



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

Προηγμένες Μεθοδολογίες Δοκιμής για Αρχιτεκτονικές Μικροεπεξεργαστών με Αριθμητική Υψηλών Επιδόσεων

Διδακτορική Διατριβή

Γεώργιος Ξενούλης



Εργαστήριο Υπολογιστικών Συστημάτων

Πειραιάς, Μάρτιος 2009



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

Προηγμένες Μεθοδολογίες Δοκιμής για Αρχιτεκτονικές Μικροεπεξεργαστών με Αριθμητική Υψηλών Επιδόσεων

Διδακτορική Διατριβή του
Γεωργίου Ξενούλη
Πτυχιούχου Τμήματος Πληροφορικής
Πανεπιστημίου Πειραιώς

Τριμελής Συμβουλευτική Επιτροπή

Δημήτριος Γκιζόπουλος, *Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς (επιβλέπων)*
Χρήστος Δουλγιέρης, *Καθηγητής Πανεπιστημίου Πειραιώς*
Γεώργιος Τσιχριντζής, *Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς*

Επταμελής Εξεταστική Επιτροπή (έγκριση Μέρα-Μήνας-Έτος)

...
Χρήστος Δουλγιέρης
Καθηγητής
ΠΑ.ΠΕΙ.

...
Αντώνιος Πασχάλης
Καθηγητής
Ε.Κ.Π.Α.

...
Κιαμάλ Πεκμεστζή
Καθηγητής
Ε.Μ.Π.

...
Δημήτριος Γκιζόπουλος
Αναπληρωτής Καθηγητής
ΠΑ.ΠΕΙ.

...
Γεώργιος Τσιχριντζής
Αναπληρωτής Καθηγητής
ΠΑ.ΠΕΙ.

...
Άγγελος Πικράκης
Λέκτορας
ΠΑ.ΠΕΙ.

...
Μιχαήλ Ψαράκης
Λέκτορας
ΠΑ.ΠΕΙ.



Εργαστήριο Υπολογιστικών Συστημάτων

Πειραιάς, Μάρτιος 2009

Μεταπτυχιακή Διατριβή

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Περίληψη

Οι εξελίξεις στην επιστήμη της μικροηλεκτρονικής οδηγούν σε τεχνολογίες κατασκευής ολοκληρωμένων κυκλωμάτων σε συνεχώς μικρότερες διαστάσεις. Οι προβλέψεις του International Technology for Roadmap of Semiconductors - ITRS μιλούν για σχεδίαση στα 10nm το 2020. Καθώς οι διαστάσεις κατασκευής μικραίνουν, εμφανίζονται συχνότερα ψεγάδια ή ατέλειες που μπορούν να επηρεάσουν όχι μόνο την ορθή λειτουργία των κυκλωμάτων αλλά και τη συχνότητα λειτουργίας και κατ' επέκταση την απόδοση ενός ορθά κατασκευασμένου κυκλώματος. Η δοκιμή των ολοκληρωμένων κυκλωμάτων είναι μια διαδικασία, που στοχεύει στην ανίχνευση των κυκλωμάτων που περιέχουν ατέλειες ή λειτουργούν σε χαμηλότερη συχνότητα, και πραγματοποιείται στο εργοστάσιο μετά την κατασκευή τους (manufacturing testing). Επιπλέον, η δοκιμή πρέπει να πραγματοποιείται και κατά τη λειτουργία του κυκλώματος σε ένα σύστημα (on-line testing) για να αντιμετωπιστούν οι νέες προκλήσεις που έκαναν την εμφάνισή τους στις νανοτεχνολογίες, όπως είναι η συνεχιζόμενη αύξηση του ρυθμού των ήπιων λαθών (soft error rate) που οφείλεται σε σωματίδια-α και στην κοσμική ακτινοβολία (νετρόνια), και η γήρανση των υλικών που οδηγεί σε όλο και μεγαλύτερη υποβάθμιση της απόδοσης του συστήματος. Η δοκιμή των μικροεπεξεργαστών παρουσιάζει τις περισσότερες προκλήσεις από κάθε άλλο κύκλωμα, διότι σε αυτούς εφαρμόζονται οι πιο σύγχρονες τεχνολογίες κατασκευής. Λαμβάνοντας υπόψη, πως κάθε ενσωματωμένο σύστημα περιέχει τουλάχιστον έναν ή περισσότερους μικροεπεξεργαστές, από απλούς επεξεργαστές των 8-bit έως ιδιαίτερα πολύπλοκους των 32/64-bit με σύγχρονες αρχιτεκτονικές (superscalar, VLIW), η δοκιμή των μικροεπεξεργαστών έχει ιδιαίτερη αξία.

Η συνεισφορά της διατριβής αυτής εντάσσεται στο χώρο της δοκιμής των μικροεπεξεργαστών που ενσωματώνουν μονάδες κινητής υποδιαστολής για την επίτευξη αριθμητικής υψηλών επιδόσεων. Στο πρώτο μέρος της διατριβής παρουσιάζεται η μεθοδολογία ανάπτυξης των συνθηκών δοκιμαστικότητας για μονάδες κινητής υποδιαστολής. Οι συνθήκες δοκιμαστικότητας (testability conditions) αποτελούνται από τις συνθήκες ελεγχιμότητας και παρατηρησιμότητας και χρησιμοποιούνται για την εξαγωγή διανυσμάτων δοκιμής για μονάδες κινητής υποδιαστολής. Παραμένουν ανεξάρτητες από την εσωτερική υλοποίηση των μονάδων κινητής υποδιαστολής και είναι προσαρμόσιμες σε κάθε ακρίβεια αναπαράστασης. Στο δεύτερο μέρος της διατριβής, παρουσιάζονται, αρχικά, τα κριτήρια για την ανάπτυξη λογισμικού αυτοδοκιμής του μικροεπεξεργαστή κατά τη λειτουργία του (μικρό ίχνος μνήμης, μικρός χρόνος εκτέλεσης και χαμηλή κατανάλωση ενέργειας), και στη συνέχεια παρουσιάζεται μια νέα μεθοδολογία ανάπτυξης αυτοδοκιμής για μονάδες κινητής υποδιαστολής σε μικροεπεξεργαστές 32/64-bit, τόσο για αναπαράσταση απλής ακρίβειας όσο και για αναπαράσταση διπλής ακρίβειας. Τέλος, πειραματικά αποτελέσματα στις πιο ευρέως διαδεδομένες μονάδες κινητής υποδιαστολής υψηλών επιδόσεων, οι οποίες χρησιμοποιούνται στους μοντέρνους μικροεπεξεργαστές (MIPS, PowerPC, Alpha), αποδεικνύουν την αποδοτικότητα της προτεινόμενης μεθοδολογίας.

Abstract

Advances in the microelectronics science lead to manufacturing technologies with smaller feature sizes. The predictions of the International Technology Roadmap for Semiconductors - ITRS mentions feature sizes of 10 nm in 2022. As feature sizes are getting smaller, defects that hinder the correct operation of a digital circuit or affect its operating frequency, thus its performance, appear more frequently. The testing of digital circuits aims at detecting defective circuits and it is carried out after the circuit is manufactured (manufacturing testing). In addition, testing must be performed simultaneously with normal circuit operation (on-line testing), in order to face the new challenges that appear with the advent of nanotechnology, such as the continuously increasing soft error rate which is attributed to α -particles and cosmic ray (neutrons), and the ageing of materials which leads to lower system performance. Among all digital circuits, the testing of microprocessors presents the most challenges as the most aggressive manufacturing is applied first at them. Given the fact that nowadays most embedded systems contain at least one or more microprocessors ranging from simple 8-bit cores to 32/64-bit ones with modern architectures (superscalar, VLIW), the testing of microprocessors becomes even more important.

This PhD thesis contributes to the field of microprocessor testing which integrates floating point units to achieve high performance arithmetic. In the first part of the thesis, we present the methodology to develop testability conditions for floating point units. Testability conditions consist of controllability and observability conditions and are used to extract test patterns for floating point units. They are scalable to every floating point precision (single, double) and they are also independent of the internal architecture of the subcomponents that comprise the floating point unit. In the second part of the thesis, we first present the criteria (small memory footprint, small execution time and low-power consumption) for the development of self-test software for on-line microprocessor testing. Furthermore, we present a new methodology for the development of self-test software for floating point units in modern 32/64-bit microprocessors for both single and double precision units. Finally, experimental results in the high performance floating point units which are used in modern microprocessors (MIPS, PowerPC, Alpha) show the efficacy of the proposed methodology.

Ευχαριστίες

Για την όμορφη αυτή αλλά και πολλές φορές επίπονη πορεία προς την ολοκλήρωση της διδακτορικής διατριβής, θα ήθελα να ευχαριστήσω τον κ. *Δημήτρη Γκιζόπουλο*, Αναπληρωτή Καθηγητή του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς για την επίβλεψη, τη βοήθεια, την εποικοδομητική συνεργασία καθώς και για την συνεχή υποστήριξη και κατανόηση του στις δύσκολες στιγμές. Επίσης, θα ήθελα να τον ευχαριστήσω για τις πολύτιμες εμπειρίες από τη συμμετοχή και παρουσίαση των εργασιών της διατριβής σε διεθνή συνέδρια.

Θα ήθελα να ευχαριστήσω τον κ. *Μιχάλη Ψαράκη*, Λέκτορα του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς για τη βοήθεια, την εποικοδομητική συνεργασία του και την προθυμία του να μοιραστεί γνώσεις και απόψεις σε κάθε προβληματισμό μου.

Επίσης, θα ήθελα να ευχαριστήσω τον κ. *Αντώνη Πασχάλη*, Καθηγητή του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών, για τη βοήθειά του και τις συμβουλές του κατά την εκπόνηση της διδακτορικής διατριβής.

Θα ήθελα ακόμα να ευχαριστήσω τον κ. *Χρήστο Δουληγέρη*, Καθηγητή του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς και τον κ. *Γεώργιο Τσιχριντζή*, Αναπληρωτή Καθηγητή του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς για την υποστήριξή τους κατά την εκπόνηση της διατριβής.

Ακόμα θα ήθελα να ευχαριστήσω τους καλούς μου φίλους, *Ηλία*, *Θανάση*, *Νίκο* και *Σωκράτη* (η σειρά είναι αλφαβητική) για την υποστήριξή τους. Ο καθένας με το δικό του τρόπο με στήριξαν σε όλη αυτή την προσπάθεια.

Η εκπόνηση της διδακτορικής διατριβής δε θα είχε ποτέ ξεκινήσει δίχως την ηθική και υλική συμπαράσταση, την υπομονή, και την εμπιστοσύνη των γονιών μου, *Διονύση* και *Γεωργίας*. Επίσης θέλω να ευχαριστήσω τον αδερφό μου, *Κωστή* για τη βοήθεια του και τη συμπαράστασή του σε αυτήν την προσπάθεια.

Γιώργος Ξενούλης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΙΑ

στο Διονύσι, τη Γεωργία, τον Κωστή
...και την Ελένη

Περιεχόμενα

1	Εισαγωγή στην Ερευνητική Περιοχή και Συνεισφορά της Διδακτορικής Διατριβής	1
2	Δοκιμή Ψηφιακών Ολοκληρωμένων Κυκλωμάτων και Μικροεπεξεργαστών	23
2.1	Τρόποι Πραγματοποίησης Δοκιμής	26
2.2	Αξιολόγηση και Ποιότητα Δοκιμής	27
2.3	Κόστος και Δυσκολίες Δοκιμής	29
2.4	Μοντέλα ελαττωμάτων	33
2.5	Ανάλυση Δοκιμαστικότητας Κυκλωμάτων	38
2.6	Σχεδίαση για Αυξημένη Δοκιμαστικότητα	41
2.7	Ενσωματωμένη Αυτοδοκιμή	48
2.8	Αυτοδοκιμή μικροεπεξεργαστών	54
2.8.1	Η έννοια της αυτοδοκιμής με ενσωματωμένο λογισμικό	58
2.8.2	Επισκόπηση της βιβλιογραφίας	63
2.9	Ανακεφαλαίωση	67
3	Αριθμητική Κινητής Υποδιαστολής	69
3.1	Βασικές Έννοιες	70
3.1.1	Ακρίβεια και Απόκλιση	76
3.2	Στρογγυλοποίηση	76
3.2.1	Στρογγυλοποίηση προς το 0 - Αποκοπή	78
3.2.2	Στρογγυλοποίηση προς τον πλησιέστερο αριθμό	78
3.2.3	Στρογγυλοποίηση προς τον πλησιέστερο άρτιο ή περιττό	79
3.2.4	Στρογγυλοποίηση προς $\pm\infty$	80
3.3	Το πρότυπο IEEE 754	81
3.3.1	Ειδικές Τιμές	83
3.3.2	Εξαιρέσεις	84
3.4	Πρόσθεση–Αφαίρεση	85
3.4.1	Υλοποίηση Μονού Μονοπατιού για Αθροιστές Κινητής Υποδιαστολής	94

3.4.2	Υλοποίηση Διπλού Μονοπατιού για Αθροιστές Κινητής Υποδιαστολής . . .	94
3.5	Πολλαπλασιασμός	100
3.6	Διαίρεση	105
3.7	Τετραγωνική Ρίζα	108
3.8	Μετατροπές	110
3.9	Ανακεφαλαίωση	112
4	Δοκιμαστικότητα & ανάπτυξη διανυσμάτων δοκιμής για μονάδες κινητής υπο-	
	διαστολής	115
4.1	Ανασκόπηση της διεθνούς βιβλιογραφίας	116
4.2	Βασικές αρχές εξαγωγής διανυσμάτων δοκιμής για μονάδες κινητής υποδιαστολής	120
4.3	Συνθήκες δοκιμαστικότητας για αθροιστές υψηλής απόδοσης διπλού μονοπατιού	121
4.3.1	Ακέραιος αθροιστής/αφαιρέτης μακρινού μονοπατιού	123
4.3.2	Ακέραιος αφαιρέτης σύντομου μονοπατιού	130
4.3.3	Ολισθητής μετά-κανονικοποίησης σύντομου μονοπατιού	132
4.3.4	Συμπλήρωμα ως προς 2 (Σύντομο Μονοπάτι)	133
4.3.5	Αυξητής στρογγυλοποίησης σημανόμενου (Σύντομο Μονοπάτι)	135
4.3.6	Αυξητής στρογγυλοποίησης σημανόμενου (Μακρινό Μονοπάτι)	136
4.3.7	Αυξητής εκθέτη - Υπερχείλιση Στρογγυλοποίησης (Μακρινό Μονοπάτι) .	137
4.3.8	Αυξητής - Μειωτής Κανονικοποίησης εκθέτη (Μακρινό Μονοπάτι)	138
4.3.9	Ολισθητής Προ-κανονικοποίησης - Γεννήτορας Ψηφίων φρουρών (Μακρινό Μονοπάτι)	139
4.4	Εφαρμογή των συνθηκών δοκιμαστικότητας	141
4.5	Πειραματικά Αποτελέσματα	146
4.5.1	Σύγκριση με άλλες τεχνικές δοκιμής	147
4.6	Ανακεφαλαίωση	151
5	Αυτοδοκιμή Μικροεπεξεργαστών με Ενσωματωμένο Λογισμικό κατά τη λειτουρ-	
	γία	153
5.1	Δοκιμή ολοκληρωμένων κυκλωμάτων κατά τη λειτουργία	154
5.2	Αυτοδοκιμή μικροεπεξεργαστών κατά τη λειτουργία με ενσωματωμένο λογισμικό	157
5.2.1	Επιθυμητά χαρακτηριστικά λογισμικού αυτοδοκιμής	160
5.3	Ανάπτυξη ενσωματωμένου λογισμικού αυτοδοκιμής	163
5.3.1	Λογισμικό Αυτοδοκιμής βασισμένο σε εργαλεία εξαγωγής διανυσμά- των δοκιμής	164
5.3.2	Λογισμικό Αυτοδοκιμής βασισμένο σε ψευδοτυχαία διανύσματα δοκιμής .	167
5.3.3	Λογισμικό Αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα δοκιμής	168

5.4	Σύμπτυξη των αποκρίσεων	170
5.5	Πειραματικά αποτελέσματα	170
5.6	Λογισμικό αυτοδοκιμής για μονάδες κινητής υποδιαστολής	177
5.6.1	Ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες απλής ακρίβειας	179
5.6.2	Ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες διπλής ακρίβειας	186
5.6.3	Πειραματικά αποτελέσματα	192
5.7	Ανακεφαλαίωση	204
6	Συμπεράσματα και Μελλοντική Έρευνα	205

РАНЕЕЗНАМО ТЕРРА

Κατάλογος σχημάτων

1.1	Ο Κανόνας των δέκα.	5
1.2	Απεικόνιση της συνεισφοράς της διδακτορικής διατριβής στην εξαγωγή διανυσμάτων δοκιμής μονάδων κινητής υποδιαστολής και ανάπτυξη λογισμικού αυτοδοκιμής για αυτές.	19
2.1	Το πείραμα της δοκιμής.	24
2.2	Ο διαχωρισμός των ολοκληρωμένων κυκλωμάτων σε ορθά και ελαττωματικά. . .	28
2.3	Παράδειγμα ελαττώματος προσκόλλησης της γραμμής d στην τιμή 1.	34
2.4	Παράδειγμα ελαττώματος προσκόλλησης της γραμμής d στην τιμή 0.	34
2.5	Παράδειγμα ελαττώματος καθυστέρησης μετάβασης $1 \rightarrow 0$ (slow-to-fall-STF) της γραμμής d	35
2.6	Παράδειγμα κυρίαρχου ελαττώματος γεφύρωσης.	37
2.7	Ανάλυση δοκιμαστικότητας ενός πλήρους αθροιστή σύμφωνα με τη μέθοδο SCOAP.	40
2.8	Παράδειγμα εφαρμογής της κατά-περίπτωση σχεδίασης για αυξημένη δοκιμαστικότητα	42
2.9	Η κυψελίδα σάρωσης Mixed D Scan Cell.	43
2.10	Ακολουθιακό κύκλωμα με αλυσίδα σάρωσης που αποτελείται από τρεις κυψελίδες σάρωσης.	44
2.11	Η λειτουργία της αλυσίδας σάρωσης κατά την εφαρμογή των διανυσμάτων δοκιμής V1, V2.	45
2.12	Η τυπική έννοια της αυτοδοκιμής.	49
2.13	Γεννίτορες ψευδο-τυχαίων διανυσμάτων δοκιμής βασισμένοι σε καταχωρητές ολίσθησης με ανάδραση.	50
2.14	Γενική μορφή του κυψελωτού αυτόματου.	51
2.15	Σειριακή ανάλυση υπογραφής.	52
2.16	Παράλληλη ανάλυση υπογραφής.	53
2.17	Δοκιμή-ανά-σάρωση με το σύστημα STUMPS.	54
2.18	Αυτοδοκιμή μικροεπεξεργαστών με ενσωματωμένο λογισμικό	59

2.19	Ανίχνευση ελαττώματος στην ALU με ενσωματωμένο λογισμικό	60
2.20	Τα διαφορετικά περιβάλλοντα δοκιμής στα οποία εφαρμόζεται το λογισμικό αυτοδοκιμής	62
3.1	Το εύρος των αριθμών κινητής υποδιαστολής.	74
3.2	Η στρογγυλοποίηση του αριθμού x	77
3.3	Η κατανομή των ψηφίων στην απλή και διπλή ακρίβεια αναπαράστασης.	82
3.4	Στρογγυλοποίηση με όλα τα ψηφία.	93
3.5	Λανθασμένη στρογγυλοποίηση με ψηφία G, T	93
3.6	Ορθή στρογγυλοποίηση με ψηφία G, R, T	93
3.7	Πρόσθεση Κινητής Υποδιαστολής – Υλοποίηση Μονού Μονοπατιού.	95
3.8	Εξαρτήσεις στην Αρχιτεκτονική Μονού και Διπλού Μονοπατιού.	97
3.9	Πρόσθεση Κινητής Υποδιαστολής – Υλοποίηση Διπλού Μονοπατιού.	98
3.10	Γραφική αναπαράσταση των χαρακτηριστικών του μεγέθους και της καθυστέρησης.	100
3.11	Υλοποίηση Πολλαπλασιασμού κινητής υποδιαστολής.	104
3.12	Βελτιστοποιημένος υπολογισμός του αθροίσματος των εκθετών (σε πολωμένη μορφή).	105
3.13	Διάρθρωση κινητής υποδιαστολής.	108
4.1	Αθροιστής/Αφαιρέτης δύο μονοπατιών	122
4.2	Ολισθητής προ-κανονικοποίησης μακρινού μονοπατιού και γέννηση ψηφίων φρουρών G, R, T	140
4.3	Εξαγωγή διανυσμάτων δοκιμής με συνθήκες δοκιμαστικότητας.	141
4.4	Σύγκριση ποσοστού κάλυψης ελαττωμάτων για το μοντέλο μόνιμης τιμής.	149
4.5	Σύγκριση ποσοστού κάλυψης ελαττωμάτων για το μοντέλο καθυστέρησης διάδοσης μετάβασης.	150
5.1	Το περιβάλλον του μικροεπεξεργαστή κατά την αυτοδοκιμή στο εργοστάσιο κατασκευής.	158
5.2	Το περιβάλλον του μικροεπεξεργαστή στο πεδίο λειτουργίας του συστήματος.	159
5.3	Τα διανύσματα δοκιμής (ATPG) σχηματίζονται με τη βοήθεια της ψευδοεντολής li (load immediate).	165
5.4	Τα διανύσματα δοκιμής (ATPG) φορτώνονται από τη μνήμη χρησιμοποιώντας την εντολή lw (load word).	166
5.5	Τα ψευδοτυχαία διανύσματα δοκιμής εξάγονται με τη βοήθεια γραμμικών καταχωρητών ολίσθησης με ανάδραση υλοποιημένους σε λογισμικό.	168
5.6	Δομή ρουτίνας αυτοδοκιμής βασισμένη σε προ-υπολογισμένα διανύσματα δοκιμής.	169

5.7	Σύγκριση του ίχνους μνήμης για τις ρουτίνες αυτοδοκιμής του πολλαπλασιαστή.	172
5.8	Σύγκριση του χρόνου εκτέλεσης των ρουτινών αυτοδοκιμής του πολλαπλασιαστή.	173
5.9	Σύγκριση του ποσοστού κάλυψης ελαττωμάτων που επιτυγχάνει κάθε ρουτίνα αυτοδοκιμής του πολλαπλασιαστή.	174
5.10	Συγκριτικά αποτελέσματα του μεγέθους των ρουτινών αυτοδοκιμής.	175
5.11	Συγκριτικά αποτελέσματα του χρόνου εκτέλεσης των ρουτινών αυτοδοκιμής. . .	175
5.12	Συγκριτικά αποτελέσματα του ποσοστού κάλυψης ελαττωμάτων των ρουτινών αυτοδοκιμής.	176
5.13	Εφαρμογή διανυσμάτων δοκιμής από εργαλείο εξαγωγής διανυσμάτων με χρήση βρόχου.	180
5.14	Εφαρμογή διανυσμάτων δοκιμής από εργαλείο εξαγωγής διανυσμάτων δοκιμής με χρήση εντολών άμεσης διευθυνσιοδότησης.	181
5.15	Η λειτουργία της ρουτίνας <code>set_sig</code> στους καταχωρητές <code>r_dst</code> και <code>r_src</code>	182
5.16	Η λειτουργία της ρουτίνας <code>set_exp</code> στους καταχωρητές <code>r_dst</code> και <code>r_src</code>	183
5.17	Η υλοποίηση των ρουτινών <code>set significand</code> και <code>set exponent</code> για αναπαράσταση απλής ακρίβειας στον επεξεργαστή MIPS.	184
5.18	Η υλοποίηση των ρουτινών <code>set significand</code> και <code>set exponent</code> για αναπαράσταση απλής ακρίβειας στον επεξεργαστή PowerPC.	184
5.19	Ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής απλής ακρίβειας, βασισμένο σε ψευδοτυχαία διανύσματα.	185
5.20	Λογισμικό αυτοδοκιμής που εφαρμόζει τα προσαρμόσιμα διανύσματα δοκιμής της ακολουθίας S2.	187
5.21	Λογισμικό αυτοδοκιμής που εφαρμόζει τα προσαρμόσιμα διανύσματα δοκιμής της ακολουθίας S3.	188
5.22	Σχηματίζοντας διανύσματα δοκιμής εξαγόμενα από αυτόματα εργαλεία (ATPG) χρησιμοποιώντας τις εντολές άμεσης διευθυνσιοδότησης. Τα διανύσματα δοκιμής είναι για μονάδες κινητής υποδιαστολής διπλής ακρίβειας σε ένα μικροεπεξεργαστή 32-bit.	189
5.23	Σχηματίζοντας ένα διάνυσμα δοκιμής 64-bit χρησιμοποιώντας εντολές άμεσης διευθυνσιοδότησης όπου το μέγεθος του άμεσου πεδίου (<code>immediate</code>) είναι 16-bit.	190
5.24	Η λειτουργία της ρουτίνας <code>set_sig</code> στους καταχωρητές <code>r_dst_high</code> , <code>r_dst_low</code> και <code>r_src_high</code> , <code>r_src_low</code>	190
5.25	MIPS 32-bit—Οι ρουτίνες <code>set significand</code> και <code>set exponent</code> για μονάδες κινητής υποδιαστολής διπλής ακρίβειας.	191
5.26	PowerPC 32-bit—Οι ρουτίνες <code>set significand</code> και <code>set exponent</code> για μονάδες κινητής υποδιαστολής διπλής ακρίβειας.	192

5.27 Alpha 64-bit—Οι ρουτίνες set significand και set exponent για μονάδες κινητής υποδιαστολής διπλής ακρίβειας.	193
5.28 Συγκριτικά αποτελέσματα ποσοστού κάλυψης ελαττωμάτων για αθροιστές-αφαιρέτες κινητής υποδιαστολής απλής και διπλής ακρίβειας.	195
5.29 Συγκριτικά αποτελέσματα ποσοστού κάλυψης ελαττωμάτων για πολλαπλασιαστές κινητής υποδιαστολής απλής και διπλής ακρίβειας.	196
5.30 Συγκριτικά αποτελέσματα χρόνου εκτέλεσης (κύκλοι) για αθροιστές-αφαιρέτες κινητής υποδιαστολής απλής και διπλής ακρίβειας.	199
5.31 Συγκριτικά αποτελέσματα χρόνου εκτέλεσης (κύκλοι) για πολλαπλασιαστές κινητής υποδιαστολής απλής και διπλής ακρίβειας.	200
5.32 Συγκριτικά αποτελέσματα κατανάλωσης ενέργειας για αθροιστές-αφαιρέτες κινητής υποδιαστολής απλής και διπλής ακρίβειας.	201
5.33 Συγκριτικά αποτελέσματα κατανάλωσης ενέργειας πολλαπλασιαστική κινητής υποδιαστολής απλής και διπλής ακρίβειας.	202

Κατάλογος πινάκων

1.1	Εφαρμογές των αριθμών κινητής υποδιαστολής	12
1.2	Συνδυασμοί επεξεργαστών και μονάδων κινητής υποδιαστολής για τους οποίους έγινε ανάπτυξη λογισμικού αυτοδοκιμής.	18
2.1	Κανόνες υπολογισμού ελεγχιμότητας.	39
2.2	Κανόνες υπολογισμού παρατηρησιμότητας.	39
3.1	Μέγιστοι και ελάχιστοι αριθμοί κινητής υποδιαστολής.	74
3.2	Αποκοπή και σφάλμα στρογγυλοποίησης.	78
3.3	Στρογγυλοποίηση προς τον πλησιέστερο αριθμό και σφάλμα στρογγυλοποίησης.	79
3.4	Στρογγυλοποίηση προς τον πλησιέστερο άρτιο και σφάλμα στρογγυλοποίησης.	80
3.5	Στρογγυλοποίηση προς $\pm\infty$ και σφάλμα στρογγυλοποίησης.	81
3.6	Τα βασικά χαρακτηριστικά της αναπαράστασης Μονής και Διπλής ακρίβειας σύμφωνα με το πρότυπο IEEE 754.	84
3.7	Υπολογισμός των Ψηφίων Φρουρών Guard (G), Round (R), Sticky (T).	87
3.8	Πραγματική Πράξη ανάλογα με τα πρόσσημα και την πράξη κινητής υποδιαστολής.	88
3.9	Πρόσθεση/Αφαίρεση ειδικών τιμών.	93
3.10	Μέγεθος και απόδοση των: πολλαπλασιαστική, αθροιστική μονού μονοπατιού και αθροιστική διπλού μονοπατιού.	99
3.11	Ειδικές Τιμές και πολλαπλασιασμός κινητής υποδιαστολής.	103
3.12	Ειδικές Τιμές και διαίρεση κινητής υποδιαστολής.	107
4.1	Πειραματικά αποτελέσματα του συστήματος Socrates για την δοκιμή των πολλαπλασιαστών κινητής υποδιαστολής.	116
4.2	Οι λύσεις της ισότητας $E = E_{faulty}$	127
4.3	Τα ζεύγη διανυσμάτων δοκιμής για το ζεύγος πυλών που σχηματίζουν το ψηφίο K_3	145
4.4	Αποτελέσματα σύνθεσης στη βιβλιοθήκη TSMC 0.18um.	146
4.5	Στατιστικά στοιχεία σχετικά με το πλήθος των ελαττωμάτων.	147

4.6	Ποσοστά κάλυψης ελαττωμάτων μόνιμης τιμής (stuck-at) και καθυστέρησης μετάβασης (transition delay) για αθροιστές κινητής και διπλής ακρίβειας (DP-FPAdder).	148
4.7	Ποσοστά κάλυψης ελαττωμάτων των ψευδοτυχαίων διανυσμάτων δοκιμής.	149
4.8	Συγκριτικά αποτελέσματα μεθόδων δοκιμής	150
5.1	Τα χαρακτηριστικά του λογισμικού αυτοδοκιμής ανάλογα με τον τρόπο ανάπτυξής του.	172
5.2	Στατιστικά στοιχεία λογισμικού αυτοδοκιμής του μικροεπεξεργαστή Plasma.	177
5.3	Συνδυασμοί επεξεργαστών και μονάδων κινητής υποδιαστολής για τους οποίους έγινε ανάπτυξη λογισμικού αυτοδοκιμής.	179
5.4	Σύνολο διανυσμάτων δοκιμής για τον αφαιρέτη στο σύντομο μονοπάτι του αθροιστή/αφαιρέτη κινητής υποδιαστολής με αρχιτεκτονική υλοποίησης δύο μονοπατιών.	185
5.5	Πλήθος εντολών που απαιτούνται για την υλοποίηση των ρουτινών set_sig και set_exp στους επεξεργαστές MIPS, PowerPC και Alpha.	192
5.6	Στατιστικά σύνθεσης για αθροιστές και πολλαπλασιαστές κινητής υποδιαστολής απλής και διπλής ακρίβειας.	194
5.7	Το πλήθος των διανυσμάτων δοκιμής.	196
5.8	Μέγεθος λογισμικού αυτοδοκιμής (byte).	197
5.9	Διευθετήσεις συστήματος κρυφής μνήμης.	198
5.10	Συμπεράσματα από τα πειραματικά αποτελέσματα.	203

Κεφάλαιο 1

Εισαγωγή στην Ερευνητική Περιοχή και Συνεισφορά της Διδακτορικής Διατριβής

Πάντοτε, η ανθρώπινη φύση επεδίωκε την ανάπτυξη μηχανών για την επεξεργασία δεδομένων και εξαγωγή χρήσιμων πληροφοριών. Σήμερα, η ποσότητα της πληροφορίας που καλείται να επεξεργαστεί ένας άνθρωπος είναι μεγαλύτερη από ποτέ. Έτσι, η σύγχρονη ζωή των ανθρώπων χαρακτηρίζεται από την καθημερινή επαφή τους με ένα μεγάλο σύνολο ηλεκτρονικών συσκευών για την επεξεργασία των δεδομένων στο κέντρο των οποίων βρίσκεται ένας ή και περισσότεροι μικροεπεξεργαστές (microprocessors). Πολλές φορές γίνεται λόγος για την έννοια της διάχυτης υπολογιστικής (pervasive computing) όπου οι έννοιες του υπολογισμού και της επεξεργασίας δεδομένων βρίσκονται πίσω σχεδόν από κάθε δραστηριότητα του ανθρώπου. Συνάμα, γίνεται λόγος και για τα ενσωματωμένα συστήματα (embedded systems), που επιτρέπουν την υλοποίηση της διάχυτης υπολογιστικής. Τα ενσωματωμένα συστήματα αποτελούνται από το υλικό (hardware) το οποίο διαθέτει ένα τουλάχιστον μικροεπεξεργαστή και το λογισμικό (software), κατάλληλα συνδυασμένα για την επεξεργασία δεδομένων και εξαγωγή χρήσιμων πληροφοριών ώστε να καλύπτουν τις απαιτήσεις μιας συγκεκριμένης εφαρμογής. Μια σημαντική ένδειξη για το βαθμό της διείσδυσης των μικροεπεξεργαστών στη ζωή του ανθρώπου αποτελεί η απάντηση στην ακόλουθη ερώτηση, *Πόσους μικροεπεξεργαστές χρησιμοποιείτε καθημερινά;* Αν και οι περισσότεροι θα απαντούσαν πως χρησιμοποιούν συνολικά 20 ή 30 μικροεπεξεργαστές καθημερινά, η πραγματικότητα είναι διαφορετική καθώς ένας άνθρωπος έρχεται καθημερινά σε επαφή με περισσότερους από 100 μικροεπεξεργαστές [1]. Ένα σύγχρονο αυτοκίνητο περιλαμβάνει τουλάχιστον 40 μικροεπεξεργαστές. Διάφορες οικογένειες μικροεπεξεργαστών ποικίλων μεγεθών (8, 16, 32 ακόμα και 64-bit) χρησιμοποιούνται

σε πλήθος εφαρμογές ακόμα και σε σημεία που πριν μερικά χρόνια θα θεωρούνταν αδιανόητη η παρουσία των μικροεπεξεργαστών, όπως στην ηλεκτρική οδοντόβουρτσα όπου έχει χρησιμοποιηθεί ο μικροεπεξεργαστής Zilog Z8 των 8-bit, ή στα σύγχρονα πλυντήρια όπου έχει βρει εφαρμογή ο μικροεπεξεργαστής των 8-bit 68HC05 της Motorola [2]. Τα συστήματα πλοήγησης που όλο και πιο συχνά ενσωματώνονται στα αυτοκίνητα, χρησιμοποιούν μικροεπεξεργαστές των 16-bit, ενώ σε συσκευές ψυχαγωγίας, όπως η μηχανή αναπαραγωγής μουσικής RIO800 MP3 Player της εταιρείας SONICblue χρησιμοποιεί μικροεπεξεργαστή των 32-bit. Ακόμα και επεξεργαστές των 64-bit έχουν χρησιμοποιηθεί σε συσκευές ψυχαγωγίας, όπως στο γνωστό σκύλο robot - AIBO της Sony, το οποίο χρησιμοποιεί ένα μικροεπεξεργαστή MIPS των 64-bit. Οι υψηλότερες απαιτήσεις των χρηστών των συσκευών οδηγούν προς τη χρήση συστημάτων τα οποία διαθέτουν περισσότερους από έναν μικροεπεξεργαστές. Τα νέα κινητά τηλέφωνα τρίτης γενιάς-(3rd Generation-3G) συνήθως ενσωματώνουν δύο επεξεργαστές, έναν επεξεργαστή γενικού σκοπού και έναν επεξεργαστή ψηφιακού σήματος (Digital Signal Processor - DSP) για να καλύψουν τις ανάγκες της επικοινωνίας και των εφαρμογών πολυμέσων. Η νέα κονσόλα παιχνιδιών PlayStation3 χρησιμοποιεί τον επεξεργαστή Cell της IBM που ενσωματώνει ένα επεξεργαστή PowerPC των 64-bit και οχτώ συνεργατικούς επεξεργαστές (synergistic processors), που είναι επιφορτισμένοι με τη μαζική επεξεργασία πραγματικών αριθμών και πολυμεσικών δεδομένων [3]. Φυσικά, η απαρίθμηση των συσκευών που διαθέτουν έναν ή και περισσότερους μικροεπεξεργαστές και με τις οποίες ο άνθρωπος αλληλεπιδρά καθημερινά δεν έχει τέλος.

Όλα αυτά τα τεχνολογικά επιτεύγματα οφείλουν την ύπαρξή τους, κατά κύριο λόγο, στις αλματώδεις εξελίξεις στην τεχνολογία των ημιαγωγών (semiconductors) η οποία με τη σειρά της οφείλει την ανάπτυξή της στις επιστήμες της φυσικής, της χημείας και ειδικότερα της επιστήμης υλικών και της μικροηλεκτρονικής. Χάρη στην ανάπτυξη όλων αυτών των επιστημών είναι δυνατή η τοποθέτηση εκατομμυρίων τρανζίστορ σε ένα και μόνο κύκλωμα. Η δυνατότητα ενσωμάτωσης των τρανζίστορ εκφράζεται με την κλίμακα ολοκλήρωσης (scale of integration-SI) των τρανζίστορ σε ένα ολοκληρωμένο κύκλωμα (Integrated Circuit-IC). Ξεκινώντας από την ενσωμάτωση μερικών δεκάδων τρανζίστορ (Small SI - SSI), στη συνέχεια μερικών εκατοντάδων (Medium SI - MSI) και έπειτα μερικών χιλιάδων τρανζίστορ (Large SI - LSI), σήμερα, γίνεται λόγος για κυκλώματα με πολύ υψηλή κλίμακα ολοκλήρωσης (Very Large SI - VLSI) όπου εκατοντάδες χιλιάδες τρανζίστορ τοποθετούνται σε ένα κύκλωμα. Βέβαια, ακόμα και ο ορισμός αυτός είναι παρωχημένος καθώς τη δεκαετία του 1990 ο πρώτος μικροεπεξεργαστής Pentium της Intel ολοκλήρωνε ήδη τρία εκατομμύρια τρανζίστορ [4]. Η αλματώδης πρόοδος στην τεχνολογία των ημιαγωγών έχει επιτύχει τη μείωση των φυσικών διαστάσεων του τρανζίστορ, την αύξηση στην ταχύτητα εναλλαγής της κατάστασής του (state switching) αλλά και τη μείωση της κατανάλωσης ενέργειάς του και συνολικά έχει μειώσει τις αποστάσεις των στοιχείων στο ολοκληρωμένο κύκλωμα. Τα τελευταία νέα από τη βιομηχανία των ημιαγωγών αναφέρονται στη σχεδίαση κυκλωμάτων σε διαστάσεις των 32 νανομέτρων (1nm

= 10^{-9} m), ενώ οι μελλοντικές προβλέψεις μιλούν για σχεδίαση στα 10 νανόμετρα το 2022 [5]. Ο πρώτος επεξεργαστής Pentium ήταν κατασκευασμένος σε τεχνολογία των 0,8μm ή 800 nm ($1\mu\text{m}=10^{-6}$ m), ενώ σήμερα ένας επεξεργαστής με δύο πυρήνες (cores) κατασκευάζεται στα 45 nm. Άμεση συνέπεια της ραγδαίας μείωσης των γεωμετρικών διαστάσεων ήταν η ενσωμάτωση τόσο μεγάλης ποσότητας υλικού (επεξεργαστές, μνήμες, στοιχεία εισόδου-εξόδου) ώστε να καταστεί δυνατή η υλοποίηση ενός συστήματος σε ένα μόνο ολοκληρωμένο κύκλωμα (System-on-Chip) ανοίγοντας με αυτό τον τρόπο νέους ορίζοντες στην ανάπτυξη προϊόντων με πολυδιάστατες δυνατότητες. Η σχεδίαση και η κατασκευή των ολοκληρωμένων κυκλωμάτων, των μικροεπεξεργαστών και των συστημάτων σε ένα κύκλωμα είναι μια διαδικασία που χαρακτηρίζεται από υψηλό βαθμό πολυπλοκότητας. Κατά τη διάρκεια της σχεδίασης και κατασκευής κάθε ολοκληρωμένου κυκλώματος ένα πλήθος παραγόντων επηρεάζει την όλη διαδικασία και καθορίζει το χρονικό διάστημα από τη σύλληψη της αρχικής ιδέας μέχρι την εμφάνιση του προϊόντος στην αγορά (time-to-market) με άμεσο αποτέλεσμα την αποτυχία ή την επιτυχία του προϊόντος.

Τα σφάλματα κατά το στάδιο της σχεδίασης αποτελούν σημαντικό παράγοντα που μπορεί να επηρεάσει την αποτυχία ή την επιτυχία ενός προϊόντος. Χαρακτηριστικότερο των σχεδιαστικών σφαλμάτων αποτελεί το διάσημο πλέον σφάλμα στη σχεδίαση της μονάδας κινητής υποδιαστολής του Pentium, το 1994, όπου η διαίρεση μεταξύ δύο αριθμών κινητής υποδιαστολής παράγαγε λανθασμένη ακρίβειας αποτελέσματα [6]. Τα σχεδιαστικά σφάλματα αντιμετωπίζονται στο στάδιο της επαλήθευσης της σχεδίασης (design verification), όπου εφαρμόζονται διάφορα σενάρια κατά την προσομοίωση του κυκλώματος (simulation based verification). Καθώς όμως η πολυπλοκότητα των κυκλωμάτων συνεχώς αυξάνει, η αποτελεσματικότητα της επαλήθευσης της σχεδίασης με βάση την προσομοίωση μειώνεται. Την αδυναμία αυτή καλύπτουν οι τυπικές μέθοδοι για την επαλήθευση της σχεδίασης (formal design verification) οι οποίες βασίζονται σε μαθηματικά μοντέλα και μπορούν να αποδείξουν αν η σχεδίαση ενός κυκλώματος ικανοποιεί τις προδιαγραφές που έχουν τεθεί.

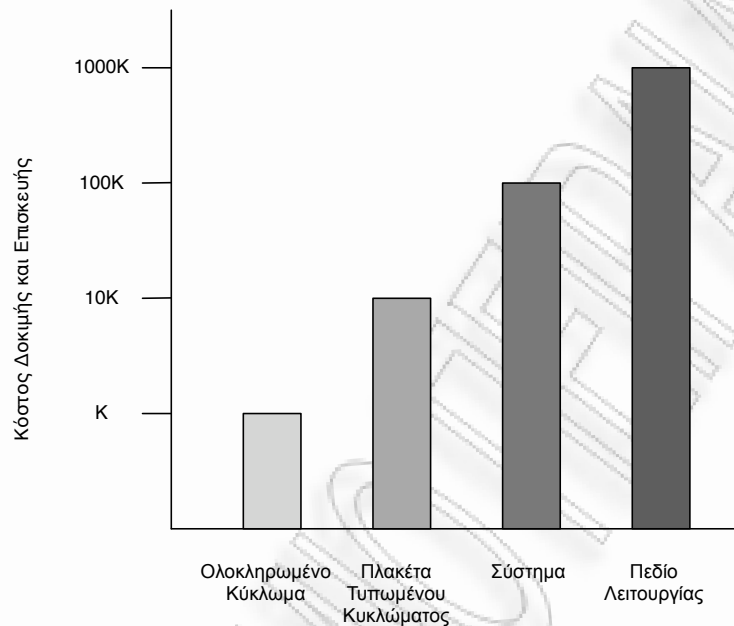
Πέρα από τα σχεδιαστικά σφάλματα που οφείλονται στον άνθρωπο, οι *ατέλειες* ή *ψεγάδια* (*defects* ή *physical faults*) που εμφανίζονται κατά τη διαδικασία της κατασκευής των κυκλωμάτων μπορεί να υπονομεύσουν την λειτουργία ενός ορθά σχεδιασμένου κυκλώματος. Η εμφάνιση των ατελειών οφείλεται κυρίως στην ατελή διαδικασία κατασκευής όπου για παράδειγμα μία λανθασμένα τυπωμένη ηλεκτρική διασύνδεση, ή η επικάλυψη ενός σωματιδίου σκόνης, ή ένα ελαττωματικό τρανζίστορ μπορούν να προκαλέσουν την εσφαλμένη λειτουργία του κυκλώματος. Ψεγάδια μπορεί να προκύψουν και εξαιτίας περιβαλλοντικών παραγόντων αλλά και λόγω της φυσικής γήρανσης των υλικών [7]. Βέβαια, καθώς οι επιστήμες της φυσικής, της χημείας και των υλικών εξελίσσονται παρέχουν όλο και αριότερα υλικά για την κατασκευή των ολοκληρωμένων κυκλωμάτων. Έτσι θα περίμενε κανείς το πλήθος των ατελειών στα ολοκληρωμένα κυκλώματα να μειώνεται διαρκώς. Όμως όχι μόνο δεν παρατηρείται κάποια τέτοια μείωση, αλλά αντιθέτως εμφανίζονται και νέοι τύποι ατελειών. Οι αιτίες στις οποίες

οφείλεται η εμφάνιση των νέων τύπων ατελειών είναι οι ίδιες που οδήγησαν στην υψηλή ολοκλήρωση πολλών τρανζίστορ σε ένα ολοκληρωμένο κύκλωμα. Έτσι νέα ψεγάδια εμφανίζονται καθώς η τεχνολογία κατασκευής ολοκληρωμένων κυκλωμάτων και μικροεπεξεργαστών οδεύει στις πάρα πολύ μικρές διαστάσεις της τάξης των λίγων δεκάδων νανομέτρων. Παράλληλα, τα ολοκληρωμένα κυκλώματα γίνονται όλο και περισσότερο ευάλωτα σε περιβαλλοντικούς και εξωγενείς παράγοντες, όπως η κοσμική ακτινοβολία. Τα σωματίδια- α (α -particles) μπορεί να επηρεάσουν τη λειτουργία των ολοκληρωμένων κυκλωμάτων αλλάζοντας την τιμή ενός στοιχείου μνήμης [8]. Οι ατέλειες ακόμα και αν επιδρούν σε ένα και μόνο τρανζίστορ, είναι ικανές να επηρεάσουν τη λειτουργία των ολοκληρωμένων κυκλωμάτων είτε αλλοιώνοντας τον ορθό τρόπο επεξεργασίας των δεδομένων είτε μειώνοντας την απόδοσή τους (μειωμένη συχνότητα λειτουργίας).

Αναπόσπαστο μέρος της κατασκευής των ολοκληρωμένων κυκλωμάτων αποτελεί το στάδιο της δοκιμής τους. Η δοκιμή των ολοκληρωμένων κυκλωμάτων (IC testing) στοχεύει στην ανίχνευση ατελειών που πιθανόν να υπάρχουν σε ένα ολοκληρωμένο κύκλωμα και τα οποία επηρεάζουν την ορθή λειτουργία του ή την απόδοσή του. Τα ολοκληρωμένα κυκλώματα που χαρακτηρίζονται ελαττωματικά χρησιμοποιούνται στο στάδιο της διάγνωσης των ελαττωμάτων (fault diagnosis), όπου εξετάζονται οι αιτίες που οδήγησαν στην εμφάνιση των ατελειών. Με αυτό τον τρόπο, η διάγνωση συνεισφέρει στη βελτιστοποίηση της διαδικασίας κατασκευής έτσι ώστε να αποφεύγονται ή έστω να περιορίζονται οι κατασκευαστικές ατέλειες.

Είναι ευνόητο πως η δοκιμή των ολοκληρωμένων κυκλωμάτων εξασφαλίζει την υψηλή αξιοπιστία του προϊόντος και είναι όχι μόνο επιθυμητή αλλά και επιβεβλημένη τόσο από την πλευρά του κατασκευαστή όσο και από την πλευρά του καταναλωτή του εκάστοτε προϊόντος. Τα ποσά που δαπανώνται από τις εταιρείες κατασκευής ολοκληρωμένων κυκλωμάτων κατά το στάδιο της δοκιμής τους είναι ιδιαίτερος υψηλά και μπορεί να φθάσουν ακόμα και το 70% του τελικού κόστους του προϊόντος. Ενδεικτικά σημειώνουμε πως η Intel αναφέρει ότι ο συνδυασμός της επαλήθευσης του σχεδιασμού και της δοκιμής αποτελούν το κύριο κόστος ενός νέου προϊόντος και όχι η αξία 2 δισεκατομμυρίων δολαρίων γραμμί παραγωγής [9]. Οι υψηλές αυτές δαπάνες είναι απολύτως δικαιολογημένες αν λάβει κάποιος υπόψη του τον κανόνα των δέκα (*rule of ten*) [10], ο οποίος περιγράφει το κόστος δοκιμής και επισκευής (test and repair) των ολοκληρωμένων κυκλωμάτων κατά τη διάρκεια της ζωής τους και απεικονίζεται στην Εικόνα 1.1.

Ο κανόνας των δέκα δηλώνει πως αν το κόστος για τη δοκιμή και διόρθωση του ολοκληρωμένου κυκλώματος είναι K , τότε το κόστος για δοκιμή και επισκευή του ολοκληρωμένου κυκλώματος όταν αυτό έχει τοποθετηθεί σε μια κάρτα-πλακέτα είναι $10 \times K$. Αντίστοιχα, το κόστος δοκιμής και επισκευής του κυκλώματος όταν έχει τοποθετηθεί στο σύστημα είναι $100 \times K$, ενώ όταν το κύκλωμα βρίσκεται στο πεδίο λειτουργίας του, όπου και θα περάσει το μεγαλύτερο μέρος της ζωής του, το κόστος για τη δοκιμή και επισκευή είναι ίσο με $1000 \times K$. Το κόστος της δοκιμής και επισκευής των ολοκληρωμένων κυκλωμάτων αυξάνε-



Εικόνα 1.1: Ο Κανόνας των δέκα.

ται δραματικά καθώς το κύκλωμα διανύει τα στάδια χρήσης του εξαιτίας της δυσκολότερης προσπέλασής του όταν πλέον αυτό βρίσκεται ενσωματωμένο σε ένα πολύπλοκο σύστημα. Το πρόβλημα αυτό γίνεται πιο έντονο όταν το ολοκληρωμένο κύκλωμα είναι ένας μικροεπεξεργαστής, διότι οι μικροεπεξεργαστές αποτελούν κομβικά σημεία σε ένα πολύπλοκο σύστημα και καθώς πλασιώνονται από ένα μεγάλο πλήθος κυκλωμάτων η απομόνωσή τους ώστε να δοκιμαστούν γίνεται ακόμα δυσκολότερη. Γενικότερα, η διόγκωση του προβλήματος σήμερα είναι τόσο σημαντική ώστε έχει προταθεί η αναπροσαρμογή του κανόνα των δέκα σε κανόνα των είκοσι (rule of twenty) εξαιτίας της αυξανόμενης πολυπλοκότητας των συστημάτων.

Είναι εύκολα αντιληπτό πως η αποτελεσματική δοκιμή των ολοκληρωμένων κυκλωμάτων και των μικροεπεξεργαστών στα πρώτα στάδια της ζωής τους, δηλαδή διπλαδί στο εργοστάσιο αμέσως μετά την κατασκευή τους (IC manufacturing testing), αυξάνει την αξιοπιστία (reliability) τους και αποφέρει σημαντικά οικονομικά οφέλη αφού μειώνει το κόστος για τη δοκιμή και διόρθωση στο πεδίο λειτουργίας του κυκλώματος. Όμως τα οφέλη από την αποτελεσματική δοκιμή δεν περιορίζονται στο πεδίο της οικονομίας. Σε εφαρμογές που απαιτούν υψηλή αξιοπιστία, όπως αυτόματα συστήματα βιομηχανικού ελέγχου, συστήματα που σχετίζονται με τα μέσα μαζικής μεταφοράς (αεροπλάνα, τρένα, αυτοκίνητα), στρατιωτικές και διαστημικές εφαρμογές κλπ, η χρήση ολοκληρωμένων κυκλωμάτων που περιέχουν ελαττώματα ή μπορεί να εμφανίσουν στο μέλλον, μπορεί να οδηγήσει ακόμα και σε απώλειες ανθρώπινων ζωών. Τα πολλαπλά οφέλη από τη δοκιμή των ολοκληρωμένων κυκλωμάτων έχουν από καιρό αναγνω-

ριστεί από τις εταιρείες κατασκευής κυκλωμάτων. Η έκφραση *κανένα κύκλωμα δε σχεδιάζεται αν δεν έχει ληφθεί μέριμνα για τη δοκιμή του (you cannot design it if you cannot test it)* είναι ενδεικτική της υψηλής σημασίας της δοκιμής.

Η δοκιμή των ολοκληρωμένων κυκλωμάτων πρέπει να πραγματοποιείται και όταν το ολοκληρωμένο κύκλωμα ή ο μικροεπεξεργαστής βρίσκονται στο πεδίο λειτουργίας τους (field testing). Στην περίπτωση αυτή, η δοκιμή μπορεί να εκτελείται περιοδικά όταν το σύστημα είναι αδρανές ή μπορεί να εκτελείται παράλληλα με την λειτουργία του συστήματος. Εξωγενείς παράγοντες, όπως η κοσμική ακτινοβολία, το περιβάλλον λειτουργίας (αυξημένη θερμοκρασία, υγρασία) αλλά και η φυσική γήρανση (ageing) των υλικών που οδηγεί στη φθορά τους (wear out), μπορούν να δημιουργήσουν ατέλειες σε ένα κύκλωμα το οποίο έχει δοκιμαστεί επιτυχώς προτού φύγει από το εργοστάσιο κατασκευής. Η δοκιμή ενός κυκλώματος στο πεδίο της λειτουργίας μπορεί να εξασφαλίσει την ορθή λειτουργία του συστήματος αποφεύγοντας την επεξεργασία λανθασμένων δεδομένων που είναι να δυνατόν να παραχθούν από ένα ελαττωματικό κύκλωμα. Ειδικότερα δε σε εφαρμογές που απαιτούν υψηλή αξιοπιστία και όπου είναι απαραίτητη η αδιάλειπτη λειτουργία του συστήματος λαμβάνεται μέριμνα ώστε το ολοκληρωμένο κύκλωμα να έχει την ικανότητα όχι μόνο να ανιχνεύει τα ελαττώματα αλλά και να διορθώνει τα λάθη που προκύπτουν εξαιτίας των ελαττωμάτων αυτών. Τα συστήματα αυτά χαρακτηρίζονται ως ανεκτικά στα ελαττώματα (fault tolerant systems).

Η δοκιμή των ολοκληρωμένων κυκλωμάτων πραγματοποιείται με την βοήθεια του αυτόματου εξοπλισμού δοκιμής ολοκληρωμένων κυκλωμάτων (Automatic Test Equipment ATE ή IC Testers). Ο αυτόματος εξοπλισμός δοκιμής εφαρμόζει στο ολοκληρωμένο κύκλωμα ένα σύνολο *διανυσμάτων δοκιμής (test vector set)* και στη συνέχεια *συγκρίνει* τις αποκρίσεις του κυκλώματος με τις αναμενόμενες τιμές που έχουν αποθηκευτεί στη μνήμη του¹. Οποιαδήποτε διαφορά μεταξύ των αναμενόμενων τιμών και των πραγματικών αποκρίσεων του ολοκληρωμένου κυκλώματος σημαίνει πως το κύκλωμα περιέχει ελαττώματα και είναι ακατάλληλο για χρήση. Τα διανύσματα δοκιμής εφαρμόζουν κατάλληλες τιμές στις εισόδους του ολοκληρωμένου κυκλώματος με στόχο την *ενεργοποίηση του ελαττώματος (fault activation)* και τη *διάδοση της λανθασμένης τιμής ή των λανθασμένων τιμών (fault effect propagation)* που προκύπτουν από την παρουσία του ελαττώματος, σε κάποιες από τις εξόδους του κυκλώματος τις οποίες και παρακολουθεί ο εξοπλισμός δοκιμής. Η διαδικασία εξαγωγής των κατάλληλων τιμών εισόδου ονομάζεται *γέννηση διανυσμάτων δοκιμής (test pattern generation-TPG)* και όταν αυτή πραγματοποιείται με τη βοήθεια εργαλείων λογισμικού καλείται *αυτόματη γέννηση διανυσμάτων δοκιμής (automatic test pattern generation-ATPG)*. Εξαιτίας του απαγορευτικά μεγάλου αριθμού φυσικών ελαττωμάτων η γέννηση των διανυσμάτων δοκιμής διενεργείται βάσει κάποιου μοντέλου ελαττώματος (fault model). Τα μοντέλα ελαττωμάτων προσπαθούν να μοντελοποιήσουν την πραγματική συμπεριφορά των φυσικών ελαττωμάτων αντιστοιχίζοντας τα φυσικά ελαττώματα σε λογικά ελαττώματα (logical faults). Το σύνολο των λογικών ελαττωμάτων είναι

¹Οι αναμενόμενες τιμές προκύπτουν από την προσομοίωση της σχεδίασης του κυκλώματος.

σημαντικά μικρότερο από το πλήθος των φυσικών ελαττωμάτων και η συμπεριφορά των λογικών ελαττωμάτων είναι σαφώς ορισμένη. Γενικότερα, όσο καλύτερα ένα μοντέλο εκφράζει την συμπεριφορά των φυσικών ελαττωμάτων και όσο περισσότερο μειώνει την πολυπλοκότητα της γέννησης των διανυσμάτων δοκιμής τόσο μεγαλύτερη επιτυχία και αποδοχή έχει. Τα ζητούμενα χαρακτηριστικά συγκρούονται γι' αυτό το λόγο και η ανάπτυξη μοντέλων ελαττωμάτων αποτελεί μια δύσκολη διαδικασία. Κανένα μοντέλο δεν είναι δυνατό να μοντελοποιήσει πλήρως τη συμπεριφορά των φυσικών ελαττωμάτων. Έτσι έχουν αναπτυχθεί διάφορα μοντέλα για την αναπαράσταση των φυσικών ελαττωμάτων. Η υιοθέτηση του μοντέλου ελαττωμάτων για την παραγωγή των διανυσμάτων σχετίζεται άμεσα με την τεχνολογία με την οποία κατασκευάζεται ένα κύκλωμα. Η τάση που επικρατεί σήμερα στη βιομηχανία κατασκευής κυκλωμάτων είναι πως σε κάθε κύκλωμα πρέπει να εφαρμόζονται διανύσματα δοκιμής που έχουν εξαχθεί βάσει αρκετών διαφορετικών μοντέλων ελαττωμάτων. Με αυτόν τον τρόπο εξασφαλίζεται η μεγαλύτερη δυνατή ανίχνευση φυσικών ελαττωμάτων και αυξάνεται η αποτελεσματικότητα της δοκιμής με αποτέλεσμα την κατασκευή κυκλωμάτων υψηλής αξιοπιστίας.

Παρόλο που η διαδικασία της δοκιμής έπεται της φάσης της σχεδίασης οι βιομηχανίες κατασκευής διαπίστωσαν πως η εξαγωγή των διανυσμάτων δοκιμής είναι ευκολότερη εάν έχει πραγματοποιηθεί ανάλυση της δοκιμαστικότητας (testability analysis) του κυκλώματος κατά το στάδιο της σχεδίασης. Η δοκιμαστικότητα (testability) εκφράζει την ευκολία με την οποία είναι δυνατό να ανιχνευτούν τα ελαττώματα του ολοκληρωμένου κυκλώματος και εκφράζεται με δύο επιμέρους έννοιες, την ελεγχσιμότητα (controllability) και την παρατηρησιμότητα (observability). Η ελεγχσιμότητα εκφράζει την ευκολία με την οποία μπορούν να εφαρμοστούν τα απαραίτητα διανύσματα δοκιμής για την ενεργοποίηση του ελαττώματος. Αντίστοιχα, η παρατηρησιμότητα εκφράζει την ευκολία με την οποία μπορούν να παρατηρηθούν στις εξόδους του κυκλώματος οι λανθασμένες τιμές που προκύπτουν εξαιτίας της ύπαρξης του ελαττώματος. Γενικότερα, όσο πιο βαθιά ενσωματώνεται ένα κύκλωμα σε ένα άλλο, τόσο μικρότερη είναι η ελεγχσιμότητα και η παρατηρησιμότητά του, άρα και η δοκιμαστικότητά του. Εάν διαπιστωθεί πως το κύκλωμα παρουσιάζει χαμηλή δοκιμαστικότητα, τότε εφαρμόζονται συγκεκριμένες τεχνικές κατά το στάδιο της σχεδίασης ώστε να αυξηθεί η δοκιμαστικότητά του. Αυτές είναι γνωστές ως τεχνικές σχεδίασης για αυξημένη δοκιμαστικότητα (*Design for Testability techniques*). Χαρακτηριστικά κυκλώματα τα οποία είναι χαμηλής δοκιμαστικότητας είναι εκείνα τα οποία περιέχουν στοιχεία μνήμης (flip-flop), δηλαδή τα ακολουθιακά (sequential) κυκλώματα. Η εξαγωγή διανυσμάτων δοκιμής είναι ιδιαίτερα δύσκολη για τα ακολουθιακά κυκλώματα εξαιτίας του εκθετικού πλήθους των καταστάσεων στις οποίες μπορεί να βρεθεί το κύκλωμα. Μια από τις σημαντικότερες μεθόδους για την αύξηση της δοκιμαστικότητας είναι εκείνη της ενσωματωμένης αυτοδοκιμής (Built-In Self-Test, BIST) όπου στο ολοκληρωμένο κύκλωμα προστίθεται επιπλέον υλικό για τη γέννηση και την εφαρμογή των διανυσμάτων δοκιμής αλλά και για την παρατήρηση των αποκρίσεων. Στην πραγματικότητα το επιπλέον υλικό παίζει το ρόλο του συστήματος δοκιμής, με αποτέλεσμα ο εξοπλισμός δοκιμής να ενσωματώνεται στο κύκλωμα.

Η ενσωμάτωση αυτή παρέχει δύο σημαντικά πλεονεκτήματα. Πρώτον, είναι δυνατό να γίνει δοκιμή στο πεδίο λειτουργίας του κυκλώματος (είτε κατά την εκκίνηση είτε στα αδρανή διαστήματα) αφού τώρα ο εξοπλισμός δοκιμής βρίσκεται μέσα στο κύκλωμα και μπορεί να χρησιμοποιηθεί πολλές φορές. Δεύτερον, και εξίσου σημαντικό, είναι δυνατό να πραγματοποιηθεί η δοκιμή στην πραγματική ταχύτητα λειτουργίας του κυκλώματος (at-speed testing). Πειραματικές μετρήσεις [11], [12], [13] έχουν αποδείξει πως όταν η δοκιμή πραγματοποιείται στην πραγματική συχνότητα λειτουργίας του κυκλώματος ανιχνεύονται ελαττώματα που δεν είναι δυνατό να ανιχνευθούν από τον εξοπλισμό δοκιμής όταν αυτός εφαρμόζει τα διανύσματα δοκιμής σε χαμηλότερη συχνότητα από τη συχνότητα λειτουργίας του κυκλώματος. Το κόστος των συστημάτων δοκιμής με δυνατότητα εφαρμογής διανυσμάτων δοκιμής και παρατήρησης των αποκρίσεων σε υψηλή συχνότητα είναι ιδιαίτερα υψηλό σε σημείο που μπορεί να χαρακτηριστεί ως και απαγορευτικό για κάποιες εταιρείες κατασκευής κυκλωμάτων. Ενδεικτικό του υψηλού κόστους ενός συστήματος δοκιμής υψηλής ταχύτητας είναι πως το κόστος ενός ακροδέκτη (pin) που μπορεί να εφαρμόζει διανύσματα δοκιμής και να παρατηρεί αποκρίσεις στην ταχύτητα του 1 GHz ανέρχονταν το έτος 2000 στα 3000 δολάρια. Τα συστήματα δοκιμής υψηλής ταχύτητας μπορεί να διαθέτουν μέχρι και 1000 ακροδέκτες ώστε να χρησιμοποιηθούν για τη δοκιμή ενός μεγάλου εύρους ολοκληρωμένων κυκλωμάτων. Το 2003, το κόστος ενός ακροδέκτη εξοπλισμού δοκιμής για ένα μοντέρνο μικροεπεξεργαστή κυμαινόνταν μεταξύ 2.500 και 7000 δολαρίων [14]. Σήμερα, καθώς ο αυτόματος εξοπλισμός δοκιμής περιλαμβάνει ακροδέκτες διαφορετικών τύπων για διαφορετικά είδη δοκιμής, το κόστος ανά ακροδέκτη δεν είναι ενδεικτικό του κόστους του εξοπλισμού [5]. Βέβαια αυτό δε σημαίνει πως ο εξοπλισμός δοκιμής έγινε οικονομικότερος.

Λαμβάνοντας παράλληλα υπόψη το γεγονός πως πολλά ολοκληρωμένα κυκλώματα αλλά και μικροεπεξεργαστές λειτουργούν σε πολύ μεγαλύτερες ταχύτητες από εκείνες του εξωτερικού συστήματος δοκιμής και επομένως δεν είναι δυνατό να δοκιμαστούν στη μέγιστη συχνότητα λειτουργίας τους, αντιλαμβάνεται κανείς πως η αξία της ενσωματωμένης αυτοδοκιμής είναι πολύ μεγάλη.

Όμως, η ενσωματωμένη αυτοδοκιμή έχει και την αρνητική της πλευρά, την ανάγκη για ενσωμάτωση επιπλέον υλικού για την αύξηση της δοκιμαστικότητας. Το επιπλέον υλικό σημαίνει πως το νέο κύκλωμα θα έχει αυξημένη δοκιμαστικότητα αλλά παράλληλα θα είναι μεγαλύτερο σε μέγεθος (area) και θα έχει και υψηλότερη κατανάλωση ενέργειας (power consumption). Η υψηλότερη κατανάλωση ενέργειας αποτελεί σημαντικό πρόβλημα σε περιπτώσεις όπου το ολοκληρωμένο κύκλωμα θα χρησιμοποιηθεί σε ένα σύστημα το οποίο τροφοδοτείται από κάποια μπαταρία. Παράλληλα, μπορεί να τεθούν και θέματα ψύξης (cooling) του κυκλώματος. Επίσης, οι σχεδιαστικές αλλαγές είναι δυνατό να επηρεάσουν και τη συχνότητα λειτουργίας του κυκλώματος. Αν και τα περισσότερα κυκλώματα επιδέχονται εύκολα σχεδιαστικές αλλαγές, οι μικροεπεξεργαστές αποτελούν την εξαίρεση στον κανόνα καθώς σχεδιάζονται με κύριο μέλημα την υψηλή επίδοση διατηρώντας τα επίπεδα κατανάλωσης ενέργειας όσο το δυνατό

χαμηλότερα. Έτσι, οποιοσδήποτε σχεδιαστικές αλλαγές που επιφέρει η ενσωμάτωση επιπλέον υλικού για αύξηση της δοκιμαστικότητας δεν είναι πάντα αποδεκτές παρόλο που οι σύγχρονες μέθοδοι κατασκευής κυκλωμάτων επιτρέπουν ολοκλήρωση περισσότερου υλικού. Τα τελευταία χρόνια έχει προταθεί η ιδέα της αυτοδοκιμής των μικροεπεξεργαστών χρησιμοποιώντας ενσωματωμένο λογισμικό (embedded software) που ονομάζεται *software-based self-test* ή *instruction-based self-test*. Το ενσωματωμένο λογισμικό βρίσκεται στην ενσωματωμένη μνήμη (on-chip memory cache) του μικροεπεξεργαστή και αναλαμβάνει το ρόλο της γέννησης και εφαρμογής των διανυσμάτων δοκιμής. Οι αποκρίσεις των διανυσμάτων δοκιμής αποθηκεύονται στην εσωτερική μνήμη και έπειτα με τη βοήθεια ενός απλού συστήματος δοκιμής χαμηλών επιδόσεων από πλευράς συχνότητας λειτουργίας, συγκρίνονται με τις αναμενόμενες αποκρίσεις για την ανίχνευση των ελαττωμάτων. Τα οφέλη από τη δοκιμή των μικροεπεξεργαστών μέσω του ενσωματωμένου λογισμικού είναι πολλαπλής φύσεως. Καταρχήν, επιτυγχάνεται η δοκιμή στην πραγματική συχνότητα λειτουργίας και όχι κατ' ανάγκη στη συχνότητα που υποστηρίζει ο εξοπλισμός δοκιμής, και επιπλέον δεν είναι απαραίτητο να γίνουν τροποποιήσεις στον αρχικό σχεδιασμό του μικροεπεξεργαστή. Επίσης, δεν απαιτείται τροποποίηση του κυκλώματος που μπορεί να επιφέρει μείωση στην απόδοση ή αύξηση της περιοχής. Τέλος, το ενσωματωμένο λογισμικό μπορεί να βρίσκεται σε μια οποιαδήποτε μη πτητική μνήμη (nonvolatile memory) έτσι ώστε η δοκιμή να πραγματοποιείται και στο πεδίο λειτουργίας του μικροεπεξεργαστή είτε κατά την έναρξη/τεματισμό της λειτουργίας του συστήματος, είτε κατά τα χρονικά διαστήματα τα οποία το σύστημα παραμένει αδρανές. Το ενσωματωμένο λογισμικό θα πρέπει να έχει όσο το δυνατό μικρότερες απαιτήσεις σε μνήμη καθώς συνυπάρχει μαζί με το ενσωματωμένο λογισμικό για τις εφαρμογές χρήστη. Παράλληλα, θα πρέπει να ελαχιστοποιεί την επικοινωνία με το σύστημα μνήμης, διότι το σύστημα μνήμης μαζί με το σύστημα διαύλου (bus) καταναλώνουν μεγάλα ποσά ενέργειας. Λαμβάνοντας υπόψη πως πολλά συστήματα τροφοδοτούνται από μπαταρίες, η ελαχιστοποίηση της κατανάλωσης ενέργειας αποτελεί καθοριστικό παράγοντα για τη βέλτιστη εφαρμογή του ενσωματωμένου λογισμικού αυτοδοκιμής.

Η ευρύτερη ερευνητική περιοχή στην οποία εντάσσεται η συνεισφορά της διδακτορικής διατριβής είναι η δοκιμή των μικροεπεξεργαστών που διαθέτουν μονάδες κινητής υποδιαστολής για την επίτευξη αριθμητικής υψηλών επιδόσεων.

Οι αριθμητικές μονάδες κινητής υποδιαστολής είναι ιδιαίτερα σημαντικές καθώς η παρουσία τους είναι καθοριστική για την ορθή και με μεγάλη ακρίβεια επεξεργασία των δεδομένων πραγματικών αριθμών. Δύο είναι οι κύριοι παράγοντες που οδήγησαν στην ευρεία χρήση των μονάδων κινητής υποδιαστολής τόσο σε επεξεργαστές γενικού σκοπού όσο και σε επεξεργαστικά συστήματα ειδικού σκοπού. Ο πρώτος παράγοντας έχει να κάνει με την καθιέρωση του προτύπου IEEE 754 το 1985 για την αναπαράσταση των πραγματικών αριθμών. Το πρότυπο IEEE 754 επιτρέπει την αναπαράσταση των πραγματικών αριθμών είτε με απλή είτε με διπλή ακρίβεια (single/double precision) χρησιμοποιώντας 32 ψηφία για την απλή ακρίβεια

και 64 ψηφία για τη διπλή². Με την αναθεώρηση του προτύπου, τον Αύγουστο του 2008, προστέθηκαν δύο νέες ακρίβειες, της μισής ακρίβειας με 16 ψηφία και της τετραπλής ακρίβειας με 128 ψηφία αναπαράστασης. Πριν από την καθιέρωση του προτύπου IEEE 754, κάθε κατασκευαστής ανέπτυξε τις μονάδες κινητής υποδιαστολής σύμφωνα με τις ανάγκες της εφαρμογής που έπρεπε να υποστηρίξει. Έτσι, οι μονάδες κινητής υποδιαστολής διέφεραν μεταξύ τους τόσο ως προς το εύρος και την ακρίβεια αναπαράστασης όσο και ως προς τις μεθόδους στρογγυλοποίησης κατά την εκτέλεση των αριθμητικών πράξεων. Το αποτέλεσμα ήταν ένα σύνολο ασύμβατων μεταξύ τους μονάδων κινητής υποδιαστολής, το κόστος κατασκευής των οποίων ήταν μεγάλο καθώς τα περιθώρια επαναχρησιμοποίησής τους ήταν πολύ περιορισμένα. Αντίστοιχα, το λογισμικό που χρησιμοποιούσε τις μονάδες κινητής υποδιαστολής είχε υψηλό κόστος ανάπτυξης αλλά το σημαντικότερο ίσως γεγονός ήταν πως όταν το ίδιο λογισμικό εκτελούνταν από διαφορετικούς επεξεργαστές παρήγαγε διαφορετικά αποτελέσματα. Άμεσος αντίκτυπος ήταν η αδυναμία επαναχρησιμοποίησης του λογισμικού με αποτέλεσμα το κόστος ανάπτυξής του να είναι υψηλό. Σήμερα, με τη σχεδόν καθολική αποδοχή του προτύπου IEEE 754 οι δυσκολίες αυτές έχουν ξεπεραστεί και έτσι έχει καταστεί δυνατή η ανάπτυξη πληθώρας εφαρμογών που χρησιμοποιούν τις μονάδες κινητής υποδιαστολής.

Ο δεύτερος παράγοντας για την ευρεία χρήση των μονάδων κινητής υποδιαστολής έχει σχέση με την αλματώδη ανάπτυξη της βιομηχανίας των ημιαγωγών η οποία έχει καταστήσει εφικτή την ενσωμάτωση μιας ή και περισσότερων μονάδων κινητής υποδιαστολής και πολλών επεξεργαστών σε ένα ολοκληρωμένο κύκλωμα. Αρκεί να θυμηθούμε πως μόλις πριν από δύο δεκαετίες η Intel κατασκεύαζε δύο διαφορετικές εκδόσεις για τους επεξεργαστές των οικογενειών 80386 και 80486. Οι επεξεργαστές με το επίθεμα SX δε διέθεταν μονάδα επεξεργασίας αριθμών κινητής υποδιαστολής, γνωστή τότε με το όνομα μαθηματικός συνεπεξεργαστής (mathematic co-processor) σε αντίθεση με τους επεξεργαστές με το επίθεμα DX οι οποίοι είχαν μονάδα κινητής υποδιαστολής και προορίζονταν κυρίως για την επιτάχυνση του λογισμικού που είχε ανάγκη την γρήγορη εκτέλεση αριθμητικών πράξεων μεταξύ πραγματικών αριθμών. Αντίθετα σήμερα, όχι μόνο η μονάδα κινητής υποδιαστολής βρίσκεται ενσωματωμένη στο ίδιο ολοκληρωμένο κύκλωμα με τον επεξεργαστή αλλά πολλοί επεξεργαστές ενσωματώνουν περισσότερες μονάδες κινητής υποδιαστολής για την αποτελεσματική εκτέλεση επιστημονικών και βιομηχανικών εφαρμογών με μεγάλες απαιτήσεις σε πράξεις μεταξύ πραγματικών αριθμών. Επίσης, οι μονάδες κινητής υποδιαστολής βρίσκονται στους ενσωματωμένους μικροεπεξεργαστές αλλά και στους ψηφιακούς επεξεργαστές σήματος, είτε υποστηρίζοντας πλήρως το πρότυπο IEEE 754, είτε μέρη αυτού τα οποία κρίνονται απαραίτητα για το εύρος των εφαρμογών που καλύπτει ο εκάστοτε μικροεπεξεργαστής. Ενδεικτικά μοντέλα τέτοιων επεξεργαστών αποτελούν οι ενσωματωμένοι μικροεπεξεργαστές Cortex-R4F, ARM1026EJ-S, ARM946E-S της εταιρίας ARM [15] και οι ψηφιακοί επεξεργαστές σήματος των οικογενειών TMS320C672x και

²Σύμφωνα με το πρότυπο IEEE 754 ένας αριθμός κινητής υποδιαστολής εκφράζεται με το πρόσημο, τον εκθέτη και το κλασματικό μέρος.

TMS320C67x της εταιρίας Texas Instruments [16].

Όπως προαναφέρθηκε το πρότυπο IEEE 754 αναθεωρήθηκε με την προσθήκη της μισής και τετραπλής ακρίβειας αναπαράστασης των αριθμών κινητής υποδιαστολής. Η αναθεώρηση του προτύπου που ξεκίνησε το 2000 και ολοκληρώθηκε το 2008, κρίθηκε αναγκαία καθώς το υπολογιστικό τοπίο έχει αλλάξει από την πρώτη παρουσίαση του προτύπου το 1985. Η ανάπτυξη της τεχνολογίας των ημιαγωγών έχει καταστήσει δυνατή την ενσωμάτωση πολλών μονάδων κινητής υποδιαστολής σε ένα ολοκληρωμένο κύκλωμα. Ενδεικτικά, η πρώτη μονάδα κινητής υποδιαστολής της Intel περιελάμβανε 40.000 τρανζίστορ [17] ενώ σήμερα το μέγεθος μπορεί να φθάσει μέχρι και το 1 εκατομμύριο τρανζίστορ. Βέβαια, η ύπαρξη κάθε τρανζίστορ είναι απολύτως δικαιολογημένη για την απαραίτητη αύξηση της επίδοσης των μονάδων κινητής υποδιαστολής.

Παράλληλα με την εξέλιξη της τεχνολογίας των ημιαγωγών έχει εξελιχθεί και το υπολογιστικό τοπίο. Ο Πίνακας 1.1 παρουσιάζει συνοπτικά μερικές από τις εφαρμογές όπου χρησιμοποιούνται αριθμοί κινητής υποδιαστολής. Παλαιότερα, οι αριθμοί κινητής υποδιαστολής χρησιμοποιούνταν κατά κύριο λόγο σε στρατιωτικές και διαστημικές εφαρμογές, για παράδειγμα σε συστήματα πλοήγησης πυραύλων ή αεροσκαφών και σε συστήματα ανίχνευσης όπως radar και sonar, όπου υπάρχει η ανάγκη για αναπαράσταση μεγάλων αριθμών που δεν μπορούν να αναπαρασταθούν με ακέραιους. Όμως σήμερα, εξαιτίας της αλματώδους ανάπτυξης των ημιαγωγών οι αριθμοί κινητής υποδιαστολής χρησιμοποιούνται σε πιο καθημερινές εφαρμογές. Χαρακτηριστικό παράδειγμα αποτελεί η σύγχρονη βιομηχανία ανάπτυξης ηλεκτρονικών παιχνιδιών, όπου η λεπτομερής απεικόνιση των ηρώων καθώς και η όσο το δυνατό πιο ρεαλιστική κίνησή τους καθιστά απαραίτητη την επεξεργασία των δεδομένων στο σύνολο των πραγματικών αριθμών. Τα συστήματα επεξεργασίας γραφικών (graphics processing units-GPU), οι κονσόλες παιχνιδιών (Playstation3, XBOX) αποτελούν συστήματα τα οποία κάνουν ευρεία χρήση των αριθμών κινητής υποδιαστολής για την απόδοση γεωμετρικών σχημάτων και μοντέλων σε 3 διαστάσεις (3D-rendering). Επίσης, τα συστήματα επεξεργασίας και ανάλυσης εικόνας, ιδιαίτερα στο χώρο της ιατρικής, χρησιμοποιούν αριθμούς κινητής υποδιαστολής. Ο χώρος της ψηφιακής φωτογραφίας αποτελεί μια νέα περιοχή στην οποία χρησιμοποιούνται αριθμοί κινητής υποδιαστολής, ιδιαίτερα στον τομέα της επεξεργασίας εικόνας υψηλής ευκρίνειας σύμφωνα με την τεχνική high dynamic range-HDR³. Οι αριθμοί κινητής υποδιαστολής χρησιμοποιούνται για την αναπαράσταση των τιμών των εικονοστοιχείων (pixels). Επίσης, τα συστήματα επεξεργασίας ήχου και σύνθεσης φωνής χρησιμοποιούν αριθμούς κινητής υποδιαστολής. Το ανθρώπινο αυτί έχει τη δυνατότητα να διακρίνει μικρές διακυμάνσεις στον ήχο και οι αριθμοί κινητής υποδιαστολής είναι κατάλληλοι για να αναπαραστήσουν με ακρίβεια τις συχνότητες των ηχητικών σημάτων. Τα συστήματα πλοήγησης (Global Positioning Systems-GPS) που όλο και πιο συχνά συναντώνται σε σύγχρονα οχήματα χρησιμοποιούν και αυτά αριθμούς

³ Στην τεχνική high dynamic range αυξάνεται το εύρος των εκθέσεων (exposures) μεταξύ των φωτεινών και σκοτεινών σημείων σε μια εικόνα. Με αυτό τον τρόπο η αυξάνεται η ευκρίνεια της ψηφιακής εικόνας.

κινητής υποδιαστολής. Οι παραπάνω εφαρμογές αναφέρονται κυρίως σε περιβάλλοντα με ενσωματωμένους μικροεπεξεργαστές. Προσθέτοντας στις παραπάνω εφαρμογές, τις εφαρμογές γραφείου που χρησιμοποιούν αριθμούς κινητής υποδιαστολής, τα εργαλεία σχεδιασμού με τη βοήθεια υπολογιστή (Computer Aided Design-CAD και Computer Aided Manufacturing-CAM) και τα εργαλεία μαθηματικών και προσομοίωσης γίνεται εμφανές πως οι εφαρμογές των αριθμών κινητής υποδιαστολής είναι πραγματικά αμέτρητες. Αξιοσημείωτο είναι το γεγονός πως σε ορισμένες εφαρμογές, όπως στις μελέτες των συστημάτων ασφαλείας των αυτοκινήτων κατά την πρόσκρουση, υπάρχει ανάγκη για την αναπαράσταση των πραγματικών με τόσο μεγάλη ακρίβεια όπου 64 ψηφία για την αναπαράσταση δεν είναι αρκετά και συνήθως χρησιμοποιούνται 128 ψηφία για την αναπαράσταση των αριθμών κινητής υποδιαστολής.

Αριθμοί κινητής υποδιαστολής και Εφαρμογές

Στρατιωτικές και Διαστημικές εφαρμογές (radar, sonar)
Βιομηχανικές εφαρμογές, ρομποτικός έλεγχος
Συστήματα Επεξεργασίας Εικόνας, Συστήματα Γραφικών (GPU), Παιχνιδομηχανές (Playstation3), Ψηφιακή Φωτογραφία (HDR)
Συστήματα Επεξεργασίας Ήχου, Σύνθεση Φωνής
Συστήματα Ασφάλειας με Βιομετρικά Χαρακτηριστικά
Συστήματα Πλοήγησης (Global Positioning System - GPS)
Συστήματα Προσομοίωσης (π.χ. Αντοχή Υλικών)

Πίνακας 1.1: Εφαρμογές των αριθμών κινητής υποδιαστολής

Η ορθή και αποδοτική υλοποίηση των μονάδων κινητής υποδιαστολής αλλά και των εφαρμογών που τις χρησιμοποιούν δεν είναι εύκολη. Πέρα από το διάσημο πλέον σχεδιαστικό σφάλμα στη μονάδα κινητής υποδιαστολής του επεξεργαστή Pentium το 1994, υπάρχουν περιπτώσεις όπου η λανθασμένη χρήση των αριθμών κινητής υποδιαστολής οδήγησε σε καταστροφικά αποτελέσματα. Χαρακτηριστικό παράδειγμα αποτελεί η μη επανδρωμένη διαστημική αποστολή Ariane-5, όπου η υπερχείλιση μιας μεταβλητής αριθμού κινητής υποδιαστολής, οδήγησε σε μια σειρά λανθασμένων ενεργειών η οποία είχε ως αποτέλεσμα την έκρηξη του πυραύλου έπειτα από 37 περίπου δευτερόλεπτα μετά την απογείωσή του [18]. Μια άλλη μοιραία αποτυχία η οποία κόστισε ανθρώπινες ζωές αποδίδεται σε διαδοχικά σφάλματα στρωγγυλοποίησης αριθμών κινητής υποδιαστολής στη μονάδα μέτρησης χρόνου ενός πυραύλου Patriot που χρησιμοποιείται στην αεράμυνα. Η μέτρηση του χρόνου στον συγκεκριμένο πύραυλο γίνονταν μετρώντας δέκατα του δευτερολέπτου. Όμως ο αριθμός 0,1 δεν μπορεί να αναπαρασταθεί με απόλυτη ακρίβεια στο δυαδικό σύστημα άρα ούτε και από αριθμούς κινητής υποδιαστολής σε υπολογιστή. Διαδοχικές στρωγγυλοποιήσεις έπειτα από 100 ώρες λειτουργίας οδήγησαν στην απόκλιση του ρολογιού κατά 0,34 δευτερόλεπτα, τα οποία ήταν αρκετά για να επηρεάσουν λανθασμένα το σύστημα ανίχνευσης του Patriot. Ο πύραυλος Patriot απέτυχε να αναχαιτίσει

τον πυραύλο Scud (απέκλινε κατά 500 περίπου μέτρα) με αποτέλεσμα ο Scud να χτυπήσει ένα στρατόνα [19].

Εκτός από την εμφανή ανάγκη για σωστή διαχείριση των αριθμών κινητής υποδιαστολής, οι υψηλές απαιτήσεις για την όσο το δυνατό ταχύτερη επεξεργασία των πραγματικών αριθμών οδήγησαν στην εξέλιξη και εμφάνιση νέων αρχιτεκτονικών υλοποίησης των μονάδων κινητής υποδιαστολής. Οι νέες αρχιτεκτονικές έχουν στόχο τους την επιτάχυνση της πρόσθεσης και της αφαίρεσης αριθμών κινητής υποδιαστολής καθώς οι δύο πράξεις αυτές όχι μόνο παρουσιάζουν μεγαλύτερη συχνότητα εμφάνισης αλλά είναι και οι δύο πιο χρονοβόρες πράξεις. Πειραματικές μετρήσεις έχουν δείξει πως ο χρόνος εκτέλεσης για τις πράξεις του πολλαπλασιασμού και της διαίρεσης κινητής υποδιαστολής είναι της ίδιας τάξης με τις αντίστοιχες ακέραιες πράξεις. Αντίθετα, η πρόσθεση/αφαίρεση κινητής υποδιαστολής απαιτεί πολύ περισσότερο χρόνο για την ολοκλήρωσή της σε σχέση με την αντίστοιχη ακέραια πράξη, χρόνος που είναι περίπου ίσος με αυτόν του πολλαπλασιασμού κινητής υποδιαστολής.

Έτσι, αντίθετα με ότι συμβαίνει στις αριθμητικές μονάδες που διαχειρίζονται ακέραιους όπου η πράξη της πρόσθεσης εκτελείται ταχύτατα, η πρόσθεση των αριθμών κινητής υποδιαστολής αποτελεί έναν ανασχετικό παράγοντα για την επίτευξη μονάδων κινητής υποδιαστολής υψηλών επιδόσεων. Σήμερα, όλες οι μονάδες κινητής υποδιαστολής υψηλών επιδόσεων που βρίσκονται σε σύγχρονους επεξεργαστές βασίζονται στην αρχιτεκτονική υλοποίησης των δύο μονοπατιών (two-path architecture) [20], [21], [22]. Πειραματικά αποτελέσματα που παρουσιάζονται στο Κεφάλαιο 3 δείχνουν επιτάχυνση της πρόσθεσης ή αφαίρεσης μέχρι και 30% για αριθμούς κινητής υποδιαστολής μονής ακρίβειας σε σχέση με την απλούστερη σχεδίαση του ενός μονοπατιού (one-path architecture). Είναι εύκολα αντιληπτό από τις προηγούμενες παραγράφους πως οι μονάδες κινητής υποδιαστολής υψηλών επιδόσεων έχουν ιδιαίτερα σημαντικό ρόλο στην αποτελεσματική επεξεργασία των πραγματικών αριθμών ενός μεγάλου εύρους εφαρμογών.

Παρόλο που οι μονάδες κινητής υποδιαστολής έχουν κυρίαρχο ρόλο σε μια πληθώρα εφαρμογών, η διεθνής βιβλιογραφία παρουσιάζει ελλείψεις σχετικά με τις μεθόδους δοκιμής και αυτοδοκιμής τους. Η δοκιμή των μονάδων κινητής υποδιαστολής παρουσιάζει ιδιαίτερες δυσκολίες εξαιτίας της πολύπλοκης αρχιτεκτονικής τους. Στις αρχές της δεκαετίας του 1990, η εξαγωγή διανυσμάτων για τη δοκιμή ενός πολλαπλασιαστική κινητής υποδιαστολής από ένα αυτόματο εργαλείο διαρκούσε περίπου 5 ημέρες. Οι εργασίες που παρουσιάζονται στη διεθνή βιβλιογραφία χωρίζονται σε δύο κατηγορίες. Η πρώτη κατηγορία περιλαμβάνει εργασίες που εφαρμόζουν τεχνικές σχεδίασης για αυξημένη δοκιμαστικότητα και προσθέτουν επιπλέον υλικό. Αν και οι τεχνικές σχεδίασης για αυξημένη δοκιμαστικότητα είναι αποδεκτές για κυκλώματα με χαμηλή συχνότητα λειτουργίας δεν ισχύει το ίδιο για τις μονάδες κινητής υποδιαστολής οι οποίες σχεδιάζονται με πρωτεύον κριτήριο την απόδοση. Η παρατήρηση αυτή αποτέλεσε το κίνητρο για τη δεύτερη κατηγορία που περιλαμβάνει εργασίες που δεν εφαρμόζουν τεχνικές σχεδίασης για αυξημένη δοκιμαστικότητα σε ευαίσθητες ως προς την απόδοση μονάδες

κινητής υποδιαστολής. Επιπλέον καμία από τις εργασίες που παρουσιάζονται στη διεθνή βιβλιογραφία δεν αντιμετωπίζει τη δοκιμή των μονάδων κινητής υποδιαστολής σε όλο το εύρος των ψηφίων ακρίβειας αναπαράστασης (μισή, απλή, διπλή, τετραπλή ακρίβεια). Μια άλλη σημαντική παρατήρηση που προκύπτει από τη μελέτη της διεθνούς βιβλιογραφίας είναι πως καμία εργασία δε λαμβάνει υπόψη τις σύγχρονες αρχιτεκτονικές σχεδίασης υψηλής απόδοσης, όπως η αρχιτεκτονική υλοποίησης δύο μονοπατιών που έχει κυριαρχήσει στην πράξη της πρόσθεσης. Τα κενά της βιβλιογραφίας ήταν το κίνητρο για την ενασχόληση με τις μονάδες κινητής υποδιαστολής υψηλών επιδόσεων στα πλαίσια της διατριβής αυτής. Τα κενά αυτά καλύπτει το πρώτο μέρος της παρούσας διατριβής, όπου μελετάται διεξοδικά και με γενικό τρόπο η δοκιμαστικότητα των μονάδων κινητής υποδιαστολής υψηλών επιδόσεων. Συγκεκριμένα, το πρώτο μέρος της διατριβής πραγματεύεται τη δοκιμαστικότητα και την εξαγωγή διανυσμάτων δοκιμής για μονάδες κινητής υποδιαστολής με βάση τους ακόλουθους στόχους:

- την αύξηση της δοκιμαστικότητας χωρίς τη χρήση επιπλέον υλικού,
- την επαναχρησιμοποίηση των υπάρχοντων διανυσμάτων δοκιμής για ακέραιες αριθμητικές μονάδες (που χρησιμοποιούνται ευρύτατα στο εσωτερικό των μονάδων κινητής υποδιαστολής),
- τη γενικότητα των προτεινόμενων μεθόδων ως προς την κλιμάκωση της ακρίβειας των αριθμών κινητής υποδιαστολής (απλή, διπλή, τετραπλή ή ειδικά διαμορφωμένη ακρίβεια) πάντα στα πλαίσια του προτύπου IEEE 754,
- την ανεξαρτησία από την εσωτερική υλοποίηση των ακεραίων αριθμητικών μονάδων που συνθέτουν τη μονάδα κινητής υποδιαστολής,
- την ανεξαρτησία από το μοντέλο ελαττωμάτων.

Η αύξηση της δοκιμαστικότητας και η εξαγωγή διανυσμάτων δοκιμής πρέπει να πραγματοποιηθεί δίχως επιπλέον υλικό, διότι, όπως έχει γίνει αντιληπτό οι μονάδες κινητής υποδιαστολής παρουσιάζουν μηδενική ανοχή σε τεχνικές σχεδίασης για αυξημένη δοκιμαστικότητα οι οποίες επηρεάζουν τη συχνότητα λειτουργίας και το μέγεθος του κυκλώματος. Η επαναχρησιμοποίηση διανυσμάτων δοκιμής για ακέραιες αριθμητικές μονάδες είναι επιθυμητή καθώς μπορεί να διευκολύνει την ανάπτυξη διανυσμάτων δοκιμής για μονάδες κινητής υποδιαστολής. Ταυτόχρονα, η εξαγωγή διανυσμάτων για τις μονάδες κινητής υποδιαστολής θα πρέπει να είναι ευέλικτη και να προσαρμόζεται ανάλογα με την ακρίβεια αναπαράστασης των αριθμών κινητής υποδιαστολής, η οποία καθορίζεται από τις ανάγκες της εκάστοτε εφαρμογής. Οποιοσδήποτε αλλαγές στην υλοποίηση των ακεραίων αριθμητικών μονάδες δε θα πρέπει να επηρεάζουν την ανάπτυξη διανυσμάτων δοκιμής.

Η προτεινόμενη στη διατριβή αυτή μεθοδολογία βασίζεται σε ένα σύνολο συνθηκών δοκιμαστικότητας (testability conditions) που έχουν αναπτυχθεί για τις επιμέρους ακέραιες αριθμητικές μονάδες (αθροιστές, αφαιρέτες, συγκριτές, ολισθητές) από τις οποίες αποτελείται μια

μονάδα κινητής υποδιαστολής. Οι συνθήκες δοκιμαστικότητας αποτελούνται από δύο επιμέρους συνθήκες, αυτές της ελεγχιμότητας (controllability) και της παρατηρησιμότητας (observability). Πιο συγκεκριμένα, οι συνθήκες ελεγχιμότητας περιγράφουν την επιτρεπτή μορφή των διανυσμάτων δοκιμής τα οποία είναι δυνατό να εφαρμοστούν σε μια ακέραια αριθμητική μονάδα που βρίσκεται ενσωματωμένη σε μια μονάδα κινητής υποδιαστολής. Οι συνθήκες παρατηρησιμότητας από την άλλη εκφράζουν πότε είναι δυνατόν ένα ελάττωμα να επηρεάσει την ορθή λειτουργία της μονάδας κινητής υποδιαστολής και να δώσει λανθασμένη τιμή στις εξόδους της. Οι συνθήκες δοκιμαστικότητας έχουν εξαχθεί βάσει της αρχιτεκτονικής υλοποίησης της μονάδας κινητής υποδιαστολής δίχως να προϋποθέτουν την ύπαρξη επιπλέον υλικού για αύξηση της δοκιμαστικότητας. Αν λάβουμε υπόψη το γεγονός ότι αρκετοί επεξεργαστές ενσωματώνουν δυο μονάδες κινητής υποδιαστολής, όπως ο PowerPC 970FX [23] της IBM, αλλά και την εμφάνιση πολλαπλών επεξεργαστών σε ένα ολοκληρωμένο κύκλωμα όπου ο καθένας ξεχωριστά ενσωματώνει μια μονάδα κινητής υποδιαστολής, όπως ο Niagara-2 (UltraSparc) [24] της Sun, η απουσία του επιπλέον υλικού για την αύξηση της δοκιμαστικότητας παρέχει μεγάλη εξοικονόμηση στο συνολικό μέγεθος του ολοκληρωμένου κυκλώματος. Οι συνθήκες δοκιμαστικότητας που προτείνονται στη διατριβή αυτή παρέχουν μεγάλη ευελιξία καθώς είναι ανεξάρτητες της εσωτερικής υλοποίησης των ακέραιων αριθμητικών μονάδων που περιλαμβάνει η μονάδα κινητής υποδιαστολής. Η ιδιότητα αυτή είναι σημαντική διότι επιτρέπει την μελλοντική αντικατάσταση των ακέραιων μονάδων από άλλες, είτε για λόγους που έχουν να κάνουν με την επίδοση της μονάδας, είτε για λόγους που έχουν σχέση με το μέγεθός της, δίχως όμως να επηρεάζονται οι συνθήκες δοκιμαστικότητας της μονάδας κινητής υποδιαστολής. Ένα άλλο πλεονέκτημα που διαθέτουν οι συνθήκες δοκιμαστικότητας, το οποίο σχετίζεται άμεσα με την ολοκλήρωση της αναθεώρησης του προτύπου IEEE 754 όσον αφορά την ακρίβεια αναπαράστασης των πραγματικών αριθμών, είναι η δυνατότητα αξιοποίησης των συνθηκών ανεξάρτητα από την ακρίβεια αναπαράστασης, καθώς οι προτεινόμενες συνθήκες δοκιμαστικότητας βασίζονται στην αρχιτεκτονική υλοποίησης και είναι ανεξάρτητες της ακρίβειας αναπαράστασης. Όπως αναφέρθηκε σε προηγούμενες παραγράφους, υπάρχουν εφαρμογές που απαιτούν πολύ μεγάλη ακρίβεια αναπαράστασης, για παράδειγμα τετραπλή ακρίβεια και άλλες που χρησιμοποιούν μικρότερη ακρίβεια από αυτήν που ορίζει το πρότυπο. Εκτός από τις συνθήκες δοκιμαστικότητας, το πρώτο μέρος της διατριβής πραγματεύεται επίσης την εξαγωγή διανυσμάτων δοκιμής (test vector generation) με βάση αυτές τις συνθήκες. Σχετικά με την εξαγωγή διανυσμάτων για τις μονάδες κινητής υποδιαστολής, η διεθνής βιβλιογραφία παρουσιάζει ολιγάριθμες εργασίες. Οι εργασίες αυτές βασίζονται στα αυτόματα εργαλεία εξαγωγής διανυσμάτων για τις μονάδες κινητής υποδιαστολής. Αν και τα εργαλεία αυτόματης εξαγωγής διανυσμάτων επιτυγχάνουν υψηλά ποσοστά κάλυψης ελαττωμάτων για κάποια μοντέλα ελαττωμάτων, όπως για τα μοντέλα προσκόλλησης (stuck-at) και καθυστερημένης μετάβασης (transition delay) (αναλύονται στο Κεφάλαιο 2), παραμένει το ερώτημα της εφαρμογής των διανυσμάτων αυτών. Η παρούσα διδακτορική διατριβή στοχεύει με τη βοήθεια

των συνθηκών δοκιμαστικότητας στην επαναχρησιμοποίηση των διανυσμάτων δοκιμής που έχουν αναπτυχθεί για τις ακέραιες αριθμητικές μονάδες. Εξαιτίας της έμφυτης κανονικής, επαναληπτικής δομής που παρουσιάζουν οι ακέραιες αριθμητικές μονάδες, τα διανύσματα δοκιμής χαρακτηρίζονται και αυτά από κάποια κανονικότητα, η οποία δίνει μια ευκολία αναπαγωγής τους τόσο από το υλικό όσο και από το λογισμικό για οποιοδήποτε μέγεθος λέξης. Κάτι τέτοιο δεν είναι εφικτό για τα διανύσματα δοκιμής που παράγονται από τα αυτόματα εργαλεία εξαγωγής διανυσμάτων και αποτελεί σημαντικό μειονέκτημα της χρήσης τους.

Οι αριθμητικές μονάδες κινητής υποδιαστολής σπάνια βρίσκονται σαν αυτόνομες μονάδες αλλά βρίσκονται σχεδόν πάντα ενσωματωμένες σε έναν επεξεργαστή είτε γενικού σκοπού, είτε ειδικού σκοπού, είτε σε έναν ψηφιακό επεξεργαστή σήματος. Την τελευταία δεκαετία παρουσιάζεται στη διεθνή βιβλιογραφία ένα διαρκώς αυξανόμενο σύνολο εργασιών που πραγματεύονται την αυτοδοκιμή των μικροεπεξεργαστών με τη βοήθεια του ενσωματωμένου λογισμικού. Ένα μεγάλο εύρος μικροεπεξεργαστών, από απλούς επεξεργαστές των 8-bit έως και πολύ πολύπλοκους των 32-bit με μηχανισμούς διοχέτευσης (pipeline) και κρυφές μνήμες (cache memories) έχει αποτελέσει το ερευνητικό πεδίο των εργασιών αυτών. Εντούτοις, η βιβλιογραφία παρουσιάζει κενά σχετικά με την αυτοδοκιμή των μονάδων κινητής υποδιαστολής υψηλών επιδόσεων με ενσωματωμένο λογισμικό. Ένας μικρός αριθμός εργασιών έχει ασχοληθεί με τις μονάδες κινητής υποδιαστολής και την παραγωγή λογισμικού για την αυτοδοκιμή τους ενώ καμία εργασία δεν αναφέρεται στη σύγχρονη αρχιτεκτονική δύο μονοπατιών. Αντίθετα, όλες οι εργασίες που πραγματεύονται την αυτοδοκιμή των μικροεπεξεργαστών με ενσωματωμένο λογισμικό εστιάζουν στις αριθμητικές μονάδες ακεραίων αριθμών.

Το δεύτερο μέρος της διατριβής πραγματεύεται τη συστηματική ανάπτυξη ενσωματωμένου λογισμικού για την αποδοτική αυτοδοκιμή των ακέραιων μονάδων και των μονάδων κινητής υποδιαστολής, ώστε το ενσωματωμένο λογισμικό να είναι κατάλληλο για αυτοδοκιμή στο πεδίο κατά τη διάρκεια λειτουργίας (on-line testing) του μικροεπεξεργαστή. Η δοκιμή του μικροεπεξεργαστή κατά τη διάρκεια λειτουργίας σε ένα σύστημα είναι τελείως διαφορετική από,τι στο εργοστάσιο. Σε ένα σύστημα, ο μικροεπεξεργαστής πλαισιώνεται από μνήμες, χρονομετρητές, ελεγκτές διαύλων και άλλα περιφερειακά. Επιπλέον, οι μνήμες δεν περιέχουν μόνο το λογισμικό αυτοδοκιμής, αλλά μπορεί να περιέχουν το λειτουργικό σύστημα και το λογισμικό της εφαρμογής. Έτσι, το λογισμικό αυτοδοκιμής συναγωνίζεται το υπόλοιπο λογισμικό για πόρους του συστήματος. Για αυτό το λόγο, το λογισμικό αυτοδοκιμής θα πρέπει να έχει μικρό ίχνος μνήμης αλλά και μικρό χρόνο εκτέλεσης ώστε να μην επιβαρύνει την απόδοση του συστήματος. Επίσης το λογισμικό αυτοδοκιμής θα πρέπει να έχει χαμηλή κατανάλωση ενέργειας καθώς πολλά ενσωματωμένα συστήματα τροφοδοτούνται από μπαταρίες.

Η ανάπτυξη του ενσωματωμένου λογισμικού για τις μονάδες κινητής υποδιαστολής διαφέρει από αυτή των ακέραιων μονάδων και εισάγει νέες προκλήσεις. Στις περισσότερες περιπτώσεις το σύνολο των εντολών για επεξεργασία αριθμών κινητής υποδιαστολής παρέχει λιγότερες δυνατότητες σε σχέση με αυτές που παρέχονται από το σύνολο εντολών για επεξεργασία ακέ-

ρικών αριθμών. Για παράδειγμα δεν υπάρχει δυνατότητα ολίσθησης ενός καταχωρητή κινητής υποδιαστολής ενώ η ολίσθηση ακεραίων υλοποιείται πάντα με μια απλή εντολή. Επιπλέον, κανένα σύνολο εντολών δεν παρέχει τη δυνατότητα για επεξεργασία των επιμέρους τμημάτων ενός καταχωρητή κινητής υποδιαστολής (πρόσημο, εκθέτης, κλασματικό μέρος). Μια άλλη σημαντική πρόκληση που παρουσιάζει η αυτοδοκιμή των μονάδων κινητής υποδιαστολής με λογισμικό είναι η περίπτωση όπου ένας μικροεπεξεργαστής 32-bit ενσωματώνει μια μονάδα κινητής υποδιαστολής διπλής ακρίβειας (όπου απαιτούνται 64-bit για την αναπαράσταση). Έτσι, προκύπτει το ερώτημα πώς μπορεί να εφαρμοστεί ένα διάνυσμα δοκιμής 64-bit τη στιγμή που ο επεξεργαστής χειρίζεται 32-bit τιμές.

Η συστηματική ανάπτυξη του ενσωματωμένου λογισμικού για την αυτοδοκιμή των μονάδων κινητής υποδιαστολής που προτείνει το δεύτερο μέρος της διατριβής υποδιαστολής βασίζεται στις συνθήκες δοκιμαστικότητας, και στην εξαγωγή και επαναχρησιμοποίηση διανυσμάτων δοκιμής τα οποία είναι εύκολο να αναπαραχθούν με τη βοήθεια του συνόλου εντολών του μικροεπεξεργαστή. Έτσι, διατηρούνται τα πλεονεκτήματα των συνθηκών δοκιμαστικότητας, όπως η κλιμάκωση σε διαφορετικής ακρίβειας αναπαραστάσεις καθώς και η ανεξαρτησία από την εσωτερική υλοποίηση των επιμέρους δομικών στοιχείων που απαρτίζουν τη μονάδα κινητής υποδιαστολής. Επιπλέον, η μεθοδολογία ανάπτυξης λογισμικού αυτοδοκιμής εισάγει την επεξεργασία σε επίπεδο δυαδικών ψηφίων (bit level manipulation) για να επεκτείνει τις δυνατότητες του συνόλου εντολών επεξεργασίας αριθμών κινητής υποδιαστολής ώστε α) να δημιουργήσει νέες ψευδοεντολές (ακολουθίες εντολών) οι οποίες επιτρέπουν την επεξεργασία των επιμέρους τμημάτων των καταχωρητών κινητής υποδιαστολής (πρόσημο, εκθέτης, κλασματικό μέρος), β) να δημιουργήσει νέες ψευδοεντολές (ακολουθίες εντολών) οι οποίες επιτρέπουν την επεξεργασία ενός ή περισσότερων καταχωρητών σαν έναν ενιαίο καταχωρητή. Κατ' αυτόν τον τρόπο, αντιμετωπίζεται και η ανάπτυξη λογισμικού αυτοδοκιμής για μικροεπεξεργαστές αρχιτεκτονικής 32-bit που διαθέτουν μονάδες κινητής υποδιαστολής διπλής ακρίβειας. Επιπλέον, η διατριβή εξετάζει τις δυνατότητες των συνόλων εντολών αντιπροσωπευτικών σύγχρονων μικροεπεξεργαστών όπως MIPS, Alpha, PowerPC. Ο Πίνακας 1.2 παρουσιάζει αντιπροσωπευτικούς συνδυασμούς μικροεπεξεργαστών και μονάδων κινητής υποδιαστολής για τις οποίες αναπτύχθηκε λογισμικό αυτοδοκιμής.

Η ανάπτυξη του ενσωματωμένου λογισμικού πραγματοποιείται με χρήση της συμβολικής γλώσσας μηχανής του μικροεπεξεργαστή και στηρίζεται σε βρόχους επανάληψης που συνδυάζουν εντολές ακέραιας πρόσθεσης, αφαίρεσης, ολίσθησης που χρησιμοποιούνται για τη γέννηση των επιμέρους τμημάτων (πρόσημο, εκθέτης, κλασματικό μέρος) των διανυσμάτων δοκιμής. Οι εντολές ακέραιας πρόσθεσης, αφαίρεσης και ολίσθησης υλοποιούνται από όλους τους μικροεπεξεργαστές με αποτέλεσμα την εύκολη μεταφερσιμότητα του ενσωματωμένου λογισμικού σε οποιονδήποτε μικροεπεξεργαστή. Μόνο το πλήθος των επαναλήψεων των βρόχων εξαρτάται αποκλειστικά από την ακρίβεια αναπαράστασης των πραγματικών αριθμών. Έτσι η μεθοδολογία ανάπτυξης του λογισμικού μπορεί να εφαρμοστεί ανεξάρτητα από την

Επεξεργαστής	Μέγεθος	Ακρίβεια Υποδιαστολής	Μονάδας Κινητής	Ειδικές εντολές επεξεργασίας σε επίπεδο bit
MIPS	32-bit	Απλή (32-bit)		Όχι
MIPS	32-bit	Διπλή (64-bit)		Όχι
PowerPC	32-bit	Απλή (32-bit)		Ναι
PowerPC	32-bit	Διπλή (64-bit)		Ναι
Alpha	64-bit	Διπλή (64-bit)		Όχι

Πίνακας 1.2: Συνδυασμοί επεξεργαστών και μονάδων κινητής υποδιαστολής για τους οποίους έγινε ανάπτυξη λογισμικού αυτοδοκιμής.

ακρίβεια αναπαράστασης, ακόμα και στη νέα τετραπλή ακρίβεια του αναθεωρημένου προτύπου IEEE 754. Το χαρακτηριστικό αυτό αποτελεί ένα σημαντικό πλεονέκτημα καθώς πολλές εφαρμογές χρειάζονται μια ειδικά διαμορφωμένη ακρίβεια.

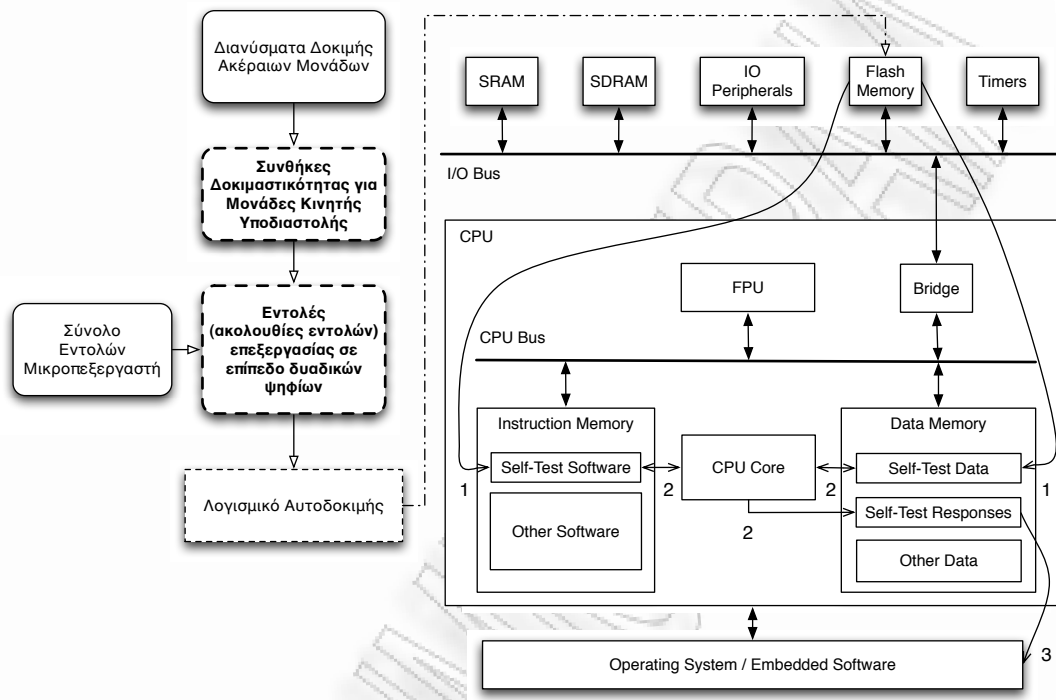
Η μεθοδολογία ανάπτυξης λογισμικού που προτείνεται πλεονεκτεί έναντι μεθοδολογιών που απλά εφαρμόζουν διανύσματα δοκιμής που έχουν εξαχθεί από εργαλεία αυτόματης εξαγωγής διανυσμάτων καθώς η χρήση επαναληπτικών βρόχων μειώνει το συνολικό μέγεθος του λογισμικού και το καθιστά κατάλληλο για χρήση και κατά τη λειτουργία (on-line testing). Αναλυτικά πειραματικά αποτελέσματα που παρουσιάζονται στο Κεφάλαιο 5 δείχνουν μείωση του μεγέθους του λογισμικού αυτοδοκιμής της τάξης του 40% και 50% για τις μονάδες πρόσθεσης/αφαίρεσης απλής και διπλής ακρίβειας, αντίστοιχα. Η μείωση του μεγέθους του ενσωματωμένου λογισμικού είναι πολλαπλής σημασίας. Όταν ο έλεγχος πραγματοποιείται αμέσως μετά από την κατασκευή του κυκλώματος, το ενσωματωμένο λογισμικό με χαμηλές απαιτήσεις σε μνήμη θα φορτωθεί ταχύτερα μειώνοντας το συνολικό χρόνο της δοκιμής. Υπενθυμίζουμε πως στην πλειοψηφία των περιπτώσεων ο εξοπλισμός δοκιμής λειτουργεί σε χαμηλότερη συχνότητα από αυτήν του κυκλώματος. Από την άλλη πλευρά, όταν πραγματοποιείται δοκιμή κατά τη διάρκεια λειτουργίας του κυκλώματος, είναι αναγκαίο το ενσωματωμένο λογισμικό να έχει χαμηλές απαιτήσεις σε μνήμη αλλά και να πραγματοποιεί όσο το δυνατό λιγότερες προσπελάσεις στο σύστημα της μνήμης. Δε θα πρέπει να ξεχνάμε πως το ενσωματωμένο λογισμικό κατά τη δοκιμή χρησιμοποιεί τους πόρους του συστήματος στο οποίο συνυπάρχει ενσωματωμένο λογισμικό της εφαρμογής χρήστη.

Η Εικόνα 1.2 συνοψίζει τη συνεισφορά της διδακτορικής διατριβής στη δοκιμή των μονάδων κινητής υποδιαστολής υψηλών επιδόσεων, την εύρεση συνθηκών δοκιμαστικότητας και την ανάπτυξη αποδοτικού λογισμικού αυτοδοκιμής.

Η διάρθρωση της διατριβής σε κεφάλαια είναι η ακόλουθη:

Στο Κεφάλαιο 2 παρουσιάζονται οι θεμελιώδεις έννοιες και οι μεθοδολογίες για τη δοκιμή των ολοκληρωμένων κυκλωμάτων. Στη συνέχεια, αναπτύσσεται η έννοια της αυτοδοκιμής των μικροεπεξεργαστών με ενσωματωμένο λογισμικό.

Στο Κεφάλαιο 3 παρουσιάζεται το πρότυπο IEEE 754 για την αναπαράσταση των πραγ-



Εικόνα 1.2: Απεικόνιση της συνεισφοράς της διδακτορικής διατριβής (με έντονη διαγράμμιση στο σχήμα) στην εξαγωγή διανυσμάτων δοκιμής για μονάδες κινητής υποδιαστολής και ανάπτυξη λογισμικού αυτοδοκιμής για αυτές στο πλαίσιο της δοκιμής κατά τη λειτουργία. Στο βήμα 1 ο μικροεπεξεργαστής μεταφέρει το λογισμικό αυτοδοκιμής στην κρυφή μνήμη. Στο βήμα 2, ο μικροεπεξεργαστής εκτελεί το λογισμικό αυτοδοκιμής και παράγει τις αποκρίσεις δοκιμής. Τέλος, στο βήμα 3, οι αποκρίσεις δοκιμής συγκρίνονται με τις αναμενόμενες αποκρίσεις ώστε να ανιχνευθούν τυχόν ελαττώματα.

ματικών αριθμών. Επίσης, αναλύονται οι αριθμητικές πράξεις μεταξύ των αριθμών κινητής υποδιαστολής και περιγράφονται οι αλγόριθμοι υλοποίησης των πράξεων από το υλικό. Ιδιαίτερη έμφαση δίνεται στις μονάδες άθροισης/αφαίρεσης των αριθμών κινητής υποδιαστολής οι οποίες υλοποιούν τη συννηθέστερη αλλά και μια από τις περισσότερο χρονοβόρες πράξεις. Έπειτα, αναλύεται διεξοδικά η σύγχρονη αρχιτεκτονική του αθροιστή/αφαιρέτη βασισμένη στην υλοποίηση δύο μονοπατιών. Επίσης, παρουσιάζονται πειραματικά αποτελέσματα που δείχνουν τη βελτίωση της απόδοσης του αθροιστή/αφαιρέτη δύο μονοπατιών έναντι της απόδοσης του αθροιστή ενός μονοπατιού.

Στο Κεφάλαιο 4 γίνεται αρχικά μια σύντομη επισκόπηση της βιβλιογραφίας σχετικά με τις μεθοδολογίες δοκιμής μονάδων κινητής υποδιαστολής και αναλύονται τα μειονεκτήματά τους. Στη συνέχεια, εισάγονται αναλυτικά οι συνθήκες δοκιμαστικότητας για τις μονάδες κινητής υποδιαστολής και έπειτα παρουσιάζεται η μεθοδολογία για την εξαγωγή των διανυσμάτων δοκιμής αλλά και επαναχρησιμοποίησης διανυσμάτων δοκιμής για τις ακέραιες αριθμητικές μονάδες. Τα αποτελέσματα του κεφαλαίου αυτού που αποτελούν και το πρώτο μέρος της

διατριβής έχουν δημοσιευτεί στις εργασίες [25] και [26].

Στο Κεφάλαιο 5 παρουσιάζεται η μεθοδολογία ανάπτυξης ενσωματωμένου λογισμικού για την αυτοδοκιμή των μονάδων κινητής υποδιαστολής. Ιδιαίτερη έμφαση δίνεται στην ανάπτυξη ενσωματωμένου λογισμικού ώστε αυτό να συμβαδίζει με τις απαιτήσεις για την αυτοδοκιμή κατά τη διάρκεια λειτουργίας (on-line testing) για τις μονάδες κινητής υποδιαστολής διπλής ακρίβειας όπου οι επεξεργαστές δεν παρέχουν τις απαιτούμενες εντολές για την διαχείριση καταχωρητών αντίστοιχης ακρίβειας. Έπειτα, δίνονται πειραματικές μετρήσεις για το παραγόμενο ενσωματωμένο λογισμικό και γίνεται η σύγκριση με άλλες τεχνικές για την ανάπτυξη λογισμικού αυτοδοκιμής που εξάγεται χρησιμοποιώντας διανύσματα ελέγχου από αυτόματα εργαλεία εξαγωγής αλλά και γεννιότερες ψευδοτυχαίων διανυσμάτων. Επίσης παρουσιάζονται αποτελέσματα που δείχνουν πως το λογισμικό αυτοδοκιμής επηρεάζεται από αντιπροσωπυτικά σύνολα εντολών σύγχρονων μικροεπεξεργαστών.

Τα αποτελέσματα του κεφαλαίου αυτού που αποτελεί το δεύτερο μέρος της διδακτορικής διατριβής έχουν δημοσιευτεί στις εργασίες [27], [28] και [29].

Λίστα Δημοσιεύσεων

- Instruction-based On-line Periodic Self-testing of Microprocessors With Floating-point Units, G. Xenoulis, D. Gizopoulos, M. Psarakis, A. Paschalis, IEEE Transactions on Dependable and Secure Computing, accepted for publication.
- Testability Analysis and Scalable Test Generation for High-Speed Floating Point Units, G. Xenoulis, D. Gizopoulos, M. Psarakis and A. Paschalis, IEEE Transactions on Computers, vol. 55, no. 11, pp. 1449-1457, November 2006.
- On-Line Periodic Self-Testing of High-Speed Floating-Point Units in Microprocessors, G. Xenoulis, M. Psarakis, D. Gizopoulos, A. Paschalis, IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFTS 2007), Rome, Italy, September 2007.
- Test Generation Methodology for High-Speed Floating Point Adders, G. Xenoulis, M. Psarakis, D. Gizopoulos, A. Paschalis, in Proceedings of the 11th IEEE International On-Line Testing Symposium 2005, San Rafael, France, July 2005.
- Low-Cost On-Line Software-Based Self-Testing of Embedded Processor Cores, G. Xenoulis, D. Gizopoulos, N. Kranitis, A. Paschalis, in Proceedings of the 9th IEEE International On-Line Testing Symposium 2003, Kos, Greece, 7-9 July 2003.

Άλλες Δημοσιεύσεις

- Software-Based Self-Test of Embedded Processors, N. Kranitis, A. Paschalis, D. Gizopoulos, G. Xenoulis, in Processor Design – System-On-Chip Computing for ASICs and FPGAs, edited by Jari Nurmi, ISBN 978-1-4020-5529-4, Hardcover, Springer May 2007.
- Software-Based Self-Testing of Embedded Processors, N. Kranitis, G. Xenoulis, A. Paschalis, D. Gizopoulos, IEEE Transactions on Computers, vol. 54, no. 4, pp. 461-475, April 2005.
- Low-Cost Software-Based Self-Testing of RISC Processor Cores, N. Kranitis, G. Xenoulis, D. Gizopoulos, A. Paschalis, Y. Zorian, IEE Computers and Digital Techniques, vol. 150, no. 5, Invited Paper, September 2003.
- Software-Based Self-Test for Pipelined Processors: A Case Study, M.Hatzimihail, G. Xenoulis, A. Paschalis, M. Psarakis, D. Gizopoulos, in Proceedings of the IEEE International Symposium on Defect and Fault Tolerance (DFTS), Monterey, CA, USA, October 2005.
- Application and Analysis of RT-Level Software-Based Self-Testing for Embedded Processor Cores, N. Kranitis, G. Xenoulis, A. Paschalis, D.Gizopoulos, Y. Zorian, in Proceedings of the IEEE International Test Conference (ITC) 2003, Charlotte, NC, USA, September 30 October 2, 2003.
- Low-Cost Software-Based Self-Testing of RISC Processor Cores, N. Kranitis, G. Xenoulis, D. Gizopoulos, A. Paschalis, Y. Zorian, IEEE Design Automation and Test in Europe Conference (DATE2003), Munich, Germany, March 2003.
- Software-Based Self-Testing of Large Register Banks in RISC Processor Cores, N. Kranitis, G. Xenoulis, D. Gizopoulos, A. Paschalis, Y. Zorian, in Proceedings of the 4th IEEE Latin American Test Workshop 2003, Natal, Brazil, 10-13 February 2003.
- Embedded Software-Based Self-Testing of Processors Cores: Application to a RISC Architecture, G. Xenoulis, N. Kranitis, D. Gizopoulos, A. Paschalis, Y. Zorian, IEEE International Test Resource Partitioning Workshop 2002 (TRP2002), Baltimore, MD, USA, pp. 2.2.1 2.2.5, October 10-11, 2002.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

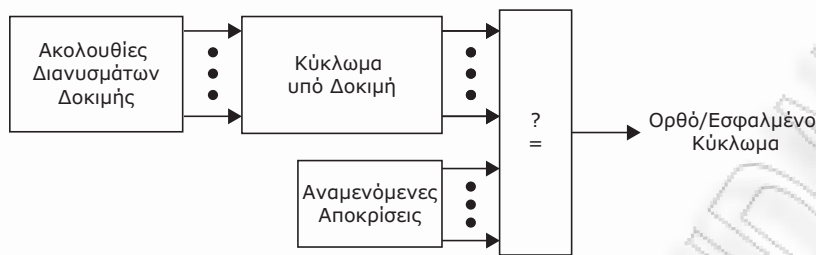
Κεφάλαιο 2

Δοκιμή Ψηφιακών Ολοκληρωμένων Κυκλωμάτων και Μικροεπεξεργαστών

Η δοκιμή των ολοκληρωμένων κυκλωμάτων (*IC test*) είναι ένα πείραμα το οποίο στοχεύει στο διαχωρισμό των ορθών κυκλωμάτων από τα ελαττωματικά κυκλώματα [30]. Το ολοκληρωμένο κύκλωμα τίθεται σε δοκιμασία και η απόκρισή του αναλύεται ώστε να διαπιστωθεί αν είναι η αναμενόμενη [31]. Όταν ένα ολοκληρωμένο δεν παρέχει την αναμενόμενη λειτουργία ή υπηρεσία τότε γίνεται λόγος για *αστοχία (failure)* του κυκλώματος. Ένα *λάθος (error)* του κυκλώματος είναι μια λανθασμένη τιμή στις εξόδους του. Ένα κύκλωμα περιέχει *ελαττώματα (faults)* όταν υπάρχουν φυσικές διαφορές ανάμεσα στο κύκλωμα που λειτουργεί ορθά και σε αυτό που λειτουργεί εσφαλμένα. Οι φυσικές διαφορές καλούνται *ατέλειες* ή *ψεγάδια (defects)*. Το κύκλωμα που δεν παρέχει την αναμενόμενη υπηρεσία/λειτουργία εξαιτίας ελαττωμάτων καλείται *εσφαλμένο (faulty)*, ενώ στην αντίθετη περίπτωση καλείται *ορθό* ή *ελεύθερο από ελαττώματα (fault-free)*.

Η δοκιμή των ολοκληρωμένων κυκλωμάτων αποτελεί μια καίρια φάση κατά τη διαδικασία της κατασκευής τους. Η διαδικασία κατασκευής των κυκλωμάτων είναι τόσο πολύπλοκη που σε καμία περίπτωση δεν είναι δυνατό να κατασκευαστεί ένα κύκλωμα ελεύθερο από ελαττώματα σε ποσοστό 100%. Η δοκιμή είναι το μόνο μέσο το οποίο μπορεί να εξασφαλίσει την παρουσία αξιόπιστων κυκλωμάτων στην αγορά και είναι τόσο σημαντική ώστε κανένα κύκλωμα δε σχεδιάζεται αν δεν έχει ληφθεί μέριμνα για τη δοκιμή του.

Οι ατέλειες και τα ψεγάδια είναι αναπόσπαστο κομμάτι της διαδικασίας κατασκευής των ολοκληρωμένων κυκλωμάτων και εκφράζονται κατά κύριο λόγο ως *βραχυκυκλώματα (shorts)* και *ανοιχτές συνδέσεις (opens)*. Τα βραχυκυκλώματα οφείλονται στο επιπλέον αγωγίμο υλικό



Εικόνα 2.1: Το πείραμα της δοκιμής.

(αλουμίνιο ή χαλκός) που μπορεί κατά λάθος να υπάρχει ή στην απουσία μονωτικού υλικού (insulating material). Αντίστοιχα, οι ανοιχτές συνδέσεις οφείλονται στην απουσία αγωγικού υλικού ή στο επιπλέον μονωτικό υλικό. Λάθη εκτύπωσης κατά τη φάση της φωτολιθογραφίας (photolithographic printing errors), ατελής χάραξη (incomplete etch) και ατελής λείανση των συνδέσεων (incomplete metal polish) είναι μερικές από τις αιτίες που δημιουργούν βραχυκυκλώματα.

Η δοκιμή, όπως φαίνεται στην Εικόνα 2.1, πραγματοποιείται εφαρμόζοντας στο ολοκληρωμένο κύκλωμα, το οποίο καλείται σε αυτή την περίπτωση *κύκλωμα υπό δοκιμή* (circuit under test-CUT), κατάλληλες τιμές ή αλλιώς ερεθίσματα (stimulus). Αν οι έξοδοι του κυκλώματος υπό δοκιμή διαφέρουν από τις αναμενόμενες τιμές¹, τότε έχει συμβεί κάποιο σφάλμα και έχει επιτευχθεί η ανίχνευση του ελαττώματος που το προκάλεσε. Οι τιμές που εφαρμόζονται στις εισόδους του κυκλώματος υπό δοκιμή καλούνται *διανύσματα δοκιμής* (test vectors) ή *πρότυπα δοκιμής* (test patterns), ενώ οι τιμές στις εξόδους του κυκλώματος καλούνται *αποκρίσεις δοκιμής* (test responses). Τα κυκλώματα τα οποία χαρακτηρίζονται εσφαλμένα έπειτα από την ολοκλήρωση της δοκιμής οδηγούνται στο στάδιο της *διάγνωσης* (diagnosis) όπου ο στόχος είναι να βρεθούν ποια ελαττώματα ή φυσικές ατέλειες οδήγησαν στην εσφαλμένη συμπεριφορά. Η διάγνωση είναι ιδιαίτερα σημαντική όταν γίνεται μετάβαση σε μια νέα τεχνολογία κατασκευής, για παράδειγμα από την τεχνολογία κατασκευής στα 130nm στα 65nm όπου δεν υπάρχει καμία γνώση για τις ατέλειες στη νέα τεχνολογία². Τα αποτελέσματα της διάγνωσης αναδεικνύουν τα προβλήματα στην τεχνολογία κατασκευής και οδηγούν στη βελτίωσή της.

Τα διανύσματα δοκιμής αναπτύσσονται βάσει *λογικών μοντέλων ελαττωμάτων* (logical fault models) τα οποία προσεγγίζουν την πραγματική συμπεριφορά των ατελειών. Η εξαγωγή διανυσμάτων δοκιμής βάσει των ατελειών δεν είναι εφικτή εξαιτίας του απαγορευτικά μεγάλου πλήθους αλλά και εξαιτίας της ιδιαίτερα πολύπλοκης συμπεριφοράς τους. Η εξαγωγή των διανυσμάτων δοκιμής είναι μια διαδικασία η οποία αποτελείται από δύο φάσεις. Η πρώτη φάση

¹Οι αναμενόμενες τιμές προκύπτουν από την εξομοίωση του κυκλώματος βάσει των τιμών στις κύριες εισόδους του κυκλώματος.

²Οι τεχνολογίες κατασκευής είναι ενδεικτικές και έχουν ξεπεραστεί. Ήδη οι μικροεπεξεργαστές κατασκευάζονται σε τεχνολογία των 45nm.

σχετίζεται με την *ενεργοποίηση του ελαττώματος* (*fault activation* ή *excitation*) ενώ η δεύτερη φάση αφορά τη *διάδοση του αποτελέσματος του ελαττώματος* (*fault effect propagation* ή *fault propagation*) στις εξόδους του κυκλώματος. Παράδειγμα ενεργοποίησης των ελαττωμάτων και της διάδοσης των αποτελεσμάτων των ελαττωμάτων παρουσιάζονται στην συνέχεια.

Τα ελαττώματα χωρίζονται σε τρεις κατηγορίες. Τα *μόνιμα ελαττώματα* (*permanent faults*) όπως φανερώνει το όνομά τους διαρκούν επ' άπειρον από τη στιγμή της εμφάνισής τους και επηρεάζουν μόνιμα τη λειτουργία του κυκλώματος [32], [33]. Η εμφάνιση των μόνιμων ελαττωμάτων αποδίδεται σε κατασκευαστικές ατέλειες, δηλαδή σε λανθασμένες συνδέσεις μεταξύ των στοιχείων του κυκλώματος, όπως βραχυκυκλώματα (*shorts*) και ανοιχτές συνδέσεις (*opens*). Σπασίματα (*cracks*) στα στοιχεία του κυκλώματος (*broken parts*) δημιουργούν επίσης μόνιμα ελαττώματα. Άλλες αιτίες εμφάνισης μόνιμων ελαττωμάτων είναι οι λανθασμένα ευθυγραμμισμένες μάσκες (*misaligned masks*) και οι εσωτερικές συνδέσεις μεταξύ των μετάλλων και του πυριτίου κατά τη διαδικασία της κατασκευής. Τέλος, τα σχεδιαστικά σφάλματα αποτελούν πηγή δημιουργίας μόνιμων ελαττωμάτων. Τα μόνιμα ελαττώματα δεν είναι δυνατό να αφαιρεθούν παρά μόνο με αντικατάσταση του ελαττωματικού κυκλώματος.

Τα *περιοδικά ελαττώματα* (*intermittent faults*) είναι ελαττώματα τα οποία διαρκούν για περιορισμένη χρονική διάρκεια και διέπονται από περιοδικότητα, δηλαδή εμφανίζονται ανά τακτά χρονικά διαστήματα. Επιπλέον, εμφανίζονται σε συγκεκριμένα σημεία του κυκλώματος [32], [33]. Η διάρκειά τους είναι τέτοια ώστε να συμπεριφέρονται ως μόνιμα ελαττώματα και να επηρεάζουν την λειτουργία του κυκλώματος. Συνήθως, τα περιοδικά ελαττώματα αποτελούν προπομπό των μόνιμων ελαττωμάτων. Η εμφάνιση των περιοδικών ελαττωμάτων αποδίδεται σε περιβαλλοντικούς παράγοντες, όπως υγρασία, αυξημένη θερμοκρασία αλλά και στην *φυσική γήρανση* (*aging*) των υλικών που συνθέτουν το κύκλωμα. Γενικότερα, ένα περιοδικό ελάττωμα μπορεί να θεωρηθεί ως μόνιμο ελάττωμα κατά τη χρονική διάρκεια της εμφάνισής του.

Τα *παροδικά ελαττώματα* (*transient faults*) εμφανίζονται με τυχαίο τρόπο και σε τυχαία σημεία σε ένα ολοκληρωμένο κύκλωμα, γεγονός που καθιστά την ανίχνευση τους αρκετά δύσκολη [32], [33]. Περιβαλλοντικοί παράγοντες αλλά και ηλεκτρομαγνητικές παρεμβολές, ο στατικός ηλεκτρισμός και απότομες μεταβολές στην τροφοδοσία ευθύνονται για την εμφάνιση των παροδικών ελαττωμάτων. Ένας άλλος παράγοντας που οδηγεί στην εμφάνιση παροδικών ελαττωμάτων είναι η κοσμική ακτινοβολία και τα σωματίδια-α. Ιδιαίτερα ευάλωτες είναι οι μνήμες τύπου RAM και τα ακολουθιακά στοιχεία flip-flop. Τα αποτελέσματα των παροδικών ελαττωμάτων στα στοιχεία αποθήκευσης είναι γνωστά ως *soft errors*.

2.1 Τρόποι Πραγματοποίησης Δοκιμής

Η δοκιμή ενός κυκλώματος μπορεί να πραγματοποιηθεί με ποικίλους τρόπους. Ανάλογα με την κατάσταση στην οποία βρίσκεται το κύκλωμα, η δοκιμή μπορεί να χωριστεί στη *δοκιμή εντός λειτουργίας (on-line testing)* και στη *δοκιμή εκτός λειτουργίας (off-line testing)*. Η δοκιμή εντός λειτουργίας πραγματοποιείται όταν το κύκλωμα βρίσκεται σε ένα σύστημα σε κατάσταση λειτουργίας. Η *ταυτόχρονη* δοκιμή εντός λειτουργίας (*concurrent on-line testing*) πραγματοποιείται ταυτόχρονα με την κανονική λειτουργία του κυκλώματος, δηλαδή όταν αυτό επεξεργάζεται δεδομένα από το περιβάλλον λειτουργίας. Η *μη ταυτόχρονη* δοκιμή στο πεδίο λειτουργίας (*non-concurrent on-line testing*) πραγματοποιείται όταν το κύκλωμα βρίσκεται σε αδρανή κατάσταση (*idle mode*). Αντίθετα, η δοκιμή εκτός λειτουργίας λαμβάνει χώρα θέτοντας το κύκλωμα σε ειδική κατάσταση δοκιμής (*test mode*). Στη δοκιμή εκτός λειτουργίας ανιχνεύονται τα μόνιμα ελαττώματα και τα περιοδικά ελαττώματα τα οποία έχουν αρκετά μεγάλη χρονική διάρκεια ώστε να θεωρούνται μόνιμα κατά τη χρονική διάρκεια της δοκιμής. Τα παροδικά ελαττώματα ανιχνεύονται μόνο κατά την ταυτόχρονη δοκιμή εντός λειτουργίας.

Η δοκιμή βασίζεται στην εφαρμογή κατάλληλων διανυσμάτων δοκιμής και στην παρατήρηση των αποκρίσεων του κυκλώματος. Η εφαρμογή των διανυσμάτων δοκιμής και η παρατήρηση των αποκρίσεων πραγματοποιείται από τον *αυτόματο εξοπλισμό δοκιμής ολοκληρωμένων κυκλωμάτων (IC Automatic Test Equipment—IC-ATE)* ή (*IC tester*). Το κύκλωμα υπό δοκιμή τοποθετείται σε ειδικές κάρτες ή αλλιώς πλακέτες φόρτωσης (*loadboards*) που συνδέονται με τον εξοπλισμό δοκιμής. Ο εξοπλισμός δοκιμής διαθέτει εσωτερική μνήμη στην οποία βρίσκονται αποθηκευμένα τα διανύσματα δοκιμής αλλά και οι ορθές αποκρίσεις (όπως αυτές εξάγονται από την προσομοίωση των διανυσμάτων δοκιμής). Στη συνέχεια, ο εξοπλισμός δοκιμής εφαρμόζει τα διανύσματα δοκιμής στο κύκλωμα και παρατηρεί τις αποκρίσεις. Οποιαδήποτε διαφορά μεταξύ των αποθηκευμένων αποκρίσεων και των αποκρίσεων στις εξόδους του κυκλώματος υπό δοκιμή σηματοδοτεί την ανίχνευση των ελαττωμάτων. Η δοκιμή που πραγματοποιείται με τη βοήθεια του εξοπλισμού δοκιμής καλείται και *εξωτερική δοκιμή (external testing)*.

Αντίθετα από την εξωτερική δοκιμή, όταν το κύκλωμα υπό δοκιμή περιέχει επιπλέον υλικό ώστε να εφαρμόζει τα διανύσματα δοκιμής στον εαυτό του αλλά και να συγκρίνει τις αποκρίσεις, τότε η δοκιμή καλείται *αυτοδοκιμή (self-test)* ή *εσωτερική δοκιμή (internal testing)* ή *ενσωματωμένη δοκιμή (embedded testing)*.

Η ταχύτητα με την οποία εφαρμόζονται τα διανύσματα δοκιμής στο κύκλωμα διαχωρίζει τη δοκιμή στη *στατική δοκιμή (DC testing, static testing)* και στη *δοκιμή στη συχνότητα λειτουργίας (at-speed testing)*. Στη στατική δοκιμή η συχνότητα εφαρμογής των διανυσμάτων δοκιμής είναι συνήθως πολύ χαμηλότερη της συχνότητας λειτουργίας του κυκλώματος. Αντίθετα, όταν τα διανύσματα εφαρμόζονται στην πραγματική λειτουργία του κυκλώματος γίνεται λόγος για δοκιμή στην (πραγματική) ταχύτητα ή συχνότητα λειτουργίας. Η πραγματοποίηση της δοκι-

μής στη συχνότητα λειτουργίας εξασφαλίζει ποιοτική δοκιμή [11], [12], [13] διότι ανιχνεύονται ελαττώματα που επηρεάζουν την συχνότητα λειτουργίας του κυκλώματος και τα οποία δεν εκδηλώνονται σε χαμηλότερες συχνότητες λειτουργίας.

Ένας άλλος διαχωρισμός της δοκιμής μπορεί να γίνει ανάλογα με τον τρόπο που παράγονται τα διανύσματα δοκιμής. Όταν τα διανύσματα βρίσκονται αποθηκευμένα σε κάποια μνήμη τότε γίνεται λόγος για *δοκιμή με αποθηκευμένα διανύσματα (stored pattern testing)*, ενώ όταν τα διανύσματα δοκιμής παράγονται κατά τη διαδικασία της δοκιμής γίνεται λόγος για *αλγοριθμική δοκιμή (algorithmic testing)*.

Η ανίχνευση των ελαττωμάτων γίνεται με την παρατήρηση των αποκρίσεων δοκιμής του κυκλώματος. Όταν το πλήθος των αποκρίσεων είναι μεγάλο, τότε εφαρμόζονται τεχνικές σύμπτυξης των αποκρίσεων και η σύγκριση γίνεται βάσει των συνεπτυγμένων αποκρίσεων. Στην περίπτωση αυτή γίνεται λόγος για *συνεπτυγμένη δοκιμή (compaction testing)*.

2.2 Αξιολόγηση και Ποιότητα Δοκιμής

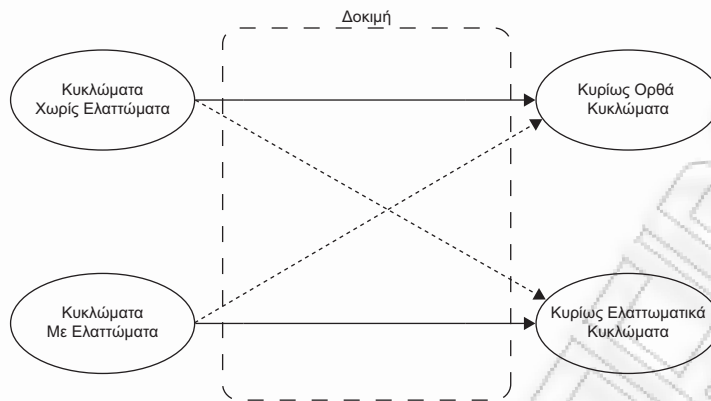
Η αξιολόγηση ενός συνόλου διανυσμάτων δοκιμής πραγματοποιείται με τη μέτρηση του ποσοστού κάλυψης ελαττωμάτων σύμφωνα με ένα μοντέλο ελαττωμάτων. Το *ποσοστό κάλυψης ελαττωμάτων (fault coverage)* ορίζεται ως η αναλογία των ελαττωμάτων που ανιχνεύονται έναντι του συνολικού αριθμού ελαττωμάτων στο κύκλωμα :

$$\text{Ποσοστό Κάλυψης Ελαττωμάτων} = \frac{\text{Ελαττώματα που ανιχνεύθηκαν}}{\text{Συνολικό Πλήθος Ελαττωμάτων}}$$

Το εφικτό/μέγιστο ποσοστό κάλυψης ελαττωμάτων μπορεί να είναι μικρότερο του 100% εξαιτίας της πιθανής ύπαρξης μη-ανιχνεύσιμων (undetectable) ελαττωμάτων. Το *πραγματικό ποσοστό κάλυψης ελαττωμάτων (effective fault coverage ή test coverage)* ορίζεται ως η αναλογία των ελαττωμάτων που ανιχνεύθηκαν από τα διανύσματα δοκιμής έναντι των ανιχνεύσιμων (detectable) ελαττωμάτων :

$$\text{Πραγματικό ποσ. κάλ.} = \frac{\text{Ελαττώματα που ανιχνεύθηκαν}}{\text{Συνολικό πλήθος ελαττωμάτων} - \text{Πλήθος μη-ανιχνεύσιμων ελαττωμάτων}}$$

Η μέτρηση του ποσοστού κάλυψης των ελαττωμάτων γίνεται προσομοιώνοντας το κύκλωμα για κάθε ελάττωμα εφαρμόζοντας το σύνολο δοκιμής σε καθένα από τα εσφαλμένα κυκλώματα. Η διαδικασία είναι γνωστή ως *προσομοίωση ελαττωμάτων (fault simulation)*. Στην πιο απλή της μορφή έχει σειριακό χαρακτήρα, όπου για κάθε ελάττωμα σχηματίζεται ένα αντίγραφο του αρχικού κυκλώματος στο οποίο εισάγεται το ελάττωμα και έπειτα εφαρμόζεται το σύνολο δοκιμής. Η σειριακή προσομοίωση δεν είναι εύχρηστη καθώς ένα ξεχωριστό αντίγραφο σχηματίζεται για κάθε ελάττωμα. Αποδοτικότερες τεχνικές είναι η *παράλληλη προσομοίωση (parallel)* [34], η *προσομοίωση με παράλληλη εφαρμογή διανυσμάτων (parallel-pattern)* [35], η *συμπερασματική (deductive)* [36] και η *σύγχρονη (concurrent)* προσομοίωση ελαττωμάτων



Εικόνα 2.2: Ο διαχωρισμός των ολοκληρωμένων κυκλωμάτων σε ορθά και ελαττωματικά.

[37]. Όλες αυτές οι τεχνικές μειώνουν το χρόνο προσομοίωσης ελαττωμάτων καθώς επεξεργάζονται είτε περισσότερα του ενός διανύσματα δοκιμής είτε περισσότερες από μια εκδοχές του κυκλώματος με διαφορετικά ελαττώματα.

Το ποσοστό κάλυψης ελαττωμάτων αποτελεί το μέτρο αξιολόγησης των διανυσμάτων δοκιμής αλλά από μόνο του δεν επαρκεί για την εξασφάλιση της ποιότητας της δοκιμής. Η εσοδεία και ο ρυθμός απόρριψης των κυκλωμάτων από την αγορά είναι δύο παράμετροι που πρέπει να ληφθούν υπόψη για την ποιότητα της δοκιμής.

Η εσοδεία (*yield*) της κατασκευής κυκλωμάτων ορίζεται ως η αναλογία των κυκλωμάτων τα οποία είναι ελεύθερα από ελαττώματα (*defect free*) έναντι των κυκλωμάτων που κατασκευάστηκαν. Επειδή είναι πολύ δύσκολο να πραγματοποιηθεί δοκιμή για όλες τις φυσικές ατέλειες η εσοδεία προσεγγίζεται με την αναλογία των κυκλωμάτων που πέρασαν επιτυχώς τη δοκιμή έναντι των κυκλωμάτων που κατασκευάστηκαν :

$$\text{Εσοδεία} = \frac{\text{πλήθος αποδεκτών κυκλωμάτων}}{\text{κυκλώματα που κατασκευάστηκαν συνολικά}}$$

Αναμφίβολα, κάποια κυκλώματα που περιέχουν ελαττώματα θα περάσουν επιτυχώς τη διαδικασία της δοκιμής, ενώ θα υπάρξουν και κυκλώματα τα οποία ενώ λειτουργούν σύμφωνα με τις προδιαγραφές θα ανιχνευθούν ως ελαττωματικά. Αυτό συμβαίνει διότι τα μοντέλα ελαττωμάτων προσεγγίζουν τη συμπεριφορά των ατελειών και δεν την περιγράφουν επακριβώς και επιπλέον η εξαγωγή διανυσμάτων μπορεί να μην είναι υπολογιστικά εφικτή για όλα τα ελαττώματα. Η Εικόνα 2.2 παρουσιάζει με γραφικό τρόπο το γεγονός πως η δοκιμή μπορεί να χαρακτηρίσει ορθά κυκλώματα ως ελαττωματικά αλλά και το αντίστροφο. Οι διακεκομμένες γραμμές παρουσιάζουν την λανθασμένη κατηγοριοποίηση των κυκλωμάτων.

Ο ρυθμός απόρριψης (*reject rate*), ο οποίος είναι γνωστός και ως *στάθμη/επίπεδο ατελειών* (*defect level*) ορίζεται ως η αναλογία των εσφαλμένων κυκλωμάτων που πέρασαν τη δοκιμή

έναντι των συνολικών που πέρασαν επιτυχώς τη δοκιμή :

$$\text{Στάθμη ατελειών} = \frac{\text{πλήθος εσφαλμένων κυκλωμάτων που δοκιμάστηκαν επιτυχώς}}{\text{πλήθος κυκλωμάτων που δοκιμάστηκαν επιτυχώς}}$$

Η στάθμη ατελειών είναι ενδεικτική της ποιότητας της δοκιμής. Γενικά, μια στάθμη ατελειών της τάξης των 500 κυκλωμάτων ανά εκατομμύριο κυκλωμάτων (parts per million-PPM ή defects per million-DPM) θεωρείται αποδεκτή. Όταν η στάθμη ατελειών είναι της τάξης των 100 κυκλωμάτων ανά εκατομμύριο κυκλωμάτων (100 PPM) τότε λέγεται πως η δοκιμή είναι υψηλής ποιότητας. Η στάθμη ατελειών γίνεται γνωστή αφού τα κυκλώματα φθάσουν στην αγορά, τοποθετηθούν σε συστήματα και παρουσιάσουν αστοχίες στις υπηρεσίες που πρέπει να παρέχουν. Επιπλέον, πριν τα εσφαλμένα κυκλώματα επιστρέψουν στο εργοστάσιο κατασκευής για τη διαδικασία της διάγνωσης μπορεί να εμφανίσουν και άλλες ατέλειες λόγω της πιθανής κακομεταχείρισης κατά την επιστροφή. Η στάθμη ατελειών από μόνη της δε μπορεί να παρέχει μια χρήσιμη μέτρηση για την ποιότητα της δοκιμής.

Το ποσοστό κάλυψης ελαττωμάτων (fault coverage-FC) συνδέεται με την εσοδεία, Y, και με το ρυθμό απόρριψης ή αλλιώς τη στάθμη ατελειών, DL, σύμφωνα με την εξίσωση [38]

$$\text{Στάθμη ατελειών} = 1 - \text{Εσοδεία}^{(1 - \text{Ποσοστό κάλυψης ελαττωμάτων})}$$

Η εξίσωση αποτελεί ένα χρήσιμο τρόπο μέτρησης της ποιότητας της δοκιμής. Υψηλά ποσοστά κάλυψης ελαττωμάτων οδηγούν σε μείωση του ρυθμού απόρριψης. Αν μια διαδικασία κατασκευής παρουσιάζει εσοδεία 50% και το ποσοστό κάλυψης ελαττωμάτων είναι 80%, ο ρυθμός απόρριψης είναι 0,1295, δηλαδή το 12,95% των κυκλωμάτων (δηλαδή 129500 DPM) που στέλνονται στην αγορά περιέχουν ελαττώματα. Αυξάνοντας το ποσοστό κάλυψης στο 90%, ο ρυθμός απόρριψης μειώνεται στο 6,7%, ενώ περαιτέρω αύξηση του ποσοστού κάλυψης ελαττωμάτων στο 99% ο ρυθμός απόρριψης μειώνεται στο 0,69% (δηλαδή 6900 DPM).

Η εξίσωση δεν αντιπροσωπεύει πάντα την ποιότητα της δοκιμής, διότι δε λαμβάνει υπόψη της την ταχύτητα με την οποία εφαρμόζονται τα διανύσματα δοκιμής. Στην πραγματικότητα η δοκιμή στην πραγματική ταχύτητα λειτουργίας παρέχει υψηλότερης ποιότητας δοκιμή σε σχέση με τη δοκιμή σε αρκετά χαμηλότερη συχνότητα παρόλο που η εξίσωση εκτιμά τον ίδιο ρυθμό απόρριψης.

2.3 Κόστος και Δυσκολίες Δοκιμής

Η πρόοδος στην τεχνολογία κατασκευής CMOS VLSI έχει επιτρέψει την υλοποίηση πολύπλοκων ολοκληρωμένων κυκλωμάτων που περιέχουν δισεκατομμύρια τρανζίστορ και λειτουργούν σε πολύ υψηλές ταχύτητες, της τάξης των GHz. Οι τεχνολογικές ανακαλύψεις επηρεάζουν αρνητικά το πείραμα της δοκιμής και εισάγουν νέες προκλήσεις για την αποτελεσματική δοκιμή. Η Intel αναφέρει πως το μεγαλύτερο τμήμα του κόστους κατασκευής ενός προϊόντος

δαπανάται στην επαλίθευση του σχεδιασμού και στη δοκιμή, και όχι στην αξία 2 δισεκατομμυρίων δολαρίων γραμμή παραγωγής [9]. Το κόστος της δοκιμής ενός ολοκληρωμένου κυκλώματος μπορεί να φθάσει ακόμα και το 70% του συνολικού κόστους κατασκευής του. Στη συνέχεια παρουσιάζονται μερικές από τις σημαντικότερες προκλήσεις της δοκιμής των κυκλωμάτων όπως αυτές εισάγονται από την πρόοδο της τεχνολογίας κατασκευής CMOS VLSI.

- Καθώς τα κυκλώματα μεγαλώνουν σε μέγεθος και πολυπλοκότητα αυξάνεται ο αριθμός των κυρίων εισόδων καθώς και το πλήθος των στοιχείων αποθήκευσης (flip-flop). Η εφαρμογή όλων των δυνατών συνδυασμών στις εισόδους του κυκλώματος δεν είναι σε καμία περίπτωση εφικτή. Το σύνολο διανυσμάτων δοκιμής πρέπει να είναι όσο το δυνατό μικρότερο ώστε ο χρόνος δοκιμής στον εξοπλισμό δοκιμής (IC-ATE, IC tester) να ελαχιστοποιείται. Τα κυκλώματα με μεγάλο ακολουθιακό βάθος παρουσιάζουν πολύ μεγάλη δυσκολία στην δοκιμή τους. Σε αυτή την περίπτωση οι αλγόριθμοι εξαγωγής διανυσμάτων έχουν να αντιμετωπίσουν ένα μεγάλο σύνολο τιμών, που αποτελείται από το συνδυασμό των 2^N δυνατών τιμών στις εισόδους, και τις $2^{N_{ff}}$ καταστάσεις στις οποίες μπορεί να βρίσκεται το κύκλωμα, όπου N_{ff} το πλήθος των flip-flop. Ο χρόνος εξαγωγής διανυσμάτων δοκιμής στη χειρότερη περίπτωση έχει εκθετική συμπεριφορά.

Η δοκιμή δυσκολεύει καθώς ο χρόνος εξαγωγής διανυσμάτων μεγαλώνει και παράλληλα η επίτευξη υψηλών ποσοστών κάλυψης γίνεται ολοένα και πιο δύσκολη. Η χαμηλή δοκιμαστικότητα αντιμετωπίζεται με σχεδίαση για αυξημένη δοκιμαστικότητα (εξετάζεται στις επόμενες ενότητες) η οποία επηρεάζει το κόστος του κυκλώματος.

- Η συχνότητα λειτουργίας του συστήματος δοκιμής αποτελεί ανασχετικό παράγοντα για τη δοκιμή των κυκλωμάτων. Τα συστήματα δοκιμής έχουν πολύ υψηλό κόστος αγοράς και συντήρησης. Το κόστος αγοράς ανά ακροδέκτη του συστήματος δοκιμής με συχνότητα λειτουργίας 1GHz ανέρχονταν στα 3000 δολάρια. Ο εξοπλισμός δοκιμής έχει συνήθως 1000 τέτοιους ακροδέκτες. Η συχνότητα λειτουργίας των κυκλωμάτων έχει ξεπεράσει κατά πολύ τη συχνότητα του 1GHz. Έτσι τίθεται το κείμερο ερώτημα της πραγματοποίησης της δοκιμής στην πραγματική συχνότητα λειτουργίας (at-speed) η οποία παρέχει υψηλής ποιότητας δοκιμή. Η κατασκευή εξοπλισμού δοκιμής σε υψηλότερες συχνότητες λειτουργίας δεν είναι οικονομικά εφικτή. Ήδη το κόστος του εξοπλισμού δοκιμής με δυνατότητα εφαρμογής διανυσμάτων δοκιμής και παρατήρησης αποκρίσεων σε συχνότητες της τάξης του GHz φθάνει ακόμα και τα 5 εκατομμύρια δολάρια. Επιπλέον, σε πολύ υψηλές συχνότητες λειτουργίας του εξοπλισμού δοκιμής τίθεται το ερώτημα της ακρίβειας των μετρήσεων. Ανακριβείς μετρήσεις μειώνουν την εσοδεία. Η αυτοδοκιμή (παρουσιάζεται σε επόμενες ενότητες) λύνει τα περισσότερα προβλήματα του εξοπλισμού δοκιμής αυξάνοντας όμως το κόστος σε υλικό.

- Τα ολοκληρωμένα κυκλώματα κατασκευάζονται σε ολοένα και μικρότερες διαστάσεις. Νέα ψεγάδια και ατέλειες εμφανίζονται. Η διαδικασία της κατασκευής γίνεται πιο ευαίσθητη σε αιωρούμενα σωματίδια. Έτσι, λαμβάνεται μέριμνα ώστε οι χώροι κατασκευής να περιέχουν μικρό αριθμό σωματιδίων ανά μονάδα όγκου αέρα (*καθαρός χώρος*). Επειδή ο αριθμός των αιωρούμενων σωματιδίων δεν μπορεί να μηδενισθεί πάντα υπάρχει η πιθανότητα επικάλυψης δημιουργώντας βραχυκυκλώματα. Νέα μοντέλα ελαττωμάτων χρειάζονται για να περιγράψουν την συμπεριφορά των ατελειών. Το μοντέλο ελαττωμάτων μόνιμης τιμής (*stuck-at fault model*) αδυνατεί να περιγράψει τη συμπεριφορά των ατελειών σε διαστάσεις μερικών δεκάδων νανομέτρων. Τα μοντέλα ελαττωμάτων καθυστέρησης (*delay*) και γεφύρωσης (*bridging*) αντιπροσωπεύουν με μεγαλύτερη ακρίβεια την συμπεριφορά των ατελειών που εμφανίζονται σε διαστάσεις κατασκευής μερικών νανομέτρων. Η εξαγωγή διανυσμάτων βάσει των μοντέλων αυτών είναι δυσκολότερη σε σχέση με το μοντέλο ελαττωμάτων μόνιμης τιμής. Επιπλέον, αυξάνεται το συνολικό πλήθος των διανυσμάτων δοκιμής αυξάνοντας το χρόνο δοκιμής και κατά συνέπεια το κόστος. Αξιοσημείωτο είναι το γεγονός πως ακόμα και αν εφαρμοστούν όλα τα δυνατά διανύσματα δοκιμής στις εισόδους του κυκλώματος δεν είναι εγγυημένη η ανίχνευση των ελαττωμάτων καθυστέρησης και γεφύρωσης.
- Ο νόμος του Moore, [39], [40] δείχνει πως περίπου κάθε 18 μήνες το πλήθος των τρανζίστρων που τοποθετούνται σε ένα ολοκληρωμένο κύκλωμα διπλασιάζεται. Έχει παρατηρηθεί πως η αναλογία των ακροδεκτών έναντι των λογικών πυλών ενός κυκλώματος έχει καθοδική τάση. Αυτό σημαίνει πως η πρόσβαση στις υπομονάδες ενός κυκλώματος δυσχεραίνεται, μειώνοντας την ελεγχσιμότητα (*controllability*)³ και την παρατηρησιμότητα (*observability*)⁴, δηλαδή τη δοκιμασιμότητα. Άμεση συνέπεια είναι η δυσκολότερη επίτευξη υψηλών ποσοστών κάλυψης ελαττωμάτων.

Οι παραπάνω παράγοντες επηρεάζουν το πείραμα της δοκιμής με αρνητικό τρόπο αυξάνοντας το κόστος του κυκλώματος και επηρεάζουν τον χρόνο-προς-την αγορά (*time-to-market*). Το κόστος της δοκιμής είναι συνάρτηση πολλών παραμέτρων. Ενδεικτικά, αναφέρουμε το κόστος ανάπτυξης των διανυσμάτων δοκιμής, το κόστος της σχεδίασης για αυξημένη δοκιμασιμότητα, την επιβάρυνση του μεγέθους του κυκλώματος και τη μείωση της ταχύτητας, και το κόστος από την καθυστερημένη είσοδο του προϊόντος στην αγορά [41]. Αν το αρχικό κύκλωμα έχει κόστος κατασκευής, C_O , οι τροποποιήσεις για σχεδίαση με αυξημένη δοκιμασιμότητα μειώνουν την εσοδεία Y και αυξάνουν το μέγεθος του κυκλώματος από A_O σε A_E . Στην περίπτωση αυτή

³Η ελεγχσιμότητα αναφέρεται στην ευκολία με την οποία μπορεί μια γραμμή να λάβει την τιμή 0 ή 1 από τις κύριες εισόδους.

⁴Η παρατηρησιμότητα αναφέρεται στην ευκολία με την οποία μια λανθασμένη τιμή σε μία γραμμή του κυκλώματος εξαιτίας ενός ελαττώματος, μπορεί να παρατηρηθεί στις κύριες εξόδους.

το κόστος για το νέο κύκλωμα με αυξημένη δοκιμαστικότητα, C_A , είναι :

$$C_A = C_O \cdot (A_E/A_O \cdot Y^{1-\sqrt{A_E/A_O}} - 1).$$

Το ακόλουθο παράδειγμα παρουσιάζει την αύξηση του κόστους για ένα κύκλωμα του οποίου το μέγεθος αυξάνεται κατά 15% λόγω σχεδίασης με αυξημένη δοκιμαστικότητα. Το αρχικό κόστος κατασκευής του κυκλώματος είναι €50 και το 50% του κόστους αποδίδεται στην κατασκευή του κυκλώματος στο πλακίδιο πυριτίου. Αν η εσοδεία είναι 60% τότε το κόστος αυξάνει κατά :

$$C_A = €50 \cdot 50\% \cdot (1,15 \cdot 60\%^{1-\sqrt{1,15}} - 1) = €4,83.$$

ανά κύκλωμα.

Επίσης όταν ένα κύκλωμα σχεδιάζεται με αυξημένη δοκιμαστικότητα μειώνεται η συχνότητα λειτουργίας του. Αν το κόστος του κυκλώματος είναι P_O και η συχνότητα του κυκλώματος από F_O γίνει F_D τότε το κόστος του νέου κυκλώματος είναι :

$$C_S = P_O \cdot (1 - F_D/F_O).$$

Έτσι, αν το κόστος ενός κυκλώματος είναι 1.000 ευρώ και η σχεδίαση για αυξημένη δοκιμαστικότητα μειώσει την συχνότητα λειτουργίας κατά 5% τότε το κόστος του νέου κυκλώματος αυξάνει κατά:

$$C_S = €1000 \cdot (1 - 0.95) = 50€.$$

Πιθανές καθυστερήσεις στην εμφάνιση του προϊόντος στην αγορά εξαιτίας της δοκιμής ή και μιας επανασχεδίασης για αυξημένη δοκιμαστικότητα συνεπάγονται απώλειες κερδών από την πώληση του προϊόντος. Η απώλεια των εσόδων, (lost revenue- LR), μπορεί να υπολογιστεί σύμφωνα με την παρακάτω εξίσωση:

$$LR = ER \cdot \frac{D(3W - D)}{2W^2},$$

όπου ER (expected revenue) είναι τα αναμενόμενα έσοδα, D είναι ο χρόνος καθυστέρησης στην αγορά και W είναι το μέσο του χρονικού διαστήματος στο οποίο το προϊόν θα διατίθεται στην αγορά. Η παραπάνω εξίσωση γίνεται πιο κατανοητή λαμβάνοντας υπόψη το εξής παράδειγμα. Μια εταιρία κατασκευάζει συνολικά 2000 ολοκληρωμένα κυκλώματα καθένα από τα οποία έχει κόστος κατασκευής €10.000. Αν το χρονικό διάστημα στο οποίο το προϊόν θα παραμείνει στην αγορά είναι 36 μήνες, δηλαδή $W = 36/2$ μήνες, τότε μια πιθανή καθυστέρηση κατά δύο μήνες, $D = 2$ μήνες, εξαιτίας της δοκιμής, συνεπάγεται απώλειες εσόδων

$$LR = €20.000.000 \frac{2(3 \cdot 18 - 2)}{2 \cdot 18^2} = €3.209.887$$

ή περίπου 1.600 ευρώ ανά ολοκληρωμένο κύκλωμα.

2.4 Μοντέλα ελαττωμάτων

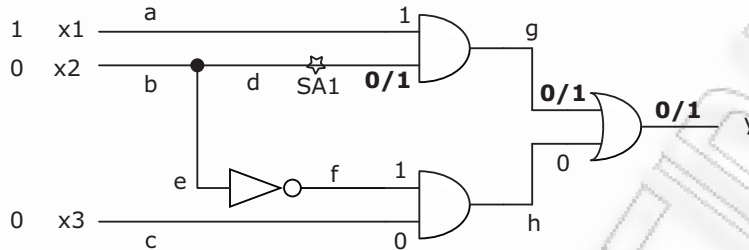
Τα λογικά μοντέλα ελαττωμάτων (*logical fault models*) προσεγγίζουν την συμπεριφορά των φυσικών ατελειών καθιστώντας υπολογιστικά εφικτή την εξαγωγή διανυσμάτων δοκιμής. Βέβαια, τα μοντέλα ελαττωμάτων αποτελούν μια προσέγγιση και δε μπορούν να περιγράψουν με απόλυτη ακρίβεια τη συμπεριφορά των ψεγαδιών/ατελειών σε ένα κύκλωμα με αποτέλεσμα κυκλώματα που περνούν επιτυχώς τη δοκιμή και αποστέλλονται στην αγορά να περιέχουν ατέλειες. Στην προσπάθεια να προσεγγιστεί όσο το δυνατό καλύτερα η συμπεριφορά των ατελειών έχουν αναπτυχθεί διάφορα μοντέλα ελαττωμάτων. Μια κατηγοριοποίηση μπορεί να γίνει βάσει του αφαιρετικού επιπέδου (*abstraction level*) στο οποίο περιγράφεται ένα κύκλωμα. Τα δομικά μοντέλα ελαττωμάτων (*structural fault models*) αναφέρονται στον τρόπο με τον οποίο επηρεάζεται η δομή του κυκλώματος, ενώ τα λειτουργικά μοντέλα ελαττωμάτων (*functional fault models*) εκφράζουν πώς αλλοιώνεται η λειτουργία (η λογική συνάρτηση) ή η συμπεριφορά του κυκλώματος.

Το πιο διαδεδομένο μοντέλο ελαττωμάτων που χρησιμοποιείται στη βιομηχανία, είναι το μοντέλο μόνιμης τιμής μιας μόνο γραμμής (*single stuck-at fault model*) [42]. Το μοντέλο μόνιμης τιμής ορίζει πως μία και μόνο μία γραμμή του κυκλώματος βρίσκεται μόνιμα είτε στην τιμή 0 (*stuck-at-0-SA0*) είτε στην τιμή 1 (*stuck-at-1-SA1*). Αν ένα κύκλωμα αποτελείται από k γραμμές το πλήθος των ελαττωμάτων μόνιμης τιμής είναι $2 \times k$. Το πλήθος μπορεί να μειωθεί κατά 40%-50% λαμβάνοντας υπόψη πως πολλά από τα παραπάνω ελαττώματα είναι ισοδύναμα (*equivalent*)⁵. Στην πραγματικότητα, περισσότερες από μία γραμμές μπορεί να βρίσκονται μόνιμα σε μία τιμή. Το μοντέλο μόνιμης τιμής πολλαπλών γραμμών (*multiple stuck-at fault model*) εκφράζει την παραπάνω κατάσταση. Αν ένα κύκλωμα έχει k γραμμές, το συνολικό πλήθος ελαττωμάτων σε πολλαπλές γραμμές είναι $3^k - 1$. Είναι ευνόητο πως ακόμα και για μικρά κυκλώματα το πλήθος των ελαττωμάτων μόνιμης τιμής σε πολλαπλές γραμμές είναι απαγορευτικά μεγάλο για εξαγωγή διανυσμάτων δοκιμής. Ευτυχώς, έχειδειχθεί πως σε μεγάλα κυκλώματα τα διανύσματα δοκιμής για ελαττώματα μόνιμης τιμής μιας μόνο γραμμής ανιχνεύουν τουλάχιστον το 99,6% των ελαττωμάτων μόνιμης τιμής πολλαπλών γραμμών [43], [44].

Η Εικόνα 2.3 παρουσιάζει ένα κύκλωμα του οποίου η γραμμή d έχει ένα ελάττωμα μόνιμης τιμής 1, δηλαδή $d: SA-1$. Η ενεργοποίηση του ελαττώματος, απαιτεί την ανάθεση της αντίθετης τιμής από αυτή στην οποία είναι προσκολλημένη η γραμμή d , δηλαδή στην είσοδο x_2 πρέπει να τεθεί η τιμή 0. Για τη διάδοση του αποτελέσματος του ελαττώματος $d: SA-1$, οι είσοδοι x_1 , x_3 πρέπει να λάβουν τις τιμές 1 και 0, αντίστοιχα. Το διάνυσμα δοκιμής $(x_1, x_2, x_3) = (1, 0, 0)$ ανιχνεύει το ελάττωμα $d: SA-1$. Ο συμβολισμός 0/1 συμβολίζει πως η αναμενόμενη (ορθή) τιμή της γραμμής είναι 0 αλλά εξαιτίας ελαττώματος έχει αλλάξει τιμή και έχει τιμή 1. Αντίστοιχα

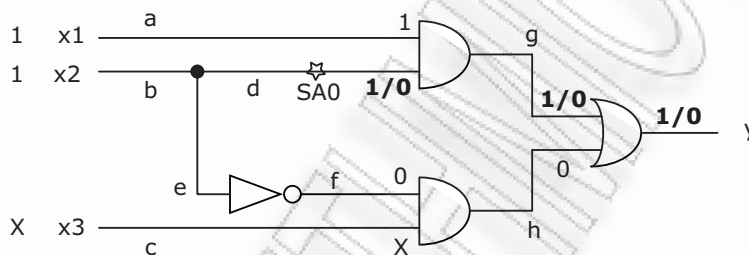
⁵ Δύο ελαττώματα είναι ισοδύναμα όταν οι αποκρίσεις του κυκλώματος για κάθε ελάττωμα είναι ακριβώς ίδιες. Για παράδειγμα το ελάττωμα SA0 στην είσοδο μιας πύλης AND είναι ισοδύναμο με το ελάττωμα SA0 στην έξοδό της.

και για το συμβολισμό 1/0.



Εικόνα 2.3: Παράδειγμα ελαττώματος προσκόλλησης της γραμμής d στην τιμή 1.

Το ελάττωμα προσκόλλησης d: SA-0, στην Εικόνα 2.4 ανιχνεύεται με το διάνυσμα δοκιμής $(x_1, x_2, x_3)=(1, 1, X)$, όπου X είναι αδιάφορος όρος εισόδου. Οποιαδήποτε τιμή αν λάβει η είσοδος x_3 το ελάττωμα d:SA-0 θα ανιχνευθεί.

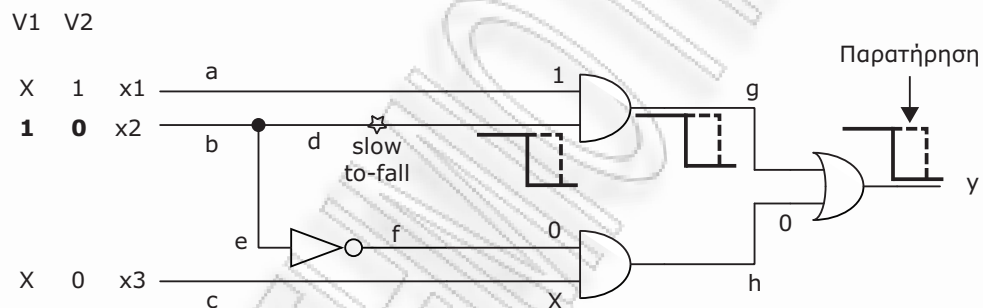


Εικόνα 2.4: Παράδειγμα ελαττώματος προσκόλλησης της γραμμής d στην τιμή 0.

Το μοντέλο ελαττωμάτων μόνιμης τιμής αποτελεί το δημοφιλέστερο μοντέλων ελαττωμάτων και έτσι έχουν αναπτυχθεί αρκετοί αλγόριθμοι εξαγωγής διανυσμάτων για αυτό. Οι αλγόριθμοι *D-Alg* [45], *PODEM* [46], *FAN* [47] εξάγουν διανύσματα δοκιμής για ελαττώματα μόνιμης τιμής, σε συνδυαστικά κυκλώματα.

Ένα κύκλωμα ελεύθερο από ατέλειες δεν αρκεί να παράγει ορθές τιμές αλλά πρέπει να τις παράγει και στις σωστές χρονικές στιγμές. Τα ελαττώματα καθυστέρησης (*delay faults*) προσεγγίζουν τη συμπεριφορά των ατελειών που επηρεάζουν την χρονικά ορθή λειτουργία του κυκλώματος. Τα ελαττώματα καθυστέρησης γνωρίζουν μια ολοένα αυξανόμενη αποδοχή καθώς τα κυκλώματα λειτουργούν σε πολύ υψηλές συχνότητες. Το μοντέλο ελαττώματος καθυστέρησης μετάβασης (*transition delay fault model*) [48] υποθέτει πως μόνο μία πύλη σε ένα κύκλωμα εμφανίζει ένα ελάττωμα καθυστέρησης που είτε αποτρέπει την μετάβαση από την τιμή 0 στην τιμή 1 (*slow-to-rise*) είτε την μετάβαση από την τιμή 1 στην τιμή 0 (*slow-to-fall*). Επίσης το μοντέλο υποθέτει πως η καθυστέρηση είναι τόσο μεγάλη, που ανεξάρτητα από το μονοπάτι που ακολουθεί η μετάβαση, δε θα παρατηρηθεί στις κύριες εξόδους του κυκλώματος τη χρονική στιγμή που γίνεται η παρατήρηση.

Η Εικόνα 2.5 παρουσιάζει την ανίχνευση του ελαττώματος καθυστέρησης μετάβασης $1 \rightarrow 0$ στην γραμμή d. Απαιτούνται δυο διανύσματα δοκιμής για την ανίχνευση του ελαττώματος. Αρχικά το διάνυσμα V1 θέτει την τιμή 1 στην είσοδο x2, οπότε και η γραμμή d λαμβάνει την τιμή 1. Στη συνέχεια το διάνυσμα δοκιμής θέτει την τιμή 0 στην είσοδο x2 δημιουργώντας μια μετάβαση $1 \rightarrow 0$ στην γραμμή d. Οι τιμές των εισόδων x1, x3 λαμβάνουν κατάλληλες τιμές ώστε η μετάβαση να διαδοθεί στην κύρια έξοδο y. Εξαιτίας του ελαττώματος καθυστέρησης μετάβασης η μετάβαση $1 \rightarrow 0$ αργεί να ολοκληρωθεί (η καθυστέρηση παρουσιάζεται με διακεκομμένη γραμμή στην Εικόνα 2.5). Τη χρονική στιγμή της παρατήρησης, η τιμή της κύριας εξόδου είναι 1 ενώ θα έπρεπε να είναι 0. Η παρατήρηση αυτή οδηγεί στο συμπέρασμα πως υπάρχει κάποιο ελάττωμα καθυστέρησης μετάβασης στο κύκλωμα, επομένως το ελαττώματα καθυστέρησης μετάβασης $1 \rightarrow 0$ στην γραμμή d έχει ανιχνευθεί.



Εικόνα 2.5: Παράδειγμα ελαττώματος καθυστέρησης μετάβασης $1 \rightarrow 0$ (slow-to-fall-STF) της γραμμής d

Το μοντέλο ελαττώματος καθυστέρησης πύλης (*gate delay fault model*) [49] επίσης υποθέτει πως υπάρχει ένα μόνο ελάττωμα καθυστέρησης σε μία μόνο πύλη που καθυστερεί την μετάβαση $0 \rightarrow 1$ και $1 \rightarrow 0$. Αντίθετα από το μοντέλο καθυστέρησης μετάβασης, η χρονική καθυστέρηση δεν είναι βέβαιο ότι θα παρατηρηθεί στις κύριες εξόδους. Η παρατήρηση της καθυστέρησης εξαρτάται από την καθυστέρηση του μονοπατιού από το οποίο θα διαδοθεί η μετάβαση. Το μοντέλο ελαττώματος καθυστέρησης πύλης βασίζεται στα φυσικά χαρακτηριστικά καθυστέρησης των πυλών. Το πλήθος των ελαττωμάτων καθυστέρησης σύμφωνα με τα παραπάνω μοντέλα είναι ανάλογο των γραμμών του κυκλώματος.

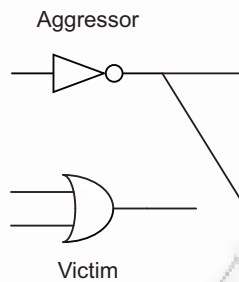
Το μοντέλο ελαττώματος καθυστέρησης μονοπατιού (*path delay fault model*) [50] θεωρεί πως πολλαπλά ελαττώματα καθυστέρησης στις γραμμές και τις πύλες ενός μονοπατιού μπορεί να επηρεάσουν μια μετάβαση στο μονοπάτι αυτό. Τα ελαττώματα καθυστέρησης μονοπατιού προσεγγίζουν με μεγαλύτερη ακρίβεια τις ατέλειες που οδηγούν σε καθυστέρηση σε σχέση με τα προηγούμενα μοντέλα. Καθώς το πλήθος των μονοπατιών σε ένα κύκλωμα είναι πολύ μεγάλο και αυξάνει εκθετικά με το πλήθος των πυλών, η εξαγωγή διανυσμάτων δοκιμής είναι πολύ πιο δύσκολη.

Τα παραπάνω μοντέλα χρησιμοποιούνται όταν η περιγραφή του κυκλώματος δίνεται σε

λογικές πύλες (netlist of logic gates). Όταν η περιγραφή του κυκλώματος γίνεται σε τρανζίστορ το μοντέλο ελαττώματος μόνιμης τιμής δεν επαρκεί καθώς μια πύλη αποτελείται από περισσότερα από ένα τρανζίστορ. Το μοντέλο ελαττώματος μόνιμα ανοικτού τρανζίστορ (*stuck-open/stuck-off transistor fault model*) προσεγγίζει τη συμπεριφορά των ατελειών που δεν επιτρέπουν στο τρανζίστορ να άγει ενώ το μοντέλο ελαττωμάτων μόνιμα κλειστού τρανζίστορ (*stuck-short/stuck-on fault model*) προσεγγίζει τη συμπεριφορά των ατελειών που δημιουργούν βραχυκύκλωμα μεταξύ της τάσης και της γείωσης. Τα ελαττώματα μόνιμα ανοικτού τρανζίστορ σε ένα συνδυαστικό κύκλωμα οδηγούν σε ακολουθιακή συμπεριφορά του κυκλώματος. Από την άλλη πλευρά τα ελαττώματα μόνιμα κλειστού τρανζίστορ δεν είναι πάντα βέβαιο ότι θα διαδοθούν, διότι η λανθασμένη τάση στην έξοδο του τρανζίστορ μπορεί να μη γίνει αντιληπτή από την πύλη που ακολουθεί. Η ανίχνευση των ελαττωμάτων μόνιμα κλειστού τρανζίστορ γίνεται με μέτρηση του ρεύματος που αντλεί το τρανζίστορ. Η δοκιμή αυτή είναι γνωστή ως δοκιμή I_{DDQ} (I_{DDQ} testing) [51], [52].

Καθώς οι διαστάσεις κατασκευής των κυκλωμάτων ολοένα και μικραίνουν είναι πολύ πιθανό να δημιουργηθούν βραχυκυκλώματα μεταξύ των γραμμών που συνδέουν τα στοιχεία του κυκλώματος. Το μοντέλο ελαττωμάτων γεφύρωσης (*bridging fault model*) [53] προσεγγίζει την συμπεριφορά του κυκλώματος όταν οι ατέλειες δημιουργούν βραχυκυκλώματα. Ελαττώματα γεφύρωσης μπορεί να εξαχθούν είτε στο επίπεδο περιγραφής του κυκλώματος σε λογικές πύλες είτε στο επίπεδο περιγραφής με τρανζίστορ. Η πιο αποδοτική εξαγωγή των ελαττωμάτων γεφύρωσης επιτυγχάνεται όταν το κύκλωμα περιγράφεται σε επίπεδο διάταξης (layout), όπου είναι γνωστό ποιες γραμμές βρίσκονται σε μικρή απόσταση. Μια αρχική προσέγγιση μοντελοποίησης των βραχυκυκλωμάτων ήταν η "τοποθέτηση" μιας εικονικής πύλης AND ή OR στο σημείο του βραχυκυκλώματος με την έξοδο της πύλης να οδηγεί τα δύο βραχυκυκλωμένα σήματα. Όμως αυτή η μοντελοποίηση δεν είναι ακριβής για την τεχνολογία CMOS. Στην τεχνολογία CMOS VLSI η προσέγγιση της συμπεριφοράς των βραχυκυκλωμάτων γίνεται με τα κυρίαρχα ελαττώματα γεφύρωσης (*dominant bridging fault*) όπου η τιμή της μιας γραμμής κυριαρχεί (*dominates*) στην τιμή της άλλης. Η συμπεριφορά των βραχυκυκλωμάτων, αν και ακριβέστερη, είναι πιο δύσκολο να παρατηρηθεί αφού η παρατήρηση γίνεται μόνο στην γραμμή η οποία είναι το θύμα (*victim*) στο βραχυκύκλωμα. Η Εικόνα 2.6 παρουσιάζει ένα κυρίαρχο ελάττωμα γεφύρωσης όπου η έξοδος της πύλης NOT κυριαρχεί (*dominates*) στην έξοδο της πύλης OR.

Σχεδόν όλα τα παραπάνω μοντέλα ορίζουν ελαττώματα τα οποία μπορούν να παρατηρηθούν μετρώντας την τάση στις εξόδους του κυκλώματος. Όμως, υπάρχουν ελαττώματα τα οποία δεν μπορούν να ανιχνευθούν μετρώντας απλά τάσεις στις εξόδους, όπως τα ελαττώματα μόνιμα κλειστού τρανζίστορ. Η ανίχνευση τέτοιων ελαττωμάτων απαιτεί την μέτρηση ηλεκτρικών ρευμάτων η οποία είναι η βάση της μεθόδου δοκιμής I_{DDQ} [51], [52]. Τα κυκλώματα που υλοποιούνται με την τεχνολογία CMOS έχουν το χαρακτηριστικό πως όταν βρίσκονται σε αδρανή κατάσταση (*quiescent state*) αντλούν πάρα πολύ μικρή ποσότητα ρεύματος, η οποία



Εικόνα 2.6: Παράδειγμα κυρίαρχου ελαττώματος γεφύρωσης.

είναι γνωστή ως *ρεύμα διαρροής (leakage current)*. Ελαττώματα όπως το ελάττωμα μόνιμα κλειστού τρανζίστορ αλλά και *εσωτερικά ελαττώματα γεφύρωσης** δημιουργούν αγωγή μονοπάτια μεταξύ της τάσης (V_{dd}) και της γείωσης. Έτσι το ρεύμα που αντλεί το τρανζίστορ σε κατάσταση ηρεμίας είναι πολύ μεγαλύτερο σε σχέση με το ρεύμα που θα αντλούσε αν δεν υπήρχαν ελαττώματα. Κατά τη δοκιμή με μέτρηση ρευμάτων, τίθεται εκ' των προτέρων ένα κατώφλι (threshold) ρεύματος. Όσα κυκλώματα παρουσιάζουν ρεύματα διαρροής μεγαλύτερα του κατωφλίου ανιχνεύονται ως ελαττωματικά.

Η δοκιμή με μέτρηση των ρευμάτων, I_{DDQ} testing, αντιμετωπίζει σημαντικά προβλήματα καθώς οι διαστάσεις της τεχνολογίας κατασκευής των κυκλωμάτων συνεχώς μικραίνουν. Ακόμα και στην τεχνολογία κατασκευής των 0.18 μm τα ρεύματα διαρροής είναι αρκετά μεγάλα ώστε ο διαχωρισμός των εσφαλμένων από τα ορθά κυκλώματα να γίνεται ολοένα δυσκολότερος. Σε τεχνολογίες κατασκευής με ακόμα μικρότερες διαστάσεις, τα ρεύματα διαρροής μεγαλώνουν, δυσχεραίνοντας ακόμα περισσότερο τη δοκιμή με μέτρηση ρευμάτων.

Στον αντίποδα των δομικών μοντέλων ελαττωμάτων βρίσκονται τα μοντέλα ελαττωμάτων λειτουργίας και συμπεριφοράς. Το *μοντέλο ελαττωμάτων λειτουργίας (functional fault model)* προσεγγίζει τον τρόπο που οι φυσικές ατέλειες αλλοιώνουν την ορθή λειτουργία του κυκλώματος, δηλαδή τον πίνακα αληθείας του. Τα μοντέλα λειτουργικών ελαττωμάτων αναπτύσσονται κατά περίπτωση (ad-hoc) ανά κύκλωμα βάσει της λειτουργίας του κυκλώματος και όχι βάσει της δομής του. Το *οικουμενικό μοντέλο ελαττωμάτων (universal fault model)* ορίζει πως οι ατέλειες μπορούν να μετατρέψουν τον πίνακα αληθείας του κυκλώματος με οποιοδήποτε τρόπο αρκεί το κύκλωμα να παραμείνει συνδυαστικό. Η ανίχνευση των λειτουργικών ελαττωμάτων απαιτεί συχνά εξαντλητική ή ψευδο-εξαντλητική δοκιμή.

Ένα άλλο διαδεδομένο μοντέλο ελαττωμάτων είναι το *μοντέλο ελαττώματος κυψελίδας (cell fault model-CFM)* [54], το οποίο βρίσκει εφαρμογή στις επαναληπτικές διατάξεις λογικής. Οι *επαναληπτικές διατάξεις λογικής (Iterative Logic Arrays-ILAs)* αποτελούνται από πολλαπλά αντίγραφα ενός κυκλώματος το οποίο καλείται *κυψελίδα (cell)*. Ο αθροιστής ρι-

*Ονομάζονται εσωτερικά διότι τουλάχιστον η μία γραμμή που συμμετέχει στο βραχυκύκλωμα βρίσκεται εντός του τρανζίστορ.

πής κρατουμένου (ripple carry adder) αποτελεί μια επαναληπτική διάταξη λογικής, όπου η κυψελίδα είναι ο πλήρης αθροιστής (full adder) ο οποίος επαναλαμβάνεται σε σειρά⁷. Το μοντέλο ελαττώματος κυψελίδας (*Cell fault model-CFM*) ορίζει πως οποιοδήποτε συνδυασμός ατελειών μπορεί να επηρεάσει μια μόνο κυψελίδα αρκεί η συμπεριφορά της να παραμείνει συνδυαστική. Η ανίχνευση των ελαττωμάτων κυψελίδας απαιτεί την εξαντλητική εφαρμογή διανυσμάτων δοκιμής στις εισόδους της κυψελίδας.

Το μοντέλο ελαττωμάτων συμπεριφοράς (*behavioral fault model*) ορίζεται στο υψηλότερο επίπεδο αφαίρεσης (abstraction level), όπου η συμπεριφορά του κυκλώματος περιγράφεται σε μία γλώσσα περιγραφής υλικού όπως στη VHDL ή στη Verilog. Τα ελαττώματα συμπεριφοράς περιγράφουν πώς επηρεάζονται οι δομές (constructs) των γλωσσών περιγραφής υλικού. Για παράδειγμα, ένα ελάττωμα συμπεριφοράς σε μια δομή επιλογής όπως if (C) then S1 else S2 ορίζει πως η δήλωση S1 θα εκτελείται πάντα ανεξάρτητα από τη συνθήκη C.

Τα μοντέλα ελαττωμάτων σε υψηλό επίπεδο αξιολογούνται μετρώντας το ποσοστό κάλυψης ελαττωμάτων σε χαμηλότερο επίπεδο κυρίως με βάση το μοντέλο ελαττωμάτων μόνιμης τιμής.

2.5 Ανάλυση Δοκιμαστικότητας Κυκλωμάτων

Η δοκιμαστικότητα (*testability*) είναι ένα σχετικό μέγεθος το οποίο εκφράζει τη δυσκολία της δοκιμής ενός κυκλώματος. Γενικότερα, αντιπροσωπεύει τη δυσκολία στην εφαρμογή των επιθυμητών τιμών στους κόμβους (nodes) του κυκλώματος αλλά και της παρατήρησης των εσφαλμένων τιμών που μπορεί να έχουν εξαιτίας των ελαττωμάτων. Η δοκιμαστικότητα εκφράζεται με δύο επιμέρους έννοιες, της ελεγχιμότητας και της παρατηρησιμότητας. Η ελεγχιμότητα (*controllability*) εκφράζει την δυσκολία ανάθεσης τιμής σε ένα κόμβο του κυκλώματος από τις κύριες εισόδους. Αντίστοιχα, η παρατηρησιμότητα (*observability*) εκφράζει τη δυσκολία στη διάδοση μιας εσφαλμένης τιμής ενός κόμβου στις κύριες εξόδους. Η ανάλυση δοκιμαστικότητας (*testability analysis*) έχει στόχο της τη μέτρηση της ελεγχιμότητας και της παρατηρησιμότητας σε όλους τους κόμβους του κυκλώματος. Τα αποτελέσματα της μελέτης δοκιμαστικότητας χρησιμοποιούνται από τους αλγορίθμους εξαγωγής δεδομένων οι οποίοι είναι σε θέση να γνωρίζουν ποιοι κόμβοι είναι πιο εύκολο να λάβουν τις απαραίτητες τιμές. Επιπλέον, χρησιμοποιούνται από τις τεχνικές σχεδίασης για αυξημένη δοκιμαστικότητα (εξετάζονται στη συνέχεια) ώστε να ανιχνευθούν δύσκολα δοκιμάσιμες περιοχές στο κύκλωμα και να ληφθεί ειδική σχεδιαστική μέριμνα για αυτά.

Η πιο διαδεδομένη μέθοδος μελέτης της δοκιμαστικότητας είναι η *τοπολογική (topological)*⁸ τεχνική *Sandia Controllability/Observability Analysis Program* ή αλλιώς τεχνική *SCOAP* [55].

⁷Οι αθροιστές, αφαιρέτες, αυξητές (incrementers), μειωτές (decrementers), πολλαπλασιαστές αποτελούν παραδείγματα επαναληπτικών διατάξεων λογικής.

⁸Ονομάζεται τοπολογική διότι λαμβάνει υπόψη της τη δομή του κυκλώματος.

Η τεχνική SCOAP υπολογίζει έξι μεγέθη για κάθε κόμβο s του κυκλώματος. Η *συνδυαστική ελεγχσιμότητα-0* (0 -controllability/ $CC0$) και η *συνδυαστική ελεγχσιμότητα-1* (1 -controllability/ $CC1$) αναφέρονται στην δυσκολία να λάβει ένας κόμβος την τιμή 0 ή 1 αντίστοιχα. Η τιμή *συνδυαστική παρατηρησιμότητα* (C -observability/ CCO) εκφράζει τη δυσκολία να παρατηρηθεί ή να διαδοθεί η τιμή ενός κόμβου στις κύριες εξόδους. Αντίστοιχα ορίζονται και τα υπόλοιπα μεγέθη για τα ακολουθιακά κυκλώματα. Στα συνδυαστικά κυκλώματα, η δυσκολία που περιγράφουν οι τιμές ελεγχσιμότητας και παρατηρησιμότητας αναφέρονται με βάση τον αριθμό των υπολοίπων κόμβων που χρειάζεται να λάβουν συγκεκριμένες τιμές ώστε είτε ο κόμβος s να λάβει την επιθυμητή τιμή είτε η εσφαλμένη τιμή του να διαδοθεί στις εξόδους. Στα ακολουθιακά κυκλώματα η δυσκολία αναφέρεται στον αριθμό των κύκλων ρολογιού που απαιτούνται ώστε ο κόμβος s να ελεγχθεί ή να παρατηρηθεί. Τα μεγέθη δοκιμαστικότητας λαμβάνουν θετικές τιμές. Όσο μεγαλύτερη είναι η τιμή του μεγέθους δοκιμαστικότητας τόσο μεγαλύτερη δυσκολία εκφράζει. Η ελεγχσιμότητα των κύριων εισόδων λαμβάνει πάντοτε την τιμή 1 ενώ η παρατηρησιμότητα λαμβάνει πάντα την τιμή 0.

Οι Πίνακες 2.1 και 2.2 παρουσιάζουν τους κανόνες βάσει των οποίων υπολογίζεται η ελεγχσιμότητα των εξόδων των πυλών και η παρατηρησιμότητα των εισόδων των πυλών.

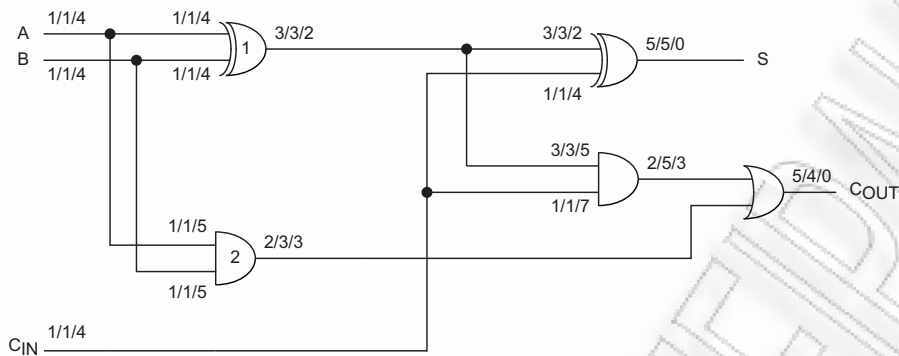
Έξοδος Πύλης	Ελεγχσιμότητα-0	Ελεγχσιμότητα-1
AND	$\min\{CC0(a), CC0(b)\} + 1$	$\sum\{CC1(a), CC1(b)\} + 1$
OR	$\sum\{CC0(a), CC0(b)\} + 1$	$\min\{CC1(a), CC1(b)\} + 1$
XOR	$\min\{CC1(a) + CC1(b), CC0(a) + CC0(b)\} + 1$	$\min\{CC0(a) + CC1(b), CC1(a) + CC0(b)\} + 1$
Διακλάδωση	$CC0(\text{κόμβος διακλ.})$	$CC1(\text{κόμβος διακλ.})$

Πίνακας 2.1: Κανόνες υπολογισμού ελεγχσιμότητας.

Είσοδος Πύλης	Παρατηρησιμότητα
AND	$\sum\{CCO, CC1(\text{υπόλοιπων εισόδων})\} + 1$
OR	$\sum\{CCO, CC0(\text{υπόλοιπων εισόδων})\} + 1$
XOR	$a : \sum(CCO, \min\{CC0(b), CC1(b)\}) + 1$ $b : \sum(CCO, \min\{CC0(a), CC1(a)\}) + 1$
Κόμβος Διακλάδωσης	$\min\{CCO(\text{διακλαδώσεων})\}$

Πίνακας 2.2: Κανόνες υπολογισμού παρατηρησιμότητας.

Η Εικόνα 2.7 παρουσιάζει τα αποτελέσματα της ανάλυσης δοκιμαστικότητας ενός πλήρους αθροιστή (full adder) σύμφωνα με την τεχνική SCOAP. Ο συμβολισμός $v_1/v_2/v_3$ αναφέρεται στις τιμές ελεγχσιμότητα-0/ελεγχσιμότητα-1/παρατηρησιμότητα. Σύμφωνα με τον Πίνακα 2.1, η ελεγχσιμότητα-0/1 των εισόδων της πύλης AND(2) είναι ίδια με την ελεγχσιμότητα των κυρίων



Εικόνα 2.7: Ανάλυση δοκιμαστικότητας ενός πλήρους αθροιστή σύμφωνα με τη μέθοδο SCOAP.

εισόδων A και B , αφού οι εισόδους της πύλης προέρχονται από διακλαδώσεις. Η ελεγχιμότητα-0 της εξόδου της πύλης AND είναι η ελάχιστη ελεγχιμότητα-0 των εισόδων αυξημένη κατά 1, άρα η ελεγχιμότητα-0 της εξόδου είναι 2. Αντίστοιχα, η ελεγχιμότητα-1 της εξόδου είναι το άθροισμα της ελεγχιμότητας-1 των εισόδων αυξημένο κατά 1, επομένως η ελεγχιμότητα-1 είναι 3.

Η ελεγχιμότητα-0 της εξόδου της πύλης XOR(1) είναι 3 διότι σύμφωνα με τον Πίνακα 2.1 είναι $\min\{CC1(a) + CC1(b), CC0(a) + CC0(b)\} + 1 = \min\{1 + 1, 1 + 1\} + 1 = 3$. Με αντίστοιχο τρόπο υπολογίζεται και η ελεγχιμότητα-1.

Συνεχίζοντας με τον ίδιο τρόπο υπολογίζονται όλες οι τιμές της ελεγχιμότητας-0/1 των γραμμών του κυκλώματος.

Η παρατηρησιμότητα της εισόδου a της πύλης OR(5), σύμφωνα με τον Πίνακα 2.2 είναι το άθροισμα της παρατηρησιμότητας της εξόδου και της ελεγχιμότητας-0 των υπολοίπων εισόδων αυξημένο κατά 1, επομένως $0+2+1=3$. Αντίστοιχα, η παρατηρησιμότητα της εισόδου b της πύλης AND(4) είναι το άθροισμα της παρατηρησιμότητας της εξόδου και της ελεγχιμότητας-1 της εισόδου a , αυξημένο κατά 1 επομένως $4+3+1=7$.

Με αντίστοιχο τρόπο υπολογίζονται οι τιμές της παρατηρησιμότητας για τις υπόλοιπες γραμμές του κυκλώματος.

Ένας άλλος τρόπος μελέτης της δοκιμαστικότητας είναι με χρήση προσομοίωσης, όπου στο κύκλωμα εφαρμόζονται διανύσματα δοκιμής είτε τυχαία είτε ίδη υπάρχοντα. Με την προσομοίωση μετρώνται οι εμφανίσεις των τιμών 0, 1 και των μεταβάσεων $0 \rightarrow 1$, $1 \rightarrow 0$. Οι κόμβοι που εμφανίζουν χαμηλό αριθμό μεταβάσεων ή παρουσιάζουν χαμηλή συχνότητα εμφάνισης των τιμών 0 ή 1 υποδεικνύουν προβλήματα χαμηλής δοκιμαστικότητας.

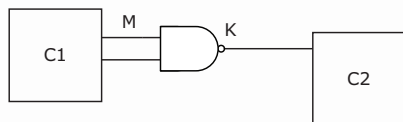
2.6 Σχεδίαση για Αυξημένη Δοκιμαστικότητα

Η εξαγωγή των διανυσμάτων δοκιμής δυσκολεύει όσο μεγαλώνει το κύκλωμα υπό δοκιμή. Αν το κύκλωμα περιέχει και ακολουθιακή λογική τότε η δυσκολία για εξαγωγή διανυσμάτων δοκιμής μεγαλώνει εκθετικά, εξαιτίας του ακολουθιακού βάθους, σε σημείο που η εξαγωγή διανυσμάτων δεν είναι υπολογιστικά εφικτή. Οι *τεχνικές σχεδίασης με αυξημένη δοκιμαστικότητα (Design for Testability—DFT)* προτείνουν αλλαγές στη σχεδίαση του κυκλώματος ώστε να είναι εφικτή η αποτελεσματική εξαγωγή διανυσμάτων, εφαρμόζονται δε συνήθως σε ακολουθιακά κυκλώματα. Σχεδίαση για αυξημένη δοκιμαστικότητα πραγματοποιείται και για συνδυαστικά κυκλώματα αλλά σε μικρότερο βαθμό και σε σημεία που η ελεγχσιμότητα ή/και η παρατηρησιμότητα είναι χαμηλή.

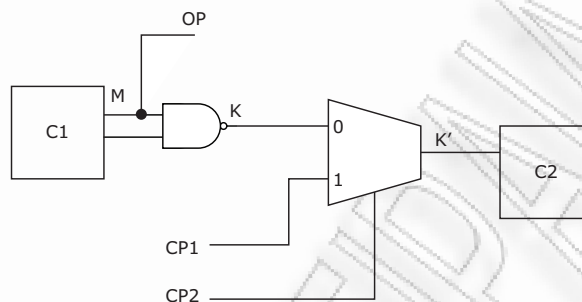
Αρχικά, το πρόβλημα της μειωμένης δοκιμαστικότητας αντιμετωπίστηκε με την *κατά-περίπτωση σχεδίαση αυξημένης δοκιμαστικότητας (ad-hoc DFT)*. Η τεχνική της κατά-περίπτωσης αυξημένης δοκιμαστικότητας βασίζεται στην εύρεση των σημείων εκείνων που παρουσιάζουν πρόβλημα στη δοκιμαστικότητά τους χρησιμοποιώντας τεχνικές ανάλυσης της δοκιμαστικότητας, όπως η τεχνική SCOAP. Στη συνέχεια γίνονται αλλαγές στη σχεδίαση ώστε να αυξηθεί η ελεγχσιμότητα και η παρατηρησιμότητα. Αναλυτικότερα, η κατά-περίπτωση σχεδίαση για αυξημένη δοκιμαστικότητα βασίζεται σε κάποιες "καλές" πρακτικές σχεδίασης, όπως αυτές έχουν προκύψει από εμπειρικές παρατηρήσεις. Ενδεικτικά περιλαμβάνουν την αποφυγή της ασύγχρονης λογικής, την αποφυγή συνδυαστικών βρόχων ανάδρασης (combinational feedback loops), την αποφυγή πλεονάζουσας λογικής (redundant logic), τη διαμέριση μεγάλων τμημάτων λογικής σε μικρότερα ώστε να απλουστευτεί η εξαγωγή διανυσμάτων και την *εισαγωγή σημείων δοκιμής (test point insertion)*. Τα σημεία δοκιμής εισάγονται στα σημεία εκείνα που έχει διαπιστωθεί πως έχουν χαμηλή δοκιμαστικότητα (ελεγχσιμότητα ή παρατηρησιμότητα). Εκτενής συζήτηση γύρω από το θέμα των σημείων δοκιμής υπάρχει στο [31].

Στο κύκλωμα της Εικόνας 2.8.1 που αποτελείται από τα συνδυαστικά κυκλώματα $C1$, $C2$ και μια πύλη NAND, διαπιστώθηκε έπειτα από μέτρηση της δοκιμαστικότητας πως ο κόμβος K έχει μειωμένη ελεγχσιμότητα τόσο ως προς την τιμή 0 όσο και ως προς την τιμή 1 ενώ ο κόμβος M έχει χαμηλή παρατηρησιμότητα. Το κύκλωμα τροποποιείται όπως φαίνεται στην Εικόνα 2.8.2 εισάγοντας δύο σημεία ελεγχσιμότητας, $CP1$ και $CP2$ και ένα σημείο παρατηρησιμότητας OP καθώς και έναν πολυπλέκτη. Κατά την κανονική λειτουργία ο ακροδέκτης $CP2$ λαμβάνει την τιμή 0 ενώ η τιμή που λαμβάνει ο ακροδέκτης $CP1$ είναι αδιάφορη. Κατά τη δοκιμή, ο ακροδέκτης $CP2$ λαμβάνει την τιμή 1 και η είσοδος του $C2$ λαμβάνει την επιθυμητή τιμή (ανάλογα με το διάνυσμα δοκιμής) από τον ακροδέκτη $CP1$. Επιπλέον κατά τη δοκιμή παρατηρείται και ο ακροδέκτης OP ώστε να ανιχνευθεί τυχόν ελάττωμα στο $C1$.

Η κατά-περίπτωση σχεδίαση για αυξημένη δοκιμαστικότητα δεν αποτελεί εύκολα υιοθετήσιμη λύση καθώς όπως φαίνεται και από την Εικόνα 2.8 αυξάνει το πλήθος των ακροδεκτών του αρχικού κυκλώματος. Μια λύση στο πρόβλημα αυτό, είναι η χρήση πολύπλεξης για τα



2.8.1: Αρχικό κύκλωμα



2.8.2: Τροποποιημένο κύκλωμα με αυξημένη δοκιμαστικότητα

Εικόνα 2.8: Παράδειγμα εφαρμογής της κατά-περίπτωση σχεδίασης για αυξημένη δοκιμαστικότητα

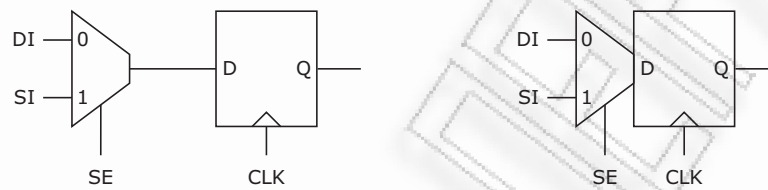
σημεία παρατήρησης αποκρίσεων όπου με τη βοήθεια πολυπλέκτη μόνο ένα σημείο παρατήρησης οδηγείται κάθε φορά σε ένα ακροδέκτη. Αντίστοιχη τεχνική εφαρμόζεται και στα σημεία ελεγχιμότητας [31]. Δυστυχώς, με αυτό τον τρόπο αυξάνεται ο χρόνος εκτέλεσης της δοκιμής. Γενικότερα, το πλήθος των ακροδεκτών για σημεία δοκιμής και ο χρόνος εκτέλεσης της δοκιμής αποτελούν ένα λεπτό σημείο διαπραγμάτευσης. Η κατά-περίπτωση σχεδίαση για αυξημένη δοκιμαστικότητα, αν και επιτυγχάνει το στόχο της, δεν είναι εύκολο να εφαρμοστεί με γενικό τρόπο σε οποιοδήποτε κύκλωμα αφού βασίζεται σε εμπειρικές αποφάσεις. Επιπλέον, δεν είναι δυνατό να προβλέψει κανείς αν και πότε θα εφαρμοστεί αποτελεσματικά σε ένα κύκλωμα υπό δοκιμή.

Σήμερα, η αποτελεσματική αύξηση της δοκιμαστικότητας, πραγματοποιείται με τη *συστηματική ή δομημένη τεχνική σχεδίασης για αυξημένη δοκιμαστικότητα (systematic, structured DFT)* [56]. Η συστηματική σχεδίαση μπορεί να εφαρμοστεί σε κάθε ακολουθιακό κύκλωμα με αυτοματοποιημένο τρόπο και αποτελεί την πιο αποτελεσματική σχεδίαση για αυξημένη δοκιμαστικότητα αφού μετατρέπει το πρόβλημα της εξαγωγής διανυσμάτων δοκιμής από ακολουθιακό σε συνδυαστικό, άρα με μικρότερη πολυπλοκότητα.

Η συστηματική τεχνική σχεδίασης για αυξημένη δοκιμαστικότητα μετατρέπει τα ακολουθιακά στοιχεία (flip-flop) σε *κυψελίδες σάρωσης (scan cells)* για να σχηματίσει *αλυσίδες σάρωσης (scan chains)* ή αλλιώς *καταχωρητές σάρωσης (scan registers)*. Η κυψελίδα σάρωσης είναι ένα στοιχείο αποθήκευσης το οποίο έχει δύο εισόδους δεδομένων (data inputs), την είσοδο δεδομένων και την είσοδο σάρωσης καθώς και μια είσοδο ελέγχου (control input). Η είσοδος ελέγχου επιλέγει ποια από τις δύο γραμμές εισόδου θα αποθηκευτεί στο στοιχείο αποθήκευσης (flip-flop). Οι κυψελίδες σάρωσης αντικαθιστούν τα στοιχεία αποθήκευσης σε ένα ακολουθιακό κύκλωμα. Η έξοδος του συνδυαστικού κυκλώματος που παράγει την επόμενη κατάσταση συνδέεται στην είσοδο δεδομένων της κυψελίδας σάρωσης. Οι αλυσίδες σάρωσης σχηματίζονται συνενώνοντας την έξοδο της προηγούμενης κυψελίδας σάρωσης στην είσοδο σάρωσης της τρέχουσας κυψελίδας σάρωσης. Όταν όλα τα ακολουθιακά στοιχεία μετατρέ-

πονται σε κυψελίδες σάρωσης τότε γίνεται λόγος για *σχεδιασμό πλήρους σάρωσης (full scan design)* ενώ όταν κάποια από τα ακολουθιακά στοιχεία μετατρέπονται σε κυψελίδες σάρωσης γίνεται λόγος για *σχεδιασμό μερικής σάρωσης (partial scan design)*.

Ο πιο διαδεδομένος τύπος κυψελίδας σάρωσης αποτελείται από το *συνδυασμό ενός πολυπλέκτη και ενός ακμοπυροδοτούμενου flip-flop τύπου D (Mixed D Scan Cell)*. Η Εικόνα 2.9 παρουσιάζει την αρχιτεκτονική της κυψελίδας σάρωσης και το συμβολισμό της.



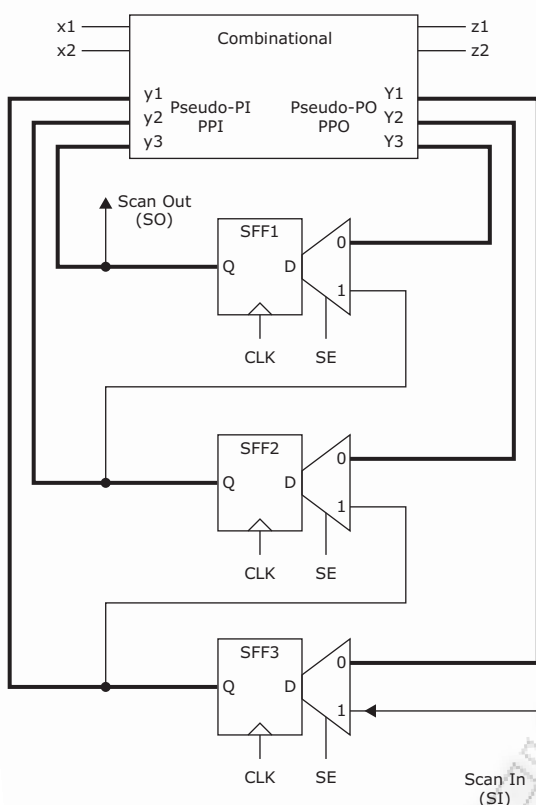
Εικόνα 2.9: Η κυψελίδα σάρωσης Mixed D Scan Cell.

Ο πολυπλέκτης χρησιμοποιεί την είσοδο Scan Enable-SE για να επιλέξει μεταξύ της εισόδου δεδομένων, Data Input-DI, και της εισόδου στην οποία συνδέεται η έξοδος της προηγούμενης κυψελίδας σάρωσης, Scan Input-SI. Κατά την κανονική λειτουργία, η είσοδος SE λαμβάνει την τιμή 0 επιλέγοντας την κανονική είσοδο δεδομένων, ενώ όταν πραγματοποιείται η δοκιμή λαμβάνει την τιμή 1. Το μεγαλύτερο πλεονέκτημα της κυψελίδας σάρωσης με πολυπλέκτη είναι πως είναι πολύ εύκολο να εφαρμοστεί από αυτόματα εργαλεία εισαγωγής αλυσίδων σάρωσης καθώς βασίζεται στο D flip-flop. Όμως ο πολυπλέκτης αποτελεί αρνητικό σημείο καθώς μειώνει την ταχύτητα διάδοσης της γραμμής εισόδου δεδομένων DI.

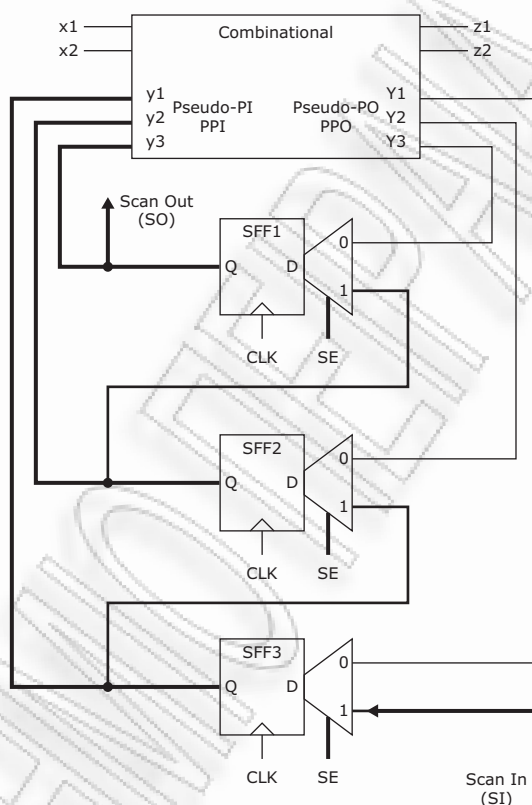
Ένας άλλος τύπος κυψελίδας σάρωσης είναι η *κυψελίδα σάρωσης με (επιπλέον) ρολόι (Clocked Scan Cell)* όπου δε χρησιμοποιείται πολυπλέκτης αλλά η επιλογή εισόδου γίνεται βάσει των ρολογιών Data-CLK και Shift-CLK. Κατά την κανονική λειτουργία ή κατά τη λειτουργία αποθήκευσης (capture) χρησιμοποιείται το ρολόι DCLK ενώ κατά την λειτουργία ολίσθησης το ρολόι SCLK.

Οι παραπάνω κυψελίδες σάρωσης χρησιμοποιούνται σε κυκλώματα που είναι υλοποιημένα με ακμοπυροδοτούμενα flip-flop. Για τα κυκλώματα που υλοποιούνται με κυκλώματα μαντάλωσης (latch based designs) χρησιμοποιείται η κυψελίδα σάρωσης τύπου Level Sensitive Scan Design-LSSD. Η κυψελίδα σάρωσης LSSD αποτελείται από δύο μανταλωτές σε συνδεσμολογία αφέντη-σκλάβου (master-slave) και από τρία ρολόγια τα οποία χρησιμοποιούνται για την επιλογή της εισόδου των δεδομένων ή της εισόδου της αλυσίδας σάρωσης.

Η Εικόνα 2.10 παρουσιάζει ένα ακολουθιακό κύκλωμα το οποίο περιέχει μια αλυσίδα σάρωσης με τρεις κυψελίδες σάρωσης. Οι εισοδοί x1 και x2 είναι οι *κύριες εισοδοί (primary inputs-PIs)* ενώ οι z1 και z2 είναι οι *κύριες έξοδοι (primary outputs-POs)*. Οι γραμμές y1, y2, y3 αποτελούν την παρούσα κατάσταση ενώ οι Y1, Y2, Y3 αντιπροσωπεύουν την επόμενη κατάσταση. Στην κανονική λειτουργία η είσοδος SE λαμβάνει την τιμή 0 ενεργοποιώντας τα



2.10.1: Η αλυσίδα σάρωσης στην κανονική λειτουργία του κυκλώματος



2.10.2: Η αλυσίδα σάρωσης κατά τη δοκιμή

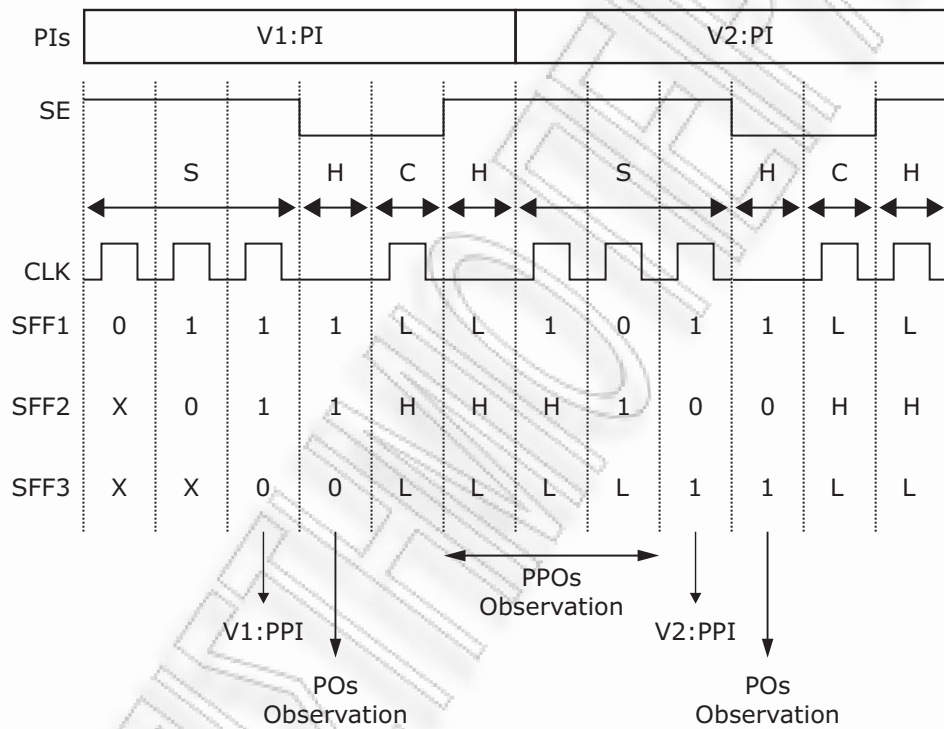
Εικόνα 2.10: Ακολουθιακό κύκλωμα με αλυσίδα σάρωσης που αποτελείται από τρεις κυψελίδες σάρωσης.

κατάλληλα μονοπάτια (με έντονο χρώμα στην Εικόνα 2.10.1) ώστε η επόμενη κατάσταση στις γραμμές Y1, Y2, Y3 να αποθηκευτεί στις κυψελίδες σάρωσης. Εκτός από την καθυστέρηση του πολυπλέκτη στην κυψελίδα σάρωσης, το κύκλωμα λειτουργεί όπως θα λειτουργούσε και πριν αν δεν υπήρχαν οι κυψελίδες σάρωσης αλλά D flip-flop. Όταν πραγματοποιείται η δοκιμή, η γραμμή SE λαμβάνει την τιμή 1, ενεργοποιώντας τα κατάλληλα μονοπάτια (με έντονο χρώμα στην Εικόνα 2.10.2) ώστε οι κυψελίδες σάρωσης να σχηματίσουν την αλυσίδα σάρωσης.

Για την εξαγωγή διανυσμάτων δοκιμής χρησιμοποιούνται αλγόριθμοι για συνδυαστικά κυκλώματα και όχι για ακολουθιακά. Οι y_1 , y_2 , y_3 θεωρούνται *ψευδο-κύριες εισόδοι (pseudo-primary inputs-PPIs)* στις οποίες οι συνδυαστικοί αλγόριθμοι εξαγωγής διανυσμάτων μπορούν να εφαρμόσουν οποιαδήποτε τιμή. Αντίστοιχα, οι Y1, Y2, Y3 θεωρούνται *ψευδο-κύριες εξόδοι (pseudo-primary outputs-PPOs)* οι οποίοι είναι πλήρως παρατηρήσιμοι κατά τη διαδικασία της εξαγωγής διανυσμάτων. Με αυτό τον τρόπο το πρόβλημα της εξαγωγής διανυσμάτων δοκιμής μετατρέπεται από ακολουθιακό σε συνδυαστικό. Τα διανύσματα δοκιμής που προκύπτουν αποτελούνται από δύο μέρη, τις τιμές για τις κύριες εισόδους και τις τιμές για τις ψευδο-

κύριες εισόδους. Έτσι, το διάνυσμα δοκιμής V1 χωρίζεται στις τιμές των κύριων εισόδων V1:PI, δηλαδή x1, x2, και στις τιμές των ψευδο-κύριων εισόδων, V1:PPI, δηλαδή y1, y2, y3.

Το ακόλουθο παράδειγμα στην Εικόνα 2.11 [57] παρουσιάζει το τρόπο με τον οποίο εφαρμόζεται η δοκιμή εφαρμόζοντας δύο διανύσματα δοκιμής, V1 και V2 στο ακολουθιακό κύκλωμα της Εικόνας 2.10.



Εικόνα 2.11: Η λειτουργία της αλυσίδας σάρωσης κατά την εφαρμογή των διανυσμάτων δοκιμής V1, V2.

Η αλυσίδα σάρωσης τίθεται σε κατάσταση ολίσθησης (shift mode) θέτοντας στην είσοδο scan enable-SE την τιμή 1. Οι τιμές των ψευδο-κύριων εισόδων του διανύσματος V1, V1:PPI εφαρμόζονται στην είσοδο scan input-SI με αντίστροφη σειρά, δηλαδή πρώτα εφαρμόζεται η τιμή για την κυψελίδα σάρωσης SFF3. Έπειτα από εφαρμογή τριών κύκλων ρολογιού, οι τιμές του V1:PPI βρίσκονται στις κυψελίδες σάρωσης. Επίσης ήδη από την έναρξη της κατάστασης ολίσθησης, το τμήμα V1:PI έχει εφαρμοστεί στις κύριες εισόδους. Στη συνέχεια ακολουθεί ένας κύκλος διατήρησης (hold cycle) κατά τον οποίο δεν εφαρμόζεται παλμός ρολογιού και η είσοδος SE γίνεται 0. Στο σημείο αυτό όλο το διάνυσμα V1 έχει εφαρμοστεί στο κύκλωμα. Ο κύκλος διατήρησης είναι απαραίτητος ώστε το κύκλωμα να προλάβει να σταθεροποιηθεί. Έπειτα από τη σταθεροποίηση του κυκλώματος γίνεται και η παρατήρηση των κύριων εξόδων του κυκλώματος. Η αλυσίδα σάρωσης τίθεται σε κατάσταση σύλληψης ή αποτύπωσης (capture

mode) όπου εφαρμόζεται ένας παλμός ρολογιού και οι τιμές των ψευδο-εξόδων αποθηκεύονται στις κυψελίδες σάρωσης. Άλλος ένας κύκλος διατήρησης ακολουθεί, όπου δεν εφαρμόζεται παλμός ρολογιού και η είσοδος SE λαμβάνει την τιμή 1. Παράλληλα γίνεται και η παρατήρηση της εξόδου της αλυσίδας σάρωσης scan out-SO.

Η αλυσίδα τίθεται ξανά σε κατάσταση ολίσθησης και το τμήμα των ψευδο-εισόδων του διανύσματος V2, V2:PPI εφαρμόζεται στην είσοδο SI. Καθώς οι νέες τιμές ολισθαίνουν στις κυψελίδες σάρωσης οι αποκρίσεις του διανύσματος V1 ολισθαίνουν στην έξοδο της αλυσίδας όπου και παρατηρούνται. Η παρατήρηση ολοκληρώνεται στους δύο πρώτους κύκλους της κατάστασης ολίσθησης. Η ίδια διαδικασία ακολουθείται ωστόσο εφαρμοστούν όλα τα διανύσματα δοκιμής.

Ο χρόνος περάτωσης της δοκιμής εξαρτάται από το πλήθος των διανυσμάτων δοκιμής και από το μέγεθος της αλυσίδας, δηλαδή από τον αριθμό των κυψελίδων σάρωσης που περιέχει. Έτσι αν μια αλυσίδα σάρωσης περιέχει k κυψελίδες σάρωσης ο χρόνος εφαρμογής N διανυσμάτων δοκιμής είναι $N \times (k + 3) + (k - 1)$. Για κάθε διάνυσμα δοκιμής χρειάζονται $k + 3$ κύκλοι για την εφαρμογή του, όπου k το πλήθος των κυψελίδων σάρωσης. Οι τρεις επιπλέον κύκλοι είναι δύο κύκλοι διατήρησης και ένας κύκλος για την αποθήκευση των αποκρίσεων στις κυψελίδες σάρωσης. Για το τελευταίο διάνυσμα δοκιμής χρειάζονται επιπλέον $k - 1$ κύκλοι ώστε να παρατηρηθούν τα περιεχόμενα των $k - 1$ κυψελίδων σάρωσης στην έξοδο της αλυσίδας σάρωσης. Το περιεχόμενο της κυψελίδας k της οποίας η έξοδος είναι η έξοδος της αλυσίδας σάρωσης παρατηρήθηκε κατά τον τελευταίο παλμό αποτύπωσης.

Σε μεγάλα ακολουθιακά κυκλώματα, τα στοιχεία μνήμης διαμερίζονται σε περισσότερες από μια αλυσίδες σάρωσης (multiple scan chains) ώστε να μειωθεί ο συνολικός χρόνος της δοκιμής. Αυτό σημαίνει πως αυξάνεται ο αριθμός των απαιτούμενων ακροδεκτών για την υλοποίηση των αλυσίδων αλλά μειώνεται ο χρόνος δοκιμής. Ένα παράδειγμα πολλαπλών αλυσίδων σάρωσης παρουσιάζεται στην Εικόνα 2.17 (σελ. 54). Για να μειωθεί ο αριθμός των επιπλέον ακροδεκτών εφαρμόζονται τεχνικές πολύπλεξης μεταξύ των διαθέσιμων ακροδεκτών και των ακροδεκτών των κυρίων εξόδων, αλλά έτσι αυξάνει η πολυπλοκότητα του κυκλώματος. Η κατανομή των στοιχείων μνήμης σε πολλαπλές αλυσίδες σάρωσης είναι ένα πρόβλημα NP-complete, όπως έχει δείχθει στην [58] η οποία μελετά το πρόβλημα της κατανομής σε πολλαπλές αλυσίδες ώστε να μειωθεί ο συνολικός χρόνος εκτέλεσης της δοκιμής. Περισσότερες πληροφορίες σχετικά με τη διαμέριση των στοιχείων μνήμης σε πολλαπλές αλυσίδες σάρωσης υπάρχουν στην αναφορά [30].

Η σχεδίαση για αυξημένη δοκιμαστικότητα βασισμένη σε αλυσίδες σάρωσης επιτυγχάνει το στόχο της για υψηλό ποσοστό κάλυψης ελαττωμάτων. Η επιτυχία αυτή συνοδεύεται και με το αντίστοιχο κόστος. Οι αλυσίδες σάρωσης επιβαρύνουν το συνολικό μέγεθος του κυκλώματος με αποτέλεσμα να κατασκευάζονται λιγότερα κυκλώματα σε μια επιφάνεια πυριτίου γεγονός που επηρεάζει αρνητικά την εσοδεία. Το ποσοστό της αύξησης του μεγέθους του κυκλώματος κυμαίνεται μεταξύ 2% και 10%. Επιπλέον, οι αλυσίδες σάρωσης απαιτούν επι-

πλέον ειδικού σκοπού ακροδέκτες (dedicated pins) στο κύκλωμα. Όσο περισσότερες αλυσίδες σάρωσης υπάρχουν τόσο περισσότεροι ακροδέκτες είναι απαραίτητοι. Οι αλυσίδες σάρωσης επιβαρύνουν και τη συχνότητα λειτουργίας του κυκλώματος. Η αντικατάσταση των στοιχείων μνήμης με κυψελίδες σάρωσης στα κρίσιμα μονοπάτια (critical paths) ενός κυκλώματος έχει ως αποτέλεσμα την αύξηση του κύκλου ρολογιού. Η δοκιμή με αλυσίδες σάρωσης θέτει το κύκλωμα σε ειδική κατάσταση λειτουργίας όπου η κατανάλωση ενέργειας είναι πολύ μεγαλύτερη από αυτή που καταναλώνεται κατά τη φυσιολογική κατάσταση λειτουργίας. Η αυξημένη κατανάλωση μπορεί να επηρεάσει τη δοκιμή.

Παρ' όλα αυτά, οι αλυσίδες σάρωσης είναι η πλέον επιτυχημένη τεχνική για αυξημένη δοκιμαστικότητα αφού μετατρέπει το πρόβλημα της εξαγωγής διανυσμάτων από ακολουθιακό σε συνδυαστικό. Συνάμα, οι αλυσίδες σάρωσης παρέχουν μεγάλη πληροφορία για το τι συμβαίνει στο εσωτερικό ενός ακολουθιακού κυκλώματος και για αυτό το λόγο χρησιμοποιούνται ευρέως κατά την διαδικασία της *αποσφαλμάτωσης των πρωτοτύπων (first silicon debug)*, δηλαδή των πρώτων κατασκευασμένων ολοκληρωμένων κυκλωμάτων.

Το κόστος από την εφαρμογή της σχεδίασης αλυσίδων πλήρους σάρωσης μπορεί να μειωθεί επιλέγοντας ένα υποσύνολο των στοιχείων μνήμης για το σχηματισμό αλυσίδων σάρωσης. Στην περίπτωση αυτή γίνεται λόγος για *σχεδιασμό αλυσίδων μερικής σάρωσης (partial scan design)*. Το πρόβλημα της εξαγωγής των διανυσμάτων δοκιμής παραμένει ακολουθιακό με μικρότερη όμως πολυπλοκότητα αφού μειώνεται το ακολουθιακό βάθος, ενώ σε μερικές περιπτώσεις μπορεί να μετατραπεί και σε συνδυαστικό, όπως σε κυκλώματα που υιοθετούν την τεχνική της διοχέτευσης (pipeline) για αύξηση της διεκπεραιωτικής ικανότητας (throughput). Γενικότερα, ακόμα αν και το πρόβλημα παραμένει ακολουθιακό είναι πιο προσιτό για τους ακολουθιακούς αλγορίθμους εξαγωγής διανυσμάτων. Το κύριο ερώτημα για την επιτυχημένη εφαρμογή της σχεδίασης μερικής σάρωσης είναι ποια στοιχεία μνήμης θα επιλεγούν για να σχηματίσουν τις αλυσίδες σάρωσης. Ο *γράφος δομής (structure graph-s+graph)* χρησιμοποιείται για να περιγράψει ένα ακολουθιακό κύκλωμα. Οι κόμβοι του γράφου είναι τα στοιχεία μνήμης ενώ οι ακμές του γράφου αντιπροσωπεύουν τα συνδυαστικά μονοπάτια (combinational paths) που συνδέουν τα στοιχεία μνήμης. Το πρόβλημα της επιλογής των στοιχείων μνήμης ανάγεται στο πρόβλημα της εύρεσης του ελάχιστου συνόλου κόμβων με ανάδραση (minimum feedback vertex set), οι οποίοι όταν αφαιρεθούν θα μετατρέψουν το γράφο από κυκλικό (cyclic) σε ακυκλικό (acyclic). Οι εργασίες [59], [60], [61] πραγματεύονται την εύρεση των στοιχείων μνήμης που όταν αφαιρεθούν μετατρέπουν το ακολουθιακό κύκλωμα από κυκλικό σε ακυκλικό. Επίσης στην εργασία [62] έχει αποδειχθεί πως δεν είναι απαραίτητο να αφαιρεθούν όλοι οι κόμβοι του γράφου που συμμετέχουν σε κύκλους. Κύκλοι οι οποίοι ξεκινούν και καταλήγουν στον ίδιο κόμβο μπορούν να παραμείνουν στο δομικό γράφο χωρίς να δυσκολεύουν την εξαγωγή διανυσμάτων δοκιμής.

Ένας άλλος τρόπος εφαρμογής του σχεδιασμού αλυσίδων μερικής σάρωσης είναι χρησιμοποιώντας τη λειτουργική διαμέριση (functional partitioning). Στη λειτουργική διαμέριση, το

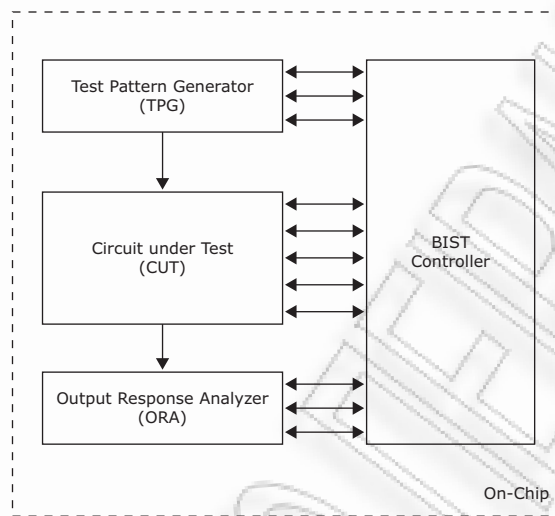
ακολουθιακό κύκλωμα θεωρείται πως αποτελείται από μονοπάτια δεδομένων (datapaths) και από μονάδες ελέγχου (control logic units) οι οποίες καθορίζουν την λειτουργία των μονοπατιών δεδομένων. Αλυσίδες σάρωσης εφαρμόζονται μόνο στα στοιχεία μνήμης των μονάδων ελέγχου και όχι στα μονοπάτια δεδομένων. Με αυτό τον τρόπο επιτυγχάνεται η ευκολότερη εξαγωγή διανυσμάτων δοκιμής για τις μονάδες ελέγχου (οι οποίες είναι πιο πολύπλοκες) δίχως όμως να επηρεάζεται η απόδοση των μονοπατιών δεδομένων.

2.7 Ενσωματωμένη Αυτοδοκιμή

Η ενσωματωμένη αυτοδοκιμή (*Built-in Self-test-BIST*) αναφέρεται σε ένα σύνολο τεχνικών για αυξημένη δοκιμαστικότητα όπου το ίδιο το κύκλωμα αποκτά τη δυνατότητα να πραγματοποιήσει τη δοκιμή του εαυτού του [30], [31], [57]. Στην ενσωματωμένη αυτοδοκιμή η εξαγωγή των διανυσμάτων δοκιμής και η σύγκριση και ανάλυση των αποκρίσεων γίνεται από το ίδιο το κύκλωμα υπό δοκιμή. Υπάρχουν δύο κατηγορίες αυτοδοκιμής. Στην *εξωτερική αυτοδοκιμή (external BIST)*, στο αρχικό κύκλωμα εισάγεται περιφερειακά επιπλέον υλικό το οποίο πραγματοποιεί την εξαγωγή των διανυσμάτων και την συλλογή των αποκρίσεων. Αντίθετα στην *εσωτερική αυτοδοκιμή (internal BIST)* μέρη του αρχικού κυκλώματος τροποποιούνται ώστε να εφαρμόζουν διανύσματα δοκιμής και να συλλέγουν τις αποκρίσεις.

Τα πλεονεκτήματα από την υιοθέτηση της ενσωματωμένης αυτοδοκιμής είναι πολλαπλά. Εφόσον το κύκλωμα δοκιμάζει τον εαυτό του δεν είναι απαραίτητος ο εξοπλισμός δοκιμής. Επιπλέον, η δοκιμή μπορεί να πραγματοποιηθεί στην πραγματική συχνότητα λειτουργίας ή αν μη τι άλλο η συχνότητα πραγματοποίησής της δοκιμής δεν εξαρτάται από τον εξοπλισμό δοκιμής αλλά από το ίδιο το κύκλωμα. Καθώς υποβαθμίζεται ο ρόλος του εξοπλισμού δοκιμής το κόστος της δοκιμής μειώνεται δραματικά αφού δεν είναι αναγκαίο ο εξοπλισμός δοκιμής να εφαρμόζει διανύσματα στην ταχύτητα λειτουργίας του κυκλώματος. Η αυτοδοκιμή δεν περιορίζεται μόνο στο εργοστάσιο κατασκευής αλλά συνοδεύει το ολοκληρωμένο κύκλωμα σε όλο τον κύκλο ζωής του. Με αυτό τον τρόπο η αυτοδοκιμή μπορεί να πραγματοποιείται στο περιβάλλον λειτουργίας του κυκλώματος καθιστώντας περισσότερο αξιόπιστη τη λειτουργία του. Αυτό συμβαίνει διότι μπορεί να ανιχνευθούν ελαττώματα που εμφανίζονται αργότερα στον κύκλο ζωής του κυκλώματος.

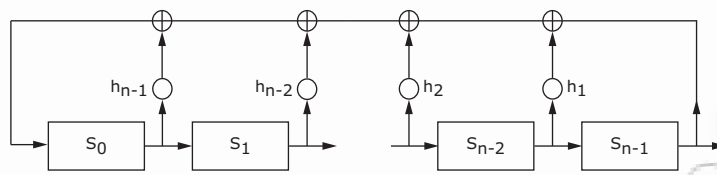
Η αυτοδοκιμή βασίζεται σε τροποποίηση του αρχικού κυκλώματος γεγονός που εισάγει δύο μειονεκτήματα ήδη γνωστά και από τις αλυσίδες σάρωσης. Κατά πρώτον εισάγει επιβάρυνση στο μέγεθος του κυκλώματος, γεγονός που επηρεάζει το πλήθος των κυκλωμάτων που μπορούν να κατασκευαστούν άρα και την τελική εσοδεία. Κατά δεύτερον, οι τροποποιήσεις για αυξημένη δοκιμαστικότητα είναι πολύ πιθανό να επηρεάσουν τα κρίσιμα μονοπάτια του κυκλώματος και έτσι να μειωθεί η συχνότητα λειτουργίας του κυκλώματος. Η αυτοδοκιμή δεν είναι τόσο κατάλληλη για την αποσφαλμάτωση των πρωτότυπων ολοκληρωμένων κυκλωμάτων



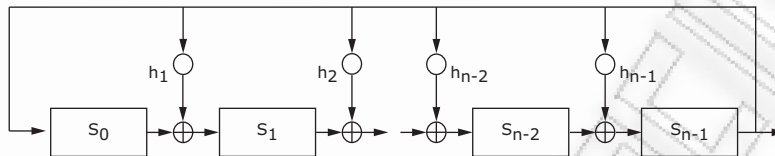
Εικόνα 2.12: Η τυπική έννοια της αυτοδοκιμής.

αμέσως μετά την κατασκευή (silicon debug), διότι η σύγκριση των αποκρίσεων των διανυσμάτων πραγματοποιείται εντός του κυκλώματος στην αυτοδοκιμή και, όπως θα αναλυθεί στη συνέχεια, αυτή η σύγκριση πραγματοποιείται σε συμπυκνωμένες μορφές των αποκρίσεων.

Η αρχιτεκτονική ενός κυκλώματος σχεδιασμένου για να πραγματοποιεί αυτοδοκιμή παρουσιάζεται στην Εικόνα 2.12. Ο γεννήτορας διανυσμάτων δοκιμής (*test pattern generator-TPG*) είναι συνήθως ένας γραμμικός καταχωρητής ολίσθησης με ανάδραση (*linear feedback shift register-LFSR*) [63]. Οι αποκρίσεις των διανυσμάτων δοκιμής συλλέγονται από το κύκλωμα ανάλυσης αποκρίσεων (*output response analyzer-ORA*), το οποίο *συμπύσσει (compact)* όλες τις αποκρίσεις σε μία. Η μοναδική αυτή απόκριση βάσει της οποίας εξετάζεται αν το κύκλωμα περιέχει ελαττώματα ονομάζεται *υπογραφή (signature)*. Η όλη διαδικασία της αυτοδοκιμής συγχρονίζεται από ένα ελεγκτή (*BIST controller*) και λαμβάνει χώρα εντός του ολοκληρωμένου κυκλώματος (*on-chip*). Όπως αναφέραμε ήδη, οι γεννιότερες εξαγωγές διανυσμάτων βασίζονται κατά κύριο λόγο στους *γραμμικούς καταχωρητές ολίσθησης με ανάδραση (linear feedback shift registers-LFSR)*. Οι καταχωρητές αυτοί παράγουν ακολουθίες διανυσμάτων δοκιμής οι οποίες μπορεί να χρησιμοποιηθούν για *εξαντλητική (exhaustive)* δοκιμή ή για *ψευδο-τυχαία (pseudo-random)* δοκιμή. Η εξαντλητική δοκιμή απαιτεί την εφαρμογή 2^n διανυσμάτων, όπου n ο αριθμός των εισόδων του κυκλώματος και εγγυάται ποσοστό κάλυψης ελαττωμάτων μόνη τιμή 100%. Η εξαντλητική δοκιμή είναι εφικτή μόνο όταν ο αριθμός n είναι μικρός. Εξαντλητική δοκιμή μπορεί να εφαρμοστεί και με διανύσματα που παράγονται από έναν μετρητή (*counter*). Μια παραλλαγή της εξαντλητικής δοκιμής είναι η *ψευδο-εξαντλητική (pseudo-exhaustive)* δοκιμή και εφαρμόζεται όταν κάθε κύρια έξοδος εξαρτάται από ένα μικρό αριθμό κερών εισόδων στις οποίες εφαρμόζονται όλα τα διανύσματα δοκιμής. Επειδή συνήθως η



2.13.1: Εξωτερικός καταχωρητής



2.13.2: Εσωτερικός καταχωρητής

Εικόνα 2.13: Γεννήτορες ψευδο-τυχαίων διανυσμάτων δοκιμής βασισμένοι σε καταχωρητές ολίσθησης με ανάδραση.

εξαντλητική δοκιμή είναι απαγορευτική λόγω του υπερβολικά μεγάλου χρόνου που απαιτεί, εφαρμόζεται η ψευδοτυχαία δοκιμή, όπου ένα υποσύνολο των 2^n διανυσμάτων εφαρμόζεται στο κύκλωμα. Η αξιολόγηση των διανυσμάτων πραγματοποιείται με την προσομοίωση ελαττωμάτων. Σε αρκετές περιπτώσεις ο χρόνος πραγματοποίησης της ψευδοτυχαίας δοκιμής μπορεί να γίνει αρκετά μεγάλος ως και απαγορευτικός ενώ και το ποσοστό κάλυψης ελαττωμάτων μπορεί να μη φθάνει σε ικανοποιητικά επίπεδα εξαιτίας της *αντίστασης* που παρουσιάζουν τα κυκλώματα σε ψευδο-τυχαία διανύσματα (random pattern resistant) και παρά το μεγάλο αριθμό διανυσμάτων δοκιμής δεν επιτυγχάνεται μεγάλη κάλυψη ελαττωμάτων.

Κάθε γραμμικός καταχωρητής ολίσθησης με ανάδραση περιγράφεται από ένα χαρακτηριστικό πολυώνυμο της μορφής $1 + h_1x + h_2x^2 + \dots + h_{n-1}x^{n-1} + h_nx^n$, με βαθμό n . Οι γραμμικοί καταχωρητές ολίσθησης με ανάδραση υλοποιούνται με n D flip-flop και πύλες XOR⁹, συνδεδεμένα σε ένα βρόχο ανάδρασης. Υπάρχουν δύο τύποι τέτοιων καταχωρητών, ο εξωτερικός καταχωρητής και ο εσωτερικός, όπως φαίνεται στις Εικόνες 2.13.1 και 2.13.2 αντίστοιχα. Οι διαφορές τους έγκεινται στην ταχύτητα και το μέγεθος υλοποίησης. Οι συντελεστές του χαρακτηριστικού πολυωνύμου, $h_1, h_2, \dots, h_{n-1}, h_n$ καθορίζουν αν υπάρχει σύνδεση στην αντίστοιχη πύλη XOR (οι κύκλοι στην Εικόνα 2.13)

Ο καταχωρητής αρχικοποιείται σε μία αρχική τιμή (seed). Σε κάθε παλμό ρολογιού παράγεται και μια νέα τιμή η οποία μπορεί να χρησιμοποιηθεί ως διάνυσμα δοκιμής. Ανάλογα με το χαρακτηριστικό πολυώνυμο οι τιμές που παράγονται από τον καταχωρητή επαναλαμβάνονται έπειτα από ένα αριθμό κύκλων ρολογιού. Μήκος της επανάληψης καλείται η περίοδος του LFSR, και όταν το χαρακτηριστικό πολυώνυμο είναι πρωταρχικό (primitive) τότε παράγονται όλες οι δυνατές τιμές, $2^n - 1$, όπου n το πλήθος των flip-flop, εκτός από την μηδενική. Στην περίπτωση αυτή λέγεται πως έχει παραχθεί η μέγιστη δυνατή ακολουθία (maximal length

⁹Ο χαρακτηρισμός γραμμικοί αποδίδεται στις πύλες αποκλειστικό-Η (XOR).

sequence) ενώ ο καταχωρητής χαρακτηρίζεται μέγιστου μήκους (maximal length).

Οι καταχωρητές ολίσθησης μέγιστου μήκους είναι ιδανικοί για ψευδοτυχαία δοκιμή καθώς υλοποιούνται πολύ εύκολα και έχουν μικρό μέγεθος. Το πρόβλημα που παρουσιάζεται είναι πως πολλά κυκλώματα παρουσιάζουν "αντίσταση" στα ψευδοτυχαία διανύσματα, δηλαδή όσα ψευδοτυχαία διανύσματα και αν εφαρμοστούν κάποια ελαττώματα δεν ανιχνεύονται. Για παράδειγμα, η ανίχνευση του ελαττώματος μόνιμης τιμής 0 στην έξοδο μιας AND των 5 εισόδων απαιτεί κάθε είσοδος να λάβει την τιμή 1. Η πιθανότητα να παραχθεί το συγκεκριμένο διάνυσμα είναι 2^{-5} .

Οι σταθμισμένοι γραμμικοί καταχωρητές ολίσθησης με ανάδραση (weighted LFSR) επιτρέπουν την εμφάνιση των 0 και 1 στις εξόδους του καταχωρητή με διαφορετική πιθανότητα από αυτή του απλού καταχωρητή [64], [65]. Στους γραμμικούς καταχωρητές της Εικόνας 2.13 η πιθανότητα εμφάνισης της τιμής 0 ή 1 σε κάθε έξοδο είναι 0,5. Οι σταθμισμένοι LFSR παράγουν τις τιμές 0 ή 1 σε κάθε έξοδο με διαφορετική πιθανότητα ώστε τα διανύσματα που θα προκύψουν να ανιχνεύσουν επιπλέον ελαττώματα. Το πρόβλημα με τους σταθμισμένους καταχωρητές είναι πως χρειάζονται αρκετές σταθμίσεις ή βάρη ώστε να επιτευχθεί το επιθυμητό υψηλό ποσοστό κάλυψης.

Εναλλακτικά, η εξαγωγή ψευδοτυχαίων διανυσμάτων δοκιμής μπορεί να πραγματοποιηθεί χρησιμοποιώντας κυψελωτά αυτόματα (cellular automata-CA) [66]. Τα κυψελωτά αυτόματα παράγουν ακολουθίες με μεγαλύτερο βαθμό τυχαιότητας και έτσι έχουν μεγαλύτερη πιθανότητα να ανιχνεύσουν ελαττώματα σε κυκλώματα ανθεκτικά σε ψευδοτυχαία διανύσματα. Αποτελούνται από κυψελίδες οι οποίες περιέχουν ένα D flip-flop και μια πύλη XOR. Κάθε κυψελίδα συνδέεται μόνο με την αμέσως προηγούμενη και την αμέσως επόμενη κυψελίδα, όπως φαίνεται στην Εικόνα 2.14.

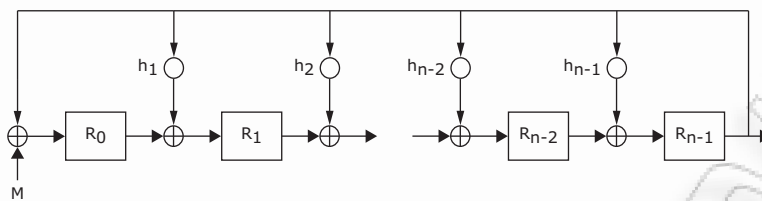


Εικόνα 2.14: Γενική μορφή του κυψελωτού αυτόματου.

Οι συνδέσεις μεταξύ των κυψελίδων περιγράφονται με τη βοήθεια των κανόνων (rules), οι οποίοι καθορίζουν την τιμή που θα λάβει η κυψελίδα στον επόμενο παλμό ρολογιού. Αν x_i είναι η τρέχουσα κυψελίδα, η τιμή που θα λάβει σύμφωνα με τον κανόνα-90 είναι $x_i(t+1) = x_{i+1}(t) \oplus x_{i-1}(t)$,¹⁰ ενώ σύμφωνα με τον κανόνα-150 είναι $x_i(t+1) = x_{i+1}(t) \oplus x_i(t) \oplus x_{i-1}(t)$ ¹¹. Έχει αποδειχθεί πως ένα κυψελωτό αυτόματο που χρησιμοποιεί τον κανόνα-90 και τον κανόνα-150 και αποτελείται από n κυψελίδες μπορεί να παράγει τη μέγιστη δυνατή ακολουθία μήκους $2^n - 1$ [66].

¹⁰Ονομάζεται κανόνας-90 διότι για τις τιμές εισόδου {111, 110, 101, 100, 011, 010, 001, 000} το αποτέλεσμα $x_i(t+1) = (0, 1, 0, 1, 1, 0, 1, 0)_2 = 90_{10}$.

¹¹Αντίστοιχα ο κανόνας-150 δίνει $\{1, 0, 0, 1, 0, 1, 1, 0\}_2 = 150_{10}$.

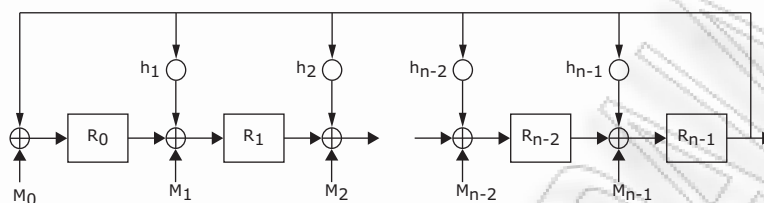


Εικόνα 2.15: Σειριακή ανάλυση υπογραφής.

Καθώς εφαρμόζονται τα διανύσματα δοκιμής παράγουν αποκρίσεις δοκιμής στις εξόδους του κυκλώματος. Κατά τη δοκιμή, ο εξοπλισμός δοκιμής είναι εκείνος που συγκρίνει τις αποκρίσεις με τις ορθές αποκρίσεις που βρίσκονται στην εσωτερική του μνήμη. Στην αυτοδοκιμή, κάτι τέτοιο δεν είναι εφικτό, διότι αφενός μεν χρειάζεται αρκετά μεγάλη μνήμη για την αποθήκευση των ορθών αποκρίσεων, αφετέρου δε ο χρόνος σύγκρισης είναι αρκετά μεγάλος. Τα κυκλώματα ανάλυσης αποκρίσεων (output response analyzers-ORA's) *συμπύσσουν (compact)* τις αποκρίσεις και παράγουν μια τελική απόκριση η οποία είναι γνωστή ως υπογραφή (signature). Η σύμπτυξη είναι μια διαδικασία συμπίεσης των αποκρίσεων κατά την οποία μπορεί να χαθεί πληροφορία. Έτσι είναι ιδιαίτερα σημαντικό να εξασφαλίζεται πως η υπογραφή που προκύπτει από το ορθό κύκλωμα και αυτή που προκύπτει από το εσφαλμένο να είναι διαφορετικές ώστε να είναι δυνατή η ανίχνευση ελαττωμάτων. Εάν είναι ίδιες λέγεται πως έχει συμβεί *απόκρυψη λαθών (error masking)* και η υπογραφή που παράγεται από το εσφαλμένο κύκλωμα αποτελεί *ψευδώνυμο (alias)* της ορθής υπογραφής.

Η σύμπτυξη των αποκρίσεων μπορεί να γίνει *μετρώντας τα ψηφία με τιμή 1 (ones count compaction)* σε ένα κύκλωμα το οποίο έχει μία έξοδο ή μετρώντας τον αριθμό των εναλλαγών από $0 \rightarrow 1$ και $1 \rightarrow 0$ (*transition count compaction*) αντίστοιχα σε ένα κύκλωμα με μια έξοδο. Οι πιο διαδεδομένες μέθοδοι σύμπτυξης αποκρίσεων είναι η *σειριακή ανάλυση υπογραφής (serial signature analysis)* και η *παράλληλη ανάλυση υπογραφής (parallel signature analysis)*, όπου η μεν πρώτη εφαρμόζεται όταν το κύκλωμα έχει μια έξοδο, ενώ η δεύτερη χρησιμοποιείται όταν το κύκλωμα έχει πολλαπλές εξόδους. Η σειριακή ανάλυση υπογραφής αποτελείται από ένα *καταχωρητή υπογραφής μονής εισόδου (single input signature register-SISR)* ο οποίος δεν είναι τίποτε άλλο από ένα γραμμικό καταχωρητή ολίσθησης με ανάδραση με μία επιπλέον πύλη XOR, όπως φαίνεται στην Εικόνα 2.15. Στην πραγματικότητα η σειριακή ανάλυση υπογραφής είναι ένας γεννιότερος κυκλικού ελέγχου πλεονασμού (cyclic redundancy check-CRC) και η υπογραφή είναι το υπόλοιπο της διαίρεσης του πολυωνύμου που ορίζεται από την ακολουθία τιμών της εξόδου και του χαρακτηριστικού πολυωνύμου του καταχωρητή. Η πιθανότητα ψευδωνυμίας της σειριακής ανάλυσης υπογραφής είναι $P(n) = (2^{L-n} - 1)/(2^L - 1)$ όπου L είναι το πλήθος των αποκρίσεων και n τα flip-flop του καταχωρητή. Όταν $L \gg n$ ισχύει $P(n) \approx 2^{-n}$. Για $n = 20$, η πιθανότητα ψευδωνυμίας είναι $P(n) < 2^{-20} = 0.0001\%$.

Η παράλληλη ανάλυση υπογραφής αποτελείται από ένα *καταχωρητή υπογραφής πολλα-*



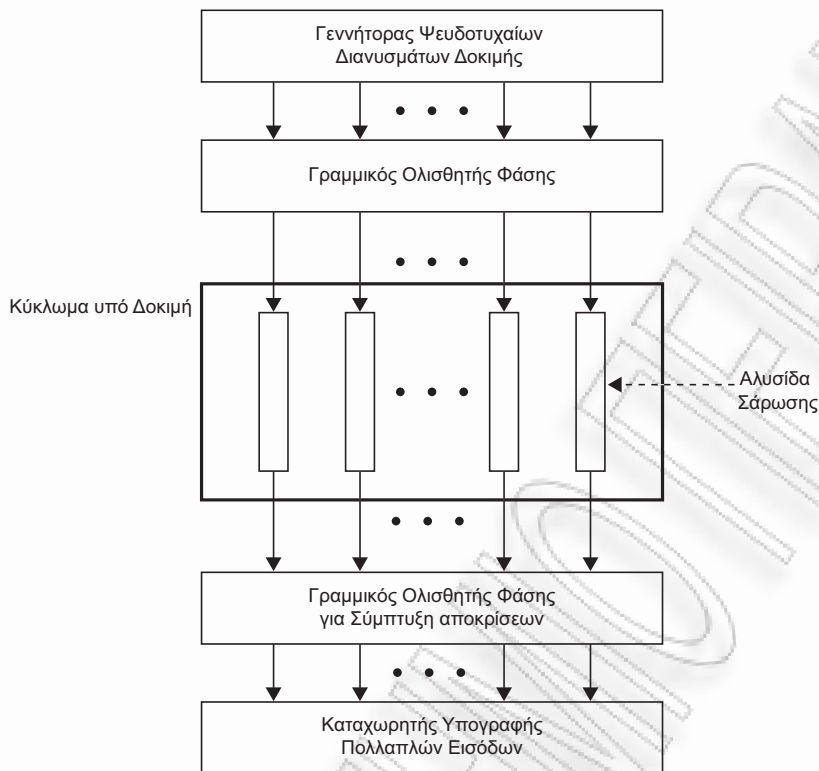
Εικόνα 2.16: Παράλληλη ανάλυση υπογραφής.

πλών, m εισόδων (*multiple input signature register-MISR*). Ο καταχωρητής υπογραφής αποτελείται από ένα γραμμικό καταχωρητή ολίσθησης και επιπλέον πύλες XOR, μια για κάθε είσοδο όπως φαίνεται στην Εικόνα 2.16. Η πιθανότητα ψευδωνυμίας για την παράλληλη ανάλυση υπογραφών είναι $P(n) = (2^{(mL-n)} - 1)/(2^{mL} - 1)$ και για $L \gg n$ ισχύει $P(n) \approx 2^{-n}$.

Η ενσωματωμένη αυτοδοκιμή μπορεί να εφαρμοστεί σε συνδυαστικά ή ακολουθιακά κυκλώματα είτε περιέχουν αλυσίδες σάρωσης είτε όχι. Στα κυκλώματα που δεν περιέχουν αλυσίδες σάρωσης η αυτοδοκιμή εφαρμόζεται τοποθετώντας κυκλώματα γέννησης διανυσμάτων δοκιμής και ανάλυσης αποκρίσεων πριν από τις κύριες εισόδους και έπειτα από τις κύριες εξόδους, αντίστοιχα. Αντιπροσωπευτικές τεχνικές παρουσιάζονται στις εργασίες [67], [68] ενώ το πλεονέκτημα που διαθέτουν είναι πως δεν απαιτείται αλλαγή στο κύκλωμα υπό δοκιμή. Όταν το κύκλωμα διαθέτει αλυσίδες σάρωσης, τότε αυτές μπορεί να χρησιμοποιηθούν για αυτοδοκιμή. Η αυτοδοκιμή σε αυτή την περίπτωση καλείται *δοκιμή-ανά-σάρωση (test-per-scan)*. Η πιο διαδεδομένη αρχιτεκτονική αυτοδοκιμής είναι το σύστημα *STUMPS*¹² [69]. Η αρχιτεκτονική *STUMPS* χρησιμοποιεί πολλαπλές αλυσίδες σάρωσης. Ένας γεννήτορας ψευδοτυχαίων διανυσμάτων τροφοδοτεί παράλληλα τις αλυσίδες σάρωσης. Όταν εφαρμοστούν τα διανύσματα, ενεργοποιούνται το ρολόι ή τα ρολόγια του συστήματος. Οι αποκρίσεις κάθε αλυσίδας σάρωσης συλλέγονται σε ένα καταχωρητή υπογραφής πολλαπλών εισόδων. Η αρχιτεκτονική παρουσιάζεται στην Εικόνα 2.17. Οι γραμμικοί ολισθητές φάσης (*linear phase shifters*) είναι προαιρετικοί και χρησιμοποιούνται ώστε να αυξήσουν το βαθμό τυχαιότητας των ψευδοτυχαίων διανυσμάτων δοκιμής αλλά και να μειώσουν την πιθανότητα απόκρυψης (*error masking*) κατά τη σύμπτυξη. Συνήθως υλοποιούνται με πύλες αποκλειστικό-Η (XOR).

Αν το κύκλωμα δεν περιέχει αλυσίδες σάρωσης, τότε τα ακολουθιακά του στοιχεία μπορούν να τροποποιηθούν ώστε να λειτουργούν είτε ως γεννήτορες διανυσμάτων είτε ως καταχωρητές υπογραφής. Στην περίπτωση αυτή η αυτοδοκιμή καλείται *δοκιμή-ανά-κύκλο (test-per-cycle)*. Στην εργασία [70] η δοκιμή-ανά-κύκλο πραγματοποιείται διαμερίζοντας το κύκλωμα σε ανεξάρτητα τμήματα (*modules*) τα οποία έχουν τους δικούς τους καταχωρητές εισόδου και εξόδου. Οι καταχωρητές επανασχεδιάζονται ώστε να πραγματοποιούν τις ακόλουθες λειτουργίες: 1) κανονική λειτουργία, 2) αλυσίδα σάρωσης, 3) γεννήτορας διανυσμάτων ή ανάλυση υπογραφής

¹²Self Testing Using MISR and Parallel SRSR (Shift Register Sequence Generator)-STUMPS.



Εικόνα 2.17: Δοκιμή-ανά-σάρωση με το σύστημα STUMPS.

και 4) κατάσταση επαναφοράς. Οι καταχωρητές που πραγματοποιούν τις παραπάνω λειτουργίες είναι γνωστοί ως *built-in logic block observer (BILBO)*. Οι καταχωρητές τροποποιημένοι (*Modified BILBO-MBILBO*) [71] και σύγχρονος (*concurrent BILBO-CBILBO*) [72] αποτελούν παραλλαγές των καταχωρητών BILBO και έχουν τη δυνατότητα να πραγματοποιούν ταυτόχρονα γέννηση διανυσμάτων δοκιμής και σύμπτυξη αποκρίσεων.

2.8 Αυτοδοκιμή μικροεπεξεργαστών

Οι μικροεπεξεργαστές ξεχωρίζουν από όλα τα υπόλοιπα λογικά κυκλώματα διότι αποτελούν την "καρδιά" κάθε υπολογιστικού συστήματος και είναι εκείνα τα κυκλώματα που παρέχουν υπολογιστικές δυνατότητες και καθιστούν την εκτέλεση του λογισμικού δυνατή. Ακόμα και η πιο απλή ενσωματωμένη εφαρμογή κρύβει ένα μικροεπεξεργαστή. Οι αυξημένες ανάγκες για υπολογιστική ισχύ αλλά και οι διαφορετικού είδους υπολογισμοί έχουν οδηγήσει στην ενσωμάτωση μικροεπεξεργαστών γενικού σκοπού και επεξεργασίας σήματος σε ένα ολοκληρωμένο κύκλωμα, το οποίο πλαισιώνεται από μνήμες και περιφερειακά και καλείται σύστημα-σε-τσιπ (System-on-Chip-SoC).

Αυτή η μοναδικότητα των μικροεπεξεργαστών καθιστά και την δοκιμή τους ιδιαίτερη και ξεχωριστή σε σχέση με τα υπόλοιπα λογικά κυκλώματα. Αν και συχνά οι μικροεπεξεργαστές αποτελούν ένα μικρό μέρος ενός ενσωματωμένου συστήματος και καταλαμβάνουν περίπου το 20% ενός συστήματος σε ένα ολοκληρωμένο κύκλωμα SoC, η αποτελεσματική δοκιμή τους είναι υψηλότερης αξίας σε σχέση με τα υπόλοιπα κυκλώματα. Για παράδειγμα, εάν διαπιστωθεί πως υπάρχουν ελαττώματα σε κάποια τμήματα της μνήμης, τότε αυτά τα τμήματα είναι δυνατό να απομονωθούν ώστε να μην χρησιμοποιούνται. Κάτι τέτοιο δεν μπορεί να γίνει με ένα μικροεπεξεργαστή. Επίσης, τα ελαττώματα στον μικροεπεξεργαστή, ανάλογα με το σημείο στο οποίο βρίσκονται, μπορεί να αλλοιώσουν την ορθή εκτέλεση κάθε εντολής.

Οι μικροεπεξεργαστές σχεδιάζονται με στόχο να καλύψουν τις ακόλουθες παραμέτρους:

- Υψηλή απόδοση
- Χαμηλή κατανάλωση ενέργειας
- Μικρό μέγεθος

Οι τρεις αυτές παράμετροι καθορίζουν σε μεγάλο βαθμό την επιτυχία του μικροεπεξεργαστή στην αγορά. Καθώς οι μικροεπεξεργαστές καλούνται να ικανοποιήσουν τις ολοένα και αυξανόμενες απαιτήσεις σε υπολογιστική ισχύ, σχεδιάζονται με μεγάλη προσοχή ώστε να είναι όσο το δυνατό περισσότερο αποδοτικοί αυξάνοντας με αυτό τον τρόπο σε μεγάλο βαθμό την πολυπλοκότητά τους. Παράλληλα, κατά τον σχεδιασμό τους λαμβάνεται σημαντικά υπόψη ο παράγοντας κατανάλωση ενέργειας. Η υψηλή κατανάλωση ενέργειας αποτελεί σημαντικό πρόβλημα για τα ενσωματωμένα συστήματα τα οποία τροφοδοτούνται με μπαταρίες, ενώ η υψηλή κατανάλωση συνεπάγεται και αυξημένη θερμοκρασία, η αντιμετώπιση της οποίας απαιτεί τεχνικές ψύξης οι οποίες επιδρούν αρνητικά τόσο στο κόστος όσο και στο μέγεθος του ενσωματωμένου συστήματος. Το μέγεθος του μικροεπεξεργαστή αποτελεί επίσης μια σημαντική παράμετρο, καθώς επιδρά στο μέγεθος της πλακέτας στην οποία βρίσκεται. Μικρότερο μέγεθος σημαίνει μικρότερη πλακέτα αλλά και περισσότερο χώρο για τα υπόλοιπα μέρη ενός ενσωματωμένου συστήματος. Η πρόοδος της τεχνολογίας των ημιαγωγών επιτρέπει την ενσωμάτωση ακόμα περισσότερων τρανζίστορ σε ένα ολοκληρωμένο κύκλωμα με αποτέλεσμα την υποβάθμιση της παραμέτρου του μεγέθους του μικροεπεξεργαστή.

Τα χαρακτηριστικά του σχεδιασμού των μικροεπεξεργαστών επηρεάζουν άμεσα και τις τεχνικές που εφαρμόζονται για τη δοκιμή τους. Οι παραδοσιακές τεχνικές δοκιμής αλλά και αυτοδοκιμής δεν είναι ιδιαίτερα φιλικές ως προς τα χαρακτηριστικά σχεδιασμού των μικροεπεξεργαστών διότι επιφέρουν σημαντικότερες επιβαρύνσεις στο μέγεθος, την κατανάλωση ενέργειας και την συχνότητα λειτουργίας. Η δοκιμή των επεξεργαστών με εξοπλισμό δοκιμής αλλά και η αυτοδοκιμή γίνεται ολοένα και δυσκολότερη παρουσιάζοντας νέες προκλήσεις.

Η ταχύτητα με την οποία ο εξοπλισμός δοκιμής εφαρμόζει τις ακολουθίες διανυσμάτων δοκιμής στον μικροεπεξεργαστή είναι καθοριστικής σημασίας για την ποιότητα της δοκιμής.

Αν η δοκιμή πραγματοποιηθεί σε χαμηλότερη συχνότητας λειτουργίας από αυτή στην οποία ο μικροεπεξεργαστής θα κληθεί να λειτουργήσει, η δοκιμή είναι χαμηλής ποιότητας. Υπάρχουν ατέλειες οι οποίες εμφανίζονται ως ελαττώματα καθυστέρησης (delay faults), τα οποία αλλοιώνουν τη λειτουργία του μικροεπεξεργαστή και μπορούν να ανιχνευθούν μόνο όταν ο μικροεπεξεργαστής λειτουργεί στην πραγματική του συχνότητα. Η δοκιμή που πραγματοποιείται στην πραγματική συχνότητα λειτουργίας ονομάζεται at-speed testing. Η ταχύτητα λειτουργίας των μικροεπεξεργαστών ακολουθεί ένα φρενιόνη ρυθμό αύξησης τον οποίο ο εξοπλισμός δοκιμής αδυνατεί να ακολουθήσουν. Όπως αναφέρεται στην εργασία [12], το 1985 για τη δοκιμή του επεξεργαστή 286 με ταχύτητα λειτουργίας 8MHz η ακρίβεια του εξοπλισμού δοκιμής για οδήγηση και παρατήρηση των ακροδεκτών του επεξεργαστή ήταν της τάξης του 1 nanosecond. Δεκαπέντε χρόνια αργότερα, το 2000 για τη δοκιμή των επεξεργαστών Pentium-III με ταχύτητα λειτουργίας 700 MHz, η ακρίβεια εξοπλισμού δοκιμής είναι περίπου 100 picosecond. Ενώ, μέσα σε αυτό το χρονικό διάστημα η ταχύτητα των μικροεπεξεργαστών σχεδόν εκατονταπλασιάστηκε η ταχύτητα του εξοπλισμού δοκιμής μόλις δεκαπλασιάστηκε. Πάντα θα υπάρχει ένας ταχύτερος μικροεπεξεργαστής από τον ταχύτερο εξοπλισμό δοκιμής.

Καθώς η ταχύτητα των μικροεπεξεργαστών αυξάνεται ο εξοπλισμός δοκιμής έρχεται αντιμέτωπος με το πρόβλημα της *ανακρίβειας των μετρήσεων (ATE/tester measurement inaccuracies)*. Η ακρίβεια των μετρήσεων είναι άμεσα συνδεδεμένη με την ευαισθησία των ηλεκτρονικών συστημάτων του εξοπλισμού δοκιμής. Καθώς όμως η συχνότητα λειτουργίας των μικροεπεξεργαστών είναι της τάξης των GHz και σε συνδυασμό με τη χαμηλότερη τάση λειτουργίας τους, ο εξοπλισμός δοκιμής καλείται να πραγματοποιήσει μετρήσεις μικρών μεταβολών τάσης σε διάστημα μερικών picosecond-ps. Η κατασκευή ηλεκτρονικού εξοπλισμού είναι ιδιαίτερα δύσκολη, δαπανηρή και αποτελεί το κυριότερο κόστος για την κατασκευή του εξοπλισμού δοκιμής. Εξαιτίας της δυσκολίας στις μετρήσεις των τάσεων στους ακροδέκτες του μικροεπεξεργαστή από τον εξοπλισμό δοκιμής, μια πιθανή απόκλιση ή ανακρίβεια, στη μέτρηση των τάσεων μπορεί να οδηγήσει τον εξοπλισμό δοκιμής να χαρακτηρίσει ένα ορθά κατασκευασμένο επεξεργαστή ως ελαττωματικό. Σε μια τέτοια περίπτωση ο επεξεργαστής δε θα προωθηθεί στην αγορά. Επομένως, οι ατελείς μετρήσεις του εξοπλισμού δοκιμής αν και είναι ανεπιθύμητες είναι υπαρκτές και οδηγούν στην μείωση της εσοδείας και στην αύξηση του κόστους του μικροεπεξεργαστή.

Οι μικροεπεξεργαστές είναι ιδιαίτερα πολύπλοκα κυκλώματα τόσο που κανένα αυτόματο εργαλείο δεν έχει τη δυνατότητα να παράγει διανύσματα δοκιμής για αυτά. Περιέχουν τόσα πολλά ακολουθιακά στοιχεία flip-flop και έχουν μεγάλο ακολουθιακό βάθος (sequential depth) που ξεπερνούν κατά πολύ τις δυνατότητες των αυτόματων εργαλείων εξαγωγής διανυσμάτων. Η σχεδίαση για αυξημένη δοκιμαστικότητα, με αλυσίδες σάρωσης αποτελεί μονόδρομο για αποτελεσματική δοκιμή με υψηλά ποσοστά κάλυψης ελαττωμάτων. Το τίμημα των αλυσίδων σάρωσης δεν είναι αμελητέο. Οι αλυσίδες σάρωσης επιβαρύνουν το μέγεθος του μικροεπεξεργαστή (area overhead) και δυσχεραίνουν τη δρομολόγηση των καλωδιώσεων (routing). Η

επιβάρυνση στο μέγεθος του επεξεργαστή συνεπάγεται τη μείωση της εσοδείας καθώς λιγότερα κυκλώματα κατασκευάζονται στα πλακίδια πυριτίου (wafer).

Παράλληλα οι αλυσίδες σάρωσης επηρεάζουν τα κρίσιμα μονοπάτια (*critical paths*) του μικροεπεξεργαστή τα οποία καθορίζουν τη συχνότητα λειτουργίας και, γενικότερα, τείνουν να μειώνουν την συχνότητα λειτουργίας (performance penalty). Οι μικροεπεξεργαστές σχεδιάζονται με κύριο γνώμονα την υψηλή απόδοση και οποιαδήποτε σχεδιαστική αλλαγή που μειώνει την επίδοση δεν είναι πάντα ευπρόσδεκτη.

Η δοκιμή με αλυσίδες σάρωσης θέτει τον μικροεπεξεργαστή σε ειδική κατάσταση λειτουργίας, με αποτέλεσμα να ενεργοποιούνται τμήματά του τα οποία θα παρέμεναν ανενεργά κατά την κανονική του λειτουργία. Το φαινόμενο αυτό έχει μια δυσάρεστη παρενέργεια η οποία καλείται *υπερδοκιμή (overtesting)*, όπου ο μικροεπεξεργαστής ελέγχεται για ελαττώματα τα οποία αν και μπορεί να υπάρχουν δεν επηρεάζουν την ορθή λειτουργία του. Η δοκιμή με αλυσίδες σάρωσης δεν έχει καμία τέτοια πληροφορία για αυτά τα ελαττώματα. Το φαινόμενο της υπερδοκιμής επιδρά αρνητικά στην εσοδεία διότι είναι πιθανό να απορριφθούν μικροεπεξεργαστές οι οποίοι αν και περιέχουν ελαττώματα, τα ελαττώματα αυτά δεν υπονομεύουν την ορθή λειτουργία του μικροεπεξεργαστή. Για άλλη μια φορά το κόστος αυξάνεται.

Η δοκιμή με αλυσίδες σάρωσης τείνει να αυξάνει το χρόνο δοκιμής καθώς ένας μεγάλος αριθμός κύκλων ρολογιού χρειάζεται για να εφαρμοστούν τα διανύσματα δοκιμής. Το πρόβλημα λύνεται χρησιμοποιώντας πολλαπλές αλυσίδες σάρωσης, αλλά σε αυτή την περίπτωση πρέπει να ληφθεί μέριμνα για επιπλέον ακροδέκτες στο ολοκληρωμένο κύκλωμα.

Εναλλακτικά, η δοκιμή ενός μικροεπεξεργαστή μπορεί να πραγματοποιηθεί εφαρμόζοντας αυτοδοκιμή. Η αυτοδοκιμή βασίζεται στη σχεδίαση για αυξημένη δοκιμαστικότητα εισάγοντας επιπλέον υλικό. Το επιπλέον υλικό μέσα στον μικροεπεξεργαστή αναλαμβάνει την γέννηση και εφαρμογή των διανυσμάτων δοκιμής καθώς και τη σύμπτυξη των αποκρίσεων. Με αυτό τον τρόπο η αυτοδοκιμή επιτυγχάνει τη δοκιμή στην πραγματική συχνότητα λειτουργίας του μικροεπεξεργαστή (at-speed test) και καλύπτει το σημαντικότερο μειονέκτημα της δοκιμής που βασίζεται σε εξοπλισμό δοκιμής υψηλής ταχύτητας. Η αυτοδοκιμή επιτυγχάνει δοκιμή χαμηλού κόστους διότι επιτρέπει τη χρήση εξοπλισμού δοκιμής χαμηλής ταχύτητας και κατ'επέκταση χαμηλού κόστους (low-speed, low cost ATE). Επιπλέον, παύει να υπάρχει το πρόβλημα της ανακρίβειας των μετρήσεων του εξοπλισμού δοκιμής. Η δοκιμή των μικροεπεξεργαστών με αυτοδοκιμή παρέχει υψηλότερης ποιότητας δοκιμής σε σχέση με τη εξωτερική δοκιμή που εξαρτάται από τη ταχύτητα του εξοπλισμού δοκιμής. Επιπλέον, με την αυτοδοκιμή είναι δυνατή η δοκιμή του μικροεπεξεργαστή στο πεδίο της λειτουργίας χωρίς να απαιτείται ειδικός εξοπλισμός. Η αυτοδοκιμή μικροεπεξεργαστών με επιπλέον υλικό έχει εφαρμοστεί στο παρελθόν στις εργασίες [73], [74]. Όμως αν και η αυτοδοκιμή των επεξεργαστών αποτελεί ένα βήμα προς την σωστή κατεύθυνση, παρουσιάζει και αυτή σημαντικά μειονεκτήματα. Κατά πρώτον αυξάνεται το μέγεθος του μικροεπεξεργαστή αφού εισάγεται επιπλέον υλικό. Κατά δεύτερον απαιτείται ειδική σχεδίαση του μικροεπεξεργαστή ώστε να μπορεί να εφαρμοστεί η

αυτοδοκιμή. Επιπλέον, στην αυτοδοκιμή ο μικροεπεξεργαστής τίθεται σε ειδική κατάσταση λειτουργίας ή αλλιώς σε κατάσταση δοκιμής (test mode), όπου καταναλώνει επιπλέον ενέργεια σε σχέση με αυτή που θα καταναλώνε στην κανονική κατάσταση λειτουργίας. Ο παράγοντας αυτός είναι ιδιαίτερα σημαντικός σε ενσωματωμένες εφαρμογές όπου τα συστήματα τροφοδοτούνται από μπαταρίες.

Τα μειονεκτήματα της αυτοδοκιμής με ενσωματωμένο υλικό έρχεται να καλύψει η αυτοδοκιμή των μικροεπεξεργαστών με ενσωματωμένο λογισμικό, χωρίς την εισαγωγή επιπλέον υλικού, διατηρώντας ταυτόχρονα τα πλεονεκτήματα που παρουσιάζει η αυτοδοκιμή.

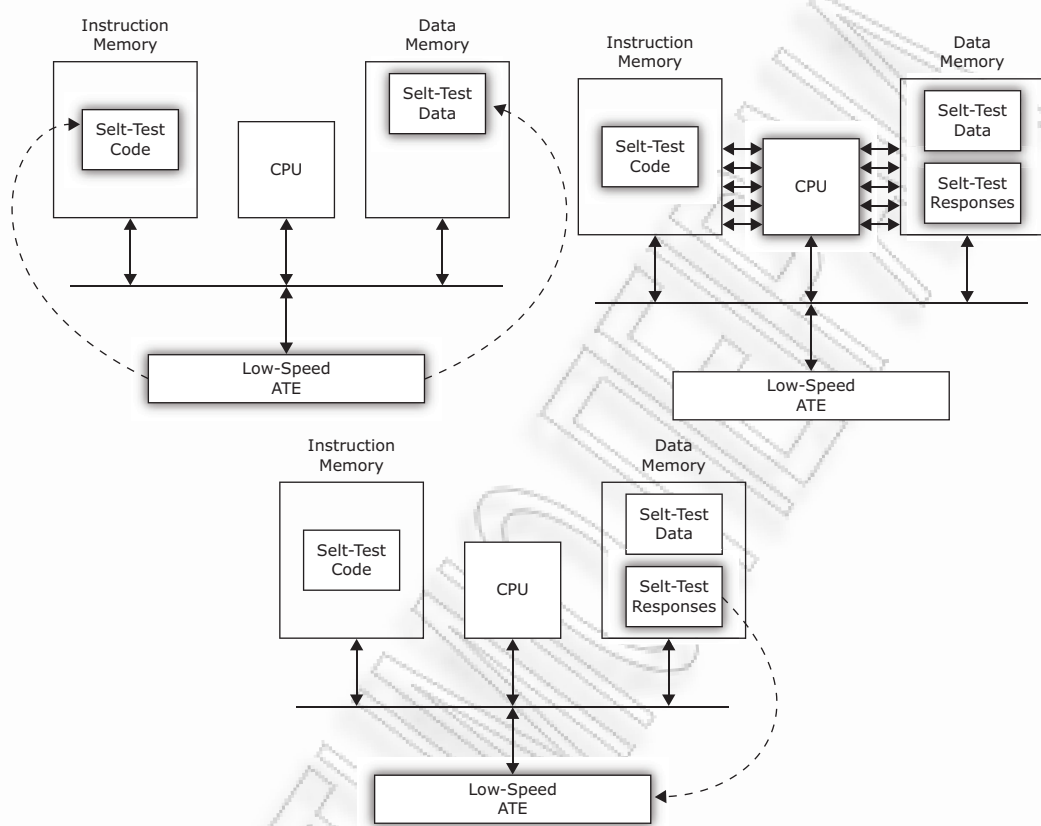
2.8.1 Η έννοια της αυτοδοκιμής με ενσωματωμένο λογισμικό

Η αυτοδοκιμή των μικροεπεξεργαστών με ενσωματωμένο λογισμικό (*Software-based Self-test—SBST*) βασίζεται στην ιδέα πως το n αρχιτεκτονική συνόλου εντολών των μικροεπεξεργαστή (*Instruction set architecture—ISA*) παρέχει τη δυνατότητα για εξαγωγή και εφαρμογή των ακολουθιών διανυσμάτων δοκιμής αλλά και τη δυνατότητα για επεξεργασία των αποκρίσεων, ώστε να διαπιστωθεί εάν ένας μικροεπεξεργαστής περιέχει ελαττώματα ή όχι. Κατά την αυτοδοκιμή με ενσωματωμένο λογισμικό, ο μικροεπεξεργαστής εκτελεί ένα σύνολο ρουτινών, οι οποίες εφαρμόζουν στις επιμέρους μονάδες του ακολουθίες διανυσμάτων δοκιμής. Στη συνέχεια, ο μικροεπεξεργαστής συλλέγει τις αποκρίσεις των διανυσμάτων και τις αποθηκεύει στη μνήμη. Εναλλακτικά, το ενσωματωμένο λογισμικό μπορεί να συμπτύσσει (compact) τις αποκρίσεις ώστε στη μνήμη να αποθηκεύεται ένας μικρός αριθμός υπογραφών (signatures).

Η αυτοδοκιμή ξεκινά με τον εξοπλισμό δοκιμής χαμηλής συχνότητας να φορτώνει το λογισμικό αυτοδοκιμής (εντολές και δεδομένα) στις αντίστοιχες μνήμες όπως φαίνεται στην Εικόνα 2.18. Έχουμε υποθέσει πως υπάρχουν ξεχωριστές μνήμες εντολών και δεδομένων. Η αυτοδοκιμή εφαρμόζεται με τον ίδιο τρόπο όταν υπάρχει κοινή μνήμη εντολών και δεδομένων. Έπειτα, ο εξοπλισμός δοκιμής ενεργοποιεί τον επεξεργαστή ο οποίος αρχίζει να εκτελεί τις εντολές του ενσωματωμένου λογισμικού δοκιμής. Κατά την εκτέλεση του λογισμικού ο μικροεπεξεργαστής αποθηκεύει στη μνήμη δεδομένων τις αποκρίσεις των διανυσμάτων δοκιμής, όπως φαίνεται στην Εικόνα 2.18. Ο μικροεπεξεργαστής "τρέχει" στην πραγματική συχνότητα λειτουργίας. Μετά την ολοκλήρωση της εκτέλεσης του λογισμικού ο εξοπλισμός δοκιμής διαβάσει τις αποκρίσεις από τη μνήμη δεδομένων και τις συγκρίνει με τις αναμενόμενες για να διαπιστώσει αν ο μικροεπεξεργαστής περιέχει ελαττώματα.

Έστω ότι η αριθμητική και λογική μονάδα (ALU) του επεξεργαστή περιέχει ένα ελάττωμα f το οποίο ανιχνεύεται με τα διανύσματα δοκιμής t_1 , και t_2 , όταν η αριθμητική και λογική μονάδα εκτελεί την πράξη λογικό-Η (logic-OR) (βλ. Εικόνα 2.19). Τα βήματα για την ανίχνευση του ελαττώματος στην αριθμητική και λογική μονάδα έχουν ως εξής:

1. $r1 \leftarrow t1, r2 \leftarrow t2$



Εικόνα 2.18: Αυτοδοκιμή μικροεπεξεργαστών με ενσωματωμένο λογισμικό

Το ενσωματωμένο λογισμικό αποθηκεύει τα διανύσματα δοκιμής $t1$, και $t2$ στους καταχωρητές $r1$ και $r2$ αντίστοιχα, χρησιμοποιώντας το σύνολο εντολών του μικροεπεξεργαστή.

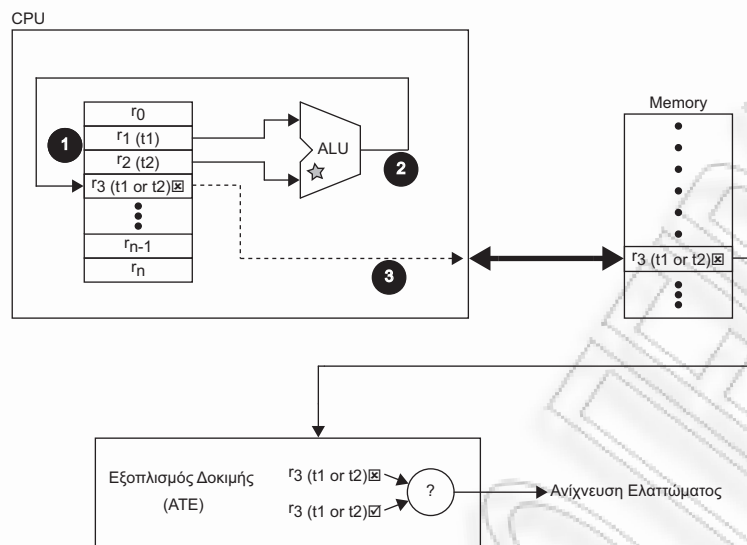
2. $r3 \leftarrow r1 \text{ OR } r2$

Καθώς εκτελείται η πράξη λογικό-Η, το ελάττωμα f στην αριθμητική και λογική μονάδα ενεργοποιείται και το λανθασμένο αποτέλεσμα της πράξης αποθηκεύεται στον καταχωρητή $r3$.

3. $\text{memory} \leftarrow r3$

Το ενσωματωμένο λογισμικό αποθηκεύει το περιεχόμενο του καταχωρητή $r3$ στη μνήμη. Ο εξοπλισμός δοκιμής διαβάζει τα περιεχόμενα της μνήμης και τα συγκρίνει με τις αναμενόμενες αποκρίσεις. Στη συγκεκριμένη περίπτωση, ο εξοπλισμός δοκιμής θα διαπιστώσει πως η τιμή της μνήμης (στην οποία έχει αποθηκευτεί ο $r3$) διαφέρει από αυτή που αναμένει, πράγμα που σηματοδοτεί την ανίχνευση του ελαττώματος f .

Η αυτοδοκιμή με ενσωματωμένο λογισμικό αποτελεί τον πιο "φυσικό" τρόπο αυτοδοκιμής των επεξεργαστών καθώς είναι τελείως διαφανής (transparent) ως προς τον μικροεπεξεργαστή



Εικόνα 2.19: Ανίχνευση ελαττώματος στην ALU με ενσωματωμένο λογισμικό

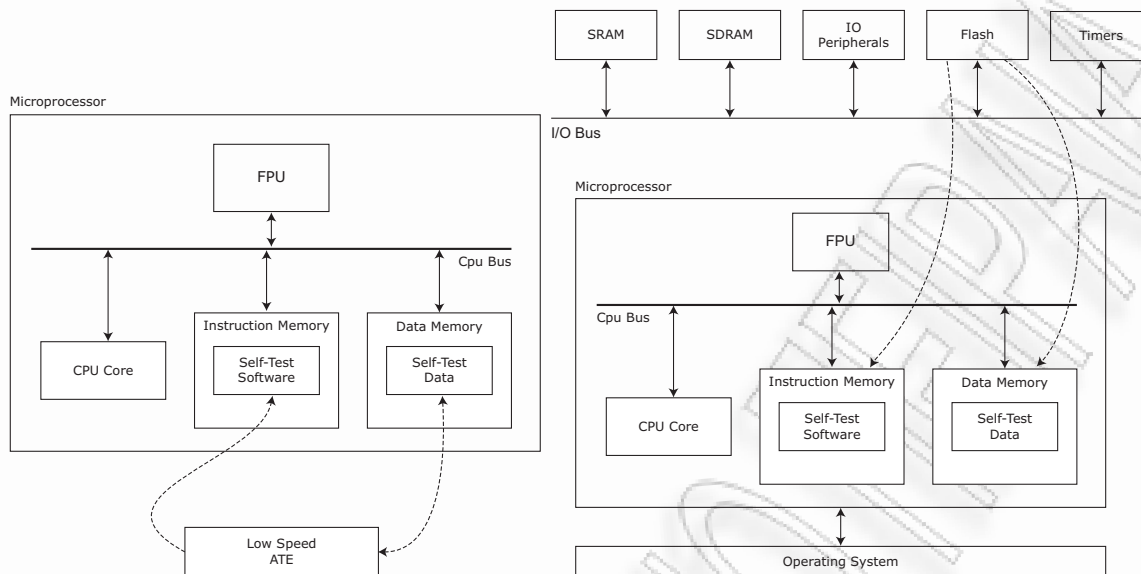
και το μόνο που απαιτεί είναι η ύπαρξη μνήμης στο ίδιο ολοκληρωμένο κύκλωμα με τον μικροεπεξεργαστή. Τα πλεονεκτήματα που παρουσιάζει η αυτοδοκιμή με ενσωματωμένο λογισμικό είναι ποικίλα.

- Η αυτοδοκιμή με ενσωματωμένο λογισμικό αποτελεί μια "φιλική" μέθοδο δοκιμής των μικροεπεξεργαστών. Βασίζεται αποκλειστικά και μόνο στο σύνολο εντολών του επεξεργαστή και δεν απαιτεί καμία μετατροπή στο σχεδιασμό του επεξεργαστή για αύξηση της δοκιμαστικότητας. Δυσάρεστες συνέπειες όπως επιβάρυνση του μεγέθους του επεξεργαστή ή της κατανάλωσης ενέργειας και μείωση της απόδοσής του παύουν να υφίστανται.
- Δεν υπάρχουν περιορισμοί ως προς τον τύπο του εξοπλισμού δοκιμής που χρησιμοποιείται κατά το πείραμα της δοκιμής. Η δοκιμή είναι ανεξάρτητη του εξοπλισμού δοκιμής και για αυτό το λόγο, η αυτοδοκιμή με ενσωματωμένο λογισμικό αποτελεί μια *χαμηλού κόστους* λύση για δοκιμή των μικροεπεξεργαστών αφού μπορεί να χρησιμοποιηθεί εξοπλισμός δοκιμής χαμηλής ταχύτητας και κατά συνέπεια χαμηλού κόστους. Η ταχύτητα με την οποία ο εξοπλισμός δοκιμής αποθηκεύει το ενσωματωμένο λογισμικό στη μνήμη και με την οποία διαβάζει τις αποκρίσεις από αυτή δεν επηρεάζει την αυτοδοκιμή. Επιπλέον, η δοκιμή δεν εξαρτάται από τυχόν ανακρίβειες στις μετρήσεις του εξοπλισμού δοκιμής οι οποίες απειλούν την εσοδεία.
- Όπως και στην αυτοδοκιμή με υλικό, η αυτοδοκιμή με λογισμικό επιτυγχάνει την πραγματοποίηση της δοκιμής στην ταχύτητα λειτουργίας του επεξεργαστή (at-speed) και όχι σε κάποια χαμηλότερη επιτρέποντας την ανίχνευση χρονικών ελαττωμάτων (delay faults)

που εμφανίζονται μόνο σε υψηλές συχνότητες λειτουργίας.

- Η αυτοδοκιμή με ενσωματωμένο λογισμικό δεν αυξάνει την κατανάλωση ενέργειας όπως η αυτοδοκιμή με υλικό καθώς η δοκιμή πραγματοποιείται στην κανονική λειτουργία και όχι σε κάποια ειδική κατάσταση δοκιμής. Η μείωση της κατανάλωσης ενέργειας είναι άμεσος αντίκτυπος της απουσίας ειδικού υλικού για δοκιμή.
- Η δοκιμή πραγματοποιείται μόνο για εκείνα τα ελαττώματα που επηρεάζουν την ορθή λειτουργία του επεξεργαστή. Το φαινόμενο της υπερδοκιμής (overtesting) αποφεύγεται και εξασφαλίζεται μεγαλύτερη εσοδεία η οποία μειώνει το κόστος δοκιμής του μικροεπεξεργαστή.
- Το ενσωματωμένο λογισμικό αυτοδοκιμής μπορεί να χρησιμοποιηθεί τόσο για τη δοκιμή αμέσως μετά την κατασκευή του μικροεπεξεργαστή όσο και στο πεδίο λειτουργίας του (on-line, in-field testing). Το λογισμικό αυτοδοκιμής εκτελείται όπως και το υπόλοιπο ενσωματωμένο λογισμικό και έτσι "ταιριάζει" περισσότερο για αυτοδοκιμή στο πεδίο λειτουργίας σε σχέση με την αυτοδοκιμή βασισμένη σε υλικό που απαιτεί να τεθεί ο μικροεπεξεργαστής σε ειδική κατάσταση δοκιμής. Έτσι, η αυτοδοκιμή με λογισμικό αποτελεί μια λύση δοκιμής η οποία συνοδεύει τον μικροεπεξεργαστή σε όλο τον κύκλο ζωής του. Καθώς, η αυτοδοκιμή δεν αυξάνει την κατανάλωση ενέργειας του μικροεπεξεργαστή, αποτελεί ιδανική επιλογή για τη δοκιμή των μικροεπεξεργαστών που βρίσκονται σε ενσωματωμένα συστήματα τα οποία τροφοδοτούνται από μπαταρίες.
- Η δοκιμή με ενσωματωμένο λογισμικό επιτρέπει την επαναχρησιμοποίηση του ίδιου του μικροεπεξεργαστή για την εξαγωγή, την εφαρμογή διανυσμάτων δοκιμής και τη συλλογή αποκρίσεων για τη δοκιμή άλλων υποσυστημάτων που βρίσκονται σε ένα σύστημα σε ένα ολοκληρωμένο κύκλωμα (SoC). Κάτι τέτοιο δεν είναι εύκολο να γίνει όταν η αυτοδοκιμή πραγματοποιείται με υλικό. Επιπλέον, εάν κατά το πείραμα της δοκιμής παραστεί η ανάγκη για βελτίωση του συνόλου των διανυσμάτων δοκιμής, ίσως για παράδειγμα για ένα άλλο μοντέλο ελαττωμάτων, το λογισμικό αυτοδοκιμής μπορεί πολύ πιο εύκολα να προσαρμοστεί σε σχέση με το υλικό.

Το ενσωματωμένο λογισμικό αυτοδοκιμής συνοδεύει το μικροεπεξεργαστή σε όλο τον κύκλο ζωής του τόσο από το στάδιο της δοκιμής στο εργοστάσιο παραγωγής όσο και στο περιβάλλον λειτουργίας του. Βέβαια οι απαιτήσεις για την ανάπτυξη του ενσωματωμένου λογισμικού στα δύο αυτά περιβάλλοντα είναι διαφορετικές. Η σημαντικότερη διαφορά είναι πως το λογισμικό αυτοδοκιμής κατά τη δοκιμή στο εργοστάσιο παραγωγής διαχειρίζεται αποκλειστικά τον επεξεργαστή και τους πόρους του ενώ αυτή η αποκλειστικότητα παύει να υπάρχει από τη στιγμή που ο επεξεργαστής τοποθετείται σε ένα σύστημα στο περιβάλλον λειτουργίας. Σε αυτή την περίπτωση το λογισμικό αυτοδοκιμής συναγωνίζεται το λογισμικό του χρήστη τόσο σε χρόνο



Εικόνα 2.20: Τα διαφορετικά περιβάλλοντα δοκιμής στα οποία εφαρμόζεται το λογισμικό αυτοδοκιμής

εκτέλεσης όσο και σε μνήμη. Τα διαφορετικά περιβάλλοντα στα οποία το λογισμικό αυτοδοκιμής καλείται να εφαρμοστεί παρουσιάζονται στην Εικόνα 2.20. Είναι εύλογο πως το λογισμικό αυτοδοκιμής θα πρέπει να είναι αποτελεσματικά σχεδιασμένο ώστε να μην επιβαρύνει τον μικροεπεξεργαστή.

Το λογισμικό αυτοδοκιμής όπως έχει γίνει αντιληπτό φροντίζει για την εφαρμογή συγκεκριμένων διανυσμάτων δοκιμής στις υπομονάδες του μικροεπεξεργαστή. Τα διανύσματα δοκιμής μπορούν να παραχθούν είτε από αυτόματα εργαλεία εξαγωγής διανυσμάτων δοκιμής, είτε από γεννιότερες ψευδοτυχαίων διανυσμάτων είτε να εξαχθούν βάσει της συμμετρικής δομής των υπομονάδων. Αντίστοιχα, το λογισμικό αυτοδοκιμής που παράγεται για την εφαρμογή των διανυσμάτων της κάθε μεθόδου έχει διαφορετικά χαρακτηριστικά, τα οποία αποτελούν αντικείμενο μελέτης του Κεφαλαίου 5 της διδακτορικής διατριβής.

Όπως φαίνεται στην Εικόνα 2.20, ο μικροεπεξεργαστής δεν βρίσκεται ποτέ απομονωμένος σε ένα σύστημα, αλλά συνοδεύεται και από άλλες μονάδες, για παράδειγμα μονάδες σειριακής επικοινωνίας, όπως UART, I2C, CAN αλλά και ταχύτερες μονάδες, όπως USB και Ethernet. Επιπλέον, δεν νοείται ένα σύστημα με μικροεπεξεργαστής δίχως μνήμες, για παράδειγμα SRAM, DRAM, Flash, Rom. Μια από τις συνηθέστερες και σημαντικότερες μονάδες που συμπληρώνουν τη λειτουργικότητα του μικροεπεξεργαστή είναι ο συνεπεξεργαστής κινητής υποδιαστολής. Δίχως τη μονάδα κινητής υποδιαστολής, ο μικροεπεξεργαστής δεν έχει τρόπο να επεξεργαστεί πραγματικούς αριθμούς εκτελώντας πράξεις με μεγάλη ακρίβεια. Είναι αντιληπτό, πως όλες αυτές οι μονάδες είναι απαραίτητο να δοκιμάζονται στο πεδίο λειτουργίας ώστε να εξασφαλίζεται η εύρυθμη λειτουργία του συστήματος. Το λογισμικό αυτοδοκιμής

αποτελεί τον καταλληλότερο τρόπο ώστε να πραγματοποιηθεί η δοκιμή. Έτσι ο μικροεπεξεργαστής αφού εκτελέσει το ενσωματωμένο λογισμικό για να δοκιμάσει τον εαυτό του, μπορεί να συνεχίσει εκτελώντας λογισμικό αυτοδοκιμής για τις υπόλοιπες μονάδες του συστήματος. Στο Κεφάλαιο 5 παρουσιάζεται η ανάπτυξη λογισμικού αυτοδοκιμής για τις μονάδες κινητής υποδιαστολής.

2.8.2 Επισκόπηση της βιβλιογραφίας

Η ιδέα της αυτοδοκιμής των μικροεπεξεργαστών με ενσωματωμένο λογισμικό δεν είναι νέα. Αν και τα πρώτα βήματα έγιναν από τις αρχές της δεκαετίας του 1970, η εργασία [75] αποτελεί την πρώτη σημαντική μελέτη της αυτοδοκιμής των μικροεπεξεργαστών με ενσωματωμένο λογισμικό. Στην εργασία [75], οι Thatte και Abraham ανέπτυξαν ένα θεωρητικό μοντέλο, το S-graph, για την αναπαράσταση του μικροεπεξεργαστή, η κατασκευή του οποίου βασίστηκε στο σύνολο εντολών και την περιγραφή του μικροεπεξεργαστή στο μοντέλο RTL. Το λογισμικό αυτοδοκιμής στόχευε σε λειτουργικά ελαττώματα (functional faults) πραγματοποιώντας λειτουργική δοκιμή (functional testing). Οι Thatte και Abraham ανέπτυξαν λογισμικό αυτοδοκιμής για ένα μικροεπεξεργαστή της εταιρίας Helwert-Packard μεγέθους 8-bit. Το λογισμικό αυτοδοκιμής είχε συνολικό μέγεθος 8K και επιτύχανε ποσοστό κάλυψης ελαττωμάτων μόνιμης τιμής 96%.

Η εργασία [76] αποτελεί την πρώτη εργασία όπου η αυτοδοκιμή πραγματοποιείται για την ανίχνευση δομικών ελαττωμάτων (structural faults). Στην εργασία αυτή, οι Tururi και Abraham χρησιμοποίησαν εμπορικά εργαλεία εξαγωγής διανυσμάτων για τις υπομονάδες του επεξεργαστή. Κατά την εξαγωγή των διανυσμάτων, εφάρμοσαν περιορισμούς (constraints) στο αυτόματο εργαλείο ώστε τα διανύσματα δοκιμής να είναι εφαρμόσιμα από το σύνολο των εντολών του μικροεπεξεργαστή.

Μια αναλυτική ιστορική ανασκόπηση των διαφόρων μεθοδολογιών ανάπτυξης ενσωματωμένου λογισμικού αυτοδοκιμής για μικροεπεξεργαστές παρουσιάζεται στην [77].

Οι μεθοδολογίες ανάπτυξης λογισμικού αυτοδοκιμής μπορεί να διαχωριστούν σε δύο μεγάλες κατηγορίες ανάλογα με το αν αναπτύσσουν το λογισμικό ακολουθώντας μια προσέγγιση από πάνω προς τα κάτω (top-down) ή από κάτω προς τα πάνω (bottom-up). Οι μεθοδολογίες που ακολουθούν την ανάπτυξη του λογισμικού από πάνω προς τα κάτω, δε χρησιμοποιούν κάποιο δομικό μοντέλο ελαττωμάτων αλλά βασίζονται σε λειτουργικά ελαττώματα, πραγματοποιώντας λειτουργική δοκιμή. Επιπλέον, δε βασίζονται στην αρχιτεκτονική του μικροεπεξεργαστή αλλά αποκλειστικά και μόνο στο σύνολο εντολών αυτού. Το λογισμικό αυτοδοκιμής αποτελείται από ψευδοτυχαίες ακολουθίες εντολών οι οποίες εφαρμόζουν ψευδοτυχαία διανύσματα. Ο στόχος του λογισμικού είναι να διεγείρει όσο το δυνατό περισσότερα τμήματα του μικροεπεξεργαστή. Το μεγάλο πλεονέκτημα αυτής της προσέγγισης είναι πως μπορεί να εφαρμοστεί αρκετά εύκολα αφού το μόνο που απαιτείται είναι η γνώση του συνόλου εντολών

το οποίο είναι πάντα διαθέσιμο. Στον αντίποδα, το λογισμικό αυτοδοκιμής απαιτεί πολλούς κύκλους ρολογιού για να εκτελεστεί εξαιτίας των μεγάλων ψευδοτυχαίων ακολουθιών. Επιπλέον, δεν εξασφαλίζεται πάντα το υψηλό ποσοστό κάλυψης δομικών ελαττωμάτων. Μια ενδεχόμενη αύξηση στον αριθμό των εντολών και των διανυσμάτων δοκιμής δεν επιφέρει την επιθυμητή αύξηση του ποσοστού κάλυψης των ελαττωμάτων διότι εφαρμόζονται ψευδοτυχαία διανύσματα δοκιμής. Ενδεικτικές εργασίες της προσέγγισης ανάπτυξης λογισμικού αυτοδοκιμής από πάνω προς τα κάτω (top-down) είναι οι [78], [79], [80], [81].

Οι Shen και Abraham στην εργασία [78] προτείνουν την ανάπτυξη λογισμικού αυτοδοκιμής δίχως να βασίζονται σε κάποιο μοντέλο δομικών ελαττωμάτων. Το λογισμικό αυτοδοκιμής περιλαμβάνει την απαρίθμηση όλων των δυνατών συνδυασμών των εντολών που ενεργοποιούν τις υπομονάδες του επεξεργαστή ενώ τα διανύσματα δοκιμής προέρχονται από γεννιότερες ψευδοτυχαίων ακολουθιών. Εφάρμοσαν τη μεθοδολογία τους στο μικροεπεξεργαστή GL85¹³. Συνολικά, το λογισμικό αυτοδοκιμής που ανέπτυξαν περιελάμβανε 360.000 εντολές και το ποσοστό κάλυψης ελαττωμάτων που επιτύγχανε ήταν 90,2%, ενώ όταν οι αποκρίσεις των διανυσμάτων δοκιμής συμπτυσσόταν, το ποσοστό κάλυψης ελαττωμάτων μειωνόταν στο 86,7%.

Οι Batcher και Papachristou στην εργασία [79], προτείνουν την ανάπτυξη λογισμικού αυτοδοκιμής σε ψευδοτυχαίες ακολουθίες εντολών που συνδυάζονται με ψευδοτυχαία διανύσματα. Η παραγωγή των διανυσμάτων γίνεται με τη βοήθεια υλικού ώστε να είναι ταχύτερη και να μειωθεί ο συνολικός χρόνος εκτέλεσης του λογισμικού. Σύμφωνα με τις πειραματικές μετρήσεις τους, το επιπλέον υλικό δεν επηρεάζει την ταχύτητα του επεξεργαστή ενώ η επιβάρυνση σε υλικό είναι περίπου 3% για ένα επεξεργαστή τύπου RISC. Στα πειράματά τους, ανέπτυξαν δύο διαφορετικά λογισμικά αυτοδοκιμής, με το πρώτο να επιτυγχάνει ποσοστό κάλυψης 92,5% έπειτα από εκτέλεση 50.000 κύκλων ρολογιού και το δεύτερο να επιτυγχάνει ποσοστό κάλυψης 94,8% έπειτα από εκτέλεση 200.000 κύκλων.

Η εργασία [81] έχει ξεχωριστή αξία διότι πραγματεύεται την ανάπτυξη λογισμικού αυτοδοκιμής για τον επεξεργαστή Intel Pentium4 για την ανίχνευση ελαττωμάτων τα οποία διέφευγαν από τις παραδοσιακές μεθόδους δοκιμής, γεγονός που αποδεικνύει την αδυναμία των παραδοσιακών τεχνικών για αποτελεσματική δοκιμή. Το σύστημα FRITS που αναπτύχθηκε παράγει λογισμικό αυτοδοκιμής βασισμένο σε ψευδοτυχαίες ακολουθίες εντολών συνδυασμένες με ψευδοτυχαία διανύσματα δοκιμής. Το λογισμικό αυτοδοκιμής φορτώνεται στην κρυφή μνήμη από τον ελεγκτή και στη συνέχεια εκτελείται από τον επεξεργαστή. Ιδιαίτερη έμφαση δόθηκε ώστε κατά την εκτέλεση του λογισμικού αυτοδοκιμής να μη δημιουργούνται αναφορές σε θέσεις μνήμης που δεν υπάρχουν στην κρυφή μνήμη και να αποφεύγεται οποιαδήποτε ανταλλαγή δεδομένων με το σύστημα των διαύλων. Η παρατήρηση απέδειξε πως η ανάπτυξη λογισμικού δεν είναι μια απλή υπόθεση. Η μεθοδολογία FRITS εφαρμόστηκε στον επεξεργαστή Pentium4 και πέτυχε ποσοστό κάλυψης ελαττωμάτων 70%. Αντίστοιχη εφαρμογή της

¹³Ο μικροεπεξεργαστής GL85 υλοποιεί το σύνολο εντολών του επεξεργαστή Intel 8085 που αποτελείται από 6.300 πύλες και 244 στοιχεία μνήμης (flip-flop).

μεθοδολογίας στις αέριες μονάδες και στις μονάδες κινητής υποδιαστολής του επεξεργαστή Intel Itanium πέτυχε ποσοστό κάλυψης ελαττωμάτων 85%.

Στην εργασία [80], ο Corno και οι υπόλοιποι ερευνητές ανέπτυξαν το αυτοματοποιημένο σύστημα ανάπτυξης ενσωματωμένου λογισμικού MicroGP το οποίο βασίζεται στη θεωρία των γενετικών αλγορίθμων. Σε πρώτη φάση αναπτύχθηκε λογισμικό αυτοδοκιμής και έπειτα αξιολογήθηκε το ποσοστό κάλυψης που επιτυγχάνει με τη βοήθεια ενός εργαλείου εξομίωσης ελαττωμάτων. Στη συνέχεια το σύστημα MicroGP λάμβανε πληροφορίες από την προσομοίωση των ελαττωμάτων την οποία αξιοποιούσε με τους γενετικούς αλγορίθμους και εξέλιξε το αρχικό λογισμικό αυτοδοκιμής σε ένα νέο. Η διαδικασία επαναλαμβάνονταν μέχρι να επιτευχθεί το επιθυμητό ποσοστό κάλυψης. Στις πειραματικές μετρήσεις, το σύστημα MicroGP παράγαγε λογισμικό αυτοδοκιμής για τον μικροελεγκτή 8051 το οποίο πέτυχε ποσοστό κάλυψης ελαττωμάτων 90%.

Η δεύτερη μεγάλη κατηγορία περιλαμβάνει τις εργασίες εκείνες που βασίζουν την ανάπτυξη του λογισμικού αυτοδοκιμής σε κάποιο μοντέλο ελαττωμάτων. Το μοντέλο ελαττωμάτων μόνιμης τιμής μονοπωλεί τις περισσότερες εργασίες αν και λογισμικό αυτοδοκιμής έχει αναπτυχθεί και για μοντέλα ελαττωμάτων καθυστέρησης μονοπατιού (path delay faults) και για μοντέλα ελαττωμάτων γεφύρωσης (bridge faults). Οι εργασίες που αναπτύσσουν λογισμικό αυτοδοκιμής βάσει μοντέλων ελαττωμάτων ακολουθούν μια προσέγγιση από κάτω προς τα πάνω (bottom-up), ξεκινώντας από τα διανύσματα δοκιμής και καταλήγοντας στο λογισμικό αυτοδοκιμής.

Στην εργασία [82] οι ερευνητές παρουσιάζουν την ανωτερότητα της αυτοδοκιμής με ενσωματωμένο λογισμικό έναντι της δοκιμής με αλυσίδες σάρωσης και της αυτοδοκιμής βασισμένη στο υλικό. Στη συνέχεια δημιουργούν λογισμικό αυτοδοκιμής το οποίο εφαρμόζει διανύσματα δοκιμής τα οποία παράγονται είτε από γεννιότερες ψευδοτυχαίων διανυσμάτων είτε από αυτόματα εργαλεία εξαγωγής διανυσμάτων. Κατά τη διαδικασία της εξαγωγής διανυσμάτων λαμβάνονται υπόψη τυχόν περιορισμοί (constraints) όπως αυτοί ορίζονται από το σύνολο εντολών του μικροεπεξεργαστή και από τις μικρο-λειτουργίες που εκτελούν οι υπομονάδες του επεξεργαστή. Το λογισμικό αυτοδοκιμής που αναπτύσσουν αποθηκεύεται στη μνήμη από όπου και εκτελείται. Η εφαρμογή της μεθοδολογίας της εργασίας [82] στον επεξεργαστή Parwan 8-bit αρχιτεκτονικής συσσωρευτή (accumulator based) παράγαγε λογισμικό αυτοδοκιμής με ποσοστό κάλυψης ελαττωμάτων 91.4%, χρόνο εκτέλεσης 137.649 κύκλων και μέγεθος μνήμης 1129 byte. Στην εργασία [83], οι ερευνητές εξελίσσουν την μεθοδολογία ανάπτυξης λογισμικού αυτοδοκιμής εφαρμόζουν πολλαπλές προσομοιώσεις προτύπων λογισμικού (program templates) συνδυάζοντας διάφορες εντολές. Έπειτα χρησιμοποιούν στατιστική ανάλυση ώστε να αναγνωρίσουν τους περιορισμούς (constraints) που εισάγονται από το σύνολο εντολών ώστε να επιτύχουν την αποτελεσματική εξαγωγή διανυσμάτων δοκιμής. Η εφαρμογή της μεθοδολογίας στο στάδιο εκτέλεσης εντολών (execution stage) του επεξεργαστή Tensilica (Xtensa™) παράγαγε λογισμικό αυτοδοκιμής με ποσοστό κάλυψης ελαττωμάτων 95.2%, χρόνο εκτέλεσης

27.248 κύκλους και μέγεθος μνήμης 20373 bytes.

Η εργασία [84] προσεγγίζει την ανάπτυξη λογισμικού με παρόμοιο τρόπο όπως η εργασία [83]. Η διαφορά έγκειται στον τρόπο εξαγωγής και χρήσης των περιορισμών κατά την εξαγωγή των διανυσμάτων δοκιμής. Αναλυτικότερα, οι ερευνητές εξάγουν γνώση χρησιμοποιώντας τα δυαδικά διαγράμματα αποφάσεων (binary decision diagrams) για το τμήμα ελέγχου (control logic) και την παρεμβολή με πολυώνυμα (polynomial interpolation) για το μονοπάτι δεδομένων (datapath). Η εφαρμογή της μεθοδολογίας τους σε ένα τμήμα της μονάδας ελέγχου και στην αριθμητική και λογική μονάδα (ALU) του επεξεργαστή OpenRisc 1200, πέτυχε ποσοστά κάλυψης ελαττωμάτων 69.39% και 99.94%, αντίστοιχα.

Στις εργασίες [85], [86] οι ερευνητές αντίστοιχα με τις προηγούμενες ερευνητικές εργασίες κατασκευάζουν λογισμικό αυτοδοκιμής βασισμένο σε διανύσματα που εξάγονται από αυτόματα εργαλεία. Αντί όμως να προσπαθούν να περιγράψουν τους περιορισμούς που ορίζει το σύνολο εντολών του μικροεπεξεργαστή, εφαρμόζουν τυπικές μεθόδους επαλήθευσης σχεδίασης (formal verification methods) χρησιμοποιώντας την έννοια του αντιπαραδείγματος (counter example). Αρχικά, περιγράφεται με τυπική γλώσσα η ιδιότητα πως ένα διάνυσμα δοκιμής δεν είναι δυνατό να εφαρμοστεί σε μια υπομονάδα του επεξεργαστή. Την παραπάνω ιδιότητα προσπαθεί να την αποδείξει ένα εργαλείο επαλήθευσης. Αν το εργαλείο αποδείξει πως η ιδιότητα δεν ισχύει, θα γεννήσει ένα αντιπαραείγμα, το οποίο περιγράφει πώς θα εφαρμοστεί το διάνυσμα δοκιμής στην υπομονάδα.

Οι παραπάνω εργασίες που παρουσιάζονται στη διεθνή βιβλιογραφία ακολουθούν είτε την προσέγγιση από κάτω προς τα επάνω (bottom up) είτε την προσέγγιση από πάνω προς τα κάτω (top down) για την ανάπτυξη λογισμικού αυτοδοκιμής. Αντίθετα, οι εργασίες [87], [88], [89] βρίσκονται ενδιάμεσα των δύο κατηγοριών προσπαθώντας να συνδυάσουν τα πλεονεκτήματα των δύο τεχνικών ανάπτυξης λογισμικού αυτοδοκιμής. Οι ερευνητές βασίζονται στη γνώση που εξάγεται μελετώντας τη σχεδίαση του μικροεπεξεργαστή σε επίπεδο μεταφοράς καταχωρητή (register transfer level-RTL) ώστε να αναγνωρίσουν τους περιορισμούς που ορίζει το σύνολο εντολών του μικροεπεξεργαστή και έπειτα αναπτύσσουν διανύσματα δοκιμής τα οποία εξάγονται βάσει της συμμετρικής δομής των υπο-μονάδων. Τέτοιες μονάδες είναι οι αθροιστές, οι αφαιρέτες, οι συγκριτές, οι αυξητές, οι μειωτές, οι πολλαπλασιαστές αλλά και το αρχείο καταχωρητών του μικροεπεξεργαστή (register file). Οι εν λόγω μονάδες αποτελούν ένα μεγάλο ποσοστό του μικροεπεξεργαστή. Καθώς οι επεξεργαστές μεταβαίνουν από αρχιτεκτονικές 32-bit σε αρχιτεκτονικές 64-bit το ποσοστό αυτό μεγαλώνει. Στην εργασία [87], οι ερευνητές εφαρμόσαν τη μεθοδολογία τους στον μικροεπεξεργαστή Parwan 8-bit και παράγαν λογισμικό αυτοδοκιμής με ποσοστό κάλυψης ελαττωμάτων 91%, χρόνο εκτέλεσης 16667 κύκλους και μέγεθος μνήμης 923 byte. Σε σχέση με τα πειραματικά αποτελέσματα της εργασίας [82], διαπιστώνεται η αποτελεσματικότητα της μεθοδολογίας αυτής αφού με πολύ μικρότερο και ταχύτερο στην εκτέλεση λογισμικό αυτοδοκιμής επιτυγχάνεται εξίσου υψηλό ποσοστό κάλυψης ελαττωμάτων. Στις εργασίες [88], [89], οι ερευνητές εξελίσσουν τη μεθοδολογία τους

και την εφαρμόζουν για πρώτη φορά σε ένα μικροεπεξεργαστή MIPS με τρία στάδια διοχέτευσης, και σε ένα μικροεπεξεργαστή με σύνολο εντολών ειδικού σκοπού (Application Specific Instruction Set-ASIP) υλοποιημένο σε αρχιτεκτονική MIPS R3000.

Στην εργασίες [90] και [91] οι ερευνητές εξελίσσουν τη μεθοδολογία των εργασιών [88] και [89] και την προσαρμόζουν για την αυτοδοκιμή των μικροεπεξεργαστών που διαθέτουν πολύπλοκους μηχανισμούς διοχέτευσης με προώθηση (forwarding). Η εφαρμογή της μεθοδολογίας στους μηχανισμούς διοχέτευσης δύο σύγχρονων μικροεπεξεργαστών με 5 στάδια διοχέτευσης, miniMips και OpenRisc 1200 αυξάνει το ποσοστό κάλυψης ελαττωμάτων κατά 22% στο μηχανισμό διοχέτευσης και συνολικά κατά 12% στον μικροεπεξεργαστή.

Καμία από τις παραπάνω εργασίες που μελετούν την αυτοδοκιμή των μικροεπεξεργαστών με λογισμικό δεν αναφέρεται στις μονάδες κινητής υποδιαστολής και δεν τις μελετά συστηματικά. Καθώς όμως οι μικροεπεξεργαστές καλούνται να καλύψουν τις ανάγκες των εφαρμογών επεξεργασίας φωνής και εικόνας (πολυμέσα) η ενσωμάτωση των μονάδων κινητής υποδιαστολής είναι επιβεβλημένη. Το κενό αυτό στη διεθνή βιβλιογραφία, δηλαδή την αυτοδοκιμή των μονάδων κινητής υποδιαστολής με ενσωματωμένο λογισμικό, έρχεται να καλύψει η παρούσα διδακτορική διατριβή.

2.9 Ανακεφαλαίωση

Στο Κεφάλαιο 2,

- έγινε η εισαγωγή στην έννοια της δοκιμής, στους τρόπους πραγματοποίησής της και στις δυσκολίες της,
- παρουσιάστηκαν τα μοντέλα ελαττωμάτων,
- παρουσιάστηκαν οι τεχνικές σχεδίασης για αυξημένη δοκιμαστικότητα, και
- έγινε η εισαγωγή στην αυτοδοκιμή και στην αυτοδοκιμή μικροεπεξεργαστών με λογισμικό.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Κεφάλαιο 3

Αριθμητική Κινητής Υποδιαστολής

Οι αριθμοί κινητής υποδιαστολής (*floating point numbers*) αποτελούν τη γέφυρα επικοινωνίας του λογισμικού με τον κόσμο των πραγματικών αριθμών. Ολοένα και περισσότερα συστήματα χρησιμοποιούν τους αριθμούς κινητής υποδιαστολής για την αναπαράσταση των πραγματικών αριθμών. Χαρακτηριστικές περιπτώσεις αποτελούν τα συστήματα επεξεργασίας γραφικών και οι κονσόλες παιχνιδιών, όπου η ανάγκη για υψηλής ποιότητας αναπαράστασης της κίνησης αλλά και των ίδιων των χαρακτήρων σημαίνει υποχρεωτική χρήση αριθμών κινητής υποδιαστολής μεγάλης ακρίβειας. Τα συστήματα σύνθεσης και επεξεργασίας φωνής τα οποία ανέκαθεν χρησιμοποιούσαν αριθμούς σταθερής υποδιαστολής (*fixed point numbers*) έχουν αρχίσει να χρησιμοποιούν και αυτά όλο και περισσότερο αριθμούς κινητής υποδιαστολής. Επίσης, με την άνθιση της ψηφιακής φωτογραφίας, οι αριθμοί κινητής υποδιαστολής χρησιμοποιούνται για να προσεγγίσουν το φάσμα του αναλογικού φιλμ. Γενικά, οι αριθμοί κινητής υποδιαστολής χρησιμοποιούνται από οποιαδήποτε λογισμικό απαιτείται να επεξεργαστεί πραγματικούς αριθμούς.

Οι ανάγκες για υπολογισμό με πραγματικούς αριθμούς έχουν οδηγήσει πολλές εταιρείες στην ενσωμάτωση περισσότερων της μιας αριθμητικών μονάδων κινητής υποδιαστολής σε έναν επεξεργαστή, όπως στην περίπτωση του PowerPC970-FX [23] της IBM (ενσωματώνει 2 μονάδες κινητής υποδιαστολής). Επίσης, ο πολυεπεξεργαστής Niagara 2 [24] ενσωματώνει μια μονάδα κινητής υποδιαστολής σε καθένα από τους 8 επεξεργαστές που διαθέτει.

Τα προβλήματα που μπορεί να ανακύψουν από τη λανθασμένη χρήση των αριθμών κινητής υποδιαστολής έγιναν γνωστά με το διάσημο πλέον σφάλμα στη μονάδα διαίρεσης αριθμών κινητής υποδιαστολής του επεξεργαστή Pentium της Intel το 1994 [6]. Έπειτα από έρευνες, διαπιστώθηκε πως ορισμένα τμήματα του πίνακα αναζήτησης (*lookup table*), που χρησιμο-

ποιείται από τον αλγόριθμο διαίρεσης SRT-4, στα οποία έπρεπε να είναι αποθηκευμένη η τιμή 2 ήταν κενά. Έτσι, κατά την προσπέλαση του πίνακα αναζήτησης αντί ο αλγόριθμος διαίρεσης να διαβάσει την τιμή 2 διάβαζε την τιμή 0 [92]. Το σφάλμα στην εν λόγω μονάδα υπολογίζεται πως κόστισε στην Intel περίπου 500 εκατομμύρια δολάρια. Ευτυχώς στην περίπτωση αυτή το κόστος ήταν μόνο οικονομικό. Αντίθετα, στην περίπτωση του πυραυλικού συστήματος Patriot μια σειρά από λανθασμένες στρογγυλοποιήσεις οδήγησαν στην απώλεια ζωών. Το χρονόμετρο (timer) του πυραύλου μετρούσε σε δέκατα του δευτερολέπτου. Όμως, ο αριθμός 0,1¹ δεν αναπαρίσταται ακριβώς στο δυαδικό σύστημα άρα ούτε από τους αριθμούς κινητής υποδιαστολής. Διαδοχικές στρογγυλοποιήσεις στο χρονόμετρο, οδήγησαν στην απόκλιση κατά 0,34 δευτερόλεπτα τα οποία ήταν αρκετά ώστε ο Patriot να αποτύχει να αναχαιτίσει τον πύραυλο Scud (απέκλινε κατά 500 μέτρα περίπου) ο οποίος έπληξε ένα στρατόνα σκοτώνοντας 28 στρατιώτες [19].

Μια άλλη λανθασμένη χρήση των αριθμών κινητής υποδιαστολής αναφέρεται στη μη επανδρωμένη διαστημική αποστολή Ariane 5 όπου ο πύραυλος εξερράγη εξαιτίας της υπερχείλισης μιας μεταβλητής κινητής υποδιαστολής στην οποία ήταν αποθηκευμένη η επιτάχυνση του πυραύλου. Εξαιτίας της υπερχείλισης, το κυρίως αλλά και το εφεδρικό σύστημα πλοήγησης κατέρρευσε με αποτέλεσμα την έκρηξη του πυραύλου και την αποτυχία της αποστολής. Αξιοσημείωτο είναι πως το ίδιο λογισμικό είχε χρησιμοποιηθεί στην παλαιότερη αποστολή Ariane 4 χωρίς κανένα πρόβλημα αφού η μεταβλητή δεν υπερχείλιζε [18].

Η ανάλυση των αριθμών κινητής υποδιαστολής στο κεφάλαιο αυτό και των πράξεων μεταξύ τους βασίζεται στις αναφορές [21], [22], [93].

3.1 Βασικές Έννοιες

Οι αριθμοί κινητής υποδιαστολής στη γενική τους μορφή αποτελούνται από τρία μέρη: τον εκθέτη (*exponent*), τη βάση του εκθέτη (*base*) και το σημανόμενο (*significand part ή mantissa*). Το σημανόμενο αποτελείται από το ακέραιο μέρος (*integer part*) και το κλασματικό μέρος (*fractional part*), αντίστοιχα. Ο πραγματικός αριθμός, x , που αναπαριστά ένας αριθμός κινητής υποδιαστολής είναι ίσος με:

$$x = M_x \times b^{E_x},$$

όπου M_x είναι το σημανόμενο, E_x ο εκθέτης και b η βάση του εκθέτη.

Για την αναπαράσταση των αρνητικών αριθμών χρησιμοποιείται η αναπαράσταση πρόσημο και μέγεθος (*sign and magnitude*). Αν και παλαιότερα χρησιμοποιήθηκε η αναπαράσταση με συμπλήρωμα ως προς 2 η αναπαράσταση πρόσημο και μέγεθος επικράτησε στο πρότυπο IEEE 754 [94] καθώς θεωρήθηκε πιο φυσιολογική για τους αριθμούς κινητής υποδιαστολής.

¹Η αναπαράσταση του δεκαδικού 0,1 στο δυαδικό σύστημα είναι 0,000110011... Εάν είναι διαθέσιμα 23 ψηφία για το κλασματικό μέρος, τότε δεκαδικός αριθμός που προσεγγίζει τον αριθμό 0,1 είναι ο αριθμός 0,0999999046325684.

Έτσι, ένας αριθμός κινητής υποδιαστολής, x , εκφράζεται από το πρόσημο, τον εκθέτη και το σημαίνόμενο

$$x = \pm M_x \times b^{E_x}.$$

Η αναπαράσταση των πραγματικών αριθμών με αριθμούς κινητής υποδιαστολής οδηγεί σε πλεονασμό καθώς ένας πραγματικός αριθμός είναι δυνατό να αναπαρασταθεί με περισσότερους από έναν αριθμούς κινητής υποδιαστολής, χρησιμοποιώντας διάφορους συνδυασμούς για τον εκθέτη και το σημαίνόμενο. Για παράδειγμα, ο αριθμός 0,5 μπορεί να αναπαρασταθεί είτε ως 5×10^{-1} είτε ως $0,5 \times 10^0$. Είναι ευνόητο πως η αναπαράσταση των αριθμών με περισσότερους από έναν τρόπους δημιουργεί προβλήματα. Γι' αυτό το λόγο, η αναπαράσταση του σημαίνόμενου απαιτείται να είναι σε κανονικοποιημένη μορφή (*normalized format*), όπου το ψηφίο του ακέραιου μέρους θα πρέπει να είναι διαφορετικό του μηδενός. Γενικότερα, αν χρησιμοποιούνται m ψηφία για την αναπαράσταση των αριθμών κινητής υποδιαστολής και από αυτά f είναι διαθέσιμα για το κλασματικό μέρος, τότε η τιμή που μπορεί να πάρει το σημαίνόμενο είναι ίση με :

$$M = \sum_{i=-f}^{m-f-1} d_i \times r^i, \quad (3.1)$$

και το εύρος των πραγματικών αριθμών που μπορεί να καλύψει είναι ίσο με :

$$r^{-f} \leq M \leq r^{m-f} - r^{-f}, \quad (3.2)$$

όπου r είναι η βάση (radix) του συστήματος που χρησιμοποιείται για το σημαίνόμενο (για το δυαδικό σύστημα $r = 2$).

Η αναπαράσταση του ακέραιου μέρους με περισσότερα του ενός ψηφία γρήγορα απορρίφθηκε για λόγους σχετικούς με την εξοικονόμηση του υλικού, με αποτέλεσμα μόνο ένα ψηφίο να διατίθεται για την αναπαράσταση του ακέραιου μέρους. Διατηρώντας την ανάγκη για κανονικοποιημένους αριθμούς, το ψηφίο αριστερά της υποδιαστολής θα πρέπει να είναι διαφορετικό του μηδενός. Στην περίπτωση αυτή, το εύρος των τιμών (μη συμπεριλαμβανομένης της τιμής 0) που μπορεί να εκφράσει το σημαίνόμενο είναι ίσο με :

$$1 + r^{-f} \leq M \leq r - r^{-f}. \quad (3.3)$$

Χρησιμοποιώντας ως βάση το δυαδικό σύστημα ($r = 2$), το σημαίνόμενο μπορεί να λάβει τιμές στο διάστημα $[1, 2)$. Όταν οι αριθμοί κινητής υποδιαστολής έχουν ένα ψηφίο διαθέσιμο για το ακέραιο μέρος το οποίο είναι μη-μηδενικό (δηλαδή είναι κανονικοποιημένοι) λέγεται πως αναπαρίστανται σε επιστημονική σημειογραφία (*scientific notation*).

Συνολικά, ένας αριθμός κινητής υποδιαστολής x σε βάση b και σε κανονικοποιημένη μορφή εκφράζεται από το πρόσημο (S), τον εκθέτη (E) και το κλασματικό μέρος (F) και παριστάνει τον πραγματικό αριθμό :

$$x = (-1)^S \times b^E \times 1.F. \quad (3.4)$$

Πριν υιοθετηθεί η κανονικοποιημένη αναπαράσταση για το σημαίνόμενο, είχε προταθεί η ιδέα πως το ψηφίο αριστερά της υποδιαστολής θα έπρεπε να είναι ίσο με μηδέν. Στην περίπτωση αυτή το σημαίνόμενο αναπαριστά το εύρος $r^{-f} \leq M \leq 1 - r^{-f}$ (μη συμπεριλαμβανομένης της τιμής 0) και στο δυαδικό σύστημα μπορεί να αναπαραστήσει πραγματικούς αριθμούς στο πεδίο $[0, 1)$. Το σημαίνόμενο αυτό είναι γνωστό και ως *mantissa*. Καθώς, έχει επικρατήσει η κανονικοποιημένη μορφή οι όροι *significand* και *mantissa* χρησιμοποιούνται πια ισοδύναμα.

Η υιοθέτηση της κανονικοποιημένης αναπαράστασης του σημαίνόμενου έχει το μειονέκτημα ότι μειώνει το εύρος των πραγματικών αριθμών που μπορεί να προσεγγίσει ένας αριθμός κινητής υποδιαστολής. Για την ακρίβεια ο μικρότερος αριθμός που μπορεί να αναπαρασταθεί σε κανονικοποιημένη μορφή χρησιμοποιώντας m ψηφία για το σημαίνόμενο από τα οποία f διατίθενται για το κλασματικό μέρος, είναι ίσος με :

$$r^{m-f-1} \times b^{E_{min}}. \quad (3.5)$$

Στην περίπτωση όπου ένα μόνο ψηφίο είναι διαθέσιμο για το ακέραιο μέρος, δηλαδή $m = f + 1$, ο μικρότερος κανονικοποιημένος αριθμός κινητής υποδιαστολής είναι ίσος με :

$$r^0 \times b^{E_{min}} = b^{E_{min}}, \quad (3.6)$$

όπου r είναι η βάση του συστήματος αναπαράστασης για το σημαίνόμενο και b η βάση του εκθέτη.

Εξαιτίας αυτού του μειονεκτήματος της κανονικοποιημένης αναπαράστασης, επιτρέπεται η χρήση μη κανονικοποιημένων αριθμών μόνο όταν αυτοί έχουν τον μικρότερο δυνατό εκθέτη. Οι αριθμοί αυτοί είναι γνωστοί με το όνομα μη κανονικοποιημένοι αριθμοί (*denormals*) ή αλλιώς υποκανονικοί (*subnormals*). Με αυτό τον τρόπο αυξάνεται το εύρος των πραγματικών αριθμών που προσεγγίζονται με αριθμούς κινητής υποδιαστολής και ο μικρότερος μη κανονικοποιημένος αριθμός κινητής υποδιαστολής είναι ίσος με :

$$r^{-f} \times b^{E_{min}}.$$

Το φαινόμενο της σταδιακής μείωσης των αριθμών κινητής υποδιαστολής καθώς εμφανίζονται όλο και περισσότερα μηδενικά στο κλασματικό μέρος ονομάζεται *βαθμιαία ανεπάρκεια (gradual underflow)*.

Η επιλογή της βάσης του εκθέτη ενός αριθμού κινητής υποδιαστολής αποτελεί μια δύσκολη απόφαση καθώς μεγαλύτερες τιμές της βάσης επιτρέπουν την αναπαράσταση ιδιαίτερως μεγάλων αριθμών. Τελικά, επικράτησε η άποψη πως η βάση του εκθέτη θα πρέπει να είναι ίση με 2, αν και στο παρελθόν κατασκευάστηκαν μονάδες κινητής υποδιαστολής όπου η βάση του εκθέτη ήταν ίση με 4, 8 ακόμα και 256. Η κύρια αιτία για την επιλογή του αριθμού 2 ως βάση του εκθέτη έχει να κάνει με λόγους απλοποίησης του υλικού των μονάδων κινητής υποδιαστολής. Ένα άλλο σημαντικό θέμα σχετικά με τον εκθέτη είναι η αναπαράσταση

των αρνητικών τιμών, όπου στην αρχή χρησιμοποιήθηκε η αναπαράσταση σε συμπλήρωμα ως προς 2. Εν τέλει επικράτησε η άποψη πως ο εκθέτης των αριθμών κινητής υποδιαστολής θα εκφράζεται σε *πολωμένη σημειογραφία (biased notation)*. Στην πολωμένη σημειογραφία οι ακέραιοι (θετικοί και αρνητικοί) αναπαρίστανται ως το άθροισμα της τιμής τους με ένα θετικό αριθμό, γνωστό ως *πόλωση (bias)*. Με αυτό τον τρόπο οι αρνητικοί ακέραιοι μετατρέπονται σε θετικοί. Καθώς ο εκθέτης βρίσκεται σε πολωμένη μορφή, δεν υπάρχει ανάγκη για διαχωρισμό μεταξύ θετικών και αρνητικών εκθετών κατά την επεξεργασία των αριθμών. Έτσι, ο αρνητικός εκθέτης μετατρέπεται σε ένα θετικό ως εξής:

$$E_{biased} = E + B,$$

όπου B είναι η πόλωση. Η πόλωση θα πρέπει να είναι ίση με το μικρότερο αρνητικό ακέραιο ώστε :

$$E_{min} + B = 0 \Rightarrow B = -E_{min}. \quad (3.7)$$

Επιπλέον για να υπάρχει συμμετρία στο εύρος των εκθετών θα πρέπει $-B \leq E \leq B$ ώστε μετά την αναπαράσταση σε πολωμένη μορφή να ισχύει $0 \leq E_{biased} \leq 2B$. Αν e ψηφία είναι διαθέσιμα για την αναπαράσταση του εκθέτη ενός αριθμού κινητής υποδιαστολής, θα πρέπει να ισχύει $2B \leq 2^e - 1$. Άρα :

$$B \leq \frac{(2^e - 1)}{2} \leq \frac{(2^e - 2)}{2} = 2^{e-1} - 1. \quad (3.8)$$

Για παράδειγμα, αν είναι διαθέσιμα 8 ψηφία για την αναπαράσταση του εκθέτη τότε για την πόλωση θα πρέπει να ισχύει $B = 127$. Έτσι, αν ο εκθέτης έχει την τιμή -2 τότε η πολωμένη αναπαράστασή του είναι $-2 + 127 = 125$, ενώ αν η τιμή του είναι 3 η πολωμένη αναπαράσταση είναι $3 + 127 = 130$.

Η πολωμένη σημειογραφία αυτή καθαυτή δεν επιβραδύνει σημαντικά τις πράξεις μεταξύ των αριθμών κινητής υποδιαστολής αλλά παρέχει δύο πολύ σημαντικά πλεονεκτήματα. Κατά πρώτον, η σύγκριση δύο αριθμών κινητής υποδιαστολής γίνεται ευκολότερα συγκρίνοντας σε πρώτη φάση τους εκθέτες (σε πολωμένη μορφή) και μόνο αν αυτοί είναι ίσοι συγκρίνονται τα κλασματικά μέρη. Με αυτό τον τρόπο επιταχύνεται η ταξινόμηση των αριθμών κινητής υποδιαστολής. Κατά δεύτερον, είναι δυνατή η αναπαράσταση του πραγματικού αριθμού μηδέν επιλέγοντας τη μικρότερη τιμή που αναπαριστά ο εκθέτης, δηλαδή την τιμή μηδέν.

Το εύρος των πραγματικών αριθμών που μπορούν να αναπαραστήσουν οι αριθμοί κινητής υποδιαστολής είναι οι αρνητικοί $[-max, -min]$ και οι θετικοί $[min, max]$, όπου :

$$max = M_{max} \times b^{E_{max}} \quad (3.9)$$

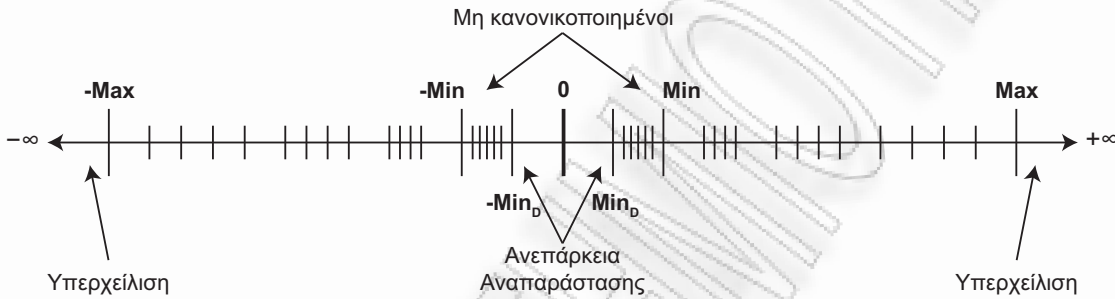
$$min = M_{min} \times b^{E_{min}}. \quad (3.10)$$

Φυσικά το ελάχιστο (μέγιστο) σημανόμενο αλλά και ο ελάχιστος (μέγιστος) εκθέτης εξαρτώνται από το πλήθος των ψηφίων τα οποία χρησιμοποιούνται για την αναπαράσταση του

Μέγιστος Αρνητικός $-max$	$-(r^{m-f} - r^{-f}) \times b^{E_{max}}$
Ελάχιστος Αρνητικός $-min$	$-r^{m-f-1} \times b^{E_{min}}$
Ελάχιστος Υποκανονικός Αρνητικός $-min_D$	$-r^{-f} \times b^{E_{min}}$
Ελάχιστος Θετικός $+min$	$r^{m-f-1} \times b^{E_{min}}$
Ελάχιστος Θετικός Υποκανονικός $+min_D$	$r^{-f} \times b^{E_{min}}$
Μέγιστος Θετικός $+max$	$(r^{m-f} - r^{-f}) \times b^{E_{max}}$

Πίνακας 3.1: Μέγιστοι και ελάχιστοι αριθμοί κινητής υποδιαστολής.

αριθμού κινητής υποδιαστολής. Η Εικόνα 3.1 παρουσιάζει την κατανομή των αριθμών κινητής υποδιαστολής στον άξονα των πραγματικών αριθμών.



Εικόνα 3.1: Το εύρος των αριθμών κινητής υποδιαστολής.

Στο κέντρο του άξονα των πραγματικών αριθμών βρίσκεται η τιμή 0, η οποία και αναπαρίσταται με ειδική μορφή θέτοντας προκαθορισμένες τιμές στα πεδία του εκθέτη και του σημαινόμενου. Ο άξονας των θετικών αριθμών περιλαμβάνει δύο κατηγορίες τιμών. Η πρώτη κατηγορία περιλαμβάνει τους κανονικοποιημένους αριθμούς, αυτούς που βρίσκονται στο διάστημα $[min, max]$ όπως φαίνεται στην Εικόνα 3.1, ενώ η δεύτερη κατηγορία περιλαμβάνει τους μη-κανονικοποιημένους αριθμούς των οποίων ο εκθέτης είναι ο μικρότερος δυνατός μη-μηδενικός. Το διάστημα αυτό είναι το $[min_D, min]$. Αντίστοιχα είναι και τα διαστήματα στην αρνητική πλευρά του άξονα των πραγματικών αριθμών. Ο Πίνακας 3.1 παρουσιάζει τις τιμές των σημείων του άξονα των πραγματικών αριθμών θεωρώντας πως m ψηφία είναι διαθέσιμα για την αναπαράσταση του σημαινόμενου και f ψηφία χρησιμοποιούνται για το κλασματικό μέρος.

Όπως φαίνεται στην Εικόνα 3.1 η κατανομή των αριθμών κινητής υποδιαστολής δεν είναι συμμετρική ως προς τον άξονα των πραγματικών αριθμών. Η αναπαράσταση των πραγματικών αριθμών είναι πιο πυκνή καθώς οι αριθμοί κινητής υποδιαστολής πλησιάζουν προς το 0, ενώ η αναπαράσταση είναι περισσότερο αραιή καθώς οι αριθμοί πλησιάζουν το $\pm\infty$. Η διαφορά, Δ , δύο διαδοχικών αριθμών κινητής υποδιαστολής, οι οποίοι έχουν τον ίδιο εκθέτη και την

φυσικά την ίδια βάση είναι ίση με :

$$\Delta = r^{-f} \times r^E = r^{E-f}. \quad (3.11)$$

Όσο περισσότερα ψηφία είναι διαθέσιμα για το κλασματικό μέρος του αριθμού κινητής υποδιαστολής τόσο μεγαλύτερη είναι και η ακρίβεια του, δηλαδή τόσο πιο «πυκνή» θα είναι η αναπαράσταση. Από την άλλη πλευρά όσο περισσότερα ψηφία είναι διαθέσιμα για τον εκθέτη τόσο περισσότερο μεγαλώνει το εύρος των πραγματικών αριθμών που είναι δυνατό να αναπαρασταθούν. Καθώς μεγαλώνει το εύρος αναπαράστασης μειώνεται η ακρίβεια με την οποία προσεγγίζονται οι πραγματικοί αριθμοί, όπως διακρίνεται από την εξίσωση της διαφοράς (3.11). (δηλαδή τόσο πιο "αραιή" γίνεται η αναπαράσταση.)

Πριν υιοθετηθεί το πρότυπο IEEE 754 για την αναπαράσταση των αριθμών κινητής υποδιαστολής, κάθε κατασκευαστής επέλεγε το πλήθος των διαθέσιμων ψηφίων για το σημεινόμενο, τον εκθέτη, τη βάση του εκθέτη, και τον τρόπο αναπαράστασης των αρνητικών σημεινόμενων και εκθετών ανάλογα με τις απαιτήσεις της εκάστοτε εφαρμογής. Ακόμα και σήμερα κατασκευάζονται μονάδες κινητής υποδιαστολής με διαφορετικά χαρακτηριστικά από αυτά του προτύπου IEEE 754 (32 ψηφία για απλή ακρίβεια και 64 ψηφία για διπλή ακρίβεια) εξαιτίας των ειδικών απαιτήσεων που μπορεί να έχει μια εφαρμογή και οι οποίες δεν καλύπτονται από το πρότυπο. Χαρακτηριστικό παράδειγμα αποτελεί η εταιρία κατασκευής υποσυστημάτων γραφικών, NVIDIA, η οποία χρησιμοποιεί αριθμούς κινητής υποδιαστολής με 16 ψηφία αναπαράστασης (μισή ακρίβεια) [95]. Στον αντίποδα, εφαρμογές όπως η προσομοίωση αντοχής των υλικών σε μεγάλες πιέσεις απαιτεί τη χρήση αριθμών κινητής υποδιαστολής με 128 ψηφία ακρίβειας (τετραπλή ακρίβεια). Σήμερα οι παραπάνω ακρίβειες έχουν ενσωματωθεί στο αναθεωρημένο πρότυπο IEEE 754 [96].

Οι αριθμοί κινητής υποδιαστολής, όπως άλλωστε και οποιοδήποτε σύστημα αναπαράστασης αριθμών, δεν είναι δυνατό να αναπαραστήσουν όλους τους πραγματικούς αριθμούς. Όταν το αποτέλεσμα μιας πράξης κινητής υποδιαστολής είναι μεγαλύτερο από το μεγαλύτερο αριθμό κινητής υποδιαστολής τότε λέγεται πως έχει συμβεί *υπερχείλιση (overflow)*. Η υπερχείλιση διαπιστώνεται ελέγχοντας τον εκθέτη του αποτελέσματος. Αν ο εκθέτης είναι μεγαλύτερος από το μεγαλύτερο δυνατό εκθέτη τότε έχει συμβεί υπερχείλιση. Αντίστοιχα, όταν το αποτέλεσμα είναι μικρότερο και από το μικρότερο δυνατό αριθμό τότε γίνεται λόγος για *ανεπάρκεια αναπαράστασης κινητής υποδιαστολής (underflow)*, όπως φαίνεται και στην Εικόνα 3.1. Η ανεπάρκεια αναπαράστασης διαπιστώνεται ελέγχοντας τον εκθέτη. Όταν ο εκθέτης είναι μικρότερος από το μικρότερο δυνατό εκθέτη τότε έχει λάβει χώρα το φαινόμενο της ανεπάρκειας αναπαράστασης.

Οι αριθμοί κινητής υποδιαστολής εκτός από τους πραγματικούς αριθμούς καλούνται να αναπαραστήσουν ορισμένες ειδικές τιμές οι οποίες μπορεί να προκύψουν από τις πράξεις κινητής υποδιαστολής. Τέτοιες τιμές είναι το μηδέν (0) και το $\pm\infty$. Επιπλέον όταν το αποτέλεσμα της πράξης κινητής υποδιαστολής δεν ορίζεται, όπως στην αφαίρεση $\infty - \infty$, χρησιμοποιείται

μια ειδική τιμή για να δείξει πως η πράξη δεν είναι έγκυρη. Οι ειδικές τιμές των αριθμών κινητής υποδιαστολής καθορίζονται με προαποφασισμένες τιμές στα πεδία του εκθέτη και του κλασματικού μέρους του σημαινόμενου. Εφόσον δεσμεύονται κάποιες τιμές για τα πεδία, η αναπαράσταση κάποιων αριθμών κινητής υποδιαστολής δεν επιτρέπεται με αυτές τις τιμές.

3.1.1 Ακρίβεια και Απόκλιση

Οι αριθμοί κινητής υποδιαστολής δεν μπορούν να προσεγγίσουν πάντα με απόλυτη ακρίβεια την τιμή ενός πραγματικού αριθμού, εξαιτίας των πεπερασμένων ψηφίων που διαθέτουν για την αναπαράσταση του κλασματικού μέρους και του εκθέτη. Επίσης, η αδυναμία απόλυτης ακρίβειας εξαρτάται και από τη βάση του αριθμητικού συστήματος που χρησιμοποιείται. Έτσι για παράδειγμα η τιμή 1×10^{-1} δεν μπορεί να αναπαρασταθεί με ακρίβεια στο δυαδικό σύστημα όσα ψηφία και αν είναι διαθέσιμα για το κλασματικό μέρος, σε αντίθεση με το δεκαδικό σύστημα. Το σφάλμα κατά την αναπαράσταση ενός πραγματικού αριθμού μετριέται συνήθως σε μονάδες στην τελευταία θέση (*units in last place - ulp*). Για παράδειγμα, όταν γίνεται αναφορά στο σύνολο των ακεραίων, τότε το τελευταίο ψηφίο είναι ίσο με τη μονάδα. Το τελευταίο ψηφίο, όταν γίνεται αναφορά σε αριθμούς κινητής υποδιαστολής είναι ίσο με τη διαφορά δύο διαδοχικών τιμών του σημαινόμενου, δηλαδή :

$$ulp_{\text{κινητής υποδιαστολής}} = r^{-f}.$$

Γενικά, αν ο αριθμός κινητής υποδιαστολής $d.dd\dots d \times b^E$ χρησιμοποιείται για να προσεγγίσει τον πραγματικό αριθμό z , τότε το σφάλμα κατά την αναπαράσταση είναι ίσο με :

$$|d.dd\dots d - (z/b^E)| \times b^{m-1} \text{ ulp.} \quad (3.12)$$

όπου m είναι τα ψηφία που έχουν δεσμευτεί για την αναπαράσταση του σημαινόμενου. Έτσι αν ο αριθμός κινητής υποδιαστολής 3.12×10^{-2} χρησιμοποιείται για να προσεγγίσει τον πραγματικό αριθμό 0.0314 και είναι διαθέσιμα 3 ψηφία για την αναπαράσταση του σημαινόμενου, τότε το σφάλμα προσέγγισης είναι ίσο με :

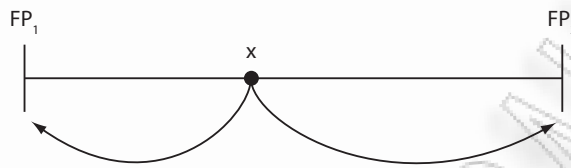
$$|3.12 - 0.0314/10^{-2}| \times 10^{3-1} = |-0.02| \times 10^2 = 2 \text{ ulp.} \quad (3.13)$$

Αντίστοιχα, εάν ο αριθμός κινητής υποδιαστολής 3.14×10^{-2} χρησιμοποιείται για να αναπαραστήσει τον πραγματικό αριθμό 0.0314159, όπου 3 ψηφία είναι δεσμευμένα για το σημαινόμενο, τότε το σφάλμα προσέγγισης είναι ίσο με :

$$|3.14 - 0.0314159/10^{-2}| \times 10^{3-1} = 0.159 \text{ ulp.} \quad (3.14)$$

3.2 Στρογγυλοποίηση

Πολλές φορές η αναπαράσταση του αποτελέσματος μιας πράξης κινητής υποδιαστολής χρειάζεται περισσότερα ψηφία από αυτά που είναι διαθέσιμα σε έναν καταχωρητή. Στην



Εικόνα 3.2: Η στρογγυλοποίηση του αριθμού x .

περίπτωση αυτή το αποτέλεσμα χρειάζεται *στρογγυλοποίηση (rounding)* ώστε όχι μόνο να απορριφθούν τα επιπλέον ψηφία, αλλά ταυτόχρονα ο νέος αριθμός που θα προκύψει θα πρέπει να προσεγγίζει όσο το δυνατόν καλύτερα τον αριθμό πριν την στρογγυλοποίηση. Έστω οι πραγματικοί αριθμοί x, y και οι αριθμοί κινητής υποδιαστολής $R(x), R(y)$ που αναπαριστούν τους x, y μετά τη στρογγυλοποίηση. Ένας *τρόπος στρογγυλοποίησης (rounding mode)* θα πρέπει να ικανοποιεί τους παρακάτω κανόνες:

- **Διάταξη (Ordering):**
Αν ισχύει $x \leq y$, τότε θα πρέπει να ισχύει $R(x) \leq R(y)$
- **Αντιπροσώπευση (Representability):** Αν ο πραγματικός αριθμός x προσεγγίζεται με απόλυτη ακρίβεια από ένα αριθμό κινητής υποδιαστολής, τότε θα πρέπει να ισχύει $R(x) = x$, δηλαδή ο στρογγυλοποιημένος αριθμός κινητής υποδιαστολής να συνεχίζει να προσεγγίζει με απόλυτη ακρίβεια τον πραγματικό x .
- **Περιεκτικότητα (Containment):** Έστω δύο διαδοχικοί αριθμοί κινητής υποδιαστολής FP_1 και FP_2 τέτοιοι ώστε $FP_1 \leq x \leq FP_2$, τότε ο στρογγυλοποιημένος αριθμός $R(x)$ θα πρέπει να είναι είτε ο FP_1 είτε ο FP_2 , όπως φαίνεται και στην Εικόνα 3.2.

Οι μέθοδοι στρογγυλοποίησης χαρακτηρίζονται από το σφάλμα αναπαράστασης που εισάγουν κατά τη διαδικασία της στρογγυλοποίησης το οποίο είναι γνωστό ως *σφάλμα στρογγυλοποίησης (round-off error)*. Το σφάλμα στρογγυλοποίησης ορίζεται ως η διαφορά του στρογγυλοποιημένου αριθμού από τον αρχικό αριθμό (πριν τη στρογγυλοποίηση). Έτσι αν x είναι ο αριθμός κινητής υποδιαστολής πριν τη στρογγυλοποίηση και $R_{mode}(x)$ ο αριθμός κινητής υποδιαστολής μετά τη στρογγυλοποίηση με μια μέθοδο, έστω τη *mode*, το σφάλμα στρογγυλοποίησης είναι:

$$\text{Round-off Error} = R_{mode}(x) - x.$$

Ένα άλλο αριθμητικό χαρακτηριστικό των τρόπων στρογγυλοποίησης αποτελεί η *πόλωση του σφάλματος στρογγυλοποίησης (round-off error bias)*. Η πόλωση του σφάλματος στρογγυλοποίησης ορίζεται ως ο μέσος όρος των σφαλμάτων στρογγυλοποίησης. Εκφράζει την τάση του τρόπου στρογγυλοποίησης προς μια συγκεκριμένη τιμή σφάλματος. Εάν το πρόσημο της πόλωσης των σφαλμάτων είναι θετικό, τότε για μεγάλο αριθμό πράξεων κινητής υποδιαστολής ο αριθμός που θα προκύψει θα είναι μεγαλύτερος από τον αριθμό που θα προέκυπτε

αν όλα τα ψηφία ακρίβειας ήταν διαθέσιμα και δεν πραγματοποιούνταν η στρογγυλοποίηση. Το αντίστροφο ισχύει όταν το πρόσημο της πόλωσης των σφαλμάτων είναι αρνητικό. Καθόλη την παρουσίαση των τρόπων στρογγυλοποίησης στις επόμενες ενότητες, η υποδιαστολή θα υποδεικνύει το τέλος των διαθέσιμων ψηφίων για την αναπαράσταση του αριθμού κινητής υποδιαστολής και δεν θα χρησιμοποιείται για να διαχωρίσει το ακέραιο από το κλασματικό μέρος του σημαινόμενου.

3.2.1 Στρογγυλοποίηση προς το 0 - Αποκοπή

Η *στρογγυλοποίηση προς το 0* (*round to 0*) ή αλλιώς *αποκοπή* (*truncation*) είναι ο απλούστερος τρόπος στρογγυλοποίησης, καθώς δεν απαιτεί επιπλέον υλικό για την πραγμάτωσή του, αλλά ταυτόχρονα είναι ο λιγότερο αποδοτικός από πλευράς ακρίβειας. Στην αποκοπή τα επιπλέον ψηφία για τα οποία δεν υπάρχει διαθέσιμος χώρος για την αποθήκευσή τους σε έναν καταχωρητή (τα ψηφία δεξιά της υποδιαστολής), απορρίπτονται δίχως κανέναν έλεγχο. Ο Πίνακας 3.2 παρουσιάζει τον αριθμό X που έχει προκύψει από μια πράξη κινητής υποδιαστολής και έχει δύο επιπλέον ψηφία τα οποία αποκόπτονται κατά τη διαδικασία της στρογγυλοποίησης. Επίσης, στον ίδιο πίνακα παρουσιάζεται και το σφάλμα στρογγυλοποίησης που προκύπτει από τη διαφορά του στρογγυλοποιημένου αριθμού $R(X)$ από τον αριθμό X .

	$R(X)$	Σφάλμα Στρογγυλοποίησης
$X.00$	X	0
$X.01$	X	$-1/4$
$X.10$	X	$-1/2$
$X.11$	X	$-3/4$

Πίνακας 3.2: Αποκοπή και σφάλμα στρογγυλοποίησης.

Το άθροισμα των σφαλμάτων στρογγυλοποίησης είναι ίσο με $-3/2$, ενώ η μέση τιμή του σφάλματος είναι ίση με $-3/8$. Εξαιτίας του αρνητικού πρόσημου της πόλωσης που δημιουργείται κατά τη στρογγυλοποίηση, η τάση έπειτα από διαδοχικές πράξεις κινητής υποδιαστολής είναι πως το τελικό αποτέλεσμα είναι μικρότερο του αποτελέσματος που θα προέκυπτε αν δε χρησιμοποιούνταν η αποκοπή.

3.2.2 Στρογγυλοποίηση προς τον πλησιέστερο αριθμό

Η *στρογγυλοποίηση προς τον πλησιέστερο αριθμό* (*rounding-to-nearest*) είναι η πιο γνωστή μέθοδος στρογγυλοποίησης, και είναι ευρύτερα γνωστή απλά ως στρογγυλοποίηση. Στη στρογγυλοποίηση προς τον πλησιέστερο αριθμό προστίθεται μισή μονάδα του τελευταίου ψηφίου ($1/2 ulp$) στον αριθμό που στρογγυλοποιείται και στη συνέχεια απορρίπτονται τα ψηφία

δεξιά της υποδιαστολής. Στο δεκαδικό σύστημα αναπαράστασης, η μονάδα του τελευταίου ψηφίου είναι ίση με 1, άρα $1/2 ulp = 0.5$. Οπότε, αν ο αριθμός προς στρογγυλοποίηση είναι ο 3.499, τότε :

$$3.499 + 0.5 = 3.999 = 3.$$

Ενώ αν ο αριθμός προς στρογγυλοποίηση είναι ο 3.999, τότε :

$$3.999 + 0.5 = 4.499 = 4.$$

Το μεγαλύτερο σφάλμα στρογγυλοποίησης παρατηρείται όταν το πρώτο ψηφίο ακρίβειας μετά την υποδιαστολή έχει τιμή 5, ενώ τα υπόλοιπα ψηφία ακρίβειας είναι μηδενικά, π.χ. $2.500 + 0.5 = 3.000 = 3$ και είναι ίσο με 0.5 (για το δεκαδικό πάντα σύστημα). Αντίστοιχα, για το δυαδικό σύστημα το μέγιστο σφάλμα στρογγυλοποίησης είναι ίσο με 0.1_2 και παρατηρείται όταν το πρώτο ψηφίο ακρίβειας μετά την υποδιαστολή είναι ίσο με 1 και τα υπόλοιπα είναι μηδενικά. Γενικά, το μέγιστο σφάλμα παρατηρείται όταν ο αριθμός προς στρογγυλοποίηση έχει την ίδια απόσταση και από τους δύο πλησιέστερους αριθμούς (βρίσκεται ακριβώς στη μέση). Σε αυτή την περίπτωση επιλέγεται ο μεγαλύτερος των πλησιέστερων αριθμών. Ο Πίνακας 3.3 παρουσιάζει τη στρογγυλοποίηση προς τον πλησιέστερο αριθμό καθώς και τα σφάλματα στρογγυλοποίησης, για τον αριθμό X (στο δυαδικό σύστημα) που έχει δύο ψηφία ακρίβειας.

	$R(X)$	Σφάλμα Στρογγυλοποίησης
$X.00$	X	0
$X.01$	X	$-1/4$
$X.10$	$X + 1$	$+1/2$
$X.11$	$X + 1$	$+1/4$

Πίνακας 3.3: Στρογγυλοποίηση προς τον πλησιέστερο αριθμό και σφάλμα στρογγυλοποίησης.

Το άθροισμα των σφαλμάτων στρογγυλοποίησης είναι ίσο με $+1/2$, ενώ η πόλωση είναι ίση με $+1/8$. Γενικότερα, αν χρησιμοποιούνται f ψηφία ακρίβειας τότε η πόλωση είναι ίση με $1/2 \times 2^{-f}$.

3.2.3 Στρογγυλοποίηση προς τον πλησιέστερο άρτιο ή περιττό

Η στρογγυλοποίηση προς τον πλησιέστερο αριθμό αποτελεί μια σημαντική βελτίωση σε σχέση με τη στρογγυλοποίηση προς το 0. Παρόλα αυτά ο μέσος όρος των σφαλμάτων στρογγυλοποίησης προς τον πλησιέστερο αριθμό, δηλαδή η πόλωση δεν είναι μηδέν. Η αιτία είναι πως ο αριθμός που έχει την ίδια απόσταση από τους δύο πλησιέστερους, πάντα στρογγυλοποιείται στο μεγαλύτερο πλησιέστερο. Τη λύση στο πρόβλημα δίνει η *στρογγυλοποίηση προς τον πλησιέστερο άρτιο (round-to-nearest even)*, όπου ο αριθμός με την ίδια απόσταση από τους πλησιέστερους, στρογγυλοποιείται πάντα προς τον άρτιο. Εναλλακτικά, ο αριθμός

με την ίδια απόσταση από τους πλησιέστερους μπορεί να στρογγυλοποιείται πάντα προς τον περιττό. Στην περίπτωση αυτή γίνεται λόγος για *στρογγυλοποίηση προς τον πλησιέστερο περιττό (round-to-nearest odd)*. Σε κάθε άλλη περίπτωση, η στρογγυλοποίηση προς τον πλησιέστερο άρτιο ή περιττό δίνει το ίδιο αποτέλεσμα με την στρογγυλοποίηση προς τον πλησιέστερο αριθμό. Ο Πίνακας 3.4 παρουσιάζει τη στρογγυλοποίηση προς τον πλησιέστερο άρτιο και πλησιέστερο περιττό για τον αριθμό X ο οποίος έχει δύο ψηφία ακρίβειας.

	$R(X)$	Σφάλμα Στρογγυλοποίησης
$X0.00$	$X0$	0
$X0.01$	$X0$	-1/4
$X0.10$	$X1$	-1/2
$X0.11$	$X1$	+1/4
$X1.00$	$X1$	0
$X1.01$	$X1$	-1/4
$X1.10$	$X1 + 1$	+1/2
$X1.11$	$X1 + 1$	+1/4

Πίνακας 3.4: Στρογγυλοποίηση προς τον πλησιέστερο άρτιο και σφάλμα στρογγυλοποίησης.

Το άθροισμα των σφαλμάτων στρογγυλοποίησης είναι 0 και για τις δύο κατηγορίες στρογγυλοποίησης, άρα η πόλωση είναι ίση με 0.

3.2.4 Στρογγυλοποίηση προς $\pm\infty$

Η *στρογγυλοποίηση προς το $\pm\infty$ (round-to $\pm\infty$)* συναντάται συχνά στην *αριθμητική διαστημάτων (interval arithmetic)*, όπου ένας πραγματικός αριθμός αναπαρίσταται με δύο αριθμούς κινητής υποδιαστολής FP_{low} και FP_{upper} οι οποίοι παρέχουν το κάτω και άνω φράγμα, αντίστοιχα για τον πραγματικό αριθμό. Στην αριθμητική διαστημάτων, οι αριθμητικές πράξεις πραγματοποιούνται βάσει των διαστημάτων. Έτσι, αν οι πραγματικοί αριθμοί A, B αναπαρίστανται από τα διαστήματα $[a_1, a_2]$ και $[b_1, b_2]$, αντίστοιχα, που ορίζονται από τους αριθμούς κινητής υποδιαστολής, a_1, a_2 και b_1, b_2 τότε οι πράξεις της πρόσθεσης και της αφαίρεσης ορίζονται ως εξής :

$$[a_1, a_2] + [b_1, b_2] = [a_1 + b_1, a_2 + b_2]$$

$$[a_1, a_2] - [b_1, b_2] = [a_1 - b_2, a_2 - b_1],$$

ενώ η πράξη του πολλαπλασιασμού ορίζεται ως :

$$[a_1, a_2] \times [b_1, b_2] = [\min\{a_1b_1, a_1b_2, a_2b_1, a_2b_2\}, \max\{a_1b_1, a_1b_2, a_2b_1, a_2b_2\}].$$

Στους παραπάνω υπολογισμούς το κάτω φράγμα στρογγυλοποιείται προς το $-\infty$, ενώ το άνω φράγμα στρογγυλοποιείται προς το $+\infty$. Στη στρογγυλοποίηση προς το $+\infty$ επιλέγεται ο

μεγαλύτερος μεταξύ των FP_1, FP_2 . Αντίθετα, στη στρογγυλοποίηση προς το $-\infty$ επιλέγεται ο μικρότερος των FP_1, FP_2 . Ο Πίνακας 3.5 παρουσιάζει τη στρογγυλοποίηση προς το $\pm\infty$ του αριθμού X με δύο ψηφία ακρίβειας τόσο για τις θετικές όσο και για τις αρνητικές τιμές του. Επίσης, παρουσιάζονται και τα σφάλματα στρογγυλοποίησης για την κάθε περίπτωση.

Προς $+\infty$			Προς $-\infty$		
	$R(X)$	Σφάλμα		$R(X)$	Σφάλμα
$X.00$	X	0	$X.00$	X	0
$X.01$	$X + 1$	$+3/4$	$X.01$	X	$-1/4$
$X.10$	$X + 1$	$+1/2$	$X.10$	X	$-1/2$
$X.11$	$X + 1$	$+1/4$	$X.11$	X	$-3/4$
$-X.00$	$-X$	0	$-X.00$	$-X$	0
$-X.01$	$-X$	$+1/4$	$-X.01$	$-X - 1$	$-3/4$
$-X.10$	$-X$	$+1/2$	$-X.10$	$-X - 1$	$-1/2$
$-X.11$	$-X$	$+3/4$	$-X.11$	$-X - 1$	$-1/4$

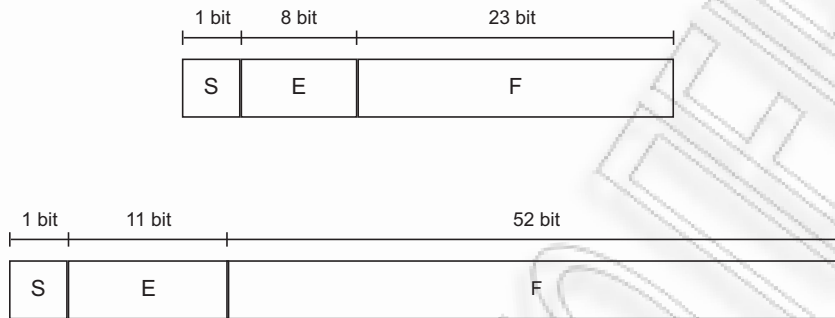
Πίνακας 3.5: Στρογγυλοποίηση προς $\pm\infty$ και σφάλμα στρογγυλοποίησης.

3.3 Το πρότυπο IEEE 754

Το πρότυπο IEEE 754 για την αναπαράσταση των πραγματικών αριθμών με τη βοήθεια των αριθμών κινητής υποδιαστολής, καθώς και ο ορισμός των πράξεων μεταξύ των τελευταίων ανακοινώθηκε το 1985. Πριν από την υιοθέτηση του προτύπου, το τοπίο στην αναπαράσταση και επεξεργασία των πραγματικών αριθμών κάθε άλλο παρά σαφές ήταν. Κάθε κατασκευαστής μονάδων κινητής υποδιαστολής χρησιμοποιούσε τη δική του αναπαράσταση για να προσεγγίσει τους πραγματικούς αριθμούς, ορίζοντας κατάλληλα το πλήθος των ψηφίων για τον εκθέτη αλλά και τη βάση του, καθώς και το πλήθος των ψηφίων για την αναπαράσταση του σημανόμενου ή του κλασματικού μέρους. Άμεσο αποτέλεσμα των αυθαίρετων ορισμών ήταν πως οι εφαρμογές που χρησιμοποιούσαν πραγματικούς αριθμούς παράγαν διαφορετικά αποτελέσματα, τα οποία ήταν εξαρτημένα από την εσωτερική αναπαράσταση των αριθμών. Φυσικά, το κόστος ανάπτυξης λογισμικού που χρησιμοποιούσε πραγματικούς αριθμούς ήταν πολύ υψηλό εξαιτίας του περιορισμένου πεδίου χρήσης. Όλα αυτά όμως άλλαξαν με την καθολική αποδοχή του προτύπου IEEE 754 για την αναπαράσταση των πραγματικών αριθμών και έτσι η ανάπτυξη εφαρμογών που παράγουν τα ίδια αποτελέσματα σε όλους τους επεξεργαστές έγινε εφικτή.

Το πρότυπο IEEE 754 παρέχει δύο βασικά μοντέλα ακρίβειας αναπαράστασης τα οποία είναι γνωστά ως, *αναπαράσταση απλής ακρίβειας* (*single precision representation*) και *αναπαράσταση διπλής ακρίβειας* (*double precision representation*). Η αναπαράσταση απλής ακρίβειας

χρησιμοποιεί 32 ψηφία για την αναπαράσταση των αριθμών κινητής υποδιαστολής, όπου το πρόσημο καταλαμβάνει 1 ψηφίο, ο εκθέτης 8 ψηφία ενώ το κλασματικό μέρος καταλαμβάνει 23 ψηφία. Αντίστοιχα, η αναπαράσταση διπλής ακρίβειας χρησιμοποιεί 64 ψηφία για την αναπαράσταση των αριθμών κινητής υποδιαστολής, όπου το πρόσημο καταλαμβάνει 1 ψηφίο, ο εκθέτης 11 ψηφία, και το κλασματικό μέρος καταλαμβάνει 52 ψηφία.



Εικόνα 3.3: Η κατανομή των ψηφίων στην απλή και διπλή ακρίβεια αναπαράστασης.

Με την αναθεώρηση του προτύπου IEEE 754 ενσωματώθηκε η αναπαράσταση των αριθμών κινητής υποδιαστολής με μισή ακρίβεια (half precision) και η αναπαράσταση με τετραπλή ακρίβεια (quaduple precision). Η αναπαράσταση των αριθμών με μισή ακρίβεια χρησιμοποιεί 16 ψηφία από τα οποία το πρόσημο καταλαμβάνει 1 ψηφίο, ο εκθέτης 5 και το κλασματικό μέρος 10. Η αναπαράσταση με μισή ακρίβεια έχει μεγάλη απήχηση στα γραφικά με υπολογιστές (computer graphics) και χρησιμοποιείται από περιβάλλοντα όπως τα OpenGL, OpenEXR, D3DX. Από την άλλη πλευρά, η αναπαράσταση τετραπλής ακρίβειας χρησιμοποιεί 128 ψηφία από τα οποία 1 καταλαμβάνει το πρόσημο, 15 ο εκθέτης και 112 το κλασματικό μέρος και χρησιμοποιείται σε προσομοιώσεις αντοχής υλικών.

Η αναπαράσταση ενός αριθμού κινητής υποδιαστολής σύμφωνα με το πρότυπο IEEE 754 γίνεται βάσει της τριάδας (S, E, F) . Το σύμβολο S αναπαριστά το πρόσημο, όπου όταν $S = 1$ ο αριθμός είναι αρνητικός. Το σύμβολο E αναπαριστά τον εκθέτη σε πολωμένη σημειογραφία και η βάση του εκθέτη είναι το 2. Η πόλωση σύμφωνα με την εξίσωση 3.8 για την αναπαράσταση απλής ακρίβειας είναι 127 ενώ για την αναπαράσταση διπλής ακρίβειας 1023. Τέλος, το σύμβολο F αναπαριστά το κλασματικό μέρος του σημαινόμενου το οποίο βρίσκεται σε κανονικοποιημένη μορφή και αποτελείται από f ψηφία (23 για απλή ακρίβεια και 52 για διπλή ακρίβεια), ενώ η μονάδα αριστερά της υποδιαστολής αναπαριστά το ακέραιο μέρος του αριθμού κινητής υποδιαστολής. Η μονάδα που βρίσκεται αριστερά της υποδιαστολής είναι γνωστή με το όνομα *κρυφή μονάδα* (hidden one) ή *κρυφό ψηφίο* (hidden bit). Η κρυφή μονάδα, αν και αποτελεί μέρος του σημαινόμενου, δεν αποθηκεύεται ως μέρος του αριθμού κινητής υποδιαστολής αλλά υπονοείται. Αυτό συμβαίνει διότι οι αριθμοί κινητής υποδιαστολής είναι σε κανονική μορφή (normalized), άρα το ψηφίο αριστερά της υποδιαστολής είναι πάντοτε η

μονάδα. Με αυτόν τον τρόπο εξοικονομείται ένα επιπλέον ψηφίο για την αναπαράσταση του κλασματικού μέρους. Η αναπαράσταση των αρνητικών αριθμών γίνεται με βάση το πρόσημο και μέγεθος (sign and magnitude), όπου το πρόσημο δηλώνεται από το σύμβολο S . Το ακέραιο μέρος μαζί με το κλασματικό μέρος αποτελούν το *σημαινόμενο* (*significand* ή *mantissa*) του αριθμού κινητής υποδιαστολής. Το κανονικοποιημένο σημαινόμενο έχει τη μορφή $1.F$ και έχει εύρος :

$$1 \leq 1.F \leq 2 - 2^{-f}.$$

Η τριάδα (S, E, F) παριστάνει τον πραγματικό αριθμό :

$$(-1)^S \times 2^{E-Bias} \times 1.F.$$

Το πρότυπο IEEE 754 υποστηρίζει την αναπαράσταση των ελάχιστων μη κανονικοποιημένων αριθμών (denormals) ή των υποκανονικών (subnormals) αριθμών θέτοντας στον εκθέτη τη μικρότερη δυνατή τιμή, ενώ το κλασματικό μέρος είναι διάφορο του μηδενός. Με αυτόν τον τρόπο είναι δυνατή η απεικόνιση των αριθμών :

$$-1^S \times 0.F \times 2^{-(Bias-1)},$$

όπου Bias είναι η πόλωση του εκθέτη E . Η υποστήριξη και υλοποίηση των μη κανονικοποιημένων αριθμών δεν είναι υποχρεωτική σύμφωνα με το πρότυπο IEEE 754, διότι η διαχείριση των υποκανονικών αριθμών αυξάνει την πολυπλοκότητα μιας μονάδας επεξεργασίας αριθμών κινητής υποδιαστολής με άμεση επίδραση στο μέγεθος και την ταχύτητά της.

Οι πράξεις κινητής υποδιαστολής που υποστηρίζει το πρότυπο μπορούν να χωριστούν στις αριθμητικές, δηλαδή, πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαίρεση, τετραγωνική ρίζα ($+$, $-$, \times , \div , $\sqrt{\quad}$) και στις πράξεις σύγκρισης ή μετατροπής από έναν αριθμό κινητής υποδιαστολής σε έναν ακέραιο και αντίστροφα. Το πρότυπο IEEE 754 υποστηρίζει τη στρογγυλοποίηση προς τον πλησιέστερο άρτιο (ο οποίος είναι ο εξ' ορισμού τρόπος στρογγυλοποίησης), τη στρογγυλοποίηση προς το 0 ή αποκοπή (truncation) καθώς και τη στρογγυλοποίηση προς το $\pm\infty$.

3.3.1 Ειδικές Τιμές

Το πρότυπο IEEE 754 χρησιμοποιεί προκαθορισμένες τιμές για την αναπαράσταση των ειδικών τιμών που μπορεί να προκύψουν έπειτα από την εκτέλεση μιας πράξης κινητής υποδιαστολής. Η τιμή 0 αναπαρίσταται θέτοντας στα ψηφία του εκθέτη και του κλασματικού μέρους του σημαινόμενου την τιμή 0. Αξιοσημείωτο είναι πως με αυτό τον τρόπο το πρότυπο επιτρέπει το διαχωρισμό του $+0$ από το -0 , παρόλο που ορίζει πως το $+0 = -0$.

Για την αναπαράσταση του $\pm\infty$ το πρότυπο ορίζει πως ο εκθέτης λαμβάνει τη μέγιστη δυνατή τιμή, θέτοντας στα ψηφία του εκθέτη την τιμή 1, ενώ αντίθετα στα ψηφία του κλασματικού μέρους τίθεται η τιμή 0. Το πρόσημο καθορίζει εάν γίνεται αναφορά στο $+\infty$ ή στο $-\infty$.

	Μονή Ακρίβεια	Διπλή Ακρίβεια
Ψηφία	32	64
Ψηφία Σημανιόμενου	$23 + 1$	$52 + 1$
Εύρος Σημανιόμενου	$[1, 2 - 2^{-23})$	$[1, 2 - 2^{-52})$
Ψηφία Εκθέτη	8	11
Πόλωση	127	1023
Μηδέν(± 0)	$E + bias = 0, F = 0$	$E + bias = 0, F = 0$
Υποκανονικός	$E + bias = 0, F \neq 0$ αναπαριστά τον $\pm 0.F \times 2^{-126}$	$E + bias = 0, F \neq 0$ αναπαριστά τον $\pm 0.F \times 2^{-1022}$
Απειρο($\pm \infty$)	$E + bias = 255, F = 0$	$E + bias = 2047, F = 0$
Όχι Αριθμός NaN	$E + bias = 255, F \neq 0$	$E + bias = 2047, F \neq 0$
Συνήθης Αριθμός	$E + bias \in [1, 254]$ $E \in [-126, 127]$ αναπαριστά τον $\pm 1.F \times 2$	$E + bias \in [1, 2046]$ $E \in [-1022, 1023]$ αναπαριστά τον $\pm 1.F \times 2$
Ελάχιστος	$2^{-126} \approx 1.2 \times 10^{-38}$	$2^{-1022} \approx 2.2 \times 10^{-308}$
Μέγιστος	$(2 - 2^{-127}) \times 2^{127} \approx 2^{128}$ $\approx 3.4 \times 10^{38}$	$(2 - 2^{-1023}) \times 2^{1023} \approx 2^{1024}$ $\approx 1.8 \times 10^{308}$

Πίνακας 3.6: Τα βασικά χαρακτηριστικά της αναπαράστασης Μονής και Διπλής ακρίβειας σύμφωνα με το πρότυπο IEEE 754.

Όταν το αποτέλεσμα της πράξης κινητής υποδιαστολής δεν ορίζεται, π.χ. $(+\infty) - (\infty)$ ή $\infty \div \infty$, το πρότυπο χρησιμοποιεί την ειδική τιμή *όχι-αριθμός (Not-a-Number (NaN))* για να δείξει πως η πράξη δεν είναι έγκυρη. Η αναπαράσταση της τιμής *όχι-αριθμός* είναι παρόμοια με την αναπαράσταση του $\pm \infty$ με τη διαφορά πως ένα τουλάχιστον από τα ψηφία του κλασματικού μέρους είναι μη μηδενικό.

Τέλος για την αναπαράσταση των μη κανονικοποιημένων αριθμών με τον ελάχιστο μη μηδενικό εκθέτη (denormals) το πρότυπο ορίζει πως τα ψηφία του εκθέτη λαμβάνουν την τιμή 0, ενώ ένα εκ των ψηφίων του κλασματικού μέρους είναι μη μηδενικό.

Ο Πίνακας 3.6 παρουσιάζει τα βασικά χαρακτηριστικά των δύο μοντέλων ακρίβειας αναπαράστασης σύμφωνα με το πρότυπο καθώς και την αναπαράσταση των ειδικών τιμών, 0, NaN, $\pm \infty$.

3.3.2 Εξαιρέσεις

Το πρότυπο IEEE 754 ορίζει πέντε τύπους εξαιρέσεων, οι οποίες εξ' ορισμού θέτουν την αντίστοιχη *σημαία εξαιρέσεων (exception flag)*. Όταν συμβεί μια εξαίρεση, είτε συνεχίζεται ο υπολογισμός κινητής υποδιαστολής, είτε ενεργοποιείται η *ρουτίνα παγίδευσης (trap handler)* για να διαχειριστεί την εξαίρεση. Οι εξαιρέσεις που υποστηρίζονται από το IEEE 754 είναι οι

ακόλουθες:

- Υπερχείλιση κινητής υποδιαστολής (floating point overflow)
Συμβαίνει όταν ο αριθμός δεν μπορεί να αναπαρασταθεί. Στην περίπτωση αυτή στο αποτέλεσμα τίθεται η τιμή $\pm\infty$.
- Ανεπάρκεια αναπαράστασης κινητής υποδιαστολής (floating point underflow)
Ενεργοποιείται όταν ο αριθμός κινητής υποδιαστολής είναι πολύ μικρός. Όταν δεν υλοποιούνται οι ελάχιστοι μη κανονικοποιημένοι αριθμοί στο αποτέλεσμα τίθεται η τιμή 0.
- Διάρθρωση με το 0 (divide by 0)
Το πρότυπο ορίζει πως το αποτέλεσμα της διαίρεσης με το 0 είναι ίσο με $\pm\infty$.
- Μη ακριβές αποτέλεσμα (inexact)
Η εξαίρεση ενεργοποιείται όταν εξαιτίας της στρογγυλοποίησης απορρίπτονται ψηφία τα οποία δεν είναι μηδενικά.
- Μη έγκυρη πράξη κινητής υποδιαστολής (invalid)
Συμβαίνει όταν εκτελείται μια πράξη κινητής υποδιαστολής η οποία δεν ορίζεται ή όταν ένας από τους τελεστέους είναι όχι-αριθμός (NaN).

3.4 Πρόσθεση–Αφαίρεση

Η πρόσθεση/αφαίρεση των αριθμών κινητής υποδιαστολής είναι η πιο πολύπλοκη και συνάμα η συννηθέστερη μεταξύ όλων των αριθμητικών πράξεων. Έστω δύο αριθμοί κινητής υποδιαστολής, A , B , οι οποίοι αναπαρίστανται στο πρότυπο IEEE 754 με τις ακόλουθες τριάδες: $A = (S_A, E_A, F_A)$ και $B = (S, E, F)$. Επειδή η αφαίρεση δύο αριθμών κινητής υποδιαστολής είναι ισοδύναμη με την πρόσθεση των αριθμών με διαφορετικά πρόσημα, η ανάπτυξη που ακολουθεί αναφέρεται μόνο στην πράξη της πρόσθεσης. Πριν παρουσιάσουμε τα αναλυτικά βήματα της πρόσθεσης δίνουμε το ακόλουθο παράδειγμα το οποίο δείχνει τις βασικές αρχές του αλγορίθμου.

Παράδειγμα: Ας υποθέσουμε πως οι αριθμοί που θα προστεθούν είναι οι δεκαδικοί αριθμοί 9.987×10^1 και 1.85×10^{-1} και συνολικά δεσμεύονται 4 ψηφία για το ακέραιο και το κλασματικό μέρος. Είναι γνωστό πως οι αριθμοί μπορεί να προστεθούν ή να αφαιρεθούν μόνο όταν έχουν τον ίδιο εκθέτη. Στο παράδειγμα ο αριθμός με το μικρότερο εκθέτη πολλαπλασιάζεται και διαιρείται με 10^2 . Ο εκθέτης είναι η απόλυτη τιμή της διαφοράς των εκθετών των δύο αριθμών, $|1 - (-1)| = 2$. Αυτό σημαίνει πως ο αριθμός με τον μικρότερο εκθέτη ολισθαίνει δεξιά κατά δύο θέσεις. Άρα :

$$1.85 \times 10^{-1} = 0.0185 \times 10^1.$$

Αφού μόνο 4 ψηφία είναι διαθέσιμα το ψηφίο 5 χάνεται και έχουμε 0.018×10^{-1} . Η απόρριψη των ψηφίων κατά την ολίσθηση είναι μια λανθασμένη ενέργεια καθώς χάνεται ακρίβεια από το τελικό αποτέλεσμα όπως θα δούμε στην συνέχεια. Καθώς οι εκθέτες είναι ίσοι μπορούμε να προχωρήσουμε στην πρόσθεση των αριθμών, άρα :

$$\begin{array}{r} 9.987 \times 10^1 \\ + \quad 0.018 \times 10^1 \\ \hline 10.005 \times 10^1 \end{array}$$

Το αποτέλεσμα δεν βρίσκεται σε κανονική μορφή. Μια ολίσθηση κατά μια θέση δεξιά και μια αύξηση του εκθέτη κατά 1 θα φέρει το αποτέλεσμα σε κανονική μορφή. Άρα, μετά την ολίσθηση έχουμε 1.0005×10^2 . Στην συνέχεια το αποτέλεσμα θα πρέπει να στρογγυλοποιηθεί ώστε να καταλαμβάνει συνολικά 4 ψηφία. Στρογγυλοποιώντας προς τον πλησιέστερο αριθμό ελέγχουμε αν το ψηφίο που περισσεύει είναι μεταξύ 5 και 9 οπότε προσθέτουμε 1 ή μεταξύ 0 και 4 οπότε δεν προσθέτουμε. Στο παράδειγμα, το ψηφίο που περισσεύει είναι το 5, άρα προσθέτουμε τη μονάδα στο τέταρτο ψηφίο, οπότε το αποτέλεσμα μετά την στρογγυλοποίηση είναι 1.001×10^2 . Δηλαδή το άθροισμα του $9.987 \times 10^1 = 99.87$ με τον $1.85 \times 10^{-1} = 0.185$ είναι ίσο με 100.1.

Γενικά, τα βήματα για την πρόσθεση των αριθμών, A , B έχουν ακολούθως:

1. Αφαίρεση των εκθετών - Εναλλαγή (Swap):

Οι εκθέτες των αριθμών A και B αφαιρούνται. Η απόλυτη τιμή της διαφοράς τους, $d = |E_A - E_B|$, χρησιμοποιείται ώστε να διαπιστωθεί εάν οι αριθμοί κινητής υποδιαστολής έχουν τον ίδιο εκθέτη, ενώ το πρόσημο χρησιμοποιείται για να επιλεγεί ο αριθμός με το μεγαλύτερο εκθέτη. Εάν η διαφορά των εκθετών είναι θετική τότε ο εκθέτης του αριθμού A , ο E_A επιλέγεται για περαιτέρω επεξεργασία στα επόμενα στάδια της πρόσθεσης. Σε αντίθετη περίπτωση, επιλέγεται ο εκθέτης του αριθμού B , ο E_B . Η απόλυτη τιμή της διαφοράς των εκθετών θα χρησιμοποιηθεί στο στάδιο της προ-κανονικοποίησης.

Επίσης, σε αυτό το στάδιο λαμβάνει χώρα η *εναλλαγή (swap)* των σημαινόμενων n οποία αν και δεν είναι απαραίτητη για την ορθή εκτέλεση της πράξης κινητής υποδιαστολής αποτελεί μια βελτιστοποίηση του αλγορίθμου της πρόσθεσης. Η εναλλαγή υλοποιείται με τη βοήθεια ενός πολυπλέκτη του οποίου οι είσοδοι είναι τα σημαινόμενα των αριθμών κινητής υποδιαστολής και η είσοδος ελέγχου είναι το πρόσημο της διαφοράς των εκθετών. Με αυτόν τον τρόπο επιλέγεται πάντα το σημαινόμενο με το μικρότερο εκθέτη το οποίο οδηγείται στον ολισθητή για το βήμα της προ-κανονικοποίησης. Επιπλέον, μπορεί να γίνει και σύγκριση των σημαινόμενων ώστε να αφαιρείται πάντα το μικρότερο σημαινόμενο από το μεγαλύτερο. Με αυτόν τον τρόπο, το αποτέλεσμα της αφαίρεσης θα είναι πάντα θετικό και δεν θα χρειάζεται επιπλέον υλικό για τον υπολογισμό του συμπληρώματος ως προς 2, ώστε το αρνητικό αποτέλεσμα να μετατραπεί σε θετικό.

Διαφορά	Σημαινόμενο	G	R	T
$d = 0$	$1.F_{f-1}F_{f-2} \dots F_1F_0$	0	0	0
$d = 1$	$0.1F_{f-1}F_{f-2} \dots F_2F_1$	F_0	0	0
$d = 2$	$0.01F_{f-1}F_{f-2} \dots F_3F_2$	F_1	F_0	0
$d = 3$	$0.001F_{f-1}F_{f-2} \dots F_4F_3$	F_2	F_1	F_0
$d \geq 4$	$\underbrace{0.00 \dots 00}_d 1F_{f-1}F_{f-2} \dots F_{d+1}F_d$	F_{d-1}	F_{d-2}	$F_{d-3} + \dots + F_0$

Πίνακας 3.7: Υπολογισμός των Ψηφίων Φρουρών Guard (G), Round (R), Sticky (T).

2. Προ-Κανονικοποίηση (PreNormalization):

Η πρόσθεση δύο αριθμών κινητής υποδιαστολής επιτρέπεται μόνο όταν οι εκθέτες τους είναι ίσοι. Στο στάδιο της *προ-κανονικοποίησης*, το κλασματικό μέρος του αριθμού με το μικρότερο εκθέτη ολισθαίνει δεξιά τόσες θέσεις όσο είναι η απόλυτη τιμή της διαφοράς των εκθετών.

Κατά τη δεξιά ολίσθηση του σημαίνόμενου με το μικρότερο εκθέτη *δεν πρέπει* να χαθούν ψηφία, τα οποία μπορεί να επηρεάσουν τη στρογγυλοποίηση του αποτελέσματος της πρόσθεσης ή αφαιρέσης των σημανόμενων. Με λίγα λόγια, η τελική στρογγυλοποίηση θα πρέπει να πραγματοποιείται σαν να ήταν διαθέσιμα όλα τα ψηφία μετά την ολοκλήρωση της προ-κανονικοποίησης. Για την επίτευξη της στρογγυλοποίησης με μεγαλύτερη ακρίβεια χρησιμοποιούνται τρία επιπλέον ψηφία, γνωστά ως *ψηφία φρουροί (guard bits)*. Έτσι, αν το σημανόμενο έχει μέγεθος $m = f + 1$ ψηφία, όπου f τα ψηφία του κλασματικού μέρους, τότε κατά την προ-κανονικοποίηση θα προκύψουν $m + 3$ ψηφία. Έπειτα από τη δεξιά ολίσθηση, το ψηφίο που βρίσκεται μια θέση δεξιότερα του τελευταίου ψηφίου του νέου σημανόμενου είναι το *ψηφίο φρουρός (guard bit-G)* και το αμέσως δεξιότερο είναι το *ψηφίο στρογγύλευσης (round bit-R)*. Τέλος, τα υπόλοιπα ψηφία που βρίσκονται δεξιότερα του ψηφίου στρογγυλοποίησης συμμετέχουν σε πράξη *λογικό-Η (logic-OR)* για να σχηματίσουν το *επίμονο ψηφίο (sticky bit-T)*. Στην πραγματικότητα, το επίμονο ψηφίο δηλώνει εάν ένα ή και περισσότερα από τα ψηφία δεξιότερα του ψηφίου στρογγύλευσης είναι μη μηδενικά. Ο Πίνακας 3.7 παρουσιάζει τον τρόπο με τον οποίο σχηματίζονται τα τρία ψηφία φρουροί ανάλογα με την απόλυτη τιμή της διαφοράς των εκθετών, $d = |E_A - E_B|$, όπου f είναι τα ψηφία για την αναπαράσταση του κλασματικού μέρους ($f = 23$ για την απλή ακρίβεια και $f = 52$ για τη διπλή ακρίβεια) και d είναι η απόλυτη τιμή της διαφοράς των εκθετών, ενώ ο τελεστής + αντιπροσωπεύει την πράξη λογικό-Η (logic-OR).

Τα ψηφία φρουροί προσκολλώνται στο τέλος των ψηφίων των σημανόμενων (αμέσως μετά το λιγότερο σημαντικό ψηφίο του σημανόμενου) και συμμετέχουν στην πράξη της πρόσθεσης ή αφαιρέσης. Τα ψηφία φρουροί για το σημανόμενο του αριθμού με το

Πράξη Κινητής Υποδιαστολής	Πρόσημα S_A, S_B	Πραγματική Πράξη
Πρόσθεση	Ομόσημα	Πρόσθεση
Πρόσθεση	Ετερόσημα	Αφαίρεση
Αφαίρεση	Ομόσημα	Αφαίρεση
Αφαίρεση	Ετερόσημα	Πρόσθεση

Πίνακας 3.8: Πραγματική Πράξη ανάλογα με τα πρόσημα και την πράξη κινητής υποδιαστολής.

μεγαλύτερο εκθέτη είναι μηδενικά.

Έστω ότι 9 ψηφία είναι διαθέσιμα για την αναπαράσταση των αριθμών κινητής υποδιαστολής. Αν η απόλυτη τιμή της διαφοράς των εκθετών είναι 4, τότε το σημειώμενο 1.00110010 έπειτα από την αριστερή ολίσθηση θα είναι ίσο με :

$$0.00010011|00(1 + 0) = 0.00010011|001,$$

όπου το σύμβολο | διαχωρίζει το κλασματικό μέρος από τα ψηφία φρουρούς τα οποία είναι $G = 0$, $R = 0$, $T = 1$.

3. Πρόσθεση—Αφαίρεση:

Τα σημειώμενα των αριθμών κινητής υποδιαστολής εκτεταμένα με τα ψηφία φρουρούς προστίθενται ή αφαιρούνται σύμφωνα με την *πραγματική πράξη (effective operation)*. Έτσι, για την αναπαράσταση απλής ακρίβειας ο αθροιστής των σημειώμενων έχει μέγεθος $23 + 1 + 3 = 27$ ψηφία, ενώ ο αντίστοιχος αθροιστής για την αναπαράσταση διπλής ακρίβειας έχει μέγεθος $52 + 1 + 3 = 56$ ψηφία. Η πραγματική πράξη διαπιστώνεται από τα πρόσημα των αριθμών A, B και από την πράξη κινητής υποδιαστολής. Ο Πίνακας 3.8 παρουσιάζει την πραγματική πράξη λαμβάνοντας υπόψη τα πρόσημα των αριθμών κινητής υποδιαστολής, S_A, S_B και την πράξη κινητής υποδιαστολής.

Για την πράξη της πρόσθεσης ή αφαίρεσης χρησιμοποιείται συνήθως ένας πολύ γρήγορος αθροιστής—αφαιρέτης με πρόβλεψη κρατουμένου (carry lookahead), ενώ η αφαίρεση πραγματοποιείται με τη βοήθεια του συμπληρώματος ως προς 2. Δε χρησιμοποιείται αθροιστής—αφαιρέτης προσίμου και μεγέθους εξαιτίας της υψηλής πολυπλοκότητάς του. Εάν κατά το στάδιο της εναλλαγής δεν έχει γίνει σύγκριση των σημειώμενων όταν οι εκθέτες είναι ίσοι, τότε μπορεί να αφαιρεθεί ένας μεγαλύτερος αριθμός από ένα μικρότερο. Στην περίπτωση αυτή λαμβάνεται το συμπλήρωμα ως προς 2 του αριθμού αυτού, ώστε να προκύψει ένας θετικός αριθμός.

Επίσης στο στάδιο αυτό διαμορφώνεται το πρόσημο του αποτελέσματος της πράξης της κινητής υποδιαστολής συναρτήσει των προσίμων των αριθμών, της πράξης κινητής υποδιαστολής και μιας λογικής μεταβλητής που εκφράζει ποιο από τα σημειώμενα είναι μεγαλύτερο κατά απόλυτη τιμή. Το πρόσημο του αποτελέσματος είναι ίσο με το

αποτέλεσμα της λογικής συνάρτησης:

$$(Abig \cdot S_A) + (S_A + Abig') \cdot (FPOp \oplus S_B),$$

όπου η $Abig = 1$ όταν $1.F_A \geq 1.F_B$, ενώ $Abig = 0$ όταν $1.F_A < 1.F_B$, και $FPOp = 0$ όταν η πράξη κινητής υποδιαστολής είναι πρόσθεση, ενώ $FPOp = 1$ όταν είναι αφαίρεση.

Τέλος, εάν οι αριθμοί κινητής υποδιαστολής είναι ίσοι, το τελικό πρόσθετο είναι θετικό, δηλαδή $+0$. Εξαίρεση στην παραπάνω περίπτωση, αποτελεί η στρογγυλοποίηση προς το $-\infty$ όπου το τελικό πρόσθετο είναι αρνητικό, -0 .

4. Μετα-Κανονικοποίηση (PostNormalization):

Το αποτέλεσμα της πρόσθεσης ή αφαίρεσης μπορεί να μην είναι κανονικοποιημένο. Αυτό συμβαίνει καθώς για τους αριθμούς κινητής υποδιαστολής A, B , ισχύει :

$$A \in [1, 2) \text{ και } B \in [1, 2) \Rightarrow A \pm B \in [0, 4).$$

Διακρίνονται οι ακόλουθες περιπτώσεις:

- Το αποτέλεσμα είναι *κανονικοποιημένο*.

Στην περίπτωση αυτή $A \pm B \in [1, 2)$, οπότε δεν απαιτείται καμία περαιτέρω ενέργεια, όπως φαίνεται και από το ακόλουθο παράδειγμα :

$$\begin{array}{r} 1.10011111 \\ + 0.00101011 \\ \hline 1.11001010 \end{array}$$

- Το αποτέλεσμα της ακέραιας πρόσθεσης των σημαϊνόμενων έχει *υπερχείλιση*, δηλαδή ο ακέραιος αθροιστής γεννά ένα κρατούμενο εξόδου (carry out).

Υπερχείλιση είναι δυνατό να συμβεί μόνο όταν η πραγματική πράξη είναι πρόσθεση. Στην περίπτωση αυτή $A + B \in [2, 4)$. Η υπερχείλιση αναφέρεται στο αποτέλεσμα της πρόσθεσης των σημαϊνόμενων και όχι στην αναπαράσταση του τελικού αριθμού κινητής υποδιαστολής. Οι ενέργειες που απαιτούνται για την κανονικοποίηση του αποτελέσματος είναι οι παρακάτω:

- Δεξιά ολίσθηση του αθροίσματος κατά μια θέση.
- Προσαύξηση του εκθέτη κατά μια μονάδα.

Η πρόσθεση των παρακάτω σημαϊνόμενων οδηγεί σε υπερχείλιση διότι το αποτέλεσμα της πρόσθεσης χρειάζεται ένα επιπλέον ψηφίο για να αναπαρασταθεί. Στην περίπτωση αυτή πραγματοποιείται δεξιά ολίσθηση. Το επιμόνο ψηφίο δεν αποκόπτεται αλλά συνυπολογίζεται μέσω της πράξης λογικό-Η με το ψηφίο στρογγύλευσης και σχηματίζει ένα νέο επιμόνο ψηφίο. Με αυτό τον τρόπο εξασφαλίζεται η

διατήρηση της ακρίβειας του τελικού αποτελέσματος :

$$\begin{array}{r}
 1.1001111 \\
 + 0.0110110 \\
 \hline
 10.0000101 \\
 \text{Κανονικοποίηση} \quad 1.0000011
 \end{array}$$

- Το αποτέλεσμα έχει αρχικά μηδενικά (*leading zeros*). Όταν η πραγματική πράξη είναι αφαίρεση τότε μπορεί να προκύψουν αρχικά μηδενικά, καθώς $A - B \in [0, 1)$. Η κανονικοποίηση της διαφοράς απαιτεί τις παρακάτω ενέργειες:

- Αριστερή ολίσθηση της διαφοράς κατά τόσες θέσεις όσα και τα αρχικά μηδενικά.
- Μείωση του εκθέτη ανάλογα με τον αριθμό των αρχικών μηδενικών.

Η διαφορά των παρακάτω σημειόμενων γεννά αρχικά μηδενικά :

$$\begin{array}{r}
 1.1001111 \\
 - 1.1001010 \\
 \hline
 0.0000101 \\
 \text{Κανονικοποίηση} \quad 1.0100000
 \end{array}$$

5. Στρογγυλοποίηση (Rounding):

Η στρογγυλοποίηση πραγματοποιείται ανάλογα με την επιλογή στρογγυλοποίησης που ορίζεται κατά την πράξη κινητής υποδιαστολής.

Έστω ότι έπειτα από την ολοκλήρωση της μετά-κανονικοποίησης το νέο σημειόμενο είναι ίσο με:

$$1.F_{f-1}F_{f-2} \dots F_1L|GRT,$$

όπου $L = F_0$, δηλαδή είναι το λιγότερο σημαντικό ψηφίο του σημειόμενου. Υπάρχει πιθανότητα να συμβεί υπερχείλιση του σημειόμενου κατά τη στρογγυλοποίηση. Σε αυτή την περίπτωση το αποτέλεσμα ολισθαίνει κατά μια θέση δεξιά και προσ αυξάνεται ο εκθέτης.

Στρογγυλοποίηση προς τον πλησιέστερο άρτιο: Πραγματοποιείται προσθέτοντας τη μονάδα στο λιγότερο σημαντικό ψηφίο, δηλαδή στο ψηφίο L , όταν:

- $G = 1$ και ένα εκ των R, T ή και τα δύο είναι μη μηδενικά, δηλαδή $G(R+T) = 1$.
- $G = 1, L = 1$ και τα ψηφία R και T είναι μηδενικά, δηλαδή $GLR'T' = 1$.

Συνδυάζοντας τις παραπάνω περιπτώσεις η στρογγυλοποίηση προς τον πλησιέ-

στερο άρτιο γίνεται όταν:

$$\begin{aligned} G(R+T) + GLR'T' &= G(R+T+LR'T') = \\ G(R+(T+T')(T+LR')) &= G(R+T+LR') = \\ G((R+R')(R+L)+T) &= G(L+R+T), \end{aligned}$$

δηλαδή όταν $G = 1$ και ένα τουλάχιστον εκ των L, R, T είναι 1.

Στρογγυλοποίηση προς το 0: Στην αποκοπή τα ψηφία G, R, T , απορρίπτονται δίχως περαιτέρω επεξεργασία.

Στρογγυλοποίηση προς το $+\infty$: Πραγματοποιείται προσθέτοντας τη μονάδα στο λιγότερο σημαντικό ψηφίο του σημαινόμενου, L , όταν το πρόστιμο του αριθμού είναι θετικό και ένα τουλάχιστον ψηφίο εκ των G, R, T είναι μη μηδενικό, δηλαδή :

$$S'(G+R+T)$$

Στρογγυλοποίηση προς το $-\infty$: Πραγματοποιείται προσθέτοντας τη μονάδα στο λιγότερο σημαντικό ψηφίο του σημαινόμενου, L , όταν το πρόστιμο του αριθμού είναι αρνητικό και ένα τουλάχιστον ψηφίο εκ των G, R, T είναι μη μηδενικό, δηλαδή :

$$S(G+R+T)$$

Στις παραπάνω συνθήκες στρογγυλοποίησης η έκφραση $R+T$ μπορεί να αντικατασταθεί με μια νέα μεταβλητή, T^* , όπου ο τελεστής $+$ είναι η πράξη λογικό-Η.

Η στρογγυλοποίηση προς τον πλησιέστερο άρτιο απαιτεί δύο επιπλέον ψηφία. Αν και διαφαίνεται πως δύο μόνο ψηφία θα έπρεπε να διατηρούνται κατά την προ-κανονικοποίηση τα παρακάτω παραδείγματα δείχνουν ότι η πρόσθεση/αφαίρεση αριθμών κινητής υποδιαστολής χωρίς και τα τρία ψηφία φρουρούς οδηγεί σε αποτελέσματα λανθασμένης ακρίβειας.

Παράδειγμα: Έστω ότι θέλουμε να προσθέσουμε τους αριθμούς 5.01×10^{-1} και 5.64×10^2 στρογγυλοποιώντας το άθροισμα προς τον πλησιέστερο άρτιο. Η απόλυτη τιμή της διαφοράς των εκθετών είναι 3 άρα ο αριθμός 5.01 θα πρέπει να ολισθήσει 3 θέσεις δεξιά. Ο πίνακας που ακολουθεί παρουσιάζει την πρόσθεση των αριθμών με δύο και τρία ψηφία φρουρούς αντίστοιχα.

Στρογγυλοποίηση με 2 ψηφία φρουρούς	Στρογγυλοποίηση με 3 ψηφία φρουρούς
$\begin{array}{r} 5.64 00 \times 10^2 \\ + 0.00 50 \times 10^2 \\ \hline 5.64 50 \times 10^2 \\ \text{Στρογγ. } 5.64 \times 10^2 \end{array}$	$\begin{array}{r} 5.64 000 \times 10^2 \\ + 0.00 501 \times 10^2 \\ \hline 5.64 501 \times 10^2 \\ \text{Στρογγ. } 5.65 \times 10^2 \end{array}$

Στην αριστερή στήλη η πρόσθεση πραγματοποιείται με δύο ψηφία φρουρούς οπότε ένα ψηφίο από τον αριθμό $5,01 \times 10^{-1}$ χάνεται. Το αποτέλεσμα της πρόσθεσης πριν την στρογγυλοποίηση είναι $5,64|50$ και βρίσκεται ακριβώς στη μέση μεταξύ των αριθμών $5,64$ και $5,65$. Επειδή πραγματοποιούμε στρογγυλοποίηση προς τον πλησιέστερο άρτιο, το τελικό αποτέλεσμα της στρογγυλοποίησης είναι $5,64$.

Αντίθετα στην δεξιά στήλη χρησιμοποιούνται τρία ψηφία φρουροί για την πρόσθεση. Το άθροισμα πριν την στρογγυλοποίηση είναι $5,64|501$ το οποίο βρίσκεται πιο κοντά στον αριθμό $5,65$. Την πληροφορία αυτή την παρέχει το επίμονο ψηφίο το οποίο στην αριστερή στήλη του πίνακα δε χρησιμοποιήθηκε με αποτέλεσμα τα δύο αθροίσματα να διαφέρουν κατά 1 μονάδα στην τελευταία θέση.

Το επίμονο ψηφίο είναι απαραίτητο για την ορθή στρογγυλοποίηση προς τον πλησιέστερο άρτιο κατά την αφαίρεση δύο αριθμών κινητής υποδιαστολής, όπως φαίνεται στο ακόλουθο παράδειγμα.

Παράδειγμα: Έστω ότι θέλουμε να αφαιρέσουμε τον αριθμό $1.10000010110 \times 2^{-6}$ από τον 1.00000101100×2^0 . Η απόλυτη τιμή της διαφοράς των εκθετών είναι 6, οπότε το σημαντικό του αριθμού με το μικρότερο εκθέτη θα πρέπει ολισθήσει 6 θέσεις δεξιά. Στην Εικόνα 3.4 παρουσιάζεται η αφαίρεση των αριθμών κινητής υποδιαστολής θεωρώντας πως είναι διαθέσιμα όλα τα ψηφία έπειτα από τη δεξιά ολίσθηση (προ-κανονικοποίηση). Η διαφορά των σημανόμενων δε βρίσκεται σε κανονική μορφή, αφού το ψηφίο αριστερά της υποδιαστολής είναι μηδενικό, επομένως πραγματοποιείται ολίσθηση κατά μια θέση αριστερά (μετά-κανονικοποίηση). Το κανονικοποιημένο αποτέλεσμα δε χρειάζεται στρογγυλοποίηση.

Η Εικόνα 3.5 παρουσιάζει την αφαίρεση των αριθμών κινητής υποδιαστολής θεωρώντας δύο ψηφία φρουρούς. Το πρώτο ψηφίο που διατηρείται είναι το ψηφίο φρουρός (guard bit) G , ενώ το δεύτερο ψηφίο είναι το επίμονο ψηφίο (sticky bit) T . Η κανονικοποιημένη διαφορά είναι $1.1111110111|1$. Η στρογγυλοποίηση προς τον πλησιέστερο άρτιο δε δίνει σωστό αποτέλεσμα αφού χρειάζεται δύο επιπλέον ψηφία ενώ στην περίπτωση που μελετάμε υπάρχει διαθέσιμο μόνο ένα. Εάν θεωρήσουμε πως τα επιπλέον ψηφία είναι $|10$, τότε η διαφορά χρειάζεται στρογγυλοποίηση οδηγώντας σε λανθασμένο αποτέλεσμα.

Στην Εικόνα 3.6, η αφαίρεση πραγματοποιείται με 3 επιπλέον ψηφία, το ψηφίο φρουρός (G), το ψηφίο στρογγυλοποίησης (R) και το επίμονο ψηφίο (T). Όπως και πριν η διαφορά δε βρίσκεται σε κανονική μορφή και χρειάζεται αριστερή ολίσθηση κατά μια θέση. Έπειτα από την αριστερή ολίσθηση υπάρχουν διαθέσιμα δύο επιπλέον ψηφία και έτσι η στρογγυλοποίηση προς τον πλησιέστερο άρτιο πραγματοποιείται σωστά, σαν να ήταν διαθέσιμα και τα 6 επιπλέον ψηφία όπως στην Εικόνα 3.4.

6. Σημαίες Εξαιρέσεων (Exception Flags) και Ειδικές Τιμές:

	1.00000101100	000000
–	0.00000110000	010110
	0.1111111011	101010
Κανονικοποίηση	1.1111110111	01010
Στρογγυλοποίηση	1.1111110111	

Εικόνα 3.4: Στρογγυλοποίηση με όλα τα ψηφία.

	G	T		G	R	T	
	1.00000101100	0	0	1.00000101100	0	0	0
–	0.00000110000	0	1	– 0.00000110000	0	1	1
	0.1111111011	1	1	0.1111111011	1	0	1
Κανον.	1.1111110111	1		Κανον.	1.1111110111	0	1
Στρογγ.	1.1111110100			Στρογγ.	1.1111110111		

Εικόνα 3.5: Λανθασμένη στρογγυλοποίηση με ψηφία G, T.

Εικόνα 3.6: Ορθή στρογγυλοποίηση με ψηφία G, R, T.

Τέλος, αφού έχει ολοκληρωθεί η πράξη κινητής υποδιαστολής, το αποτέλεσμα ελέγχεται ώστε να διαπιστωθεί αν υπάρχουν σφάλματα ακρίβειας κατά την στρογγυλοποίηση. Σφάλμα ακρίβειας συμβαίνει όταν ισχύει : $G + R + T = 1$. Παράλληλα, γίνεται έλεγχος εάν έχει συμβεί υπερχείλιση (overflow) ή ανεπάρκεια αναπαράστασης (underflow), δηλαδή ο εκθέτης του αποτελέσματος είναι μεγαλύτερος από το μεγαλύτερο δυνατό εκθέτη ή μικρότερος από το μικρότερο δυνατό εκθέτη, αντίστοιχα. Επίσης διαπιστώνεται εάν το αποτέλεσμα της πράξης κινητής υποδιαστολής είναι όχι-αριθμός (Not-a-Number-NaN). Εάν κάποιος από τους τελεστέους είναι όχι-αριθμός (NaN) τότε το αποτέλεσμα είναι όχι-αριθμός (NaN). Ο Πίνακας 3.9 παρουσιάζει τα αποτελέσματα της πρόσθεσης και αφαίρεσης των ειδικών τιμών, αντίστοιχα.

	Πρόσθεση					Αφαίρεση			
+	$-\infty$	-0	$+0$	$+\infty$	–	$-\infty$	-0	$+0$	$+\infty$
$-\infty$	$-\infty$	$-\infty$	$-\infty$	NaN	$-\infty$	NaN	$-\infty$	$-\infty$	$-\infty$
-0	$-\infty$	-0	$\pm 0^*$	$+\infty$	-0	$+\infty$	$+0$	-0	$-\infty$
$+0$	$-\infty$	$\pm 0^*$	$+0$	$+\infty$	$+0$	$+\infty$	$+0$	$+0$	$-\infty$
$+\infty$	NaN	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	NaN

* -0 όταν στρογγυλοποίηση προς $-\infty$

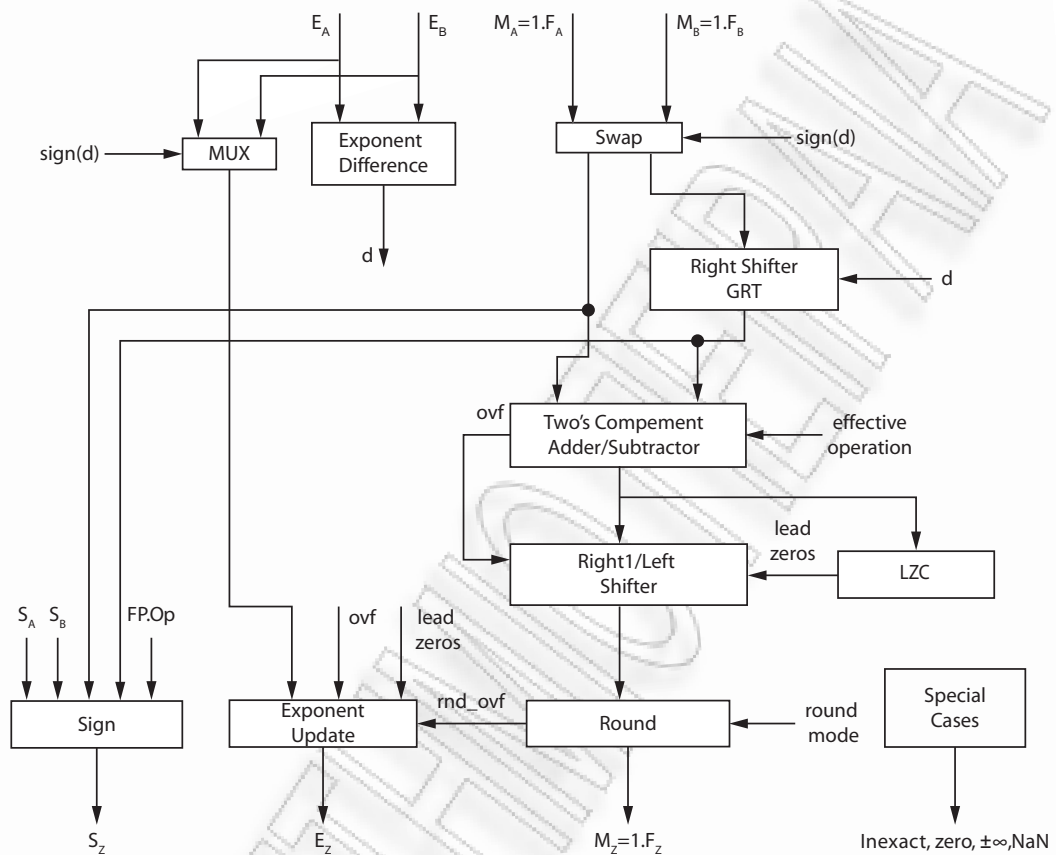
Πίνακας 3.9: Πρόσθεση/Αφαίρεση ειδικών τιμών.

3.4.1 Υλοποίηση Μονού Μονοπατιού για Αθροιστές Κινητής Υποδιαστολής

Ο αλγόριθμος πρόσθεσης/αφαίρεσης αριθμών κινητής υποδιαστολής, όπως παρουσιάστηκε στις προηγούμενες παραγράφους, οδηγεί στην απ' ευθείας υλοποίηση μονού μονοπατιού (*single path*). Ονομάζεται αρχιτεκτονική μονού μονοπατιού διότι μόνο ένα μονοπάτι δεδομένων χρησιμοποιείται για τον υπολογισμό του σημανόμενου του αποτελέσματος της πράξης κινητής υποδιαστολής, όπως φαίνεται και από την Εικόνα 3.7. Το πρόσημο της διαφοράς των εκθέτων χρησιμοποιείται ώστε να επιλεγεί ο μεγαλύτερος εκθέτης για περαιτέρω επεξεργασία. Αντίστοιχα πραγματοποιείται και η εναλλαγή. Η εναλλαγή μπορεί να περιλαμβάνει και τη σύγκριση των σημανόμενων ώστε εάν η πραγματική πράξη είναι αφαίρεση, η διαφορά να είναι πάντα θετικός αριθμός. Στη συνέχεια βάσει της διαφοράς, ο δεξιός ολισθητής (στην Εικόνα 3.7 : Right Shifter—GRT) πραγματοποιεί την ολίσθηση και δημιουργεί τα ψηφία φρουρούς, G , R , T . Η πρόσθεση/αφαίρεση των σημανόμενων πραγματοποιείται με ένα αθροιστή/αφαιρέτη με συμπλήρωμα ως προς 2. Ο εν λόγω αθροιστής υλοποιείται βάσει μιας πολύ γρήγορης αρχιτεκτονικής πρόβλεψης κρατουμένου, όπως για παράδειγμα των Carry Lookahead [97], Brent-Kung [98], Kogge-Stone [99], ή Sklansky [100]. Καθώς το αποτέλεσμα του αθροιστή/αφαιρέτη μπορεί να μην είναι κανονικοποιημένο μπορεί να χρειάζεται δεξιά ολίσθηση κατά μια θέση ή αριστερή ολίσθηση κατά τόσες θέσεις όσα είναι και τα αρχικά μηδενικά. Τα αρχικά μηδενικά υπολογίζονται από το μετρητή αρχικών μηδενικών (*leading zero counter*) (στην Εικόνα 3.7 : LZC). Ο μετρητής αρχικών μηδενικών υλοποιείται συνήθως από ένα κύκλωμα κωδικοποιητή με προτεραιότητα (*priority encoder*). Μια εναλλακτική σχεδίαση η οποία είναι ταχύτερη και καταλληλότερη για τεχνολογία VLSI παρουσιάζεται στην εργασία [101]. Έπειτα από την μετά-κανονικοποίηση (στην Εικόνα 3.7 : Right/Left Shifter) ακολουθεί η στρογγυλοποίηση (στην Εικόνα 3.7 : Round) ανάλογα βέβαια με την επιλογή της στρογγυλοποίησης. Κατά τη διαδικασία της μετά-κανονικοποίησης αλλά και της στρογγυλοποίησης ενημερώνεται και ο εκθέτης του αποτελέσματος (στην Εικόνα 3.7 : Exponent Update). Η πράξη ολοκληρώνεται με τον υπολογισμό του προσήμου του αποτελέσματος (στην Εικόνα 3.7 : Sign) και τον έλεγχο για ειδικές τιμές και σημαίες εξαιρέσεων (στην Εικόνα 3.7 : Special Cases).

3.4.2 Υλοποίηση Διπλού Μονοπατιού για Αθροιστές Κινητής Υποδιαστολής

Η υλοποίηση ενός αθροιστή/αφαιρέτη αριθμών κινητής υποδιαστολής βάσει της αρχιτεκτονικής του μονού μονοπατιού (*single-path*) δεν αποτελεί την καλύτερη επιλογή από πλευράς επίδοσης. Εξαιτίας του μεγάλου πλήθους των σειριακών βημάτων η αρχιτεκτονική μονού μονοπατιού οδηγεί σε μονάδες κινητής υποδιαστολής μειωμένης απόδοσης. Αξιοσημείωτο είναι πως ένας αθροιστής αριθμών κινητής υποδιαστολής χρειάζεται περίπου τον ίδιο χρόνο υπο-



Εικόνα 3.7: Πρόσθεση Κινητής Υποδιαστολής — Υλοποίηση Μονού Μονοπατιού.

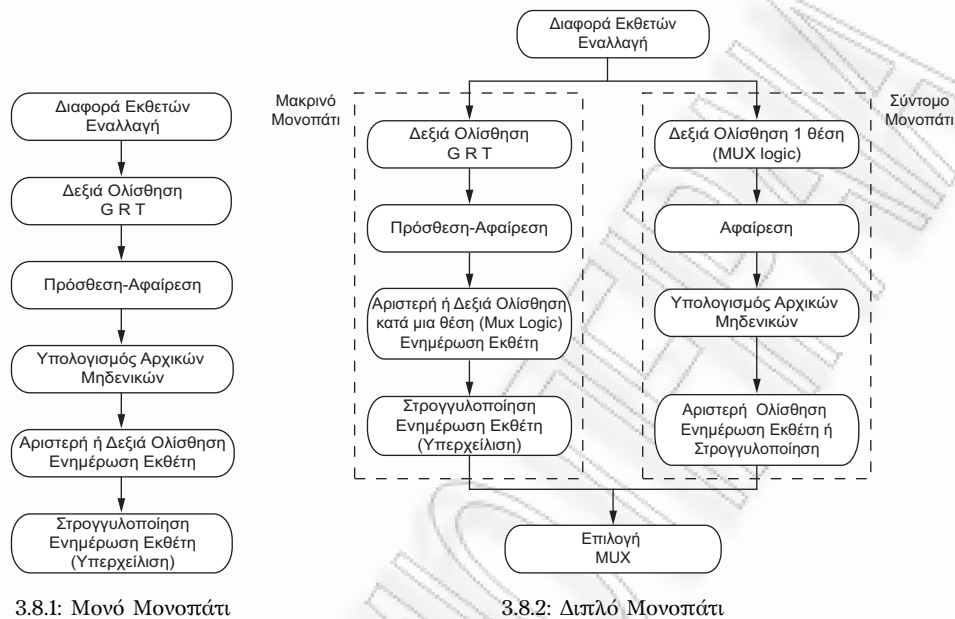
λογισμού με μία μονάδα πολλαπλασιασμού αριθμών κινητής υποδιαστολής, αντίθετα με ότι συμβαίνει για τις αντίστοιχες ακέραιες μονάδες, όπου οι πολλαπλασιαστές είναι σαφώς πιο αργό από τους απλούστερους αθροιστές. Η αιτία είναι πως τα περισσότερα βήματα για την πρόσθεση/αφαίρεση των αριθμών κινητής υποδιαστολής εκτελούνται ακολουθιακά – στην σειρά και όχι παράλληλα.

Σήμερα, σχεδόν όλες οι μονάδες κινητής υποδιαστολής υλοποιούν την πρόσθεση/αφαίρεση των αριθμών κινητής υποδιαστολής βάσει της αρχιτεκτονικής διπλού μονοπατιού (double path). Η αρχιτεκτονική διπλού μονοπατιού στοχεύει στην αύξηση της παραλληλίας των βημάτων για την πρόσθεση/αφαίρεση αριθμών κινητής υποδιαστολής. Η διαδικασία της πρόσθεσης/αφαίρεσης διαμερίζεται σε δύο ανεξάρτητα μεταξύ τους μονοπάτια, το μακρινό μονοπάτι (*far-path*) και το σύντομο μονοπάτι (*close-path*). Η αρχιτεκτονική παρουσιάστηκε από τον Farmwald [20] και βασίζεται στην ακόλουθη παρατήρηση. Όταν η πραγματική πράξη είναι αφαίρεση και η απόλυτη τιμή της διαφοράς των εκθετών είναι 0 ή 1, τότε δε χρειάζεται μονάδα ολίσθησης για προ-κανονικοποίηση. Σε κάθε άλλη περίπτωση και όταν το αποτέλεσμα της πρόσθεσης ή

αφαίρεσης χρειάζεται μετά-κανονικοποίηση, αυτή μπορεί να υλοποιηθεί με ένα πολυπλέκτη χωρίς να απαιτείται μονάδα ολίσθησης. Η λειτουργία των ανεξάρτητων μονοπατιών έχει ως εξής:

- Το μακρινό μονοπάτι (far-path) υπολογίζει το αποτέλεσμα της πρόσθεσης/αφαίρεσης αριθμών κινητής υποδιαστολής όταν $d = |E_A - E_B| > 1$. Όταν η απόλυτη τιμή της διαφοράς των εκθετών των αριθμών κινητής υποδιαστολής είναι μεγαλύτερη της μονάδας, $d = |E_A - E_B| > 1$, ανεξάρτητα από την πραγματική πράξη το αποτέλεσμα ή θα είναι κανονικοποιημένο, ή θα χρειάζεται δεξιά ολίσθηση κατά μια θέση, ή τέλος θα χρειάζεται αριστερή ολίσθηση κατά μια θέση μόνο. Με αυτό τον τρόπο, δε χρειάζεται ένας ολισθητής για την μετά-κανονικοποίηση και η λογική της κανονικοποίησης εκφυλίζεται σε λογική πολυπλέκτη (multiplexer). Επίσης, το αποτέλεσμα του ακέραιου αθροιστή/αφαιρέτη των σημειωμένων θα είναι πάντα θετικός αριθμός.
- Το σύντομο μονοπάτι (close-path) υπολογίζει το αποτέλεσμα της αφαίρεσης αριθμών κινητής υποδιαστολής όταν $d = |E_A - E_B| = 0$ ή $d = |E_A - E_B| = 1$. Όταν η απόλυτη τιμή της διαφοράς των εκθετών είναι 0, ή 1 και η πραγματική πράξη (effective operation) είναι αφαίρεση τότε το αποτέλεσμα της αφαίρεσης ή θα είναι κανονικοποιημένο ή θα χρειάζεται αριστερή ολίσθηση κατά δύο θέσεις τουλάχιστον. Με αυτό τον τρόπο χρειάζεται ένας ολισθητής πολλαπλών θέσεων για την μετά-κανονικοποίηση. Από την άλλη πλευρά δεν είναι απαραίτητος ένας ολισθητής για την προ-κανονικοποίηση αφού γίνεται η παραδοχή πως η διαφορά των εκθετών θα είναι είτε 0 είτε 1. Οπότε η προ-κανονικοποίηση μπορεί να πραγματοποιηθεί με λογική πολυπλέκτη. Το σύντομο μονοπάτι πραγματοποιεί την αφαίρεση των αριθμών, εικάζοντας πως η διαφορά των εκθετών είναι 0 ή 1 ελέγχοντας τα λιγότερο σημαντικά ψηφία των εκθετών.
- Στο τέλος ανάλογα με την τιμή της απόλυτης διαφοράς των εκθετών επιλέγεται το αποτέλεσμα είτε του μακρινού είτε του σύντομου μονοπατιού.

Στην Εικόνα 3.8 παρουσιάζονται οι εξαρτήσεις για τις αρχιτεκτονικές μονού και διπλού μονοπατιού [21]. Το στάδιο της προ-κανονικοποίησης απαιτεί την δεξιά ολίσθηση του σημειωμένου με το μικρότερο εκθέτη. Αντίθετα, στην αρχιτεκτονική του διπλού μονοπατιού μόνο το μακρινό μονοπάτι απαιτεί παρόμοια ολίσθηση ενώ στο σύντομο μονοπάτι δεν υπάρχει ολίσθηση. Στο σύντομο μονοπάτι η δεξιά ολίσθηση πραγματοποιείται πολύ εύκολα με ένα πολυπλέκτη (multiplexer). Η αρχιτεκτονική μονού μονοπατιού απαιτεί την αριστερή ολίσθηση του σημειωμένου κατά μία ή περισσότερες θέσεις ή τη δεξιά ολίσθηση κατά μια θέση όταν το σημειωμένο δεν είναι κανονικοποιημένο. Η ολίσθηση επιτυγχάνεται με ένα ολισθητή που υποστηρίζει και αριστερή και δεξιά ολίσθηση. Αντίθετα, στην αρχιτεκτονική του διπλού μονοπατιού αριστερή ολίσθηση κατά τουλάχιστον δύο θέσεις πραγματοποιείται στο σύντομο



3.8.1: Μονό Μονοπάτι

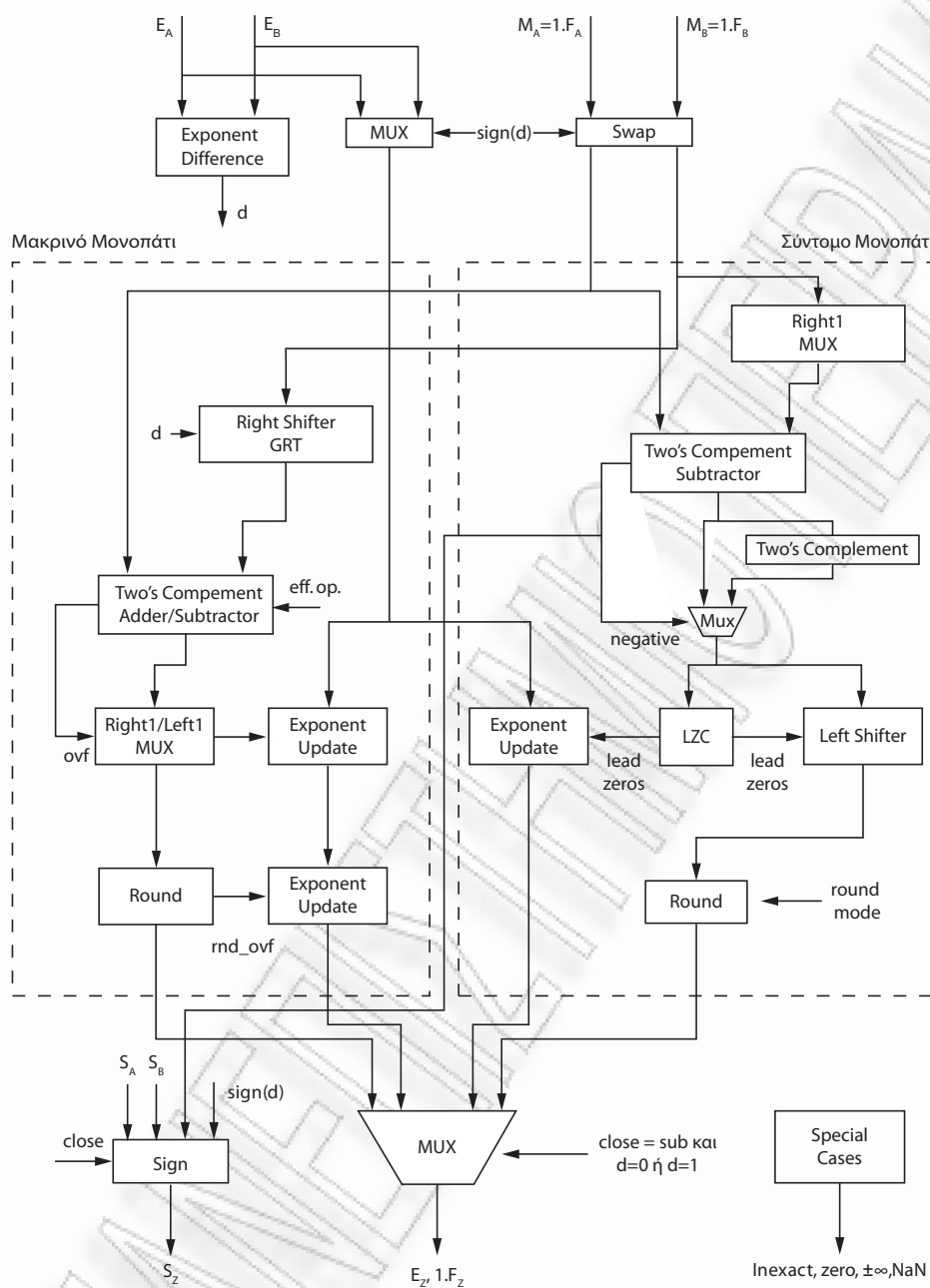
3.8.2: Διπλό Μονοπάτι

Εικόνα 3.8: Εξαρτήσεις στην Αρχιτεκτονική Μονού και Διπλού Μονοπατιού.

μονοπάτι. Από την άλλη πλευρά η μετά-κανονικοποίηση για το μακρινό μονοπάτι πραγματοποιείται χρησιμοποιώντας έναν πολυπλέκτη. Επίσης στο σύντομο μονοπάτι δεν υπάρχει περίπτωση υπερχείλισης κατά την στρογγυλοποίηση, με αποτέλεσμα την απλοποίηση του υλικού. Και τα δύο μονοπάτια έχουν σημαντικά μικρότερο μήκος/καθυστέρηση από το μονοπάτι στην αρχιτεκτονική μονού μονοπατιού και από αυτήν την ιδιότητα τους αντλούν τη μεγαλύτερη απόδοση—ταχύτητα.

Η αρχιτεκτονική ενός αθροιστή/αφαιρέτη αριθμών κινητής υποδιαστολής βασισμένη στην υλοποίηση του διπλού μονοπατιού παρουσιάζεται στην Εικόνα 3.9, [21].

Τα περισσότερα τμήματα της αρχιτεκτονικής διπλού μονοπατιού είναι παρόμοια με αυτά της αρχιτεκτονικής μονού μονοπατιού. Βέβαια υπάρχουν και διαφορές, όπως ότι στο σύντομο μονοπάτι το αποτέλεσμα της διαφοράς των σημαϊνόμενων μπορεί να είναι αρνητικό. Στην περίπτωση αυτή χρησιμοποιείται ένα κύκλωμα συμπληρώματος ως προς 2 (στην Εικόνα Two's Complement) ώστε να μετατρέψει το αρνητικό αποτέλεσμα σε θετικό. Διαφορετικός είναι και ο τρόπος με τον οποίο υπολογίζεται το πρόσημο. Το τελικό πρόσημο του αποτελέσματος της πράξης της κινητής υποδιαστολής είναι το πρόσημο του αριθμού με το μεγαλύτερο εκθέτη όταν σωστό αποτέλεσμα είναι αυτό του μακρινού μονοπατιού. Όταν το σωστό αποτέλεσμα είναι αυτό του σύντομου μονοπατιού η εύρεση του τελικού πρόσημου είναι πιο πολύπλοκη. Όταν οι εκθέτες είναι ίδιοι, τότε το πρόσημο είναι ίσο με $S_A \oplus C'_{out}$ (2's compl.), όπου C_{out} (2's compl.) είναι το κρατούμενο εξόδου του κυκλώματος που υπολογίζει το συμπλήρωμα ως προς 2. Εάν οι εκθέτες δεν είναι ίδιοι τότε το τελικό πρόσημο είναι το πρόσημο του αριθμού με το



Εικόνα 3.9: Πρόσθεση Κινητής Υποδιαστολής — Υλοποίηση Διπλού Μονοπατιού.

μεγαλύτερο εκθέτη. Εν τέλει, το πρόσημο του αποτελέσματος είναι ίσο με :

$$S_A + close(S_A \oplus C'_{out}(2's\ compl.))$$

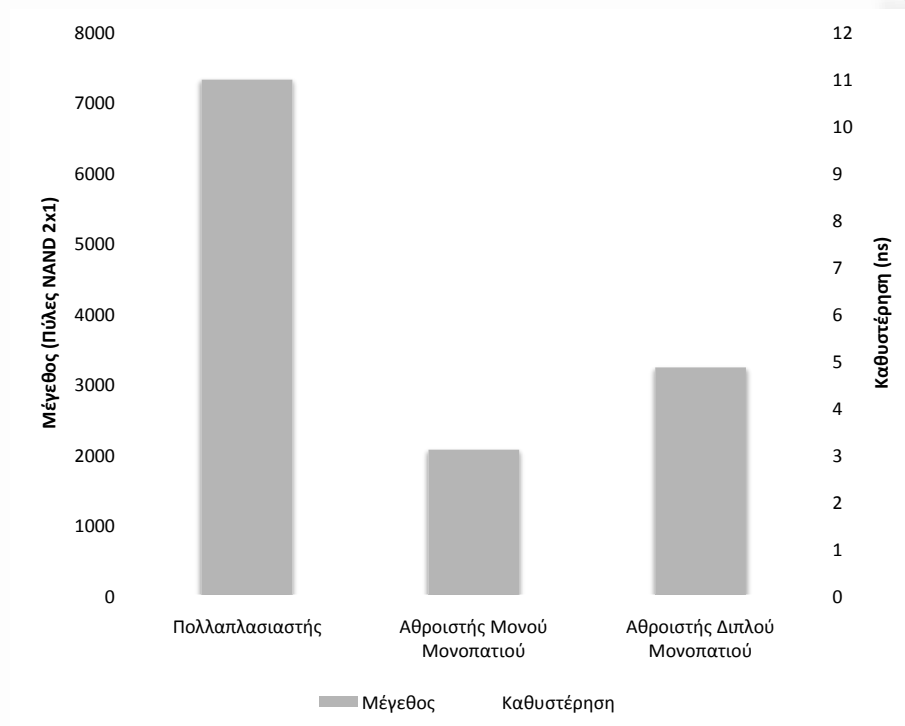
	Μέγεθος	Καθυστέρηση
Πολλαπλασιαστής	7310	11,74 ns
Αθροιστής Μονού Μονοπατιού	2057	10,83 ns
Αθροιστής Διπλού Μονοπατιού	3225	7,64 ns

Πίνακας 3.10: Μέγεθος και απόδοση των: πολλαπλασιαστή, αθροιστή μονού μονοπατιού και αθροιστή διπλού μονοπατιού.

Σύγκριση Υλοποιήσεων

Η υλοποίηση ενός αθροιστή/αφαιρέτη σύμφωνα με την αρχιτεκτονική υλοποίησης διπλού μονοπατιού είναι πιο αποδοτική από πλευράς ταχύτητας σε σχέση με την υλοποίηση μονού μονοπατιού. Η διαισθητική αυτή παρατήρηση επιβεβαιώνεται παρακάτω με την μέτρηση της απόδοσης τριών διαφορετικών μονάδων κινητής υποδιαστολής: του πολλαπλασιαστή, του αθροιστή μονού μονοπατιού και του αθροιστή διπλού μονοπατιού, όλων απλής ακρίβειας αναπαράστασης. Ο Πίνακας 3.10 παρουσιάζει το μέγεθος των κυκλωμάτων σε ισοδύναμες πύλες NAND δύο εισόδων (NAND2x1) καθώς και την καθυστέρηση (delay) τους σε nsec.

Η σύνθεση των κυκλωμάτων κινητής υποδιαστολής έγινε σε βιβλιοθήκη (design library) τεχνολογίας 0.18 μm της εταιρείας TSMC. Η Εικόνα 3.10 παρουσιάζει γραφικά τα δεδομένα του Πίνακα 3.10. Ενώ ο πολλαπλασιαστής κινητής υποδιαστολής είναι σαφώς μεγαλύτερος σε σχέση με τον αθροιστή μονού μονοπατιού, τα δύο κυκλώματα έχουν παρόμοια καθυστέρηση όπως φαίνεται και από τη μικρή κλίση της γραμμής της καθυστέρησης. Αντίθετα, ο αθροιστής διπλού μονοπατιού έχει σαφώς μικρότερη καθυστέρηση και από τον πολλαπλασιαστή και τον αντίστοιχο αθροιστή μονού μονοπατιού με επιπλέον κόστος σε υλικό το οποίο είναι λιγότερο από 1000 πύλες NAND2x1. Συνοψίζοντας, ο αθροιστής κινητής υποδιαστολής υλοποιημένος βάσει της αρχιτεκτονικής δύο μονοπατιών αυξάνει την απόδοση περίπου κατά 30% για την αναπαράσταση απλής ακρίβειας.



Εικόνα 3.10: Γραφική αναπαράσταση των χαρακτηριστικών του μεγέθους και της καθυστέρησης.

3.5 Πολλαπλασιασμός

Η πράξη του πολλαπλασιασμού δύο αριθμών κινητής υποδιαστολής, σε αντίθεση με την πράξη της πρόσθεσης, είναι απλούστερη καθώς αποτελείται από λιγότερα βήματα και ταυτόχρονα τα βήματα αυτά είναι χαμηλότερης πολυπλοκότητας από τα αντίστοιχα της πρόσθεσης. Το παράδειγμα που ακολουθεί παρουσιάζει τα βασικά χαρακτηριστικά του πολλαπλασιασμού κινητής υποδιαστολής.

Παράδειγμα: Έστω οι αριθμοί $0.5_{10} = 1.000 \times 2^{-1}$ και $-0.4375_{10} = -1.110 \times 2^{-2}$ τους οποίους θέλουμε να πολλαπλασιάσουμε. Για την αναπαράσταση του σημειόμενου δεσμεύονται συνολικά 4 ψηφία. Οι αριθμοί είναι ετερόσημοι άρα το γινόμενο τους είναι αρνητικό. Στον πολλαπλασιασμό οι εκθέτες των αριθμών κινητής υποδιαστολής προστίθενται, άρα ο τελικός εκθέτης είναι -3 (Η περίπτωση της πολωμένης σημειογραφίας εξετάζεται στην συνέχεια). Ακολουθεί ο πολλαπλασιασμός των σημαινόμενων. Το γινόμενο των σημαινόμενων όπως αυτό παράγεται από τον ακέραιο πολλαπλασιαστή είναι ίσο με 01110000 και αποτελείται από 8 ψηφία. Η υποδιαστολή τοποθετείται 6 θέσεις μετρώντας από δεξιά, άρα το γινόμενο είναι ίσο με 01.110000. Το σημαντικότερο ψηφίο του γινομένου είναι 0 και αυτό σημαίνει πως το γινόμενο βρίσκεται σε κανονική μορφή. Εάν ήταν 1 τότε το γινόμενο δε θα βρισκονταν σε κανο-

νική μορφή και θα χρειάζονταν δεξιά ολίσθηση κατά μία θέση. Το γινόμενο των σημειωμένων, 1.110|000 θα πρέπει να αναπαρασταθεί με 4 ψηφία. Τα ψηφία που βρίσκονται δεξιά του συμβόλου | θα χρησιμοποιηθούν κατά την στρογγυλοποίηση (ανάλογα με τον τρόπο στρογγυλοποίησης). Στο παράδειγμα, δεν είναι απαραίτητη η στρογγυλοποίηση καθώς όλα τα ψηφία δεξιά του συμβόλου | είναι μηδενικά. Επομένως,

$$1.000 \times 2^{-1} \times (-1.110) \times 2^{-2} = -1.110 \times 2^{-3} = -0.21875_{10}.$$

Γενικά, τα βήματα για τον πολλαπλασιασμό δύο αριθμών κινητής υποδιαστολής A , B οι οποίοι αναπαρίστανται από τις τριπλέτες, (S_A, E_A, F_A) και (S_B, E_B, F_B) , αντίστοιχα έχουν ως εξής:

1. Πρόσθεση των εκθετών:

Οι εκθέτες των αριθμών κινητής υποδιαστολής A , B αθροίζονται για να σχηματίσουν τον τελικό εκθέτη. Επίσης από το άθροισμα θα πρέπει να αφαιρεθεί η πόλωση (bias) του εκθέτη διότι κατά την πρόσθεση των εκθετών η πόλωση έχει αθροισθεί δύο φορές. Έτσι, ο τελικός εκθέτης είναι:

$$(E_A + B) + (E_B + B) - B = E_A + E_B - B.$$

2. Πρόσημο:

Όπως και στον πολλαπλασιασμό των ακεραίων αριθμών, έτσι και στον πολλαπλασιασμό των αριθμών κινητής υποδιαστολής το πρόσημο του γινομένου είναι θετικό όταν οι αριθμοί είναι ομόσημοι ενώ είναι αρνητικό όταν οι αριθμοί είναι ετερόσημοι. Το πρόσημο του γινομένου είναι:

$$S_A \oplus S_B,$$

όπου \oplus η λογική πράξη αποκλειστικό-Η (XOR).

3. Πολλαπλασιασμός των σημειωμένων:

Τα σημειωμένα των αριθμών κινητής υποδιαστολής πολλαπλασιάζονται χρησιμοποιώντας έναν ακεραίο πολλαπλασιαστή για να σχηματίσουν το νέο σημειωμένο. Καθώς τα σημειωμένα έχουν μέγεθος $m = f + 1$ ψηφίων το γινόμενο τους παράγει ένα αριθμό με $2 \times m = 2 \times (f + 1)$ ψηφία συνολικά. Βέβαια από αυτά τα ψηφία μόνο τα $f + 1$ θα χρησιμοποιηθούν για το τελικό σημειωμένο.

Τα παραπάνω βήματα είναι ανεξάρτητα μεταξύ τους και πραγματοποιούνται παράλληλα, αντίθετα απ' ό τι συμβαίνει στην πρόσθεση. Επίσης δεν υπάρχει το στάδιο της προκανονικοποίησης.

4. Μετά-Κανονικοποίηση (Post Normalization):

Το γινόμενο που προκύπτει από τον πολλαπλασιασμό των σημειωμένων μπορεί να μην

είναι κανονικοποιημένο. Αυτό συμβαίνει διότι για τους αριθμούς κινητής υποδιαστολής A, B ισχύει:

$$A \in [1, 2) \text{ και } B \in [1, 2) \Rightarrow A \times B \in [1, 4).$$

Όταν το περισσότερο σημαντικό ψηφίο (most significant bit) του γινομένου είναι 1, τότε το γινόμενο δε βρίσκεται σε κανονική μορφή. Οι ενέργειες που απαιτούνται ώστε το γινόμενο να κανονικοποιηθεί είναι οι ακόλουθες:

- Δεξιά ολίσθηση του γινομένου κατά μια θέση. Στην περίπτωση αυτή το λιγότερο σημαντικό ψηφίο δεν αποκόπτεται αλλά συνυπολογίζεται μέσω της πράξης λογικό-Η με το ψηφίο που βρίσκεται αριστερά του.
- Προσαύξηση του εκθέτη κατά μια μονάδα.

Η στρογγυλοποίηση προς τον πλησιέστερο άρτιο αριθμό απαιτεί δύο επιπλέον ψηφία. Έτσι, από τα συνολικά $2 \times m$ ψηφία του γινομένου θα χρησιμοποιηθούν $m + 2$ ψηφία. Τα ψηφία στρογγυλοποίησης που απαιτούνται είναι το ψηφίο φρουρός (Guard bit) και το επίμονο ψηφίο (Sticky bit) το οποίο είναι το αποτέλεσμα της πράξης λογικό-Η όλων των ψηφίων δεξιότερα του ψηφίου φρουρός. Έστω πως το γινόμενο, P , των σημαίνωμενων πριν από την μετά-κανονικοποίηση είναι:

$$P1P0.P_0P_1 \dots P_{m-2}P_{m-1}P_mP_{m+1} \dots P_{2m-2}.$$

Εάν $P1 = 0$ το γινόμενο είναι κανονικοποιημένο, επομένως το κανονικοποιημένο γινόμενο $P_{\text{normalized}}$ είναι:

$$P_{\text{normalized}} = P0.P_0P_1 \dots P_{m-2}L|GT,$$

όπου L το λιγότερο σημαντικό ψηφίο, $L = P_{m-1}$, G το ψηφίο φρουρός, $G = P_m$, και T το επίμονο ψηφίο, $T = P_{m+1} + P_{m+2} + \dots + P_{2m-2}$ (+ η πράξη λογικό-Η).

Εάν $P1 = 1$ τότε το γινόμενο είναι μεγαλύτερο του 2 και έπειτα από την κανονικοποίηση το νέο γινόμενο είναι :

$$P_{\text{normalized}} = P1.P0P_0P_1 \dots P_{m-3}L|GT,$$

όπου L είναι λιγότερο σημαντικό ψηφίο, $L = P_{m-2}$, G είναι το ψηφίο φρουρός, $G = P_{m-1}$, και T είναι το επίμονο ψηφίο, $T = P_m + P_{m+1} + \dots + P_{2m-2}$.

5. Στρογγυλοποίηση (Rounding)

Η στρογγυλοποίηση προς τον πλησιέστερο άρτιο αριθμό απαιτεί δύο επιπλέον ψηφία στρογγυλοποίησης ενώ η στρογγυλοποίηση προς το $\pm\infty$ μόνο ένα. Έστω πως το κανονικοποιημένο γινόμενο είναι:

$$1.F_0F_1 \dots F_{f-2}L|GT,$$

\times	$-\infty$	-0	$+0$	$+\infty$
$-\infty$	$+\infty$	NaN	NaN	$-\infty$
-0	NaN	$+0$	-0	NaN
$+0$	NaN	-0	$+0$	NaN
$+\infty$	$-\infty$	NaN	NaN	$+\infty$

Πίνακας 3.11: Ειδικές Τιμές και πολλαπλασιασμός κινητής υποδιαστολής.

όπου L το λιγότερο σημαντικό ψηφίο και τα ψηφία δεξιά του $|$ είναι το ψηφίο φρουρός και το επίμονο ψηφίο.

Στρογγυλοποίηση προς τον πλησιέστερο άρτιο: Πραγματοποιείται προσθέτοντας τη μονάδα στο ψηφίο L όταν:

$$GT + GT'L = G(T + L) = 1$$

Στρογγυλοποίηση προς το 0: Κατά την αποκοπή τα ψηφία G, T απορρίπτονται.

Στρογγυλοποίηση προς $+\infty$: Πραγματοποιείται προσθέτοντας τη μονάδα στο ψηφίο L όταν:

$$S'(G + T)$$

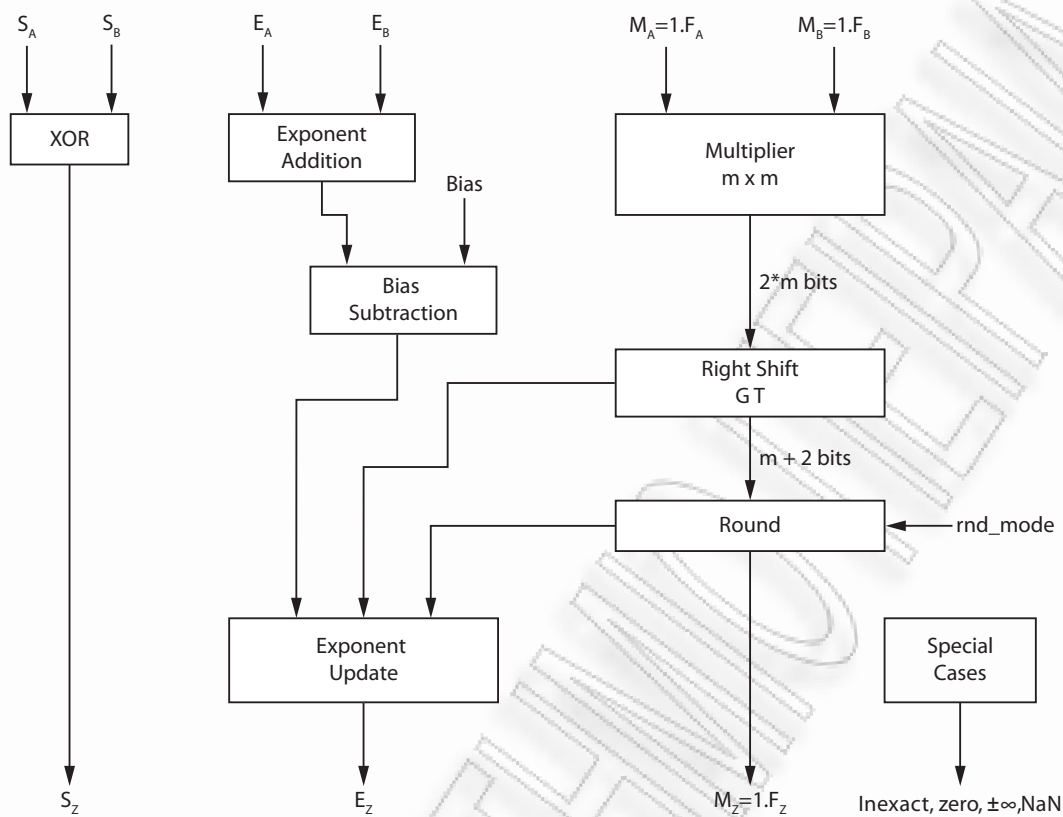
Στρογγυλοποίηση προς $-\infty$: Πραγματοποιείται προσθέτοντας τη μονάδα στο ψηφίο L όταν:

$$S(G + T)$$

6. Σημαίες Εξαιρέσεων (Exception Flags) και Ειδικές Τιμές:

Τέλος, το αποτέλεσμα του πολλαπλασιασμού κινητής υποδιαστολής ελέγχεται ώστε να διαπιστωθεί εάν υπάρχει σφάλμα κατά την στρογγυλοποίηση. Σφάλμα στρογγυλοποίησης έχει συμβεί όταν $G+T = 1$. Παράλληλα γίνεται έλεγχος για υπερχείλιση ή ανεπάρκεια αναπαράστασης του τελικού αποτελέσματος. Οι ενέργειες είναι αντίστοιχες με αυτές του ελέγχου τους αποτελέσματος της πρόσθεσης-αφαίρεσης. Ο Πίνακας 3.11 παρουσιάζει το αποτέλεσμα του πολλαπλασιασμού κινητής υποδιαστολής για τις ειδικές τιμές. Επειδή ο πολλαπλασιασμός μεταξύ του ± 0 και του $\pm \infty$ δεν ορίζεται το αποτέλεσμα της πράξης είναι όχι-αριθμός (NaN). Επίσης αν ένας ή και οι δυο τελεστέοι είναι όχι-αριθμοί (NaN) τότε το αποτέλεσμα είναι όχι-αριθμός (NaN).

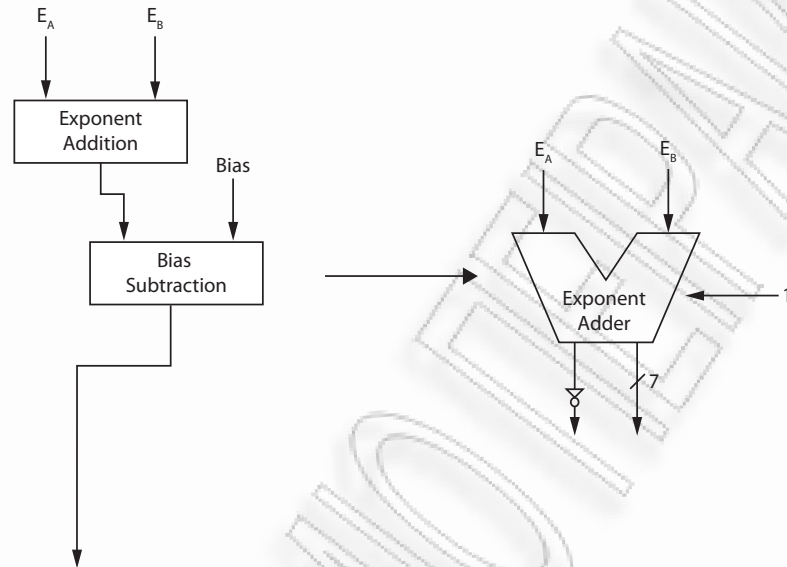
Η Εικόνα 3.11 παρουσιάζει την υλοποίηση του πολλαπλασιασμού αριθμών κινητής υποδιαστολής. Όπως φαίνεται ο υπολογισμός του γινομένου των σημειωμένων, του εκθέτη αλλά και του τελικού πρόσημου πραγματοποιούνται παράλληλα, δίχως αλληλεξαρτήσεις. Το τελικό πρόσημο υπολογίζεται πολύ εύκολα από το αποτέλεσμα της πράξης XOR των δύο



Εικόνα 3.11: Υλοποίηση Πολλαπλασιασμού κινητής υποδιαστολής.

προσέμων. Ο ακέραιος πολλαπλασιαστής υλοποιείται συνήθως βάσει κάποιας γρήγορης παράλληλης αρχιτεκτονικής, π.χ. Carry Save, Booth-Wallace, κυρίως για λόγους υψηλής απόδοσης. Σε περιπτώσεις όπου το μέγεθος του κυκλώματος έχει μεγαλύτερη σημασία από αυτή της ταχύτητας τότε μπορεί να χρησιμοποιηθεί ένας σειριακός ακέραιος πολλαπλασιαστής. Ο υπολογισμός του εκθέτη απαιτεί την πρόσθεση των δύο επιμέρους εκθετών των αριθμών κινητής υποδιαστολής και στην συνέχεια την αφαίρεση της πόλωσης από αυτό το άθροισμα. Υπενθυμίζεται πως η πόλωση (Bias) είναι $B = 127$ για την αναπαράσταση απλής ακρίβειας και $B = 1023$ για την αναπαράσταση διπλής ακρίβειας (βλ. Πίνακα 3.6). Η αφαίρεση της πόλωσης από το άθροισμα των εκθετών είναι ισοδύναμη με την πρόσθεση της μονάδας στο άθροισμα των εκθετών και την αντιστροφή του περισσότερο σημαντικού ψηφίου (Most Significant Bit) του αθροίσματος. Έτσι, στην αναπαράσταση απλής ακρίβειας, αντί να αφαιρεθεί η πόλωση $B = 127$ από το άθροισμα, προστίθεται η μονάδα και έπειτα αφαιρείται $B + 1 = 128$. Επειδή, το συμπλήρωμα ως προς 2 του 128 είναι ο εαυτός του, το περισσότερο σημαντικό ψηφίο του αθροίσματος των εκθετών αντιστρέφεται. Η Εικόνα 3.12 δείχνει την μετατροπή του μονοπατιού δεδομένων (datapath) για τον υπολογισμό του εκθέτη. Αντίστοιχη είναι η μετατροπή για

τον πολλαπλασιαστή αριθμών κινητής υποδιαστολής διπλής ακρίβειας.



Εικόνα 3.12: Βελτιστοποιημένος υπολογισμός του αθροίσματος των εκθετών (σε πολωμένη μορφή).

3.6 Διαίρεση

Η πράξη της διαίρεσης των αριθμών κινητής υποδιαστολής αποτελείται από παρόμοια βήματα με αυτά του πολλαπλασιασμού. Η πολυπλοκότητα της όμως είναι μεγαλύτερη εξαιτίας της υψηλής πολυπλοκότητας της ακέραιας διαίρεσης. Η διαίρεση δύο αριθμών κινητής υποδιαστολής παράγει το πηλίκο (quotient) και όχι το υπόλοιπο (remainder). Το υπόλοιπο της διαίρεσης συνήθως υπολογίζεται ως η διαφορά του γινομένου του διαιρέτη και του πηλίκου από τον διαιρετέο. Τα βήματα για τη διαίρεση δύο αριθμών κινητής υποδιαστολής A , B οι οποίοι αναπαρίστανται από τις τριπλέτες (S_A, E_A, F_A) , και (S_B, E_B, F_B) είναι τα ακόλουθα.

1. Αφαίρεση των εκθετών:

Οι εκθέτες των αριθμών κινητής υποδιαστολής A , B αφαιρούνται ώστε να σχηματίσουν τον τελικό εκθέτη. Επίσης στη διαφορά των εκθετών θα πρέπει να προστεθεί η πόλωση (bias) του εκθέτη διότι κατά την αφαίρεση των εκθετών η πόλωση έχει αφαιρεθεί δύο φορές. Έτσι, ο τελικός εκθέτης είναι:

$$(E_A - B) + (E_B - B) + B = E_A - E_B + B.$$

2. Πρόσημο:

Όπως στον πολλαπλασιασμό των αριθμών κινητής υποδιαστολής έτσι και στη διαίρεση,

το πρόσημο του πηλίκου γινομένου είναι θετικό όταν οι αριθμοί είναι ομόσημοι ενώ είναι αρνητικό όταν οι αριθμοί είναι ετερόσημοι. Το πρόσημο του πηλίκου είναι:

$$S_A \oplus S_B.$$

3. Διαίρεση των σημαινόμενων:

Τα σημαινόμενα των αριθμών κινητής υποδιαστολής διαιρούνται χρησιμοποιώντας έναν ακέραιο διαιρέτη ο οποίος είναι υλοποιημένος είτε με βάση τεχνικής σύγκλισης (convergence division) είτε με βάση τεχνικές επανάληψης ψηφίου (digit recurrence division). Δύο πολύ γνωστές τεχνικές διαίρεσης με επανάληψη ψηφίου είναι η διαίρεση με επαναφορά (restoring division) και η διαίρεση δίχως επαναφορά (non-restoring division). Αυτοί οι τρόποι διαίρεσης των ακεραίων μπορούν να χρησιμοποιηθούν για τη διαίρεση των σημαινόμενων με κάποια παραλλαγή. Στη διαίρεση των ακεραίων ο διαιρετέος έχει διπλάσιο πλήθος ψηφίων από το διαιρέτη. Η διαίρεση των ακεραίων ξεκινά με την ολίσθηση του διαιρετέου κατά μια θέση αριστερά και συνεχίζει με την αφαίρεση του διαιρέτη από τα υψηλά ψηφία του διαιρετέου. Η τροποποίηση που χρειάζεται για τη διαίρεση των σημαινόμενων είναι η ακόλουθη. Ο διαιρετέος δεν καταλαμβάνει διπλάσιο πλήθος ψηφίων σε σχέση με το διαιρέτη. Επιπλέον, ο διαιρετέος δεν χρειάζεται να ολισθήσει κατά μια θέση αριστερά (μόνο όμως κατά την πρώτη επανάληψη). Στη συνέχεια, ο διαιρέτης αφαιρείται από τα υψηλά ψηφία του διαιρετέου όπως και στην κλασική διαίρεση ακεραίων. Η διαίρεση κατ' αυτόν τον τρόπο παράγει ένα πηλίκο της μορφής $q_0.q_1q_2 \dots$ [102].

Η διαίρεση των σημαινόμενων πραγματοποιείται με δύο επιπλέον ψηφία, ώστε το πηλίκο να διαθέτει δύο επιπλέον ψηφία ακρίβειας, απαραίτητα για τη στρογγυλοποίηση. Το υπόλοιπο της διαίρεσης χρησιμοποιείται ώστε να σχηματίσει το επίμονο ψηφίο.

4. Μετά-Κανονικοποίηση (Post Normalization):

Το πηλίκο που προκύπτει από τη διαίρεση των σημαινόμενων μπορεί να μην είναι κανονικοποιημένο. Αυτό συμβαίνει διότι για τους αριθμούς κινητής υποδιαστολής A και B ισχύει:

$$A \in [1, 2) \text{ και } B \in [1, 2) \Rightarrow A \div B \in (1/2, 2).$$

Όταν το περισσότερο σημαντικό ψηφίο (most significant bit) του πηλίκου είναι 0, τότε το πηλίκο δεν βρίσκεται σε κανονική μορφή. Οι ενέργειες που απαιτούνται ώστε το γινόμενο να κανονικοποιηθεί είναι οι ακόλουθες:

- Αριστερή ολίσθηση του πηλίκου κατά μια θέση.
- Μείωση του εκθέτη κατά μια μονάδα.

5. Στρογγυλοποίηση:

Το κανονικοποιημένο πηλίκο στρογγυλοποιείται βάσει του επιλεγμένου τρόπου στρογγυλοποίησης. Η περίπτωση της στρογγυλοποίησης του πηλίκου προς τον πλησιέστερο

\div	$-\infty$	-0	$+0$	$+\infty$
$-\infty$	NaN	$+\infty$	$-\infty$	NaN
-0	$+\infty$	NaN	NaN	$-\infty$
$+0$	$-\infty$	NaN	NaN	$+\infty$
$+\infty$	NaN	$-\infty$	$+\infty$	NaN

Πίνακας 3.12: Ειδικές Τιμές και διαίρεση κινητής υποδιαστολής.

άρτιο δεν υφίσταται και η στρογγυλοποίηση προς τον πλησιέστερο άρτιο στην πραγματικότητα σημαίνει στρογγυλοποίηση προς τον πλησιέστερο αριθμό.

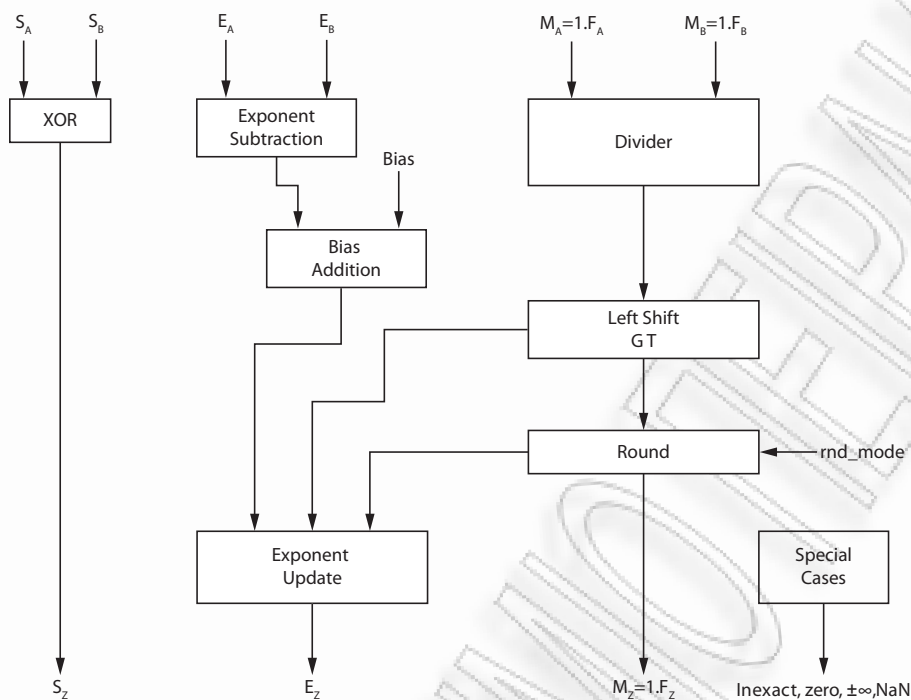
6. Σημαίες Εξαίρεσεων (Exception Flags) και Ειδικές Τιμές:

Το αποτέλεσμα της διαίρεσης υποδιαστολής ελέγχεται ώστε να διαπιστωθεί εάν υπάρχει σφάλμα κατά την στρογγυλοποίηση. Σφάλμα στρογγυλοποίησης έχει συμβεί όταν το υπόλοιπο της διαίρεσης είναι διαφορετικό του μηδέν. Παράλληλα γίνεται έλεγχος για υπερχείλιση ή ανεπάρκεια αναπαράστασης του τελικού αποτελέσματος. Οι ενέργειες είναι αντίστοιχες με αυτές του ελέγχου τους αποτελέσματος της πρόσθεσης/αφαίρεσης. Ο Πίνακας 3.12 παρουσιάζει το αποτέλεσμα της διαίρεσης κινητής υποδιαστολής για τις ειδικές τιμές. Εάν ένας ή και οι δυο τελεστέοι είναι όχι-αριθμοί (NaN) τότε το αποτέλεσμα είναι όχι-αριθμός (NaN). Μια σημαντική διαφορά μεταξύ της ακέραιας διαίρεσης και της διαίρεσης κινητής υποδιαστολής είναι η διαίρεση με το 0, όπου ενώ στην ακέραια διαίρεση συνήθως γεννιέται κάποιου είδους εξαίρεση ή διακοπή στην διαίρεση κινητής υποδιαστολής η διαίρεση με το 0 δίνει αποτέλεσμα $\pm\infty$. Εξαίρεση αποτελεί η διαίρεση του 0 με το 0 όπου η πράξη δεν ορίζεται και το αποτέλεσμα είναι όχι-αριθμός (NaN). Επίσης όχι-αριθμός είναι και το αποτέλεσμα της διαίρεσης του $\pm\infty$ με το $\pm\infty$.

Η Εικόνα 3.13 παρουσιάζει την υλοποίηση ενός διαιρέτη κινητής υποδιαστολής. Η αφαίρεση των εκθετών με συμπλήρωμα ως προς 2 και η πρόσθεση της πόλωσης μπορεί να πραγματοποιηθεί με ένα μόνο αθροιστή. Επειδή η πρόσθεση της πόλωσης, 127 για απλή ακρίβεια είναι αρκετά πολύπλοκη, αρκεί στη διαφορά των εκθετών να προστεθεί 128 και να αφαιρεθεί 1. Επειδή η αφαίρεση των εκθετών γίνεται με συμπλήρωμα ως προς 2, το κρατούμενο εισόδου αναιρείται από τη μονάδα που θα αφαιρεθεί. Η πρόσθεση με το 128 ισοδυναμεί με αντιστροφή του περισσότερου σημαντικού ψηφίου.

Παράδειγμα: Έστω οι αριθμοί κινητής υποδιαστολής σε αναπαράσταση απλής ακρίβειας, $1.00\dots00 \times 2^{128} = 2_{10}$ και $1.010\dots00 \times 2^{129} = 5_{10}$ τους οποίους θέλουμε να διαιρέσουμε. Οι εκθέτες βρίσκονται σε πολωμένη σημειογραφία, επομένως ο εκθέτης του πηλίκου σε πολωμένη σημειογραφία είναι $128-129+127=126$ ή αλλιώς -1 . Το πηλίκο της διαίρεσης του 2 με το 5 είναι:

$$0.11001100110011001100110110||011,$$



Εικόνα 3.13: Διάρθρωση κινητής υποδιαστολής.

όπου τα ψηφία μετά το σύμβολο $\|$ αντιστοιχούν στα ψηφία G, R, T . Το πηλίκο δεν βρίσκεται σε κανονική μορφή επομένως χρειάζεται αριστερή ολίσθηση κατά μια θέση. Ταυτόχρονα, αφαιρείται μια μονάδα από τον εκθέτη και ο νέος εκθέτης είναι 125 σε πολωμένη αναπαράσταση ή αλλιώς -2. Το κανονικοποιημένο πηλίκο είναι:

$$1.10011001100110011001100\|11.$$

Το επόμενο βήμα είναι η στρογγυλοποίηση του πηλίκου. Η στρογγυλοποίηση προς τον πλησιέστερο άρτιο ορίζει πως πρέπει να προστεθεί μία μονάδα στο λιγότερο σημαντικό ψηφίο του πηλίκου, δηλαδή στο 0 αριστερά του $\|$. Επομένως, το στρογγυλοποιημένο πηλίκο σε απλή αναπαράσταση είναι $1.1001100110011001101 \times 2^{125} = 0.400000005960464_{10}$.

3.7 Τετραγωνική Ρίζα

Η τελευταία πράξη κινητής υποδιαστολής που ορίζει το πρότυπο IEEE 754 είναι ο υπολογισμός της τετραγωνικής ρίζας ενός θετικού αριθμού.

Τα βήματα για τον υπολογισμό της τετραγωνικής ρίζας ενός θετικού αριθμού κινητής υποδιαστολής που αναπαρίσταται με την τριάδα (S_A, E_A, F_A) είναι τα ακόλουθα:

1. Υπολογισμός του νέου εκθέτη:

Ο νέος εκθέτης του αποτελέσματος της τετραγωνικής ρίζας του αριθμού κινητής υποδιαστολής είναι το αποτέλεσμα της παράστασης $\sqrt{2^{E_A}} = 2^{E_A/2}$, όπου E_A είναι η πολωμένη αναπαράσταση του εκθέτη. Η διαίρεση του εκθέτη με 2 σημαίνει ολίσθηση κατά μια θέση δεξιά. Αν ο εκθέτης είναι άρτιος τότε το λιγότερο σημαντικό ψηφίο είναι 0 οπότε δεν υπάρχει πρόβλημα κατά την ολίσθηση. Αντίθετα, αν ο εκθέτης είναι περιττός τότε το λιγότερο σημαντικό ψηφίο του είναι 1 και κατά την δεξιά ολίσθηση το ψηφίο θα χαθεί. Επομένως όταν ο εκθέτης είναι περιττός, μετατρέπεται σε άρτιο αφαιρώντας μια μονάδα (1) και το το σημαϊνόμενο του αριθμού κινητής υποδιαστολής ολισθαίνει κατά μία θέση αριστερά, διότι:

$$\sqrt{1.F_A \times 2^{E_A}} = \sqrt{1.F_A} \times 2^{E_A/2} = \sqrt{1.F_A} \times 2^{(E_A-1)/2} \times 2^{1/2} = \sqrt{1.F_A \times 2} \times 2^{(E_A-1)/2}$$

2. Υπολογισμός της τετραγωνικής ρίζας του νέου σημαϊνόμενου:

Στο στάδιο αυτό υπολογίζεται η τετραγωνική ρίζα του νέου σημαϊνόμενου του αριθμού κινητής υποδιαστολής. Ο υπολογισμός της τετραγωνικής ρίζας μπορεί να γίνει σύμφωνα με τεχνικές επανάληψης (recurrence), με επαναφορά (restoring), δίχως επαναφορά (non-restoring) όπως η SRT ή με τεχνικές σύγκλισης (convergence). Το αποτέλεσμα της τετραγωνικής ρίζας του σημαϊνόμενου βρίσκεται πάντοτε σε κανονική μορφή, οπότε δεν είναι απαραίτητο κάποιο στάδιο μετά-κανονικοποίησης. Όταν ο εκθέτης του αριθμού κινητής υποδιαστολής είναι άρτιος τότε $1.F_A \in [1, 2)$, ενώ όταν είναι περιττός $2 \times 1.F_A \in [2, 4)$ (αφού όταν ο εκθέτης είναι περιττός είναι απαραίτητο να ολισθήσει το σημαϊνόμενο κατά μία θέση αριστερά). Επομένως, για οποιαδήποτε τιμή του εκθέτη θα ισχύει

$$1 \leq 1.F_A < 4 \Rightarrow 1 \leq \sqrt{1.F_A} < 2,$$

οπότε η ρίζα βρίσκεται σε κανονική μορφή.

Το στάδιο του υπολογισμού της τετραγωνικής ρίζας του σημαϊνόμενου και του υπολογισμού του εκθέτη πραγματοποιούνται παράλληλα. Ο υπολογισμός της τετραγωνικής ρίζας χρειάζεται να γνωρίζει μόνο εάν ο εκθέτης είναι άρτιος ή περιττός, πληροφορία που είναι άμεσα διαθέσιμη.

3. Στρογγυλοποίηση:

Η στρογγυλοποίηση της τετραγωνικής ρίζας απαιτεί την ύπαρξη ενός επιπλέον ψηφίου, το οποίο παίζει το ρόλο του ψηφίου φρουρού (guard bit). Επιπλέον το υπόλοιπο της τετραγωνικής ρίζας² χρησιμοποιείται για να σχηματίσει το επίμονο ψηφίο (sticky bit). Το επίμονο ψηφίο είναι απαραίτητο για την στρογγυλοποίηση προς το $\pm\infty$ και για να διαπιστωθεί αν το αποτέλεσμα είναι ακριβές ή όχι ώστε να ενεργοποιηθεί η αντίστοιχη

² Η βασική εξίσωση της τετραγωνικής ρίζας είναι $z = q^2 + s$, όπου q η τετραγωνική ρίζα και s , το υπόλοιπο ($s = z - q^2$).

σημαία. Αντίθετα, το επίμονο ψηφίο δε χρησιμοποιείται κατά τη στρογγυλοποίηση προς τον πλησιέστερο άρτιο, αφού ο υπολογισμός της τετραγωνικής ρίζας εγγυάται πως ποτέ δε θα προκύψει ρίζα η οποία θα είναι ακριβώς μεταξύ δύο αριθμών εκ των οποίων ένας θα πρέπει να επιλεγεί ως το στρογγυλοποιημένο αποτέλεσμα.

4. Σημαίες εξαιρέσεων (Exception values) και Ειδικές τιμές:

Η πράξη της τετραγωνικής ρίζας ενός αριθμού κινητής υποδιαστολής ορίζεται μόνο όταν αυτός είναι θετικός. Σε αντίθετη περίπτωση ενεργοποιείται η σημαία εξαιρέσεων μη-έγκυρη πράξη (invalid) και το αποτέλεσμα λαμβάνει την ειδική τιμή NaN. Η τετραγωνική ρίζα του $+\infty$ είναι $+\infty$. Υπερχείλιση δεν συμβαίνει ποτέ κατά τον υπολογισμό της τετραγωνικής ρίζας. Ανεπάρκεια αναπαράστασης μπορεί να συμβεί μόνο όταν ο πολωμένος εκθέτης έχει την τιμή 1 η οποία είναι περιττός αριθμός.

Παράδειγμα: Έστω πως θέλουμε να υπολογίσουμε την τετραγωνική ρίζα του +37 του οποίου η απλή αναπαράσταση κινητής υποδιαστολής είναι $1.00101\dots00 \times 2^{132}$. Ο εκθέτης του αριθμού κινητής υποδιαστολής είναι 5 (132 σε πολωμένη σημειογραφία), οπότε αφαιρούμε μια μονάδα από τον εκθέτη και ολισθαίνουμε μια θέση δεξιά. Ο εκθέτης της τετραγωνικής ρίζας είναι 2 (129 σε πολωμένη σημειογραφία). Ταυτόχρονα το σημαϊνόμενο ολισθαίνει μια θέση αριστερά. Η τετραγωνική ρίζα του ολισθημένου σημαϊνόμενου, 10.0101...00, είναι ίση με 1.1000010101001011111101||1. Η στρογγυλοποίηση προς τον πλησιέστερο άρτιο πραγματοποιείται προσθέτοντας τη μονάδα στο λιγότερο σημαντικό ψηφίο του σημαϊνόμενου (το ψηφίο αριστερά του ||). Επομένως η στρογγυλοποιημένη τετραγωνική ρίζα είναι ίση με:

$$1.1000010101001011111110 \times 2^2 = 6.08276271820068_{10}.$$

3.8 Μετατροπές

Το πρότυπο IEEE 754 υποστηρίζει την μετατροπή των αριθμών κινητής υποδιαστολής σε ακέραιους αλλά και αντίστροφα³. Ιδιαίτερη προσοχή θα πρέπει να δίνεται κατά τις μετατροπές αυτές διότι τα φαινόμενα της υπερχείλισης και της ανεπάρκειας αναπαράστασης είναι πολύ πιθανό να εμφανιστούν.

Το 1997 ανακαλύφθηκε άλλο ένα σφάλμα στις μονάδες κινητής υποδιαστολής των επεξεργαστών Pentium-II το οποίο είχε σχέση με την μετατροπή των αριθμών κινητής υποδιαστολής σε ακεραίους. Το πρότυπο IEEE 754 ορίζει πως όταν ο αριθμός κινητής υποδιαστολής είναι πολύ μεγάλος ώστε να μετατραπεί σε ακέραιο, η πράξη της μετατροπής θα πρέπει να ενεργοποιεί τη σημαία εξαιρέσεων της μη-επιτρεπτής πράξης. Όμως ο επεξεργαστής Pentium-II

³Το πρότυπο δεν ορίζει το πλήθος των ψηφίων των ακεραίων. Οι επεξεργαστές των 32(64)-bit χρησιμοποιούν 32(64) ψηφία για την αναπαράσταση των ακεραίων

δεν ενεργοποιούσε τη σημαία αυτή αλλά ενεργοποιούσε την σημαία του μη-ακριβούς αποτελέσματος (imprecise flag) [103].

Η μετατροπή ενός αριθμού κινητής υποδιαστολής σε έναν ακέραιο πραγματοποιείται ολισθαίνοντας αριστερά το σημαϊνόμενο κατά τόσες θέσεις όσες και ο εκθέτης του αριθμού κινητής υποδιαστολής. Καθώς ο εκθέτης βρίσκεται σε πολωμένη σημειογραφία θα πρέπει να αφαιρεθεί η πόλωση πριν την ολίσθηση. Εάν ο αριθμός κινητής υποδιαστολής είναι αρνητικός και η μετατροπή γίνεται σε προσημισμένο ακέραιο, τότε λαμβάνεται το συμπλήρωμα ως προς 2 του σημαϊνόμενου. Έπειτα από την ολίσθηση τα ψηφία που βρίσκονται δεξιά της υποδιαστολής δεν αποκόπτονται αλλά χρησιμοποιούνται για την στρογγυλοποίηση του νέου αριθμού στον πλησιέστερο άρτιο. Παλαιότερα, είχε χρησιμοποιηθεί ο τρόπος στρογγυλοποίησης προς το 0.

Η μετατροπή ενός αριθμού κινητής υποδιαστολής σε έναν ακέραιο δεν είναι πάντοτε εφικτή. Αν για παράδειγμα ο αριθμός κινητής υποδιαστολής είναι $\pm\infty$ ή NaN, τότε η πράξη της μετατροπής δεν ορίζεται και στην περίπτωση αυτή ενεργοποιείται η σημαία εξαιρέσεων της άκυρης εντολής και το αποτέλεσμα της πράξης είναι NaN. Η σημαία εξαιρέσεων μη-ακριβές αποτέλεσμα (inexact) ενεργοποιείται όταν ψηφία του αριθμού κινητής υποδιαστολής έχουν χαθεί κατά την μετατροπή.

Παράδειγμα: Έστω ότι επιθυμούμε να μετατρέψουμε τον πραγματικό αριθμό +127216,9921875 του οποίου η αναπαράσταση σε απλή ακρίβεια είναι $S = 0$, $E = 10001111_2 = 143_{10}$ και το κλασματικό μέρος είναι $F = 1111000011110000111111$. Ο εκθέτης βρίσκεται σε πολωμένη μορφή, άρα αφαιρώντας την πόλωση για την απλή ακρίβεια $B = 127$, ο πραγματικός εκθέτης είναι 16. Το σημαϊνόμενο θα πρέπει να ολισθήσει αριστερά κατά 16 θέσεις όπως φαίνεται παρακάτω :

Αρχικό σημαϊνόμενο	1.111000011110000111111
Σημαϊνόμενο μετά την ολίσθηση	1111000011110000.111111

Τα ψηφία που βρίσκονται δεξιά της υποδιαστολής θα χρησιμοποιηθούν για την στρογγυλοποίηση προς τον πλησιέστερο άρτιο. Στην περίπτωση του παραδείγματος χρειάζεται να γίνει στρογγυλοποίηση οπότε προστίθεται μια μονάδα στο ψηφίο αριστερά της υποδιαστολής. Επομένως ο ακέραιος αριθμός που προκύπτει είναι ο $11111000011110001_2 = 127217_{10}$ και επιπλέον ενεργοποιείται η σημαία του μη-ακριβούς αποτελέσματος.

Η μετατροπή ενός ακεραίου αριθμού σε ένα αριθμό κινητής υποδιαστολής ακολουθεί την αντίστροφη διαδικασία. Εάν ο αριθμός είναι αρνητικός τότε λαμβάνεται το συμπλήρωμα ως προς 2 αυτού και το πρόσημο του αριθμού κινητής υποδιαστολής λαμβάνει την τιμή 1. Ο ακέραιος ολισθαίνει αριστερά τόσες θέσεις όσες και τα αρχικά μηδενικά που διαθέτει. Με αυτό τον τρόπο σχηματίζεται το σημαϊνόμενο, το οποίο περιλαμβάνει το ακέραιο μέρος, δηλαδή την κρυφή μονάδα και το κλασματικό μέρος. Καθώς η κρυφή μονάδα δε θα αποθηκευτεί ο ακέραιος ολισθαίνει ακόμα μια θέση αριστερά ώστε να προκύψει το κλασματικό μέρος.

Κατά τη μετατροπή ενός ακεραίου σε ένα αριθμό κινητής υποδιαστολής υπάρχει πιθανό

νότιπα να μην είναι δυνατή η αποθήκευση όλων των ψηφίων του ακεραίου, για παράδειγμα όταν ένας ακέραιος των 32 ψηφίων μετατρέπεται σε ένα αριθμό κινητής υποδιαστολής απλής ακρίβειας. Στην περίπτωση αυτή πραγματοποιείται στρογγυλοποίηση στον πλησιέστερο άρτιο όπως θα δούμε στο παράδειγμα που ακολουθεί.

Ο εκθέτης του αριθμού κινητής υποδιαστολής υπολογίζεται ως η διαφορά των αρχικών μηδενικών από το πλήθος των ψηφίων του ακεραίου. Από την διαφορά που θα προκύψει θα πρέπει να αφαιρεθεί μια μονάδα-1 η οποία αντιστοιχεί στην κρυφή μονάδα. Στον νέο εκθέτη προστίθεται η πόλωση ώστε ο εκθέτης να βρίσκεται σε πολωμένη σημειογραφία.

Παράδειγμα: Η μετατροπή του αριθμού 537857839, που βρίσκεται αποθηκευμένος σε ένα καταχωρητή των 32 ψηφίων, σε αριθμό κινητής υποδιαστολής απλής ακρίβειας ξεκινά με την ολίσθηση του αριθμού κατά δύο θέσεις δεξιά αφού έχει δύο αρχικά μηδενικά όπως φαίνεται παρακάτω :

Αρχικός ακέραιος	0010 0000 0000 1111 0000 1111 0010 1111
Ακέραιος μετά την ολίσθηση	1000 0000 0011 1100 0011 1100 1011 1100

Η πρώτη μονάδα αποτελεί το ακέραιο μέρος του αριθμού κινητής υποδιαστολής, δηλαδή την κρυφή μονάδα και δεν θα αποθηκευτεί. Τα επόμενα 23 ψηφία αριστερά της κρυφής μονάδας αποτελούν το κλασματικό μέρος του αριθμού κινητής υποδιαστολής. Τα υπόλοιπα ψηφία θα χρησιμοποιηθούν για την στρογγυλοποίηση του κλασματικού μέρους προς τον πλησιέστερο άρτιο :

Κλασματικό Μέρος	Υπόλοιπα ψηφία
0000 0000 0111 1000 0111 100	1011 1100

Όπως φαίνεται από τα παραπάνω χρειάζεται στρογγυλοποίηση άρα στο λιγότερο σημαντικό ψηφίο του κλασματικού μέρους θα πρέπει να προστεθεί η μονάδα.

Ο εκθέτης του αριθμού κινητής υποδιαστολής υπολογίζεται ως η διαφορά των αρχικών μηδενικών από το πλήθος των ψηφίων του ακεραίου, άρα $32 - 2 = 30$. Από αυτή τη διαφορά θα πρέπει να αφαιρέσουμε μια μονάδα εξαιτίας της επιπλέον ολίσθησης που πραγματοποιήθηκε ώστε να προκύψει το κλασματικό μέρος. Επομένως ο τελικός εκθέτης είναι 29 στον οποίο προστίθεται η πόλωση, 127. Ο αριθμός κινητής υποδιαστολής απλής αναπαράστασης είναι: $S = 0$, $E = 10011100_2 = 156_{10}$ και το κλασματικό μέρος $F = 0000000011110000111101$.

3.9 Ανακεφαλαίωση

Στο Κεφάλαιο 3,

- παρουσιάστηκαν οι έννοιες αριθμών κινητής υποδιαστολής και το θεωρητικό υπόβαθρό τους,

- παρουσιάστηκαν οι τεχνικές στρογγυλοποιήσης,
- παρουσιάστηκε το πρότυπο αναπαράστασης αριθμών κινητής υποδιαστολής IEEE 754, και
- παρουσιάστηκαν οι αρχιτεκτονικές υλοποιήσης μονάδων κινητής υποδιαστολής και η αρχιτεκτονική υλοποίησης αθροιστών υψηλής απόδοσης - αρχιτεκτονική δύο μονοπατιών.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Κεφάλαιο 4

Δοκιμαστικότητα και προσαρμόσιμη ανάπτυξη διανυσμάτων δοκιμής για μονάδες κινητής υποδιαστολής

Όπως έχει γίνει αντιληπτό από τα προηγούμενα κεφάλαια της διατριβής η δοκιμή των ολοκληρωμένων κυκλωμάτων και συγκεκριμένα των μικροεπεξεργαστών και των μονάδων κινητής υποδιαστολής είναι ιδιαίτερα σημαντική για την κατασκευή αξιόπιστων συστημάτων που επεξεργάζονται πραγματικούς αριθμούς. Η διεθνής βιβλιογραφία παρουσιάζει ένα ιδιαίτερα μεγάλο σύνολο εργασιών που πραγματεύονται τη δοκιμή και την αυτοδοκιμή των ακέραιων αριθμητικών μονάδων. Αντίθετα, οι εργασίες που ασχολούνται με τη δοκιμή των μονάδων κινητής υποδιαστολής είναι σαφώς λιγότερες ενώ πολύ λίγες είναι εκείνες που πραγματεύονται το ιδιαίτερα σημαντικής αξίας θέμα της αυτοδοκιμής. Συνοπτικά, οι εργασίες στον χώρο της δοκιμής των μονάδων κινητής υποδιαστολής χρησιμοποιούν κατά κύριο λόγο τεχνικές αύξησης της δοκιμαστικότητας εισάγοντας επιπλέον υλικό με άμεσες συνέπειες στο μέγεθος, την ταχύτητα αλλά και την κατανάλωση επιπλέον ενέργειας. Οι τεχνικές αύξησης της δοκιμαστικότητας έχουν το μειονέκτημα πως δεν είναι πάντα εύκολο να προσαρμοστούν και να χρησιμοποιηθούν για την δοκιμή των μονάδων κινητής υποδιαστολής ενός μικροεπεξεργαστή στο πεδίο λειτουργίας του τελευταίου. Μέχρι στιγμής, οι εργασίες που παρουσιάζονται στη βιβλιογραφία αναφέρονται στη δοκιμή των αθροιστών κινητής υποδιαστολής που είναι υλοποιημένοι σύμφωνα με την αρχιτεκτονική του μονού μονοπατιού (single path). Καμία εργασία δεν έχει αντιμετωπίσει τη δοκιμή των αθροιστών κινητής υποδιαστολής που είναι βασισμένοι

στην αρχιτεκτονική υλοποίησης διπλού μονοπατιού η οποία αυξάνει κατά 30% την απόδοση του αθροιστή (βλ. Κεφάλαιο 3).

4.1 Ανασκόπηση της διεθνούς βιβλιογραφίας

Η δοκιμή των μονάδων κινητής υποδιαστολής, εμφανίζεται για πρώτη φορά στην βιβλιογραφία στην εργασία [104]. Ο Sparmann μελετά την δοκιμή του αθροιστή/αφαιρέτη και του πολλαπλασιαστή κινητής υποδιαστολής. Η εργασία δείχνει τη δυσκολία που παρουσιάζει η δοκιμή των μονάδων κινητής υποδιαστολής και την μεγάλη υπολογιστική ισχύ που απαιτεί το σύστημα γέννησης διανυσμάτων δοκιμής Socrates [105]. Το σύστημα Socrates ήταν από τα αποτελεσματικότερα συστήματα εξαγωγής διανυσμάτων δοκιμής στις αρχές της δεκαετίας του 1990. Αναλυτικότερα, ο Sparmann στις πειραματικές του μετρήσεις αναφέρει πως η εξαγωγή διανυσμάτων δοκιμής για έναν πολλαπλασιαστή απλής ακρίβειας απαιτεί χρόνο εκτέλεσης περίπου 5 ημερών ώστε να επιτευχθεί το ποσοστό κάλυψης ελαττωμάτων της τάξης του 97,68%. Η δυσκολία της εξαγωγής των διανυσμάτων δοκιμής αναδεικνύεται περισσότερο λαμβάνοντας υπόψη πως για ένα απλούστερο πολλαπλασιαστή με 4 διαθέσιμα ψηφία για τον εκθέτη και 12 ψηφία για το κλασματικό μέρος απαιτούνται περίπου 4 ώρες για την εξαγωγή διανυσμάτων με ποσοστό κάλυψης ελαττωμάτων 99,27%. Για την αύξηση του ποσοστού κάλυψης ελαττωμάτων κατά 0,07% για το ίδιο κύκλωμα ο χρόνος εκτέλεσης του συστήματος Socrates είναι 9 φορές μεγαλύτερος. Ο Πίνακας 4.1 παρουσιάζει τα πειραματικά αποτελέσματα για την εξαγωγή διανυσμάτων δοκιμής από το σύστημα Socrates για τους πολλαπλασιαστές κινητής υποδιαστολής. Σύμφωνα με τον Sparmann παρόμοια είναι και τα αποτελέσματα για τους αθροιστές κινητής υποδιαστολής που είναι υλοποιημένοι σύμφωνα με την αρχιτεκτονική μονού μονοπατιού.

Στην εργασία [104], ο Sparmann αυξάνει τη δοκιμαστικότητα (testability) των μονάδων κινητής υποδιαστολής αυξάνοντας των αριθμό των εξόδων του κυκλώματος. Συγκεκριμένα, τα επιπλέον ψηφία τα οποία απορρίπτονται κατά τη διαδικασία της στρογγυλοποίησης τόσο στον αθροιστή όσο και στον πολλαπλασιαστή οδηγούνται στις κύριες εξόδους του κυκλώματος αυξάνοντας με αυτό τον τρόπο την παρατηρησιμότητα (observability). Για τη μείωση του υπολογιστικού χρόνου του συστήματος Socrates, ο Sparmann διαμερίζει και απλοποιεί

Πολλαπλασιαστής FPMult	Ποσοστό Κάλυψης	Χρόνος Εκτέλεσης
FPMult[12, 4]	99.27%	3 h 56 min
	99.34%	35 h 19 min
FPMult[24, 8]	97.68%	127 h 8 min

Πίνακας 4.1: Πειραματικά αποτελέσματα του συστήματος Socrates για την δοκιμή των πολλαπλασιαστών κινητής υποδιαστολής.

τη μονάδα κινητής υποδιαστολής ανάλογα με το υποκύκλωμα για το οποίο εξάγονται τα διανύσματα δοκιμής.

Οι Rajski και Tyszer, στην εργασία [106], προτείνουν κατάλληλες αλλαγές στην αρχιτεκτονική του αθροιστή, ώστε ο τελευταίος να είναι εύκολα δοκιμάσιμος με ψευδοτυχαία διανύσματα δοκιμής. Πιο συγκεκριμένα, μελετούν τα σημεία εκείνα της αρχιτεκτονικής τα οποία μειώνουν τη δοκιμασιμότητα του αθροιστή, όπως οι ολισθητές των κλασματικών μερών, τα οποία και μεταβάλλουν ώστε τα ψευδοτυχαία διανύσματα δοκιμής να είναι ικανά να ενεργοποιήσουν τα ελαττώματα και να οδηγήσουν τις λανθασμένες αποκρίσεις στις κύριες εξόδους του αθροιστή. Επιπλέον, εφαρμόζουν την *αριθμητική αυτοδοκιμή (Arithmetic Built-In Self-Test)* [65], χρησιμοποιώντας το αποτέλεσμα του αθροιστή για τη δημιουργία ενός νέου διανύσματος δοκιμής. Τα πειραματικά αποτελέσματα της εργασίας [106] δείχνουν τη δυσκολία της δοκιμής των αθροιστών κινητής υποδιαστολής με ψευδοτυχαία διανύσματα. Αναλυτικότερα, το ποσοστό κάλυψης ελαττωμάτων για ένα αθροιστή με 4 ψηφία για τον εκθέτη και 9 ψηφία για το κλασματικό μέρος παραμένει στο 97.05% ακόμα και όταν εφαρμοστούν σε αυτόν 2500 ψευδοτυχαία διανύσματα δοκιμής. Αντίθετα, για τον αθροιστή μονού μονοπατιού ο οποίος είναι σχεδιασμένος με αύξηση της δοκιμασιμότητας, τα 2500 ψευδοτυχαία διανύσματα είναι αρκετά για να παρέχουν πλήρη κάλυψη ελαττωμάτων

Η εργασία [107] είναι ιδιαίτερα σημαντική καθώς η συνεισφορά της είναι διπλή. Σε αντιδιαστολή με τις προηγούμενες εργασίες, η μονάδα κινητής υποδιαστολής δεν αντιμετωπίζεται σαν αυτόνομη μονάδα αλλά βρίσκεται ενσωματωμένη σε ένα μικροεπεξεργαστή. Το γεγονός αυτό αλλάζει και τις συνθήκες υπό τις οποίες πραγματοποιείται η δοκιμή. Η πρώτη συνεισφορά της εργασίας είναι η παρατήρηση πως αφού η μονάδα κινητής υποδιαστολής βρίσκεται ενσωματωμένη στο μικροεπεξεργαστή η εφαρμογή των διανυσμάτων δοκιμής μπορεί να πραγματοποιηθεί με τη βοήθεια του συνόλου των εντολών του. Η δεύτερη συνεισφορά της εργασίας είναι πως η δοκιμή των μονάδων κινητής υποδιαστολής πρέπει να πραγματοποιηθεί χωρίς την εφαρμογή τεχνικών αύξησης της δοκιμασιμότητας και πως δεν πρέπει να πραγματοποιηθεί καμία αλλαγή στο σχεδιασμό των μονάδων. Αναγνωρίζεται πως το επιπλέον υλικό για την αύξηση της δοκιμασιμότητας των μονάδων κινητής υποδιαστολής είναι απαγορευτικό. Η παρατήρηση αυτή είναι ιδιαίτερα σημαντική αν λάβει κανείς υπόψη πως η εργασία [107] προέρχεται από το ερευνητικό κέντρο της εταιρείας Hitachi. Τα διανύσματα δοκιμής εφαρμόζονται αποκλειστικά από το σύνολο των εντολών του μικροεπεξεργαστή, ενώ η εξαγωγή των διανυσμάτων πραγματοποιείται με τη βοήθεια ενός αυτόματου εργαλείου εξαγωγής διανυσμάτων. Η εξαγωγή των διανυσμάτων πραγματοποιείται βάσει των *περιορισμών (constraints)* στους οποίους υπόκειται η μονάδα κινητής υποδιαστολής όταν βρίσκεται ενσωματωμένη σε ένα μικροεπεξεργαστή. Για παράδειγμα, το εργαλείο εξαγωγής διανυσμάτων δε θα πρέπει να δημιουργήσει διανύσματα δοκιμής που ανιχνεύουν ελαττώματα στη μονάδα στρογγυλοποίησης προς τον πλησιέστερο άρτιο, ενώ στις γραμμές ελέγχου έχει γίνει η επιλογή να πραγματοποιείται στρογγυλοποίηση με αποκοπή. Τα πειραματικά αποτελέσματα στην εργασία [107] αναφέρουν ποσοστά κάλυψης

ελαττωμάτων 89,10% και 89,15% για τον αθροιστή/αφαιρέτη και τον πολλαπλασιαστή κινητής υποδιαστολής, αντίστοιχα.

Με αντίστοιχο τρόπο αντιμετωπίζεται και η δοκιμή των μονάδων κινητής υποδιαστολής στην εργασία [108], όπου επίσης αναγνωρίζεται η ανάγκη για δοκιμή δίχως επιπλέον υλικό για την αύξηση της δοκιμαστικότητας καθώς αυτό επηρεάζει τις επιδόσεις τόσο των ακέραιων μονάδων όσο και των μονάδων κινητής υποδιαστολής στο μηχανισμό διοχέτευσης (pipeline) του επεξεργαστή. Τεχνικές αύξησης της δοκιμαστικότητας εφαρμόζονται στα υπόλοιπα μέρη του επεξεργαστή CalmRisc32 [108]. Τα διανύσματα δοκιμής των μονάδων κινητής αποτελούνται από τα διανύσματα για την επιβεβαίωση του ορθού σχεδιασμού (verification) και τα διανύσματα δοκιμής που εξάγονται από ένα αυτόματο εργαλείο εξαγωγής διανυσμάτων.

Αντίθετα με τις εργασίες [107], [108], οι εργασίες [109], [110] οι οποίες μελετούν τη δοκιμαστικότητα των μονάδων κινητής υποδιαστολής στον επεξεργαστή ψηφιακού σήματος, Flona, χρησιμοποιούν επιπλέον υλικό για την αύξηση της δοκιμαστικότητας. Πιο συγκεκριμένα, οι εσωτερικοί καταχωρητές διοχέτευσης (pipeline registers) τροποποιούνται ώστε είτε να λειτουργούν είτε ως γεννήτορες ψευδοτυχαίων διανυσμάτων δοκιμής, είτε ως συμπιεστές αποκρίσεων των διανυσμάτων δοκιμής. Η τεχνική αυτή μοιάζει με την τεχνική αυξημένης δοκιμαστικότητας όπου οι καταχωρητές μετατρέπονται σε καταχωρητές BILBO. Με αυτό τον τρόπο υλοποιείται η ενσωματωμένη αυτοδοκιμή βασισμένη στο υλικό. Επιπλέον, διανύσματα που έχουν προκύψει από ένα αυτόματο εργαλείο εξαγωγής διανυσμάτων χρησιμοποιούνται ως αρχικές τιμές (seed) των γεννητριών ψευδοτυχαίων διανυσμάτων. Η τεχνική αυτή είναι γνωστή ως εξαγωγή διανυσμάτων δοκιμής με ποικιλία αρχικών τιμών (test pattern reseeding).

Η εργασία των Bayraktaroglu και d' Abreu, [111] πραγματεύεται την ανάπτυξη ενσωματωμένου λογισμικού για την αυτοδοκιμή των μονάδων κινητής υποδιαστολής, που βρίσκονται ενσωματωμένες σε ένα μικροεπεξεργαστή με δεδομένη την απουσία επιπλέον υλικού για αύξηση της δοκιμαστικότητας. Η εφαρμογή των διανυσμάτων δοκιμής πραγματοποιείται μέσω των εντολών κινητής υποδιαστολής του μικροεπεξεργαστή. Αναλυτικότερα, πριν από την εξαγωγή των διανυσμάτων δοκιμής, σχηματίζεται ένα πλήθος προτύπων (templates) τα οποία δεν είναι τίποτε άλλο παρά ομάδες εντολών οι οποίες εφαρμόζουν τα διανύσματα δοκιμής, τα οποία είναι αποθηκευμένα σε καταχωρητές και έπειτα αποθηκεύουν το αποτέλεσμα της μονάδας κινητής υποδιαστολής στη μνήμη για σύγκριση με την αναμενόμενη τιμή. Η εξαγωγή των προτύπων πραγματοποιείται με τη βοήθεια ενός αυτόματου εργαλείου εξαγωγής διανυσμάτων σύμφωνα με τους περιορισμούς που εισάγει το σύνολο εντολών κινητής υποδιαστολής του μικροεπεξεργαστή. Τα πειραματικά αποτελέσματα της εργασίας [111] αναφέρουν για τον πολλαπλασιαστή ποσοστό κάλυψης ελαττωμάτων περίπου 98% ενώ για τον αθροιστή/αφαιρέτη το αντίστοιχο ποσοστό κάλυψης είναι αρκετά χαμηλότερο, 88,4%.

Συνοψίζοντας, τα μειονεκτήματα που παρουσιάζουν οι ερευνητικές εργασίες που παρουσιάζονται στη διεθνή βιβλιογραφία έχουν ως εξής:

- Οι εργασίες μετατρέπουν την αρχιτεκτονική των μονάδων κινητής υποδιαστολής με τεχνικές αύξησης της δοκιμασιμότητας (Design for Testability). Όμως, με αυτόν τον τρόπο αυξάνεται το μέγεθος των μονάδων και η κατανάλωση ενέργειας αλλά και το σημαντικότερο όλων είναι πως μειώνεται η απόδοσή τους, όπως φαίνεται στο Κεφάλαιο 2. Επίσης, η μονάδα πρέπει να τεθεί σε συγκεκριμένη κατάσταση δοκιμής (test mode) στην οποία δε λειτουργεί ως μονάδα κινητής υποδιαστολής. Δυστυχώς, έτσι γίνεται δυσκολότερη η δοκιμή κατά τη λειτουργία όταν οι μονάδες κινητής υποδιαστολής βρίσκονται ενσωματωμένες σε ένα μικροεπεξεργαστή.
- Οι εργασίες που δε χρησιμοποιούν τεχνικές αύξησης της δοκιμασιμότητας βασίζονται σε αυτόματα εργαλεία εξαγωγής διανυσμάτων δοκιμής τα οποία εφαρμόζονται από το σύνολο των εντολών του μικροεπεξεργαστή. Τα διανύσματα δοκιμής αποθηκεύονται στην μνήμη του επεξεργαστή, και από εκεί μεταφέρονται στους καταχωρητές και εφαρμόζονται στην μονάδα κινητής υποδιαστολής. Όμως με αυτόν τον τρόπο, επιβαρύνεται ιδιαίτερα το σύστημα μνήμης καθώς και οι δίαυλοι επικοινωνίας (data-bus) του μικροεπεξεργαστή. Η εργασία [112] αναφέρει πως τα δύο αυτά συστήματα είναι υπεύθυνα για το μεγαλύτερο μέρος κατανάλωσης ενέργειας ενός μικροεπεξεργαστή.
- Όλες οι εργασίες που έχουν παρουσιαστεί στη διεθνή βιβλιογραφία αντιμετωπίζουν τις μονάδες κινητής υποδιαστολής σαν ένα οποιοδήποτε άλλο κύκλωμα. Όμως έτσι δεν είναι δυνατή η επαναχρησιμοποίηση διανυσμάτων δοκιμής που έχουν αναπτυχθεί για ακέραιες μονάδες. Όπως έχει γίνει αντιληπτό από το Κεφάλαιο 3 οι μονάδες κινητής υποδιαστολής αποτελούνται στο κύριο μέρος από ακέραιες αριθμητικές μονάδες οι οποίες συνδέονται μεταξύ τους με πολύπλοκη λογική. Τα διανύσματα δοκιμής για ακέραιες αριθμητικές μονάδες εξάγονται με βάση την αρχιτεκτονική των τελευταίων. Καθώς η δομή των αριθμητικών μονάδων διακρίνεται από υψηλό βαθμό κανονικότητας-συμμετρίας, αντίστοιχα και τα διανύσματα δοκιμής είναι δυνατό να εξαχθούν και να περιγραφούν με γενικό τρόπο ώστε και αυτά να εμφανίζουν μιας μορφής συμμετρία.
- Καμία από τις εργασίες για τη δοκιμή των μονάδων κινητής υποδιαστολής δεν αναφέρεται στην εξαγωγή διανυσμάτων δοκιμής λαμβάνοντας υπόψη την ακρίβεια κινητής υποδιαστολής (μισή, απλή, διπλή, τετραπλή) και τη γενικότητα της κατασκευής των μονάδων κινητής υποδιαστολής για προσαρμοσμένες ακρίβειες, όπως αυτές ορίζονται από τις ανάγκες της εκάστοτε εφαρμογής.
- Τέλος, καμία εργασία δεν πραγματεύεται την εξαγωγή διανυσμάτων δοκιμής αλλά ούτε και την ανάπτυξη ενσωματωμένου λογισμικού για την αυτοδοκιμή του αθροιστή/αφαιρέτη κινητής υποδιαστολής υψηλής απόδοσης, που είναι βασισμένος στην αρχιτεκτονική υλοποίησης διπλού μονοπατιού.

4.2 Βασικές αρχές εξαγωγής διανυσμάτων δοκιμής για μονάδες κινητής υποδιαστολής

Τα μειονεκτήματα που παρουσιάζουν οι εργασίες στη διεθνή βιβλιογραφία έρχεται να καλύψει το πρώτο μέρος της διδακτορικής διατριβής. Αναλυτικότερα, η κύρια προϋπόθεση που η προτεινόμενη μεθοδολογία προσπαθεί να ικανοποιήσει είναι η δοκιμή των μονάδων κινητής υποδιαστολής λαμβάνοντας υπόψη πως δεν επιτρέπεται καμία σχεδιαστική τροποποίηση για αύξηση της δοκιμαστικότητας (Design for Testability).

Εξίσου ίδιας υψηλής αξίας είναι και η ανάγκη για δοκιμή των μονάδων ανεξάρτητα από την ακρίβεια των αριθμών που υποστηρίζουν. Με αυτό τον τρόπο η μέθοδος εξαγωγής διανυσμάτων μπορεί να εφαρμοστεί όχι μόνο για ακρίβειες αναπαράστασης σύμφωνα με το πρότυπο IEEE 754 αλλά και σε οποιαδήποτε ακρίβεια είναι αναγκαία όπως αυτή καθορίζεται από τις απαιτήσεις της εκάστοτε εφαρμογής. Στο σημείο αυτό αξίζει να υπενθυμίσουμε πως με την αναθεώρηση του προτύπου έχει ενσωματωθεί η τετραπλή ακρίβεια αναπαράστασης με 128 ψηφία για την αναπαράσταση των αριθμών κινητής υποδιαστολής.

Ένα άλλο ζητούμενο είναι η ανεξαρτησία της εξαγωγής των διανυσμάτων δοκιμής από την εσωτερική υλοποίηση των ακέραιων αριθμητικών μονάδων από τις οποίες απαρτίζεται η μονάδα κινητής υποδιαστολής. Για παράδειγμα, αν η αρχιτεκτονική του ακέραιου αθροιστή αλλάξει για λόγους που έχουν να κάνουν με την ταχύτητα ή το μέγεθός του, τότε τα διανύσματα δοκιμής για τις υπόλοιπες εσωτερικές μονάδες δεν θα πρέπει να επηρεάζονται. Παράλληλα, με αυτό τον τρόπο επιτυγχάνεται η επαναχρησιμοποίηση διανυσμάτων δοκιμής τα οποία έχουν αναπτυχθεί για ακέραιες μονάδες. Καθώς η αρχιτεκτονική των περισσότερων ακέραιων αριθμητικών μονάδων χαρακτηρίζεται από υψηλή έμφυτη κανονικότητα, έτσι και τα διανύσματα δοκιμής αυτών, παρουσιάζουν χαρακτηριστικά κανονικότητας σε αντίθεση με τα διανύσματα που παράγονται από ένα αυτόματο εργαλείο εξαγωγής και τα οποία είναι τελείως ακανόνιστα. Η ιδιότητα αυτή είναι ιδιαίτερα σημαντική διότι, όπως θα φανεί στο Κεφάλαιο 5, οδηγεί στην ανάπτυξη ενσωματωμένου λογισμικού αυτοδοκιμής με ιδιαίτερα μικρές απαιτήσεις σε πόρους μνήμης. Αυτό βέβαια δεν σημαίνει πως το αυτόματο εργαλείο εξαγωγής διανυσμάτων δεν έχει θέση στην προτεινόμενη μεθοδολογία αλλά αντιθέτως είναι ιδιαίτερα χρήσιμο για την εξαγωγή διανυσμάτων τα οποία ανιχνεύουν ελαττώματα τα οποία δεν ανιχνεύονται από τα υπόλοιπα διανύσματα δοκιμής. Ας μη ξεχνάμε πως οι μονάδες κινητής υποδιαστολής εκτός από ακέραιες αριθμητικές μονάδες αποτελούνται και από μονάδες ελέγχου (control logic units) των οποίων τα ελαττώματα μπορεί να μην ανιχνεύονται από τα διανύσματα της προτεινόμενης μεθοδολογίας.

Τέλος, ένα επιπλέον ζητούμενο είναι η ανεξαρτησία της εξαγωγής των διανυσμάτων δοκιμής από το μοντέλο ελαττωμάτων που χρησιμοποιείται. Είναι γνωστό πως όσα περισσότερα διανύσματα δοκιμής διαφόρων μοντέλων ελαττωμάτων εφαρμόζονται σε ένα ολοκληρωμένο

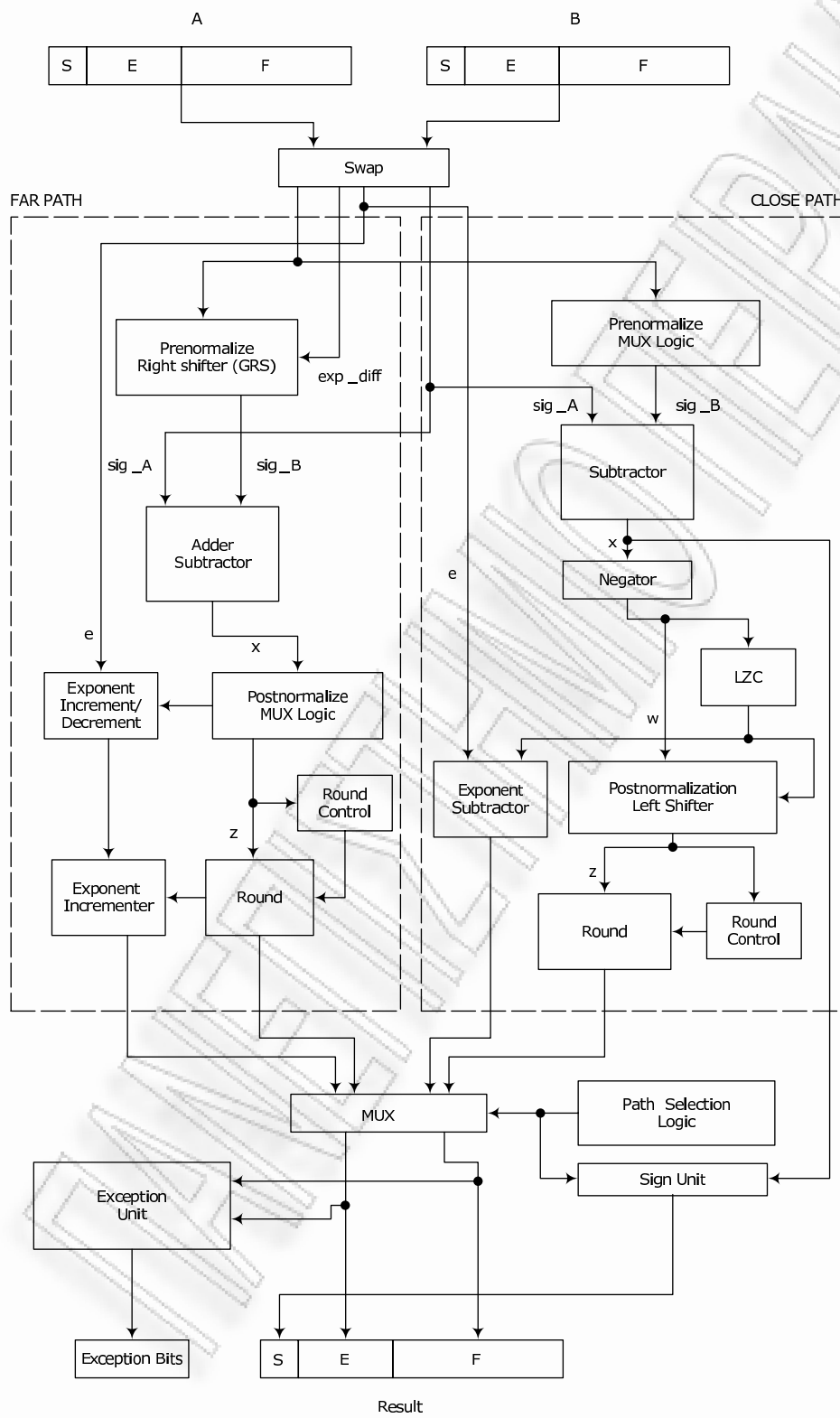
κύκλωμα τόσο περισσότερο διασφαλίζεται η αξιόπιστη λειτουργία του.

Η ανάπτυξη της μεθόδου εξαγωγής διανυσμάτων που αναλύεται στη συνέχεια βασίζεται στις συνθήκες δοκιμασιμότητας οι οποίες αναπτύσσονται για κάθε υπομονάδα της μονάδας κινητής υποδιαστολής ξεχωριστά. Οι συνθήκες δοκιμασιμότητας εξάγονται με βάση την αρχιτεκτονική της μονάδας κινητής υποδιαστολής και δεν εξαρτώνται από την εσωτερική υλοποίηση των επιμέρους υπομονάδων. Πιο συγκεκριμένα, οι συνθήκες δοκιμασιμότητας αποτελούνται από δυο επιμέρους συνθήκες, αυτές της ελεγχιμότητας (controllability) και της παρατηρησιμότητας (observability). Οι συνθήκες ελεγχιμότητας περιγράφουν τους περιορισμούς στις εισόδους των υπομονάδων της μονάδας κινητής υποδιαστολής και ορίζουν την μορφή που θα πρέπει να έχουν οι αριθμοί κινητής υποδιαστολής ώστε το επιθυμητό διάνυσμα δοκιμής να εφαρμοστεί στην εσωτερική υπομονάδα. Οι συνθήκες παρατηρησιμότητας εξετάζουν αν είναι δυνατή η παρατήρηση μιας εσφαλμένης απόκρισης μιας εσωτερικής υπομονάδας στις κύριες εξόδους της μονάδας κινητής υποδιαστολής.

Οι αθροιστές κινητής υποδιαστολής διπλού μονοπατιού όπως έχει φανεί στο Κεφάλαιο 3 αποτελούν την αποδοτικότερη αρχιτεκτονική υλοποίησης, καθώς η απόδοσή τους είναι αυξημένη κατά 30% περίπου έναντι της αρχιτεκτονικής μονού μονοπατιού. Επιπλέον, οι αθροιστές κινητής υποδιαστολής αποτελούν τις πολυπλοκότερες μονάδες κινητής υποδιαστολής, εξαιτίας της παρεμβολής των μονάδων ελέγχου (control logic modules) ανάμεσα στις ακέραιες μονάδες επεξεργασίας, όπως οι αθροιστές, αφαιρέτες και ολισθητές. Γι' αυτό το λόγο οι συνθήκες δοκιμασιμότητας αναπτύσσονται πρώτα για τους αθροιστές κινητής υποδιαστολής διπλού μονοπατιού.

4.3 Συνθήκες δοκιμασιμότητας για αθροιστές υψηλής απόδοσης διπλού μονοπατιού

Οι συνθήκες δοκιμασιμότητας παρουσιάζονται έχοντας ως αναφορά την αρχιτεκτονική δύο μονοπατιών όπως αυτή απεικονίζεται στην Εικόνα 4.1. Η Εικόνα 4.1 παρουσιάζει με περισσότερη λεπτομέρεια την αρχιτεκτονική του αθροιστή δύο μονοπατιών που απεικονίζεται στην Εικόνα 3.9.



Εικόνα 4.1: Αθροιστής/Αφαιρέτης δύο μονοπατιών

Γ.Δ. Ξενούλης - Διδακτορική Διατριβή

4.3.1 Ακέραιος αθροιστής/αφαιρέτης μακρινού μονοπατιού

Ο ακέραιος αθροιστής/αφαιρέτης του μακρινού μονοπατιού υλοποιείται συνήθως με μια γρήγορη αρχιτεκτονική πρόβλεψης κρατούμενου και έχει μέγεθος $N+3$, όπου N είναι ο αριθμός των ψηφίων του σημαιόμενου, συμπεριλαμβανομένου και του κρυφού ψηφίου.

Οι είσοδοι του ακέραιου αθροιστή, sigA , sigB δεν είναι πλήρως ελέγξιμοι (controllable) από τις κύριες εισόδους του αθροιστή εξαιτίας της προ-κανονικοποίησης. Ο τελεστής sigA έχει το σημαντικότερο ψηφίο του πάντα ίσο με τη μονάδα (1), η οποία είναι η κρυφή μονάδα. Επίσης τα τρία επιπλέον ψηφία που αντιστοιχούν στα ψηφία φρουρούς είναι μηδενικά. Οι τιμές που μπορεί να λάβει ο τελεστής sigA περιγράφονται από την εξίσωση :

$$\text{sigA} = 1.f_{p-1}f_{p-2} \dots f_1f_0\|000, \quad (4.1)$$

όπου με το σύμβολο p συμβολίζεται το πλήθος των ψηφίων ακρίβειας αναπαράστασης ($p = 23$ για μονή ακρίβεια και $p = 52$ για διπλή ακρίβεια αναπαράστασης).

Αντίστοιχα, οι τιμές που μπορεί να λάβει ο τελεστής sigB εξαρτώνται από τη διαφορά των εκθετών. Υπενθυμίζεται πως το σημαιόμενο του αριθμού κινητής υποδιαστολής με το μικρότερο εκθέτη ολισθαίνει δεξιά και σχηματίζει τα ψηφία φρουρούς, G , R , T . Όταν η διαφορά των εκθετών είναι μεγαλύτερη ή ίση του 4, τότε τα ψηφία που ολισθαίνουν εκτός του ολισθητή δεν απορρίπτονται, αλλά όλα μαζί σχηματίζουν το επίμονο ψηφίο, T , μέσω της πράξης λογικό-Η. Οι τιμές που μπορεί να λάβει ο τελεστής, sigB , περιγράφονται από την εξίσωση :

$$\text{sigB} = \begin{cases} 1.f_{p-1} \dots f_1f_0\|000 & \text{όταν } d = 0 \\ 0.1f_{p-1} \dots f_2f_1\|f_000 & \text{όταν } d = 1 \\ 0.01f_{p-1} \dots f_3f_2\|f_1f_00 & \text{όταν } d = 2 \\ 0.001f_{p-1} \dots f_4f_3\|f_2f_1f_0 & \text{όταν } d = 3 \\ \underbrace{0.0 \dots 0}_d 1f_p \dots f_d\|f_{d-1}f_{d-2}T & \text{όταν } d \geq 4 \end{cases} \quad (4.2)$$

όπου $T = (f_{d-3} + \dots + f_0)$ είναι η τιμή που λαμβάνει το επίμονο ψηφίο όταν η απόλυτη τιμή της διαφοράς των εκθετών, d , είναι μεγαλύτερη του 4 ή ίση (η πράξη + αναφέρεται στην πράξη λογικό-Η (OR)).

Οι μόνες επιτρεπτές τιμές που είναι δυνατό να λάβουν οι είσοδοι, sigA και sigB , του ακέραιου αθροιστή/αφαιρέτη του μακρινού μονοπατιού περιγράφονται από τις εξισώσεις 4.1 και 4.2.

Η διάδοση των εσφαλμένων τιμών του ακέραιου αθροιστή/αφαιρέτη και κατά συνέπεια η ανίχνευση των ελαττωμάτων δεν είναι πάντοτε εφικτή εξαιτίας των σταδίων της μετά-κανονικοποίησης και της στρογγυλοποίησης. Για να διαδοθεί ένα ελάττωμα στον αθροιστή/αφαιρέτη πρέπει να διαδοθεί στις εξόδους της υπομονάδας της στρογγυλοποίησης (Round

Unit) (βλ. Εικόνα 4.1) ή/και στις εξόδους της υπομονάδας αύξησης του εκθέτη (Exponent Incrementer). Το αποτέλεσμα του ακέραυ αθροιστή/αφαιρέτη δεν επηρεάζει το τελικό πρόσμιο.

Η έξοδος του αθροιστή/αφαιρέτη συμβολίζεται στην Εικόνα 4.1 με X και έχει τη μορφή :

$$X = x_1 x_0 \cdot f_{p-1} f_{p-1} \dots f_1 f_0 \| GRT.$$

Το ψηφίο x_1 συμβολίζει το κρατούμενο εξόδου του αθροιστή/αφαιρέτη. Το αποτέλεσμα του αθροιστή/αφαιρέτη μπορεί να μη βρίσκεται σε κανονική μορφή και να χρειάζεται αριστερή ή δεξιά ολίσθηση κατά μία θέση. Όταν, οι αριθμοί κινητής υποδιαστολής είναι ομόσημοι, δηλαδή $S_A \oplus S_B = 0$ και $x_1 = 1$ τότε το αποτέλεσμα χρειάζεται δεξιά ολίσθηση κατά μια θέση. Η τιμή του X μετά τη δεξιά ολίσθηση συμβολίζεται ως \vec{Y} και έχει τη μορφή $\vec{Y} = 1 \cdot x_0 f_{p-1} \dots f_1 \| f_0 G(R + S)$. Η δυαδική μεταβλητή \vec{p} είναι ίση με τη μονάδα μόνο όταν είναι απαραίτητη η δεξιά ολίσθηση.

Αντίθετα, αν οι αριθμοί κινητής υποδιαστολής είναι ετερόσημοι, δηλαδή $S_A \oplus S_B = 1$ και $x_1 = 0$, τότε το αποτέλεσμα, X , χρειάζεται αριστερή ολίσθηση κατά μια θέση ώστε να βρεθεί σε κανονική μορφή. Η τιμή του X μετά την αριστερή ολίσθηση συμβολίζεται ως $\overleftarrow{Y} = 1 \cdot f_{p-1} f_{p-2} \dots f_0 \| RS0$. Η δυαδική μεταβλητή \overleftarrow{p} είναι ίση με τη μονάδα μόνο όταν είναι απαραίτητη η αριστερή ολίσθηση. Οι δυαδικοί συντελεστές \vec{p} και \overleftarrow{p} δεν μπορεί να είναι ταυτόχρονα ίσοι με τη μονάδα, αφού το αποτέλεσμα, X , του αθροιστή δεν είναι δυνατό να χρειάζεται και αριστερή αλλά και δεξιά ολίσθηση την ίδια στιγμή. Τέλος, με το σύμβολο Y συμβολίζεται το αποτέλεσμα του αθροιστή/αφαιρέτη όταν αυτό βρίσκεται σε κανονική μορφή.

Οι συμβολισμοί Z , \vec{Z} , \overleftarrow{Z} αντιπροσωπεύουν τους Y , \vec{Y} , \overleftarrow{Y} έπειτα από την αποκοπή των ψηφίων φρουρών (τα ψηφία μετά το σύμβολο $\|$).

Έπειτα από το στάδιο της μετά-κανονικοποίησης ακολουθεί το στάδιο της στρογγυλοποίησης. Οι δυαδικές μεταβλητές, r , \vec{r} , \overleftarrow{r} λαμβάνουν την τιμή "1" όταν το κανονικοποιημένο αποτέλεσμα του αθροιστή/αφαιρέτη χρειάζεται στρογγυλοποίηση. Αναλυτικότερα και σύμφωνα με την στρογγυλοποίηση προς τον πλησιέστερο άρτιο (βλέπε Κεφάλαιο 3), ισχύει $r = G(R + T + f_0)$, $\vec{r} = f_0(f_1 + G + R + T)$ και $\overleftarrow{r} = R(G + T)$. Επιπλέον, υπάρχει περίπτωση να συμβεί υπερχείλιση κατά τη στρογγυλοποίηση. Υπερχείλιση συμβαίνει όταν όλα τα ψηφία του σημασιόμενου είναι ίσα με "1". Στην περίπτωση αυτή ο εκθέτης προσαυξάνεται κατά μια μονάδα και η έξοδος του αθροιστή κινητής υποδιαστολής λαμβάνει την τιμή $C^{OV} = 1.00 \dots 00$ (2^{23} για μονή ακρίβεια αναπαράστασης και 2^{52} για διπλή ακρίβεια). Η δυαδική μεταβλητή r^{OV} λαμβάνει την τιμή "1" όταν έχει συμβεί υπερχείλιση κατά την στρογγυλοποίηση.

Μέχρι στιγμής έχουμε περιγράψει με γενική μορφή τον τρόπο με τον οποίο επηρεάζεται το αποτέλεσμα του ακέραυ αθροιστή/αφαιρέτη στον μακρινό μονοπάτι. Εν τέλει το τελικό σημανόμενο, $1.F$, που φθάνει στις κύριες εξόδους του αθροιστή κινητής υποδιαστολής περιγράφεται από τις δύο υποπεριπτώσεις της εξίσωσης 4.3, δηλαδή όταν δεν έχει συμβεί

υπερχείλιση κατά τη στρογγυλοποίηση ($r^{OV} = 0$) και όταν έχει συμβεί υπερχειλίση ($r^{OV} = 1$) :

$$1.F = \begin{cases} (Z+r)(1 - \overrightarrow{p} - \overleftarrow{p}) + (\overleftarrow{Z} + \overleftarrow{r})\overleftarrow{p} + (\overrightarrow{Z} + \overrightarrow{r})\overrightarrow{p} & \text{όταν } r^{OV} = 0. \\ Z+1 = C^{OV}, \overleftarrow{Z}+1 = C^{OV}, \overrightarrow{Z}+1 = C^{OV} & \text{όταν } r^{OV} = 1. \end{cases} \quad (4.3)$$

Παράλληλα, η μετά-κανονικοποίηση και η στρογγυλοποίηση επηρεάζουν την τιμή του εκθέτη. Η τελική τιμή που λαμβάνει ο εκθέτης, E , περιγράφεται από την εξίσωση 4.4, όπου e είναι ο μεγαλύτερος εκθέτης όπως αυτός έχει υπολογιστεί από τη μονάδα εναλλαγής (Swap) (βλ. Εικόνα 4.1) :

$$E = e + \overrightarrow{p} - \overleftarrow{p} + r^{OV}. \quad (4.4)$$

Τα στάδια της μετά-κανονικοποίησης και της στρογγυλοποίησης είναι δυνατό να εμποδίσουν τη διάδοση μιας εσφαλμένης απόκρισης στις εξόδους του ακέραιου αθροιστή/αφαιρέτη στις κύριες εξόδους του αθροιστή κινητής υποδιαστολής. Αν συμβεί κάτι τέτοιο τα ελαττώματα στο εσωτερικό του ακέραιου αθροιστή δεν ανιχνεύονται. Στην περίπτωση αυτή οι αποκρίσεις του κυκλώματος που περιέχει ελαττώματα (faulty) και αποκρίσεις του κυκλώματος δίχως ελαττώματα είναι ίδιες, δηλαδή ισχύει $(E, 1.F) = (E, 1.F)_{faulty}$.

Λήμμα 1. Ένα ελάττωμα f δεν είναι δυνατό να ανιχνευθεί όταν οι ορθές και εσφαλμένες έξοδοι :

1. οδηγούν στην ίδια λειτουργία μετά-κανονικοποίησης, δηλαδή $\overleftarrow{p} = \overleftarrow{p}_f, \overrightarrow{p} = \overrightarrow{p}_f$
2. δεν οδηγούν σε υπερχειλίση κατά την στρογγυλοποίηση, δηλαδή $r^{OV} = r_f^{OV} = 0$
3. και τέλος ισχύει μια από τις συνθήκες :

$$(\alpha') |Z - Z_f| = 0, r = r_f$$

$$(\beta') |Z - Z_f| = 1, r \neq r_f$$

Οι συνθήκες 3α', 3β' ισχύουν όταν δεν πραγματοποιείται μετά-κανονικοποίηση, δηλαδή $\overleftarrow{p} = \overrightarrow{p} = 0$. Ακριβώς οι ίδιες συνθήκες ισχύουν όταν $\overleftarrow{p} = 1$ με τη διαφορά ότι τα Z, Z_f, r, r_f αντικαθίστανται από τα $\overleftarrow{Z}, \overleftarrow{Z}_f, \overleftarrow{r}, \overleftarrow{r}_f$. Ομοίως όταν $\overrightarrow{p} = 1$.

Απόδειξη. Ένα ελάττωμα f στον ακέραιο αθροιστή/αφαιρέτη του μακρινού μονοπατιού δεν ανιχνεύεται όταν $E = E_{faulty}$ και $1.F = 1.F_{faulty}$. Με τη βοήθεια της εξίσωσης 4.4 διακρίνονται δύο περιπτώσεις. Στην πρώτη περίπτωση το αποτέλεσμα του ακέραιου αθροιστή/αφαιρέτη είναι σε κανονική μορφή, δηλαδή $\overleftarrow{p} = \overrightarrow{p} = 0$. Συνεπώς, σύμφωνα με την εξίσωση 4.4 :

$$E^1 = e + r^{OV}.$$

Στη δεύτερη περίπτωση το αποτέλεσμα του αθροιστή/αφαιρέτη δε βρίσκεται σε κανονική μορφή και χρειάζεται είτε αριστερή είτε δεξιά κανονικοποίηση, δηλαδή $\overrightarrow{p} + \overleftarrow{p} = 1$. Επομένως,

σύμφωνα με την εξίσωση 4.4 :

$$E^2 = e + 2\vec{p} - 1 + r^{OV} = e - 2\overleftarrow{p} + 1 + r^{OV}.$$

Για να διαπιστώσουμε πότε ισχύει $E = E_{faulty}$ θα πρέπει να εξετάσουμε τις ακόλουθες περιπτώσεις.

$$\begin{aligned} E^1 &= E_{faulty}^1, & E^2 &= E_{faulty}^2 \\ E^2 &= E_{faulty}^1, & E^1 &= E_{faulty}^2 \end{aligned}$$

$E^1 = E_{faulty}^1$: Σε αυτό το σενάριο, τόσο η ορθή όσο και η εσφαλμένη έξοδος του ακέραου αθροιστή/αφαιρέτη βρίσκονται σε κανονική μορφή, άρα :

$$E^1 = E_{faulty}^1 \Rightarrow e + r^{OV} = e + r_f^{OV} \Rightarrow r^{OV} = r_f^{OV}.$$

Η παραπάνω ισότητα ισχύει για τις τιμές $(\overleftarrow{p}, \vec{p}, r^{OV}) = (0, 0, 0)$ και $(\overleftarrow{p}_f, \vec{p}_f, r_f^{OV}) = (0, 0, 0)$ ή για τις τιμές $(\overleftarrow{p}, \vec{p}, r^{OV}) = (0, 0, 1)$ και $(\overleftarrow{p}_f, \vec{p}_f, r_f^{OV}) = (0, 0, 1)$. (Οι τιμές παρουσιάζονται στις γραμμές 1 και 2 του Πίνακα 4.2).

$E^2 = E_{faulty}^2$: Σε αυτή την περίπτωση τόσο η ορθή όσο και η εσφαλμένη έξοδος του αθροιστή/αφαιρέτη χρειάζονται κανονικοποίηση, είτε αριστερά είτε δεξιά. Άρα $\vec{p} + \overleftarrow{p} = 1$. Αντικαθιστώντας τον συντελεστή \overleftarrow{p} με $1 - \vec{p}$ στην εξίσωση 4.4 έχουμε :

$$E^2 = E_{faulty}^2 \Rightarrow e + 2p^R - 1 + r^{OV} = e + 2p_f^R - 1 + r_f^{OV} \Rightarrow 2(p^R - p_f^R) = r_f^{OV} - r_f.$$

Η ισότητα ικανοποιείται για τις τιμές $(\overleftarrow{p}, \vec{p}, r^{OV}) = (0, 1, 1)$ και $(\overleftarrow{p}_f, \vec{p}_f, r_f^{OV}) = (0, 1, 1)$ ή για τις τιμές $(\overleftarrow{p}, \vec{p}, r^{OV}) = (1, 0, 0)$ και $(\overleftarrow{p}_f, \vec{p}_f, r_f^{OV}) = (1, 0, 0)$. (Οι τιμές παρουσιάζονται στις γραμμές 3 και 4 του Πίνακα 4.2).

Αντικαθιστώντας τον συντελεστή \vec{p} με $1 - \overleftarrow{p}$ στην εξίσωση 4.4 έχουμε :

$$E^2 = E_{faulty}^2 \Rightarrow e - 2\overleftarrow{p} + 1 + r^{OV} = e - \Rightarrow 2(\overleftarrow{p} - \overleftarrow{p}_f) = r^{OV} - r_f^{OV}.$$

Η ισότητα ικανοποιείται για τις τιμές $(\overleftarrow{p}, \vec{p}, r^{OV}) = (1, 0, 1)$ και $(\overleftarrow{p}_f, \vec{p}_f, r_f^{OV}) = (1, 0, 1)$ ή για τις τιμές $(\overleftarrow{p}, \vec{p}, r^{OV}) = (0, 1, 0)$ και $(\overleftarrow{p}_f, \vec{p}_f, r_f^{OV}) = (0, 1, 0)$. (Οι τιμές παρουσιάζονται στις γραμμές 5 και 6 του Πίνακα 4.2).

$E^2 = E_{faulty}^1$: Σε αυτό το σενάριο η λανθασμένη έξοδος του αθροιστή/αφαιρέτη βρίσκεται σε κανονική μορφή ενώ αντίθετα η ορθή έξοδος χρειάζεται κανονικοποίηση. Αντικαθιστώντας το συντελεστή \overleftarrow{p} με $1 - \vec{p}$ στην εξίσωση 4.4 έχουμε:

$$E_{faulty}^1 = E^2 \Rightarrow e + r_f^{OV} = e + 2\vec{p} - 1 + r^{OV} \Rightarrow 2\vec{p} = r_f^{OV} - r^{OV} + 1.$$

Η ισότητα ισχύει για τις τιμές $(\overleftarrow{p}, \overrightarrow{p}, r^{OV}) = (0, 1, 0)$ και $(\overleftarrow{p}_f, \overrightarrow{p}_f, r_f^{OV}) = (0, 0, 1)$ ή για τις τιμές $(\overleftarrow{p}, \overrightarrow{p}, r^{OV}) = (1, 0, 1)$ και $(\overleftarrow{p}_f, \overrightarrow{p}_f, r_f^{OV}) = (0, 0, 0)$. (Οι τιμές παρουσιάζονται στις γραμμές 7 και 8 του Πίνακα 4.2).

Αντικαθιστώντας το συντελεστή \overrightarrow{p} με $1 - \overleftarrow{p}$ φθάνουμε ξανά στις ίδιες τιμές.

$E^1 = E_{faulty}^2$: Τέλος, σε αυτή την περίπτωση η εσφαλμένη έξοδος του αθροιστή/αφαιρέτη χρειάζεται κανονικοποίηση αλλά η ορθή έξοδος βρίσκεται σε κανονική μορφή. Αντικαθιστώντας το συντελεστή \overleftarrow{p} με $1 - \overrightarrow{p}$ στην 4.4, έχουμε

$$E_{faulty}^1 = E^2 \Rightarrow e + r^{OV} = e + 2\overrightarrow{p}_f - 1 + r_f^{OV} \Rightarrow 2\overrightarrow{p}_f = r^{OV} - r_f^{OV} + 1$$

Η παραπάνω ισότητα ισχύει για τιμές $(\overleftarrow{p}, \overrightarrow{p}, r^{OV}) = (0, 0, 1)$ και $(\overleftarrow{p}_f, \overrightarrow{p}_f, r_f^{OV}) = (0, 1, 0)$ ή για τιμές $(\overleftarrow{p}, \overrightarrow{p}, r^{OV}) = (0, 0, 0)$ και $(\overleftarrow{p}_f, \overrightarrow{p}_f, r_f^{OV}) = (1, 0, 1)$. (Οι τιμές παρουσιάζονται στις γραμμές 9 και 10 του Πίνακα 4.2)

Αντικαθιστώντας το συντελεστή \overrightarrow{p} με $1 - \overleftarrow{p}$ στις ίδιες τιμές.

Ο Πίνακας 4.2 συνοψίζει τις τιμές εκείνες των $(\overleftarrow{p}, \overrightarrow{p}, r^{OV})$ και $(\overleftarrow{p}, \overrightarrow{p}, r^{OV})_{faulty}$ για τις οποίες ισχύει $E = E_{faulty}$.

	$(\overleftarrow{p}, \overrightarrow{p}, r^{OV})$	$(\overleftarrow{p}, \overrightarrow{p}, r^{OV})_{faulty}$
1	(0, 0, 0)	(0, 0, 0)
2	(0, 0, 1)	(0, 0, 1)
3	(0, 1, 1)	(0, 1, 1)
4	(1, 0, 0)	(1, 0, 0)
5	(1, 0, 1)	(1, 0, 1)
6	(0, 1, 0)	(0, 1, 0)
7	(0, 1, 0)	(0, 0, 1)
8	(1, 0, 1)	(0, 0, 0)
9	(0, 0, 1)	(0, 1, 0)
10	(0, 0, 0)	(1, 0, 1)

Πίνακας 4.2: Οι λύσεις της ισότητας $E = E_{faulty}$

Στη συνέχεια, για κάθε ζεύγος τιμών του Πίνακα 4.2 μελετούμε τις λύσεις της ισότητας $1.F = 1.F_{faulty}$.

Το τρέχον λήμμα αποδεικνύεται λαμβάνοντας υπόψη τα ζεύγη τιμών που παρουσιάζονται στις γραμμές 1, 4, και 6 του Πίνακα 4.2. Παρουσιάζουμε την απόδειξη για το ζευγάρι τιμών της πρώτης γραμμής του Πίνακα 4.2. Με παρόμοιο τρόπο, αναπτύσσεται η απόδειξη για τα ζεύγη τιμών των γραμμών 4 και 6.

Αφού, τόσο η ορθή όσο και η εσφαλμένη έξοδος του ακέρατου αθροιστή/αφαιρέτη βρίσκονται σε κανονικοποιημένη μορφή, δηλαδή $(\overleftarrow{p} = \overleftarrow{p}_f = \overrightarrow{p} = \overrightarrow{p}_f = 0)$, και δε συμβαίνει υπερχείλιση

κατά την στρογγυλοποίηση, δηλαδή ($r^{OV} = r_f^{OV} = 0$), το ελάττωμα f δεν μπορεί να ανιχνευθεί όταν σύμφωνα με την εξίσωση 4.3 :

$$1.F = 1.F_f \Rightarrow Z + r = Z_f + r_f \Rightarrow |Z - Z_f| = |r - r_f| \Rightarrow \begin{cases} 0 & r = r_f \\ 1 & r \neq r_f \end{cases}$$

□

Λήμμα 2. Ένα ελάττωμα f δε μπορεί να ανιχνευθεί όταν n ορθή και n εσφαλμένη έξοδος του ακεραίου αθροιστή/αφαιρέτη:

1. οδηγούν στην ίδια λειτουργία μετά-κανονικοποίησης, δηλαδή $\overleftarrow{p} = \overleftarrow{p}_f$, και $\overrightarrow{p} = \overrightarrow{p}_f$
2. οδηγούν σε υπερχείλιση κατά τη στρογγυλοποίηση, δηλαδή $r^{OV} = r_f^{OV} = 1$
3. $Z + 1 = Z_f + 1 = C^{OV}$

Η συνθήκη 3 ισχύει όταν $\overrightarrow{p} = \overleftarrow{p} = 0$. Η ίδια συνθήκη ισχύει για τα $\overrightarrow{Z}, \overrightarrow{Z}_f$, όταν $\overrightarrow{p} = 1$ και για τα $\overleftarrow{Z}, \overleftarrow{Z}_f$, όταν $\overleftarrow{p} = 1$.

Απόδειξη. Η απόδειξη του λήμματος γίνεται λαμβάνοντας υπόψη τα ζεύγη τιμών στις γραμμές 2, 3 και 5 του Πίνακα 4.2. Θα δείξουμε την απόδειξη για το ζεύγος τιμών της γραμμής 5 του Πίνακα 4.2. Αφού τόσο n ορθή όσο και n εσφαλμένη έξοδος του αθροιστή/αφαιρέτη χρειάζονται αριστερή ολίσθηση για να βρεθούν σε κανονική μορφή, δηλαδή $\overleftarrow{p} = \overleftarrow{p}_f = 1$, και υπερχειλίζουν κατά τη στρογγυλοποίηση, δηλαδή $r^{OV} = r_f^{OV} = 1$, τότε σύμφωνα με την εξίσωση 4.3 ένα ελάττωμα f δεν ανιχνεύεται όταν :

$$1.F = 1.F_f \Rightarrow \overleftarrow{Z} + 1 = \overleftarrow{Z}_f + 1 = C^{OV}.$$

Αντίστοιχη είναι η απόδειξη για τα ζεύγη τιμών των γραμμών 2 και 3. □

Λήμμα 3. Ένα ελάττωμα f δε μπορεί να ανιχνευτεί όταν:

1. n ορθή (εσφαλμένη) έξοδος πραγματοποιεί δεξιά ολίσθηση για μετά-κανονικοποίηση, ενώ n εσφαλμένη (ορθή) έξοδος βρίσκεται σε κανονική μορφή, δηλαδή $\overleftarrow{p} = \overleftarrow{p}_f = 0$, και $\overrightarrow{p} \neq \overrightarrow{p}_f$
2. n ορθή (εσφαλμένη) έξοδος δεν χρειάζεται στρογγυλοποίηση ενώ n εσφαλμένη (ορθή) έξοδος υπερχειλίζει κατά τη στρογγυλοποίηση, δηλαδή $r^{OV} \neq r_f^{OV}$
3. και ισχύει μία από τις συνθήκες :

$$(\alpha') \overrightarrow{Z} = Z_f + 1 = C^{OV}$$

$$(\beta') Z + 1 = \overrightarrow{Z}_f = C^{OV}$$

Η συνθήκη 3α' ισχύει όταν $\overrightarrow{p} = 1$ και $r^{OV} = 1$, ενώ η συνθήκη 3β' ισχύει όταν $\overrightarrow{p} = 0$ και $r^{OV} = 0$.

Απόδειξη. Τα ζεύγη τιμών των γραμμών 7 και 9 του Πίνακα 4.2 αποτελούν τη βάση της απόδειξης του τρέχοντος λήμματος. Δείχνουμε την απόδειξη για το ζεύγος τιμών της γραμμής 7. Εφόσον η ορθή έξοδος του αθροιστή/αφαιρέτη δεν χρειάζεται κανονικοποίηση, δηλαδή ($\overrightarrow{p} = 0$), και δεν υπερχειλίζει κατά την στρογγυλοποίηση, δηλαδή $r^{OV} = 0$, τότε σύμφωνα με την εξίσωση 4.3 ισχύει $1.F = \overrightarrow{Z}$. Αντίθετα, όταν η εσφαλμένη έξοδος του αθροιστή/αφαιρέτη βρίσκεται σε κανονική μορφή, δηλαδή $\overleftarrow{p}_f = \overrightarrow{p}_f = 0$, αλλά υπερχειλίζει κατά την στρογγυλοποίηση, δηλαδή $r_f^{OV} = 1$ τότε σύμφωνα με την εξίσωση 4.3 ισχύει $1.F_f = Z_f + 1 = C^{OV}$. Ένα ελάττωμα f δεν ανιχνεύεται όταν :

$$1.F = 1.F_f \Rightarrow \overrightarrow{Z} = Z_f + 1 = C^{OV}.$$

Με αντίστοιχο τρόπο αποδεικνύεται και το δεύτερο μέρος του λήμματος. □

Λήμμα 4. Ένα ελάττωμα f δεν ανιχνεύεται όταν:

1. ορθή (εσφαλμένη) έξοδος πραγματοποιεί αριστερή ολίσθηση για την μετά-κανονικοποίηση ενώ η εσφαλμένη (ορθή) έξοδος είναι κανονικοποιημένη, δηλαδή: $\overleftarrow{p} \neq \overleftarrow{p}_f$, $\overrightarrow{p} = \overrightarrow{p}_f = 0$
2. η ορθή (εσφαλμένη) έξοδος υπερχειλίζει κατά την στρογγυλοποίηση ενώ η εσφαλμένη (ορθή) έξοδος δε χρειάζεται στρογγυλοποίηση, δηλαδή: $r^{OV} \neq r_f^{OV}$
3. και ισχύει μια από τις παρακάτω συνθήκες :

$$(\alpha') \overleftarrow{Z} + 1 = Z_f = C^{OV}$$

$$(\beta') Z = \overleftarrow{Z}_f + 1 = C^{OV}$$

Η συνθήκη 3α' ισχύει όταν $\overleftarrow{p} = 1$ και $r^{OV} = 1$, ενώ η συνθήκη 3β' ισχύει όταν $\overleftarrow{p} = 0$ και $r^{OV} = 1$.

Απόδειξη. Τα ζεύγη τιμών στις γραμμές 8 και 10 του Πίνακα 4.2 αποτελούν τη βάση για την απόδειξη του λήμματος. Στη συνέχεια παρουσιάζουμε το δεύτερο μέρος του λήμματος. Εφόσον η εσφαλμένη έξοδος του αθροιστή/αφαιρέτη χρειάζεται αριστερή ολίσθηση για μετά-κανονικοποίηση, δηλαδή \overleftarrow{p}_f , και υπερχειλίζει κατά την στρογγυλοποίηση, δηλαδή $r_f^{OV} = 1$ τότε σύμφωνα με την εξίσωση 4.3 ισχύει : $1.F_f = Z_f^I + 1 = C^{OV}$. Αντίθετα, όταν η ορθή έξοδος του αθροιστή/αφαιρέτη βρίσκεται σε κανονική μορφή, δηλαδή $\overleftarrow{p} = \overrightarrow{p} = 0$ και δεν υπερχειλίζει κατά τη στρογγυλοποίηση, τότε ισχύει : $1.F = Z$. Επομένως, ένα ελάττωμα f δεν ανιχνεύεται όταν :

$$1.F = 1.F_f \Rightarrow Z = Z_f^I + 1 = C^{OV}.$$

Με τον ίδιο τρόπο αποδεικνύεται και το πρώτο μέρος του λήμματος. □

Λήμμα 5. Σε όλες τις υπόλοιπες περιπτώσεις η διάδοση των εσφαλμένων αποκρίσεων του ακέραiou αθροιστή/αφαιρέτη εξασφαλίζεται καθώς είτε τα σημαινόμενα του αθροιστή δίχως ελαττώματα και του αθροιστή με ελαττώματα θα είναι διαφορετικά είτε οι εκθέτες θα είναι διαφορετικοί είτε και τα δύο.

Απόδειξη. Έχοντας εξετάσει όλα τα ζεύγη τιμών του Πίνακα 4.2 που επαληθεύουν την ισότητα $E = E_{faulty}$ στα προηγούμενα λήμματα, αποδεικνύεται πως σε όλες τις υπόλοιπες περιπτώσεις ένα ελάττωμα f στον αθροιστή/αφαιρέτη θα διαφοροποιεί την ορθή από την εσφαλμένη έξοδο επομένως το ελάττωμα f θα ανιχνεύεται πάντοτε. Επιπλέον, στα προηγούμενα λήμματα έχουμε προσδιορίσει τις τιμές της ορθής και της εσφαλμένης εξόδου οι οποίες εμποδίζουν την ανίχνευση του ελαττώματος f . Έτσι, είναι εμφανές πως σε όλες τις υπόλοιπες περιπτώσεις είτε ο εκθέτης είτε το σημαινόμενο είτε και τα δύο θα διαφοροποιούνται εξαιτίας του ελαττώματος στον ακέραιο αθροιστή/αφαιρέτη και επομένως το ελάττωμα θα ανιχνεύεται. \square

4.3.2 Ακέραιος αφαιρέτης σύντομου μονοπατιού

Ο ακέραιος αφαιρέτης του σύντομου μονοπατιού υλοποιείται όπως και όλες οι ακέριες αριθμητικές μονάδες στη μονάδα κινητής υποδιαστολής, με ένα αθροιστή υψηλής απόδοσης με πρόβλεψη κρατούμενου. Το μέγεθός του είναι $N + 1$, όπου N είναι ο αριθμός των ψηφίων του σημαινόμενου συμπεριλαμβανομένης και της κρυφής μονάδας. Υπενθυμίζουμε πως ο αφαιρέτης υπολογίζει τη διαφορά των σημαινόμενων όταν η απόλυτη τιμή της διαφοράς των εκθετών είναι 0 ή 1 και η πραγματική πράξη είναι αφαίρεση.

Οι είσοδοι του αφαιρέτη δεν είναι πλήρως ελέγξιμοι εξαιτίας του σταδίου της προ-κανονικοποίησης για το σημαινόμενο του αριθμού με το μικρότερο εκθέτη. Στην Εικόνα 4.1 παρουσιάζονται οι είσοδοι του ακέραiou αφαιρέτη του σύντομου μονοπατιού, συμβολίζονται δε με $sigA$ και $sigB$ αντίστοιχα. Ο είσοδος $sigA$ και $sigB$ έχουν ένα ψηφίο φρουρό το οποίο για τον τελεστέο $sigA$ είναι πάντα 0. Αντίθετα, το ψηφίο φρουρός για τον τελεστέο $sigB$ καθορίζεται από τη διαφορά των εκθετών. Οι εξισώσεις 4.5, 4.6 περιγράφουν την μορφή που πρέπει να έχουν τα διανύσματα δοκιμής για τον ακέραιο αφαιρέτη του σύντομου μονοπατιού :

$$sigA = 1.f_{p-1}f_{p-2} \dots f_1f_0 \parallel 0 \quad (4.5)$$

$$sigB = \begin{cases} 1.f_{p-1} \dots f_1f_0 \parallel 0 & \text{όταν } d = 0 \\ 0.1f_{p-1} \dots f_2f_1 \parallel f_0 & \text{όταν } d = 1 \end{cases} \quad (4.6)$$

όπου p είναι ο αριθμός των ψηφίων ακρίβειας αναπαράστασης και d η διαφορά των εκθετών.

Η διάδοση των εσφαλμένων τιμών του ακέραiou αφαιρέτη και κατά συνέπεια η ανίχνευση των ελαττωμάτων δεν είναι πάντοτε εφικτή εξαιτίας των σταδίων της μετά-κανονικοποίησης

και της στρογγυλοποίησης. Για να διαδοθεί ένα ελάττωμα στον αφαιρέτη πρέπει να διαδοθεί στις εξόδους της υπομονάδας της στρογγυλοποίησης (Round Unit) (βλ. Εικόνα 4.1) ή και στις εξόδους της υπομονάδας αφαιρέτη του εκθέτη (Exponent Subtractor) στο σύντομο μονοπάτι.

Η έξοδος του ακέραιου αφαιρέτη στο σύντομο μονοπάτι συμβολίζεται στην Εικόνα 4.1 με X και έχει τη μορφή :

$$X = x_0 \cdot f_{p-1} f_{p-2} \dots f_1 f_0 \parallel R.$$

Όμως το αποτέλεσμα του αφαιρέτη μπορεί να μη βρίσκεται σε κανονική μορφή. Τότε το αποτέλεσμα του αφαιρέτη ολισθαίνει αριστερά κατά τόσες θέσεις όσες και ο αριθμός των αρχικών μηδενικών (leading zeroes) που υπολογίζονται από τη μονάδα μέτρησης αρχικών μηδενικών (Leading Zero Counter - LZC) (βλ. Εικόνα 4.1). Με το σύμβολο Y εκφράζεται η τιμή X έπειτα από τη μετά-κανονικοποίηση και ισχύει :

$$Y = 2^{LZC(X)} X.$$

Στρογγυλοποίηση πραγματοποιείται όταν τα δύο δεξιότερα ψηφία του Y είναι μη μηδενικά. Η στρογγυλοποίηση και η μετά-κανονικοποίηση είναι αμοιβαία αποκλειόμενες εργασίες. Κατά την αριστερή ολίσηση τα δεξιότερα ψηφία μετατοπίζονται αριστερά και στη θέση τους εισέρχονται μηδενικά ψηφία. Τώρα τα δεξιότερα ψηφία του Y είναι μηδενικά και έτσι δεν πραγματοποιείται στρογγυλοποίηση.

Η είσοδος του υποκυκλώματος στρογγυλοποίησης (Round Unit) στο σύντομο μονοπάτι, συμβολίζεται με Z , και είναι ίση με την τιμή του Y έχοντας απορρίψει το ψηφίο φρουρό, R , (δεξιά του συμβόλου \parallel). Η τελική τιμή του σημασιόμενου που εμφανίζεται στις εξόδους του σύντομου μονοπατιού δίνεται από την :

$$1.F = Z + r, \quad (4.7)$$

όπου r είναι ένας δυαδικός συντελεστής ο οποίος είναι ίσος με "1" μόνο όταν πραγματοποιείται στρογγυλοποίηση.

Αντίστοιχα, η τελική τιμή του εκθέτη στις κύριες εξόδους του σύντομου μονοπατιού δίνεται από την :

$$E = e - LZC(X). \quad (4.8)$$

Κατά τη στρογγυλοποίηση στο σύντομο μονοπάτι δε συμβαίνει υπερχείλιση.

Ένα ελάττωμα στον ακέραιο αφαιρέτη δεν είναι δυνατό να ανιχνευθεί όταν $(E, 1.F) = (E, 1.F)_{faulty}$.

Λήμμα 6. Ένα ελάττωμα f στον ακέραιο αφαιρέτη δεν ανιχνεύεται όταν:

1. η ελεύθερη από ελαττώματα έξοδος και η ελαττωματική έξοδος έχουν τον ίδιο αριθμό αρχικών μηδενικών, δηλαδή: $LZC(X) = LZC(X_f)$.
2. και ισχύει μια εκ των συνθηκών :

$$(\alpha') |Z - Z_f| = 0, r = r_f = 0$$

$$(\beta') |Z - Z_f| = 1, r \neq r_f$$

Απόδειξη. Ένα ελάττωμα f δεν μπορεί να ανιχνευθεί όταν $E = E_f$ και $1.F = 1.F_f$. Επομένως,

$$E = E_f \Rightarrow e - LZC(X) = e - LZC(X_f) \Rightarrow LZC(X) = LZC(X_f),$$

το οποίο είναι και το πρώτο μέρος του λήμματος. Επίσης για να είναι αδύνατη η ανίχνευση του ελαττώματος f θα πρέπει να ισχύει :

$$1.F = 1.F_f \Rightarrow Z + r = Z_f + r_f.$$

Όταν $r = r_f = 0$ η εξίσωση ισχύει όταν $|Z - Z_f| = 0$ ενώ όταν $r \neq r_f$ η εξίσωση ισχύει μόνο όταν $|Z - Z_f| = 1$. Οι δύο περιπτώσεις αποτελούν το δεύτερο μέρος του λήμματος. Όταν $r = r_f = 1$ τότε το ελάττωμα f θα ανιχνεύεται πάντοτε διότι :

$$Z + 1 = Z_f + 1 \Rightarrow \lfloor X/2 \rfloor = \lfloor X_f/2 \rfloor \Rightarrow X - X \bmod 2 = X_f - X_f \bmod 2.$$

Εφόσον πραγματοποιείται στρογγυλοποίηση και στις δύο περιπτώσεις τα δύο δεξιότερα ψηφία των X και X_f θα είναι ίσα με "1", άρα $X \bmod 2 = X_f \bmod 2 = 1$. Αφού όμως $X \neq X_f$ το ελάττωμα ανιχνεύεται σε κάθε περίπτωση. \square

4.3.3 Ολισθητής μετά-κανονικοποίησης σύντομου μονοπατιού

Η μετά-κανονικοποίηση στο σύντομο μονοπάτι, δεν είναι τίποτα περισσότερο από έναν αριστερό ολισθητή ο οποίος ολισθαίνει το αποτέλεσμα της αφαίρεσης κατά τόσες θέσεις όσες και ο αριθμός των αρχικών μηδενικών του. Η είσοδος του ολισθητή αποτελείται από $N + 2$ ψηφία, όπου N είναι ο αριθμός των ψηφίων του σημασιόμενου, ενώ η έξοδος του αποτελείται από $N + 1$ ψηφία, τα οποία αποτελούν την είσοδο για την υπομονάδα στρογγυλοποίησης του σύντομου μονοπατιού.

Η θέση του ολισθητή στο μονοπάτι δεδομένων του σύντομου μονοπατιού είναι τέτοια, που οδηγεί σε σοβαρά προβλήματα ελεγχιμότητας. Όταν η διαφορά των εκθετών είναι ίση με 0, τότε για τους αριθμούς κινητής υποδιαστολής, a, b ισχύει :

$$\left. \begin{array}{l} 1 \leq a < 2 \\ 1 \leq b < 2 \end{array} \right\} \Rightarrow |a - b| < 1.$$

Επομένως είναι βέβαιο, πως όταν οι αριθμοί κινητής υποδιαστολής έχουν τον ίδιο εκθέτη η διαφορά θα έχει αρχικά μηδενικά.

Αντίστοιχα, όταν η διαφορά των εκθετών είναι 1, τότε ισχύει :

$$\left. \begin{array}{l} 1 \leq a < 2 \\ 0.5 \leq b < 1 \end{array} \right\} \Rightarrow 0 < |a - b| < 1.5.$$

Επομένως η διαφορά μπορεί να έχει αρχικά μηδενικά αλλά μπορεί και όχι. Στην περίπτωση που δεν υπάρχουν αρχικά μηδενικά τα δύο σημαντικότερα ψηφία της εισόδου του ολισθητή, w , (Εικόνα 4.1, σύντομο μονοπάτι) έχουν την τιμή "1.0".

Η εξίσωση 4.9 παρουσιάζει τις επιτρεπτές τιμές που λαμβάνει η είσοδος, w , του ολισθητή.

$$w = \begin{cases} 1.0f_{p-1}f_{p-2} \dots f_1f_0, & \text{όταν } LZC(w) = 0 \\ \underbrace{0.0 \dots 0}_{LZC(w)} f_{p-1}f_{p-2} \dots f_{LZC(w)}, & \text{όταν } LZC(w) \neq 0, \end{cases} \quad (4.9)$$

όπου p είναι ο αριθμός των ψηφίων ακρίβειας αναπαράστασης του αριθμού κινητής υποδιαστολής.

Η διάδοση των εσφαλμένων τιμών στις εξόδους του ολισθητή μπορεί να θεωρηθεί ως υποπερίπτωση της διάδοσης των εσφαλμένων τιμών στις εξόδους του ακέραιου αφαιρέτη με την ακόλουθη διαφορά. Μια εσφαλμένη τιμή στις εξόδους του ολισθητή μπορεί να ανιχνευθεί μόνο στις κύριες εξόδους του σημασιόμενου και όχι σε αυτές του εκθέτη. Η ανίχνευση των ελαττωμάτων στον ολισθητή εμποδίζεται μόνο από τη μονάδα στρογγυλοποίησης. Το επόμενο λήμμα ορίζει υπό ποιες συνθήκες δεν είναι δυνατή η ανίχνευση ενός ελαττώματος στον ολισθητή.

Λήμμα 7. Ένα ελάττωμα f δεν είναι δυνατό να ανιχνευθεί όταν ισχύει μια από τις ακόλουθες συνθήκες :

1. $|Z - Z_f| = 0, r = r_f = 0$
2. $|Z - Z_f| = 1, r \neq r_f$

Απόδειξη. Η απόδειξη του λήμματος είναι ίδια με αυτή του ακέραιου αφαιρέτη του σύντομου μονοπατιού. □

4.3.4 Συμπλήρωμα ως προς 2 (Σύντομο Μονοπάτι)

Η μονάδα συμπληρώματος ως προς 2 στο σύντομο μονοπάτι χρησιμοποιείται για να μετατρέψει την αρνητική διαφορά των σημασιόμενων σε θετική. Το βήμα αυτό είναι απαραίτητο διότι οι αριθμοί κινητής υποδιαστολής αναπαρίστανται με το πρόσημο και μέγεθος (sign and magnitude) όπου το μέγεθος είναι πάντα ένας θετικός αριθμός.

Η υλοποίηση της μονάδας συμπληρώματος ως προς 2 γίνεται με τη βοήθεια ενός αυξητή (incrementer) με κρατούμενο εισόδου 1, του οποίου η είσοδος είναι το αποτέλεσμα του αφαιρέτη αντεστραμμένο. Το μέγεθος της μονάδας συμπληρώματος ως προς 2 είναι $N + 1$, όπου N είναι το πλήθος των ψηφίων του σημασιόμενου.¹

Το λήμμα που ακολουθεί αποτελεί τη συνθήκη ελεγχιμότητας των ελαττωμάτων της μονάδας συμπληρώματος ως προς 2 και ορίζει τη μορφή των διανυσμάτων δοκιμής.

¹Το επιπλέον ψηφίο είναι απαραίτητο για την στρογγυλοποίηση.

Λήμμα 8. Ένα ελάττωμα f στη μονάδα συμπληρώματος ως προς 2 μπορεί να ενεργοποιηθεί μόνο όταν:

1. Η απόλυτη τιμή της διαφοράς των εκθετών είναι 0 και η πραγματική πράξη είναι αφαίρεση
2. $1.F_A < 1.F_B$
3. Τα διανύσματα δοκιμής (δηλαδή η είσοδος της μονάδας συμπληρώματος ως προς 2) έχουν το περισσότερο σημαντικό ψηφίο τους ίσο με 1 και το λιγότερο σημαντικό ψηφίο τους ίσο με 0

Απόδειξη. Ένα ελάττωμα f στην μονάδα συμπληρώματος ως προς 2 ενεργοποιείται μόνο όταν η απόλυτη τιμή της διαφοράς των εκθετών είναι 0, δηλαδή $|E_A - E_B| = 0$, και η πραγματική πράξη είναι αφαίρεση. Εάν $|E_A - E_B| = 1$ τότε η διαφορά των σημειωμένων θα είναι θετική, αφού

$$\left. \begin{array}{l} 1 \leq A < 2 \\ 0.5 \leq B < 1 \end{array} \right\} \Rightarrow 0 < A - B < 1.5,$$

και επομένως η έξοδος της μονάδας συμπληρώματος δε θα χρησιμοποιηθεί. Έτσι προκύπτει το πρώτο μέρος του λήμματος. Το δεύτερο μέρος του λήμματος είναι προφανές διότι για να προκύψει αρνητική διαφορά θα πρέπει: $1.F_A - 1.F_B < 0 \Rightarrow 1.F_A < 1.F_B$.

Αφού $|E_A - E_B| = 0$ δεν πρόκειται να συμβεί προ-κανονικοποίηση (δεξιά ολίσθηση). Άρα το λιγότερο σημαντικό ψηφίο της εισόδου της μονάδας, δηλαδή το ψηφίο φρουρός, G , θα είναι 0. Επίσης, το περισσότερο σημαντικό ψηφίο θα είναι 1 αφού το αποτέλεσμα του αφαιρέτη των σημειωμένων θα είναι αρνητικό. \square

Το παρακάτω λήμμα καθορίζει την παρατηρησιμότητα των εσφαλμένων τιμών στις εξόδους της μονάδας συμπληρώματος ως προς 2.

Λήμμα 9. Ένα ελάττωμα f στην μονάδα συμπληρώματος ως προς 2 που έχει ενεργοποιηθεί θα ανιχνεύεται πάντα.

Απόδειξη. Ένα ελάττωμα f στη μονάδα συμπληρώματος ως προς 2 δεν μπορεί να ανιχνευθεί όταν $E = E_f$ και $1.F = 1.F_f$, όπου η τελική τιμή του σημειωμένου και του εκθέτη στο σύντομο μονοπάτι δίνονται από τις εξισώσεις 4.7 και 4.8 αντίστοιχα. Επομένως για να μην είναι δυνατή η ανίχνευση ενός ελαττώματος f στην μονάδα συμπληρώματος ως προς 2 θα πρέπει:

$$E = E_f \Rightarrow e - LZC(X) = e - LZC(X_f) \Rightarrow LZC(X) = LZC(X_f),$$

όπου X και X_f η ορθή και η εσφαλμένη έξοδος της μονάδας συμπληρώματος, αντίστοιχα. Επιπλέον, θα πρέπει να ισχύει:

$$1.F = 1.F_f \Rightarrow Z + r = Z_f + r_f,$$

όπου Z η έξοδος της μονάδας στρογγυλοποίησης του σύντομου μονοπατιού και r ο δυαδικός συντελεστής που καθορίζει αν πραγματοποιείται στρογγυλοποίηση. Αντίστοιχα ορίζονται και οι μεταβλητές με δείκτη f για την εσφαλμένη περίπτωση. Η ορθή έξοδος X θα έχει τουλάχιστον ένα αρχικό μηδενικό αφού για τους αριθμούς κινητής υποδιαστολής a, b ισχύει :

$$\left. \begin{array}{l} 1 \leq a < 2 \\ 1 \leq b < 2 \end{array} \right\} \Rightarrow 0 < |a - b| < 1.$$

Άρα θα συμβεί αριστερή ολίσθηση κατά ένα τουλάχιστον ψηφίο. Η μετά-κανονικοποίηση και η στρογγυλοποίηση είναι αμοιβαία αποκλειόμενες πράξεις στο σύντομο μονοπάτι. Επομένως, η ορθή έξοδος δε θα χρειαστεί στρογγυλοποίηση. Άρα $r = 0$. Η εσφαλμένη έξοδος X_f δε χρειάζεται ούτε αυτή στρογγυλοποίηση αφού $LZC(X) = LZC(X_f) \neq 0$ (δηλαδή και η εσφαλμένη έξοδος θα ολισθήσει). Άρα $r_f = 0$. Επιπλέον, εξαιτίας της ολίσθησης το ψηφίο φρουρός του X και του X_f θα είναι 0. Άρα για να μην είναι δυνατή η ανίχνευση ενός ελαττώματος f θα πρέπει :

$$Z = Z_f \Rightarrow Y = Y_f \Rightarrow 2^{LZC(X)} X = 2^{LZC(X_f)} X_f,$$

το οποίο είναι αδύνατο αφού $X \neq X_f$ και $LZC(X) = LZC(X_f)$. Επομένως όλα τα ελαττώματα της μονάδας συμπληρώματος ως προς 2 που έχουν ενεργοποιηθεί ανιχνεύονται. \square

4.3.5 Αυξητής στρογγυλοποίησης σημαίνονομου (Σύντομο Μονοπάτι)

Ο αυξητής (incrementer) του σημαίνονομου του σύντομου μονοπατιού χρησιμοποιείται για να προσαυξήσει το σημαίνονομο όταν αυτό χρειάζεται στρογγυλοποίηση. Υπενθυμίζουμε πως στο σύντομο μονοπάτι δεν υπάρχει περίπτωση υπερχείλισης κατά τη στρογγυλοποίηση. Το μέγεθος του αυξητή είναι N , όπου N είναι το πλήθος των ψηφίων του σημαίνονομου.

Το παρακάτω λήμμα καθορίζει την ελεγχσιμότητα των ελαττωμάτων f στον αυξητή στρογγυλοποίησης της διαφοράς.

Λήμμα 10. Ένα ελάττωμα f στον αυξητή του σημαίνονομου μπορεί να ενεργοποιηθεί μόνο όταν:

1. Η απόλυτη τιμή της διαφοράς των εκθετών είναι 1 και η πραγματική πράξη είναι αφαίρεση.
2. Η διαφορά των σημαίνονομων δεν έχει αρχικά μηδενικά, δηλαδή $LZC(1.F_A - 1.F_B) = 0$.
3. Η έξοδος του ολισθητή μετά-κανονικοποίησης έχει τα λιγότερο σημαντικά του ψηφία ίσα με 1. Απαραίτητη προϋπόθεση είναι τα σημαίνονομα των αριθμών κινητής υποδιαστολής να έχουν μια από τις ακόλουθες μορφές:

$$(α') 1.F_A = 1.x_2x_2x_{21} \dots x_2x_10 \text{ και } 1.F_B = 1.x_2x_2x_{21} \dots x_201 \text{ ή}$$

$$(\beta') 1.F_A = 1.x_{22}x_{21} \dots x_{11} \text{ και } 1.F_B = 1.x_{22}x_{21} \dots x_{211}$$

Απόδειξη. Για να ενεργοποιηθεί ένα ελάττωμα f στον αυξητή της στρογγυλοποίησης, η διαφορά των σημειόμενων θα πρέπει να είναι σε κανονική μορφή, δηλαδή να μην έχει αρχικά μηδενικά, $LZC(1.F_A - 1.F_B) = 0$. Αυτό σημαίνει πως θα πρέπει να ισχύει $|E_A - E_B| = 1$. Σε αντίθετη περίπτωση, δηλαδή εάν, $|E_A - E_B| = 0$ ισχύει :

$$\left. \begin{array}{l} 1 \leq A < 2 \\ 1 \leq B < 2 \end{array} \right\} \Rightarrow 0 < |A - B| < 1.$$

Άρα η διαφορά έχει αρχικά μηδενικά και δεν πραγματοποιείται στρογγυλοποίηση.

Η στρογγυλοποίηση της διαφοράς πραγματοποιείται μόνο όταν τα δύο λιγότερο σημαντικά ψηφία της διαφοράς έχουν την τιμή 1 (βλέπε Κεφάλαιο 3). Καθώς η διαφορά των εκθετών είναι 1 το σημειόμενο του αριθμού B , $1.F_B$, θα ολισθήσει μια θέση δεξιά. Επομένως τα δύο λιγότερο σημαντικά ψηφία των σημειόμενων της μορφής (3α') είναι $0||0$ και $0||1$. Άρα τα δύο λιγότερο σημαντικά ψηφία της διαφοράς είναι $1||1$. Αντίστοιχα, τα λιγότερο σημαντικά ψηφία για τα σημειόμενα της μορφής (3β') είναι $1||0$ και $1||1$. Επομένως τα λιγότερο σημαντικά ψηφία της διαφοράς είναι $1||1$. Το ψηφίο δεξιά του συμβόλου $||$ είναι το ψηφίο φρουρός G . \square

Το λήμμα που ακολουθεί ορίζει την παρατηρησιμότητα των εσφαλμένων τιμών στις εξόδους του αυξητή.

Λήμμα 11. Κάθε ελάττωμα f του αυξητή που έχει ενεργοποιηθεί ανιχνεύεται.

Απόδειξη. Η έξοδος του αυξητή, Z , οδηγείται στον πολυπλέκτη επιλογής μονοπατιού και είναι άμεσα παρατηρήσιμη στις κύριες εξόδους του αθροιστή κινητής υποδιαστολής. Επομένως, κάθε ελάττωμα f που ενεργοποιείται ανιχνεύεται. \square

4.3.6 Αυξητής στρογγυλοποίησης σημειόμενου (Μακρινό Μονοπάτι)

Ο αυξητής (incrementer) του σημειόμενου στο μακρινό μονοπάτι χρησιμοποιείται για να προσαυξήσει το σημειόμενο όταν αυτό χρειάζεται στρογγυλοποίηση. Έχει μέγεθος N , όπου N είναι το πλήθος των ψηφίων του σημειόμενου. Το κρατούμενο εξόδου της μονάδας αύξησης ελέγχεται ώστε να διαπιστωθεί εάν έχει συμβεί υπερχείλιση κατά την προσαύξηση.

Λήμμα 12. Ένα ελάττωμα f στην μονάδα αύξησης του σημειόμενου μπορεί να ενεργοποιηθεί όταν:

1. Η απόλυτη τιμή της διαφοράς των εκθετών είναι ίση με το πλήθος των ψηφίων του σημειόμενου, δηλαδή $|E_A - E_B| = 24$ για απλή ακρίβεια και $|E_A - E_B| = 53$ για διπλή ακρίβεια.

2. Το σημαίνόμενο του αριθμού κινητής υποδιαστολής έχει όλα τα ψηφία του ίσα με 1, δηλαδή είναι της μορφής $1.F_B = 1.11 \dots 11$.
3. Τα διανύσματα δοκιμής έχουν το περισσότερο σημαντικό ψηφίο τους ίσο με 1. Επίσης, αποτελούν το κλασματικό μέρος του αριθμού κινητής υποδιαστολής A .

Απόδειξη. Τα δύο πρώτα μέρη του λήμματος εξασφαλίζουν πως το αποτέλεσμα του ακέραιου αθροιστή-αφαιρέτη θα χρειαστεί στρογγυλοποίηση. Το σημαίνόμενο του αριθμού B , $1.F_B$, θα ολισθήσει δεξιά και τα ψηφία φρουροί G, R, T θα λάβουν όλα την τιμή 1. Έτσι το σημαίνόμενο του αριθμού B έπειτα από την προ-κανονικοποίηση θα έχει τη μορφή $1.F_B = 0.00 \dots 00 \parallel 111$. Το αποτέλεσμα του ακέραιου αθροιστή-αφαιρέτη θα βρίσκεται σε κανονική μορφή και θα χρειάζεται στρογγυλοποίηση, διότι $G(R + T + f_0) = 1$. Υποθέτουμε πως πραγματοποιείται στρογγυλοποίηση προς τον πλησιέστερο άρτιο. Επιπλέον, τα διανύσματα δοκιμής, δηλαδή η είσοδος της μονάδας αύξησης, θα πρέπει να έχουν το σημαντικότερο ψηφίο τους ίσο με 1, αφού το σημαίνόμενο που φθάνει στην είσοδο της μονάδας αύξησης βρίσκεται πάντα σε κανονική μορφή. Τα διανύσματα δοκιμής αποτελούν το κλασματικό μέρος του αριθμού κινητής υποδιαστολής A διότι το άθροισμα του $1.F_A$ με το κανονικοποιημένο $1.F_B$ αποτελεί την είσοδο της μονάδας αύξησης του σημαίνόμενου. \square

Το λήμμα που ακολουθεί εξασφαλίζει πως κάθε ελάττωμα στην μονάδα αύξησης του σημαίνόμενου το οποίο έχει ενεργοποιηθεί θα ανιχνεύεται πάντοτε.

Λήμμα 13. Κάθε ελάττωμα f στην μονάδα αύξησης του σημαίνόμενου που ενεργοποιείται ανιχνεύεται πάντα.

Απόδειξη. Η έξοδος της μονάδας αύξησης οδηγείται στον πολυπλέκτη επιλογής μονοπατιού και από εκεί είναι άμεσα παρατηρήσιμη στις εξόδους του αθροιστή κινητής υποδιαστολής. \square

4.3.7 Αυξητής εκθέτη - Υπερχείλιση Στρογγυλοποίησης (Μακρινό Μονοπάτι)

Ο αυξητής (incrementer) του εκθέτη χρησιμοποιείται για να προσαυξήσει τον εκθέτη όταν κατά τη στρογγυλοποίηση του σημαίνόμενου έχει συμβεί υπερχείλιση. Υπερχείλιση έχει συμβεί όταν το κρατούμενο εξόδου της μονάδας αύξησης του σημαίνόμενου είναι ίσο με 1.

Λήμμα 14. Ένα ελάττωμα f στην μονάδα αύξησης του εκθέτη μπορεί να ενεργοποιηθεί όταν :

- το κανονικοποιημένο αποτέλεσμα του ακέραιου αθροιστή-αφαιρέτη υπερχειλίζει κατά τη στρογγυλοποίηση. Η υπερχείλιση εξασφαλίζεται όταν :
 - το σημαίνόμενο του αριθμού κινητής υποδιαστολής A έχει τη μορφή $1.F_A = 1.00 \dots 00$, και

- το σημαίνόμενο του αριθμού κινητής υποδιαστολής B έχει τη μορφή $1.F_B = 1.11\dots 11$, και
- η απόλυτη τιμή της διαφοράς των εκθέτων είναι 1 και η πραγματική πράξη είναι πρόσθεση, $|E_A - E_B| = 1$.
- Τα διανύσματα δοκιμής μπορούν να λάβουν οποιαδήποτε τιμή εκτός της τιμής $11\dots 11$ και $00\dots 00$, οι οποίες αποτελούν ειδικές τιμές για τον εκθέτη ενός αριθμού κινητής υποδιαστολής.

Απόδειξη. Θέτοντας τα σημαίνόμενα στις μορφές που περιγράφονται στο λήμμα εξασφαλίζεται πως το άθροισμα των σημανόμενων θα χρειάζεται στρογγυλοποίηση κατά την οποία θα συμβεί υπερχείλιση, διότι :

$$\begin{array}{r}
 \text{GRS} \\
 1.00\dots 00\|000 \\
 + 0.11\dots 11\|100 \\
 \hline
 1.11\dots 11\|100
 \end{array}$$

άρα αποτέλεσμα χρειάζεται στρογγυλοποίηση διότι, $G(R + T + f_0) = 1$. Επειδή όλα τα ψηφία του είναι 1 το κρατούμενο εξόδου της μονάδας αύξησης του σημανόμενου είναι 1, επομένως συμβαίνει υπερχείλιση. \square

Λήμμα 15. Κάθε ελάττωμα f στην μονάδα αύξησης του εκθέτη που έχει ενεργοποιηθεί ανιχνεύεται σε κάθε περίπτωση.

Απόδειξη. Η έξοδος της μονάδας αύξησης του εκθέτη αποτελεί τον τελικό εκθέτη του αποτελέσματος της πράξης κινητής υποδιαστολής, άρα κάθε ελάττωμα f που έχει ενεργοποιηθεί ανιχνεύεται πάντα. \square

4.3.8 Αυξητής - Μειωτής Κανονικοποίησης εκθέτη (Μακρινό Μονοπάτι)

Η μονάδα αύξησης/μείωσης του εκθέτη χρησιμοποιείται κατά το στάδιο της μετά-κανονικοποίησης του αποτελέσματος του ακέραίου αθροιστή/αφαιρέτη. Η είσοδος της μονάδας είναι ο μεγαλύτερος εκθέτης των αριθμών κινητής υποδιαστολής A , και B όπως αυτός προκύπτει από την μονάδα εναλλαγής (Swap). Η μονάδα έχει μια επιπλέον είσοδο ελέγχου n οποία ορίζει αν ο εκθέτης θα αυξηθεί ή θα μειωθεί ή θα μείνει ανέπαφος. Υπενθυμίζουμε πως στο μακρινό μονοπάτι η μετά-κανονικοποίηση (όταν χρειάζεται) πραγματοποιείται είτε με την ολίσθηση του αποτελέσματος του αθροιστή/αφαιρέτη κατά μια θέση αριστερά και μείωση του εκθέτη είτε με ολίσθηση κατά μία θέση δεξιά και αύξηση του εκθέτη.

Λήμμα 16. Ένα ελάττωμα f στη μονάδα αύξησης/μείωσης του εκθέτη μπορεί να ενεργοποιηθεί όταν τα διανύσματα δοκιμής (δηλαδή ο εκθέτης) λαμβάνουν οποιαδήποτε τιμή εκτός των ειδικών τιμών $00\dots 00$ και $11\dots 11$. Αν το διάνυσμα δοκιμής που θα εφαρμοστεί :

- απαιτεί την αύξηση του εκθέτη, τότε η έξοδος του ακέραιου αθροιστή/αφαιρέτη θα πρέπει να χρειάζεται δεξιά ολίσθηση. Αυτό μπορεί να γίνει προσθέτοντας δύο αριθμούς κινητής υποδιαστολής για τους οποίους ισχύει $|E_A - E_B| = 0$.
- απαιτεί τη μείωση του εκθέτη τότε η έξοδος του ακέραιου αθροιστή/αφαιρέτη θα πρέπει να χρειάζεται αριστερή ολίσθηση.
- δεν απαιτεί τη μεταβολή του εκθέτη, τότε η έξοδος του ακέραιου αθροιστή/αφαιρέτη θα πρέπει να βρίσκεται σε κανονική μορφή.

Λήμμα 17. Κάθε ελάττωμα f στην μονάδα αύξησης/μείωσης του εκθέτη που ενεργοποιείται ανιχνεύεται.

Απόδειξη. Σύμφωνα με την εξίσωση 4.4 για να είναι αδύνατη η ανίχνευση ενός ελαττώματος στην μονάδα αύξησης/μείωσης θα πρέπει να ισχύει :

$$E = E_{faulty} \Rightarrow e + \overrightarrow{p} - \overleftarrow{p} + r^{OV} = e_f + \overrightarrow{p}_f - \overleftarrow{p}_f + r_f^{OV}.$$

Τα ελαττώματα στην μονάδα αύξησης/μείωσης δεν επηρεάζουν του συντελεστές \overrightarrow{p} , \overleftarrow{p} , r^{OV} , οι οποίοι είναι ίσοι για την ορθή και εσφαλμένη έξοδο της μονάδας. Επομένως, ο μόνος τρόπος για να είναι αδύνατη η ανίχνευση των ελαττωμάτων είναι να ισχύει $e = e_f$, που δεν ισχύει. \square

4.3.9 Ολισθητής Προ-κανονικοποίησης - Γεννήτορας Ψηφίων φρουρών (Μακρινό Μονοπάτι)

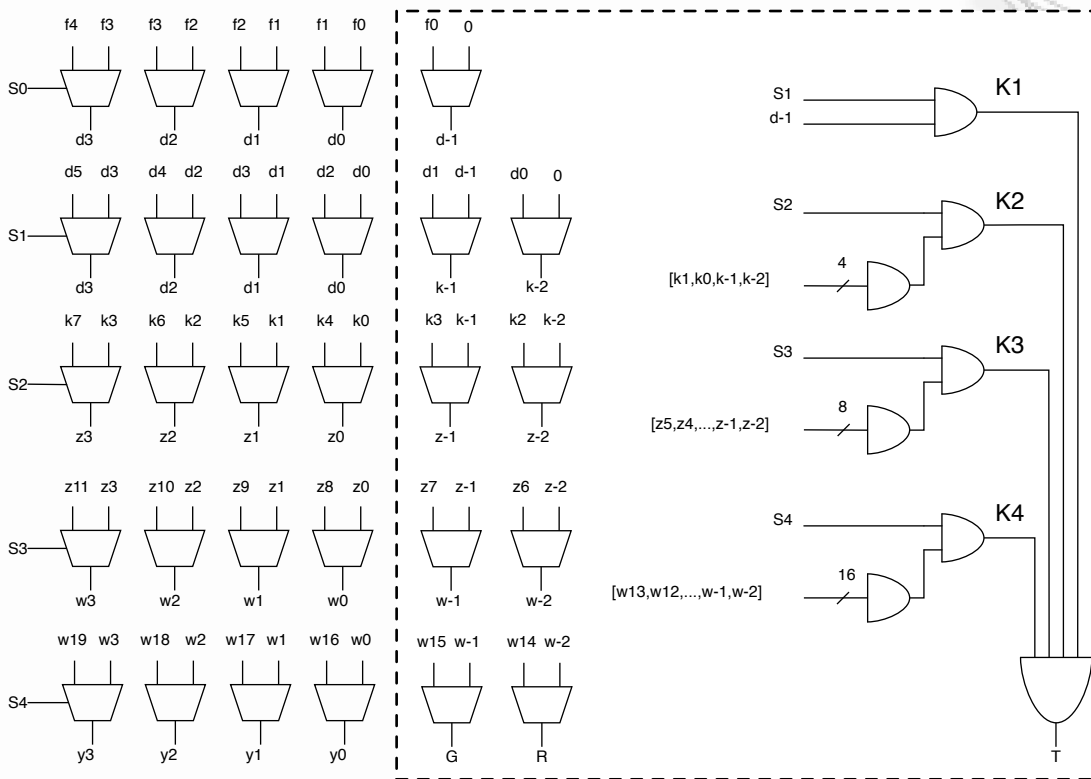
Ο ολισθητής προ-κανονικοποίησης του μακρινού μονοπατιού είναι ένας αριστερός ολισθητής ο οποίος ολισθαίνει το σημειώμενο του αριθμού κινητής υποδιαστολής με το μικρότερο εκθέτη τόσες θέσεις όσο και η απόλυτη διαφορά των εκθετών. Η μονάδα ολίσθησης συμπληρώνεται με επιπλέον λογικές πύλες ώστε να παράγει τα ψηφία φρουρούς, G, R, T , όπως φαίνεται στην Εικόνα 4.2.

Οι είσοδοι του ολισθητή προ-κανονικοποίησης έχουν μέγεθος $N + 1$ για το σημειώμενο, όπου είναι το πλήθος των ψηφίων του σημειώμενου και K για τη διαφορά των εκθετών όπου K είναι το πλήθος των ψηφίων του εκθέτη. Η έξοδος του ολισθητή προ-κανονικοποίησης έχει μέγεθος $N + 3$.

Τα λήμματα που ακολουθούν παρουσιάζουν τις συνθήκες ελεγχιμότητας και παρατηρησιμότητας για την μονάδα προ-κανονικοποίησης.

Λήμμα 18. Οι είσοδοι της μονάδας προ-κανονικοποίησης είναι άμεσα ελέγξιμοι. Τα διανύσματα δοκιμής θα πρέπει να έχουν το σημαντικότερο ψηφίο τους ίσο με 1.

Απόδειξη. Η θέση της μονάδας προ-κανονικοποίησης είναι τέτοια που δεν υπάρχουν περιορισμοί ως προς τις τιμές που μπορούν να λάβουν οι είσοδοί της. Ο μοναδικός περιορισμός



Εικόνα 4.2: Ολισθητής προ-κανονικοποίησης μακρινού μονοπατιού και γέννηση ψηφίων φρουρών G, R, T .

είναι πως το σημαντικότερο ψηφίο της εισόδου θα πρέπει να είναι ίσο με 1 αφού οι αριθμοί κινητής υποδιαστολής βρίσκονται σε κανονική μορφή. □

Λήμμα 19. Οι συνθήκες παρατηρησιμότητας για τις εσφαλμένες αποκρίσεις της μονάδας προ-κανονικοποίησης είναι ίδιες με αυτές του ακέραιου αθροιστή/αφαιρέτη του μακρινού μονοπατιού.

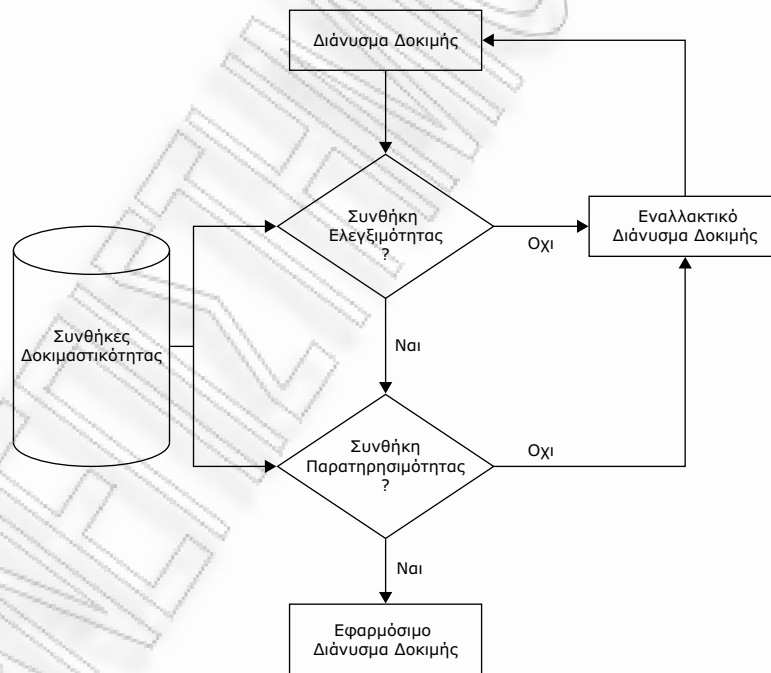
Απόδειξη. Η έξοδος της μονάδας προ-κανονικοποίησης αποτελεί είσοδο του ακέραιου αθροιστή/αφαιρέτη του μακρινού μονοπατιού. Σε κάθε περίπτωση μια εσφαλμένη τιμή στην έξοδο της μονάδας προ-κανονικοποίησης θα διαδοθεί και στις εξόδους του αθροιστή/αφαιρέτη των σημανόμενων. Ας υποθέσουμε πως η ορθή έξοδος της μονάδας προ-κανονικοποίησης είναι y και η εσφαλμένη y_f . Η έξοδος του ακέραιου αθροιστή/αφαιρέτη είναι ίση με $x \pm y$ και $x \pm y_f$ για την ορθή και εσφαλμένη περίπτωση, αντίστοιχα. Ο μόνος τρόπος για να μην είναι δυνατή η διάδοση της εσφαλμένης τιμής y_f μέσω του αθροιστή/αφαιρέτη είναι να ισχύει $x \pm y = x \pm y_f \Rightarrow y = y_f$, το οποίο είναι αδύνατο. Επομένως, η μελέτη για την εξαγωγή των συνθηκών παρατηρησιμότητας μιας εσφαλμένης τιμής στην έξοδο της μονάδας προ-

κανονικοποίησης είναι η ίδια με τη μελέτη παρατηρησιμότητας μιας εσφαλμένης τιμής στην έξοδο του ακέραιου αθροιστή/αφαιρέτη. □

4.4 Εφαρμογή των συνθηκών δοκιμαστικότητας

Η ανάπτυξη διανυσμάτων δοκιμής ξεκινά θεωρώντας πως υπάρχουν διαθέσιμα σύνολα διανυσμάτων δοκιμής για τις υπομονάδες του αθροιστή/αφαιρέτη κινητής υποδιαστολής. Η εξαγωγή των διανυσμάτων για τις υπομονάδες πραγματοποιείται θεωρώντας πως δεν υπάρχουν προβλήματα σχετικά με την ελεγχιμότητα και την παρατηρησιμότητα, δηλαδή οι είσοδοι είναι άμεσα ελέγξιμοι και οι έξοδοι άμεσα παρατηρήσιμοι.

Έπειτα, κάθε διάνυσμα δοκιμής ελέγχεται σύμφωνα με τις συνθήκες δοκιμαστικότητας της υπομονάδας της οποίας ανιχνεύει ελαττώματα, όπως φαίνεται στην Εικόνα 4.3.



Εικόνα 4.3: Εξαγωγή διανυσμάτων δοκιμής με συνθήκες δοκιμαστικότητας.

Αν ένα διάνυσμα δοκιμής δεν μπορεί να εφαρμοστεί εξαιτίας των περιορισμών ελεγχιμότητας τότε γίνεται η επιλογή ενός άλλου διανύσματος το οποίο ενεργοποιεί το ίδιο ελάττωμα ή εναλλακτικά πραγματοποιείται νέα εξαγωγή διανυσμάτων. Στην περίπτωση κατά την οποία κανένα διάνυσμα δεν ικανοποιεί τις συνθήκες ελεγχιμότητας της μονάδας, τότε τα ελαττώματα

χαρακτηρίζονται μη-ελέγξιμα (uncontrollable).

Στη συνέχεια, τα διανύσματα που ικανοποιούν τις συνθήκες ελεγχσιμότητας της υπομονάδας, ελέγχονται με βάση τις συνθήκες παρατηρησιμότητας. Για την αξιολόγηση των διανυσμάτων δοκιμής είναι απαραίτητο να είναι γνωστή η επίδραση του ελαττώματος (fault effect) της υπομονάδας στις εξόδους της, δηλαδή να είναι γνωστή η εσφαλμένη τιμή που προκύπτει εξαιτίας του ελαττώματος. Η εσφαλμένη τιμή προκύπτει από τη προσομοίωση των ελαττωμάτων της υπομονάδας. Εάν η διάδοση του αποτελέσματος του ελαττώματος (fault effect propagation) στις εξόδους της μονάδας κινητής υποδιαστολής δεν είναι εφικτή επιλέγεται ένα νέο διάνυσμα δοκιμής. Σε περίπτωση που κανένα διάνυσμα δοκιμής δεν ικανοποιεί τις συνθήκες παρατηρησιμότητας της υπομονάδας τα ελαττώματα χαρακτηρίζονται ως μη-παρατηρήσιμα (unobservable).

Όλα τα διανύσματα που ικανοποιούν τις συνθήκες ελεγχσιμότητας και δοκιμασιμότητας χαρακτηρίζονται ως εφαρμόσιμα και μετατρέπονται σε αριθμούς κινητής υποδιαστολής οι οποίοι μπορούν να εφαρμοστούν στις εισόδους της μονάδας κινητής υποδιαστολής και να ανιχνεύσουν ελαττώματα στις υπομονάδες.

Ένα υποσύνολο του συνόλου διανυσμάτων δοκιμής το οποίο ανιχνεύει ελαττώματα προσκόλλησης στη δενδρική δομή γέννησης κρατουμένου απαιτεί την εφαρμογή 27 διανυσμάτων δοκιμής όταν ο αθροιστής/αφαιρέτης πραγματοποιεί αφαίρεση (στην απλή ακρίβεια ο αθροιστής/αφαιρέτης έχει μέγεθος 27 ψηφίων, 24 ψηφία για το σημανόμενο και 3 ψηφία για τα ψηφία φρουρούς).

X	Y
(000...00	100...00)
(000...00	010...00)
(000...00	001...00)
...	...
(000...00	000...01)

Το σύνολο των παραπάνω διανυσμάτων μπορεί να μετατραπεί σε αριθμούς κινητής υποδιαστολής θέτοντας το πρόσημο του αριθμού κινητής υποδιαστολής A ίσο με 0 και το πρόσημο του αριθμού B ίσο με 1, ώστε η πραγματική πράξη να είναι αφαίρεση. Σύμφωνα με την ανάλυση της ελεγχσιμότητας του ακέραιου αθροιστή/αφαιρέτη για το $\text{sig}A$, το πρώτο μέρος των διανυσμάτων (στην αριστερή στήλη- X) δε μπορεί να εφαρμοστεί, αφού το σημαντικότερο ψηφίο είναι 0. Η συνθήκη ελεγχσιμότητας, εξίσωση 4.1 ορίζει ότι το σημαντικότερο ψηφίο για το $\text{sig}A$ θα πρέπει να είναι 1. Επομένως, όλα τα ελαττώματα του ακέραιου αθροιστή/αφαιρέτη τα οποία ανιχνεύονται με διανύσματα δοκιμής των οποίων το σημαντικότερο ψηφίο είναι 1, χαρακτηρίζονται ως μη ανιχνεύσιμα, όταν ο ακέραιος αθροιστής/αφαιρέτης βρίσκεται στην μονάδα κινητής υποδιαστολής. Η τιμή 00...00 εφαρμόζεται μόνο για το κλασματικό μέρος του αριθμού A .

για αναπαράσταση απλής ακρίβειας. Όλα τα ψηφία από κάθε επίπεδο, τα οποία απορρίπτονται κατά την ολίσθηση οδηγούνται σε μια πύλη OR, όπως φαίνεται στην Εικόνα 4.2. Τα ψηφία K_1, K_2, K_3 , και K_4 οδηγούνται εκ νέου σε μια πύλη OR της οποίας η έξοδος είναι το επίμονο ψηφίο.

Ένα ελάττωμα f στη μονάδα υπολογισμού του επίμονου ψηφίου ανιχνεύεται μόνο όταν επηρεάζει τη στρογγυλοποίηση. Υπενθυμίζουμε πως η στρογγυλοποίηση προς τον πλησιέστερο άρτιο εξαρτάται από το αποτέλεσμα της έκφρασης $G(L + R + T)$, όπου L είναι το λιγότερο σημαντικό ψηφίο του σημεινόμενου αριστερά των ψηφίων φρουρών. Επομένως για τα διανύσματα δοκιμής θα πρέπει να ισχύει $G = 1$ και $L = R = 0$, ώστε να είναι δυνατή η διάδοση των ελαττωμάτων στη λογική του επίμονου ψηφίου.

Η δοκιμή της πύλης AND στο επάνω μέρος της Εικόνας 4.2 χρειάζεται 3 διανύσματα δοκιμής. Τα διανύσματα θέτουν τις ακόλουθες τιμές στις εισόδους της πύλης: $\{(S_1, d_{-1}) = (01), (10), (11)\}$. Οι αριθμοί κινητής υποδιαστολής που παρουσιάζονται παρακάτω θέτουν τα απαραίτητα διανύσματα στην πύλη AND.

FP Number A (S, E, F)	FP Number B (S, E, F)
0, 00001111, 00...00	0, 00001110, 00...001
0, 00001111, 00...00	0, 00001101, 00...010
0, 00001111, 00...00	0, 00001100, 00...101

Η τιμή του αριθμού κινητής υποδιαστολής A παραμένει σταθερή σε όλα τα ζεύγη διανυσμάτων δοκιμής. Οι τιμές που λαμβάνουν οι εκθέτες είναι ενδεικτικές. Είναι απαραίτητο η διαφορά των εκθετών να λαμβάνει τις τιμές 1, 2, και 3 διαδοχικά. Η προσαρμογή των διανυσμάτων δοκιμής σε διπλή ακρίβεια γίνεται με την επέκταση του εκθέτη στα 11 ψηφία και του σημεινόμενου στα 52 ψηφία αντίστοιχα.

Τα ζεύγη των πυλών AND και OR δοκιμάζονται θέτοντας την διαφορά των εκθετών $|E_A - E_B| = 2^l + 2$, όπου $l, l \geq 2$, είναι το επίπεδο του ολισθητή από το οποίο απορρίπτονται ψηφία. Το κλασματικό μέρος του αριθμού A τίθεται ίσο με 00...00. Το ψηφίο $2^l + 2$ του κλασματικού μέρους του αριθμού B λαμβάνει την τιμή 1. Με αυτό τον τρόπο σχηματίζεται το πρώτο ζεύγος διανυσμάτων δοκιμής. Τα υπόλοιπα ζεύγη σχηματίζονται προσθέτοντας στο κλασματικό μέρος του αριθμού B την ποσότητα 2^i , όπου $i = 0, 1, \dots, l$. Συνολικά για κάθε ζεύγος πυλών AND και OR εφαρμόζονται $2^l + 1$ ζεύγη διανυσμάτων.

Για παράδειγμα το ζεύγος πυλών AND και OR που δημιουργεί το ψηφίο K_3 χρειάζεται συνολικά $2^3 + 1 = 9$ ζεύγη διανυσμάτων. Για τη διαφορά των εκθετών θα πρέπει να ισχύει $|E_A - E_B| = 2^3 + 2 = 10$. Ο Πίνακας 4.3 παρουσιάζει τα ζεύγη διανυσμάτων δοκιμής για το ζεύγος πυλών που σχηματίζει το ψηφίο K_3 .

Οι αριθμοί κινητής υποδιαστολής είναι ομόσημοι (στον Πίνακα 4.3 είναι θετικοί) ώστε να επεξεργάζονται από το μακρινό μονοπάτι.

Αντίστοιχα, για το ζεύγος πυλών που σχηματίζουν το ψηφίο K_5 (στον αθροιστή διπλής

FP Number A (S, E, F)	FP Number B (S, E, F)
(0, 00111111, $\overbrace{0\dots0}^{23 \text{ bits}}$)	(0, 00110101, $\overbrace{0\dots0[1]000000000}^{13 \text{ bits}}$)
(0, 00111111, 0...0)	(0, 00110101, 0...0[1]000000001)
(0, 00111111, 0...0)	(0, 00110101, 0...0[1]000000010)
(0, 00111111, 0...0)	(0, 00110101, 0...0[1]000000100)
(0, 00111111, 0...0)	(0, 00110101, 0...0[1]000001000)
(0, 00111111, 0...0)	(0, 00110101, 0...0[1]000010000)
(0, 00111111, 0...0)	(0, 00110101, 0...0[1]000100000)
(0, 00111111, 0...0)	(0, 00110101, 0...0[1]001000000)
(0, 00111111, 0...0)	(0, 00110101, 0...0[1]010000000)

Πίνακας 4.3: Τα ζεύγη διανυσμάτων δοκιμής για το ζεύγος πυλών που σχηματίζουν το ψηφίο K_3 .

ακρίβειας) χρειάζονται συνολικά $2^5 + 1 = 33$ ζεύγη διανυσμάτων δοκιμής. Για τη διαφορά των εκθετών θα πρέπει να ισχύει $|E_A - E_B| = 2^5 + 2 = 34$. Το κλασματικό μέρος του αριθμού A λαμβάνει την τιμή $00\dots00$ και το κλασματικό μέρος του αριθμού B λαμβάνει αρχικά την τιμή $(\underbrace{0\dots0}_{18\text{-ψηφία}} [1]00\dots00)$. Έπειτα το κλασματικό μέρος του αριθμού λαμβάνει διαδοχικά τις τιμές $(0\dots0[1]0\dots01), \dots, (0\dots0[1]010\dots00)$.

Όλα τα παραπάνω διανύσματα που ενεργοποιούν ελαττώματα στην μονάδα υπολογισμού του επίμονου ψηφίου διαδίδουν τις εσφαλμένες τιμές στις εξόδους του αθροιστή κινητής υποδιαστολής, δηλαδή ανιχνεύουν τα ελαττώματα.

Τόσο η ορθή όσο και η εσφαλμένη έξοδος του ακέραιου/αθροιστή αφαιρέτη θα βρίσκονται σε κανονική μορφή επομένως : $\overleftarrow{p} = \overrightarrow{p} = \overleftarrow{p}_f = \overrightarrow{p}_f = 0$. Επιπλέον, μόνο μια από τις ορθές και εσφαλμένες αποκρίσεις του ακέραιου αθροιστή θα χρειάζεται στρογγυλοποίηση, δηλαδή $r = 0$ και $r_f = 1$ ή $r = 1$ και $r_f = 0$. Σύμφωνα με το Λήμμα 1 (συνθήκη 3β') ένα ελάττωμα δεν ανιχνεύεται όταν $|Z - Z_f| = 1$. Ένα ελάττωμα στη μονάδα υπολογισμού του επίμονου ψηφίου δεν μπορεί να επηρεάσει την μεταβλητή Z , αφού η μεταβλητή Z σχηματίζεται απορρίπτοντας τα ψηφία φρουρούς. Επομένως, $Z = Z_f$, και άρα τα ελαττώματα στην μονάδα υπολογισμού του επίμονου ψηφίου ανιχνεύονται από τα διανύσματα δοκιμής. Οι συνθήκες δοκιμαστικότητας, όπως φάνηκε από την παραπάνω ανάλυση, δεν εξαρτώνται από την ακρίβεια αναπαράστασης και χαρακτηρίζονται προσαρμόσιμες σε κάθε ακρίβεια. Σε κανένα σημείο δεν έγινε αναφορά στην ακρίβεια του αθροιστή κινητής υποδιαστολής τόσο στις συνθήκες ελεγχιμότητας όσο και στις συνθήκες παρατηρησιμότητας.

Υπομονάδα	SP-FPadder	DP-FPadder
Conv-A	60	122
Conv-B	60	122
Swap	337	571
<hr/>		
Far-Path		
Shift GRS	368	924
Adder-Subtractor	430	924
Exp. Inc/Dec	59	83
Postnormalizer	98	195
Round Sig. Inc	38	54
Round Exp. Inc	117	267
<hr/>		
Close-Path		
Prenormalizer	55	122
Subtractor	305	685
Negator & MUX	196	434
LZC	119	265
Left Shifter	333	864
Exp. Subtractor	305	138
Round Sig. Inc	105	257
<hr/>		
Path Select	93	160
Exception	119	210
<hr/>		
Total	3225	6869

Πίνακας 4.4: Αποτελέσματα σύνθεσης στη βιβλιοθήκη TSMC 0.18um.

4.5 Πειραματικά Αποτελέσματα

Η αξιολόγηση των συνθηκών δοκιμαστικότητας πραγματοποιήθηκε χρησιμοποιώντας δύο συνδυαστικά (combinational) μοντέλα πρόσθεσης/αφαίρεσης αριθμών κινητής υποδιαστολής δύο μονοπατιών (απλής και διπλής ακρίβειας). Τα κυκλώματα αναπτύχθηκαν με τη βοήθεια της βιβλιοθήκης FPLibrary, η οποία παρέχει ένα αθροιστή κινητής υποδιαστολής δύο μονοπατιών προσαρμόσιμης ακρίβειας καθώς επίσης και μοντέλα μετατροπής (conversion modules) τα οποία επεκτείνουν τους αριθμούς κινητής υποδιαστολής με δύο επιπλέον ψηφία τα οποία δηλώνουν αν ο εν λόγω αριθμός είναι έγκυρος ή αν είναι ειδική τιμή, δηλαδή 0, $\pm\infty$, ή NaN. Οι ακέραιες αριθμητικές μονάδες (αθροιστές, αφαιρέτες, αυξητές, μειωτές) υλοποιήθηκαν σύμφωνα με την αρχιτεκτονική υψηλής ταχύτητας Brent-Kung [98]. Ο Πίνακας 4.4 παρουσιάζει το μέγεθος των αθροιστών κινητής υποδιαστολής απλής ακρίβειας SP-FPadder και διπλής ακρίβειας DP-FPadder σε ισοδύναμες πύλες NAND δύο εισόδων (NAND2x1). Η σύνθεση (logic synthesis) των μοντέλων SP-FPadder και DP-FPadder έγινε στη βιβλιοθήκη τεχνολογίας TSMC 0.18um. Η καθυστέρηση διάδοσης είναι 7,64 ns για τον αθροιστή απλής ακρίβειας SP-FPadder και 9,94 ns για τον αθροιστή διπλής ακρίβειας DP-FPadder. Ο Πίνακας 4.5 παρουσιάζει στοιχεία σχετικά με το πλήθος των ελαττωμάτων των αθροιστών SP-FPadder και DP-FPadder μόνιμης τιμής καθώς και το πλήθος των ελαττωμάτων τα οποία χαρακτηρί-

	SP-FPAdder	DP-FPAdder
Πλήθος ελαττωμάτων	12.208	26.350
Πλεονάζοντα ελαττώματα μόνιμης τιμής	208	263
Πλεονάζοντα ελαττώματα καθυστέρησης μετάβασης	161	223

Πίνακας 4.5: Στατιστικά στοιχεία σχετικά με το πλήθος των ελαττωμάτων.

ζονται ως πλεονάζοντα (redundant faults)². Το πλήθος των ελαττωμάτων μόνιμης τιμής είναι ίσο με το πλήθος των ελαττωμάτων καθυστέρησης μετάβασης. Οι υπόλοιπες γραμμές του Πίνακα 4.5 απαριθμούν το πλήθος των πλεοναζόντων ελαττωμάτων. Συνολικά αναπτύχθηκαν δύο σύνολα διανυσμάτων δοκιμής για τους αθροιστές απλής και τους αθροιστές διπλής ακρίβειας τα οποία αποτελούνται από 622 και 1080 διανύσματα δοκιμής, αντίστοιχα. Οι μέθοδοι εξαγωγής διανυσμάτων δοκιμής για αθροιστές τύπου Brent-Kung [113] και για ολισθητές [114] χρησιμοποιήθηκαν σε συνδυασμό με τις συνθήκες δοκιμασιμότητας για την ανάπτυξη των τελικών συνόλων των διανυσμάτων δοκιμής. Ο Πίνακας 4.6 παρουσιάζει τα ποσοστά κάλυψης ελαττωμάτων για τους αθροιστές απλής (SP-FPAdder) και διπλής ακρίβειας (DP-FPAdder) για τα μοντέλα ελαττωμάτων προσκόλλησης και καθυστέρησης μετάβασης. Οι υπομονάδες για τις οποίες πραγματοποιήθηκε ανάπτυξη διανυσμάτων δοκιμής σημειώνονται με ✓.

Το ίδιο σύνολο διανυσμάτων δοκιμής για τον αθροιστή απλής ακρίβειας αξιολογήθηκε σε ένα τροποποιημένο αθροιστή ώστε να δειχθεί πειραματικά η ανεξαρτησία των συνθηκών δοκιμασιμότητας από την εσωτερική υλοποίηση των υπομονάδων. Συγκεκριμένα, η μονάδα ακέρραιας πρόσθεσης/αφαίρεσης του μακρινού μονοπατιού και η μονάδα ακέρραιας αφαίρεσης του σύντομου μονοπατιού αντικαταστάθηκαν από αντίστοιχες μονάδες υλοποιημένες σύμφωνα με την αρχιτεκτονική Sklansky [100]. Οι ακέρραιοι αθροιστές που είναι υλοποιημένοι σύμφωνα με την αρχιτεκτονική Sklansky είναι ταχύτεροι έναντι εκείνων που είναι υλοποιημένοι σύμφωνα με την αρχιτεκτονική Brent-Kung καθώς διαθέτουν ταχύτερες μονάδες γέννησης των ενδιάμεσων κρατουμένων. Ο νέος αθροιστής απλής ακρίβειας SP-FPAdder_{Sklansky} αποτελείται από 3315 ισοδύναμες πύλες NAND δύο εισόδων (NAND2x1) και χρειάζεται 7,55 ns για την ολοκλήρωση της πράξης κινητής υποδιαστολής. Περιέχει συνολικά 11776 ελαττώματα μόνιμης τιμής και το ποσοστό κάλυψης ελαττωμάτων είναι 98,01%. Το πείραμα αυτό δείχνει πως οι συνθήκες δοκιμασιμότητας είναι ανεξάρτητες της εσωτερικής υλοποίησης των υπομονάδων που αποτελούν τον αθροιστή κινητής υποδιαστολής.

4.5.1 Σύγκριση με άλλες τεχνικές δοκιμής

Η δοκιμασιμότητα των αθροιστών κινητής υποδιαστολής δύο μονοπατιών αξιολογήθηκε με δύο εναλλακτικές μεθόδους δοκιμής, τη δοκιμή βασισμένη σε ψευδοτυχαία διανύσματα δο-

²Πλεονάζοντα ελαττώματα (redundant faults) καλούνται τα ελαττώματα τα οποία δεν επηρεάζουν την λογική συνάρτηση ενός κυκλώματος παρόλο που μπορεί να υπάρχουν στο κύκλωμα.

Component Name	Stuck-At		Transition Delay	
	SP-FPAdder	DP-FPAdder	SP-FPAdder	DP-FPAdder
Conv A ✓	100.00	100.00	100.00	100.00
Conv B ✓	100.00	100.00	100.00	100.00
Swap ✓	98.75	98.08	98.65	97.96
Far-Path				
Shifter GRS ✓	99.64	99.79	99.63	99.78
Adder/Subtractor ✓	99.90	99.95	99.89	99.95
Exp Inc/Dec ✓	100.00	100.00	97.62	95.83
Postnormalizer	98.10	99.05	97.73	98.39
Round Sig Inc ✓	98.20	99.78	98.20	99.55
Round Exp Inc ✓	100.00	98.70	88.98	94.44
Close-Path				
Prenormalizer	100.00	100.00	100.00	100.00
Subtractor ✓	100.00	100.00	100.00	100.00
Negator & Mux ✓	100.00	100.00	100.00	98.85
LZC	100.00	99.79	99.76	99.47
Left-Shifter ✓	97.82	98.77	97.64	98.66
Exp Sub ✓	97.27	76.15	97.51	74.20
Round Sig Inc ✓	100.00	100.00	99.70	100.00
Path Select	97.54	97.47	95.49	93.76
Exception Unit	88.92	87.38	81.74	82.19
Total	98.09	97.94	97.49	97.38

Πίνακας 4.6: Ποσοστά κάλυψης ελαττωμάτων μόνιμης τιμής (stuck-at) και καθυστέρησης μετάβασης (transition delay) για αθροιστές κινητής και διπλής ακρίβειας (DP-FPAdder).

κιμής και τη δοκιμή βασισμένη σε διανύσματα που εξάγονται από αυτόματα εργαλεία δοκιμής. Οι μέθοδοι δοκιμής εφαρμόστηκαν σε αθροιστές απλής και σε αθροιστές διπλής ακρίβειας τόσο για το μοντέλο ελαττωμάτων προσκόλλησης όσο και για το μοντέλο ελαττωμάτων καθυστέρησης μετάβασης.

Το πρώτο πείραμα αξιολόγησης περιλαμβάνει την αξιολόγηση της δοκιμαστικότητας των αθροιστών κινητής υποδιαστολής με ψευδοτυχαία διανύσματα όπως αυτά παράγονται από ένα γραμμικό καταχωρητή ολίσθησης με ανάδραση (LFSR). Οι Rajski και Tyszer στην εργασία [106] αναφέρουν πως οι αθροιστές κινητής υποδιαστολής δεν είναι εύκολα δοκιμάσιμοι με ψευδοτυχαία διανύσματα δοκιμής και πως είναι απαραίτητες μετατροπές στα κυκλώματα ώστε να βελτιωθεί η δοκιμαστικότητα. Η διαπίστωσή τους επιβεβαιώνεται από τα πειραματικά αποτελέσματα που παρουσιάζονται στον Πίνακα 4.7. Όπως φαίνεται Πίνακα 4.7, όσα ψευδοτυχαία διανύσματα και αν εφαρμοστούν, ακόμα και ένα εκατομμύριο, τα ποσοστά κάλυψης ελαττωμάτων μόνιμης τιμής και καθυστέρησης μετάβασης για τους αθροιστές απλής (SP-FPAdder) και αθροιστές διπλής ακρίβειας (DP-FPAdder) δεν ξεπερνούν τα όρια του 90% και 85% αντίστοιχα.

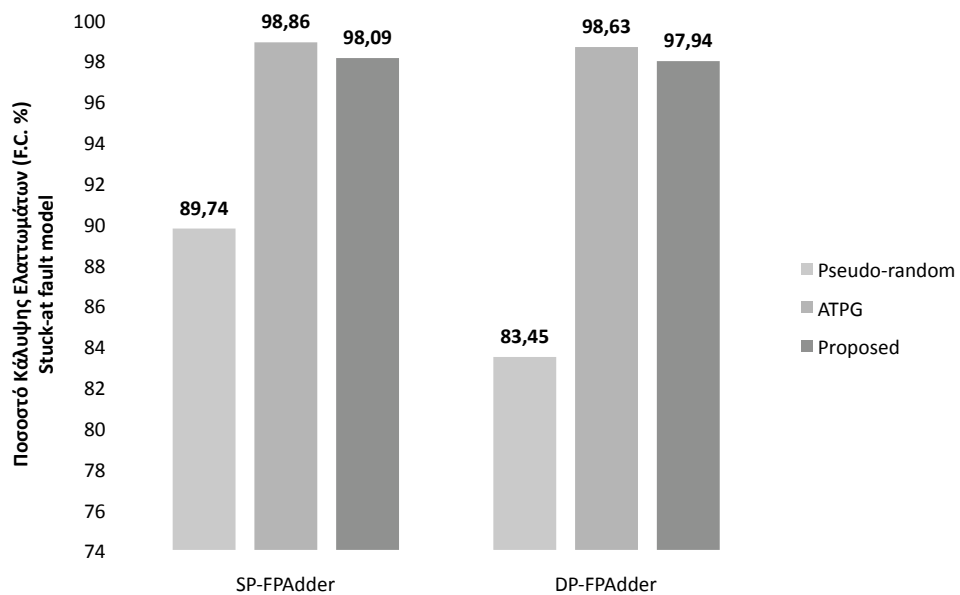
Το δεύτερο πείραμα περιλαμβάνει την αξιολόγηση της δοκιμαστικότητας των αθροιστών κινητής υποδιαστολής με διανύσματα δοκιμής όπως αυτά παράγονται από αυτόματα εργαλεία

Patterns	Stuck-At T.C(%)		Transition Delay T.C(%)	
	SP-FPAdder	DP-FPAdder	SP-FPAdder	DP-FPAdder
1000	75.39	44.01	72.90	40.41
10000	82.76	74.77	80.53	75.52
25000	88.14	81.46	87.45	79.21
100000	88.72	82.06	87.97	79.74
500000	89.51	82.97	88.77	80.88
1000000	89.74	83.45	89.00	81.42

Πίνακας 4.7: Ποσοστά κάλυψης ελαττωμάτων των ψευδοτυχαίων διανυσμάτων δοκιμής.

εξαγωγής διανυσμάτων δοκιμής. Τα αυτόματα εργαλεία εξαγωγής διανυσμάτων δοκιμής έχουν μεγάλη αποτελεσματικότητα όσον αφορά την εξαγωγή διανυσμάτων δοκιμής όταν πρόκειται για συνδυαστικά κυκλώματα, δηλαδή επιτυγχάνουν υψηλά ποσοστά κάλυψης ειδικά για το μοντέλο ελαττωμάτων μόνιμης τιμής.

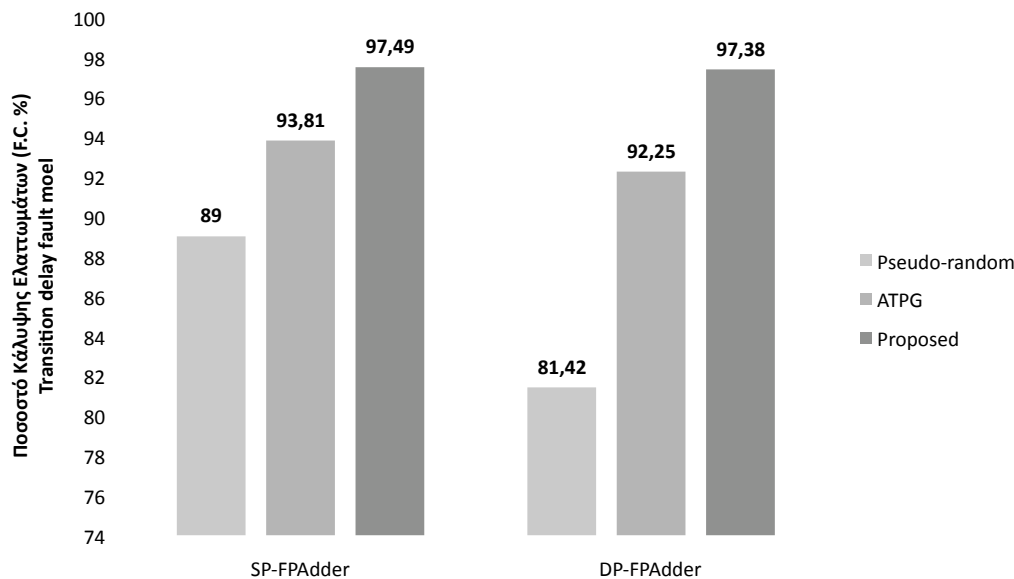
Οι Εικόνες 4.4 και 4.5 παρουσιάζουν τα ποσοστά κάλυψης ελαττωμάτων για τους αθροιστές απλής και διπλής ακρίβειας για τα μοντέλα ελαττωμάτων προσκόλλησης και καθυστέρησης διάδοσης.



Εικόνα 4.4: Σύγκριση ποσοστού κάλυψης ελαττωμάτων για το μοντέλο μόνιμης τιμής.

Ο Πίνακας 4.8 παρουσιάζει το πλήθος των διανυσμάτων δοκιμής που εφαρμόζονται για την επίτευξη των ποσοστών κάλυψης ελαττωμάτων.

Το αυτόματο εργαλείο εξαγωγής διανυσμάτων παράγει διανύσματα με τα μεγαλύτερα ποσοστά κάλυψης για το μοντέλο ελαττωμάτων τόσο για τον αθροιστή απλής ακρίβειας όσο και



Εικόνα 4.5: Σύγκριση ποσοστού κάλυψης ελαττωμάτων για το μοντέλο καθυστέρησης διάδοσης μετάβασης.

FP Unit	Approach	Stuck-At Patterns	Transition Delay Patterns
SP-FPAdder	Pseudo-random	1000000	1000000
	ATPG	413	525
	Proposed	622	622
DP-FPAdder	Pseudo-random	1000000	1000000
	ATPG	754	813
	Proposed	1080	1080

Πίνακας 4.8: Συγκριτικά αποτελέσματα μεθόδων δοκιμής

τον αθροιστή διπλής ακρίβειας. Το αποτέλεσμα αυτό είναι αναμενόμενο καθώς τα εργαλεία εξαγωγής εξετάζουν κάθε πύλη του κυκλώματος. Τα ποσοστά κάλυψης των διανυσμάτων δοκιμής που έχουν αναπτυχθεί βάσει των συνθηκών δοκιμαστικότητας είναι και αυτά αρκετά υψηλά. Ειδικότερα για το μοντέλο ελαττωμάτων καθυστέρησης μετάβασης τα διανύσματα δοκιμής που προκύπτουν βάσει των συνθηκών δοκιμαστικότητας παρέχουν υψηλότερα ποσοστά κάλυψης ελαττωμάτων έναντι των διανυσμάτων που παράγονται από αυτόματα εργαλεία εξαγωγής διανυσμάτων. Το γεγονός αυτό οφείλεται στη σχετικά υψηλότερη πολυπλοκότητα του μοντέλου καθυστέρησης μετάβασης και στην υψηλή πολυπλοκότητα της αρχιτεκτονικής δύο μονοπατιών.

Σε κάθε περίπτωση είναι εμφανές πως τα ψευδοτυχαία διανύσματα αν και παράγονται εύκολα δεν παρέχουν επαρκή ποσοστά κάλυψης ελαττωμάτων.

Σχετικά με το πλήθος των διανυσμάτων δοκιμής τα αυτόματα εργαλεία παράγουν το μικρότερο δυνατό αριθμό διανυσμάτων. Όμως τα διανύσματα που παράγονται βάσει των συνθηκών δοκιμασιμότητας έχουν άλλα πλεονεκτήματα όπως την ανεξαρτησία από την εσωτερική αρχιτεκτονική υλοποίησης των υπομονάδων και την ανεξαρτησία από την ακρίβεια της μονάδας κινητής υποδιαστολής. Επιπλέον τα εργαλεία εξαγωγής διανυσμάτων λειτουργούν σε επίπεδο πύλης και δεν μπορούν να χρησιμοποιηθούν σε λειτουργικά μοντέλα ελαττωμάτων, όπως το μοντέλο ελαττώματος κυψελίδας (Cell fault model) [54].

Συνοψίζοντας, θα πρέπει να αναφέρουμε πως η εξαγωγή διανυσμάτων με βάση τις συνθήκες δοκιμασιμότητας δεν έχει στόχο την αντικατάσταση των αυτόματων εργαλείων εξαγωγής διανυσμάτων δοκιμής. Αντίθετα, επιτρέπει την συνδυασμένη εφαρμογή διανυσμάτων που προκύπτουν και από τις δύο τεχνικές παραγωγής διανυσμάτων ώστε να επιτυγχάνονται τα μεγαλύτερα δυνατά ποσοστά κάλυψης ελαττωμάτων.

4.6 Ανακεφαλαίωση

Στο Κεφάλαιο 4 παρουσιάστηκαν,

- η ανασκόπηση της βιβλιογραφίας σχετικά με τις τεχνικές δοκιμής των μονάδων κινητής υποδιαστολής και οι αδυναμίες που παρουσιάζουν οι υπάρχουσες τεχνικές δοκιμής,
- οι βασικές αρχές για προσαρμόσιμη εξαγωγή διανυσμάτων δοκιμής, και
- οι συνθήκες δοκιμασιμότητας για τους αθροιστές κινητής υποδιαστολής υψηλής απόδοσης.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Κεφάλαιο 5

Αυτοδοκιμή Μικροεπεξεργαστών με Ενσωματωμένο Λογισμικό κατά τη Λειτουργία

Η αυτοδοκιμή των μικροεπεξεργαστών με ενσωματωμένο λογισμικό γνωρίζει μεγάλη αποδοχή τα τελευταία χρόνια όπως φαίνεται από το πλήθος των εργασιών που παρουσιάζονται στη διεθνή βιβλιογραφία, ενώ ήδη εφαρμόζεται και στη βιομηχανία, τουλάχιστον στις εταιρείες Intel και Sun όπως φαίνεται και από τις εργασίες [81] και [115] αντίστοιχα. Το μεγάλο ενδιαφέρον για την αυτοδοκιμή των μικροεπεξεργαστών είναι ευνόητο αν λάβει κανείς υπόψη του τα σημαντικά πλεονεκτήματα που παρουσιάζει έναντι των παραδοσιακών τεχνικών αυτοδοκιμής που βασίζονται στην προσθήκη επιπλέον υλικού (hardware BIST).

Η αυτοδοκιμή με ενσωματωμένο λογισμικό εξασφαλίζει την εφαρμογή των διανυσμάτων δοκιμής στην πραγματική συχνότητα λειτουργίας του μικροεπεξεργαστή (at-speed testing), όπως άλλωστε και η αυτοδοκιμή με την προσθήκη επιπλέον υλικού. Η αυτοδοκιμή με ενσωματωμένο λογισμικό είναι *διαφανής (transparent)* προς τον μικροεπεξεργαστή, δηλαδή δεν επηρεάζει την αρχιτεκτονική του και δεν απαιτεί μετατροπές ώστε το κύκλωμα να είναι έτοιμο για αυτοδοκιμή (κάτι που είναι απαραίτητο στην αυτοδοκιμή με υλικό όπου το κύκλωμα πρέπει να είναι BIST-ready). Η εργασία [116] δείχνει πως τέτοιες μετατροπές είναι πολύπλοκες και επηρεάζουν την απόδοση των κυκλωμάτων και των μικροεπεξεργαστών. Εύλογο είναι πως η αυτοδοκιμή των μικροεπεξεργαστών με λογισμικό δεν επηρεάζει την ταχύτητά τους και δεν οδηγεί σε αύξηση του μεγέθους τους. Επομένως, δεν αυξάνεται η κατανάλωση ενέργειας, σε αντίθεση με την αυτοδοκιμή βασισμένη στο υλικό όπου ο μικροεπεξεργαστής βρίσκεται σε ειδική λειτουργία δοκιμής (test mode) και καταναλώνει περισσότερη ενέργεια απ' ό,τι καταναλώνει όταν βρίσκεται στην κανονική κατάσταση λειτουργίας.

Επίσης, όταν η αυτοδοκιμή πραγματοποιείται στην κανονική κατάσταση λειτουργίας και όχι σε ειδική κατάσταση δοκιμής αποφεύγεται το δυσάρεστο φαινόμενο της *υπερδοκιμής* (*overtesting*). Υπενθυμίζουμε πως το φαινόμενο της υπερδοκιμής εμφανίζεται όταν πραγματοποιείται δοκιμή για ελαττώματα τα οποία δεν επηρεάζουν την κανονική λειτουργία του μικροεπεξεργαστή. Η απόρριψη των μικροεπεξεργαστών εξαιτίας αυτών των ελαττωμάτων οδηγεί στη μείωση της εσοδείας (*yield*) (βλ. Κεφάλαιο 2) με αρνητικές οικονομικές συνέπειες.

Η αυτοδοκιμή των μικροεπεξεργαστών με ενσωματωμένο λογισμικό χαρακτηρίζεται ως τεχνική αυτοδοκιμής *χαμηλού κόστους* (*low-cost*) αφού καθιστά δυνατή τη χρήση εξωτερικού εξοπλισμού δοκιμής χαμηλού κόστους για την αυτοδοκιμή. Ο εξοπλισμός δοκιμής αναλαμβάνει μόνο τη φόρτωση του λογισμικού δοκιμής στην εσωτερική μνήμη του μικροεπεξεργαστή (συνήθως στην *cache*) και την ανάγνωση των αποκρίσεων δοκιμής από τη μνήμη για περαιτέρω αξιολόγηση, μετά το τέλος της αυτοδοκιμής.

Τέλος, η αυτοδοκιμή των επεξεργαστών με λογισμικό εφαρμόζεται με μεγαλύτερη ευκολία κατά τη λειτουργία του μικροεπεξεργαστή (*in-field, on-line testing*) από ό,τι η αυτοδοκιμή βασισμένη στο υλικό αφού δεν είναι απαραίτητο να τεθεί ο μικροεπεξεργαστής σε ειδική κατάσταση δοκιμής, όπου απομονώνεται από το υπόλοιπο σύστημα.

5.1 Δοκιμή ολοκληρωμένων κυκλωμάτων κατά τη λειτουργία

Η ορθή λειτουργία των ολοκληρωμένων κυκλωμάτων γενικότερα αλλά και των μικροεπεξεργαστών ειδικότερα δεν είναι εξασφαλισμένη στο πεδίο λειτουργίας, ακόμα και όταν ο μικροεπεξεργαστής έχει περάσει με επιτυχία το στάδιο της δοκιμής μετά την κατασκευή (*manufacturing testing*). Ατέλειες που εμφανίζονται στο πεδίο λειτουργίας και έχουν τις ρίζες τους σε περιβαλλοντικούς παράγοντες, όπως ηλεκτρομαγνητικές παρεμβολές, κοσμική ακτινοβολία, υψηλές θερμοκρασίες, αλλά και άλλοι παράγοντες, όπως η αναπόφευκτη γήρανση των υλικών, μπορεί να υπονομεύσουν την ορθή λειτουργία του μικροεπεξεργαστή. Τα ελαττώματα που εμφανίζονται εξαιτίας των ατελειών στο πεδίο λειτουργίας είναι γνωστά ως *λειτουργικά ελαττώματα* (*operational faults*) και χωρίζονται σε τρεις κατηγορίες. Τα *μόνιμα ελαττώματα* (*permanent faults*) διαρκούν επ' άπειρον από τη στιγμή που εμφανίζονται και δεν υπάρχει τρόπος να αφαιρεθούν παρά μόνο με αντικατάσταση του κυκλώματος που τα περιέχει [32], [33]. Τα *περιοδικά ελαττώματα* (*intermittent faults*) εμφανίζονται, εξαφανίζονται και επανεμφανίζονται επαναλαμβανόμενα. Η διάρκεια τους μπορεί να είναι αρκετά μεγάλη ώστε να θεωρούνται μόνιμα ελαττώματα μικρής διάρκειας. Επιπλέον, τείνουν να εμφανίζονται στα ίδια σημεία. Συνήθως ένα περιοδικό ελάττωμα είναι προπομπός ενός μόνιμου ελαττώματος [32], [33]. Αντίθετα με τα περιοδικά ελαττώματα, τα *παροδικά ελαττώματα* (*transient faults*) εμφανίζονται με τυχαίο τρόπο και σε τυχαία σημεία σε ένα κύκλωμα. Η ανίχνευση τους

είναι δυσκολότερη σε σχέση με τους άλλους τύπους ελαττωμάτων. Τα παροδικά ελαττώματα επηρεάζουν συνήθως τα στοιχεία μνήμης σε ένα κύκλωμα, δηλαδή τις μνήμες και τα flip-flop και κατά συνέπεια τους καταχωρητές. Τα παροδικά ελαττώματα δημιουργούν σφάλματα τα οποία καλούνται *ήπια* (soft errors)¹.

Η δοκιμή των ολοκληρωμένων κυκλωμάτων και των μικροεπεξεργαστών κατά τη λειτουργία (on-line) ή δοκιμή στο πεδίο (in field testing) εξασφαλίζει την αξιόπιστη λειτουργία των μικροεπεξεργαστών, καθώς ανιχνεύοντας τα ελαττώματα αποτρέπεται η λανθασμένη λήψη αποφάσεων εξαιτίας της εσφαλμένης λειτουργίας. Η δοκιμή κατά τη λειτουργία μπορεί να γίνεται *ταυτόχρονα με την λειτουργία* (concurrent on-line testing) του ολοκληρωμένου κυκλώματος ή του μικροεπεξεργαστή. Εναλλακτικά, μπορεί να πραγματοποιείται απομονώνοντας το ολοκληρωμένο κύκλωμα ή το μικροεπεξεργαστή από το υπόλοιπο σύστημα ώστε η αυτοδοκιμή να πραγματοποιείται *ετερόχρονα από την κανονική λειτουργία* (non-concurrent on-line testing).

Η δοκιμή ταυτόχρονα με τη λειτουργία του μικροεπεξεργαστή επιτυγχάνεται είτε με εισαγωγή επιπλέον υλικού είτε με χρήση επιπλέον λογισμικού. Η εισαγωγή επιπλέον υλικού εκτός από την ανίχνευση ελαττωμάτων μπορεί να παρέχει και δυνατότητα ανοχής στα ελαττώματα (fault tolerance). Η πιο διαδεδομένη τεχνική με βάση το υλικό που παρέχει δυνατότητα ανοχής στα ελαττώματα (hardware redundancy) είναι ο *τριπλασιασμός της μονάδας* (triple modular redundancy—TMR) που πρέπει να προστατευτεί και η προσθήκη ενός κυκλώματος πλειοψηφίας (majority vote). Το κύκλωμα πλειοψηφίας επιλέγει την τιμή που πλειοψηφεί η οποία είναι σωστή αν μόνο μία από τις μονάδες περιέχει ελαττώματα. Στην περίπτωση που δύο από τις τρεις μονάδες περιέχουν ελαττώματα τότε η τιμή που θα προκύψει από το κύκλωμα πλειοψηφίας θα είναι λανθασμένη. Το κόστος της εφαρμογής του τριπλασιασμού της μονάδας σε υλικό είναι 200%. Στην γενική του μορφή, ο πλεονασμός υλικού βασίζεται στη χρήση πολλαπλών μονάδων, αντιγράφων των μονάδων που πρέπει να προστατευθεί από λειτουργικά ελαττώματα. Στην περίπτωση όπου χρησιμοποιούνται N αντίγραφα, το κόστος σε επιπλέον υλικό είναι της τάξης του $N \times 100\%$. Μια άλλη παραλλαγή της τεχνικής του πλεονασμού που μειώνει το κόστος είναι η μέθοδος του *διπλασιασμού-και-σύγκρισης* (duplication and compare) κατά τον οποίο λειτουργούν παράλληλα η μονάδα που πρέπει να προστατευθεί από λειτουργικά ελαττώματα και ένα αντίγραφο της. Τα αποτελέσματα των δύο μονάδων συγκρίνονται διαρκώς και η οποιαδήποτε διαφορά σημαίνει την ανίχνευση ελαττωμάτων (γίνεται μόνο ανίχνευση και όχι διόρθωση). Το κόστος σε υλικό της μεθόδου είναι 100% και χαρακτηρίζεται απαγορευτικό για εφαρμογές χαμηλού κόστους.

Η ταυτόχρονη δοκιμή με την λειτουργία του μικροεπεξεργαστή μπορεί να πραγματοποιείται εφαρμόζοντας μεθοδολογίες *πλεονασμού χρόνου* (time redundancy). Οι μεθοδολογίες πλεονασμού χρόνου, γνωστές και ως *διπλή εκτέλεση* (double execution) ή *επανάληψη* (retry) δεν απαιτούν επιπλέον υλικό αλλά, συνήθως, αυξάνουν τον χρόνο εκτέλεσης του λογισμικού ανάλογα

¹Ονομάζονται ήπια σφάλματα διότι δεν είναι μόνιμα.

με την επιθυμητή προστασία του μικροεπεξεργαστή από λειτουργικά ελαττώματα. Οι εντολές του λογισμικού επανεκτελούνται σε διάφορα χρονικά σημεία και τα αποτελέσματά τους ελέγχονται ώστε να διαπιστωθεί εάν ο μικροεπεξεργαστής έχει παράγει εσφαλμένα αποτελέσματα εξαιτίας λειτουργικών ελαττωμάτων. Η διπλή εκτέλεση των εντολών μπορεί να ανιχνεύσει λειτουργικά ελαττώματα που επηρεάζουν υπολογιστικές υπομονάδες του μικροεπεξεργαστή αλλά και ελαττώματα που επηρεάζουν τον έλεγχο ροής (*control flow*) του λογισμικού. Οι τεχνικές πλεονασμού χρόνου ανιχνεύουν τα περιοδικά και τα παροδικά ελαττώματα αλλά όχι και τα μόνιμα. Η διπλή εκτέλεση των εντολών επηρεασμένη από ένα μόνιμο ελάττωμα θα παράγει λανθασμένες αποκρίσεις και τις δύο φορές, δεδομένο ότι εκτελούνται οι ίδιες ακριβώς εντολές. Η σύγκριση των λανθασμένων αποκρίσεων δε θα δείξει διαφορές και έτσι το μόνιμο ελάττωμα δεν πρόκειται να ανιχνευθεί².

Οι παραπάνω μεθοδολογίες για την ταυτόχρονη δοκιμή με τη λειτουργία συμπληρώνονται με τις μεθοδολογίες πλεονασμού της πληροφορίας (*information redundancy*), οι οποίες προσθέτουν επιπλέον κωδικοποιημένη πληροφορία η οποία βοηθά στην ανίχνευση λειτουργικών ελαττωμάτων. Χαρακτηριστικό παράδειγμα των μεθόδων πλεονασμού πληροφορίας αποτελεί το ψηφίο ισοτιμίας (*parity bit*). Οι μέθοδοι πλεονασμού πληροφορίας έχουν βρει μεγάλη απήχηση σε ακέραιες αριθμητικές μονάδες, για παράδειγμα σε αθροιστές, σε αφαιρετές, σε αυξητές, σε μειωτές, και σε πολλαπλασιαστές. Συνοπτικά αναφέρουμε πως στις αριθμητικές μονάδες έχουν εφαρμοστεί γραμμικοί αριθμητικοί κώδικες (*AN codes*), κώδικες υπολοίπων (*residue codes*), κώδικες *Berger*, *Bose-Lin* και *Borden* και ψηφία ισοτιμίας [117], [118].

Αντίθετα από την ταυτόχρονη δοκιμή των μικροεπεξεργαστών η ετερόχρονη αυτοδοκιμή κατά τη λειτουργία πραγματοποιείται ή σποραδικά (*sporadic non-concurrent on-line testing*) ή περιοδικά (*periodic non-concurrent on-line testing*). Όταν η δοκιμή πραγματοποιείται σποραδικά, δηλαδή κατά την έναρξη λειτουργίας του συστήματος (*start-up*) ή κατά τη λήξη λειτουργίας (*shutdown*) του μπορεί να ανιχνευθούν μόνιμα ελαττώματα ενώ όταν πραγματοποιείται περιοδικά υπάρχει η δυνατότητα ανίχνευσης και περιοδικών ελαττωμάτων (*intermittent faults*). Η περιοδική εκτέλεση της δοκιμής έχει ιδιαίτερη αξία όταν εφαρμόζεται σε συστήματα τα οποία λειτουργούν αδιάλειπτα για μεγάλα χρονικά διαστήματα, όπου δεν είναι αποδεκτή η διακοπή και επανεκκίνησή τους για λόγους αυτοδοκιμής. Επιπλέον, η ετερόχρονη αυτοδοκιμή κατά τη λειτουργία επιτρέπει την ανίχνευση πολλαπλών ελαττωμάτων τα οποία όταν συσσωρεύονται, είναι πολύ πιθανόν να μην ανιχνεύονται με μεθόδους και τεχνικές της ταυτόχρονης αυτοδοκιμής [119]. Για παράδειγμα, η ταυτόχρονη αυτοδοκιμή δεν μπορεί να ανιχνεύσει την ύπαρξη δύο (ή γενικά άρτιου πλήθους) ελαττωμάτων όταν χρησιμοποιείται ένα μόνο ψηφίο ισοτιμίας (*parity bit*).

Γενικά, είναι αποδεκτό πως όταν εφαρμόζεται αυτοδοκιμή κατά τη λειτουργία θα πρέπει να εφαρμόζεται τόσο η ταυτόχρονη αυτοδοκιμή (*concurrent online testing*) όσο και η ετερόχρονη

²Αν εκτελεστούν διαφορετικές εντολές, από διαφορετικές μονάδες, τότε η ανίχνευση των μόνιμων ελαττωμάτων είναι δυνατή.

(non-concurrent online testing). Ο συνδυασμός και των δύο μεθόδων δοκιμής εξασφαλίζει την αξιόπιστη λειτουργία όχι μόνο των μικροεπεξεργαστών αλλά και των άλλων ολοκληρωμένων κυκλωμάτων του συστήματος [33], [119].

5.2 Αυτοδοκιμή μικροεπεξεργαστών κατά τη λειτουργία με ενσωματωμένο λογισμικό

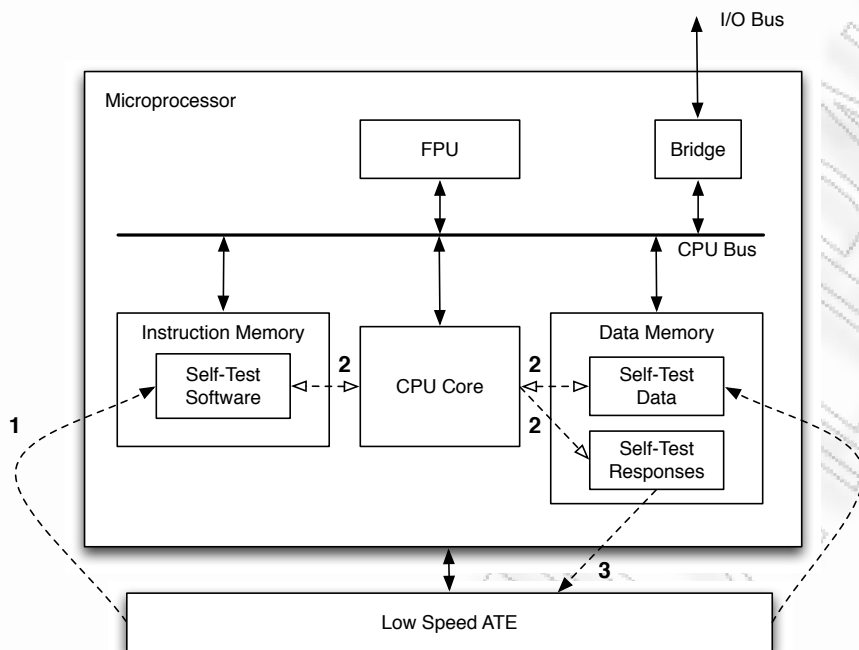
Παραδοσιακά, οι μέθοδοι αυτοδοκιμής των μικροεπεξεργαστών κατά τη λειτουργία είναι τεχνικές αυτοδοκιμής με βάση υλικό (hardware BIST) οι οποίες εφαρμόζονται αμέσως μετά την κατασκευή (IC manufacturing testing). Δυστυχώς, όμως τα μειονεκτήματα που χαρακτηρίζουν τις μεθοδολογίες αυτοδοκιμής με βάση το υλικό, κληρονομούνται στην αυτοδοκιμή των μικροεπεξεργαστών κατά τη λειτουργία. Η αυτοδοκιμή των μικροεπεξεργαστών με ενσωματωμένο λογισμικό παρουσιάζει σημαντικά πλεονεκτήματα έναντι της αυτοδοκιμής με βάση το υλικό όπως αναλύθηκε στο Κεφάλαιο 2 αλλά και στην εισαγωγή του τρέχοντος κεφαλαίου.

Στη συνέχεια παρουσιάζονται τα χαρακτηριστικά, όπως αυτά έχουν αναπτυχθεί στην εργασία [27], τα οποία θα πρέπει να διαθέτει το ενσωματωμένο λογισμικό, ώστε να είναι επιτυχημένη η ετερόχρονη αυτοδοκιμή κατά τη λειτουργία.

Η αυτοδοκιμή των μικροεπεξεργαστών με ενσωματωμένο λογισμικό κατά τη λειτουργία διαφέρει σημαντικά από την αυτοδοκιμή αμέσως μετά την κατασκευή του επεξεργαστή στο εργοστάσιο. Οι διαφορές αυτές οφείλονται στις διαφορετικές του περιβάλλοντος στο οποίο βρίσκεται ο μικροεπεξεργαστής κατά την αυτοδοκιμή στο εργοστάσιο κατασκευής από το περιβάλλον λειτουργίας του συστήματος, όπως φαίνεται από τις Εικόνες 5.1 και 5.2.

Το περιβάλλον κατά την αυτοδοκιμή του μικροεπεξεργαστή αμέσως μετά την κατασκευή του είναι λιτό όπως εύκολα διαπιστώνει κανείς από την Εικόνα 5.1. Ο εξοπλισμός δοκιμής (στην περίπτωση που μελετούμε πρόκειται για εξοπλισμό δοκιμής χαμηλής ταχύτητας low-speed ATE) αναλαμβάνει τη φόρτωση του ενσωματωμένου λογισμικού αυτοδοκιμής από την εσωτερική του μνήμη στην μνήμη του μικροεπεξεργαστή (βήμα 1 στην Εικόνα 5.1). Έπειτα ενεργοποιεί τον μικροεπεξεργαστή ο οποίος εκτελεί το λογισμικό αυτοδοκιμής και αποθηκεύει τις αποκρίσεις των συνόλων των διανυσμάτων δοκιμής στην μνήμη δεδομένων (βήμα 2 στην Εικόνα 5.1). Τέλος, ο εξοπλισμός δοκιμής αναλαμβάνει να διαβάσει τις αποκρίσεις από την μνήμη και να τις συγκρίνει με τις αναμενόμενες αποκρίσεις ώστε να ανιχνευθούν τυχόν ελαττώματα στον μικροεπεξεργαστή (βήμα 3 στην Εικόνα 5.1).

Αντίθετα, το περιβάλλον της ετερόχρονης αυτοδοκιμής κατά τη λειτουργία είναι πολύ πιο πολύπλοκο και εισάγει νέες απαιτήσεις και χαρακτηριστικά από το ενσωματωμένο λογισμικό. Όπως φαίνεται από την Εικόνα 5.2, ο μικροεπεξεργαστής επικοινωνεί με ποικίλες εξωτερικές μονάδες και βρίσκεται υπό τον έλεγχο ενσωματωμένου λογισμικού το οποίο μπορεί να έχει τη μορφή ενός λειτουργικού συστήματος, κάποιες φορές πραγματικού χρόνου. Η ετερόχρονη αυ-

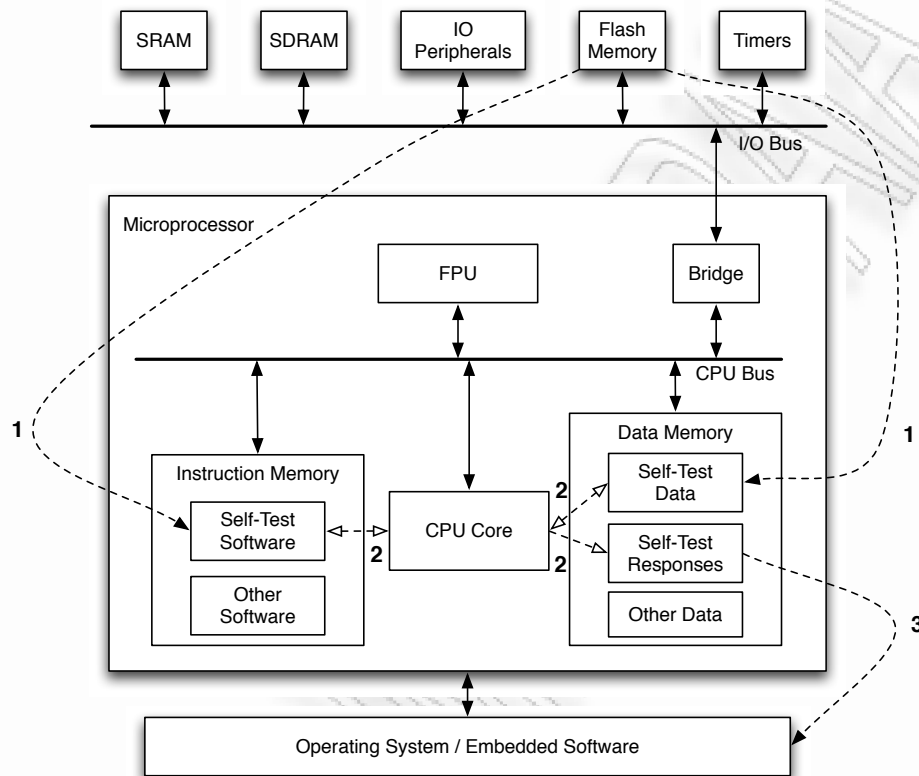


Εικόνα 5.1: Το περιβάλλον του μικροεπεξεργαστή κατά την αυτοδοκιμή στο εργοστάσιο κατασκευής.

τοδοκιμή μπορεί να πραγματοποιείται κατά την εκκίνηση ή κατά τον τερματισμό λειτουργίας του μικροεπεξεργαστή, εξασφαλίζοντας με αυτό τον τρόπο την μελλοντική ορθή λειτουργία. Εναλλακτικά, το λειτουργικό σύστημα εφαρμόζει την αυτοδοκιμή σε αδρανή χρονικά διαστήματα (idle cycles), όπως αυτά αναγνωρίζονται από το χρονοδρομολογητή (scheduler) του λειτουργικού συστήματος. Επιπλέον, εάν στο σύστημα υπάρχουν διαθέσιμοι χρονομετρητές (timers), τότε αυτοί είναι δυνατό να προγραμματιστούν ώστε η αυτοδοκιμή να εφαρμόζεται ανά τακτά χρονικά διαστήματα. Καθώς οι χρονομετρητές είναι προγραμματιζόμενοι, η περίοδος επανάληψης της αυτοδοκιμής μπορεί να μεταβάλλεται δυναμικά δίχως την επανεκκίνηση του συστήματος.

Όταν η αυτοδοκιμή πραγματοποιείται κατά την εκκίνηση ή κατά τον τερματισμό, μόνο μόνιμα ελαττώματα που προϋπήρχαν ή εμφανίστηκαν στο πεδίο λειτουργίας είναι δυνατό να ανιχνευθούν. Από την άλλη πλευρά όταν η αυτοδοκιμή εφαρμόζεται περιοδικά τότε ανιχνεύονται και περιοδικά ελαττώματα (δεδομένο ότι διαρκούν αρκετά ώστε να θεωρηθούν μόνιμα). Πάντως, συμπτωματικά και μόνο είναι δυνατό να ανιχνευθούν παροδικά ελαττώματα και στις δύο περιπτώσεις.

Το ενσωματωμένο λογισμικό αυτοδοκιμής βρίσκεται συνήθως σε μια μη-πτητική μνήμη (non-volatile), για παράδειγμα τύπου Flash ή ROM. Σε συστήματα χαμηλού κόστους οι μνήμες αυτές δεν είναι ιδιαίτερα γρήγορες ενώ και το εύρος του διαύλου είναι μικρότερο από το εύρος του



Εικόνα 5.2: Το περιβάλλον του μικροεπεξεργαστή στο πεδίο λειτουργίας του συστήματος.

μικροεπεξεργαστή. Ο μικροεπεξεργαστής, είτε με τη βοήθεια του λειτουργικού συστήματος είτε με τη βοήθεια των χρονομετρητών αναλαμβάνει να εκτελέσει το λογισμικό αυτοδοκιμής το οποίο μεταφέρεται σταδιακά στην εσωτερική μνήμη του μικροεπεξεργαστή (βήμα 1 στην Εικόνα 5.2). Έπειτα από την εκτέλεση του λογισμικού αυτοδοκιμής (βήμα 2 στην Εικόνα 5.2) παράγονται οι αποκρίσεις της δοκιμής. Τελικά, αυτές συγκρίνονται με τις αναμενόμενες αποκρίσεις από το ενσωματωμένο λογισμικό ή από το λειτουργικό σύστημα (βήμα 3 στην Εικόνα 5.2) ώστε να ανιχνευθούν τυχόν ελαττώματα. Παράλληλα, στις κρυφές μνήμες εντολών και δεδομένων βρίσκονται όχι μόνο το λογισμικό αυτοδοκιμής και οι αποκρίσεις δοκιμής αλλά και το ενσωματωμένο λογισμικό ή το λειτουργικό σύστημα καθώς και το λογισμικό των εφαρμογών του χρήστη.

Ένας άλλος σημαντικός παράγοντας στον οποίο διαφέρει η αυτοδοκιμή στο εργοστάσιο κατασκευής από την αυτοδοκιμή στο πεδίο λειτουργίας, είναι η σύμπτυξη (compaction) των αποκρίσεων. Στη δοκιμή αμέσως μετά την κατασκευή του μικροεπεξεργαστή, το σύστημα δοκιμής αναλαμβάνει να συγκρίνει τις αποκρίσεις του μικροεπεξεργαστή, δηλαδή τις αποκρίσεις

που έχουν αποθηκευτεί στην εσωτερική μνήμη που διαθέτει ο μικροεπεξεργαστής, με αυτές που βρίσκονται προ-αποθηκευμένες στη μνήμη του συστήματος δοκιμής. Η σύμπτυξη των αποκρίσεων των διανυσμάτων δοκιμής μπορεί να χαρακτηριστεί προαιρετική. Αντίθετα, στη δοκιμή κατά τη λειτουργία του μικροεπεξεργαστή η σύμπτυξη των αποκρίσεων είναι μονόδρομος, διότι δεν είναι δυνατό να υπάρχουν κάπου προ-αποθηκευμένες οι σωστές αποκρίσεις. Κάτι τέτοιο θα απαιτούσε αρκετά μεγάλες μνήμες οι οποίες θα επιβάρυναν το σύστημα, τόσο ως προς το μέγεθός του όσο και ως προς την συνολική κατανάλωση ενέργειας. Μόνο ένας μικρός αριθμός υπογραφών (signatures) μπορεί να βρίσκεται αποθηκευμένος σε κάποια μη-πτητική μνήμη του συστήματος. Η σύμπτυξη των αποκρίσεων των διανυσμάτων δοκιμής μπορεί να πραγματοποιηθεί σύμφωνα με κάποια από τις τεχνικές που παρουσιάζονται στο Κεφάλαιο 2.

Είναι ευνόητο ότι το ενσωματωμένο λογισμικό αυτοδοκιμής ανταγωνίζεται για πόρους του συστήματος μαζί με τις υπόλοιπες εφαρμογές, γεγονός που δε συμβαίνει κατά την αυτοδοκιμή μετά το στάδιο της κατασκευής. Συνεπώς, το ενσωματωμένο λογισμικό αυτοδοκιμής όταν δεν εφαρμόζεται με αποδοτικό τρόπο προκαλεί επιβάρυνση της λειτουργίας του συνολικού συστήματος.

5.2.1 Επιθυμητά χαρακτηριστικά λογισμικού αυτοδοκιμής

Το λογισμικό αυτοδοκιμής θα πρέπει να διαθέτει τα παρακάτω χαρακτηριστικά:

- μικρό ίχνος μνήμης,
- χαμηλή κατανάλωση ενέργειας, και
- μικρό χρόνο εκτέλεσης,

ώστε η εφαρμογή του να μην προκαλεί επιβάρυνση στην λειτουργία και την απόδοση του συστήματος αλλά να αποτελεί μια αποτελεσματική λύση χαμηλού κόστους για ετερόχρονη αυτοδοκιμή στο πεδίο λειτουργίας.

Μικρό ίχνος μνήμης

Το ενσωματωμένο λογισμικό αυτοδοκιμής θα πρέπει να έχει μικρό ίχνος μνήμης (*small memory footprint*) καθώς θα πρέπει να συνυπάρχει μαζί με τις εφαρμογές του χρήστη ή/και με το λειτουργικό σύστημα. Γενικότερα, όταν το ενσωματωμένο λογισμικό έχει μεγάλο ίχνος μνήμης είναι πιθανό να δημιουργηθούν *αστοχίες (misses)* στην κρυφή μνήμη οι οποίες επηρεάζουν το χρόνο εκτέλεσης του ενσωματωμένου λογισμικού. Επιπλέον, εξαιτίας των αστοχιών στην κρυφή μνήμη είναι πολύ πιθανό τα δεδομένα των υπολοίπων εφαρμογών να χρειαστεί να μετακινηθούν εκτός κρυφής μνήμης ώστε να δημιουργηθεί ο απαιτούμενος χώρος που χρειάζεται το λογισμικό αυτοδοκιμής. Οι αρνητικές επιπτώσεις συνεχίζονται όταν το λογισμικό του

χρήστη συνεχίζει την εκτέλεσή του, αλλά τώρα, δε βρίσκει στην κρυφή μνήμη τα τμήματα των εντολών και των δεδομένων που χρειάζεται, διότι έχουν αντικατασταθεί από εντολές και δεδομένα του λογισμικού αυτοδοκιμής. Κατ' αυτό τον τρόπο δημιουργούνται νέες αστοχίες και η κρυφή μνήμη προβαίνει στην έναρξη νέων *συναλλαγών (transactions)* με την κύρια μνήμη. Εν τέλει, αυξάνεται ο χρόνος εκτέλεσης των εφαρμογών του συστήματος.

Το ίχνος του ενσωματωμένου λογισμικού έχει επιπτώσεις στην επίδοση του συστήματος ακόμα και όταν δε χρησιμοποιείται κρυφή μνήμη. Στην περίπτωση αυτή, το ενσωματωμένο λογισμικό μεταφέρεται από μια μη πτητική μνήμη (Flash, ROM) σε μια γρήγορη πτητική μνήμη, για παράδειγμα SRAM, ώστε να εκτελεστεί από εκεί. Είναι εύλογο πως όσο μεγαλύτερο είναι το ίχνος του λογισμικού αυτοδοκιμής τόσο λιγότερος χώρος μνήμης θα είναι διαθέσιμος για τις υπόλοιπες εφαρμογές του συστήματος. Επιπλέον, η μεταφορά του λογισμικού αυτοδοκιμής από την αργή μη-πτητική μνήμη σε μια γρήγορη πτητική λαμβάνει χώρα κατά την εκκίνηση του συστήματος. Δεδομένου ότι οι μη-πτητικές μνήμες είναι αργές και δεν έχουν μεγάλο εύρος διαύλου δεδομένων όσο μεγαλύτερο είναι το ίχνος του λογισμικού αυτοδοκιμής τόσο μεγαλύτερος είναι και ο χρόνος εκκίνησης του συστήματος.

Αντίθετα, όταν η αυτοδοκιμή πραγματοποιείται έπειτα από την κατασκευή του μικροεπεξεργαστή, τα θέματα μνήμης που τίθενται είναι λιγότερο αυστηρά, καθώς η εσωτερική μνήμη στο ολοκληρωμένο κύκλωμα του μικροεπεξεργαστή είναι εξ' ολοκλήρου διαθέσιμη για το λογισμικό αυτοδοκιμής³.

Χαμηλή κατανάλωση ενέργειας

Η κατανάλωση ενέργειας (power consumption) αποτελεί μείζον θέμα της αυτοδοκιμής των μικροεπεξεργαστών στο πεδίο λειτουργίας, αλλά και γενικότερα όλων των εφαρμογών ενσωματωμένου λογισμικού, δεδομένου ότι οι περισσότερες εφαρμογές χρησιμοποιούν μπαταρίες για την τροφοδοσία. Η υψηλή κατανάλωση ενέργειας συνεπάγεται και υψηλότερη έκλυση θερμότητας από τον μικροεπεξεργαστή, ένα ανεπιθύμητο γεγονός διότι η εκλυόμενη θερμότητα θα πρέπει να απαχθεί. Η μελέτη, που παρουσιάζεται στην εργασία [112], σχετικά με την κατανάλωση ενέργειας σε φορητούς επεξεργαστές δείχνει πως το 33% της συνολικής ενέργειας ενός φορητού υπολογιστή δαπανάται στο μικροεπεξεργαστή ο οποίος διαθέτει κρυφή μνήμη δύο επιπέδων (2-level cache system). Από αυτό το ποσοστό, περίπου το 20% με 30% καταναλώνεται από το υποσύστημα της κρυφής μνήμης, ενώ περίπου 30% καταναλώνεται από το κύκλωμα χρονισμού (clock circuitry).

Είναι εμφανές πως η κατανάλωση ενέργειας σε ένα μικροεπεξεργαστή σχετίζεται άμεσα με το υποσύστημα της κρυφής και κύριας μνήμης. Όσο περισσότερες είναι οι αστοχίες στην κρυφή μνήμη, τόσο μεγαλύτερα τα ποσά ενέργειας που δαπανώνται, καθώς επιπλέον ενέργεια

³ Αν η εσωτερική μνήμη του μικροεπεξεργαστή είναι πολύ μικρή, τότε υπάρχει περίπτωση να μη χωρά το λογισμικό αυτοδοκιμής σε αυτή, ακόμα και όταν η δοκιμή πραγματοποιείται στο εργοστάσιο κατασκευής.

απαιτείται για την οδήγηση των κυκλωμάτων χρονισμού και των υποσυστημάτων διαύλων για την επικοινωνία κρυφής και κύριας μνήμης. Το λογισμικό αυτοδοκιμής θα πρέπει να είναι κατάλληλα σχεδιασμένο ώστε να μειώνει κατά το μέγιστο δυνατό τις αστοχίες στην κρυφή μνήμη και, γενικότερα, τις συναλλαγές με την κύρια μνήμη, μειώνοντας με αυτό τον τρόπο την κατανάλωση ενέργειας.

Μικρός χρόνος εκτέλεσης

Όπως έχουμε ήδη αναλύσει η αυτοδοκιμή των μικροεπεξεργαστών κατά τη λειτουργία αποτελεί τροχοπέδη στις επιδόσεις του συστήματος, καθώς το λογισμικό αυτοδοκιμής συναγωνίζεται με το λογισμικό των εφαρμογών για κύκλους εκτέλεσης στον μικροεπεξεργαστή. Έτσι, είναι αναγκαίο το λογισμικό αυτοδοκιμής να εκτελείται γρήγορα. Ιδανικά, το λογισμικό αυτοδοκιμής δε θα πρέπει να έχει χρόνο εκτέλεσης μεγαλύτερο από ένα κβάντο χρόνου (*quantum time*), θεωρώντας ένα λειτουργικό σύστημα με *εξυπηρέτηση εκ' περιτροπής (round robin)*. Τυπικές τιμές των κβάντων χρόνου σε λειτουργικά συστήματα που χρησιμοποιούνται σε ενσωματωμένες εφαρμογές, είναι της τάξης μερικών δεκάδων χιλιοστών του δευτερολέπτου (*millisecond*). Ο χρονοδρομολογητής του λειτουργικού συστήματος μπορεί εύκολα να εντοπίσει αδρανή χρονικά διαστήματα και να εκκινήσει την εκτέλεση του λογισμικού αυτοδοκιμής. Παρόλο, που ο χρόνος εκτέλεσης του λογισμικού αυτοδοκιμής είναι δυνατό να εκτείνεται σε περισσότερα από ένα κβάντα χρόνου, κάτι τέτοιο δεν είναι επιθυμητό καθώς αυξάνεται ο *χρόνος ανίχνευσης των ελαττωμάτων (fault detection latency)*, τόσο των μόνιμων όσο και των περιοδικών ελαττωμάτων. Επιπλέον, οι εναλλαγές περιβάλλοντος (*context switch*) επιβαρύνουν το χρόνο εκτέλεσης των υπολοίπων εφαρμογών αλλά και το σύστημα μνήμης, καθώς κατά την εναλλαγή περιβάλλοντος τα περιεχόμενα των καταχωρητών αποθηκεύονται στη μνήμη.

Αν η ενσωματωμένη εφαρμογή του συστήματος δε συνοδεύεται από λειτουργικό σύστημα, η αυτοδοκιμή μπορεί να πραγματοποιηθεί με τη βοήθεια των χρονομετρητών. Είναι εύλογο ότι και σε αυτή την περίπτωση, όσο ταχύτερα εκτελείται το λογισμικό αυτοδοκιμής τόσο μικρότερη είναι και η επιβάρυνση στο σύνολο του συστήματος. Συνάμα, το λογισμικό αυτοδοκιμής μπορεί να εκτελείται συχνότερα όταν δεν απαιτεί μεγάλο χρόνο εκτέλεσης και έτσι μειώνεται ο χρόνος ανίχνευσης ελαττωμάτων.

Αξιολόγηση των χαρακτηριστικών

Το ενσωματωμένο λογισμικό αυτοδοκιμής θα πρέπει να εφαρμόζεται σε ένα σύστημα με το μικρότερο δυνατό κόστος, ώστε να αποτελεί μια πρακτική λύση για την ετερόχρονη αυτοδοκιμή κατά τη λειτουργία. Τα βασικά χαρακτηριστικά που θα πρέπει να διαθέτει είναι το μικρό ίχνος μνήμης, η χαμηλή κατανάλωση ενέργειας και ο μικρός χρόνος εκτέλεσης. Όμως τα παραπάνω χαρακτηριστικά δεν είναι πάντα εύκολο να επιτευχθούν καθώς αλληλεπιδρούν μεταξύ τους και η βελτιστοποίηση ενός από τους παραπάνω παράγοντες συνήθως επιδρά

αρνητικά στους υπόλοιπους. Για παράδειγμα, η μείωση του χρόνου εκτέλεσης συνήθως αυξάνει το ίχνος μνήμης, ενώ όταν μειώνεται το ίχνος μνήμης παρατηρείται αύξηση του χρόνου εκτέλεσης.

Καθώς οι μικροεπεξεργαστές λειτουργούν σε αρκετά υψηλές συχνότητες, είναι προτιμότερο το ενσωματωμένο λογισμικό να βελτιστοποιείται ως προς τις απαιτήσεις σε μνήμη, και ταυτόχρονα, να ελαχιστοποιούνται οι συναλλαγές με το σύστημα της κύριας μνήμης για μείωση της κατανάλωσης ενέργειας. Μια τέτοια βελτιστοποίηση επιτυγχάνεται σχεδιάζοντας το λογισμικό αυτοδοκιμής ώστε να εκμεταλλεύεται την *χρονική τοπικότητα* (*temporal locality*) και την *χωρική τοπικότητα* (*spatial locality*). Η χρονική και η χωρική τοπικότητα επιτυγχάνονται όταν το λογισμικό αυτοδοκιμής χρησιμοποιεί συμπαγείς (compact) ρουτίνες οι οποίες βασίζονται σε βρόχους επανάληψης (loops). Παράλληλα, όταν το λογισμικό αυτοδοκιμής χαρακτηρίζεται από χρονική και χωρική τοπικότητα, ελαχιστοποιούνται και οι συναλλαγές της κρυφής μνήμης με την κύρια μνήμη και έτσι επιτυγχάνεται μείωση της κατανάλωσης ενέργειας που είναι σημαντικός παράγοντας για ενσωματωμένα φορητά συστήματα που τροφοδοτούνται από μπαταρίες.

5.3 Ανάπτυξη ενσωματωμένου λογισμικού αυτοδοκιμής

Η ανάπτυξη του ενσωματωμένου λογισμικού αυτοδοκιμής είναι άμεσα εξαρτώμενη από την προέλευση ή αλλιώς τον τρόπο ανάπτυξης των διανυσμάτων δοκιμής. Το πλήθος των εργασιών που πραγματεύονται την αυτοδοκιμή των μικροεπεξεργαστών με λογισμικό μπορεί να χωριστεί στις παρακάτω κατηγορίες ανάλογα με τον τρόπο ανάπτυξης των διανυσμάτων δοκιμής. Έτσι, χωρίζονται σε εργασίες που:

- χρησιμοποιούν εργαλεία αυτόματης εξαγωγής διανυσμάτων δοκιμής (ATPG CAD tools),
- εξάγουν διανύσματα δοκιμής με γεννίτορες ψευδοτυχαίων διανυσμάτων (pseudorandom pattern generators),
- εξάγουν διανύσματα δοκιμής με ντετερμινιστικό τρόπο βασισμένες στην έμφυτη συμμετρική δομή που παρουσιάζουν σχεδόν όλες οι υπολογιστικές μονάδες (computational units).

Στη συνέχεια του κεφαλαίου εξετάζεται η καταλληλότητα του λογισμικού αυτοδοκιμής, όπως αυτό αναπτύσσεται ανάλογα με τα χαρακτηριστικά της εκάστοτε κατηγορίας. Τα τμήματα του λογισμικού που θα παρουσιαστούν στη συνέχεια δεν εξαρτώνται από κάποιο συγκεκριμένο σύνολο εντολών αλλά ούτε και χρησιμοποιούν εντολές οι οποίες είναι μοναδικές σε μια και μόνο αρχιτεκτονική. Το λογισμικό εύκολα μπορεί να μεταφερθεί σε έναν άλλο μικροεπεξεργαστή του ίδιου τύπου, για παράδειγμα RISC, αλλάζοντας τα ονόματα των εντολών και των καταχωρητών όπου είναι απαραίτητο. Η ανάπτυξη του ενσωματωμένου λογισμικού

δοκιμής παρουσιάζεται για τον ακέραιο πολλαπλασιαστή του μικροεπεξεργαστή Plasma, ο οποίος υλοποιεί το σύνολο εντολών του μικροεπεξεργαστή MIPS-I, εκτός από τις εντολές φόρτωσης και αποθήκευσης σε μη ευθυγραμμισμένες διευθύνσεις (unaligned load-store). Ο μικροεπεξεργαστής Plasma είναι ένας επεξεργαστής τύπου RISC, ο οποίος υλοποιεί τρία στάδια διοχέτευσης (pipeline). Οι ίδιες αρχές ανάπτυξης λογισμικού αυτοδοκιμής για τον πολλαπλασιαστή εφαρμόζονται και στις υπόλοιπες λειτουργικές μονάδες του μικροεπεξεργαστή.

5.3.1 Λογισμικό Αυτοδοκιμής βασισμένο σε εργαλεία εξαγωγής διανυσμάτων δοκιμής

Η ανάπτυξη του λογισμικού αυτοδοκιμής ξεκινά με την εφαρμογή αυτόματων εργαλείων εξαγωγής διανυσμάτων δοκιμής για μια υπομονάδα του μικροεπεξεργαστή λαμβάνοντας υπόψη τους περιορισμούς (constraints), όπως αυτοί τίθενται από το σύνολο των εντολών του μικροεπεξεργαστή. Ένα παράδειγμα περιορισμού περιγράφεται στη συνέχεια.

Ας υποθέσουμε πως όλες οι λογικές και αριθμητικές λειτουργίες, που περιγράφονται από το σύνολο εντολών του μικροεπεξεργαστή και πραγματοποιούνται από μια αριθμητική και λογική μονάδα, κωδικοποιούνται με 3 bit και συνολικά χρησιμοποιούνται έξι συνδυασμοί από τους οχτώ δυνατούς συνδυασμούς. Οι δύο αχρησιμοποίητοι συνδυασμοί δε θα εμφανιστούν ποτέ κατά τη λειτουργία του μικροεπεξεργαστή. Η πληροφορία αυτή χρησιμοποιείται από το αυτόματο εργαλείο εξαγωγής διανυσμάτων ώστε να *μη δημιουργήσει* διανύσματα δοκιμής που εφαρμόζουν τους δύο αχρησιμοποίητους συνδυασμούς.

Η ανάπτυξη λογισμικού αυτοδοκιμής με βάση εργαλεία εξαγωγής διανυσμάτων δοκιμής προϋποθέτει πως είναι διαθέσιμη η περιγραφή της μονάδας υπό δοκιμή σε επίπεδο λογικών πυλών (gate level description). Αυτό είναι απαραίτητο διότι τα εργαλεία εξαγωγής διανυσμάτων δοκιμής χρειάζονται την περιγραφή του κυκλώματος σε επίπεδο πυλών ώστε να εξάγουν τα διανύσματα δοκιμής.

Αφού εξαχθούν τα διανύσματα δοκιμής για τη μονάδα υπό δοκιμή, θα πρέπει να μετατραπούν σε λογισμικό αυτοδοκιμής χρησιμοποιώντας κάποιο από τα πρότυπα ανάπτυξης λογισμικού αυτοδοκιμής που παρουσιάζονται στη συνέχεια.

Για τον πολλαπλασιαστή του μικροεπεξεργαστή Plasma, το αυτόματο εργαλείο εξαγωγής διανυσμάτων παράγει διανύσματα δοκιμής από τα οποία ορισμένα εφαρμόζονται με τη χρήση της εντολής του απρόσημου πολλαπλασιασμού (multu) και τα υπόλοιπα με τη χρήση της εντολής του προσημασμένου πολλαπλασιασμού (mult). Τα διανύσματα δοκιμής μετατρέπονται σε ενσωματωμένο λογισμικό με δύο τρόπους. Στην πρώτη περίπτωση, εντολές άμεσης διευθυνσιοδότησης (immediate address mode)⁴, όπως οι εντολές addi, ori, lui (load upper imme-

⁴Οι εντολές άμεσης διευθυνσιοδότησης εκτελούν μια λειτουργία μεταξύ ενός καταχωρητή προέλευσης (source register) και μιας τιμής των 16-bit και αποθηκεύουν το αποτέλεσμα σε ένα καταχωρητή προορισμού (destination register).

diate), φορτώνουν στους καταχωρητές τμηματικά τα διανύσματα δοκιμής. Πιο συγκεκριμένα, ένα διάνυσμα δοκιμής 32 ψηφίων, φορτώνεται σε έναν καταχωρητή με το συνδυασμό των εντολών lui και ori. Η εντολή lui register, #16-bit value φορτώνει μια τιμή 16 bit στα υψηλά 16 bit του καταχωρητή ενώ η εντολή ori συμπληρώνει τα χαμηλά 16 bit του καταχωρητή. Στη δεύτερη περίπτωση τα διανύσματα δοκιμής βρίσκονται στη μνήμη, από όπου μεταφέρονται στους καταχωρητές χρησιμοποιώντας την εντολή lw (load word)⁵.

Έστω ότι ένα αυτόματο εργαλείο εξαγωγής διανυσμάτων δοκιμής έχει δημιουργήσει N διανύσματα δοκιμής που αντιστοιχούν στην πράξη του προσημασμένου πολλαπλασιασμού (mult). Το λογισμικό αυτοδοκιμής στην Εικόνα 5.3 χρησιμοποιεί τις εντολές lui και ori (γραμμές 1-2 και 3-4) για να φορτώσει στους καταχωρητές \$s0 και \$s1 το πρώτο διάνυσμα δοκιμής. Η εντολή mult (γραμμή 5) εφαρμόζει το πρώτο διάνυσμα δοκιμής στον πολλαπλασιαστή.

```

1  lui $s0, pattern_1_X_high_bits
2  ori $s0, $s0, pattern_1_X_low_bits
3  lui $s1, pattern_1_Y_high_bits
4  ori $s1, $s1, pattern_1_Y_low_bits
5  mult $s0, $s1
6  # compaction code
7
8  lui $s0, pattern_2_X_high_bits
9  ori $s0, $s0, pattern_2_X_low_bits
10 lui $s1, pattern_2_Y_high_bits
11 ori $s1, $s1, pattern_2_Y_high_bits
12 mult $s0, $s1
13 # compaction code
14
15
16
17 lui $s0, pattern_N_X_high_bits
18 ori $s0, $s0, pattern_N_X_low_bits
19 lui $s1, pattern_N_Y_high_bits
20 ori $s1, $s1, pattern_N_Y_low_bits
21 mult $s0, $s1
22 # compaction code
23
24
25

```

Εικόνα 5.3: Τα διανύσματα δοκιμής (ATPG) σχηματίζονται με τη βοήθεια της ψευδοεντολής li (load immediate).

Το γινόμενο του πολλαπλασιαστή, που αποτελείται από 64 ψηφία, αποθηκεύεται σε δύο

⁵Μια λέξη έχει μέγεθος 4 byte.

```

1  la $s0, mult_patterns
2  addi $t0, $zero, 0x0001
3
4  mult_loop:
5  lw $s1, 0($s0)
6  addiu $s0, $s0, 0x0004
7  lw $s2, 0($s0)
8  addiu $s0, $s0, 0x0004
9  mult $s1, $s2
10 # compaction code
11
12 sltiu $t1, $t0, number_of_test_patterns
13 bnez $t1, mul_loop
14 addiu $t0, $t0, 0x0001

```

Εικόνα 5.4: Τα διανύσματα δοκιμής (ATPG) φορτώνονται από τη μνήμη χρησιμοποιώντας την εντολή lw (load word).

ειδικούς καταχωρητές, με όνομα Hi (High) και Lo (Low). Ο καταχωρητής Hi περιέχει τα 32 υψηλά ψηφία του γινομένου, ενώ ο Lo τα 32 χαμηλά ψηφία.

Έπειτα ακολουθεί η *σύμπτυξη (compaction)* (γραμμή 6) των αποκρίσεων των διανυσμάτων δοκιμής. Στο σημείο αυτό έχει ολοκληρωθεί η εφαρμογή του πρώτου ζεύγους διανυσμάτων δοκιμής στον πολλαπλασιαστή. Αντίστοιχα, στις γραμμές 8-12 εφαρμόζεται το δεύτερο ζευγάρι διανυσμάτων δοκιμής και η απόκριση του πολλαπλασιαστή συμπύσσεται με την απόκριση του πρώτου ζεύγους διανυσμάτων. Η διαδικασία επαναλαμβάνεται μέχρι να εφαρμοστεί και το τελευταίο ζεύγος διανυσμάτων δοκιμής.

Μια άλλη παραλλαγή της ανάπτυξης των διανυσμάτων δοκιμής είναι η αποθήκευσή τους στη μνήμη. Το λογισμικό αυτοδοκιμής είναι ένας βρόχος επανάληψης ο οποίος εκτελείται τόσες φορές όσα και τα ζεύγη διανυσμάτων αυτοδοκιμής. Εσωτερικά, τα διανύσματα δοκιμής μεταφέρονται από τη μνήμη στους καταχωρητές με την εντολή lw (load word). Έπειτα εφαρμόζονται στον πολλαπλασιαστή και η απόκριση συμπύσσεται με τις προηγούμενες αποκρίσεις. Το λογισμικό, το οποίο εφαρμόζει τα διανύσματα δοκιμής από τη μνήμη, παρουσιάζεται στην Εικόνα 5.4.

Αρχικά, τα διανύσματα δοκιμής τοποθετούνται στην μνήμη στη θέση `mult_patterns`. Στις γραμμές 1-2, φορτώνεται στον καταχωρητή `$s0` η διεύθυνση στην οποία βρίσκονται τα διανύσματα δοκιμής, ενώ ο καταχωρητής `$t0` μετράει τα ζεύγη διανυσμάτων δοκιμής που έχουν εφαρμοστεί στον πολλαπλασιαστή.

Έχοντας ολοκληρώσει την αρχικοποίηση των μεταβλητών του βρόχου ακολουθεί η φόρτωση του διανύσματος δοκιμής στους καταχωρητές `$s1` και `$s2` αντίστοιχα (γραμμές 5-8). Η εντολή lw χρειάζεται δύο κύκλους ρολογιού για να μεταφέρει τα δεδομένα από τη μνήμη στον

καταχωρητή προορισμού. Για το λόγο αυτό ο καταχωρητής προορισμού δε μπορεί να χρησιμοποιηθεί στην αμέσως επόμενη εντολή. Εάν μια εντολή προσπαθήσει να προσπελάσει τα περιεχόμενα του καταχωρητή προορισμού, τότε θα δημιουργηθεί *καθυστερήση διοχέτευσης (pipeline stall)*. Οι καθυστερήσεις διοχέτευσης είναι ανεπιθύμητες καθώς αυξάνουν το χρόνο εκτέλεσης του λογισμικού. Ένας τρόπος για να αποφύγουμε τις καθυστερήσεις διοχέτευσης είναι να επαναδιατάξουμε τις εντολές τοποθετώντας αμέσως μετά την εντολή lw μια εντολή που αυξάνει το δείκτη της θέσης μνήμης ώστε να δείχνει στο επόμενο διάνυσμα δοκιμής (γραμμές 6 και 8). Η εντολή mult εφαρμόζει το ζεύγος διανυσμάτων δοκιμής. Οι υπόλοιπες εντολές (γραμμές 12-14) υλοποιούν τη συνθήκη ελέγχου επανάληψης του βρόχου. Ο βρόχος επαναλαμβάνεται μέχρι να εφαρμοστούν όλα τα ζεύγη των διανυσμάτων δοκιμής.

5.3.2 Λογισμικό Αυτοδοκιμής βασισμένο σε ψευδοτυχαία διανύσματα δοκιμής

Η δοκιμή των κυκλωμάτων με ψευδοτυχαία διανύσματα αποτελεί μια ελκυστική προσέγγιση, αφού στην πραγματικότητα δεν απαιτείται μελέτη του κυκλώματος για εξαγωγή διανυσμάτων δοκιμής (όπως γίνεται με τα αυτόματα εργαλεία εξαγωγής διανυσμάτων). Δημοφιλείς γεννήτορες για ακολουθίες ψευδοτυχαίων αριθμών είναι οι *γραμμικοί καταχωρητές ολίσθησης με ανάδραση (linear feedback shift registers-LFSRs)* και τα *κνυελωτά αυτόματα (Cellular Automata-CA)* (βλ. Κεφάλαιο 2). Η υλοποίηση των παραπάνω μονάδων παραγωγής ψευδοτυχαίων διανυσμάτων με χρήση λογισμικού είναι ιδιαίτερα απλή.

Το λογισμικό αυτοδοκιμής που παρουσιάζεται στην Εικόνα 5.5 εφαρμόζει ψευδοτυχαίες ακολουθίες διανυσμάτων δοκιμής που παράγονται από μια ρουτίνα η οποία υλοποιεί, με εντολές μηχανής, ένα γραμμικό καταχωρητή ολίσθησης με ανάδραση. Στις γραμμές 1-2 γίνεται η αρχικοποίηση του γραμμικού καταχωρητή ολίσθησης. Η τιμή *polynomial* αναφέρεται στους συντελεστές του χαρακτηριστικού πολυωνύμου, ενώ η τιμή *seed* είναι η αρχική τιμή του καταχωρητή. Αλλάζοντας είτε την αρχική τιμή, είτε το χαρακτηριστικό πολυώνυμο, παράγονται νέες ακολουθίες ψευδοτυχαίων αριθμών.

Ο καταχωρητής \$t0 (γραμμή 3) λειτουργεί ως μετρητής διανυσμάτων που έχουν εφαρμοστεί στον πολλαπλασιαστή. Η εξαγωγή των διανυσμάτων γίνεται στις γραμμές 5 και 6. Διαφορετικές ακολουθίες παράγονται για κάθε τελεστέο. Τα διανύσματα αποθηκεύονται στους καταχωρητές \$s0 και \$s1 και εφαρμόζονται στον πολλαπλασιαστή (γραμμή 7). Το υπόλοιπο τμήμα του λογισμικού στις γραμμές 9-11 υλοποιεί ένα βρόχο επανάληψης μέχρι να εφαρμοστεί το επιθυμητό πλήθος διανυσμάτων δοκιμής και να επιτευχθεί το απαραίτητο ποσοστό κάλυψης ελαττωμάτων.

```

1  li $s2, seed
2  li $s3, polynomial
3  addi $t0, $t0, 0x0001
4  mul_loop:
5  # call LFSR routine for $s0
6  # call LFSR routine for $s1
7  mult $s0, $s1
8  # compaction code
9  sltiu $t1, $t0, number_of_test_patterns
10 bnez $t1, mul_loop
11 addiu $t0, $t0, 0x0001

```

Εικόνα 5.5: Τα ψευδοτυχαία διανύσματα δοκιμής εξάγονται με τη βοήθεια γραμμικών καταχωρητών ολίσθησης με ανάδραση υλοποιημένους σε λογισμικό.

5.3.3 Λογισμικό Αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα δοκιμής

Το λογισμικό αυτοδοκιμής εφαρμόζει προ-υπολογισμένα (pre-computed) διανύσματα δοκιμής στις υπομονάδες του μικροεπεξεργαστή. Τα προ-υπολογισμένα διανύσματα δοκιμής είναι ιδιαίτερα χρήσιμα όταν δεν είναι διαθέσιμη η περιγραφή του κυκλώματος υπό δοκιμή σε επίπεδο πυλών και δεν είναι δυνατό να χρησιμοποιηθεί ένα αυτόματο εργαλείο παραγωγής διανυσμάτων.

Η βιβλιογραφία παρουσιάζει πλήθος εργασιών που πραγματεύονται την ανάπτυξη προ-υπολογισμένων διανυσμάτων δοκιμής [113], [120], [121], [122], [114], [123]. Η ανάγκη για εύκολη και αποδοτική υλοποίηση της αυτοδοκιμής με υλικό οδήγησε στην ανάπτυξη προ-υπολογισμένων διανυσμάτων δοκιμής. Τα διανύσματα που παράγονται από ένα αυτόματο εργαλείο εξαγωγής διανυσμάτων είναι ακανόνιστα, δεν παρουσιάζουν καμία συμμετρία και έτσι δεν είναι δυνατό να χρησιμοποιηθούν για υλοποίηση της αυτοδοκιμής με υλικό. Αντίθετα, τα προ-υπολογισμένα διανύσματα δοκιμής έχουν αναπτυχθεί βάσει της συμμετρίας και της κανονικής δομής που παρουσιάζουν οι περισσότερες (κυρίως αριθμητικές) μονάδες. Η συμμετρία των μονάδων αντικατοπτρίζεται στα διανύσματα δοκιμής τα οποία σχετίζονται μεταξύ τους και εύκολα παράγονται το ένα από το άλλο με συνδυασμούς απλών αριθμητικών και λογικών πράξεων. Επιπλέον, τα προ-υπολογισμένα διανύσματα διαθέτουν την ιδιότητα της *προσαρμοστικότητας (scalability)*, δηλαδή, ένα σύνολο προ-υπολογισμένων διανυσμάτων δοκιμής εύκολα μπορεί να προσαρμοστεί ανάλογα με το μέγεθος της μονάδας υπό δοκιμή.

Η Εικόνα 5.6 παρουσιάζει τη δομή μιας ρουτίνας αυτοδοκιμής, η οποία εφαρμόζει προ-υπολογισμένα διανύσματα δοκιμής.

Η ρουτίνα αυτοδοκιμής αρχικοποιεί τους καταχωρητές \$s0, \$s1 με τα αρχικά διανύσματα δοκιμής από το σύνολο των προ-υπολογισμένων διανυσμάτων (γραμμές 1-2). Οι καταχωρητές

```

1  li $s0, initial_value_X
2  li $s1, initial_value_Y
3
4  addiu $t0, $zero, 0x0001
5  addiu $t1, $zero, 0x0001
6  loop:
7  mult $s0, $s1
8  # compaction code
9  sltui $t2,$t0,#_Y_test_patterns
10 # prepare next Y pattern
11 bnez $t2, loop
12 addiu $t1, $t1, 0x0001
13
14 add $s1, $zero, $zero
15 add $t1, $zero, $zero
16 sltu $t2,$t1,#_X_test_patterns
17 # prepare next X pattern
18 bnez $t2, loop
19 addiu $t0, $t0, 0x0001

```

Εικόνα 5.6: Δομή ρουτίνας αυτοδοκιμής βασισμένη σε προ-υπολογισμένα διανύσματα δοκιμής.

`$t0`, `$t1` χρησιμοποιούνται ως μετρητές του πλήθους των διανυσμάτων που θα εφαρμοστούν (γραμμές 4-5). Στην συνέχεια πραγματοποιείται η εφαρμογή του ζεύγους δοκιμής στον πολλαπλασιαστί μέσω της εντολής `mult` (γραμμή 7) και έπειτα ακολουθεί η διαδικασία της σύμπτυξης των αποκρίσεων (γραμμή 8). Η εντολή `sltui` (γραμμή 9) σε συνδυασμό με τις εντολές στις γραμμές 11-12 υλοποιούν ένα βρόχο επανάληψης ο οποίος εφαρμόζει διανύσματα δοκιμής. Το σχόλιο `prepare next Y pattern` (γραμμή 10) αναφέρεται στην εξαγωγή μιας νέας τιμής, ενός νέου διανύσματος δοκιμής για τον τελεστέο του καταχωρητή `$s1`. Η διαδικασία της εξαγωγής του νέου διανύσματος μπορεί να αποτελείται από την πρόσθεση (αφαίρεση) μιας σταθεράς στον (από τον) καταχωρητή `$s1` ή μιας ολίσθησης ή ένας συνδυασμός απλών εντολών οι οποίες δημιουργούν το επόμενο διάνυσμα δοκιμής. Εφόσον γίνεται λόγος για προ-υπολογισμένα διανύσματα δοκιμής τα οποία έχουν αναπτυχθεί βάσει της συμμετρίας των υπομονάδων, οι εντολές που παράγουν το επόμενο διάνυσμα δοκιμής είναι ιδιαίτερα απλές και το πλήθος τους δεν ξεπερνά συνήθως τις 5 εντολές. Σε ορισμένες περιπτώσεις η πρόσθεση μιας σταθεράς σε ένα καταχωρητή αρκεί για τη δημιουργία του νέου διανύσματος δοκιμής.

Οι εντολές στις γραμμές 16-19 υλοποιούν ένα βρόχο επανάληψης ο οποίος δημιουργεί ένα νέο διάνυσμα δοκιμής για τον τελεστέο του καταχωρητή `$s0`. Αντίστοιχα, το σχόλιο `prepare next X pattern` (γραμμή 17) αναφέρεται στην εξαγωγή του νέου διανύσματος δοκιμής.

Συνοψίζοντας, η ρουτίνα αυτοδοκιμής εφαρμόζει συνολικά `#_X_test_patterns` × `#_Y_test_patterns`, όπου για καθένα διάνυσμα από τα `#_X_test_patterns` του καταχωρητή `$s0`

εφαρμόζονται `#_Y_test_patterns` του καταχωρητή `$s1`.

5.4 Σύμπτυξη των αποκρίσεων

Οι αποκρίσεις των διανυσμάτων δοκιμής θα πρέπει να συμπτύσσονται ώστε να σχηματίζεται ένας μικρός αριθμός τελικών υπογραφών βάσει των οποίων θα πραγματοποιείται η σύγκριση με προαποθηκευμένες υπογραφές. Η σύμπτυξη των αποκρίσεων μπορεί να γίνει μετά την ολοκλήρωση του λογισμικού αυτοδοκιμής, όπου οι αποκρίσεις βρίσκονται αποθηκευμένες στη μνήμη του μικροεπεξεργαστή. Όμως, κάτι τέτοιο δεν αποτελεί καλή επιλογή, διότι κατά πρώτον μπορεί να μην υπάρχει διαθέσιμος χώρος στη μνήμη για την αποθήκευση των αποκρίσεων, ενώ, κατά δεύτερον, ακόμα και όταν υπάρχει διαθέσιμος χώρος δημιουργούνται άσκοπες συναλλαγές με το σύστημα της μνήμης. Γενικότερα, η σύμπτυξη των διανυσμάτων μετά την εκτέλεση του λογισμικού αυτοδοκιμής οδηγεί σε λογισμικό το οποίο έρχεται σε αντίθεση με τα επιθυμητά χαρακτηριστικά που παρουσιάστηκαν στις προηγούμενες ενότητες.

Η σύμπτυξη των αποκρίσεων όπως φαίνεται στις ρουτίνες αυτοδοκιμής των Σχημάτων 5.3, 5.4, 5.5 και 5.6 πραγματοποιείται αμέσως μετά τη δημιουργία της απόκρισης από την εφαρμογή του ζεύγους δοκιμής. Δεν πραγματοποιείται καμία συναλλαγή με το σύστημα της μνήμης, αφού καμία απόκριση δεν αποθηκεύεται σε αυτή. Η πραγματοποίηση της σύμπτυξης αμέσως μετά την γέννηση των αποκρίσεων μπορεί να δυσχεράνει την ανάπτυξη του λογισμικού αυτοδοκιμής καθώς θα πρέπει να δεσμευθούν επιπλέον καταχωρητές για την επεξεργασία των υπογραφών. Τέτοια προβλήματα μπορεί να προκύψουν μόνο όταν η δοκιμή πραγματοποιείται με προ-υπολογισμένα διανύσματα και χρησιμοποιούνται όλοι οι καταχωρητές για την ανάπτυξη των διανυσμάτων. Σε αυτήν την περίπτωση, απαιτείται μια διαμέριση της εκτέλεσης της δοκιμής η οποία αυξάνει το χρόνο εκτέλεσης της δοκιμής. Ευτυχώς, σχεδόν σε όλες τις περιπτώσεις το πλήθος των διαθέσιμων καταχωρητών είναι αρκετό ώστε η σύμπτυξη να πραγματοποιείται αμέσως μετά την γέννηση της απόκρισης με χρήση των καταχωρητών και μόνο.

5.5 Πειραματικά αποτελέσματα

Καθεμία από τις τεχνικές ανάπτυξης λογισμικού για ακέραιες μονάδες οδηγεί σε λογισμικό αυτοδοκιμής με διαφορετικά χαρακτηριστικά. Το λογισμικό αυτοδοκιμής που χρησιμοποιεί διανύσματα δοκιμής από ένα αυτόματο εργαλείο και τα εφαρμόζει σε ένα βρόχο επανάληψης-`ATPG_LOOP` (βλ. Εικόνα 5.4) είναι αποδοτικό ως προς την κρυφή μνήμη εντολών αλλά όχι και προς την κρυφή μνήμη δεδομένων. Ο βρόχος επανάληψης χαρακτηρίζεται από χρονική τοπικότητα (*temporal locality*) και έπειτα από ένα μικρό αριθμό υποχρεωτικών αστοχιών (*compulsory misses*), οι εντολές του βρόχου θα βρίσκονται εξ' ολοκλήρου στην κρυφή μνήμη δεδομένων. Αντίθετα, οι αναφορές στην μνήμη δεδομένων δεν παρουσιάζουν ούτε τοπικότητα στο χρόνο

ούτε τοπικότητα στο χώρο, καθώς η ανάγνωση στις θέσεις μνήμης πραγματοποιείται διαδοχικά. Ο αριθμός των αστοχιών στην κρυφή μνήμη εντολών εξαρτάται κατά κύριο λόγο από τον αριθμό των διανυσμάτων δοκιμής.

Η ακριβώς αντίθετη συμπεριφορά παρατηρείται όταν τα διανύσματα δοκιμής ενός αυτόματου εργαλείου που εφαρμόζονται χρησιμοποιώντας εντολές άμεσης διευθυνσιοδότησης-ATPG_{IMM} (βλ. Εικόνα 5.3). Η έκδοση αυτή είναι στην ουσία η ξετυλιγμένη (unrolled) έκδοχή του βρόχου επανάληψης. Αναφορές στην κρυφή μνήμη δεδομένων δεν υπάρχουν, οπότε δεν αναμένονται και αστοχίες (στην κρυφή μνήμη). Αστοχίες αναμένονται στην κρυφή μνήμη εντολών, αφού η ανάπτυξη του λογισμικού με εντολές άμεσης διευθυνσιοδότησης δεν χαρακτηρίζεται από τοπικότητα ούτε στο χώρο ούτε και στο χρόνο. Το πλήθος των αστοχιών στην κρυφή μνήμη εντολών εξαρτάται από τον αριθμό των διανυσμάτων.

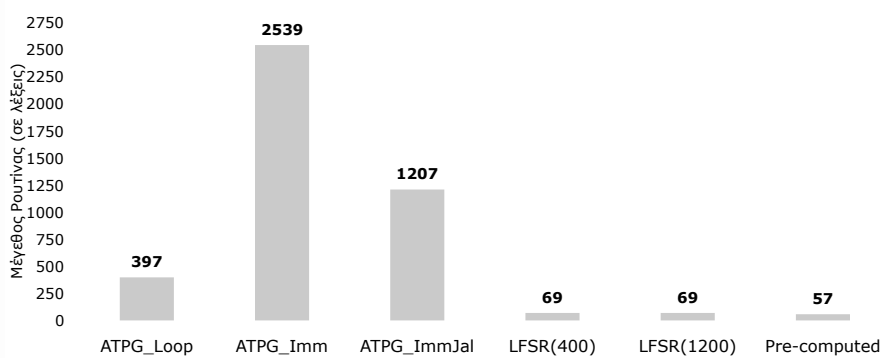
Το λογισμικό αυτοδοκιμής που εφαρμόζει ψευδοτυχαία διανύσματα δοκιμής-LFSR (βλ. Εικόνα 5.5) παρουσιάζει ιδανικά χαρακτηριστικά. Είναι συμπαγές (compact), αφού βασίζεται σε ένα βρόχο επανάληψης εκμεταλλευόμενο τη χρονική τοπικότητα και δεν πραγματοποιεί καμία αναφορά στην μνήμη δεδομένων. Μετά από ένα μικρό αριθμό αστοχιών στην κρυφή μνήμη εντολών, όλες οι εντολές του βρόχου θα βρίσκονται στην κρυφή μνήμη. Το μειονέκτημα που παρουσιάζει η συγκεκριμένη τεχνική ανάπτυξης λογισμικού είναι η εκτέλεση για ένα μεγάλο αριθμό κύκλων εντολών, που μπορεί να φτάσει μέχρι και πολλές εκατοντάδες χιλιάδες κύκλους εντολών. Η αιτία είναι πως δεν είναι όλα τα κυκλώματα εύκολα δοκιμάσιμα με ψευδοτυχαία διανύσματα και πρέπει να εφαρμοστούν μεγάλες ακολουθίες ψευδοτυχαίων διανυσμάτων για να επιτευχθεί ένα υψηλό ποσοστό κάλυψης ελαττωμάτων.

Τα προ-υπολογισμένα ή αλλιώς ντετερμινιστικά διανύσματα δοκιμής επιτρέπουν την ανάπτυξη λογισμικού αυτοδοκιμής-Pre-computed (βλ. Εικόνα 5.6) με παρόμοια χαρακτηριστικά με αυτά του λογισμικού που εφαρμόζει ψευδοτυχαία διανύσματα. Η ανάπτυξη του λογισμικού αυτοδοκιμής στηρίζεται σε βρόχους επανάληψης που παράγουν τα προ-υπολογισμένα διανύσματα όπως θα τα παράγαγε το υλικό. Το λογισμικό παραμένει συμπαγές (compact) χωρίς να πραγματοποιεί τόσες αναφορές στην μνήμη δεδομένων όσες και το λογισμικό που εφαρμόζει διανύσματα από αυτόματα εργαλεία. Επίσης, η χρήση των βρόχων επανάληψης οδηγεί σε τοπικότητα στο χρόνο και έτσι μετά από ένα μικρό αριθμό αστοχιών στην μνήμη εντολών, το λογισμικό αυτοδοκιμής θα εκτελείται από την κρυφή μνήμη. Σε αντίθεση με τα ψευδοτυχαία διανύσματα, τα προ-υπολογισμένα διανύσματα παρέχουν υψηλότερα ποσοστά κάλυψης ελαττωμάτων. Ο Πίνακας 5.1 παρουσιάζει τη συμπεριφορά του λογισμικού αυτοδοκιμής όπως αυτό προκύπτει από κάθε τεχνική ανάπτυξης, σε σχέση με τις αστοχίες στην κρυφή μνήμη εντολών και δεδομένων καθώς και με το πλήθος των διανυσμάτων δοκιμής που εφαρμόζονται.

Στην συνέχεια συγκρίνονται έξι διαφορετικές ρουτίνες αυτοδοκιμής του πολλαπλασιαστή στον επεξεργαστή Plasma. Ο μικροεπεξεργαστής Plasma υλοποιεί το σύνολο εντολών του MIPS-I. Ο πολλαπλασιαστής του μικροεπεξεργαστή Plasma είναι ένας παράλληλος πολλαπλασιαστής με κωδικοποίηση Booth, ο οποίος χρησιμοποιεί δεντρικές δομές τύπου Wallace

Λογισμικό Αυτοδοκιμής	Ρυθμός Αστοχίας κρυφή μνήμη εντολών	Ρυθμός Αστοχίας κρυφή μνήμη δεδομένων	Χρόνος Εκτέλεσης
ATPG _{IMM}	Υψηλός	Χαμηλός	Χαμηλός
ATPG _{LOOP}	Χαμηλός	Υψηλός	Χαμηλός
LFSR	Χαμηλός	Χαμηλός	Υψηλός
Pre-computed	Χαμηλός	Χαμηλός	Χαμηλός

Πίνακας 5.1: Τα χαρακτηριστικά του λογισμικού αυτοδοκιμής ανάλογα με τον τρόπο ανάπτυξής του.



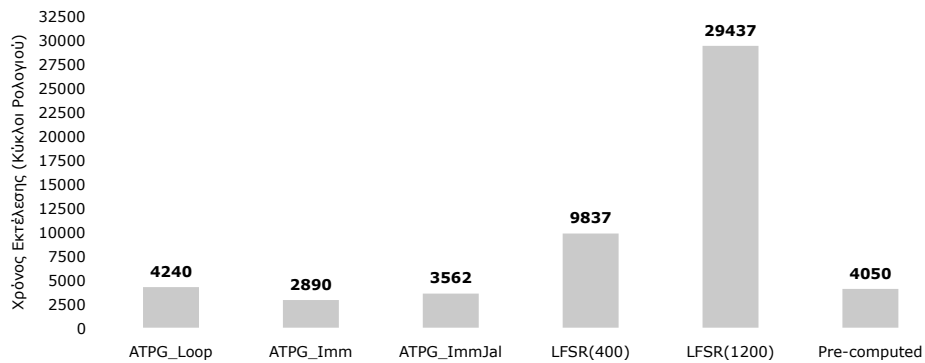
Εικόνα 5.7: Σύγκριση του ίχνους μνήμης για τις ρουτίνες αυτοδοκιμής του πολλαπλασιαστή.

για την άθροιση των μερικών γινομένων. Το αυτόματο εργαλείο εξαγωγής διανυσμάτων της MentorGraphics παράγαγε 168 διανύσματα δοκιμής⁶. Τα διανύσματα δοκιμής μετατράπηκαν σε ρουτίνες αυτοδοκιμής με τρεις διαφορετικούς τρόπους. Η ρουτίνα TP_{ATPG_Loop} εφαρμόζει τα διανύσματα δοκιμής με ένα βρόχο επανάληψης, ενώ η ρουτίνα TP_{ATPG_Imm} χρησιμοποιεί εντολές άμεσης διευθυνσιοδότησης για την εφαρμογή των διανυσμάτων. Η ρουτίνα TP_{ATPG_ImmJal} αποτελεί βελτιστοποίηση της προηγούμενης ρουτίνας όπου το τμήμα των εντολών που πραγματοποιεί τη σύμπτυξη των αποκρίσεων έχει μετατραπεί σε μια ξεχωριστή ρουτίνα η οποία καλείται αμέσως μετά την εντολή του πολλαπλασιασμού. Οι ρουτίνες TP_{LFSR(400)} και TP_{LFSR(1200)} εφαρμόζουν 400 και 1200 ζεύγη ψευδοτυχαίων διανυσμάτων δοκιμής στον πολλαπλασιαστή. Η τελευταία ρουτίνα TP_{pre-computed} εφαρμόζει τα προ-υπολογισμένα διανύσματα της εργασίας [121] στον πολλαπλασιαστή.

Η Εικόνα 5.7 παρουσιάζει το ίχνος μνήμης (εντολές και δεδομένα) σε λέξεις για καθεμία από τις παραπάνω ρουτίνες αυτοδοκιμής.

Οι ρουτίνες που εφαρμόζουν διανύσματα δοκιμής από εργαλεία αυτόματης εξαγωγής διαθέτουν πολύ μεγαλύτερο ίχνος, ως και 40 φορές μεγαλύτερο, σε σχέση με τις ρουτίνες που

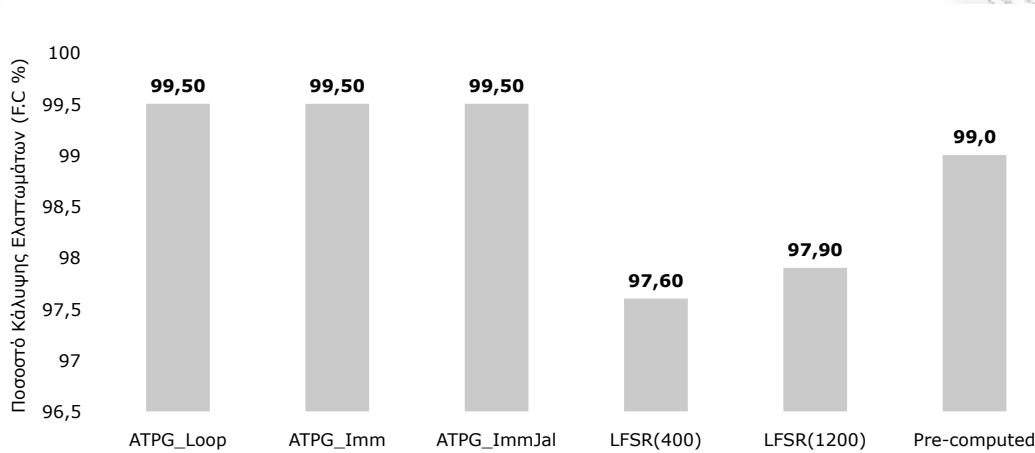
⁶Ένα διάνυσμα δοκιμής αποτελείται από δύο μέρη, ένα για τον τελεστή X και ένα για τον τελεστή Y.



Εικόνα 5.8: Σύγκριση του χρόνου εκτέλεσης των ρουτινών αυτοδοκιμής του πολλαπλασιαστή.

εφαρμόζουν ψευδοτυχαίες ακολουθίες διανυσμάτων και προ-υπολογισμένα διανύσματα. Το ίχνος μνήμης των ρουτινών εξαρτάται άμεσα από το πλήθος των ζευγών διανυσμάτων δοκιμής. Ο καλύτερος τρόπος να εφαρμόσει κανείς διανύσματα από αυτόματα εργαλεία εξαγωγής είναι αυτός που παρουσιάζεται στην Εικόνα 5.4. Η ρουτίνα TP_{ATPG_Loop} χρειάζεται συνολικά 397 διαθέσιμες λέξεις στην μνήμη (εντολές και δεδομένα). Από τις 397 λέξεις μνήμης, οι 336 (2×168 ζεύγη διανυσμάτων) χρειάζονται για την αποθήκευση των διανυσμάτων δοκιμής στη μνήμη και συνολικά πραγματοποιούνται 337 αναφορές στη μνήμη δεδομένων. Οι ρουτίνες TP_{ATPG_Imm} και TP_{ATPG_ImmJal} έχουν το μεγαλύτερο ίχνος μνήμης, αφού για κάθε ένα από τα 336 διανύσματα δοκιμής χρειάζονται δύο εντολές ώστε να σχηματιστεί το διάνυσμα σε ένα καταχωρητή. Αντίθετα, οι ρουτίνες TP_{LFSR} και $TP_{pre-computed}$ έχουν μικρό ίχνος μνήμης και δεν πραγματοποιούν αναφορές στη μνήμη δεδομένων παρά μόνο για να αποθηκεύσουν την υπογραφή και είναι περισσότερο κατάλληλες για την αυτοδοκιμή του μικροεπεξεργαστή στο πεδίο λειτουργίας.

Η Εικόνα 5.8 παρουσιάζει το πλήθος των κύκλων που απαιτούνται για την εκτέλεση της κάθε ρουτίνας αυτοδοκιμής. Όλες οι ρουτίνες αυτοδοκιμής, εκτός από τις ρουτίνες που εφαρμόζουν ακολουθίες ψευδοτυχαίων διανυσμάτων δοκιμής, οι $TP_{LFSR}(400)$ και $TP_{LFSR}(1200)$, εκτελούνται από τον μικροεπεξεργαστή σε λίγες χιλιάδες κύκλους ρολογιού, με τη ρουτίνα TP_{ATPG_Imm} να εκτελείται πιο γρήγορα από όλες. Οι ρουτίνες $TP_{LFSR}(400)$ και $TP_{LFSR}(1200)$ χρειάζονται περισσότερο χρόνο εκτέλεσης από τις υπόλοιπες ρουτίνες διότι εφαρμόζουν περισσότερα ζεύγη διανυσμάτων δοκιμής. Ο χρόνος εκτέλεσης εξαρτάται από το πλήθος των διανυσμάτων που εφαρμόζονται. Η ρουτίνα $TP_{LFSR}(1200)$ χρειάζεται σχεδόν τριπλάσιο χρόνο εκτέλεσης σε σχέση με την $TP_{LFSR}(400)$. Καθώς η ταχύτητα λειτουργίας των μικροεπεξεργαστών γίνεται ολοένα και πιο γρήγορη, το μεγάλος πλήθος κύκλων ρολογιού που απαιτούν οι ρουτίνες TP_{LFSR} υποβαθμίζεται. Οι ρουτίνες που εφαρμόζουν ψευδοτυχαία διανύσματα δοκιμής



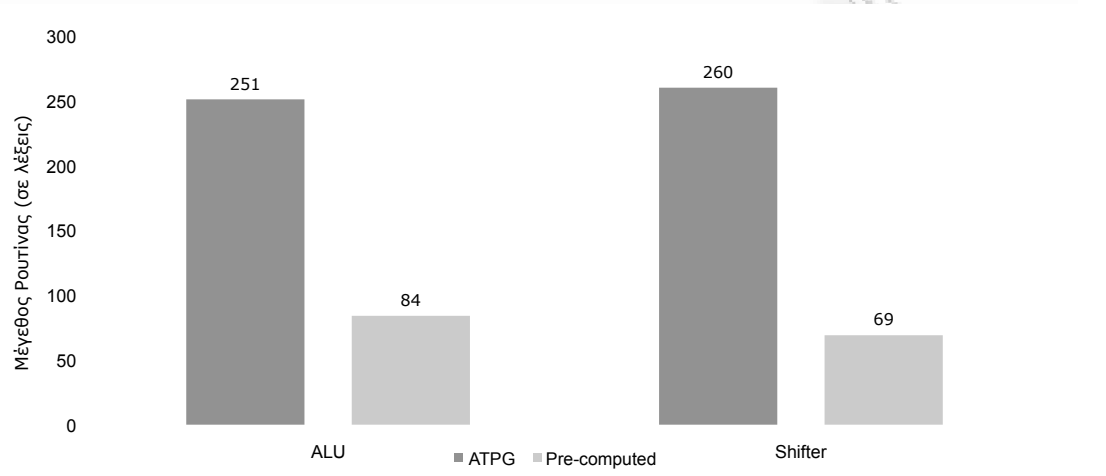
Εικόνα 5.9: Σύγκριση του ποσοστού κάλυψης ελαττωμάτων που επιτυγχάνει κάθε ρουτίνα αυτοδοκιμής του πολλαπλασιαστή.

παρουσιάζουν και μια άλλη αδυναμία. Δεν εγγυώνται υψηλά ποσοστά κάλυψης ελαττωμάτων όπως φαίνεται στην Εικόνα 5.9. Η ρουτίνα $TP_{LFSR(400)}$ εφαρμόζει 400 ζεύγη διανυσμάτων επιτυγχάνοντας ποσοστό κάλυψης ελαττωμάτων 97,6% ενώ όταν εφαρμόζεται τριπλάσιος αριθμός ζευγαριών διανυσμάτων με τη ρουτίνα $TP_{LFSR(1200)}$ το ποσοστό κάλυψης αυξάνεται μόλις κατά 0,3%. Στον αντίποδα, οι ρουτίνες που εφαρμόζουν διανύσματα που εξάγονται από αυτόματα εργαλεία εξασφαλίζουν το υψηλότερο ποσοστό κάλυψης ελαττωμάτων.

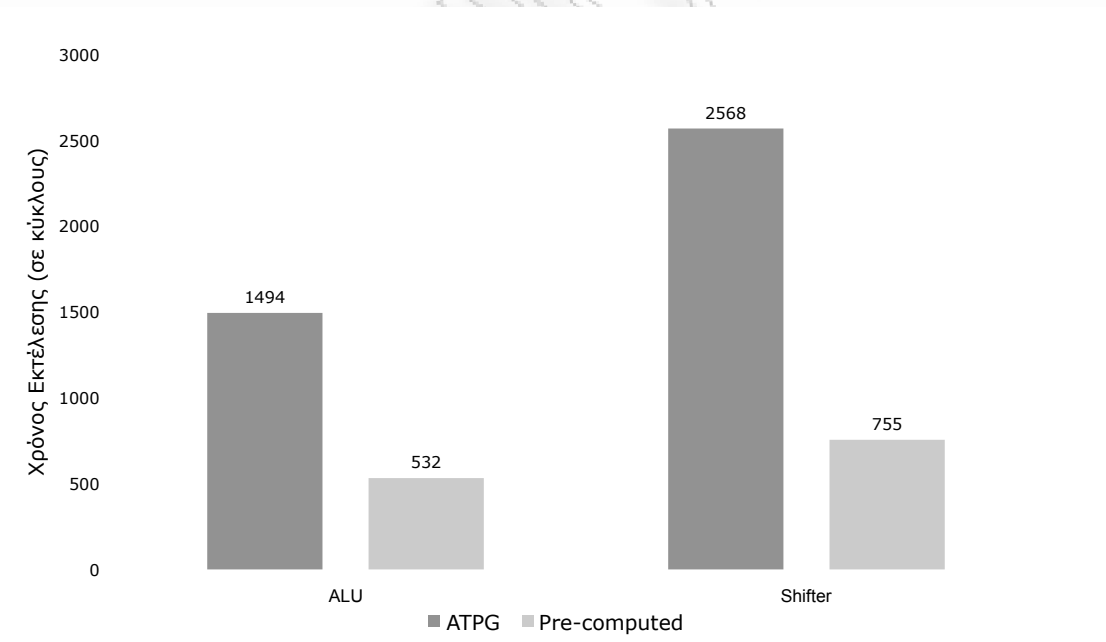
Η ρουτίνα $TP_{pre-computed}$ επιτυγχάνει και αυτή υψηλό ποσοστό κάλυψης ελαττωμάτων, 99,0% και υπολείπεται μόνο κατά 0,5% σε σχέση με τις ρουτίνες TP_{ATPG_Loop} , TP_{ATPG_Imm} και TP_{ATPG_ImmJal} . Όσον αφορά το ίχνος μνήμης η ρουτίνα $TP_{pre-computed}$ είναι περίπου 85,6% μικρότερη σε σχέση με την TP_{ATPG_Loop} και έχει μικρότερο χρόνο εκτέλεσης περίπου 4,4%. Ακόμα δεν πραγματοποιεί καμία επιπλέον αναφορά στη μνήμη δεδομένων σε αντίθεση με την ρουτίνα TP_{ATPG_Loop} .

Οι ιδιότητες του λογισμικού αυτοδοκιμής που αναπτύσσεται βάσει προ-υπολογισμένων διανυσμάτων δοκιμής διατηρούνται και στις υπόλοιπες λειτουργικές μονάδες των μικροεπεξεργαστών, όπως είναι η αριθμητική και λογική μονάδα και ο ολισθητής⁷. Οι Εικόνες 5.10, 5.11, και 5.12 παρουσιάζουν τα στατιστικά των ρουτινών αυτοδοκιμής για την αριθμητική και λογική μονάδα (ALU) και τον ολισθητή (shifter). Με έντονο γκρι παρουσιάζονται τα στοιχεία των ρουτινών τα οποία έχουν αναπτυχθεί βάσει διανυσμάτων δοκιμής που έχουν εξαχθεί από αυτόματα εργαλεία ενώ με ανοιχτό γκρι τα στοιχεία των ρουτινών που έχουν αναπτυχθεί βάσει προ-υπολογισμένων διανυσμάτων δοκιμής.

⁷Ο ολισθητής (shifter) του Plasma ολισθαίνει το περιεχόμενο ενός καταχωρητή από 0 ως 31 θέσεις ή αριστερά ή δεξιά. Η δεξιά ολίσθηση μπορεί να είναι λογική ολίσθηση ή αριθμητική ολίσθηση.

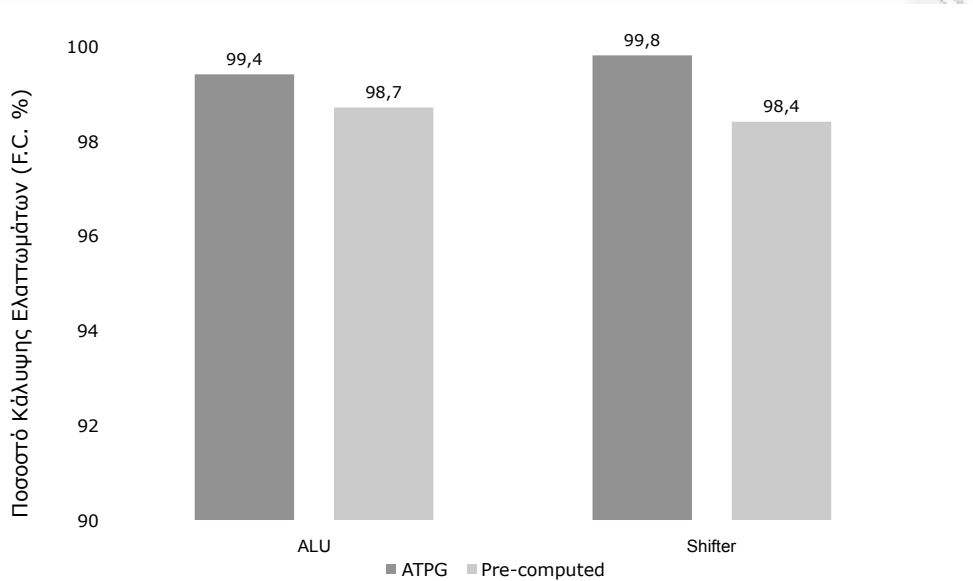


Εικόνα 5.10: Συγκριτικά αποτελέσματα του μεγέθους των ρουτινών αυτοδοκιμής.



Εικόνα 5.11: Συγκριτικά αποτελέσματα του χρόνου εκτέλεσης των ρουτινών αυτοδοκιμής.

Συνοψίζοντας, το λογισμικό αυτοδοκιμής που βασίζεται σε προ-υπολογισμένα διανύσματα έχει σαφώς μικρότερο ίχνος μνήμης από το αντίστοιχο που βασίζεται σε διανύσματα δοκιμής από αυτόματα εργαλεία, όπως φαίνεται από τις Εικόνες 5.7 και 5.10. Το λογισμικό αυτοδοκιμής από διανύσματα δοκιμής εξαγόμενα από αυτόματα εργαλεία έχει αναπτυχθεί σύμφωνα με την ενότητα 5.3.1. Αναλυτικότερα, το ίχνος του λογισμικού αυτοδοκιμής βασισμένο σε



Εικόνα 5.12: Συγκριτικά αποτελέσματα του ποσοστού κάλυψης ελαττωμάτων των ρουτινών αυτοδοκιμής.

προ-υπολογισμένα διανύσματα είναι κατά μέσο όρο 70% μικρότερο σε σχέση με αυτό που αναπτύσσεται βάσει διανυσμάτων που εξάγονται από εργαλεία εξαγωγής και χρειάζεται κατά μέσο όρο 67,5% μικρότερο χρόνο εκτέλεσης.

Τα ποσοστά κάλυψης ελαττωμάτων των διανυσμάτων δοκιμής που παράγονται από αυτόματα εργαλεία είναι ελαφρώς υψηλότερα από αυτά των προ-υπολογισμένων διανυσμάτων, καθώς τα εργαλεία εξαγωγής αναλύουν κάθε κόμβο του κυκλώματος. Όπως φαίνεται από τις Εικόνες 5.9 και 5.12, η διαφορά στα ποσοστά κάλυψης είναι κατά μέσο όρο 1% και είναι ιδιαίτερα μικρή. Επιπλέον, τα αυτόματα εργαλεία εξαγωγής διανυσμάτων αντιμετωπίζουν δυσκολίες όταν μελετούν ακολουθιακά κυκλώματα και μπορεί και μην επιτύχουν την εξαγωγή διανυσμάτων με υψηλά ποσοστά κάλυψης. Ένα χαρακτηριστικό παράδειγμα είναι το αρχείο καταχωρητών (register file) για το οποίο τα εργαλεία εξαγωγής αποτυγχάνουν να παράξουν διανύσματα με υψηλά ποσοστά κάλυψης. Αντίθετα, στην εργασία [89] παρουσιάζεται η εξαγωγή προ-υπολογισμένων διανυσμάτων δοκιμής που επιτυγχάνουν ποσοστά κάλυψης μεγαλύτερα του 99%.

Ολοκληρώνοντας την ανάλυση των πειραματικών αποτελεσμάτων επί του λογισμικού αυτοδοκιμής ακεραίων μονάδων, ο Πίνακας 5.2 παρουσιάζει τα στατιστικά στοιχεία του λογισμικού αυτοδοκιμής όπως αυτό αναπτύχθηκε για όλες τις μονάδες του μικροεπεξεργαστή Plasma. Το λογισμικό αυτοδοκιμής έχει μικρό ίχνος μνήμης και καταλαμβάνει περίπου 3,7Kb μνήμης. Επιτυγχάνει ποσοστό κάλυψης ελαττωμάτων 94,6% ενώ ο συνολικός χρόνος εκτέλεσης του λογισμικού αυτοδοκιμής ανέρχεται σε 13.077 κύκλους. Υποθέτοντας μια συχνότητα λειτουρ-

γίας στα 50MHz ο χρόνος εκτέλεσης του λογισμικού αυτοδοκιμής ανέρχεται στα 74,4μsec, χρόνος πολύ μικρότερος από το κβάντο χρόνου σε λειτουργικό σύστημα με εξυπηρέτηση εκ περιτροπής.

Ίχνος μνήμης	930 λέξεις (3720 bytes)
Χρόνος Εκτέλεσης	13 077 (κύκλοι)
Ποσοστό Κάλυψης Ελαττωμάτων	94,6%

Πίνακας 5.2: Στατιστικά στοιχεία λογισμικού αυτοδοκιμής του μικροεπεξεργαστή Plasma.

5.6 Λογισμικό αυτοδοκιμής για μονάδες κινητής υποδιαστολής

Οι τεχνικές ανάπτυξης λογισμικού αυτοδοκιμής που παρουσιάστηκαν στις προηγούμενες ενότητες αναφέρονται σε μικροεπεξεργαστές που δε διαθέτουν μονάδες κινητής υποδιαστολής. Η ανάπτυξη του λογισμικού αυτοδοκιμής εστιάστηκε κυρίως στο μονοπάτι επεξεργασίας ακεραίων αριθμών (integer datapath). Όμως σήμερα οι μικροεπεξεργαστές ενσωματώνουν μονάδες κινητής υποδιαστολής ώστε να ανταποκριθούν στις ανάγκες των εφαρμογών (επεξεργασία εικόνας και ήχου) οι οποίες απαιτούν επεξεργασία πραγματικών αριθμών. Οι τεχνικές ανάπτυξης λογισμικού αυτοδοκιμής που παρουσιάστηκαν δεν εφαρμόζονται αυτούσιες στις μονάδες κινητής υποδιαστολής και έτσι απαιτείται ξεχωριστή μελέτη για την ανάπτυξη του λογισμικού εξαιτίας των ακόλουθων προκλήσεων.

- Περιορισμένες δυνατότητες του συνόλου των εντολών.

Το σύνολο εντολών των μικροεπεξεργαστών παρέχει περιορισμένες δυνατότητες για επεξεργασία των αριθμών κινητής υποδιαστολής. Γενικότερα οι εντολές του συνόλου εντολών μπορούν να χωριστούν στις εξής κατηγορίες: *αριθμητικές πράξεις*, όπως πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση, σύγκριση, απόλυτη τιμή, και αντιστροφή πρόσημου, *πράξεις μεταφοράς* αριθμών κινητής υποδιαστολής στη μνήμη και αντίστροφα, και *πράξεις μετατροπής* αριθμών κινητής υποδιαστολής σε ακεραίους και αντίστροφα. Όμως, τα σύνολα εντολών δεν παρέχουν τη δυνατότητα για επεξεργασία των επιμέρους πεδίων ενός αριθμού κινητής υποδιαστολής, όπως του εκθέτη ή του σημαινόμενου. Επομένως, δεν είναι δυνατή η εφαρμογή ψευδοτυχαίων τιμών μόνο στο σημαινόμενο.

- Επικοινωνία του μικροεπεξεργαστή με τη μονάδα κινητής υποδιαστολής.

Οι μονάδες κινητής υποδιαστολής συνοδεύονται από ένα αρχείο καταχωρητών (register file) το οποίο είναι διαφορετικό από αυτό των ακεραίων αριθμών. Η μεταφορά δεδομένων από το ένα αρχείο στο άλλο μπορεί να γίνει με δύο τρόπους: ή με ειδικές εντολές που

παρέχει το σύνολο εντολών, ή με ακολουθίες εντολών που μεταφέρουν τα δεδομένα από ακέραιο καταχωρητή στη μνήμη και μετά στον καταχωρητή κινητής υποδιαστολής, και αντίστροφα. Η μεταφορά των δεδομένων του καταχωρητή κινητής υποδιαστολής σε ακέραιο καταχωρητή είναι απαραίτητη ώστε να πραγματοποιηθεί η σύμπτυξη των αποκρίσεων (compaction). Υπενθυμίζουμε ότι τα σύνολα εντολών δεν παρέχουν εντολές λογικών πράξεων και ολίσθησης σε καταχωρητές κινητής υποδιαστολής.

- Πολυπλοκότητα των μονάδων κινητής υποδιαστολής.
Η πολυπλοκότητα των μονάδων κινητής υποδιαστολής είναι πολύ μεγαλύτερη σε σχέση με αυτή των ακέραιων μονάδων. Επομένως απαιτούνται περισσότερα διανύσματα δοκιμής για την αποτελεσματική δοκιμή τους. Έτσι όμως αυξάνονται κατά πολύ οι απαιτήσεις του λογισμικού αυτοδοκιμής σε μνήμη. Η παρατήρηση αυτή επιβεβαιώνεται από τα πειραματικά δεδομένα της διατριβής που θα δούμε στη συνέχεια.
- Ακρίβεια κινητής υποδιαστολής και αρχιτεκτονική επεξεργαστή.
Πολλές φορές οι μικροεπεξεργαστές ενσωματώνουν μονάδες κινητής υποδιαστολής διπλής ακρίβειας όπου απαιτούνται 64-bit. Παρόλ' αυτά οι επεξεργαστές αυτοί είναι υλοποιημένοι σε αρχιτεκτονική των 32-bit. Επομένως η υιοθέτηση του λογισμικού αυτοδοκιμής βασισμένο σε ψευδοτυχαίες ακολουθίες τιμών ή σε προ-υπολογισμένα διανύσματα γίνεται δυσκολότερη αφού πρέπει να συνενωθούν δύο καταχωρητές των 32-bit για να σχηματίσουν ένα καταχωρητή 64-bit. Είναι αντιληπτό πως τέτοια θέματα δεν τίθενται κατά την αυτοδοκιμή των ακέραιων αριθμητικών μονάδων.

Στις ενότητες 5.6.1 και 5.6.2 που ακολουθούν παρουσιάζονται οι μέθοδοι ανάπτυξης λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής απλής και διπλής ακρίβειας που βρίσκονται σε επεξεργαστές 32/64-bit που είτε διαθέτουν ειδικές εντολές επεξεργασίας bit είτε όχι. Ο Πίνακας 5.3 παρουσιάζει αντιπροσωπευτικούς συνδυασμούς μικροεπεξεργαστών και μονάδων κινητής υποδιαστολής για τις οποίες πραγματοποιείται ανάπτυξη λογισμικού αυτοδοκιμής. Αναλυτικότερα, ο μικροεπεξεργαστής MIPS έχει μέγεθος 32-bit και μπορεί να ενσωματώνει μονάδα κινητής υποδιαστολής είτε απλής είτε διπλής ακρίβειας, ενώ το σύνολο εντολών του δε διαθέτει ειδικές εντολές επεξεργασίας bit. Αντίθετα, το σύνολο εντολών του μικροεπεξεργαστή PowerPC, ο οποίος έχει μέγεθος 32-bit, διαθέτει ειδικές εντολές επεξεργασίας bit και όμοια με τον MIPS ενσωματώνει μονάδα κινητής υποδιαστολής είτε απλής είτε διπλής ακρίβειας. Τέλος, ο επεξεργαστής Alpha έχει μέγεθος 64-bit και ενσωματώνει μονάδα κινητής υποδιαστολής διπλής ακρίβειας ενώ το σύνολο εντολών του δε διαθέτει ειδικές εντολές επεξεργασίας ψηφίων.

Επεξεργαστής	Μέγεθος	Ακρίβεια Υποδιαστολής	Μονάδας Κινητής	Ειδικές εντολές επεξεργασίας σε επίπεδο bit
MIPS	32-bit	Απλή (32-bit)		Όχι
MIPS	32-bit	Διπλή (64-bit)		Όχι
PowerPC	32-bit	Απλή (32-bit)		Ναι
PowerPC	32-bit	Διπλή (64-bit)		Ναι
Alpha	64-bit	Διπλή (64-bit)		Όχι

Πίνακας 5.3: Συνδυασμοί επεξεργαστών και μονάδων κινητής υποδιαστολής για τους οποίους έγινε ανάπτυξη λογισμικού αυτοδοκιμής.

5.6.1 Ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες απλής ακρίβειας

Όπως αναφέραμε παραπάνω, το σύνολο εντολών των μικροεπεξεργαστών παρέχει περιορισμένες δυνατότητες όσον αφορά την επεξεργασία των πεδίων ενός καταχωρητή κινητής υποδιαστολής. Γενικότερα οι δυνατότητες που παρέχονται είναι αρκετές για την ανάπτυξη λογισμικού αυτοδοκιμής βασισμένο σε διανύσματα που εξάγονται από αυτόματα εργαλεία εξαγωγής διανυσμάτων δοκιμής.

Σε αυτή την περίπτωση το λογισμικό αυτοδοκιμής αποτελείται ή από ένα βρόχο, ο οποίος διαβάζει τα διανύσματα δοκιμής από τη μνήμη και τα εφαρμόζει, ή από διαδοχικές εντολές που σχηματίζουν τα διανύσματα δοκιμής χρησιμοποιώντας την άμεση διευθυνσιοδότηση των εντολών. Στη δεύτερη περίπτωση, τα διανύσματα δοκιμής σχηματίζονται σε ακέραιους καταχωρητές, των οποίων τα περιεχόμενα μεταφέρονται σε καταχωρητές κινητής υποδιαστολής. Οι Εικόνες 5.13 και 5.14 παρουσιάζουν τα πρότυπα ανάπτυξης λογισμικού αυτοδοκιμής σε ψευδοκώδικα για τη δοκιμή μιας μονάδας κινητής υποδιαστολής που υλοποιεί την πράξη FP_OP. Η ψευδοεντολή MOVE_FP που εμφανίζεται και στις δύο εικόνες χρησιμοποιείται για τη μεταφορά ενός καταχωρητή κινητής υποδιαστολής σε έναν ακέραιο. Η μεταφορά αυτή είναι απαραίτητη για την σύμπτυξη των αποτελεσμάτων διότι το σύνολο εντολών δεν επιτρέπει ούτε λογικές πράξεις αλλά ούτε και πράξεις ολίσθησης σε καταχωρητές κινητής υποδιαστολής. Η ψευδοεντολή MOVE_INT, στην Εικόνα 5.14 μεταφέρει ένα ακέραιο καταχωρητή σε έναν καταχωρητή κινητής υποδιαστολής.

Υπάρχουν δύο τρόποι για τη μεταφορά ενός καταχωρητή κινητής υποδιαστολής σε έναν ακέραιο και αντίστροφα:

- είτε με χρήση ειδικών εντολών που παρέχονται από το σύνολο εντολών,
- είτε μέσω της μνήμης.

Για παράδειγμα, το σύνολο εντολών του μικροεπεξεργαστή MIPS παρέχει τις ειδικές εντολές mtc1 (μεταφορά στο συνεπεξεργαστή 1) και mfc1 (μεταφορά από το συνεπεξεργαστή 1) για

```

1  R1 := AddressOfPatternsInMemory
2  R2 := NumberOfPatterns
3  # R3 holds the compaction signature
4  AtpgLoop:
5  FP_R1 := MEM[R1]
6  FP_R2 := MEM[R1 + 4]
7  FP_R3 := FP_R1 FP_OP FP_R2
8  R3 := MOVE_FP(FP_R3)
9  CompactionRoutine(R3)
10 R1 := R1 + 8
11 R2 := R2 - 2
12 BranchNotEqual R2, 0, AtpgLoop

```

Εικόνα 5.13: Εφαρμογή διανυσμάτων δοκιμής από εργαλείο εξαγωγής διανυσμάτων με χρήση βρόχου.

τη μεταφορά δεδομένων μεταξύ ακέραιων καταχωρητών και κινητής υποδιαστολής⁸. Παρόμοιες εντολές παρέχει και το σύνολο εντολών του επεξεργαστή Alpha. Αντίθετα, τα σύνολα εντολών των επεξεργαστών PowerPC και SPARC δεν παρέχουν ειδικές εντολές. Σε αυτή την περίπτωση η ακολουθία εντολών *αποθήκευση ακεραίου-φόρτωση κινητής υποδιαστολής (store integer-load FP)* υλοποιεί τη μεταφορά από ακέραιο σε κινητής υποδιαστολής. Η ακολουθία *αποθήκευση κινητής υποδιαστολής-ανάγνωση ακεραίου (store FP-load integer)* υλοποιεί την αντίθετη μεταφορά. Είναι ευνόητο πως κατά τη μεταφορά μέσω της μνήμης χρησιμοποιούνται επιπλέον κύκλοι ρολογιού και ο χρόνος εκτέλεσης του λογισμικού αυτοδοκιμής μεγαλώνει σημαντικά.

Η ψευδοεντολή Assemble στην Εικόνα 5.14 είναι μια ακολουθία εντολών άμεσης διευθυνσιοδότησης η οποία σταδιακά σχηματίζει έναν αριθμό κινητής υποδιαστολής σε έναν ακέραιο καταχωρητή. Στον επεξεργαστή MIPS η ψευδοεντολή αποτελείται από τις εντολές lui (load upper immediate) και ori (logic OR immediate).

Αντίθετα με τα πρότυπα ανάπτυξης λογισμικού αυτοδοκιμής με διανύσματα δοκιμής από εργαλεία εξαγωγής τα οποία μοιάζουν με αυτά της ενότητας 5.3.1, τα πρότυπα ανάπτυξης λογισμικού αυτοδοκιμής των ενότητων που βασίζεται σε ψευδοτυχαία διανύσματα (βλ. ενότητα 5.3.2) και σε προ-υπολογισμένα διανύσματα (βλ. ενότητα 5.3.3) δεν είναι άμεσα εφαρμόσιμα στις μονάδες κινητής υποδιαστολής εξαιτίας των περιορισμένων δυνατοτήτων του συνόλου εντολών. Και στις δύο περιπτώσεις είναι απαραίτητη η *επεξεργασία σε επίπεδο ψηφίων (bit level manipulation)* σε ακέραιους καταχωρητές. Μια ακολουθία εντολών για επεξεργασία σε επίπεδο ψηφίων διαχωρίζει νοητά έναν ακέραιο καταχωρητή στα τρία πεδία του αριθμού κινητής υποδιαστολής, το πρόσμιο, τον εκθέτη και το σημαινόμενο (βλ. Κεφ. 3). Στη συνέχεια,

⁸Ο συνεπεξεργαστής 1 στον MIPS είναι η μονάδα κινητής υποδιαστολής.

```

1  R1 := Assemble(Pattern1_OpA)
2  FP_R1 := MOVE_INT(R1)
3
4  R1 := Assemble(Pattern1_OpB)
5  FP_R2 := MOVE_INT(R1)
6
7  FP_R3 := FP_R1 FP_OP FP_R2
8  R2 := MOVE_FP(FP_R3)
9  CompactionRoutine(R2)
10 .
11 .
12 .
13 R1 := Assemble(PatternN_OpA)
14 FP_R1 := MOVE_INT(R1)
15
16 R1 := Assemble(PatternN_OpB)
17 FP_R2 := MOVE_INT(R1)
18
19 FP_R3 := FP_R1 FP_OP FP_R2
20 R2 := MOVE_FP(FP_R3)
21 CompactionRoutine(R2)

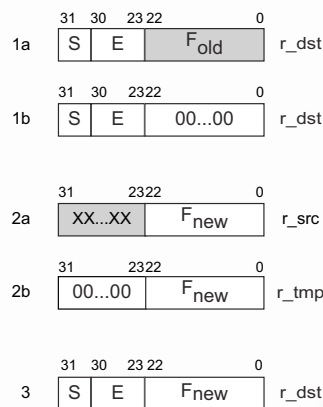
```

Εικόνα 5.14: Εφαρμογή διανυσμάτων δοκιμής από εργαλείο εξαγωγής διανυσμάτων δοκιμής με χρήση εντολών άμεσης διευθυνσιοδότησης.

κάθε πεδίο λαμβάνει την επιθυμητή τιμή ώστε να σχηματιστεί το απαιτούμενο διάνυσμα δοκιμής. Έπειτα, οι ακέραιοι καταχωρητές μεταφέρονται σε καταχωρητές κινητής υποδιαστολής για την τελική εφαρμογή των διανυσμάτων δοκιμής στη μονάδα κινητής υποδιαστολής.

Οι ρουτίνες `set_exp` (set exponent) και `set_sig` (set significand) υλοποιούν την ανάθεση τιμών μόνο στο πεδίο του εκθέτη και του σημαϊνόμενου, αντίστοιχα. Οι ρουτίνες επεξεργάζονται δύο καταχωρητές. Ο πρώτος καταχωρητής διατηρεί τον αριθμό κινητής υποδιαστολής ενώ ο δεύτερος διατηρεί το νέο σημαϊνόμενο ή το νέο εκθέτη που θα αντικαταστήσει την αντίστοιχη τιμή στον πρώτο καταχωρητή. Οι εντολές που απαρτίζουν τις παραπάνω ρουτίνες εξαρτώνται από το σύνολο εντολών του μικροεπεξεργαστή. Δε θα ασχοληθούμε με νέες ρουτίνες που επεξεργάζονται το πρόσημο του αριθμού κινητής υποδιαστολής αφού τέτοια δυνατότητα παρέχεται πάντοτε από το σύνολο εντολών (απόλυτη τιμή αριθμού κινητής υποδιαστολής και μετατροπή αριθμού κινητής υποδιαστολής στον αντίστοιχο αρνητικό).

Οι Εικόνες 5.15 και 5.16 παρουσιάζουν την λειτουργία των ρουτινών `set_sig` και `set_exp` στους ακέραιους καταχωρητές, αντίστοιχα. Ο καταχωρητής `r_dst` διατηρεί τον αριθμό κινητής υποδιαστολής, ο καταχωρητής `r_src` διατηρεί το σημαϊνόμενο ή τον εκθέτη και ο καταχωρητής `r_tmp` χρησιμοποιείται ως βοηθητικός.



Εικόνα 5.15: Η λειτουργία της ρουτίνας `set_sig` στους καταχωρητές `r_dst` και `r_src`.

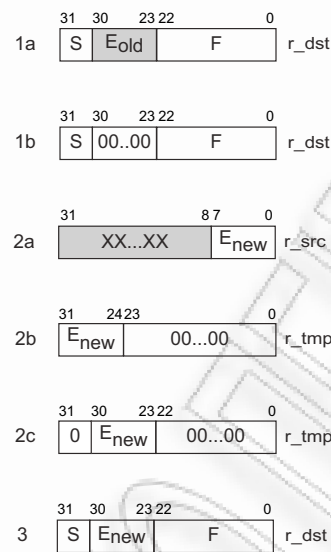
Στην Εικόνα 5.15, ο καταχωρητής `r_src` διατηρεί την τιμή του νέου σημαινόμενου (ψηφία 22 ως 0 στην απλή ακρίβεια αναπαράστασης) που θα αντικαταστήσει το αντίστοιχο πεδίο στον καταχωρητή `r_dst`. Αρχικά, ο καταχωρητής `r_dst` περιέχει έναν αριθμό κινητής υποδιαστολής με το παλαιό σημαινόμενο (ετικέτα 1a) και ο καταχωρητής `r_src` το νέο σημαινόμενο (ετικέτα 2a). Το παλαιό σημαινόμενο στον `r_dst` διαγράφεται (μηδενίζεται) (ετικέτα 1b). Με παρόμοιο τρόπο τα ψηλά ψηφία του `r_src` επίσης διαγράφονται και η νέα τιμή αποθηκεύεται στον καταχωρητή `r_tmp` (ετικέτα 2b). Ενώνοντας τους καταχωρητές `r_dst` και `r_tmp` σχηματίζεται ένας νέος αριθμός κινητής υποδιαστολής όπου το πρόσημο και ο εκθέτης δεν έχουν αλλάξει (ετικέτα 3).

Η ρουτίνα `set_exp` υλοποιείται με αντίστοιχο τρόπο, αλλά αυτήν τη φορά είναι απαραίτητες δύο ολισθήσεις (ετικέτες 2β και 2γ) για τη μεταφορά του νέου εκθέτη στην σωστή θέση. Η συνένωση του καταχωρητή `r_tmp` με τον `r_dst` σχηματίζει το νέο αριθμό κινητής υποδιαστολής.

Η Εικόνα 5.17 παρουσιάζει την υλοποίηση των ρουτινών `set_sig` και `set_exp` χρησιμοποιώντας το σύνολο εντολών του επεξεργαστή MIPS. Τα σχόλια σε κάθε γραμμή δηλώνουν τα αντίστοιχα βήματα στις Εικόνες 5.15 και 5.16.

Όταν το σύνολο εντολών ενός επεξεργαστή προσφέρει εντολές επεξεργασίας σε επίπεδο ψηφίων, τότε η υλοποίηση των ρουτινών `set_sig` και `set_exp` απλοποιείται σε μεγάλο βαθμό. Για παράδειγμα, το σύνολο εντολών του επεξεργαστή PowerPC⁹ παρέχει εντολές που συνδυάζουν την επεξεργασία σε επίπεδο ψηφίων σε συνδυασμό με λογικές μάσκες (*logic masks*) [124]. Μεταξύ των εντολών αυτών, η εντολή `rlwimi`, *rotate left immediate then mask insert*, είναι ιδανική για την υλοποίηση των `set_sig` και `set_exp`. Η σύνταξη της εντολής είναι `rlwimi RA, RS, SH, BM`. Η εντολή `rlwimi` περιστρέφει αριστερά (*rotate left*) τα περιεχόμενα του ακέραιου καταχωρητή `RS` τόσες θέσεις όπως αυτές ορίζονται από την παράμετρο `SH`. Έπειτα, αποθηκεύει τα δεδομένα στον καταχωρητή `RA` υπό τον έλεγχο της μάσκας ψηφίων (32-bit mask) η οποία ορίζεται

⁹Αν και γενικά οι μικροεπεξεργαστές PowerPC ενσωματώνουν μονάδες κινητής υποδιαστολής διπλής ακρίβειας, είναι συνηθισμένο σε ενσωματωμένες εκδόσεις τους να ενσωματώνουν μονάδες κινητής υποδιαστολής απλής ακρίβειας.



Εικόνα 5.16: Η λειτουργία της ρουτίνας set_exp στους καταχωρητές r_dst και r_src .

στο πεδίο BM. Αν το ψηφίο i της μάσκας έχει την τιμή 1, τότε το αντίστοιχο ψηφίο i της περιστραμμένης τιμής αντικαθιστά το ψηφίο i στον καταχωρητή RA. Αντίθετα, όταν το ψηφίο i στη μάσκα έχει την τιμή 0, το αντίστοιχο ψηφίο i του καταχωρητή RA μένει αμετάβλητο. Επομένως, οι ρουτίνες set_sig και set_exp εύκολα υλοποιούνται με την εντολή $rlwimi$ στους καταχωρητές r_dst και r_src και τις αντίστοιχες τιμές στις παραμέτρους SH και BM, όπως δείχνει η Εικόνα 5.18.

Οι ρουτίνες set_sig και set_exp είναι απαραίτητες για την ανάπτυξη λογισμικού αυτοδοκιμής βασισμένο σε ψευδοτυχαία ή σε προ-υπολογισμένα διανύσματα. Η δοκιμή των μονάδων κινητής υποδιαστολής με ψευδοτυχαία διανύσματα δοκιμής μπορεί να γίνει με δύο τρόπους. Είτε παράγοντας διανύσματα δοκιμής για όλο το μήκος του καταχωρητή κινητής υποδιαστολής, είτε παράγοντας διανύσματα για τα επιμέρους πεδία του καταχωρητή κινητής υποδιαστολής, δηλαδή για τον εκθέτη και το σημανόμενο. Ένα πρότυπο ανάπτυξης λογισμικού βασισμένο σε ψευδοτυχαία διανύσματα για τη δεύτερη περίπτωση παρουσιάζεται στην Εικόνα 5.19.

Οι ρουτίνες $PseudoExpGen$ και $PseudoSigGen$ παράγουν ψευδοτυχαίες τιμές για τον εκθέτη και το σημανόμενο, αντίστοιχα, οι οποίες αποθηκεύονται στον καταχωρητή R1. Στον καταχωρητή R2 διατηρείται ο αριθμός κινητής υποδιαστολής. Οι ρουτίνες set_exp και set_sig αντικαθιστούν το αντίστοιχο πεδίο του καταχωρητή R2 με την τιμή του καταχωρητή R1. Η εντολή FP_OP εφαρμόζει το διάνυσμα δοκιμής και παράγει την απόκριση στον καταχωρητή κινητής υποδιαστολής FP_R3 . Στο τέλος ο καταχωρητής FP_R3 μεταφέρεται στον καταχωρητή R3 και μέσω της ρουτίνας $CompactionRoutine$ συμπύσσεται ώστε να σχηματιστεί η τελική υπογραφή. Οι εντολές στην Εικόνα 5.19 εκτελούνται ως μέρος ενός βρόχου που εφαρμόζει τον

```

1 set_sig:
2   # r_clear-sig: 0xff80000
3   and r_dst, r_dst, r_clear-sig   #(1a, 1b)
4   # r_clear-sign-exp: 0x007fffff
5   and r_tmp, r_src, r_clear-sign-exp #(2a, 2b)
6   or r_dst, r_dst, r_tmp          #(3)
7
8 set_exp:
9   # r_clear-exp: 0x807fffff
10  and r_dst, r_dst, r_clear-exp   #(1a, 1b)
11  sll r_tmp, r_src, 24            #(2a, 2b)
12  srl r_tmp, r_tmp, 1             #(2b, 2c)
13  or r_dst, r_dst, r_tmp          #(3)

```

Εικόνα 5.17: Η υλοποίηση των ρουτινών set significand και set exponent για αναπαράσταση απλής ακρίβειας στον επεξεργαστή MIPS.

```

1 set_sig:
2   rliwimi r_dst, r_src, 0, 0x007FFFFF
3
4 set_exp:
5   rlwimi r_dst, r_src, 23, 0x7F800000

```

Εικόνα 5.18: Η υλοποίηση των ρουτινών set significand και set exponent για αναπαράσταση απλής ακρίβειας στον επεξεργαστή PowerPC.

επιθυμητό αριθμό διανυσμάτων δοκιμής στη μονάδα κινητής υποδιαστολής.

Μια εναλλακτική προσέγγιση του προτύπου ανάπτυξης λογισμικού αυτοδοκιμής της Εικόνας 5.19, είναι η προ-ανάθεση επιθυμητών τιμών στον εκθέτη ή το σημαινόμενο, ώστε να εφαρμόζονται ψευδοτυχαία διανύσματα μόνο στο σημαινόμενο ή τον εκθέτη και αντίστροφα.

Το λογισμικό αυτοδοκιμής για μονάδες κινητής υποδιαστολής μπορεί να βασιστεί σε προ-υπολογισμένα διανύσματα εκμεταλλευόμενο την συμμετρική δομή των υπομονάδων που αποτελούν τη μονάδα κινητής υποδιαστολής. Οι μονάδες κινητής υποδιαστολής αποτελούνται από αθροιστές, αφαιρέτες, αυξητές, μειωτές, ολισθητές για τους οποίους είναι δυνατή η εξαγωγή διανυσμάτων δοκιμής βασισμένη στην συμμετρική τους δομή. Τα προ-υπολογισμένα διανύσματα αναπτύσσονται με βάση τις συνθήκες δοκιμαστικότητας σύμφωνα με τις εργασίες [26], [28]. Η συμμετρία των διανυσμάτων δοκιμής επιτρέπει τη δημιουργία τους χρησιμοποιώντας αριθμητικές εντολές και εντολές ολίσθησης. Η ιδιότητα αυτή είναι ιδιαίτερα σημαντική όταν αναπτύσσουμε κλιμακούμενο (scalable) λογισμικό αυτοδοκιμής για διαφορετικές ακρίβειας αναπαραστάσεις. Με την άφιξη της τετραπλής ακρίβειας αναπαραστάσης (128-bit) στο

```

1 PseudoSelfTest:
2   PseudoExpGen(R1)
3   set_exp(R2, R1)
4   PseudoSigGen(R1)
5   set_sig(R2, R1)
6   FP_R1 := MOVE_INT(R2)
7
8   PseudoExpGen(R1)
9   set_exp(R2, R1)
10  PseudoSigGen(R1)
11  set_sig(R2, R1)
12  FP_R2 := MOVE_INT(R2)
13
14  FP_R3 := FP_R1 FP_OP FP_R2
15  R3 := MOVE_FP(FP_R3)
16  CompactionRoutine(R3)
17  # Repeat until all patterns are applied
    
```

Εικόνα 5.19: Ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής απλής ακρίβειας, βασισμένο σε ψευδοτυχαία διανύσματα.

	sigA	sigB	opA	opB
S1	(0)10...00	(0)01...00	$\frac{23}{52}$ ↔	$\frac{23}{52}$ ↔
S2	00...00	(0)01...00	Σταθερό	$\frac{23}{52}$ ↔
S3	11...11	(1)00...00	Σταθερό	$\frac{23}{52}$ ↔
S4	(0)01...11	(0)10...00	$\frac{23}{52}$ ↔	$\frac{23}{52}$ ↔

Πίνακας 5.4: Σύνολο διανυσμάτων δοκιμής για τον αφαιρέτη στο σύντομο μονοπάτι του αθροιστή/αφαιρέτη κινητής υποδιαστολής με αρχιτεκτονική υλοποίησης δύο μονοπατιών.

αναθεωρημένο πρότυπο αναπαράστασης αριθμών κινητής υποδιαστολής IEEE 754, η προσαρμοσμένη αυτοδοκιμή για οποιαδήποτε ακρίβεια αποκτά μεγαλύτερη σημασία.

Ένα τμήμα ενός προσαρμοσμένου συνόλου δοκιμής παρουσιάζεται στον Πίνακα 5.4. Το σύνολο δοκιμής είναι μέρος ενός μεγαλύτερου συνόλου δοκιμής του ακέραίου αφαιρέτη στον αθροιστή/αφαιρέτη κινητής υποδιαστολής [26] (βλ. Κεφ. 4), που ακολουθεί την υλοποίηση των δύο μονοπατιών [21] (βλ. Κεφ. 3).

Το σύνολο των διανυσμάτων δοκιμής αποτελείται από τέσσερις ακολουθίες : S1, S2, S3, και S4. Οι ακολουθίες αυτές προκύπτουν έπειτα από διαδοχικές δεξιές ολισθήσεις των αρχικών τιμών των σημειόμενων. Οι στίλες sigA και sigB παρουσιάζουν τις αρχικές τιμές των σημειόμενων των αριθμών κινητής υποδιαστολής A και B για κάθε ακολουθία διανυσμάτων δοκιμής, ενώ οι στίλες opA και opB επισημαίνουν αν το σημειόμενο θα χρειαστεί δεξιά

ολίσθηση. Για παράδειγμα, στην ακολουθία S2 του Πίνακα 5.4, το σημαίνόμενο sigA παραμένει σταθερό, ενώ το σημαίνόμενο sigB θα ολισθαίνει δεξιά κατά μια θέση. Η κάθε ολίσθηση παράγει ένα διάνυσμα δοκιμής. Το ψηφίο στην παρένθεση που βρίσκεται στη στήλη του σημαίνόμενου υποδηλώνει την τιμή του νέου δυαδικού ψηφίου που θα αντικαταστήσει το αριστερότερο ψηφίο του σημαίνόμενου. Οι αριθμοί επάνω από το σύμβολο \mapsto δηλώνουν το πλήθος των ολισθήσεων που χρειάζονται για τη δημιουργία της ακολουθίας διανυσμάτων δοκιμής. Για παράδειγμα, στην ακολουθία διανυσμάτων δοκιμής S2 το ψηφίο στην παρένθεση είναι 0, επομένως θα πραγματοποιηθεί λογική δεξιά ολίσθηση και το αριστερότερο ψηφίο θα λάβει την τιμή 0. Συνολικά θα πραγματοποιηθούν 23 δεξιές ολισθήσεις για απλή ακρίβεια αναπαράστασης και 52 ολισθήσεις για διπλή ακρίβεια αναπαράστασης με το σημαίνόμενο sigB να λαμβάνει τις τιμές (010...00), (001...00), ..., (00...01), (00...00). Στην ακολουθία S3, το ψηφίο στην παρένθεση είναι 1. Επομένως θα πραγματοποιηθεί αριθμητική ολίσθηση και το αριστερότερο ψηφίο θα λάβει την τιμή 1. Συνολικά, το σημαίνόμενο sigB θα λάβει τις παρακάτω τιμές: (000...00), (100...00), (110...00), ..., (111...10), (111...11). Στις ακολουθίες S1 και S4 ολισθαίνουν και τα δύο σημαίνόμενα.

Υποθέτοντας μια μονάδα κινητής υποδιαστολής τετραπλής ακρίβειας όπου από τα 128-bit, τα 112-bit χρησιμοποιούνται για το σημαίνόμενο θα έπρεπε να ολισθήσουμε το σημαίνόμενο του αριθμού κινητής υποδιαστολής B, sigB, 112 φορές ώστε να σχηματιστεί η ακολουθία S2.

Η Εικόνα 5.20 παρουσιάζει το λογισμικό αυτοδοκιμής για την εφαρμογή της ακολουθίας διανυσμάτων δοκιμής S2. Αρχικά, ο καταχωρητής κινητής υποδιαστολής FP_R1 λαμβάνει την αρχική τιμή. Ο καταχωρητής R2 διατηρεί το δεύτερο αριθμό κινητής υποδιαστολής και ο καταχωρητής R3 το σημαίνόμενο το οποίο θα αντικαταστήσει το σημαίνόμενο στον καταχωρητή R2. Οι καταχωρητές κινητής υποδιαστολής FP_R1, FP_R2 έχουν τον ίδιο εκθέτη (0x3F) αλλά διαφορετικά πρόσημα ώστε η πράξη κινητής υποδιαστολής να είναι η αφαίρεση. Έπειτα, μια δεξιά αριθμητική ολίσθηση του καταχωρητή R3 δημιουργεί ένα νέο σημαίνόμενο το οποίο γίνεται παράμετρος στην ρουτίνα set_sig. Κατ' αυτόν τον τρόπο δημιουργείται ένας νέος αριθμός κινητής υποδιαστολής.

Το λογισμικό αυτοδοκιμής της Εικόνας 5.20, χρησιμοποιείται και για την εφαρμογή των διανυσμάτων δοκιμής της ακολουθίας S3, όπως φαίνεται στην Εικόνα 5.21. Οι μόνες αλλαγές που είναι απαραίτητες αφορούν τα σημαίνόμενα.

5.6.2 Ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες διπλής ακρίβειας

Η ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής διπλής ακρίβειας μπορεί να είναι αρκετά διαφορετική από αυτήν για απλή ακρίβεια. Η αιτία βρίσκεται στην αρχιτεκτονική του μικροεπεξεργαστή. Για παράδειγμα, ένας μικροεπεξεργαστής MIPS των 32-bit, χρησιμοποιεί δύο καταχωρητές κινητής υποδιαστολής απλής ακρίβειας για να σχηματίσει έναν καταχωρητή διπλής ακρίβειας. Για αυτό το λόγο, η ανάπτυξη λογισμικού αυτοδοκιμής με

```

1 applyS2:
2   # sign: 0
3   # exponent: 0x7f
4   # significand : 00...00
5   R1 := Assemble(0x3f800000)
6   FP_R1 := MOVE_INT(R1)
7
8   # sign: 1
9   # exponent: 0x07f
10  # significand: 0100...00
11  R2 := Assemble(0xbfa00000)
12
13  R3 := Assemble(0x00200000)
14 ShiftLoop:
15  FP_R2 := MOVE_INT(R2)
16
17  FP_R3 := FP_R1 FP_ADD_OP FP_R2
18  R4 := MOVE_FP(FP_R3)
19  CompactionRoutine(R4)
20  R3 := ShiftRightArith(R3, 1)
21  R2 := set_sig(R3)
22  # Repeat ShiftLoop for 23 times

```

Εικόνα 5.20: Λογισμικό αυτοδοκιμής που εφαρμόζει τα προσαρμόσιμα διανύσματα δοκιμής της ακολουθίας S2.

ψευδοτυχαίες ακολουθίες ή με προ-υπολογισμένα διανύσματα ή διανύσματα που παράγονται από αυτόματα εργαλεία απαιτεί τη διαχείριση δύο καταχωρητών 32-bit για το σχηματισμό ενός νοπού καταχωρητή διπλής ακρίβειας.

Από την άλλη πλευρά, το λογισμικό αυτοδοκιμής που εφαρμόζει διανύσματα δοκιμής από αυτόματα εργαλεία δεν αντιμετωπίζει τέτοια προβλήματα στις μονάδες κινητής υποδιαστολής διπλής ακρίβειας. Σχεδόν όλοι οι μικροεπεξεργαστές παρέχουν εντολές για τη μεταφορά αριθμών διπλής ακρίβειας από τη μνήμη σε ένα ζεύγος καταχωρητών 32-bit ή σε ένα καταχωρητή 64-bit. Επομένως, το λογισμικό αυτοδοκιμής που χρησιμοποιεί βρόχους για την εφαρμογή διανυσμάτων δοκιμής (από αυτόματα εργαλεία) που βρίσκονται στη μνήμη έχει παρόμοια μορφή με το λογισμικό αυτοδοκιμής στην Εικόνα 5.13.

Αντίθετα, το λογισμικό αυτοδοκιμής που εφαρμόζει διανύσματα δοκιμής από αυτόματα εργαλεία, σχηματίζοντάς τα σε ακέραιους καταχωρητές εξαρτάται από την αρχιτεκτονική του μικροεπεξεργαστή. Όταν χρησιμοποιούμε δύο καταχωρητές των 32-bit το λογισμικό αυτοδοκιμής της Εικόνας 5.14 μετατρέπεται σε αυτό της Εικόνας 5.22. Σε αρχιτεκτονικές μικροεπεξεργαστών των 64-bit, όπως στους MIPS-64 και Alpha, ο σχηματισμός των διανυσμάτων δοκιμής

```

1 applyS3:
2   # sign: 0
3   # exponent: 0x7f
4   # significand : 11...11
5   R1 := Assemble(0x3f8ffff)
6   FP_R1 := MOVE_INT(R1)
7
8   # sign: 1
9   # exponent: 0x07f
10  # significand: 0000...00
11  R2 := Assemble(0xbf800000)
12
13  R3 := Assemble(0xffe00000)
14 ShiftLoop:
15  FP_R2 := MOVE_INT(R2)
16
17  FP_R3 := FP_R1 FP_ADD_OP FP_R2
18  R4 := MOVE_FP(FP_R3)
19  CompactionRoutine(R4)
20  R3 := ShiftRightArith(R3, 1)
21  R2 := set_sig(R3)
22  # Repeat ShiftLoop for 23 times

```

Εικόνα 5.21: Λογισμικό αυτοδοκιμής που εφαρμόζει τα προσαρμόσιμα διανύσματα δοκιμής της ακολουθίας S3.

χρησιμοποιώντας την άμεση διευθυνσιοδότηση των εντολών, γίνεται δυσκολότερος διότι το άμεσο πεδίο (immediate) παραμένει 16-bit ενώ ο καταχωρητής είναι 64-bit. Η Εικόνα 5.23 παρουσιάζει την ακολουθία εντολών για την ανάθεση μιας τιμής 64-bit σε έναν καταχωρητή. Μια τέτοια ακολουθία είναι απαραίτητη για κάθε τελεστέο. Επιπλέον στους επεξεργαστές Alpha το άμεσο πεδίο έχει μέγεθος 8-bit μόνο. Σε αυτή την περίπτωση η ακολουθία εντολών της Εικόνας 5.23 διπλασιάζεται, ενώ ο καταχωρητής R1 ολισθαίνει αριστερά κατά 8 θέσεις αντί για 16.

Η εφαρμογή του λογισμικού αυτοδοκιμής που βασίζεται σε ψευδοτυχαία ή προ-υπολογισμένα διανύσματα σε μονάδες διπλής ακρίβειας, απαιτεί την αναθεώρηση των ρουτινών `set_sig` και `set_exp`. Στους μικροεπεξεργαστές 32-bit, οι ρουτίνες `set_sig` και `set_exp` πρέπει να τροποποιηθούν ώστε να χρησιμοποιούν δύο καταχωρητές σαν ένα νοητό καταχωρητή 64-bit. Το πρώτο όρισμα των ρουτινών `set_sig` και `set_exp` είναι ένα ζεύγος καταχωρητών το οποίο διατηρεί τον αριθμό κινητής υποδιαστολής διπλής ακρίβειας. Ο πρώτος καταχωρητής στο ζεύγος καταχωρητών διατηρεί τα υψηλά 32-bit, ενώ ο δεύτερος τα χαμηλά 32-bit. Το δεύτερο όρισμα της `set_sig` είναι ξανά ένα ζεύγος καταχωρητών το οποίο διατηρεί το σημαντικό με μέγε-

```

1 R1 := Assemble(Pattern1_OpA_High32)
2 FP_R1 := MOVE_INT(R1)
3 R1 := Assemble(Pattern1_OpA_Low32)
4 FP_R2 := MOVE_INT(R1)
5
6 R1 := Assemble(Pattern1_OpB_High32)
7 FP_R3 := MOVE_INT(R1)
8 R1 := Assemble(Pattern1_OpB_Low32)
9 FP_R4 := MOVE_INT(R1)
10
11 <FP_R5,FP_R6> :=
12   <FP_R1, FP_R2> FP_OP <FP_R3, FP_R4>
13
14 R2 := MOVE_FP(FP_R5)
15 CompactionRoutine(R2)
16 R2 := MOVE_FP(FP_R6)
17 CompactionRoutine(R2)

```

Εικόνα 5.22: Σχηματίζοντας διανύσματα δοκιμής εξαγόμενα από αυτόματα εργαλεία (ATPG) χρησιμοποιώντας τις εντολές άμεσης διευθυνσιοδότησης. Τα διανύσματα δοκιμής είναι για μονάδες κινητής υποδιαστολής διπλής ακρίβειας σε ένα μικροεπεξεργαστή 32-bit.

θος 52-bit, ενώ στην ρουτίνα `set_exp` ένας καταχωρητής μεγέθους 32-bit είναι αρκετός για την αποθήκευση του εκθέτη, μεγέθους 11-bit. Η λειτουργία της τροποποιημένης ρουτίνας `set_sig` για διπλή ακρίβεια αναπαράστασης παρουσιάζεται στην Εικόνα 5.24. Η Εικόνα 5.25 παρουσιάζει την υλοποίηση των ρουτινών `set_sig` και `set_exp` σύμφωνα με το σύνολο εντολών του επεξεργαστή MIPS, το οποίο δεν παρέχει εντολές επεξεργασίας σε επίπεδο ψηφίων.

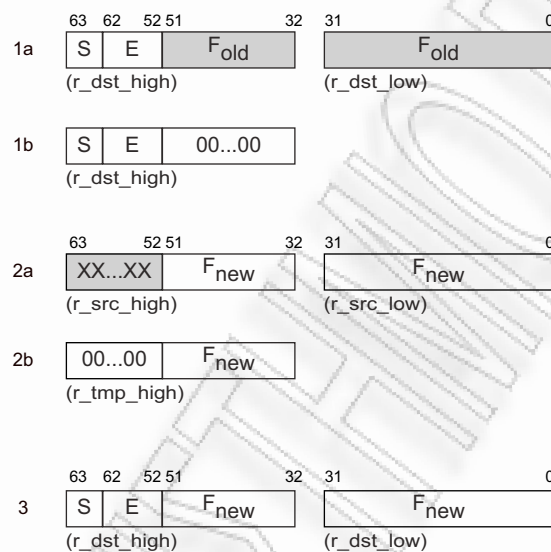
Στην Εικόνα 5.25, το ζεύγος καταχωρητών `r2, r3` διατηρεί τον αριθμό κινητής υποδιαστολής διπλής ακρίβειας και το ζεύγος καταχωρητών `r4, r5` το σημειώμενο. Στην ρουτίνα `set_exp`, ο καταχωρητής `r5` διατηρεί την τιμή του εκθέτη που θα αντικαταστήσει τον εκθέτη στο ζεύγος καταχωρητών `r2, r3`. Οι καταχωρητές `r20, r22` και `r24` διατηρούν τις κατάλληλες λογικές μάσκες (bit masks).

Όταν το σύνολο εντολών του μικροεπεξεργαστή παρέχει εντολές επεξεργασίας σε επίπεδο ψηφίων (bit level manipulation instructions), όπως στην περίπτωση του μικροεπεξεργαστή PowerPC, οι ρουτίνες `set_sig` και `set_exp` γίνονται ξανά πιο συμπαγείς και πιο απλές, όπως φαίνεται στην Εικόνα 5.26. Έτσι πρέπει να τροποποιηθούν μόνο οι λογικές μάσκες (bit masks) και, επιπλέον, χρειάζεται ένας καταχωρητής για το σχηματισμό ενός νοητού καταχωρητή 64-bit.

Μέχρι στιγμής, συζητήσαμε για αρχιτεκτονικές μικροεπεξεργαστών 32-bit με διπλή ακρίβεια αναπαράστασης, όπως οι MIPS και PowerPC. Σε αρχιτεκτονικές μικροεπεξεργαστών 64-bit, όπως οι Alpha και MIPS-64, οι ρουτίνες `set_sig` και `set_exp` για διπλή ακρίβεια αναπαράστασης

-
- 1 OR Immediate R1, R1, Pattern1_Bits_63-48
 - 2 Shift Left R1, R1, 16
 - 3 OR Immediate R1, R1, Pattern1_Bits_47-32
 - 4 Shift Left R1, R1, 16
 - 5 OR Immediate R1, R1, Pattern1_Bits_31-15
 - 6 Shift Left R1, R1, 16
 - 7 OR Immediate R1, R1, Pattern1_Bits_15-0
-

Εικόνα 5.23: Σχηματίζοντας ένα διάνυσμα δοκιμής 64-bit χρησιμοποιώντας εντολές άμεσης διευθυνσιοδότησης όπου το μέγεθος του άμεσου πεδίου (immediate) είναι 16-bit.



Εικόνα 5.24: Η λειτουργία της ρουτίνας `set_sig` στους καταχωρητές `r_dst_high`, `r_dst_low` και `r_src_high`, `r_src_low`.

υλοποιούνται όπως ακριβώς στις αρχιτεκτονικές 32-bit (βλέπε Εικόνα 5.17). Η Εικόνα 5.27 παρουσιάζει τις ρουτίνες `set_sig` και `set_exp` για τον μικροεπεξεργαστή Alpha¹⁰.

Ο Πίνακας 5.5 συνοψίζει το πλήθος των εντολών που απαιτούνται για την υλοποίηση των `set_exp` και `set_sig` σε αρχιτεκτονικές μικροεπεξεργαστών των 32 και των 64-bit για μονάδες κινητής υποδιαστολής απλής και διπλής ακρίβειας αναπαράστασης. Είναι εμφανές πως το σύνολο εντολών του μικροεπεξεργαστή PowerPC επιτρέπει την αποτελεσματικότερη υλοποίηση των ρουτινών `set_sig` και `set_exp` για απλή και διπλή ακρίβεια. Συγκρίνοντας τις Εικόνες 5.25, 5.26, και 5.27, συμπεραίνουμε πως το σύνολο εντολών του μικροεπεξεργαστή παίζει σημαντικό ρόλο στην αποτελεσματική υλοποίηση των ρουτινών `set_sig` και `set_exp`.

Το λογισμικό αυτοδοκιμής βασισμένο σε ψευδοτυχαία ή προ-υπολογισμένα διανύσματα για

¹⁰Η μορφή των εντολών διευθυνσιοδότησης καταχωρητή στο σύνολο εντολών του Alpha είναι `instr Rs, Rt, Rd` για εντολές άμεσης διευθυνσιοδότησης είναι `instr Rt, 8-bit Imm, Rd`.

```

1 set_sig:
2 # <r2,r3> holds the dp fp number
3 # <r4,r5> holds the significand value(52 bits)
4
5 # Clear significand part in pair <r2,r3>
6 # r20: 0xfff00000
7 and r2, r2, r20 # high order bits (1a, 1b)
8
9 # Clear sign & exponent part in <r4>
10 # Store it in r9
11 # r22: 0x000fffff
12 and r9, r4, r22 # high order bits (2a, 2b)
13
14 # Merge pair <r2,r3> with <r9,r5>
15 or r2, r2, r9 # high order bits (3)
16 or r3, r3, r5 # low order bits (3)
17
18 set_exp:
19 # <r2,r3> hold the dp fp number
20 # r5 holds the exponent value(11 bits)
21
22 # Clear sign and significand part in
23 # register r2
24 # r24: 0x800fffff
25 and r2, r2, r24
26
27 # shift operations to place exponent in
28 # proper position
29 slli r11, r5, 21
30 srli r11, r11, 1
31
32 # Merge r2 and r11
33 or r2, r2, r11

```

Εικόνα 5.25: MIPS 32-bit—Οι ρουτίνες set significand και set exponent για μονάδες κινητής υποδιαστολής διπλής ακρίβειας.

μονάδες κινητής υποδιαστολής διπλής ακρίβειας αναπαράστασης μοιάζει με το αντίστοιχο λογισμικό αυτοδοκιμής για απλή ακρίβεια αναπαράστασης (Εικόνες 5.19 και 5.20). Σε αρχιτεκτονικές μικροεπεξεργαστών 64-bit τα πρότυπα ανάπτυξης του λογισμικού αυτοδοκιμής είναι άμεσα εφαρμόσιμα αφού αναπροσαρμοστούν οι λογικές μάσκες. Αντίθετα, σε αρχιτεκτονικές μικροεπεξεργαστών 32-bit τα πρότυπα ανάπτυξης λογισμικού αυτοδοκιμής χρειάζονται

```

1 # <r2,r3> holds the double precision FP number
2 # <r4,r5> holds the significand value(52 bits)
3 set_sig:
4   add r3, r0, r5 # low 32-bits
5   rlwimi r2, r4, 0, 0x000FFFFF # high 32-bits
6
7 # r5: holds the exponent value (11 bits)
8 set_exp:
9   rlwimi r2, r5, 20, 0x7FF00000

```

Εικόνα 5.26: PowerPC 32-bit—Οι ρουτίνες set significand και set exponent για μονάδες κινητής υποδιαστολής διπλής ακρίβειας.

		set_sig	set_exp
Απλή Ακρίβεια	MIPS	3	4
	PowerPC	1	1
Διπλή Ακρίβεια	MIPS	4	4
	PowerPC	2	1
	Alpha	3	4

Πίνακας 5.5: Πλήθος εντολών που απαιτούνται για την υλοποίηση των ρουτινών set_sig και set_exp στους επεξεργαστές MIPS, PowerPC και Alpha.

τροποποιήσεις. Στο λογισμικό αυτοδοκιμής βασισμένο σε ψευδοτυχαία διανύσματα, η ρουτίνα PseudoSigGen θα πρέπει να τροποποιηθεί ώστε να δημιουργεί ψευδοτυχαίους αριθμούς των 52-bit και να τους αποθηκεύει σε ένα ζεύγος καταχωρητών. Επιπλέον οι ψευδοεντολές MOVE_INT και MOVE_FP πρέπει να τροποποιηθούν ώστε να ενεργούν σε ζεύγη καταχωρητών. Αντίστοιχες τροποποιήσεις είναι απαραίτητες για το λογισμικό αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα δοκιμής στην Εικόνα 5.20.

5.6.3 Πειραματικά αποτελέσματα

Σε αυτήν την ενότητα αξιολογούμε τις διάφορες τεχνικές ανάπτυξης λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής ως προς τα ακόλουθα κριτήρια:

- ποσοστό κάλυψης ελαττωμάτων,
- ίχνος μνήμης,
- χρόνος εκτέλεσης, και
- κατανάλωση ενέργειας,

```

1 # r23 holds the double precision FP number
2 # r3  holds the significand value (52 bits)
3 # r14 used as auxiliary register
4 set_sig:
5   # clear the significand part in r23
6   # r24 : 0xfff0000000000000
7   and r23, r24, r23
8   # clear sign and exponent parts in r3
9   # bic r3, r24, r14 is equivalent to
10  # and r3, r24, NOT(14)
11  bic r3, r24, r14
12  # merge r23 with r14
13  or r23, r14, r23
14
15 # r2 holds the exponent value (11 bits)
16 set_exp:
17  # clear the exponent part in r23
18  # r25 : 0x800fffffffffffff
19  and r23, r25, r23
20  # shift left 53 positions register r2
21  # then shift right one position
22  sll r2, 53, r14
23  srl r14, 1, r14
24  # merge registers
25  or r14, r23, r23

```

Εικόνα 5.27: Alpha 64-bit—Οι ρουτίνες set significand και set exponent για μονάδες κινητής υποδιαστολής διπλής ακρίβειας.

τα οποία είναι τα κριτήρια για την επιτυχημένη ετερόχρονη περιοδική αυτοδοκιμή κατά τη λειτουργία.

Οι συννηθέστερες πράξεις κινητής υποδιαστολής είναι η πράξη της πρόσθεσης (αφαίρεσης) και του πολλαπλασιασμού. Επίσης, όπως έχει γίνει αντιληπτό από το Κεφάλαιο 3, τα κυκλώματα που υλοποιούν τις παραπάνω πράξεις είναι και αυτά με τη μεγαλύτερη πολυπλοκότητα. Αναπτύξαμε μονάδες πρόσθεσης και πολλαπλασιασμού κινητής υποδιαστολής, τόσο απλής όσο και διπλής ακρίβειας αναπαράστασης. Η ανάπτυξη των μονάδων κινητής υποδιαστολής έγινε χρησιμοποιώντας την ελεύθερα διαθέσιμη βιβλιοθήκη FPLibrary [125]. Ο Πίνακας 5.6 παρουσιάζει τα αποτελέσματα της σύνθεσης (logic synthesis) των κυκλωμάτων σε βιβλιοθήκη τεχνολογίας 0.18 μm , το μέγεθος του κυκλώματος (σε πύλες NAND2x1) και την καθυστέρηση διάδοσης (σε ns). Επιπλέον παρουσιάζεται και ο αριθμός των ελαττωμάτων προσκόλλησης (stuck-at).

	Ακρίβεια	Nand2X1 gate equivalents	Delay (ns)	S@ Faults
FP	Απλή	3235	7.64	12208
Adder	Διπλή	6869	9.94	26350
FP Mult	Απλή	5009	14.65	11815
	Διπλή	23178	33.43	50559

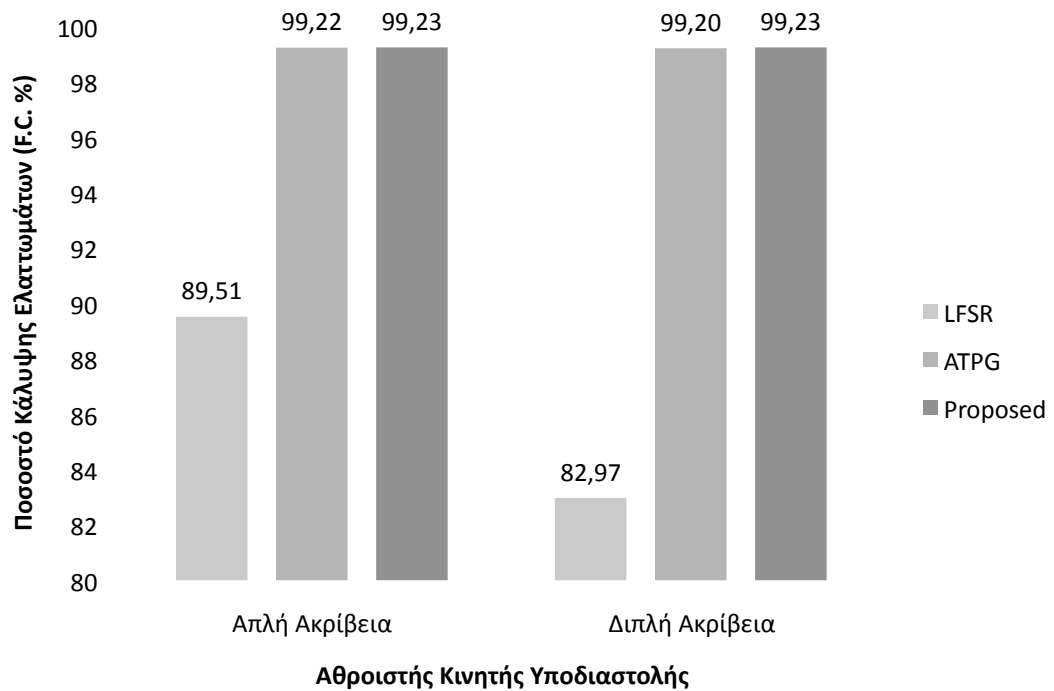
Πίνακας 5.6: Στατιστικά σύνθεσης για αθροιστές και πολλαπλασιαστές κινητής υποδιαστολής απλής και διπλής ακρίβειας.

Έχουμε αναπτύξει λογισμικό αυτοδοκιμής για μονάδες κινητής υποδιαστολής σύμφωνα με τις αρχές ανάπτυξης λογισμικού της Ενότητας 5.3, χρησιμοποιώντας διανύσματα δοκιμής από αυτόματα εργαλεία, γεννιότερες ψευδοτυχαίων διανυσμάτων και προ-υπολογισμένα διανύσματα δοκιμής. Τα προ-υπολογισμένα διανύσματα αναπτύχθηκαν σύμφωνα με την εργασία [26] (βλ. Κεφ. 4). Η ανάπτυξη των διανυσμάτων δοκιμής έγινε τόσο για μονάδες κινητής υποδιαστολής απλής όσο και διπλής ακρίβειας. Επιπλέον αναπτύξαμε διάφορες παραλλαγές του λογισμικού αυτοδοκιμής χρησιμοποιώντας το σύνολο εντολών των επεξεργαστών MIPS 32-bit και Alpha 64-bit. Ειδικά, για τον επεξεργαστή MIPS θεωρήσαμε δύο πιθανές εκδοχές του. Στην πρώτη εκδοχή του ενσωματώνει μονάδα κινητής υποδιαστολής απλής ακρίβειας, ενώ στη δεύτερη εκδοχή του διπλής ακρίβειας. Με αυτόν τον τρόπο έχουμε καλύψει όλους τους συνδυασμούς μεγέθους αρχιτεκτονικής (32/64-bit) και ακρίβειας (απλή/διπλή). Το λογισμικό αυτοδοκιμής που αναπτύξαμε σύμφωνα με κάθε τεχνική, το αξιολογήσαμε ως προς το ποσοστό κάλυψης ελαττωμάτων, το μέγεθος μνήμης, το χρόνο εκτέλεσης και, τέλος, ως προς την κατανάλωση ενέργειας.

Ο χρόνος εκτέλεσης του λογισμικού αυτοδοκιμής καθώς και η κατανάλωση ενέργειάς τους υπολογίστηκαν με το εργαλείο Wattch [126], το οποίο βασίζεται στο σύνολο εργαλείων SimpleScalar¹¹ [127]. Το εργαλείο Wattch μοντελοποιεί έναν υπερβαθμωτό (superscalar) επεξεργαστή εκτέλεσης εκτός σειράς (out-of-order execution). Κατά την αξιολόγηση του λογισμικού αυτοδοκιμής υποθέσαμε έναν επεξεργαστή με μία μονάδα κινητής υποδιαστολής—με έναν αθροιστή/αφαιρέτη και έναν πολλαπλασιαστή.

Οι Εικόνες 5.28 και 5.29 παρουσιάζουν τα ποσοστά κάλυψης ελαττωμάτων για αθροιστές και πολλαπλασιαστές κινητής υποδιαστολής απλής και διπλής ακρίβειας. Ο Πίνακας 5.7 παρουσιάζει το πλήθος των διανυσμάτων δοκιμής που εφαρμόζονται στους αθροιστές και στους πολλαπλασιαστές απλής και διπλής ακρίβειας σύμφωνα με την εκάστοτε τεχνική ανάπτυξης ενσωματωμένου λογισμικού (βασισμένο σε διανύσματα δοκιμής από αυτόματα (ATPG), σε

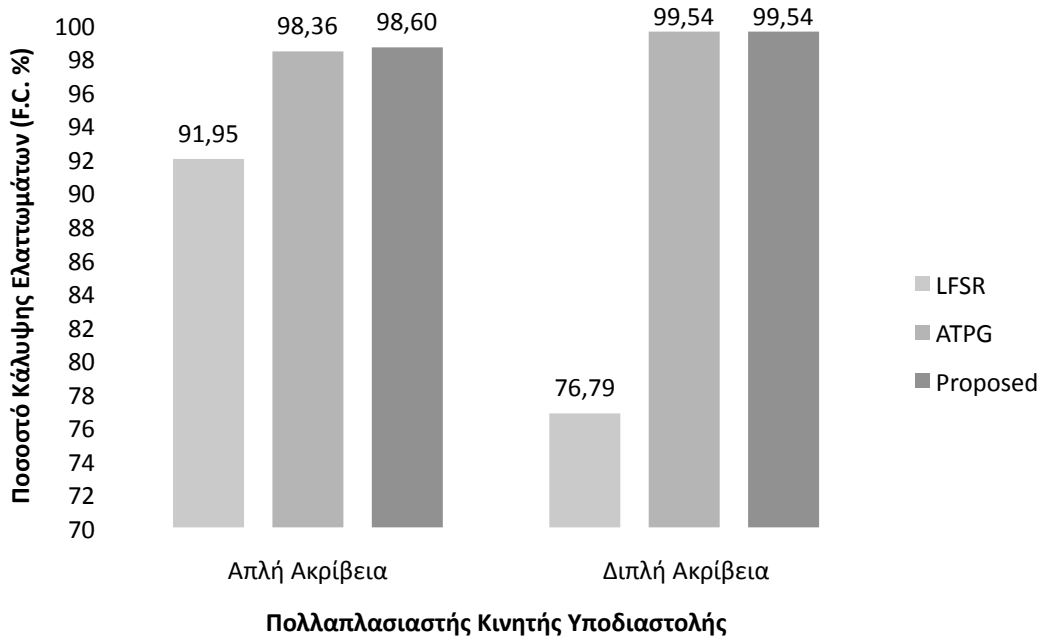
¹¹Το σύνολο εντολών Portable ISA (PISA) που ορίζεται στο εργαλείο SimpleScalar και Wattch είναι αντίστοιχο με αυτό του επεξεργαστή MIPS. Η διαφορά τους είναι πως το σύνολο εντολών PISA χρησιμοποιεί 64-bit για την κωδικοποίηση των εντολών. Τα εργαλεία SimpleScalar και Wattch επιτρέπουν την προσομοίωση του επεξεργαστή θεωρώντας την κωδικοποίηση του συνόλου εντολών με 32-bit, χρησιμοποιώντας την παράμετρο -cache:icompress.



Εικόνα 5.28: Συγκριτικά αποτελέσματα ποσοστού κάλυψης ελαττωμάτων για αθροιστές-αφαιρέτες κινητής υποδιαστολής απλής και διπλής ακρίβειας.

ψευδοτυχαία διανύσματα (LFSR) και σε προ-υπολογισμένα διανύσματα από συνθήκες δοκιμαστικότητας (Proposed)). Το σύνολο των διανυσμάτων δοκιμής που αποτελείται από προ-υπολογισμένα διανύσματα δοκιμής επεκτάθηκε με ένα μικρό αριθμό διανυσμάτων δοκιμής από αυτόματο εργαλείο εξαγωγής για την ανίχνευση ελαττωμάτων στο τμήμα της λογικής ελέγχου (control logic part) της μονάδας κινητής υποδιαστολής. Σε ορισμένες περιπτώσεις, ο συνδυασμός των προ-υπολογισμένων διανυσμάτων δοκιμής με διανύσματα δοκιμής από αυτόματο εργαλείο παρέχει κάπως υψηλότερα ποσοστά κάλυψης ελαττωμάτων. Σε κάθε περίπτωση το ποσοστό κάλυψης ελαττωμάτων του λογισμικού αυτοδοκιμής με ψευδοτυχαία διανύσματα δοκιμής είναι σαφώς χαμηλότερο σε σχέση με τις υπόλοιπες τεχνικές ανάπτυξης λογισμικού αυτοδοκιμής. Επομένως, συμπεραίνουμε πως το λογισμικό αυτοδοκιμής με διανύσματα από αυτόματα εργαλεία ή προ-υπολογισμένα διανύσματα παρέχει υψηλά ποσοστά κάλυψης ελαττωμάτων, ενώ το λογισμικό αυτοδοκιμής με ψευδοτυχαία διανύσματα δεν είναι κατάλληλο για τη δοκιμή των μονάδων κινητής υποδιαστολής στο πεδίο λειτουργίας.

Συνεχίζουμε την ανάλυση συγκρίνοντας τα διαφορετικά λογισμικά αυτοδοκιμής ως προς το ίχνος μνήμης. Ο Πίνακας 5.8 παρουσιάζει τις απαιτήσεις των λογισμικών αυτοδοκιμής σε byte για τον επεξεργαστή MIPS αρχιτεκτονικής 32-bit και τον επεξεργαστή Alpha αρχιτεκτονικής 64-bit. Οι αριθμοί στις παρενθέσεις, στον Πίνακα 5.8, διπλώνουν την κατανομή των απαιτή-



Εικόνα 5.29: Συγκριτικά αποτελέσματα ποσοστού κάλυψης ελαττωμάτων για πολλαπλασιαστές κινητής υποδιαστολής απλής και διπλής ακρίβειας.

	FP Adder		FP Multiplier	
	Απλή Ακρ.	Διπλή Ακρ.	Απλή Ακρ.	Διπλή Ακρ.
ATPG	361	682	200	379
Proposed (Pre-computed+ATPG)	521+65	937+213	432+69	474+158
LFSR		100000		

Πίνακας 5.7: Το πλήθος των διανυσμάτων δοκιμής.

σεων σε μνήμη για εντολές (I), και δεδομένα (D) των λογισμικών αυτοδοκιμής. Σε όλες τις περιπτώσεις το λογισμικό αυτοδοκιμής που σχηματίζεται από τη σύνθεση των διανυσμάτων δοκιμής χρησιμοποιώντας την άμεση διευθυνσιοδότηση των εντολών (ATPG immediate-ATPG Imm) έχει το μεγαλύτερο ίχνος μνήμης. Αυτή η παρατήρηση είναι αναμενόμενη καθώς η μορφή του λογισμικού αυτοδοκιμής αποτελεί την ξετυλιγμένη έκδοση του λογισμικού αυτοδοκιμής με βρόχο (ATPG Loop). Επιπλέον, στους μικροεπεξεργαστές αρχιτεκτονικής 64-bit οι απαιτήσεις σε ίχνος μνήμης είναι μεγαλύτερες.

Οι απαιτήσεις σε μνήμη του λογισμικού αυτοδοκιμής που προκύπτει από την εφαρμογή των προ-υπολογισμένων διανυσμάτων δοκιμής είναι μικρότερες σε σχέση με το λογισμικό αυτοδοκιμής που εφαρμόζει διανύσματα δοκιμής από αυτόματα εργαλεία εκτός από την περίπτωση του πολλαπλασιαστή απλής ακρίβειας. Αυτό συμβαίνει διότι ο πολλαπλασιαστής

	FP Adder		
	Απλή ακρίβεια	Διπλή ακρίβεια	
	MIPS	MIPS	Alpha
ATPG Loop	2944 (I:48,D:2896)	10960 (I:48,D:10912)	10960 (I:48,D:10912)
ATPG Imm	10120 (I:10120,D:0)	18784 (I:18784,D:0)	30064 (I:30064,D:0)
Proposed	2336 (I:1104,D:1232)	5299 (I:1459,D:3840)	4512 (I:608,D:3904)

	FP Multiplier		
	Απλή ακρίβεια	Διπλή ακρίβεια	
	MIPS	MIPS	Alpha
ATPG Loop	1648 (I:48,D:1600)	6112 (I:48,D:6064)	6160 (I:96,D:6064)
ATPG Imm	5608 (I:5608,D:0)	19720 (I:19720,D:0)	16736 (I:16736,D:0)
Proposed	2080 (I:928,D:1152)	2288 (I:1120,D:1168)	3856 (I:1216,D:2640)

Πίνακας 5.8: Μέγεθος λογισμικού αυτοδοκιμής (byte).

κινητής υποδιαστολής χρειάζεται τα λιγότερα διανύσματα δοκιμής. Ειδικότερα στις μονάδες κινητής υποδιαστολής διπλής ακρίβειας οι διαφορές στις απαιτήσεις σε μνήμη κυμαίνονται μεταξύ 37%, στην περίπτωση του πολλαπλασιαστή όταν το λογισμικό αυτοδοκιμής υλοποιείται με το σύνολο εντολών του επεξεργαστή Alpha, και 62%, όταν το λογισμικό αυτοδοκιμής υλοποιείται στο σύνολο εντολών του MIPS.

Για την αξιολόγηση του χρόνου εκτέλεσης και της κατανάλωσης του λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής υποθέτουμε τρία διαφορετικά σενάρια ρυθμίσεων της μνήμης.

Θεωρούμε ένα σύστημα μνήμης με διαιρεμένη κρυφή μνήμη (split cache) πρώτου επιπέδου (level-1-L1) και ενοποιημένη κρυφή μνήμη δεύτερου επιπέδου (level-2-L2) με τα χαρακτηριστικά του Πίνακα 5.9. Σε όλες τις περιπτώσεις χρησιμοποιήθηκαν οι ρυθμίσεις του εργαλείου Watch όπως αυτές παρουσιάζονται στο κάτω μέρος του Πίνακα 5.9. Το τελευταίο σύνολο ρυθμίσεων, το Config.C, το οποίο ορίζει το μέγεθος της κρυφής μνήμης εντολών και δεδομένων μόλις 1Kb, είναι χρήσιμο ώστε να μελετήσουμε τη συμπεριφορά του λογισμικού αυτοδοκιμής όταν αυτό δε χωράει στην κρυφή μνήμη. Ένα τέτοιο σενάριο είναι πιθανό σε ένα ενσωματωμένο

	Config.A	Config.B	Config.C
Κρυφή μνήμη εντολών I-L1	16Kb—1-way Block Size: 16 bytes, Χρόνος Ευστοχίας: 1 cycle	8Kb—1-way	1Kb—1-way
Κρυφή μνήμη δεδομένων D-L1	16Kb—2-way Block Size: 16 bytes, Χρόνος Ευστοχίας: 1 cycle	8Kb—2-way	1Kb—2-way
Κοινή κρυφή μνήμη Unified-L2	128Kb—4-way Block Size: 32 bytes, Χρόνος Ευστοχίας: 6 cycles	64Kb—4-way	16Kb—4-way

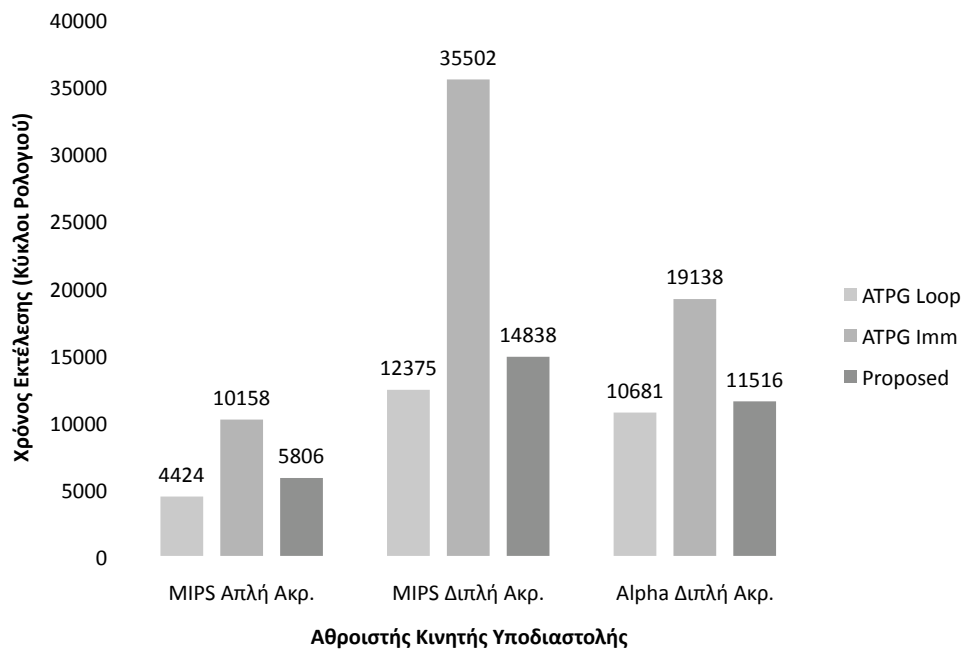
Χρόνος προσπέλασης κύριας μνήμης: πρώτη προσπέλαση: 18 cycles, υπόλοιπες προσπελάσεις: 2 cycles, Εύρος μνήμης: 8 bytes, Πλήθος Integer ALUs: 4, Πλήθος Integer Multipliers: 1, Πλήθος FP ALU: 1, Πλήθος FP Multiplier: 1

Πίνακας 5.9: Διευθετήσεις συστήματος κρυφής μνήμης.

σύστημα χαμηλού κόστους.

Οι Εικόνες 5.30 και 5.31 παρουσιάζουν το χρόνο εκτέλεσης του λογισμικού αυτοδοκιμής για μικροεπεξεργαστές MIPS, και Alpha, βασισμένο σε διανύσματα δοκιμής από αυτόματα εργαλεία και του λογισμικού αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα για αθροιστές/αφαιρέτες και πολλαπλασιαστές απλής και διπλής ακρίβειας.

Οι χρόνοι εκτέλεσης των διαφόρων τύπων λογισμικού αυτοδοκιμής δε μεταβάλλονται στις τρεις διαφορετικές διευθετήσεις της κρυφής μνήμης και γι' αυτό στις Εικόνες 5.30 και 5.31 δε γίνεται αναφορά σε αυτές. Ακόμα και όταν το λογισμικό αυτοδοκιμής δε χωρά στην κρυφή μνήμη L1 ο χρόνος εκτέλεσης δεν αλλάζει. Αυτό συμβαίνει διότι ο υπερβαθμωτός επεξεργαστής έχει διαθέσιμες εντολές προς εκτέλεση ακόμα και όταν συμβεί μια αστοχία στην κρυφή μνήμη (cache miss). Επιπλέον, κανένα λογισμικό αυτοδοκιμής δεν αποθηκεύει δεδομένα στη μνήμη, εξαιτίας της σύμπτυξης των αποκρίσεων των διανυσμάτων δοκιμής (test response compaction). Άρα, κανένα τμήμα της κρυφής μνήμης δεδομένων δε θα σημειωθεί ως ακάθαυτο (dirty) ώστε να χρειαστεί η μεταφορά του στη μνήμη. Κατά συνέπεια, ακόμα και αν ο ρυθμός αστοχίας στην κρυφή μνήμη δεδομένων αυξάνεται, δε σπαταλώνται κύκλοι για τη μεταφορά δεδομένων στην κύρια μνήμη. Το λογισμικό αυτοδοκιμής που εφαρμόζει διανύσματα δοκιμής από αυτόματα εργαλεία χρησιμοποιώντας βρόχους και το λογισμικό αυτοδοκιμής που εφαρμόζει προ-υπολογισμένα διανύσματα έχουν μικρό χρόνο εκτέλεσης. Το λογισμικό αυτοδοκιμής που εφαρμόζει διανύσματα δοκιμής από αυτόματα εργαλεία χρησιμοποιώντας βρόχους έχει μικρότερο χρόνο εκτέλεσης σε σχέση με αυτό που εφαρμόζει προ-υπολογισμένα διανύσματα. Η διαφορά αυτή κυμαίνεται μεταξύ 7% στην περίπτωση του αθροιστή/αφαιρέτη διπλής ακρίβειας όταν το λογισμικό αυτοδοκιμής είναι υλοποιημένο στο σύνολο εντολών του Alpha, και 55% στην περίπτωση του πολλαπλασιαστή απλής ακρίβειας όταν το λογισμικό αυτοδοκιμής είναι υλοποιημένο στο σύνολο εντολών του MIPS. Τόσο το λογισμικό αυτοδοκιμής που προκύπτει από προ-υπολογισμένα διανύσματα όσο και αυτό που προκύπτει από διανύσματα που



Εικόνα 5.30: Συγκριτικά αποτελέσματα χρόνου εκτέλεσης (κύκλοι) για αθροιστές-αφαιρέτες κινητής υποδιαστολής απλής και διπλής ακρίβειας.

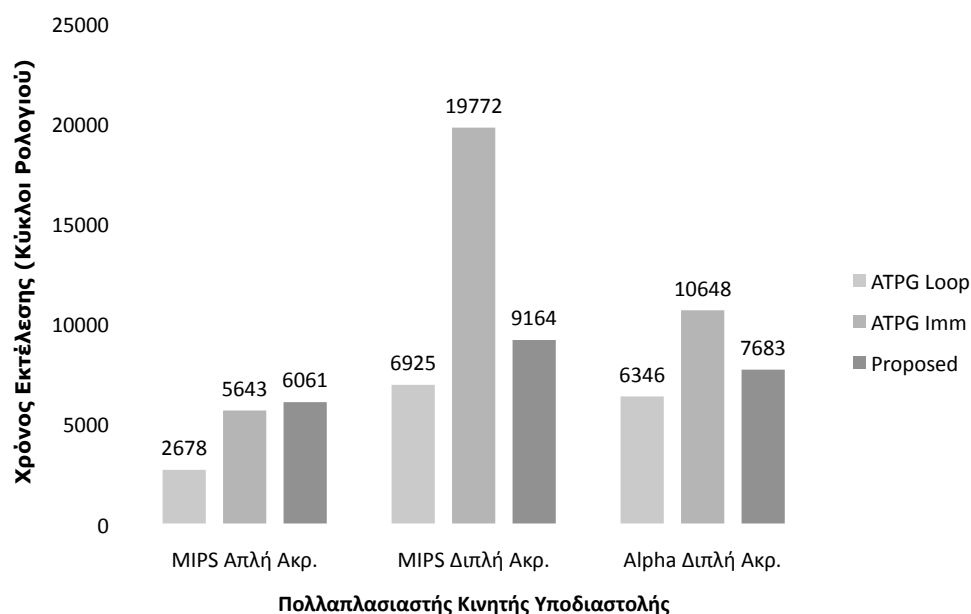
εξάγονται από αυτόματα εργαλεία έχουν μικρότερο χρόνο εκτέλεσης όταν είναι υλοποιημένα στο σύνολο εντολών του επεξεργαστή Alpha απ’ ότι στον MIPS.

Οι Εικόνες 5.32 και 5.33 παρουσιάζουν συγκριτικά αποτελέσματα σχετικά με την κατανάλωση ενέργειας των διαφόρων τύπων λογισμικού αυτοδοκιμής, για αθροιστές-αφαιρέτες και πολλαπλασιαστές κινητής υποδιαστολής απλής και διπλής ακρίβειας.

Το εργαλείο Watch υποστηρίζει διαφορετικά μοντέλα κατανάλωσης ενέργειας για διαφορετικούς τύπους αρχιτεκτονικής ρολογιού (clock gating styles). Σε όλα τα πειράματα, έχουμε επιλέξει εκείνο το μοντέλο κατανάλωσης ενέργειας σύμφωνα με το οποίο μια μονάδα ακόμα και όταν δε χρησιμοποιείται καταναλώνει το 10% της μέγιστης ενέργειας που θα καταναλώσει αν χρησιμοποιούνταν¹². Η παραπάνω επιλογή είναι αντιπροσωπευτική της κατανάλωσης ενέργειας των σύγχρονων μικροεπεξεργαστών.

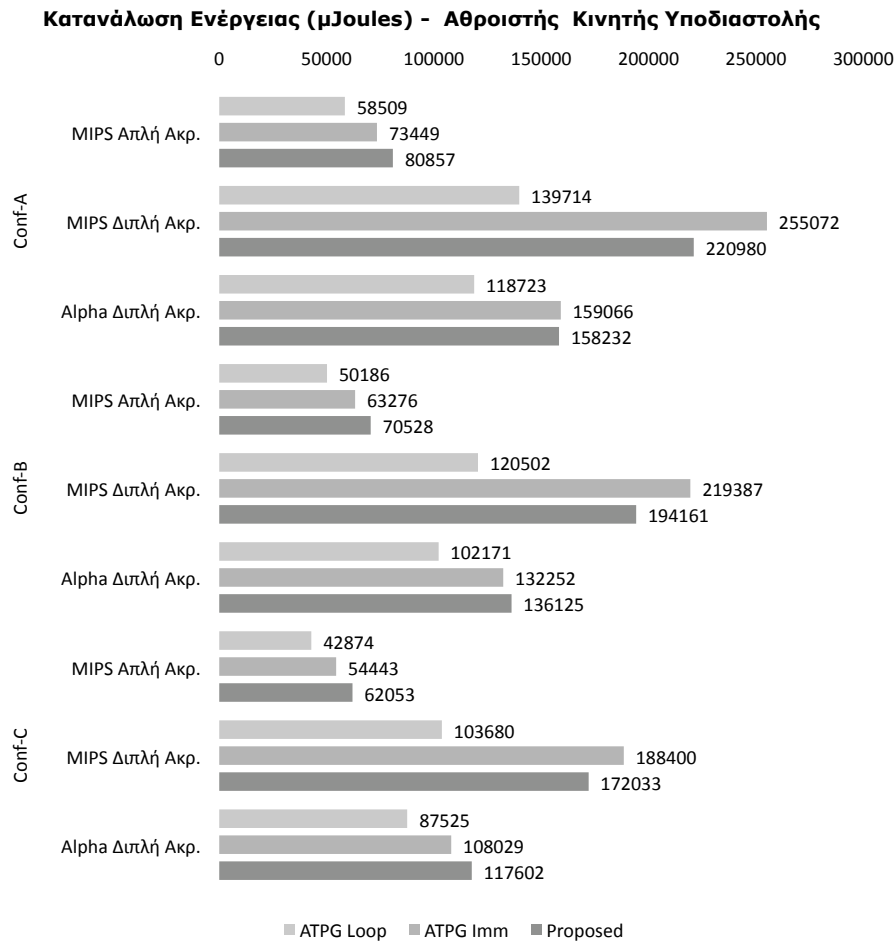
Μελετώντας τις Εικόνες 5.32 και 5.33 παρατηρούμε πως σε όλες τις περιπτώσεις το λογισμικό αυτοδοκιμής που παράγεται από διανύσματα δοκιμής που εξάγονται από αυτόματα εργαλεία και εφαρμόζονται με βρόχους καταναλώνει λιγότερη ενέργεια σε σχέση με το λογισμικό που παράγεται από προ-υπολογισμένα διανύσματα δοκιμής. Στην περίπτωση του αθροιστή/αφαιρέτη κινητής υποδιαστολής διπλής ακρίβειας, η δαπανώμενη ενέργεια σχεδόν

¹²Η συγκεκριμένη επιλογή αρχιτεκτονικής ρολογιού (clock gating mode 3) παρουσιάζεται στην έξοδο αποτελεσμάτων του εργαλείου Watch ως cc3.



Εικόνα 5.31: Συγκριτικά αποτελέσματα χρόνου εκτέλεσης (κύκλοι) για πολλαπλασιαστές κινητής υποδιαστολής απλής και διπλής ακρίβειας.

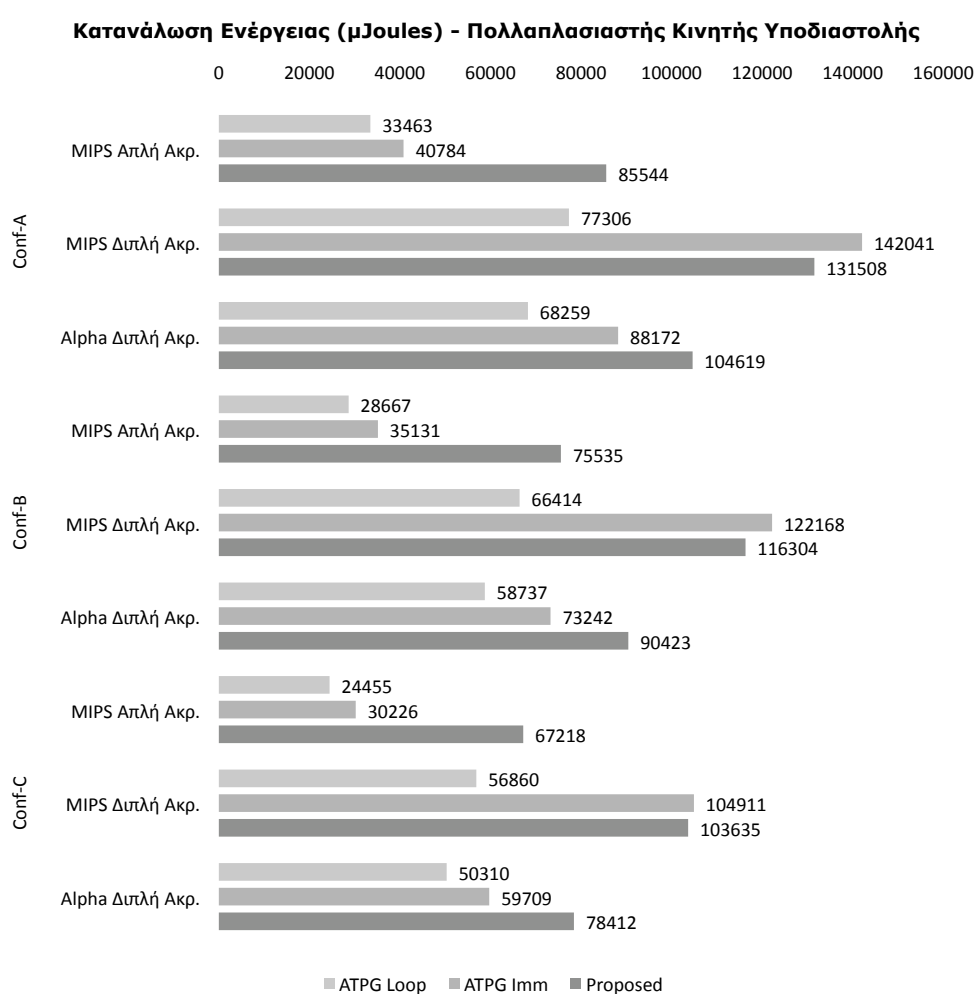
διπλασιάζεται στο λογισμικό αυτοδοκιμής με προ-υπολογισμένα διανύσματα. Αντίστοιχη είναι και η παρατήρηση για το λογισμικό αυτοδοκιμής του πολλαπλασιαστή διπλής ακρίβειας. Στην περίπτωση του αθροιστή/αφαιρέτη κινητής υποδιαστολής το λογισμικό αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα δοκιμής, χρειάζεται να εκτελέσει μια σειρά εντολών για την ανάθεση τιμών στο σημασιόμενο και τον εκθέτη (βλ. Εικόνα 5.25), καθώς και για τη μεταφορά του ακέραου καταχωρητή στον καταχωρητή κινητής υποδιαστολής. Επιπροσθέτως, ο επεξεργαστής MIPS έχει αρχιτεκτονική 32-bit, άρα οι αριθμητικές και λογικές πράξεις σε τιμές 64-bit αυξάνουν την πολυπλοκότητα, επομένως και την κατανάλωση ενέργειας. Παρατηρήστε πως όταν το λογισμικό αυτοδοκιμής είναι υλοποιημένο στο σύνολο εντολών του Alpha δεν παρατηρούμε πολύ μεγάλες διαφορές στην κατανάλωση ενέργειας μεταξύ των λογισμικών αυτοδοκιμής που προκύπτουν από τις δύο τεχνικές ανάπτυξης. Η παρατήρηση αυτή επιβεβαιώνει πως το σύνολο εντολών του μικροεπεξεργαστή καθώς και η αρχιτεκτονική του έχουν σημαντικό ρόλο στην ανάπτυξη του λογισμικού αυτοδοκιμής και της ενέργειας που καταναλώνει. Οι διαφορές στην κατανάλωση ενέργειας μεταξύ του λογισμικού που εφαρμόζει με βρόχους διανύσματα δοκιμής από αυτόματα εργαλεία και του λογισμικού που εφαρμόζει προ-υπολογισμένα είναι μεγαλύτερες στην περίπτωση του πολλαπλασιαστή κινητής υποδιαστολής, διότι η αρχιτεκτονική του πολλαπλασιαστή είναι απλούστερη σε σχέση με του αθροιστή διπλού μονοπατιού. Μια ενδιαφέρουσα παρατήρηση προκύπτει από την σύγκριση



Εικόνα 5.32: Συγκριτικά αποτελέσματα κατανάλωσης ενέργειας για αθροιστές-αφαιρέτες κινητής υποδιαστολής απλής και διπλής ακρίβειας.

της κατανάλωσης ενέργειας όταν μειώνεται το μέγεθος της κρυφής μνήμης. Καθώς το μέγεθος της κρυφής μνήμης μειώνεται, μειώνεται και η κατανάλωση ενέργειας παρόλο που αυξάνονται οι ρυθμοί αστοχίας στην κύρια μνήμη. Αυτό συμβαίνει διότι μικρότερες κρυφές μνήμες καταναλώνουν λιγότερη ενέργεια.

Πριν καταλήξουμε στην καταλληλότερη μεθοδολογία για ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής εξηγούμε γιατί δεν περιλαμβάνεται στα πειραματικά αποτελέσματα το λογισμικό αυτοδοκιμής το οποίο θα ήταν ένας συνδυασμός της εφαρμογής ψευδοτυχαίων διανυσμάτων και διανυσμάτων δοκιμής από αυτόματα εργαλεία. Όπως φαίνεται στις Εικόνες 5.28 και 5.29 το λογισμικό αυτοδοκιμής βασισμένο σε ψευδοτυχαία διανύσματα δεν επιτυγχάνει υψηλά ποσοστά κάλυψης ελαττωμάτων. Ας υποθέσουμε ότι το λογισμικό



Εικόνα 5.33: Συγκριτικά αποτελέσματα κατανάλωσης ενέργειας πολλαπλασιαστή κινητής υποδιαστολής απλής και διπλής ακρίβειας.

αυτοδοκιμής γεννά 10.000 ψευδοτυχαία διανύσματα και πως επιτυγχάνει το ίδιο ποσοστό κάλυψης ελαττωμάτων με τις άλλες τεχνικές. Σύμφωνα με το σύνολο παραμέτρων Config.A του συστήματος μνήμης (Πίνακας 5.9), το υποθετικό αυτό πρόγραμμα θα χρειαζόταν 472.480 κύκλους ρολογιού για την εκτέλεσή του ενώ θα κατανάλωνε συνολικά 4.188,705 μ Joules. Οι αριθμοί αυτοί είναι σαφώς μεγαλύτεροι σε σχέση με εκείνους των Εικόνων 5.32 και 5.33. Έπειτα από τη σύγκριση αυτή συμπεραίνουμε πως η αυτοδοκιμή των μονάδων κινητής υποδιαστολής βασισμένη σε ψευδοτυχαία διανύσματα δοκιμής δεν είναι καθόλου αποδοτική για ετερόχρονη δοκιμή κατά τη λειτουργία.

Στόχος	Τεχνική ανάπτυξης ενσωματωμένου λογισμικού που επικρατεί
Μικρό μέγεθος μνήμης	Λογισμικό αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα δοκιμής.
Χαμηλή κατανάλωση ενέργειας	Λογισμικό αυτοδοκιμής βασισμένο σε διανύσματα δοκιμής από αυτόματα εργαλεία—εφαρμογή με βρόχο.
Σύντομος χρόνος εκτέλεσης	Λογισμικό αυτοδοκιμής βασισμένο σε διανύσματα δοκιμής από αυτόματα εργαλεία—εφαρμογή με βρόχο.
Υψηλό ποσοστό κάλυψης ελαττωμάτων	Λογισμικό αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα δοκιμής ή σε διανύσματα δοκιμής από αυτόματα εργαλεία—εφαρμογή με βρόχο.

Πίνακας 5.10: Συμπεράσματα από τα πειραματικά αποτελέσματα.

Μια παρόμοια παρατήρηση μπορούμε να κάνουμε για το λογισμικό αυτοδοκιμής που εφαρμόζει διανύσματα δοκιμής από αυτόματα εργαλεία χρησιμοποιώντας την άμεση διευθυνσιοδότηση των εντολών. Όπως φαίνεται στις Εικόνες 5.30, 5.31 και 5.32, 5.33, τα εν λόγω λογισμικά αυτοδοκιμής έχουν πολύ μεγάλους χρόνους εκτέλεσης και καταναλώνουν αρκετά υψηλά ποσά ενέργειας. Λαμβάνοντας υπόψη πως το λογισμικό αυτοδοκιμής που εφαρμόζει διανύσματα δοκιμής χρησιμοποιώντας την άμεση διευθυνσιοδότηση των εντολών έχει το μεγαλύτερο ίχνος μνήμης, συμπεραίνουμε πως η συγκεκριμένη τεχνική ανάπτυξης λογισμικού αυτοδοκιμής δεν είναι αποδοτική για ετερόχρονη δοκιμή κατά τη λειτουργία.

Ολοκληρώνοντας τη μελέτη των πειραματικών αποτελεσμάτων έχουμε να επιλέξουμε μεταξύ της μεθοδολογίας ανάπτυξης λογισμικού αυτοδοκιμής που εφαρμόζει προ-υπολογισμένα διανύσματα και εκείνης που εφαρμόζει διανύσματα δοκιμής από αυτόματα εργαλεία χρησιμοποιώντας βρόχους. Το λογισμικό αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα δοκιμής επιτυγχάνει τη μείωση του ίχνους μνήμης αλλά ταυτόχρονα η εκτέλεση του δαπανά περισσότερη ενέργεια. Αντίθετα, το λογισμικό αυτοδοκιμής βασισμένο σε διανύσματα δοκιμής εξαγόμενα από αυτόματα εργαλεία επιτυγχάνει χαμηλότερη κατανάλωση ενέργειας αλλά το τίμημα είναι οι αυξημένες απαιτήσεις σε μνήμη. Διαπιστώνουμε πως η βελτιστοποίηση του λογισμικού αυτοδοκιμής ως προς μια εκ των παραμέτρων που χαρακτηρίζουν την περιοδική ετερόχρονη αυτοδοκιμή επιτυχημένη μπορεί να επηρεάσει αρνητικά μια άλλη παράμετρο. Ως προς το χρόνο εκτέλεσης δεν παρατηρούνται σημαντικές διαφορές ώστε να είναι προτιμότεα μια εκ των δύο τεχνικών ανάπτυξης ενσωματωμένου λογισμικού αυτοδοκιμής. Ο Πίνακας 5.10 συνοψίζει τα αποτελέσματα της αξιολόγησης των πειραματικών δεδομένων.

Όταν το μέγεθος μνήμης αποτελεί το κυριότερο κριτήριο για την αποτελεσματική περιοδική ετερόχρονη αυτοδοκιμή τότε το λογισμικό αυτοδοκιμής βασισμένο σε προ-υπολογισμένα διανύσματα αποτελεί την ιδανικότερη επιλογή. Αντίθετα, όταν η κατανάλωση ενέργειας έχει τον πρωτεύοντα ρόλο, το λογισμικό αυτοδοκιμής βασισμένο σε διανύσματα δοκιμής εξαγόμενα από αυτόματα εργαλεία είναι καταλληλότερο.

5.7 Ανακεφαλαίωση

Στο Κεφάλαιο 5,

- παρουσιάστηκε η αυτοδοκιμή των μικροεπεξεργαστών κατά τη λειτουργία με ενσωματωμένο λογισμικό,
- παρουσιάστηκαν τα επιθυμητά χαρακτηριστικά του λογισμικού αυτοδοκιμής,
- παρουσιάστηκαν οι τεχνικές ανάπτυξης λογισμικού αυτοδοκιμής μικροεπεξεργαστών που διαθέτουν μονάδες κινητής υποδιαστολής απλής ή διπλής ακρίβειας,
- παρουσιάστηκε η αξιολόγηση των τεχνικών ανάπτυξης λογισμικού αυτοδοκιμής ως προς το ίχνος μνήμης, το χρόνο εκτέλεσης και την κατανάλωση ενέργειας καθώς και τα πλεονεκτήματα και μειονεκτήματα της κάθε τεχνικής.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντική Έρευνα

Η παρούσα διδακτορική διατριβή ασχολήθηκε με το πρόβλημα της δοκιμής των μικροεπεξεργαστών που βρίσκονται σε ένα ευρύτερο ενσωματωμένο σύστημα και διαθέτουν μονάδες κινητής υποδιαστολής για την επίτευξη αριθμητικής υψηλών επιδόσεων.

Στο πρώτο μέρος της διατριβής, παρουσιάστηκε η μελέτη δοκιμαστικότητας των μονάδων κινητής υποδιαστολής η οποία οδήγησε στην ανάπτυξη των συνθηκών δοκιμαστικότητας. Οι συνθήκες δοκιμαστικότητας αποτελούνται από τις συνθήκες ελεγχιμότητας και παρατηρησιμότητας και εκφράζουν υπό ποιές συνθήκες τα διανύσματα δοκιμής μπορούν να ανιχνεύσουν ελαττώματα στη μονάδα κινητής υποδιαστολής. Στο δεύτερο μέρος της διατριβής, παρουσιάστηκαν τα κριτήρια για την ανάπτυξη λογισμικού δοκιμής ώστε να είναι κατάλληλο για τη δοκιμή κατά τη λειτουργία. Επίσης, παρουσιάστηκαν οι τεχνικές ανάπτυξης λογισμικού αυτοδοκιμής και δόθηκε ιδιαίτερη έμφαση σε επεξεργαστές που περιέχουν μονάδες κινητής υποδιαστολής.

Τα συμπεράσματα που προκύπτουν από το πρώτο μέρος της διατριβής είναι πως η πολυπλοκότητα των μονάδων κινητής υποδιαστολής επηρεάζει αρνητικά τη δοκιμαστικότητά τους. Η παρατήρηση επιβεβαιώνεται από το πλήθος των διανυσμάτων δοκιμής που χρειάζεται μια μονάδα κινητής υποδιαστολής ώστε να επιτευχθούν υψηλά ποσοστά κάλυψης ελαττωμάτων. Ακόμα και αν χρησιμοποιηθεί ένα αυτόματο εργαλείο εξαγωγής διανυσμάτων δεν είναι βέβαιο ότι θα επιτύχει υψηλά ποσοστά κάλυψης για κάθε μοντέλο ελαττωμάτων. Χαρακτηριστικό παράδειγμα αποτελεί το μοντέλο ελαττώματος καθυστέρησης μετάβασης (transition delay fault model) όπου το αυτόματο εργαλείο εξαγωγής διανυσμάτων επιτυγχάνει χαμηλότερα ποσοστά κάλυψης ελαττωμάτων σε σχέση με το ποσοστό κάλυψης ελαττωμάτων που επιτυγχάνουν τα διανύσματα δοκιμής που έχουν εξαχθεί βάσει των συνθηκών δοκιμαστικότητας.

Ένα άλλο συμπέρασμα που προκύπτει είναι πως τα διανύσματα δοκιμής που παράγονται από ψευδοτυχαίους γεννήτορες διανυσμάτων δε μπορούν να παρέχουν υψηλά ποσοστά κάλυψης ελαττωμάτων για αποτελεσματική δοκιμή. Τα ψευδοτυχαία διανύσματα δοκιμής μπορούν να επιτύχουν υψηλά ποσοστά κάλυψης ελαττωμάτων μόνο αν εφαρμοστούν τεχνικές σχεδίασης για αύξηση της δοκιμαστικότητας. Όμως, η εφαρμογή τέτοιων τεχνικών αυξάνει το μέγεθος του κυκλώματος και μειώνει τη συχνότητα λειτουργίας τους, γεγονός που επιδρά αρνητικά στην απόδοση του μικροεπεξεργαστή. Η εξαγωγή διανυσμάτων δοκιμής βάσει των συνθηκών δοκιμαστικότητας οδηγεί σε διανύσματα δοκιμής με υψηλά ποσοστά κάλυψης ελαττωμάτων τόσο για το μοντέλο ελαττωμάτων προσκόλλησης όσο και για το μοντέλο ελαττωμάτων καθυστέρησης μετάβασης τιμής. Επιπλέον, τα διανύσματα δοκιμής είναι προσαρμόσιμα με βάση την ακρίβεια αναπαράστασης της μονάδας κινητής υποδιαστολής, ενώ η διαδικασία εξαγωγής διανυσμάτων είναι ανεξάρτητη από την εσωτερική υλοποίηση των υπομονάδων που αποτελούν τη μονάδα κινητής υποδιαστολής. Τα παραπάνω χαρακτηριστικά είναι αποτέλεσμα της γενικότητας των συνθηκών δοκιμαστικότητας και δεν παρέχονται ούτε από την εξαγωγή διανυσμάτων δοκιμής από αυτόματα εργαλεία αλλά και ούτε από διανύσματα δοκιμής από ψευδοτυχαίους γεννήτορες διανυσμάτων.

Το κυριότερο συμπέρασμα που προκύπτει από το δεύτερο μέρος της διατριβής είναι πως το λογισμικό αυτοδοκιμής θα πρέπει να διαθέτει και επιπλέον χαρακτηριστικά εκτός από την επίτευξη υψηλού ποσοστού κάλυψης ελαττωμάτων, για να εφαρμοστεί επιτυχώς για αυτοδοκιμή κατά τη λειτουργία. Τα χαρακτηριστικά αυτά αφορούν το μικρό ίχνος μνήμης, το μικρό χρόνο εκτέλεσης και τη χαμηλή κατανάλωση ενέργειας. Από τις τρεις τεχνικές ανάπτυξης λογισμικού αυτοδοκιμής για ακέραιες αριθμητικές μονάδες, την ανάπτυξη με βάση διανύσματα που έχουν εξαχθεί από αυτόματα εργαλεία, την ανάπτυξη με ψευδοτυχαία διανύσματα και την ανάπτυξη με βάση προ-υπολογισμένα διανύσματα, τα πειράματα έδειξαν πως η ανάπτυξη λογισμικού αυτοδοκιμής με βάση προ-υπολογισμένα διανύσματα παρέχει όχι μόνο μικρό ίχνος μνήμης αλλά και μικρό χρόνο εκτέλεσης.

Ένα άλλο συμπέρασμα που προκύπτει από το δεύτερο μέρος της διατριβής είναι πως η ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής είναι αρκετά διαφορετική από αυτή για ακέραιες μονάδες. Εξάιρεση αποτελεί μόνο η ανάπτυξη λογισμικού με βάση διανύσματα που έχουν εξαχθεί από αυτόματα εργαλεία, και μόνο όταν αυτά εφαρμόζονται μέσω βρόχων. Η ανάπτυξη λογισμικού αυτοδοκιμής με βάση ψευδοτυχαία ή προ-υπολογισμένα διανύσματα δοκιμής για μονάδες κινητής υποδιαστολής απαιτεί εντολές οι οποίες θα θέτουν τιμές στο τμήμα του εκθέτη και του σημαίνονενου ενός καταχωρητή κινητής υποδιαστολής. Καθώς κανένα σύνολο εντολών δεν παρέχει τέτοιες δυνατότητες, αναπτύχθηκαν ψευδοεντολές (`set_sig` και `set_exp`) οι οποίες βασίζονται στη δυαδική επεξεργασία (*bit level manipulation*) και οι οποίες επιτρέπουν την απ' ευθείας ανάθεση τιμών στο πεδίο του εκθέτη και του σημαίνονενου. Μια σημαντική παρατήρηση που προέκυψε κατά την υλοποίηση των `set_sig` και `set_exp` αφορά στις δυνατότητες που παρέχουν τα σύνολα εντολών των μικροεπεξεργα-

στών. Για παράδειγμα, το σύνολο εντολών του μικροεπεξεργαστή PowerPC παρέχει εντολές για δυαδική επεξεργασία καταχωρητών καθιστώντας την υλοποίηση των `set_sig` και `set_exp` απλούστερη και ταχύτερη. Αντίθετα, το σύνολο εντολών του μικροεπεξεργαστή MIPS δεν παρέχει τέτοιες εντολές, με αποτέλεσμα η υλοποίηση των `set_sig` και `set_exp` να είναι πιο πολύπλοκη. Το συμπέρασμα που προκύπτει είναι πως η ανάπτυξη του λογισμικού αυτοδοκιμής με βάση ψευδοτυχαία ή προ-υπολογισμένα διανύσματα εξαρτάται άμεσα από τις δυνατότητες του συνόλου εντολών για επεξεργασία σε επίπεδο δυαδικών ψηφίων.

Μια άλλη παρατήρηση που προκύπτει από το δεύτερο μέρος της διατριβής αφορά στην ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής διπλής ακρίβειας (64-bit) που βρίσκονται σε μικροεπεξεργαστές 32-bit οι οποίοι δε διαθέτουν καταχωρητές 64-bit. Σε αυτή την περίπτωση δύο καταχωρητές των 32-bit συνδυάζονται για να σχηματίσουν ένα νοητό καταχωρητή των 64-bit. Και σε αυτήν την περίπτωση οι δυνατότητες του συνόλου εντολών παίζουν σημαντικό ρόλο.

Το συμπέρασμα που προκύπτει από την σύγκριση των τεχνικών ανάπτυξης λογισμικού αυτοδοκιμής είναι πως τόσο η ανάπτυξη λογισμικού με βάση διανύσματα δοκιμής από αυτόματα εργαλεία όσο και η ανάπτυξη λογισμικού αυτοδοκιμής με βάση προ-υπολογισμένα διανύσματα παράγουν λογισμικό αυτοδοκιμής το οποίο επιτυγχάνει πολύ υψηλά ποσοστά κάλυψης ελαττωμάτων σε αντίθεση με το λογισμικό αυτοδοκιμής που παράγεται με βάση ψευδοτυχαία διανύσματα δοκιμής. Επιπλέον, το λογισμικό αυτοδοκιμής που παράγεται με βάση προ-υπολογισμένα διανύσματα δοκιμής έχει το μικρότερο ίχνος μνήμης, κάτι που είναι ιδιαίτερα σημαντικό για ενσωματωμένα συστήματα με μικρή διαθέσιμη μνήμη. Η παρατήρηση αυτή ισχύει τόσο για τις ακέραιες αριθμητικές μονάδες όσο και για τις μονάδες κινητής υποδιαστολής. Όσον αφορά το χρόνο εκτέλεσης του λογισμικού αυτοδοκιμής, το λογισμικό με βάση προ-υπολογισμένα διανύσματα δοκιμής έχει μικρότερο χρόνο εκτέλεσης σε σχέση με το λογισμικό αυτοδοκιμής με βάση διανύσματα δοκιμής από αυτόματα εργαλεία στην περίπτωση των ακέραιων αριθμητικών μονάδων. Αντίθετα, στην περίπτωση των μονάδων κινητής υποδιαστολής, το λογισμικό αυτοδοκιμής με βάση διανύσματα από αυτόματα εργαλεία έχει μικρότερο χρόνο εκτέλεσης. Αυτό οφείλεται στην απουσία εντολών (από το σύνολο εντολών) που επιδρούν στα επιμέρους πεδία ενός καταχωρητή κινητής υποδιαστολής (εκθέτης και σημαίνόμενο). Στην απουσία αυτή αποδίδεται και η παρατήρηση πως το λογισμικό αυτοδοκιμής με βάση προ-υπολογισμένα διανύσματα δοκιμής έχει μεγαλύτερη κατανάλωση ενέργειας, σε σχέση με το λογισμικό αυτοδοκιμής που παράγεται με βάση διανύσματα δοκιμής από αυτόματα εργαλεία.

Συνοψίζοντας, το συμπέρασμα που προκύπτει είναι πως η επιλογή της τεχνικής ανάπτυξης λογισμικού για αυτοδοκιμή κατά τη λειτουργία εξαρτάται από τα επιμέρους χαρακτηριστικά του ενσωματωμένου συστήματος. Η ανάπτυξη λογισμικού αυτοδοκιμής που να συνδυάζει ταυτόχρονα μικρό ίχνος μνήμης, μικρό χρόνο εκτέλεσης και χαμηλή κατανάλωση ενέργειας δεν είναι πάντοτε εφικτή. Ιδιαίτερα στην περίπτωση πολύπλοκων μονάδων όπως οι μονάδες κινητής

υποδιαστολής, η επιλογή της τεχνικής ανάπτυξης λογισμικού αυτοδοκιμής θα πρέπει να γίνεται ανάλογα με το κριτήριο (ίχνος μνήμης, χρόνος εκτέλεσης, ή χαμηλή κατανάλωση ενέργειας) που έχει τη μεγαλύτερη σημασία σε ένα ενσωματωμένο σύστημα.

Όσον αφορά τη μελλοντική έρευνα που προκύπτει από το πρώτο μέρος της διδακτορικής διατριβής, αυτή αφορά την επέκταση των συνθηκών δοκιμαστικότητας ώστε να καλύψουν πολυπλοκότερα μοντέλα ελαττωμάτων, όπως το μοντέλο ελαττώματος καθυστέρησης μονοπατιού. Επιπλέον οι συνθήκες δοκιμαστικότητας μπορούν να χρησιμοποιηθούν για το σχεδιασμό μονάδων κινητής υποδιαστολής ώστε να είναι δυνατή η ανίχνευση ελαττωμάτων ταυτόχρονα με τη λειτουργία (concurrent on-line testing). Οι αθροιστές, αφαιρετές κινητής υποδιαστολής που είναι υλοποιημένοι σύμφωνα με την αρχιτεκτονική δύο μονοπατιών αποτελούν ιδανικό πεδίο όπου οι συνθήκες δοκιμαστικότητας μπορούν να χρησιμοποιηθούν, διότι μέχρι σήμερα δεν υπάρχει κάποια εργασία στη διεθνή βιβλιογραφία που να πραγματεύεται την ταυτόχρονη δοκιμή κατά τη λειτουργία τους.

Σχετικά με τη μελλοντική έρευνα που προκύπτει από το δεύτερο μέρος της διατριβής, αυτή επικεντρώνεται στην επέκταση του συνόλου εντολών των μικροεπεξεργαστών ώστε να επιτρέπουν λειτουργίες όπως η ανάγνωση και η εγγραφή των πεδίων του εκθέτη και του σημαίνοντος ενός καταχωρητή κινητής υποδιαστολής. Μια τέτοια επέκταση είναι εφικτή διότι δεν αναμένεται να αυξηθεί το μέγεθος του μικροεπεξεργαστή και η εισαγωγή επιπλέον υλικού δε θα επηρεάσει το κρίσιμο μονοπάτι (critical path) του μικροεπεξεργαστή. Η ύπαρξη τέτοιων εντολών αναμένεται να μειώσει το χρόνο εκτέλεσης του λογισμικού αυτοδοκιμής που έχει αναπτυχθεί με βάση προ-υπολογισμένα διανύσματα. Μια άλλη επέκταση του συνόλου εντολών αφορά την περίπτωση όπου ένας μικροεπεξεργαστής 32-bit διαθέτει μια μονάδα κινητής υποδιαστολής διπλής ακρίβειας. Στην περίπτωση αυτή, η επέκταση του συνόλου εντολών αφορά την προσθήκη εντολών οι οποίες θα μπορούν να μεταχειρισθούν δύο καταχωρητές των 32-bit σαν ένα καταχωρητή 64-bit.

Εναλλακτικά, θα πρέπει να εξεταστεί η προσθήκη καταχωρητών 64-bit και η τροποποίηση των εντολών, όπως της πρόσθεσης, και της ολίσθησης, ώστε να μεταχειρίζονται τους καταχωρητές αυτούς. Συνολικά αναμένεται πως οι παραπάνω επεκτάσεις του συνόλου εντολών, θα επιτρέψουν την ανάπτυξη λογισμικού αυτοδοκιμής για μονάδες κινητής υποδιαστολής με βάση προ-υπολογισμένα διανύσματα δοκιμής, το οποίο θα διαθέτει όχι μόνο μικρό ίχνος μνήμης αλλά ακόμα μικρότερο χρόνο εκτέλεσης και χαμηλότερη κατανάλωση ενέργειας.

Επίσης, αξίζει να μελετηθεί η ανάπτυξη λογισμικού αυτοδοκιμής τόσο για ακέραιες αριθμητικές μονάδες όσο και για μονάδες κινητής υποδιαστολής που βρίσκονται σε υπερβαθμιστούς (superscalar) μικροεπεξεργαστές. Οι υπερβαθμιστοί επεξεργαστές συνήθως ενσωματώνουν περισσότερες από μια αριθμητικές μονάδες για αύξηση της παραλληλίας εκτέλεσης των εντολών (Instruction Level Parallelism - ILP). Το λογισμικό αυτοδοκιμής θα πρέπει να είναι κατάλληλα σχεδιασμένο ώστε να εφαρμόζει διανύσματα δοκιμής σε όλες τις μονάδες και να φροντίζει για τη διάδοση τυχόν εσφαλμένων τιμών στις εξόδους του μικροεπεξεργαστή.

Η ανάπτυξη των πολυεπεξεργαστών σε επίπεδο τσιπ (Chip level MultiProcessing - CMP) [128] καθώς και της πολυνηματικής επεξεργασίας σε επίπεδο τσιπ (Chip MultiThreading - CMT) [128] σηματοδοτούν ένα νέο ερευνητικό πεδίο για την ανάπτυξη του λογισμικού αυτοδοκιμής. Το λογισμικό αυτοδοκιμής μπορεί να αναπτυχθεί σε επιμέρους τμήματα τα οποία θα εκτελούνται από πολλαπλά νήματα (threads) από πολλαπλούς επεξεργαστές με στόχο τη μείωση του συνολικού χρόνου εκτέλεσης της δοκιμής.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑ

Βιβλιογραφία

- [1] M. Bass and C. Christensen, "The future of the microprocessor business," *IEEE Spectrum*, vol. 39, no. 4, pp. 34–39, 2002.
- [2] D. Lewis, *Fundamentals of Embedded Software: Where C and Assembly Meet*. Prentice Hall, 2001, ch. 1, pp. 1–14.
- [3] J. Kahle, M. Day, H. Hofstee, C. Johns, T. Maeurer, and D. Shippy, "Introduction to the cell multiprocessor," *IBM J. Research and Development*, vol. 49, no. 4/5, pp. 589–604, 2005.
- [4] Microprocessors poster. Intel. [Online]. Available: http://download.intel.com/museum/Moores_Law/Printed_Materials/Microprocessor_Poster_24_36.pdf
- [5] International technology roadmap for semiconductors - itrs. [Online]. Available: <http://www.itrs.net/Links/2007ITRS/Home2007.htm>
- [6] Statistical analysis of floating point flaw. Intel. [Online]. Available: <http://support.intel.com/support/processors/pentium/fdiv/wp/>
- [7] D. Gizopoulos, Ed., *Advances in Electronic Testing, Challenges and Methodologies*, 1st ed., ser. Frontiers in Electronic Testing. Springer, 2006.
- [8] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design & Test of Computers*, vol. 22, no. 3, pp. 258–266, 2005.
- [9] G. Singer, "Current Trends and Future Directions in Test and DFT," in *Proc. of the 15th VLSI Test Symp.*, 1997, keynote Address.
- [10] B. Davis, *The Economics of Automatic Testing*. McGraw-Hill, 1982.
- [11] P. Nigh, W. Needham, K. Butler, P. Maxwell, and R. Aitken, "An experimental study comparing the relative effectiveness of functional, scan, iddq and delay- fault testing," in *Proc. of the 15th VLSI Test Symp*, 1997, pp. 459–464.

- [12] P. Gelsinger, "Discontinuities driven by a billion connected machines," *IEEE Des. Test. Comput.*, vol. 17, no. 1, pp. 7–15, 2000.
- [13] J. Gatej, L. Song, C. Pyron, R. Raina, and T. Munns, "Evaluating ATE features in terms of test escape rates and other cost of test culprits," in *International Test Conference*, 2002, pp. 1040–1048.
- [14] International technology roadmap for semiconductors - ITRS. [Online]. Available: <http://www.itrs.net/Links/2003ITRS/Home2003.htm>
- [15] ARM. [Online]. Available: <http://www.arm.com>
- [16] Texas Instruments. [Online]. Available: <http://www.ti.com>
- [17] Intel and floating point. Intel. [Online]. Available: <http://www.intel.com/standards/floatingpoint.pdf>
- [18] "ARIANE 5 Flight 501 Failure," 1996. [Online]. Available: <http://www.ima.umn.edu/~arnold/disasters/ariane5rep.html>
- [19] GAO/IMTEC-92-26, "Patriot Missile Software Problem," 1992. [Online]. Available: <http://www.fas.org/spp/starwars/gao/im92026.htm>
- [20] M. Farmwald, "On the design of high performance digital arithmetic circuits," Ph.D. dissertation, Stanford University, 1981.
- [21] M. Ercegovac and T. Lang, *Digital Arithmetic*. Morgan Kaufmann, 2003.
- [22] I. Koren, *Computer Arithmetic Algorithms*, 2nd ed. A K Peters Ltd, 2001.
- [23] *IBM PowerPC 970FX RISC Microprocessor User's Manual*, IBM, 2006.
- [24] Sun Niagara 2 Processor. [Online]. Available: <http://www.sun.com/processors/niagara/>
- [25] G. Xenoulis, M. Psarakis, D. Gizopoulos, and A. Paschalis, "Test generation methodology for high-speed floating point adders," in *Proc. of 11th IEEE On-Line Testing Symposium*, 2005, pp. 227–232.
- [26] —, "Testability analysis and scalable test generation for high-speed floating-point units," *IEEE Trans. Comput.*, vol. 55, no. 11, pp. 1149–1457, 2006.
- [27] G. Xenoulis, D. Gizopoulos, N. Kranitis, and A. Paschalis, "Low-cost, on-line software-based self-testing of embedded processor cores," in *Proc. of 9th IEEE On-Line Testing Symposium*, 2003, pp. 149–154.

- [28] G. Xenoulis, M. Psarakis, D. Gizopoulos, and A. Paschalis, "On-Line Periodic Self-Testing of High-Speed Floating-Point Units in Microprocessors," in *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2007, pp. 379–397.
- [29] G. Xenoulis, D. Gizopoulos, M. Psarakis, and A. Paschalis, "Instruction-based on-line periodic self-testing of microprocessors with floating-point units," *IEEE Trans. Dependable Secure Comput.*, 2008.
- [30] N. Jha and S. Gupta, *Testing of Digital Systems*. Cambridge University Press, 2002.
- [31] M. Abramovici, M. Breuer, and A. Friedman, *Digital Systems Testing & Testable Design*. Wiley-IEEE Press, 1994.
- [32] C. Constantinescu, "Trends and challenges in vlsi circuit reliability," *Micro*, vol. 23, no. 4, pp. 14–19, 2003.
- [33] H. Al-Asaad, B. Murray, and J. Hayes, "Online bist for embedded systems," *IEEE Design & Test of Computers*, vol. 15, no. 4, pp. 17–24, 1998.
- [34] S. Seshu and D. Freeman, "On improved diagnosis program," *IEEE Trans. Electron. Comput.*, vol. 7, no. 4, pp. 76–79, 1965.
- [35] J. Waicukauski, E. Eichelberger, D. Forlenza, E. Lindbloom, and T. McCarthy, "Fault simulation for structured VLSI," in *VLSI System Design*, 1985, pp. 20–32.
- [36] D. Armstrong, "A deductive method for simulating faults in logic circuits," *IEEE Trans. Comput.*, vol. 21, no. 5, pp. 464–471, 1972.
- [37] E. Ulrich and T. Baker, "Concurrent simulation of nearly identical digital networks," *IEEE Trans. Comput.*, vol. 7, no. 4, pp. 39–44, 1974.
- [38] T. Williams and N. Brown, "Defect level as a function of fault coverage," *IEEE Trans. Comput.*, vol. 30, no. 12, pp. 987–988, 1981.
- [39] G. Moore, "Cramming More Components Onto Integrated Circuits," *Electronics Magazine*, vol. 38, no. 8, pp. 114–117, 1965. [Online]. Available: <http://www.intel.com/technology/mooreslaw/index.htm>
- [40] —, in *IEEE International Solid-State Circuits Conference*, 2003, opening Talk. [Online]. Available: <http://www.intel.com/technology/mooreslaw/index.htm>
- [41] A. Ambler, M. Abadir, and S. Sastry, *Economics of design and test for electronic circuits and systems*. Ellis Horwood, 1990.

- [42] R. Eldred, "Test Routines Based on Symbolic Logical Statements," *Journal of the ACM*, vol. 6, no. 1, pp. 33–37, 1959.
- [43] V. Agarwal and A. Fung, "Multiple Fault Testing of Large Circuits by Single Fault Test Sets," *IEEE Trans. Comput.*, vol. 30, no. 11, pp. 855–865, 1981.
- [44] J. Jacob and N. N. Biswas, "GTBD Faults and Lower Bounds on Multiple Fault Coverage of Single Fault Test Sets," in *International Test Conference*, 1987, pp. 849–855.
- [45] J. Roth, "Diagnosis of automata failures: a calculus and a method," *IBM J. R&D*, vol. 10, no. 4, pp. 278–291, 1990.
- [46] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," *IEEE Trans. Comput.*, vol. 30, no. 3, pp. 215–222, 1981.
- [47] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms," *IEEE Trans. Comput.*, vol. 32, no. 12, pp. 1137–1144, 1983.
- [48] Y. Leventel and P. Menon, "Transition Faults in Combinational Circuits: Input Transition Test Generation and Fault Simulation," in *International Fault Tolerance Computing Symposium*, 1986, pp. 278–283.
- [49] J. Carter, V. Iyengar, and B. Rosen, "Efficient Test Coverage Determination for Delay Faults," in *International Test Conference*, 1987, pp. 418–427.
- [50] G. Smith, "Model for Delay Faults Based upon Paths," in *International Test Conference*, 1985, pp. 342–349.
- [51] M. Levi, "Cmos is most testable," in *International Test Conference*, 1981, pp. 217–220.
- [52] J. Soden, C. Hawkins, R. Gulati, and W. Mao, "Iddq testing: A review," *Journal of Electronic Testing: Theory and Applications*, vol. 3, no. 4, pp. 5–17, 1992.
- [53] K. Mei, "Bridging and Stuck-at Faults," *IEEE Trans. Comput.*, vol. 23, pp. 720–727, 1974.
- [54] W. Kautz, "Testing for faults in cellular logic arrays," in *8th Symposium on Switching Automata Theory*, 1967, pp. 161–174.
- [55] L. Goldstein and E. Thigpen, "SCOAP: Sandia Controllability/Observability Analysis Program," in *Design Automation Conference*, 1980, pp. 190–196.
- [56] T. Williams and K. Parker, *IEEE Trans. Comput.*, vol. 3, no. 1, pp. 2–15, 1982.
- [57] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures Design for Testability (Systems on Silicon)*. Morgan Kaufmann, 2006.

- [58] S. Narayann, R. Gupta, and M. Breuer, "Optimal configurin of multiple scan chains," *IEEE Trans. Comput.*, vol. 42, no. 9, pp. 1121–1131, 1993.
- [59] D. H. Lee and S. M. Reddy, "On determining scan flip-flops in partial scan designs," in *International Conference on Computer-Aided Design*, 1990, pp. 322–325.
- [60] S. Chakradhar, A. Balakrishnan, and V. Agrawal, "An exact algorithm for selecting partial scan flip-flops," in *Design Automation Conference*, vol. 81-86, 1994.
- [61] P. Ashar and S. Malik, "Implicit computation of minimum-cost feedback-vertex sets for partial scan and other applications," in *Design Automation Conference*, 1994, pp. 77–80.
- [62] R. Gupta and M. Breuer, "The BALLAST methodology for structured partial scan design," *IEEE Trans. Comput.*, vol. 4, no. 39, pp. 538–543, 1990.
- [63] G. Golobm, *Shift Register Sequence*. Aegean Park Press, 1982.
- [64] H. Schnurmann, E.Lindbloom, and R. Carpenter, "The weighted random test-pattern generator," *IEEE Trans. Comput.*, vol. 24, no. 7, pp. 695–700, 1975.
- [65] J. Rajski and J. Tyszer, *Arithmetic Built-in Self-Test for Embedded Systems*. Prentice Hall, 1997.
- [66] P. Hortensius, R. McLeod, W.Pries, D. Miller, and H. Card, "Cellular automata-based pseudorandom number generator for built-in self-test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 8, no. 8, pp. 842–859, 1989.
- [67] N. Benowitz, D. Clahoon, G. Alderson, J. Bauer, and C. Joeckel, "An advanced fault isolation system for digital logic," *IEEE Trans. Comput.*, vol. 24, no. 5, pp. 489–497, 1975.
- [68] C. Perkins, S. Sangani, H. Stopper, and W. Valitski, "Design for in-situ chip testing with a compact tester," in *International Test Conference*, 1980, pp. 29–41.
- [69] P. Bardell and W. McAnney, "Self-testing of multiple logic modules," in *International Test Conference*, 1982, pp. 200–204.
- [70] B. Konemann, J. Mucha, and G. Swiehoff, "Built-in logic observation techniques," in *International Test Conference*, 1979, pp. 37–41.
- [71] E. McCluskey, "Built-in self-test structures," *IEEE Des. Test. Comput.*, vol. 2, no. 2, p. 1985, 1985.
- [72] L.-T. Wang and E. McCluskey, "Concurrent built-in logic block observer," in *International Symposium on Circuits and Systems*, 1986, pp. 1054–1057.

- [73] S. Jain and A. Susskind, "Test strategy for microprocessors," in *Design Automation Conference*, 1983, pp. 703–708.
- [74] S. Adham and S. Gupta, "Dp-bist: a built in self-test for dsp data paths - a low overhead and high fault coverage technique," in *Asian Test Symposium (ATS)*, 1996, pp. 205–215.
- [75] S. Thatte and J. Abraham, "Test Generation for Microprocessors," *IEEE Trans. Comput.*, vol. 29, no. 6, pp. 429–441, 1980.
- [76] R. Tupuri and J. Abraham, "A Novel Functional Test Generation Method for Processors using Commercial ATPG," in *International Test Conference*, 1997, pp. 743–752.
- [77] D. Gizopoulos, A. Paschalis, and Y. Zorian, *Embedded Processor-Based Self-Test*. Kluwer Academic Publishers, 2004.
- [78] J. Shen and J. Abraham, "Native mode functional test generation for microprocessors with applications to self-test and design validation," in *International Test Conference*, 1998, pp. 990–998.
- [79] K. Batcher and C. Papachristou, "Instruction randomization self test for processor cores," in *VLSI Test Symposium*, 1999, pp. 34–40.
- [80] F. Corno, G. Cumani, M. S. Reorda, and G. Squillero, "Fully Automatic Test Program Generation for Microprocessor Cores," in *Design, Automation and Test in Europe*, 2003, pp. 1006–1011.
- [81] P. Parvathala, K. Maneparambil, and W. Lindsay, "FRITS – A Microprocessor Functional BIST Method," in *International Test Conference*, 2002, pp. 590–598.
- [82] L. Chen and S. Dey, "Software-based self-testing methodology for processor cores," in *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 3, 2001, pp. 369–380.
- [83] L. Chen, S. Ravi, A. Raghunathan, and S. Dey, "A scalable software-based self-testing methodology for programmable processors," in *Design Automation Conference*, 2003, pp. 548–553.
- [84] C. H.-P. Wen, L.-C. Wang, and K.-T. Cheng, "Simulation-based functional test generation for embedded processors," *IEEE Trans. Comput.*, vol. 55, no. 11, pp. 1335–1343, 2006.
- [85] S. Gurumurthy, S. Vasudevan, and J. Abraham, "Automated mapping of pre-computed module-level test sequences to processor instructions," in *International Test Conference*, 2005.
- [86] —, "Automatic generation of instruction sequences targeting hard-to-detect structural faults in a processor," in *International Test Conference*, 2006, pp. 294–303.

- [87] N. Kranitis, A. Paschalis, D. Gizopoulos, and Y. Zorian, "Instruction-based self-testing of processor cores," *Journal of Electronic Testing: Theory and Applications*, no. 19, pp. 103–112, 2002, special Issue on 20th IEEE VLSI Test Symposium 2002.
- [88] N. Kranitis, G. Xenoulis, A. Paschalis, and D. Gizopoulos, "Application and analysis of rt-level software-based self-testing for embedded processor cores," in *International Test Conference*. IEEE, 2003, pp. 431–440.
- [89] N. Kranitis, A. Paschalis, D. Gizopoulos, and G. Xenoulis, "Software-Based Self-Testing of Embedded Processors," *IEEE Trans. Comput.*, vol. 54, no. 4, pp. 461–475, 2005.
- [90] M. Psarakis, D. Gizopoulos, M. Hatzimihail, A. Paschalis, A. Raghunathan, and S. Ravi, "Systematic software-based self-test for pipelined processors," in *Design Automation Conference*, 2003, pp. 393–398.
- [91] D. Gizopoulos, M. Psarakis, M. Hatzimihail, M. Maniatakos, A. Paschalis, A. Raghunathan, and S. Ravi, "Systematic software-based self-test for pipelined processors," *IEEE Trans. VLSI Syst.*, vol. 16, no. 11, pp. 1441–1453, 2008.
- [92] D. Deley. THE PENTIUM DIVISION FLAW. [Online]. Available: <http://members.cox.net/srice1/pentbug/introduction.html>
- [93] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, 1st ed. Oxford University Press, 1999.
- [94] *IEEE-754 standard for binary floating-point arithmetic*, IEEE Std., 1985.
- [95] L. Lacassagne, D. Etiemble, and S. A. O. Kablia, "16-bit Floating Point Instructions for Embedded Multimedia Applications," in *CAMP '05: Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception (CAMP'05)*, 2005, pp. 198–203.
- [96] *IEEE-754 standard for binary floating-point arithmetic*, IEEE Std., 2008.
- [97] A. Weinberger and J. Smith, "A logic for high-speed addition," *National Bureau of Standards Circular*, vol. 591, pp. 3–12, 1958.
- [98] R. Brent and H. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. C-31, pp. 260–264, 1982.
- [99] P. Kogge and H. Stone, "A Parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans. Comput.*, vol. 22, pp. 786–793, 1973.
- [100] J. Sklansky, "Conditional sum addition logic," *IEEE Trans. Electron. Comput.*, vol. EC-9, no. 6, pp. 226–231, Jun. 1960.

- [101] V. G. Oklobdzija, "An algorithmic and novel design of a leading zero detector circuit: Comparison with logic synthesis," *IEEE Trans. VLSI Syst.*, vol. 2, no. 1, pp. 124–128, 1993.
- [102] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd ed. MORGAN KAUFFMAN, 2002, ch. Appendix H.
- [103] Pentium ii math bug? [Online]. Available: <http://www.x86.org/secrets/dan0411.htm>
- [104] U. Sparmann, "Derivation of high quality tests for large heterogeneous circuits: floating-point operations," in *3rd European Conference on Design Automation*, 1992, pp. 355–360.
- [105] M. Schulz, E. Trischler, and T. Sarfert, "SOCRATES: A highly efficient automatic test pattern generation system," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 7, no. 1, pp. 126–137, 1988.
- [106] J. Rajski and J. Tyszer, "Design of random pattern testable floating point adders," in *3rd Asian Test Symposium*, 1994, pp. 227–232.
- [107] K. Hatayama, K. Hikone, T. Miyazaki, and H. Yamada, "A practical approach to instruction-based test generation for functional modules of vlsi processors," in *VLSI Test Symposium*, 2000, pp. 17–22.
- [108] "Testability strategy and DFT methodology of CalmRISC32," in *2nd IEEE Asia Pacific Conference on ASICs*, 2000, pp. 295–298.
- [109] "Design-for-testability of the FLOVA," in *2nd IEEE Asia Pacific Conference on ASICs*, 2000, pp. 319–322.
- [110] "An efficient implementation of bist for floating point dsp processor," in *2nd IEEE Asia Pacific Conference on ASICs*, 2000, pp. 273–276.
- [111] I. Bayraktaroglu and M. d'Abreu, "ATPG based functional test for data paths: application to a floating point unit," in *9th IEEE International High-Level Design Validation and Test Workshop*, 2004, pp. 37–40.
- [112] *Mobile Power Guidelines 2000*, INTEL, 1998.
- [113] B. Becker, "Efficient testing of optimal time adders," *IEEE Trans. Comput.*, vol. 37, pp. 1113–1121, 1988.
- [114] N. Kranitis, M. Psarakis, D. Gizopoulos, A. Paschalis, and Y. Zorian, "An effective deterministic BIST scheme for shifter/accumulator pairs in datapaths," *Journal of Electronic Testing: Theory and Applications*, vol. 17, pp. 97–107, 2001.

- [115] I. Bayraktaroglu, J. Hunt, and D. Watkins, "Cache Resident Functional Microprocessor Testing: Avoiding High Speed I/O issues," in *International Test Conference*, 2006, paper 27.2.
- [116] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, "Logic BIST for Large Industrial Designs: Real Issues and Case Studies," in *International Test Conference*, 1999, pp. 358–367, paper 14.2.
- [117] T. Rao and E. Fujiwara, *Error-Control Coding for Computer Systems*. Prentice Hall, 1989.
- [118] T. Rao, *Error Coding for Arithmetic Processors*. Academic Press, 1974.
- [119] M. Nicolaidis and Y. Zorian, "On-Line Testing for VLSI – A Compendium of Approaches," *Journal of Electronic Testing: Theory & Applications*, vol. 12, pp. 7–20, 1998.
- [120] B. Becker, "On the generation of area-time optimal testable adders," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, pp. 1049–1066, 1995.
- [121] D. Gizopoulos, A. Paschalis, and Y. Zorian, "An effective built-in self-test scheme for parallel multipliers," *IEEE Trans. Comput.*, vol. 48, pp. 936–950, 1999.
- [122] D. Gizopoulos, M. Psarakis, A. Paschalis, and Y. Zorian, "Easily testable cellular carry lookahead adders," *Journal of Electronic Testing: Theory and Applications*, vol. 19, pp. 285–298, 2003.
- [123] H. Al-Asaad, J. Hayes, and B. Murray, "Scalable test generators for high-speed datapath circuits," *Journal of Electronic Testing: Theory and Applications*, vol. 12, no. 1-2, pp. 111–125, 1998.
- [124] *PowerPC Microprocessor Family: The Programming Environments for 32-Bit Microprocessors*, IBM, 2000. [Online]. Available: <http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/852569B20050FF778525699600719DF2>
- [125] "Fp-library." [Online]. Available: <http://perso.ens-lyon.fr/jeremie.detrey/FPLibrary/index.html>
- [126] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in *International Symposium on Computer Architecture*. IEEE/ACM, 2000, pp. 83–94.
- [127] D. Burger, T. Austin, and S. Bennett, "Evaluating future microprocessors: The simplescalar toolset," University of Wisconsin-Madison. Computer Sciences Department., Tech. Rep. CS-TR-1996-1308, 1996.

- [128] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd ed. MORGAN KAUFFMAN, 2002.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ