

Πανεπιστήμιο Κρήτης, Ηράκλειο
Τμήμα Επιστήμης Υπολογιστών

Αποσυγχρονισμός Βιομηχανικών Κυκλωμάτων

Μεταπτυχιακή Εργασία

Εμμανουήλ Παντελιάς
<pantelia@csd.uoc.gr>

Ηράκλειο, 4 Δεκεμβρίου 2007

Επόπτης, Καθηγητής Μανόλης Κατεβαίνης,
Πρόεδρος της Εξεταστικής Επιτροπής

Καθηγητής Απόστολος Τραγανίτης,
Μέλος της Εξεταστικής Επιτροπής

Αναπλ. Καθηγητής Πολυτεχνείου Κρήτης Διονύσης Πνευματικάτος,
Μέλος της Εξεταστικής Επιτροπής

Καθηγητής Πάνος Τραχανιάς,
Διευθυντής Μεταπτυχιακών Σπουδών



Πανεπιστήμιο Κρήτης, Ηράκλειο
Τμήμα Επιστήμης Υπολογιστών

Περίληψη

Τα ασύγχρονα κυκλώματα έχουν την ικανότητα να προσαρμόζουν τον χρονισμό τους αυτόματα, σε διαφορετικές συνθήκες μεταβλητότητας. Επίσης, στις περισσότερες περιπτώσεις, παρουσιάζουν χαμηλότερες ηλεκτρομαγνητικές εκπομπές σε σύγκριση με τα αντίστοιχα σύγχρονα. Ο αποσυγχρονισμός είναι μια αυτοματοποιημένη μέθοδος για την μετατροπή ενός σύγχρονου κυκλώματος σε ένα αντίστοιχο ασύγχρονο. Η μέθοδος αυτή χρησιμοποιεί εργαλεία, τα οποία είναι πλήρως συμβατά με την αυτοματοποιημένη ροή της βιομηχανίας.

Η εργασία αυτή επιχειρεί να γενικεύσει τη μέθοδο του αποσυγχρονισμού σε κυκλώματα βιομηχανικής κλίμακας. Εξετάζει ορισμένα προχωρημένα θέματα που δεν είχαν μελετηθεί έως τώρα, όπως ο σωστός τρόπος αντιμετώπισης της φραγής ρολογιού, εναλλακτικοί τρόποι δημιουργίας ομάδων αποσυγχρονισμού, αρχικοποίηση, επικοινωνία με σύγχρονα τμήματα και επαλήθευση ορθής λειτουργίας.

Τρία βιομηχανικά κυκλώματα αποσυγχρονίστηκαν, με στόχο τη μείωση των ηλεκτρομαγνητικών εκπομπών. Για τον σκοπό αυτό χρησιμοποιήθηκαν εργαλεία αυτόματου αποσυγχρονισμού, με τη συνεργασία ερευνητών του ΙΤΕ. Σε αυτά έγιναν οι κατάλληλες προσθήκες και αλλαγές, για την αντιμετώπιση των παραπάνω θεμάτων. Τα αποτελέσματα καταδεικνύουν την δυνατότητα χρήσης της μεθόδου για τις ανάγκες της βιομηχανίας.

Abstract

Asynchronous circuits have the ability of adapting their timing automatically, in different variability conditions. Also, in most cases, they exhibit lower electromagnetic emissions compared to the corresponding synchronous ones. Desynchronization is an automatic method for the transformation of a synchronous circuit to a corresponding asynchronous. This method uses tools, which are fully compliant to the industry's automatic flow.

This work attempts to generalize the desynchronization method to industrial scale circuits. It examines some advanced issues that have never been studied before, such as correct way of dealing with clock gating, alternative ways of creating desynchronized regions, reset, communication with synchronous modules and functional verification.

Three industrial circuits were desynchronized, with the object of reducing the electromagnetic emissions. For this purpose, automatic desynchronization tools were used, in collaboration with researchers of FORTH. The appropriate additions and modifications were made, in order to cope with the above issues. The results demonstrate the ability of this method to be employed for the industry's needs.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους γονείς μου για την αμέριστη συμπαράσταση και υποστήριξή τους καθόλη τη διάρκεια των σπουδών μου.

Θα ήθελα να ευχαριστήσω τον επιβλέποντά μου Δρ. Χρήστο Σωτηρίου για την πολύτιμη καθοδήγησή του καθόλη τη διάρκεια αυτής της εργασίας, όπως και τον Σπύρο Λυμπέρη για την διαρκή βοήθειά και υποστήριξή του. Επίσης, θα ήθελα να ευχαριστήσω και τα υπόλοιπα μέλη της Ομάδας Ασύγχρονων Κυκλωμάτων και Συστημάτων του ΙΤΕ, Ν. Ανδρίκο, Ι. Ασλανίδη, Β. Ζεμπίλη, Β. Κασσαπάκη, Ε. Κουναλάκη και Π. Ματθαιάκη.

Θα ήθελα να ευχαριστήσω το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης. Επίσης, θα ήθελα να ευχαριστήσω το ΙΤΕ-III για την υλικοτεχνική υποστήριξη και χρηματοδότηση αυτής της εργασίας.

Θα ήθελα να ευχαριστήσω την εταιρία ημιαγωγών, στην οποία έγινε μέρος αυτής της εργασίας, καθώς και τους ανθρώπους της, για τη διαρκή βοήθειά τους.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου, Ε. Αναστασοπούλου, Β. Λεκάκη, Α. Λεπίδα, Μ. Μουδάτσο και Η. Ραυτόπουλο.

Περιεχόμενα

1	Εισαγωγή	1
2	Αποσυγχρονισμός	4
2.1	Βασική Ιδέα	4
2.2	Κύρια Βήματα Αποσυγχρονισμού	5
2.2.1	Αντικατάσταση Καταχωρητών από Μανταλωτές	5
2.2.2	Ομαδοποίηση	6
2.2.3	Εισαγωγή Δικτύου Ασύγχρονων Ελεγκτών	7
2.2.4	Εισαγωγή Στοιχείων Καθυστέρησης	8
2.3	Στοιχεία C	8
2.4	Ασύγχρονοι Ελεγκτές	8
2.4.1	Γενική Περιγραφή Σημάτων	10
2.4.2	Πρωτόκολλα Χειραψίας και Είδη Ασύγχρονων Ελεγκτών	10
2.4.3	Ένας Τετραφασικός Ασύγχρονος Ελεγκτής	11
2.4.4	Ένας Διφασικός Ασύγχρονος Ελεγκτής	11
2.5	Στοιχεία Καθυστέρησης	12
2.5.1	Εύρεση Εξαρτήσεων των Ομάδων	12
2.5.2	Υπολογισμός Καθυστέρησης	13
2.5.3	Ασύμμετρα Στοιχεία Καθυστέρησης	14
2.5.4	Συμμετρικά Στοιχεία Καθυστέρησης	14
2.5.5	Χρονικό Περιθώριο και Χρήση Πολυπλέκτη	15
2.6	Ισοδυναμία Ροής	15
2.7	Δοκιμή Αποσυγχρονισμένων Κυκλωμάτων	17
2.7.1	Αλυσίδες Σάρωσης	17
2.7.2	Built-In Self-Test	17
3	Προχωρημένα Θέματα Αποσυγχρονισμού	19
3.1	Φραγή Σήματος Ρολογιού	19
3.1.1	Κόκκος Φραγής	20
3.1.2	Χρήση Μανταλωτή στην Φραγή Ρολογιού	22

3.1.3	Αποσυγχρονισμός και Φραγή Ρολογιού Μικρού Κόκκου	23
3.2	Αρχικοποίηση	24
3.3	Επικοινωνία με Σύγχρονο Τμήμα	24
3.3.1	Επικοινωνία με τη Χρήση FIFOs	25
3.3.2	Συγχρονισμός Μέσω του Αποσυγχρονισμένου Κυκλώματος	25
3.3.3	Συγχρονισμός Μέσω Εξωτερικού Ρολογιού	26
3.4	Testbench	27
3.4.1	Μετατροπές	27
3.4.2	Μέθοδος Επαλήθευσης	27
3.4.3	Προβληματικές Περιπτώσεις	27
3.5	Υλοποίηση Στοιχείων C	27
3.6	Χρονική Βελτιστοποίηση του Ασύγχρονου Δικτύου Ελέγχου	28
3.7	Ομαδοποίηση	28
3.7.1	Καθολική Λογική	29
3.7.2	Χειροκίνητη Ομαδοποίηση	29
3.7.3	Αλγόριθμος “Χαμηλής Προσπάθειας”	29
3.7.4	Αλγόριθμος “Μεσαίας Προσπάθειας”	29
3.7.5	Κόκκος Αποσυγχρονισμού	30
3.8	Σύγκριση Διαφασικών και Τετραφασικών Ασύγχρονων Ελεγκτών	30
4	Ροή Αποσυγχρονισμού	31
4.1	Ανάλυση Βιβλιοθήκης Τεχνολογίας	31
4.2	Εισαγωγή Σύγχρονου Κυκλώματος	33
4.3	Ομαδοποίηση	33
4.4	Δημιουργία Γράφου Εξαρτήσεων των Ομάδων	35
4.5	Αντικατάσταση Καταχωρητών από Μανταλωτές	35
4.6	Κατασκευή Στοιχείων Καθυστέρησης	36
4.7	Εισαγωγή Ασύγχρονου Δικτύου Ελέγχου	36
4.8	Χρονική Βελτιστοποίηση του Ασύγχρονου Δικτύου Ελέγχου	36
4.9	Εξαγωγή Αποσυγχρονισμένου Κυκλώματος	36
5	Συνεισφορά στην Ανάπτυξη του <i>Nanosync</i>	37
5.1	Ανάπτυξη Κώδικα	37
5.2	Αποσφαλμάτωση	37
6	Τρία Βιομηχανικά Κυκλώματα για Αποσυγχρονισμό	38
6.1	Επεξεργαστής IND1a	38
6.1.1	Σύντομη Περιγραφή του Κυκλώματος	38
6.1.2	Αποσυγχρονισμός και Επαλήθευση του IND1a	39

6.2	Επεξεργαστής IND1b	42
6.2.1	Σύντομη Περιγραφή του Κυκλώματος	42
6.2.2	Αποσυγχρονισμός του IND1b	42
6.2.3	Προσομοίωση και Επαλήθευση	42
6.3	Μικροελεγκτής IND2	42
6.3.1	Σύντομη Περιγραφή του Κυκλώματος	42
6.3.2	Αποσυγχρονισμός του IND2	42
6.3.3	Προσομοίωση και Επαλήθευση	42
6.4	Περίληψη	43
7	Πειράματα και Αποτελέσματα	44
7.1	Επεξεργαστής IND1b	44
7.1.1	Περίοδος Λειτουργίας	44
7.1.2	Κατανάλωση	45
7.1.3	Ηλεκτρομαγνητική Εκπομπή	45
7.1.4	Εμβαδόν	45
7.2	Μικροελεγκτής IND2	46
7.2.1	Εμβαδόν	46
8	Συμπεράσματα και Μελλοντική Εργασία	47
	Βιβλιογραφία	50
A'	Επεξεργαστής IND1b	51
A'.1	Ιεραρχία του IND1bmacro	51
A'.2	Αποσυγχρονισμός του IND1bwrap	51
A'.2.1	Scripts	51
A'.2.2	Αναφορά του <i>Nanosync</i>	51
A'.2.3	Γράφος Εξαρτήσεων των Ομάδων	51
A'.3	Αποσυγχρονισμός του IND1bCore	51
A'.3.1	Scripts	52
A'.3.2	Αναφορά του <i>Nanosync</i>	52
A'.3.3	Γράφος Εξαρτήσεων των Ομάδων	52
A'.4	Αποσυγχρονισμός του IND1bCore με τον Αλγόριθμο “Μεσαίας Προ- σπάθειας”	52
A'.4.1	Scripts	52
A'.4.2	Αναφορές του <i>Nanosync</i>	52
A'.5	Scripts του <i>Cadence Encounter</i>	52

B' Μικροελεγκτής IND2	53
B'.1 Ιεραρχία του digital_module	53
B'.2 Εμβαδόν Τμημάτων του digital_module	53
B'.3 Αποσυγχρονισμός του IND2cpu	53
B'.3.1 Scripts	53
B'.3.2 Αναφορά του <i>Nanosync</i>	53
B'.3.3 Γράφος Εξαρτήσεων των Ομάδων	53
Γ' Τεχνικά – Εμπιστευτικά Θέματα	54
Γ'.1 Ένας Διαφασικός Ασύγχρονος Ελεγκτής	54
Γ'.2 Συμμετρικά Στοιχεία Καθυστέρησης	55
Γ'.3 Αποσυγχρονισμός και Φραγή Ρολογιού Μικρού Κόκκου	56
Γ'.4 Αρχικοποίηση	59
Γ'.5 Συγχρονισμός Μέσω Εξωτερικού Ρολογιού	61
Γ'.6 Testbench	63
Γ'.6.1 Μετατροπές	63
Γ'.6.2 Μέθοδος Επαλήθευσης	64
Γ'.6.3 Προβληματικές Περιπτώσεις	65
Γ'.7 Υλοποίηση Στοιχείων C	66
Γ'.8 Χρονική Βελτιστοποίηση του Ασύγχρονου Δικτύου Ελέγχου	67
Γ'.9 Ομαδοποίηση	69
Γ'.9.1 Καθολική Λογική	69
Γ'.9.2 Χειροκίνητη Ομαδοποίηση	69
Γ'.9.3 Αλγόριθμος “Μεσαίας Προσπάθειας”	71
Γ'.9.4 Κόκκος Αποσυγχρονισμού	73
Γ'.10 Συνεισφορά στην Ανάπτυξη του <i>Nanosync</i>	73
Γ'.10.1 Ανάπτυξη Κώδικα	73
Γ'.10.2 Αποσφαλάτωση	75
Γ'.11 Επεξεργαστής IND1a	75
Γ'.11.1 Σύντομη Περιγραφή του Κυκλώματος	75
Γ'.12 Επεξεργαστής IND1b	80
Γ'.12.1 Σύντομη Περιγραφή του Κυκλώματος	80
Γ'.12.2 Αποσυγχρονισμός του IND1b	83
Γ'.12.3 Προσομοίωση και Επαλήθευση	89
Γ'.13 Μικροελεγκτής IND2	94
Γ'.13.1 Σύντομη Περιγραφή του Κυκλώματος	94
Γ'.13.2 Αποσυγχρονισμός του IND2	97
Γ'.13.3 Προσομοίωση και Επαλήθευση	98

Γ.14 Πειράματα και Αποτελέσματα για τον επεξεργαστή IND1b	99
Γ.14.1 Περίοδος Λειτουργίας	99
Γ.14.2 Κατανάλωση	100
Γ.14.3 Ηλεκτρομαγνητική Εκπομπή	101
Γ.14.4 Εμβαδόν	109
Γ.15 Πειράματα και Αποτελέσματα για τον μικροελεγκτή IND2	111
Γ.15.1 Εμβαδόν	111

Κατάλογος Σχημάτων

2.1	Παράδειγμα σύγχρονου κυκλώματος.	5
2.2	Παράδειγμα βρόχου καταχωρητών, όπου οποιαδήποτε χρονική απόκλιση δεν είναι ανεκτή.	6
2.3	Παράδειγμα αντικατάστασης καταχωρητή με υποστήριξη αλυσίδας σάρωσης για κατασκευαστική δοκιμή.	6
2.4	Γράφος εξαρτήσεων των ομάδων του Σχήματος 2.1.	7
2.5	Ασύγχρονο δίκτυο ελέγχου του Σχήματος 2.1.	8
2.6	Τρόπος σύνδεσης πολλαπλών ασύγχρονων ελεγκτών σε έναν.	9
2.7	Τρόπος σύνδεσης ενός ασύγχρονου ελεγκτή σε πολλαπλούς.	9
2.8	Διάγραμμα εισόδων–εξόδων Ασύγχρονου Ελεγκτή.	10
2.9	Ζεύγος τετραφασικών ασύγχρονων ελεγκτών για μανταλωτές Αφέντη και Σκλάβου.	12
2.10	Υπολογισμός στοιχείου καθυστέρησης.	14
2.11	Παράδειγμα ασύμμετρου στοιχείου καθυστέρησης.	14
2.12	Παράδειγμα συμμετρικού στοιχείου καθυστέρησης.	15
2.13	Στοιχείο καθυστέρησης με πολυπλέκτη.	16
2.14	Ισοδυναμία ροής.	16
2.15	Υποστήριξη ελέγχου με αλυσίδα σάρωσης για τους ασύγχρονους ελεγκτές.	17
3.1	Υλοποίηση του <i>Enable</i> με πολυπλέκτες.	20
3.2	Υλοποίηση του <i>Enable</i> με φραγή ρολογιού.	21
3.3	Φραγή ρολογιού μεγάλου κόκκου.	21
3.4	Φραγή ρολογιού μικρού κόκκου.	22
3.5	Κύκλωμα φραγής ρολογιού χωρίς μανταλωτή.	23
3.6	Κύκλωμα φραγής ρολογιού με μανταλωτή.	23
3.7	Κύκλωμα φραγής ρολογιού με μανταλωτή και υποστήριξη αλυσίδας σάρωσης.	24
3.8	Επικοινωνία σύγχρονου και αποσυγχρονισμένου κυκλώματος με τη χρήση FIFOs.	25
3.9	Τρόπος σύνδεσης σημάτων χειραψίας σε ένα αποσυγχρονισμένο κύκλωμα.	26

3.10	Γενική εικόνα υλοποίησης στοιχείου C.	28
4.1	Διάγραμμα ροής αποσυγχρονισμού.	32
4.2	Περίπτωση καθαρισμού λογικής.	35
6.1	Λανθασμένος τρόπος αντικατάστασης των καταχωρητών με <i>enable</i> . . .	41
Γ'.1	Ζεύγος διφασικών ασύγχρονων ελεγκτών για μανταλωτές Αφέντη και Σκλάβου.	54
Γ'.2	ΓΜΣ διφασικών ασύγχρονων ελεγκτών για μανταλωτές Αφέντη και Σκλάβου.	55
Γ'.3	Αποσυγχρονισμένη έκδοση για το παράδειγμα του Σχήματος 3.2. . . .	57
Γ'.4	ΣΜΓ των τετραφασικών ελεγκτών της ενότητας 2.4.3.	58
Γ'.5	Διαφοροποιημένη έκδοση των ελεγκτών του Σχήματος Γ'.1.	59
Γ'.6	Συγχρονισμός μέσω εξωτερικού ρολογιού.	61
Γ'.7	Χρονισμός εξόδου σύγχρονου τμήματος.	62
Γ'.8	Διεπαφή για την περίπτωση των ελεγκτών της Ενότητας 2.4.4.	63
Γ'.9	Παράδειγμα μεταβλητού ρολογιού σε testbench.	66
Γ'.10	Υλοποίηση κυκλώματος Preset–Clear με μανταλωτή τύπου D.	67
Γ'.11	Παράδειγμα υλοποίησης στοιχείου C με επιπλέον είσοδο αρχικοποίησης. . .	67
Γ'.12	Είδη καθυστέρησης σε ένα ασύγχρονο δίκτυο ελέγχου.	68
Γ'.13	Παράδειγμα ομαδοποίησης ενός σύγχρονου κυκλώματος.	71
Γ'.14	Εξαρτήσεις καταχωρητών.	72
Γ'.15	Παράδειγμα συνδέσεων σε έναν γράφο κυκλώματος.	76
Γ'.16	Αποτέλεσμα της <i>write_netlist</i> για τον γράφο κυκλώματος του Σχήματος Γ'.15.	77
Γ'.17	Σχηματικό διάγραμμα του επεξεργαστή IND1a.	78
Γ'.18	Σχηματικό διάγραμμα της κεντρικής μονάδας επεξεργασίας του IND1. . .	79
Γ'.19	Κύτταρο βιβλιοθήκης clockGateCell.	79
Γ'.20	Σχηματικό διάγραμμα του επεξεργαστή IND1b.	80
Γ'.21	Διάγραμμα εισόδων–εξόδων του επεξεργαστή IND1b (μέρος 1).	81
Γ'.22	Διάγραμμα εισόδων–εξόδων του επεξεργαστή IND1b (μέρος 2).	82
Γ'.23	Ιεραρχία τμημάτων του IND1bmacro.	82
Γ'.24	Ιεραρχία τμημάτων του IND1bCore.	83
Γ'.25	Αφαίρεση των καταχωρητών από το IND1bCore.	84
Γ'.26	Διπλασιασμός των clockGateCell.	85
Γ'.27	Δομή αρχείων πριν την χρησιμοποίηση του <i>Nanosync</i>	87
Γ'.28	Σχηματικό διάγραμμα του testbench.	90
Γ'.29	Ιδανικές συνδέσεις στον σύγχρονο IND1b.	91

Γ'.30	Ιδανικές συνδέσεις στον αποσυγχρονισμένο IND1b.	92
Γ'.31	Σχηματικό διάγραμμα του μικροελεγκτή IND2.	95
Γ'.32	Ιεραρχία τμημάτων του IND2.	96
Γ'.33	Ιεραρχία τμημάτων του IND2cpu.	96
Γ'.34	Ένταση ρεύματος σύγχρονου και αποσυγχρονισμένου IND1b, ως προς τον χρόνο (για 2 κύκλους ρολογιού).	102
Γ'.35	Ένταση ρεύματος σύγχρονου και αποσυγχρονισμένου IND1b, ως προς τον χρόνο (για πολλούς κύκλους ρολογιού).	103
Γ'.36	Ένταση ρεύματος σύγχρονου και αποσυγχρονισμένου IND1b, στην περιοχή των συχνοτήτων (εύρος συχνοτήτων 500MHz).	104
Γ'.37	Ένταση ρεύματος σύγχρονου και αποσυγχρονισμένου IND1b, στην περιοχή των συχνοτήτων (εύρος συχνοτήτων 1GHz).	105
Γ'.38	Ένταση ρεύματος σύγχρονου και αποσυγχρονισμένου IND1b, στην περιοχή των συχνοτήτων (εύρος συχνοτήτων 2GHz).	106
Γ'.39	Ένταση ρεύματος σύγχρονου και αποσυγχρονισμένου IND1b, στην περιοχή των συχνοτήτων (εύρος συχνοτήτων 4GHz).	107
Γ'.40	Ένταση ρεύματος σύγχρονου και αποσυγχρονισμένου IND1b, στην περιοχή των συχνοτήτων (εύρος συχνοτήτων 10GHz).	108
Γ'.41	Σχέση του πλήθους των ομάδων αποσυγχρονισμού, με το ποσοστό της επιβάρυνσης εμβადού και των στοιχείων καθυστέρησης, στον αλγόριθμο “μεσαίας προσπάθειας”.	112
Γ'.42	Σχέση του πλήθους των ομάδων αποσυγχρονισμού, με το πραγματικό και το μέγιστο δυνατό πλήθος των στοιχείων καθυστέρησης, στον αλγόριθμο “μεσαίας προσπάθειας”.	113

Κατάλογος Πινάκων

2.1	Πίνακας αληθείας στοιχείου C.	9
Γ'.1	Πίνακας αληθείας του clockGateCell.	78
Γ'.2	Περίοδος σύγχρονου και αποσυγχρονισμένου IND1b στις τρεις συνθήκες χαρακτηρισμού της βιβλιοθήκης τεχνολογίας.	100
Γ'.3	Περίοδος αποσυγχρονισμένου IND1b σε διαφορετικές τιμές τάσης.	100
Γ'.4	Κατανάλωση σύγχρονου και αποσυγχρονισμένου IND1b σε διαφορετικές συνθήκες λειτουργίας.	101
Γ'.5	Επιβάρυνση εμβαδού στον αποσυγχρονισμό του IND1bwrap.	109
Γ'.6	Επιβάρυνση εμβαδού στον αποσυγχρονισμό του IND1bCore.	110
Γ'.7	Αποτελέσματα αποσυγχρονισμού του IND1bCore με τη χρήση του αλγόριθμου “μεσαίας προσπάθειας”.	111
Γ'.8	Ποσοστό επιβάρυνσης εμβαδού του IND2cru.	113

Κεφάλαιο 1

Εισαγωγή

Η κατασκευαστική μεταβλητότητα αποτελεί ένα από τα μεγαλύτερα προβλήματα στην σχεδίαση των σύγχρονων κυκλωμάτων. Μάλιστα, το πρόβλημα γίνεται ολοένα μεγαλύτερο με την αυξανόμενη κλιμάκωση της τεχνολογίας. Μία μέθοδος που προσπαθεί να αντιμετωπίσει το πρόβλημα είναι η στατιστική στατική χρονική ανάλυση (Statistical Static Timing Analysis – SSTA), η οποία στοχεύει στην πρόβλεψη της καθυστέρησης των κυκλωμάτων, με την χρήση εμπειρικών μοντέλων.

Η σχεδίαση ασύγχρονων κυκλωμάτων αποτελεί μια εναλλακτική μέθοδο αντιμετώπισης του προβλήματος της μεταβλητότητας. Τα πλεονεκτήματά των ασύγχρονων κυκλωμάτων είναι:

- η ιδιότητα να προσαρμόζουν τον χρονισμό τους ανάλογα με τις υπάρχουσες συνθήκες διεργασίας, τάσης και θερμοκρασίας·
- η μικρότερη ηλεκτρομαγνητική παρεμβολή, η οποία προκύπτει από τη χρονική διασπορά της δραστηριότητας των λογικών πυλών.

Παρόλα αυτά, η ασύγχρονη σχεδίαση έχει και ορισμένα μειονεκτήματα, όπως είναι το μεγαλύτερο εμβαδόν και η έλλειψη κατάλληλων εργαλείων EDA, τα οποία μπορούν να χρησιμοποιηθούν στην υπάρχουσα αυτοματοποιημένη ροή. Ειδικά το τελευταίο, αποτελεί τον κυριότερο περιοριστικό παράγοντα για την αποδοχή της από την βιομηχανία.

Ο αποσυγχρονισμός [2, 3] προσπαθεί να λύσει το προηγούμενο πρόβλημα. Αποτελεί μια αυτοματοποιημένη μέθοδο μετασχηματισμού των σύγχρονων κυκλωμάτων σε ασύγχρονα, χρησιμοποιώντας εργαλεία συμβατά με την υπάρχουσα αυτοματοποιημένη ροή της βιομηχανίας. Τέτοια εργαλεία αποσυγχρονισμού είναι το *drdesync*, το οποίο παρουσιάστηκε στο [1], και το *Nanosync* της Nanochronous Logic.

Η παρούσα εργασία εφάρμοσε τη μέθοδο του αποσυγχρονισμού σε ορισμένα βιομηχανικά κυκλώματα, στο πλαίσιο μια υπάρχουσας βιομηχανικής ροής. Ο στόχοι ήταν:

- Η εξέταση και επίλυση των προβλημάτων που προκύπτουν, από τον αποσυγχρονισμό κυκλωμάτων μεγάλης πολυπλοκότητας.
- Η εξέταση της συμβατότητας των τεχνικών αποσυγχρονισμού με τις υπάρχουσες τεχνικές σχεδίασης βιομηχανικών κυκλωμάτων, και η κατάλληλη τροποποίησή τους όπου είναι απαραίτητο.
- Η εξέταση της συμβατότητας της μεθόδου, σε μια υπάρχουσα αυτοματοποιημένη βιομηχανική ροή.
- Η επιβεβαίωση των αναμενόμενων αποτελεσμάτων του αποσυγχρονισμού σε κυκλώματα βιομηχανικής κλίμακας, όσον αφορά την προσαρμοστικότητα, την κατανάλωση, την ηλεκτρομαγνητική παρεμβολή και το εμβαδόν.

Για τον αποσυγχρονισμό των συγκεκριμένων κυκλωμάτων έπρεπε να εξεταστούν ορισμένα προχωρημένα ζητήματα, τα οποία σχετίζονται με τη μέθοδο του αποσυγχρονισμού και δεν είχαν αναλυθεί από προηγούμενες εργασίες. Τα ζητήματα αυτά είναι τα εξής:

- Αντιμετώπιση της τεχνικής της φραγής ρολογιού από την μέθοδο του αποσυγχρονισμού.
- Αρχικοποίηση των αποσυγχρονισμένων κυκλωμάτων.
- Μέθοδος επικοινωνίας ενός αποσυγχρονισμένου κυκλώματος με σύγχρονα τμήματα.
- Μετατροπή του σύγχρονου testbench για τη χρήση του με το αποσυγχρονισμένο κύκλωμα.
- Μέθοδος επαλήθευσης της ορθής λειτουργίας ενός αποσυγχρονισμένου κυκλώματος.
- Χρονική βελτιστοποίηση του ασύγχρονου δικτύου ελέγχου των αποσυγχρονισμένων κυκλωμάτων.
- Εναλλακτικοί τρόποι δημιουργίας ομάδων αποσυγχρονισμού.

Τα κυκλώματα τα οποία εξετάστηκαν, ήταν δύο διαφορετικές εκδόσεις του επεξεργαστή IND1, της εταιρίας X_IND1_company_X. Οι εκδόσεις αυτές ήταν οι IND1a στα 0.13μm και IND1b στα 90nm. Επίσης, εξετάστηκε ο μικροελεγκτής IND2 στα 0.35μm, της εταιρίας X_Company_X. Σε όλες τις περιπτώσεις οι βιβλιοθήκες τεχνολογίας των κυκλωμάτων ήταν της X_Company_X. Η συγκεκριμένη εργασία πραγματοποιήθηκε στην X_Company_X, χρησιμοποιώντας την αυτοματοποιημένη ροή και τα εργαλεία της εταιρίας.

Η παρούσα αναφορά είναι οργανωμένη ως εξής: Στο Κεφάλαιο 2 παρουσιάζονται τα βασικά θέματα του αποσυγχρονισμού, ενώ στο Κεφάλαιο 3 εξετάζονται ορισμένα πιο προχωρημένα θέματα. Στο Κεφάλαιο 4 αναλύεται η ροή αποσυγχρονισμού, η οποία χρησιμοποιείται από τα εργαλεία *drdesync* και *Nanosync*. Στο Κεφάλαιο 5 παρατίθεται η συνεισφορά της εργασίας στην ανάπτυξη του *Nanosync*. Στο Κεφάλαιο 6 παρουσιάζεται η διαδικασία αποσυγχρονισμού των κυκλωμάτων, ενώ στο Κεφάλαιο 7 αναλύονται τα αποτελέσματα, τα οποία προέκυψαν. Στο Κεφάλαιο 8 παρουσιάζονται τα συμπεράσματα της εργασίας και εξετάζεται η πιθανή μελλοντική εργασία.

Κεφάλαιο 2

Αποσυγχρονισμός

Στο κεφάλαιο αυτό, παρουσιάζουμε τη βασική ιδέα του αποσυγχρονισμού και αναλύουμε τα κυριότερα βήματά του. Επίσης, εξετάζουμε ορισμένα θεμελιώδη θέματα, τα οποία αφορούν την υλοποίηση ενός αποσυγχρονισμένου κυκλώματος, όπως είναι το είδος των ασύγχρονων ελεγκτών και η υλοποίηση των στοιχείων καθυστέρησης.

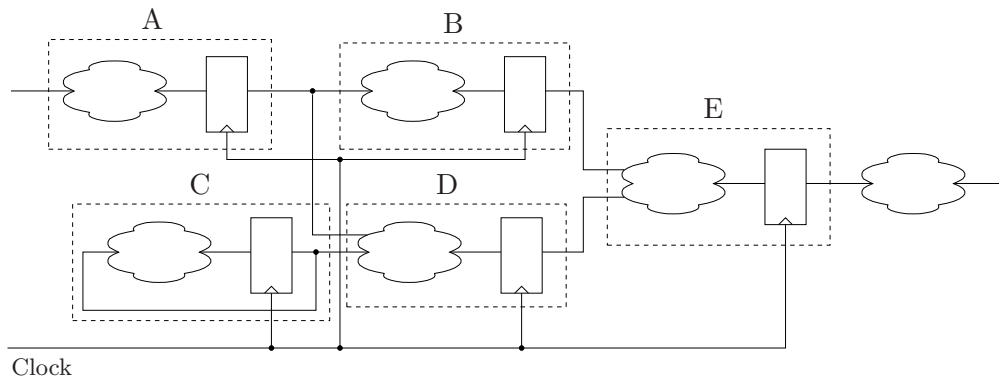
2.1 Βασική Ιδέα

Ο αποσυγχρονισμός είναι μια μέθοδος μετασχηματισμού ενός σύγχρονου κυκλώματος σε ένα αντίστοιχο ασύγχρονο [3]. Η θεμελιώδης ιδέα είναι η αντικατάσταση των γενικών δικτύων ρολογιού με ένα σύνολο ασύγχρονων ελεγκτών, χωρίς να μεταβάλλεται το τμήμα δεδομένων του κυκλώματος. Οι ελεγκτές αυτοί επικοινωνούν μεταξύ τους με σήματα χειραψίας.

Ο μετασχηματισμός αυτός διατηρεί μια ιδιότητα, η οποία ονομάζεται *ισοδυναμία ροής* [5, 2]. Σύμφωνα με αυτή, στις αντίστοιχες ακολουθιακές πύλες των δύο κυκλωμάτων, εμφανίζονται τα ίδια δεδομένα με την ίδια σειρά. Όμως, ο απόλυτος χρονισμός εμφάνισής τους μπορεί να διαφέρει.

Το Σχήμα 2.1 δείχνει ένα παράδειγμα σύγχρονου κυκλώματος. Τα ορθογώνια με τις διακεκομμένες γραμμές συμβολίζουν υπομονάδες, στις οποίες περιλαμβάνονται ένα σύνολο καταχωρητών καθώς και το τμήμα συνδυαστικής λογικής που τους τροφοδοτεί. Στα πλαίσια του αποσυγχρονισμού κάθε τέτοια ομάδα ονομάζεται *ομάδα αποσυγχρονισμού* και θα ελέγχεται από διαφορετικούς ασύγχρονους ελεγκτές. Ο τρόπος ανάθεσης των πυλών σε ομάδες αποσυγχρονισμού μπορεί να γίνει είτε χειροκίνητα, από τον ίδιο τον σχεδιαστή, είτε αυτοματοποιημένα, χρησιμοποιώντας κάποιον αλγόριθμο.

Η φάση του αποσυγχρονισμού σε ένα κύκλωμα πραγματοποιείται μετά από αυτές της σύνθεσης και του DFT (Design For Test), και πριν από τη φάση της τοποθέτησης και διασύνδεσης.



Σχήμα 2.1: Παράδειγμα σύγχρονου κυκλώματος.

2.2 Κύρια Βήματα Αποσυγχρονισμού

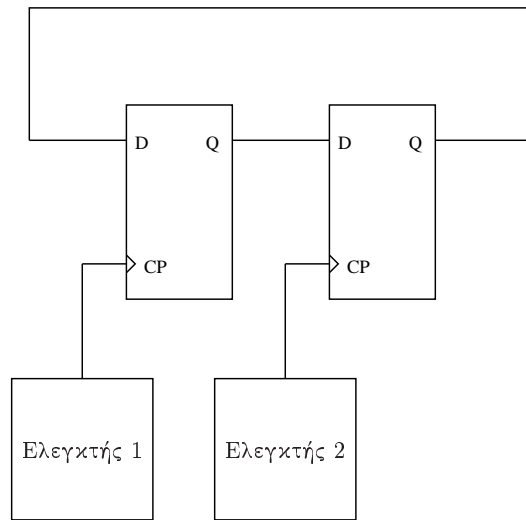
Σε αυτήν την ενότητα παρουσιάζονται τα κύρια βήματα που εκτελούνται κατά την διαδικασία του αποσυγχρονισμού.

2.2.1 Αντικατάσταση Καταχωρητών από Μανταλωτές

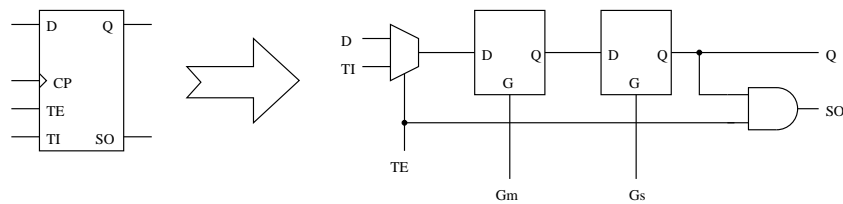
Οι καταχωρητές τύπου D αποτελούνται εσωτερικά από ένα ζεύγος μανταλωτών, τύπου Αφέντη και Σκλάβου. Τα δύο τοπικά σήματα ρολογιού, τα οποία ελέγχουν την πύλη των μανταλωτών, απορρέουν από το ίδιο εξωτερικό σήμα ρολογιού. Στον αποσυγχρονισμό είναι απαραίτητο να αποκαλυφθεί αυτή η εσωτερική δομή, έτσι ώστε:

1. Να είναι δυνατός ο ανεξάρτητος έλεγχος και των δύο τοπικών σημάτων ρολογιού. Για παράδειγμα, μια προβληματική περίπτωση, στην οποία δεν γίνεται αυτό, παρουσιάζεται στο Σχήμα 2.2. Σε αυτήν φαίνεται ότι, αν υπάρχει οποιαδήποτε χρονική απόκλιση στην πυροδότηση των δύο καταχωρητών από τους ελεγκτές, προκύπτει απώλεια δεδομένων.
2. Να υπάρχει η δυνατότητα βελτίωσης της απόδοσης, με την μετακίνηση ορισμένων μανταλωτών σε διαφορετική θέση επί της συνδυαστικής λογικής.

Σε ορισμένες βιβλιοθήκες τεχνολογίας υπάρχουν καταχωρητές, οι οποίοι δεν μπορούν να αντικατασταθούν από ανάλογους μανταλωτές. Αυτό συμβαίνει διότι δεν υπάρχουν μανταλωτές που να περιλαμβάνουν τις αντίστοιχες εισόδους και εξόδους τους. Για την σωστή αντικατάστασή αυτών των καταχωρητών πρέπει να χρησιμοποιηθούν επιπλέον ορισμένες πύλες συνδυαστικής λογικής. Μία τέτοια περίπτωση φαίνεται στο Σχήμα 2.3. Σε αυτό παρουσιάζεται ο τρόπος αντικατάστασης ενός καταχωρητή, ο οποίος υποστηρίζει αλυσίδα σάρωσης για κατασκευαστική δοκιμή.



Σχήμα 2.2: Παράδειγμα βρόχου καταχωρητών, όπου οποιαδήποτε χρονική απόκλιση δεν είναι ανεκτή.



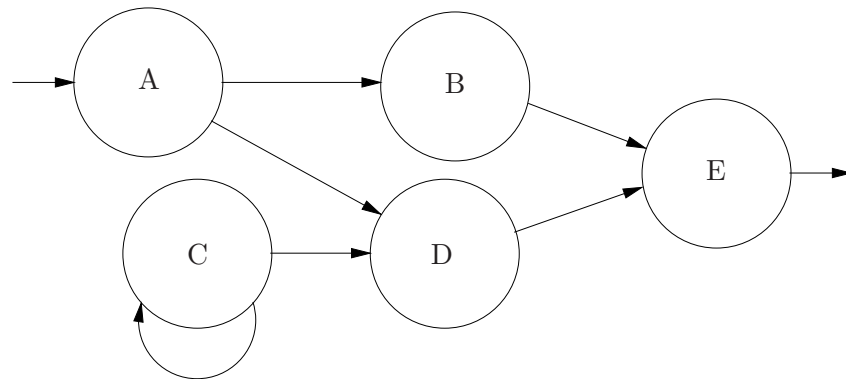
Σχήμα 2.3: Παράδειγμα αντικατάστασης καταχωρητή με υποστήριξη αλυσίδας σάρωσης για κατασκευαστική δοκιμή.

2.2.2 Ομαδοποίηση

Σε κάθε περίπτωση αποσυγχρονισμένου κυκλώματος υπάρχει η δυνατότητα ξεχωριστού ελέγχου του κάθε μανταλωτή από διαφορετικό ασύγχρονο ελεγκτή. Συνήθως όμως, η επιλογή είναι κάθε ελεγκτής να ελέγχει περισσότερους από έναν μανταλωτές, ώστε το κύκλωμα να έχει μικρότερη επιβάρυνση σε εμβαδόν. Για να επιτευχθεί αυτό θα πρέπει να γίνει ομαδοποίηση του κυκλώματος. Οι ομάδες που προκύπτουν ονομάζονται ομάδες αποσυγχρονισμού και ελέγχονται από διαφορετικό ζεύγος ασύγχρονων ελεγκτών: έναν ελεγκτή για όλους τους μανταλωτές Αφέντες και έναν για όλους τους μανταλωτές Σκλάβους. Επίσης, σε κάθε ομάδα συμπεριλαμβάνεται και η συνδυαστική λογική, η οποία τροφοδοτεί τους μανταλωτές.

Για κάθε διαφορετική ομαδοποίηση του κυκλώματος αντιστοιχεί διαφορετικός γράφος εξαρτήσεων των ομάδων. Ο γράφος αυτός παρουσιάζει τις εξαρτήσεις δεδομένων μεταξύ των διάφορων ομάδων αποσυγχρονισμού. Για κάθε μία τέτοια εξάρτηση θα πρέπει να δημιουργηθεί ένα στοιχείο καθυστέρησης, όπως εξηγείται στη συνέχεια. Για

παράδειγμα στην περίπτωση του Σχήματος 2.1, ο γράφος εξαρτήσεων των ομάδων φαίνεται στο Σχήμα 2.4.



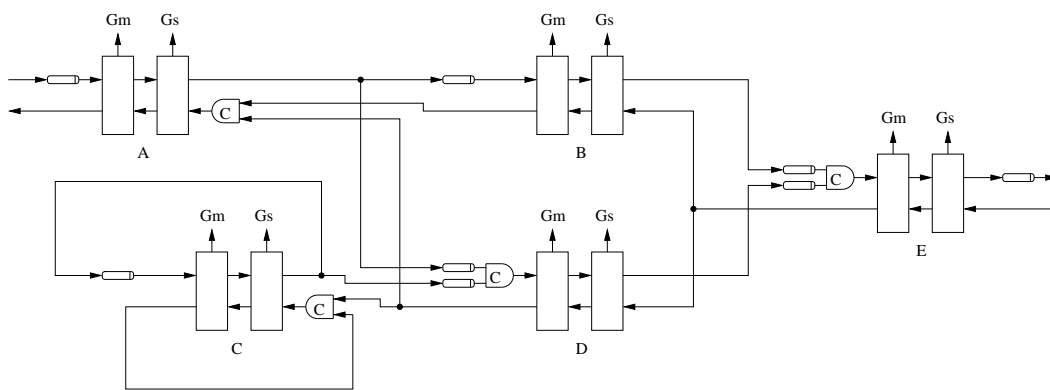
Σχήμα 2.4: Γράφος εξαρτήσεων των ομάδων του Σχήματος 2.1.

2.2.3 Εισαγωγή Δικτύου Ασύγχρονων Ελεγκτών

Από τον γράφο εξαρτήσεων των ομάδων προκύπτει το ασύγχρονο δίκτυο ελέγχου του κυκλώματος. Το δίκτυο αυτό ελέγχει τους μανταλωτές και κατά συνέπεια ορίζει την ορθή ροή δεδομένων στο κύκλωμα. Για κάθε ομάδα αποσυγχρονισμού εισάγεται ένα ζεύγος ασύγχρονων ελεγκτών. Ο πρώτος ελέγχει το άνοιγμα και κλείσιμο των μανταλωτών τύπου Αφέντη, ενώ ο δεύτερος ελέγχει το άνοιγμα και κλείσιμο των μανταλωτών τύπου Σκλάβου.

Η επικοινωνία κάθε ελεγκτή, τόσο με τους ελεγκτές που έπονται, όσο και με τους ελεγκτές που προηγούνται, γίνεται μέσω δύο σημάτων χειραψίας. Τα σήματα αυτά είναι το σήμα Αίτησης και το σήμα Αναγνώρισης. Το πρώτο δίνεται από κάποιον ελεγκτή στους επόμενους και υποδηλώνει ότι οι μανταλωτές του έχουν νέα δεδομένα. Αντίστοιχα το δεύτερο δίνεται από κάποιον ελεγκτή στους προηγούμενους και υποδηλώνει ότι οι μανταλωτές του παρέλαβαν τα δεδομένα που τους δόθηκαν. Οι συνδέσεις όλων των σημάτων χειραψίας γίνονται με ανάλογο τρόπο, ώστε να ακολουθούν τις εξαρτήσεις δεδομένων. Σε περίπτωση πολλαπλών σημάτων Αίτησης ή Αναγνώρισης που καταλήγουν σε έναν ελεγκτή, αυτά συγχωνεύονται με τη χρήση ενός στοιχείου C.

Για το παράδειγμα του Σχήματος 2.1, θα δημιουργηθεί το ασύγχρονο δίκτυο ελέγχου του Σχήματος 2.5. Στο δίκτυο χρησιμοποιούνται στοιχεία καθυστέρησης μεταξύ των ελεγκτών, τα οποία συμβολίζονται με οβάλ σχήμα και μία κάθετη γραμμή στη μία τους άκρη. Αυτά μιμούνται την καθυστέρηση των αντίστοιχων τμημάτων της συνδυαστικής λογικής.



Σχήμα 2.5: Ασύγχρονο δίκτυο ελέγχου του Σχήματος 2.1.

2.2.4 Εισαγωγή Στοιχείων Καθυστέρησης

Κάθε ακμή του γράφου εξαρτήσεων των ομάδων λαμβάνεται υπόψιν για την δημιουργία ενός στοιχείου καθυστέρησης. Για παράδειγμα, για το κύκλωμα του Σχήματος 2.1, θα δημιουργηθούν ακριβώς οχτώ στοιχεία καθυστέρησης, όσες είναι και οι ακμές του γράφου, του Σχήματος 2.4.

Το σήμα Αίτησης των ελεγκτών δίνεται όταν οι μανταλωτές τους έχουν νέα δεδομένα. Ο σκοπός, λοιπόν ενός στοιχείου καθυστέρησης είναι να καθυστερήσει το σήμα Αίτησης τόσο, όσο χρειάζεται η αντίστοιχη συνδυαστική λογική, για να υπολογίσει τα δεδομένα της. Για να επιτευχθεί αυτό θα πρέπει η καθυστέρηση, η οποία εισάγεται από κάθε τέτοιο στοιχείο, να είναι ίση ή μεγαλύτερη από την καθυστέρηση του κρίσιμου μονοπατιού της αντίστοιχης συνδυαστικής λογικής.

2.3 Στοιχεία C

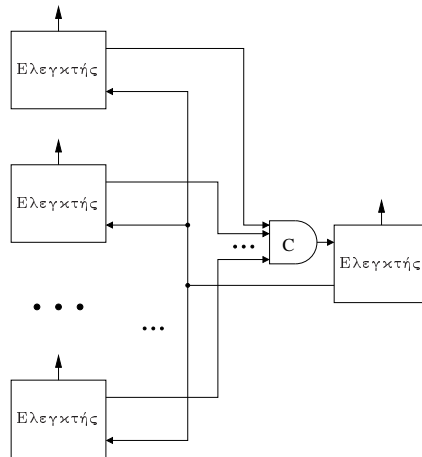
Τα στοιχεία C [7] είναι λογικές πύλες, οι οποίες χρησιμοποιούνται για τον συγχρονισμό δύο ή περισσότερων σημάτων Αίτησης, ή δύο ή περισσότερων σημάτων Αναγνώρισης. Ένα στοιχείο C περιμένει να γίνουν όλες οι εισοδοί του 0 για να δώσει στην έξοδό του 0 και αντίστοιχα περιμένει να γίνουν όλοι οι εισοδοί του 1 για να δώσει στην έξοδό του 1. Ο πίνακας αληθείας του παρουσιάζεται στον Πίνακα 2.1. Τέλος, στα Σχήματα 2.6 και 2.7, παρουσιάζονται οι τρόποι σύνδεσης των ασύγχρονων ελεγκτών με την χρήση στοιχείων C.

2.4 Ασύγχρονοι Ελεγκτές

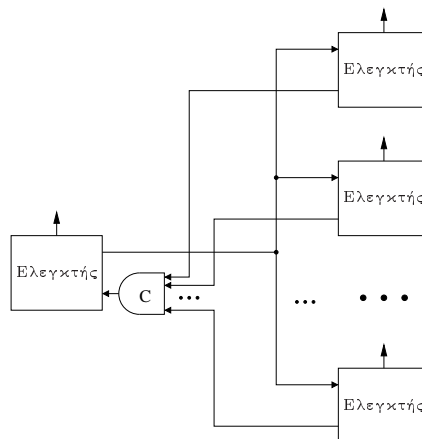
Η κύρια μονάδα, η οποία χρησιμοποιείται για την κατασκευή του ασύγχρονου δικτύου ελέγχου, είναι ο ασύγχρονος ελεγκτής. Ένα ζεύγος ασύγχρονων ελεγκτών είναι υ-

Είσοδοι	Έξοδος
Όλες 0	0
Όλες 1	1
Κάποιες 0, κάποιες 1	Αμετάβλητη

Πίνακας 2.1: Πίνακας αληθείας στοιχείου C.

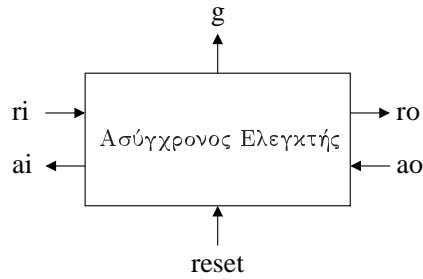


Σχήμα 2.6: Τρόπος σύνδεσης πολλαπλών ασύγχρονων ελεγκτών σε έναν. Για την σύνδεση των σημάτων Αίτησης χρησιμοποιείται ένα στοιχείο C.



Σχήμα 2.7: Τρόπος σύνδεσης ενός ασύγχρονου ελεγκτή σε πολλαπλούς. Για την σύνδεση των σημάτων Αναγνώρισης χρησιμοποιείται ένα στοιχείο C.

πεύθυνο για το άνοιγμα και κλείσιμο των μανταλωτών, τύπου Αφέντη και Σκλάβου, σε κάθε ομάδα αποσυγχρονισμού. Το διάγραμμα εισόδων–εξόδων ενός ασύγχρονου ελεγκτή φαίνεται στο Σχήμα 2.8.



Σχήμα 2.8: Διάγραμμα εισόδων-εξόδων Ασύγχρονου Ελεγκτή.

2.4.1 Γενική Περιγραφή Σημάτων

Στο Σχήμα παρουσιάζονται έξι διαφορετικά σήματα: οι εισοδοί ri (input request), ao (output acknowledge) και $reset$, και οι εξοδοί ro (output request), ai (input acknowledge) και g (πύλη των μανταλωτών).

Το $reset$ χρησιμοποιείται για την αρχικοποίηση του ελεγκτή. Τα ri και ai αποτελούν τα σήματα χειραψίας με τους προηγούμενους ελεγκτές στο ασύγχρονο δίκτυο ελέγχου. Το ri καταδεικνύει ότι τα δεδομένα που υπάρχουν στις εισόδους των μανταλωτών του ελεγκτή είναι έγκυρα, οπότε μπορούν να αποθηκευτούν. Αντίστοιχα, το ai καταδεικνύει ότι τα δεδομένα αυτά αποθηκεύτηκαν, οπότε μπορεί να γίνει αποστολή νέων. Τα σήματα ro και ao αποτελούν τα σήματα χειραψίας με τους επόμενους ελεγκτές στο ασύγχρονο δίκτυο ελέγχου και έχουν ανάλογη συμπεριφορά με τα ri και ai . Συγκεκριμένα το ro καταδεικνύει ότι οι μανταλωτές του ελεγκτή έχουν νέα δεδομένα. Αντίστοιχα το ao καταδεικνύει ότι τα δεδομένα αυτά αποθηκεύτηκαν από τους επόμενους μανταλωτές, οπότε οι μανταλωτές του ελεγκτή είναι ελεύθεροι να αποθηκεύσουν νέα δεδομένα. Τέλος, το σήμα g ελέγχει την πύλη των μανταλωτών, την οποία ανοίγει ή κλείνει ο ελεγκτής ανάλογα με την κατάσταση της ροής δεδομένων.

2.4.2 Πρωτόκολλα Χειραψίας και Είδη Ασύγχρονων Ελεγκτών

Υπάρχουν διαφορετικά είδη ασύγχρονων ελεγκτών, ανάλογα με το πρωτόκολλο χειραψίας που υλοποιούν. Τα πρωτόκολλα χειραψίας περιλαμβάνουν διάφορες κατηγορίες, όπως *semi-decoupled*, *fully-decoupled*, κτλ. [4, 2]. Αυτές διαφέρουν ως προς τον συνολικό αριθμό διαφορετικών καταστάσεων που έχουν και ως προς τον βαθμό ελευθερίας που υπάρχει, μεταξύ των σημάτων χειραψίας εισόδου και εξόδου. Η βασική τους όμως κατηγοριοποίηση αφορά σε πρωτόκολλα δύο και τεσσάρων φάσεων, με τους αντίστοιχους ελεγκτές να ονομάζονται *διφασικοί* και *τετραφασικοί* αντίστοιχα.

Στο πρωτόκολλα δύο φάσεων [10], τόσο η θετική όσο και η αρνητική ακμή του σήματος της Αίτησης σηματοδοτούν τη διαθεσιμότητα νέων δεδομένων. Αντίστοιχα,

τόσο η θετική όσο και η αρνητική ακμή του σήματος της Αναγνώρισης σηματοδοτούν το τέλος της επεξεργασίας των δεδομένων και τη δυνατότητα έναρξης νέας χειραψίας. Αντίθετα, στα πρωτόκολλα τεσσάρων φάσεων [2], η σηματοδότηση των αντίστοιχων γεγονότων γίνεται μόνο με τη μία από τις δύο ακμές των σημάτων. Αυτό σημαίνει ότι η άλλη ακμή είναι ανενεργή και αποτελεί απλά μέρος της φάσης ανάκτησης, για την έναρξη νέας χειραψίας.

Στην συνέχεια αυτής της ενότητας παρουσιάζονται οι δύο ελεγκτές που χρησιμοποιήθηκαν στον αποσυγχρονισμό των κυκλωμάτων του Κεφαλαίου 6. Ο ένας είναι τετραφασικός semi-decoupled, και ο άλλος διφασικός.

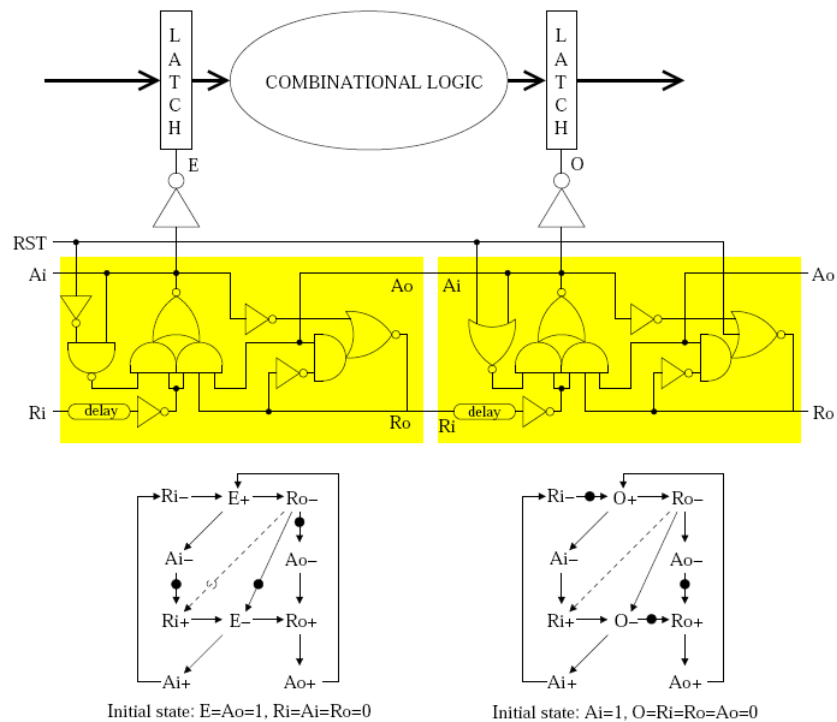
2.4.3 Ένας Τετραφασικός Ασύγχρονος Ελεγκτής

Το ζεύγος ελεγκτών, το οποίο χρησιμοποιήθηκε στον αποσυγχρονισμό του επεξεργαστή IND1a, φαίνεται στο Σχήμα 2.9. Οι μανταλωτές στο συγκεκριμένο Σχήμα (οι οποίοι αναφέρονται με το όνομα *LATCH*) θεωρούνται ότι είναι διαφανείς (transparent) στο 1 και αδιαφανείς (opaque) στο 0. Η λειτουργία του συγκεκριμένου ελεγκτή εξετάζεται αναλυτικά στο [2]. Στο Σχήμα παρουσιάζονται τόσο η έκδοση η οποία ελέγχει τους μανταλωτές Αφέντες (*E – even – ζυγοί*), όσο και η έκδοση η οποία ελέγχει τους μανταλωτές Σκλάβους (*O – odd – μονοί*). Συγκεκριμένα, δίνονται οι Γράφοι Μετάβασης Σήματος (ΓΜΣ), καθώς και η υλοποίησή τους με πύλες τεχνολογίας. Ο ίδιος ελεγκτής χρησιμοποιήθηκε και σε παλαιότερες εργασίες, για τον αποσυγχρονισμό του επεξεργαστή DLX [1, 3].

Όπως φαίνεται από τον ΓΜΣ, το πρωτόκολλο χειραψίας που υλοποιεί ο συγκεκριμένος ελεγκτής είναι τεσσάρων φάσεων. Επίσης, παρατηρούμε ότι όταν εφαρμόζεται το *reset*, οι ελεγκτές αρχικοποιούνται με τέτοιο τρόπο, ώστε οι μανταλωτές Αφέντες να είναι διαφανείς και οι μανταλωτές Σκλάβοι να είναι αδιαφανείς. Αυτό σημαίνει ότι μόνο οι μανταλωτές Σκλάβοι χρειάζεται να αρχικοποιούνται σε κάποια αρχική κατάσταση. Να σημειωθεί ότι οι αρχικές καταστάσεις των ελεγκτών συμβολίζονται με τα σύμβολα (tokens) στον ΓΜΣ. Τέλος, το *delay* το οποίο υπάρχει στο input request, είναι στοιχείο καθυστέρησης ασύμμετρου τύπου και εξαρτάται από την καθυστέρηση που έχει το κρίσιμο μονοπάτι του *COMBINATIONAL LOGIC*.

2.4.4 Ένας Διφασικός Ασύγχρονος Ελεγκτής

Το ζεύγος ελεγκτών, το οποίο χρησιμοποιήθηκε για τον αποσυγχρονισμό των κυκλωμάτων IND1b και IND2, φαίνεται στο Σχήμα Γ.1. Επίσης, στο Σχήμα Γ.2 παρουσιάζεται ο ΓΜΣ τους.



Σχήμα 2.9: Ζεύγος τετραφασικών ασύγχρονων ελεγκτών για μανταλωτές Αφέντη και Σκλάβου.

2.5 Στοιχεία Καθυστέρησης

Τα στοιχεία καθυστέρησης χρησιμοποιούνται για την μοντελοποίηση της καθυστέρησης της συνδυαστικής λογικής του κυκλώματος. Ο σκοπός τους είναι να καθυστερήσουν το σήμα Αίτησης των ασύγχρονων ελεγκτών τόσο, όσο χρειάζεται η αντίστοιχη συνδυαστική λογική για να υπολογίσει τις τιμές της.

Η δημιουργία των στοιχείων καθυστέρησης πραγματοποιείται μετά την ομαδοποίηση του κυκλώματος. Επειδή κάθε εξάρτηση δεδομένων μεταξύ των ομάδων αντιστοιχεί σε ένα στοιχείο καθυστέρησης, θα πρέπει αρχικά να βρεθούν όλες οι εξαρτήσεις των ομάδων. Ο τρόπος εύρεσής τους εξηγείται παρακάτω. Επίσης, εξηγείται ο τρόπος υπολογισμού της καθυστέρησης των στοιχείων και περιγράφονται οι διαφορετικοί τύποι τους.

2.5.1 Εύρεση Εξαρτήσεων των Ομάδων

Ο κανόνας εξάρτησης είναι ο εξής: δύο ομάδες αποσυγχρονισμού A και B , έχουν εξάρτηση $A \rightarrow B$, αν υπάρχει μονοπάτι συνδυαστικής λογικής από την έξοδο καταχωρητή της A σε είσοδο καταχωρητή της B .

Ο προηγούμενος ορισμός της εξάρτησης ισχύει και στις περιπτώσεις, στις οποίες

μία ομάδα έχει εξάρτηση με τον εαυτό της. Επίσης, ισχύει και στις περιπτώσεις, όπου μια ομάδα έχει εξάρτηση με εισόδους ή τις εξόδους του κυκλώματος. Μια και οι ομάδες εισόδου και εξόδου δεν έχουν καταχωρητές, για την εύρεση αυτών των εξαρτήσεων χρησιμοποιούνται οι ίδιοι οι εισόδοι και έξοδοι του κυκλώματος, ως ειδικού τύπου “καταχωρητές”.

Γενικά ισχύει ότι το μέγιστο δυνατό πλήθος εξαρτήσεων, για n ομάδες αποσυγχρονισμού, είναι:

$$dependencies_{max} = n^2 + 2n \quad (2.1)$$

Αυτό διότι:

- Η είσοδος μπορεί να έχει το πολύ n εξαρτήσεις προς τις n διαφορετικές ομάδες.
- Η έξοδος μπορεί να έχει το πολύ n εξαρτήσεις από τις n διαφορετικές ομάδες.
- Κάθε ομάδα ενδέχεται να έχει μία εξάρτηση με τον εαυτό της και μπορεί να έχει το πολύ $n - 1$ εξαρτήσεις προς τις $n - 1$ άλλες ομάδες.

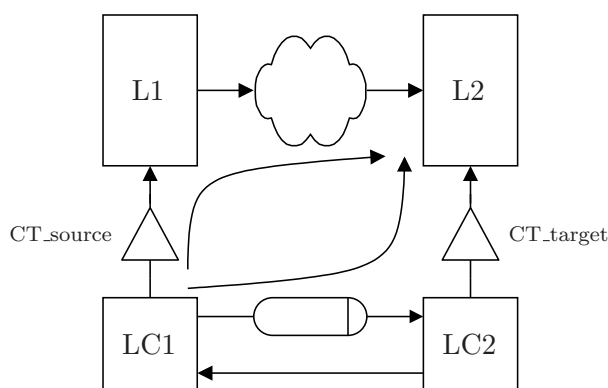
Επίσης, αν ληφθεί υπόψη και η πιθανή εξάρτηση της εισόδου προς την έξοδο, τότε η Σχέση 2.1 μεταβάλλεται σε: $dependencies_{max} = n^2 + 2n + 1$. Παρόλα αυτά, η συγκεκριμένη εξάρτηση δεν λαμβάνεται συνήθως υπόψη για την κατασκευή στοιχείου καθυστέρησης. Η τήρηση του χρονισμού της θεωρείται ότι είναι ευθύνη του ίδιου του σχεδιαστή.

2.5.2 Υπολογισμός Καθυστέρησης

Στο Σχήμα 2.10 παρουσιάζονται οι καθυστερήσεις, οι οποίες λαμβάνονται υπόψη για τη δημιουργία ενός στοιχείου καθυστέρησης. Τα CT_{source} και CT_{target} αντιστοιχούν σε τοπικά δέντρα από ενισχυτές, τα οποία εισάγονται στη φάση της τοποθέτησης και διασύνδεσης. Αν CL_{max_delay} είναι η καθυστέρηση του κρίσιμου μονοπατιού της συνδυαστικής λογικής και Req_{delay} είναι η καθυστέρηση, η οποία προκαλείται στη μετάδοση του σήματος Αίτησης από το στοιχείο καθυστέρησης, θα πρέπει να ισχύει $Req_{delay} + CT_{target} \geq CT_{source} + CL_{max_delay}$. Επίσης, αν υποθέσουμε ότι τα τοπικά δέντρα από ενισχυτές έχουν κατασκευαστεί με τέτοιο τρόπο ώστε $CT_{target} = CT_{source}$, τότε η προηγούμενη σχέση μετατρέπεται σε:

$$Req_{delay} \geq CL_{max_delay} \quad (2.2)$$

Η καθυστέρηση CL_{max_delay} μπορεί να υπολογιστεί με κάποιο εργαλείο Στατικής Χρονικής Ανάλυσης (ΣΧΑ). Επίσης, πρέπει να σημειωθεί ότι η καθυστέρηση Req_{delay} ενδέχεται να περιλαμβάνει, εκτός από αυτήν του στοιχείου καθυστέρησης, και την καθυστέρηση ενός πιθανού στοιχείου C, το οποίο χρησιμοποιείται για την συγχώνευση πολλαπλών σημάτων Αίτησης.

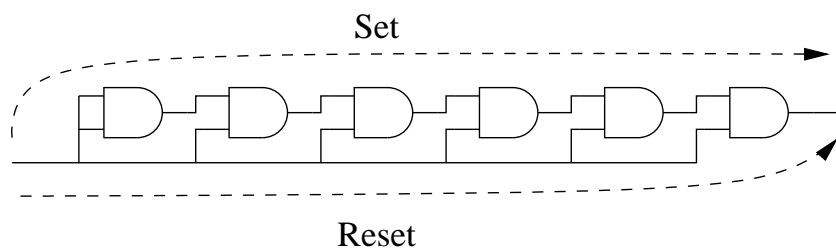


Σχήμα 2.10: Υπολογισμός στοιχείου καθυστέρησης.

2.5.3 Ασύμμετρα Στοιχεία Καθυστέρησης

Στην περίπτωση που το ασύγχρονο δίκτυο ελέγχου χρησιμοποιεί τετραφασικούς ασύγχρονους ελεγκτές, τότε τα στοιχεία καθυστέρησης πρέπει να είναι ασύμμετρου τύπου. Με αυτόν τον τρόπο καθυστερείται μόνο η άνοδος του σήματος της Αίτησης και όχι η επιστροφή στο 0.

Ένα παράδειγμα τέτοιου στοιχείου καθυστέρησης φαίνεται στο Σχήμα 2.11. Η άνοδος του σήματος Αίτησης αντιστοιχεί στο set ενώ η επιστροφή στο 0 αντιστοιχεί στο reset. Η καθυστέρηση του set καθορίζεται από την Σχέση 2.2, ενώ η καθυστέρηση του reset είναι όσο μικρότερη γίνεται. Με ανάλογο τρόπο, μπορούν να κατασκευαστούν στοιχεία καθυστέρησης ασύμμετρου τύπου, χρησιμοποιώντας πύλες οποιουδήποτε τύπου.

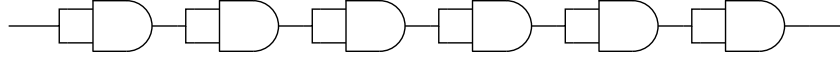


Σχήμα 2.11: Παράδειγμα ασύμμετρου στοιχείου καθυστέρησης.

2.5.4 Συμμετρικά Στοιχεία Καθυστέρησης

Στην περίπτωση που το ασύγχρονο δίκτυο ελέγχου χρησιμοποιεί διφασικούς ασύγχρονους ελεγκτές, τότε τα στοιχεία καθυστέρησης πρέπει να είναι συμμετρικού τύπου. Με αυτόν τον τρόπο καθυστερείτε τόσο η άνοδος, όσο και η κάθοδος του σήματος

Αίτησης. Τα συμμετρικά στοιχεία καθυστέρησης μπορούν να κατασκευαστούν με διάφορες πύλες, όπως π.χ. με αντιστροφείς, ενισχυτές, πύλες AND κτλ. Ένα παράδειγμα συμμετρικού στοιχείου καθυστέρησης από πύλες AND, παρουσιάζεται στο Σχήμα 2.12.



Σχήμα 2.12: Παράδειγμα συμμετρικού στοιχείου καθυστέρησης.

Ο τρόπος κατασκευής των συμμετρικών στοιχείων καθυστέρησης εξετάζεται στο Παράρτημα Γ'.2.

2.5.5 Χρονικό Περιθώριο και Χρήση Πολυπλέκτη

Όπως αναφέραμε στην Ενότητα 2.5.2, κάθε στοιχείο καθυστέρησης πρέπει να είναι τέτοιο, ώστε να ισχύει $Req_{delay} \geq CL_{max_delay}$. Εξαιτίας όμως του προβλήματος της μεταβλητότητας, θα πρέπει να χρησιμοποιηθεί ένα χρονικό περιθώριο ασφαλείας. Αυτό είναι συνήθως γύρω στο 20-30% της καθυστέρησης CL_{max_delay} για τυπικές συνθήκες, τάσης, θερμοκρασίας και διεργασίας. Οπότε, στην πράξη, για την κατασκευή αποσυγχρονισμένων κυκλωμάτων, ισχύει η ακόλουθη σχέση:

$$Req_{delay} = (1 + margin) \times CL_{max_delay} \quad (2.3)$$

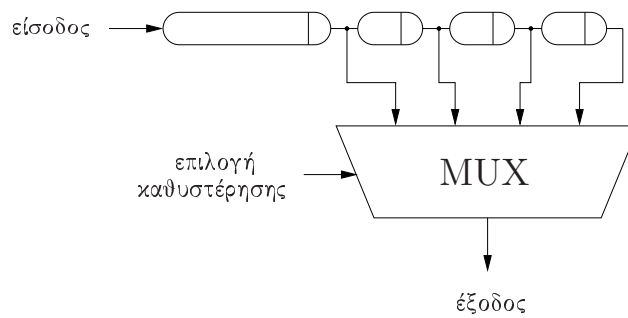
Το *margin* είναι το χρονικό περιθώριο ασφαλείας. Για παράδειγμα, στην περίπτωση του ASPIDA [3] το *margin* ήταν γύρω στο 20%. Επίσης, το εργαλείο αποσυγχρονισμού *Nanosync* χρησιμοποιεί $margin \geq 30\%$.

Αν θέλουμε να έχουμε τη δυνατότητα να διαπιστώσουμε το ελάχιστο περιθώριο των στοιχείων καθυστέρησης, στο οποίο μπορεί να λειτουργήσει ένα κύκλωμα μετά την κατασκευή του, τότε μπορούμε να χρησιμοποιήσουμε πολυπλέκτες. Οι πολυπλέκτες αυτοί επιλέγουν το τελικό μήκος των στοιχείων καθυστέρησης που χρησιμοποιείται στο κύκλωμα μέσω σημάτων ελέγχου. Η γενική εικόνα των στοιχείων καθυστέρησης με πολυπλέκτη φαίνεται στο Σχήμα 2.13.

2.6 Ισοδυναμία Ροής

Δύο κυκλώματα έχουν ισοδυναμία ροής [5] εάν:

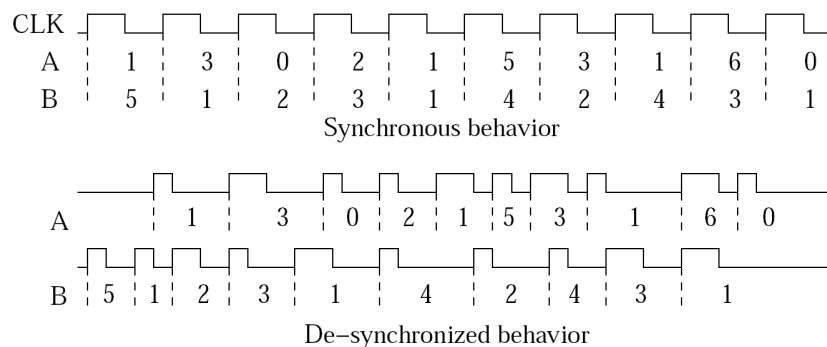
1. έχουν τον ίδιο αριθμό από μανταλωτές και
2. για κάθε αντίστοιχο μανταλωτή, η αλληλουχία δεδομένων είναι η ίδια.



Σχήμα 2.13: Στοιχείο καθυστέρησης με πολυπλέκτη.

Όπως αποδείχτηκε στο [2], κάθε αποσυγχρονισμένο κύκλωμα διατηρεί ισοδυναμία ροής με το αντίστοιχο σύγχρονό του. Αυτό συμβαίνει διότι, όλοι οι γειτονικοί μανταλωτές του αποσυγχρονισμένου κυκλώματος εναλλάσσουν τους παλμούς τους με σωστό τρόπο, ώστε να ανταλλάσσουν τα ίδια δεδομένα, τα οποία εμφανίζονται και στο σύγχρονο κύκλωμα.

Στο Σχήμα 2.14 παρουσιάζεται ένα υποθετικό παράδειγμα αλληλουχίας δεδομένων σε δύο μανταλωτές, τόσο σε ένα σύγχρονο κύκλωμα όσο και στο αντίστοιχο αποσυγχρονισμένο. Να σημειωθεί ότι, στο συγκεκριμένο παράδειγμα, οι μανταλωτές αυτοί δεν ανήκουν στην ίδια ομάδα αποσυγχρονισμού, ούτε σε γειτονικές. Όπως παρατηρούμε, τα δεδομένα των μανταλωτών στον i -οστό κύκλο ρολογιού του σύγχρονου κυκλώματος είναι τα ίδια με τα δεδομένα των μανταλωτών στους αντίστοιχους i -οστούς παλμούς του αποσυγχρονισμένου κυκλώματος. Όμως, η χρονική στιγμή στην οποία εμφανίζονται τα δεδομένα κάθε ομάδας στο αποσυγχρονισμένο κύκλωμα, μπορεί να είναι διαφορετική από αυτή στην οποία εμφανίζονται στο σύγχρονο, καθώς εξαρτάται από την ροή των δεδομένων στο κύκλωμα.



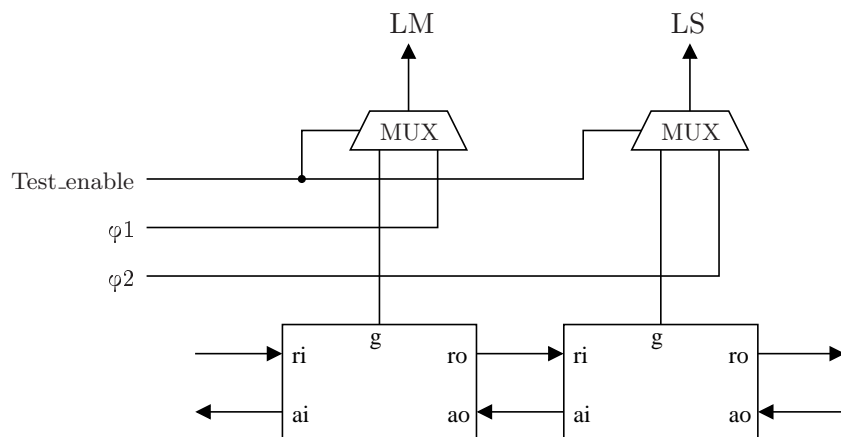
Σχήμα 2.14: Ισοδυναμία ροής.

2.7 Δοκιμή Αποσυγχρονισμένων Κυκλωμάτων

Οι ίδιες τεχνικές DFT (Design For Test), οι οποίες χρησιμοποιούνται κατά την κατασκευή των σύγχρονων κυκλωμάτων, μπορούν να χρησιμοποιηθούν και για την περίπτωση των αποσυγχρονισμένων. Παρακάτω παρουσιάζουμε τις περιπτώσεις αλυσίδας σάρωσης για κατασκευαστική δοκιμή και BIST (Built-In Self-Test).

2.7.1 Αλυσίδες Σάρωσης

Η μέθοδος του αποσυγχρονισμού χρησιμοποιεί τις υπάρχουσες συνδέσεις αλυσίδας σάρωσης των καταχωρητών, οι οποίες υπάρχουν στο σύγχρονο κύκλωμα. Στην Ενότητα 2.2.1 και στο Σχήμα 2.3 είδαμε τον τρόπο αντικατάστασης των καταχωρητών, οι οποίοι υποστηρίζουν αλυσίδα σάρωσης. Επίσης, το αποσυγχρονισμένο κύκλωμα χρησιμοποιεί τα ίδια διανύσματα, τα οποία παρήχθησαν κατά το ATPG (Automatic Test Pattern Generation), για την εξακρίβωση σφαλμάτων στο σύγχρονο κύκλωμα. Κατά την εισαγωγή και εξαγωγή των διανυσμάτων, οι μανταλωτές ελέγχονται από δύο μη επικαλυπτόμενα γενικά σήματα ρολογιού. Το ένα από αυτά ελέγχει όλους τους μανταλωτές Αφέντες του κυκλώματος, ενώ το άλλο ελέγχει όλους τους μανταλωτές Σκλάβους. Για τον λόγο αυτό, χρησιμοποιείται ένας επιπλέον πολυπλέκτης σε κάθε ασύγχρονο ελεγκτή, όπως φαίνεται στο Σχήμα 2.15.



Σχήμα 2.15: Υποστήριξη ελέγχου με αλυσίδα σάρωσης για τους ασύγχρονους ελεγκτές.

2.7.2 Built-In Self-Test

Όσον αφορά τα κυκλώματα BIST (Built-In Self-Test), αυτά δεν χρειάζεται να αποσυγχρονιστούν, διότι δεν είναι ενεργά όταν το κύκλωμα βρίσκεται σε κατάσταση λειτουργίας. Γενικά, αν αποσυγχρονιστούν δεν προκύπτει σημαντικό κέρδος (ενώ υπάρχει το

κόστος του επιπλέον εμβαδού).

Κεφάλαιο 3

Προχωρημένα Θέματα Αποσυγχρονισμού

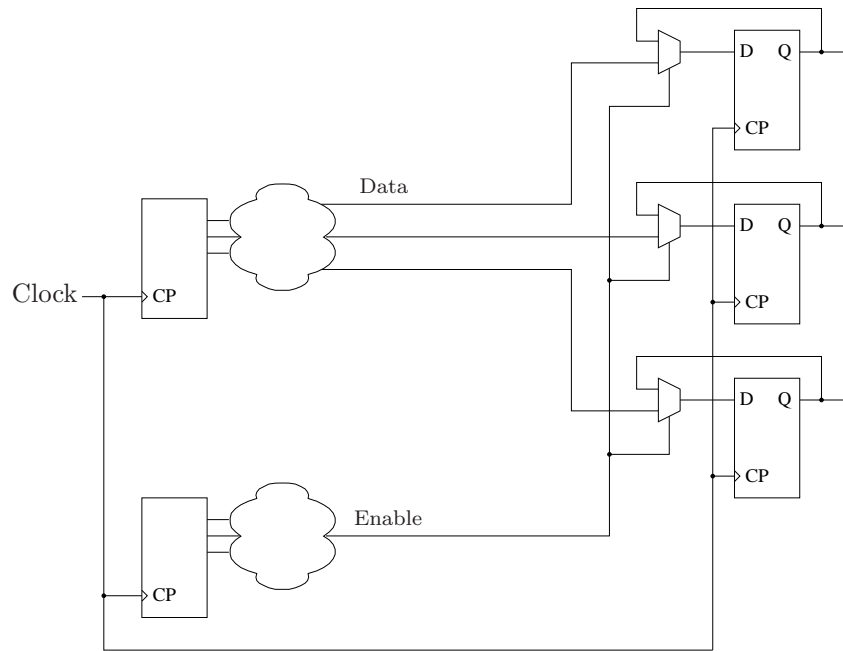
Στο προηγούμενο κεφάλαιο, παρουσιάσαμε ορισμένα θεμελιώδη θέματα του αποσυγχρονισμού, τα περισσότερα εκ των οποίων έχουν ήδη εξεταστεί από την υπάρχουσα βιβλιογραφία. Στο παρών κεφάλαιο, καταπιανόμαστε με πιο προχωρημένα θέματα, τα οποία απαιτούν βαθύτερη κατανόηση του τρόπου λειτουργίας των ασύγχρονων κυκλωμάτων και των ιδιοτήτων του αποσυγχρονισμού, και παρουσιάζονται στη βιβλιογραφία για πρώτη φορά. Αυτά προέκυψαν από την εργασία αποσυγχρονισμού των τριών βιομηχανικών κυκλωμάτων, η οποία παρουσιάζεται αναλυτικά στο Κεφάλαιο 6.

3.1 Φραγή Σήματος Ρολογιού

Η τεχνική της φραγής ρολογιού εφαρμόζεται στα σύγχρονα κυκλώματα, με στόχο την μείωση της κατανάλωσης ενέργειας. Η μείωση αυτή επιτυγχάνεται με την απενεργοποίηση της δραστηριότητας σε διάφορες ομάδες καταχωρητών, όταν αυτοί διατηρούν τα δεδομένα τους από κύκλο σε κύκλο ρολογιού.

Η φραγή ρολογιού εφαρμόζεται σε καταχωρητές, οι οποίοι μοιράζονται το ίδιο ρολόι και ελέγχονται από το ίδιο σήμα *Enable*. Αν δεν χρησιμοποιηθεί φραγή ρολογιού, τότε θα πρέπει μπροστά από τους καταχωρητές να χρησιμοποιηθούν πολυπλέκτες, οι οποίοι θα τους τροφοδοτούν με την υπάρχουσα τιμή που έχουν ή με κάποια καινούρια, ανάλογα με την τιμή που έχει το *Enable* [11]. Ένα παράδειγμα αυτής της περίπτωσης φαίνεται στο Σχήμα 3.1, στο οποίο παρουσιάζονται τρεις καταχωρητές, οι οποίοι τροφοδοτούνται από τρεις αντίστοιχους πολυπλέκτες. Με την συγκεκριμένη όμως διάταξη, όταν το *Enable* είναι 0, θα καταναλώνεται άσκοπα ενέργεια, μιας και οι καταχωρητές θα φορτώνουν σε κάθε ακμή του ρολογιού την ίδια τιμή που είχαν και πριν. Η ενέργεια αυτή θα καταναλώνεται όχι μόνο στους καταχωρητές, αλλά και στο υποδίκτυο ρολογιού που

τους ελέγχει, όπως και στους πολυπλέκτες.



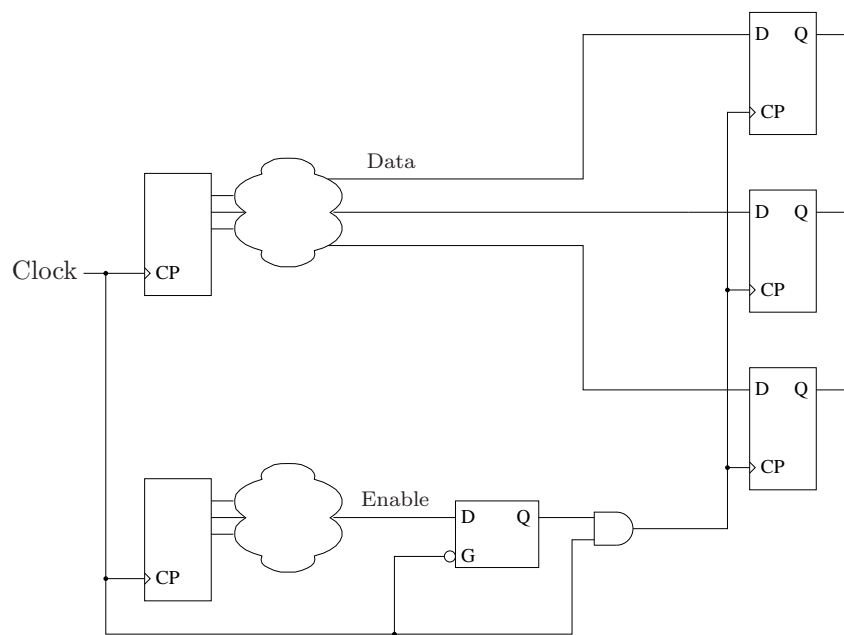
Σχήμα 3.1: Υλοποίηση του *Enable* με πολυπλέκτες.

Η εναλλακτική λύση του κυκλώματος του Σχήματος 3.1, με χρήση φραγής ρολογιού, παρουσιάζεται στο Σχήμα 3.2. Εδώ, οι καταχωρητές δεν ξαναφορτώνουν την παλιά τους τιμή όταν το *Enable* είναι 0, αλλά μένουν ανενεργοί. Με αυτόν τον τρόπο, όχι μόνο μειώνεται η κατανάλωση ενέργειας, αλλά δεν χρησιμοποιούνται και οι πολυπλέκτες. Στη θέση τους αρκεί μια πύλη AND (ή πύλη OR αν οι καταχωρητές πυροδοτούνται στην αρνητική ακμή του ρολογιού) και ένας μανταλωτής για τον έλεγχο όλων των καταχωρητών. Όπως είναι λογικό, όσο πιο πολλοί καταχωρητές ελέγχονται από το ίδιο *Enable*, τόσο προτιμότερη γίνεται η τεχνική της φραγής ρολογιού.

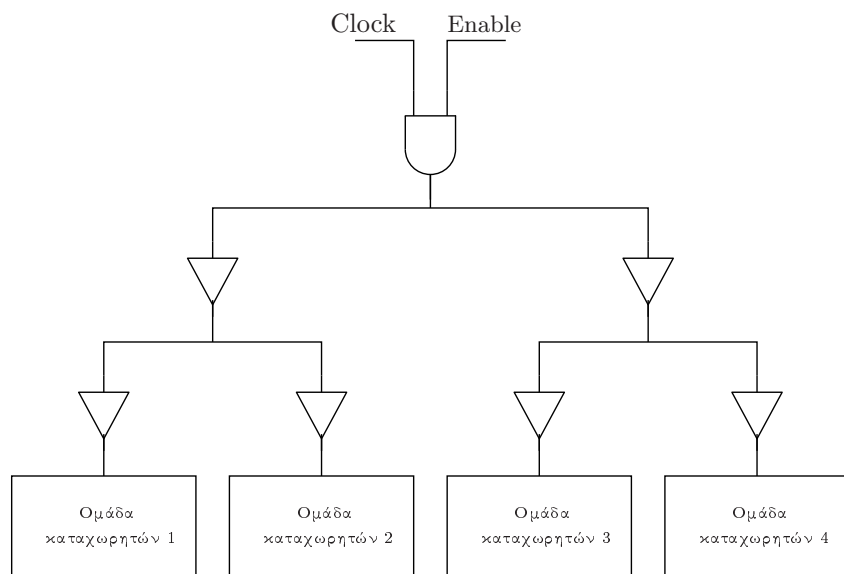
3.1.1 Κόκκος Φραγής

Ανάλογα με το εύρος κόκκου του κυκλώματος στο οποίο εφαρμόζεται, η φραγή ρολογιού μπορεί να διαχωριστεί σε δύο είδη: σε μεγάλου κόκκου ή μικρού κόκκου. Οι δύο αυτές περιπτώσεις παρουσιάζονται στα Σχήματα 3.3 και 3.4 αντίστοιχα. Και στα δύο Σχήματα, τα φύλλα του δέντρου ρολογιού αφορούν ομάδες καταχωρητών.

Η φραγή ρολογιού μεγάλου κόκκου δεν εφαρμόζεται απευθείας στα φύλλα ενός δέντρου ρολογιού αλλά σε υψηλότερα επίπεδά του. Με αυτόν τον τρόπο χρησιμοποιείται για την απενεργοποίηση μεγάλων τμημάτων του κυκλώματος. Συνήθως, και ανάλογα με το βάθος του υποδέντρου ρολογιού στο οποίο εφαρμόζεται και τη συχνότητα του ρολογιού, απαιτούνται αρκετοί κύκλοι για την απενεργοποίηση των καταχωρητών, από



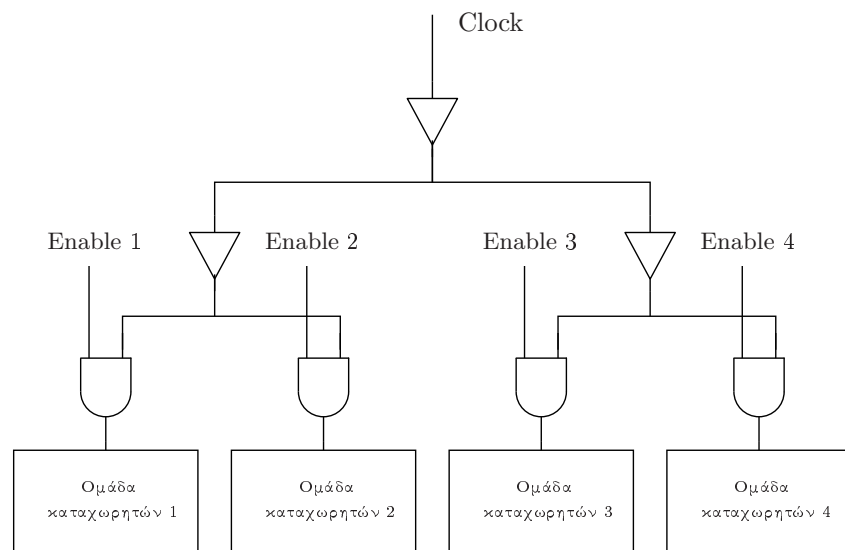
Σχήμα 3.2: Υλοποίηση του *Enable* με φραγή ρολογιού.



Σχήμα 3.3: Φραγή ρολογιού μεγάλου κόκκου.

την στιγμή που απενεργοποιείται το *Enable*. Για το λόγο αυτό, χρησιμοποιείται κυρίως σε περιπτώσεις, στις οποίες είναι επιθυμητή η εξοικονόμηση ενέργειας σε μεγάλα τμήματα του κυκλώματος και για αρκετούς κύκλους ρολογιού.

Αντιθέτως, η φραγή ρολογιού μικρού κόκκου εφαρμόζεται απευθείας στα φύλλα ενός δέντρου ρολογιού. Με αυτόν τον τρόπο ελέγχονται μικρότερες ομάδες καταχωρη-



Σχήμα 3.4: Φραγή ρολογιού μικρού κόκκου.

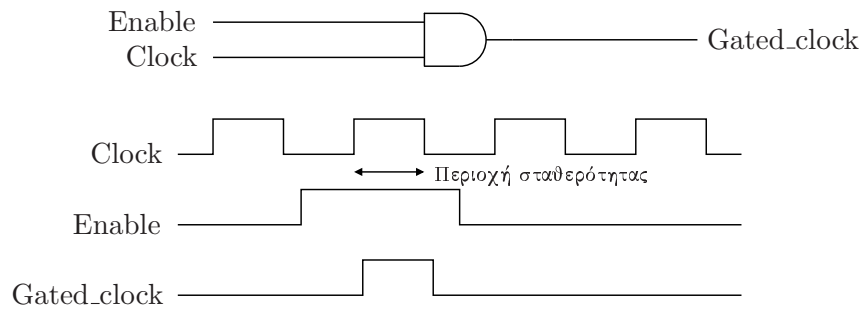
τών από διαφορετικά σήματα *Enable* (*Enable 1, 2, 3* και *4* στο Σχήμα 3.4). Επίσης, σε αντίθεση με τη φραγή ρολογιού μεγάλου κόκκου, η απενεργοποίηση των καταχωρητών γίνεται στον ίδιο κύκλο με αυτήν του *Enable*. Συνεπώς η φραγή ρολογιού μικρού κόκκου εφαρμόζεται σε μικρότερα μέρη του κυκλώματος, ακόμη και για έναν κύκλο ρολογιού.

Τέλος, να σημειωθεί ότι οι περιπτώσεις φραγής ρολογιού μεγάλου κόκκου καθορίζονται από τον ίδιο τον σχεδιαστή, ενώ αυτές της φραγής ρολογιού μικρού κόκκου καθορίζονται με αυτόματο τρόπο από τα εργαλεία λογικής σύνθεσης.

3.1.2 Χρήση Μανταλωτή στην Φραγή Ρολογιού

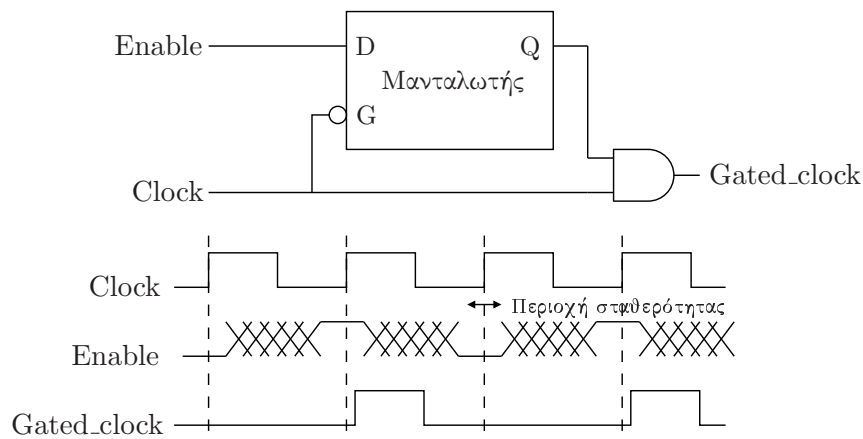
Γενικά, η φραγή ρολογιού μπορεί να υλοποιηθεί με ή χωρίς μανταλωτή [11]. Μια περίπτωση της φραγής ρολογιού χωρίς μανταλωτή παρουσιάζεται στο Σχήμα 3.5. Η τεχνική αυτή χρησιμοποιεί μια απλή πύλη AND ή OR, ανάλογα με την ακμή στην οποία πυροδοτούνται οι καταχωρητές. Όπως φαίνεται και στο Σχήμα, το σήμα του *Enable* θα πρέπει να παραμένει σταθερό από την ενεργή (θετική) ως την ανενεργή (αρνητική) ακμή του ρολογιού. Αν σε αυτό το χρονικό διάστημα η τιμή του *Enable* παρουσιάσει κλυδωνισμούς (hazards), αυτοί θα εμφανιστούν και στην τιμή του *Gated_clock*. Οι κίνδυνοι σε μια τέτοια περίπτωση είναι οι ακόλουθοι:

- Να εμφανιστούν ενεργές ή ανενεργές ακμές του *Gated_clock* σε διαφορετικές χρονικές στιγμές από τις αναμενόμενες.
- Να προκύψουν περισσότερες από μία ενεργές ακμές στο *Gated_clock*, στη διάρκεια μιας περιόδου.



Σχήμα 3.5: Κύκλωμα φραγής ρολογιού χωρίς μανταλωτή.

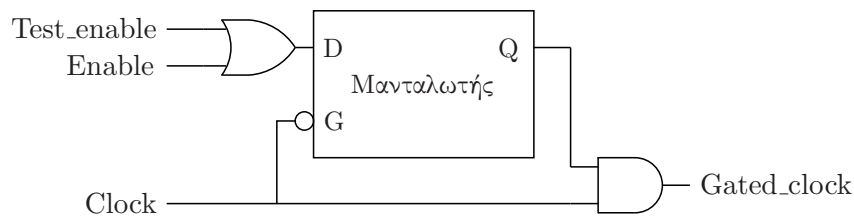
Συνήθως, στις περιπτώσεις φραγής ρολογιού χρησιμοποιείται και ένας μανταλωτής, για την αντιμετώπιση του προβλήματος των κλυδωνισμών του *Enable*. Η χρήση του φαίνεται στο Σχήμα 3.6. Ο μανταλωτής κρατάει σταθερή την τιμή του *Enable* από την ενεργή ακμή του ρολογιού μέχρι και την ανενεργή, οπότε δεν είναι απαραίτητη η προηγούμενη συνθήκη σταθερότητας. Η μόνο συνθήκη η οποία θα πρέπει να τηρείται, είναι το σήμα του *Enable* να είναι σταθερό μόνο γύρω από την ενεργή ακμή, όπως φαίνεται και στο Σχήμα, ώστε να μην προκύψει κάποια χρονική παραβίαση setup ή hold. Τέλος, στο Σχήμα 3.7, παρουσιάζεται η έκδοση του κυκλώματος φραγής ρολογιού με μανταλωτή και ένα επιπλέον σήμα ελέγχου, το *Test_enable*, για υποστήριξη αλυσίδας σάρωσης.



Σχήμα 3.6: Κύκλωμα φραγής ρολογιού με μανταλωτή.

3.1.3 Αποσυγχρονισμός και Φραγή Ρολογιού Μικρού Κόκκου

Στις προηγούμενες ενότητες αναλύθηκε ο τρόπος λειτουργίας της φραγής ρολογιού στα σύγχρονα κυκλώματα. Στην τρέχουσα ενότητα εξετάζεται το πρόβλημα αποσυγχρονισμού κυκλωμάτων, στα οποία χρησιμοποιείται η τεχνική της φραγής ρολογιού μικρού



Σχήμα 3.7: Κύκλωμα φραγής ρολογιού με μανταλωτή και υποστήριξη αλυσίδας σάρωσης.

κόκκου και παρουσιάζεται ο ενδεδειγμένος τρόπος χειρισμού της. Να σημειωθεί ότι η τεχνική αυτή δεν υποστηριζόταν έως τώρα από την υπάρχουσα μεθοδολογία του αποσυγχρονισμού.

Στο Παράρτημα Γ'3 εξετάζεται αναλυτικά το πρόβλημα του αποσυγχρονισμού κυκλωμάτων στα οποία χρησιμοποιείται η τεχνική της φραγής ρολογιού μικρού κόκκου.

3.2 Αρχικοποίηση

Το θέμα της αρχικοποίησης εξετάζεται στο Παράρτημα Γ'4.

3.3 Επικοινωνία με Σύγχρονο Τμήμα

Σε ορισμένες περιπτώσεις, είναι απαραίτητη η επικοινωνία ενός αποσυγχρονισμένου κυκλώματος με κάποιο σύγχρονο τμήμα. Τέτοιες περιπτώσεις είναι για παράδειγμα οι ακόλουθες:

- Κυκλώματα, τα οποία δεν αποσυγχρονίζονται εξολοκλήρου, αλλά ένα τμήμα τους παραμένει σύγχρονο.
- Αποσυγχρονισμός μεμονωμένων IPs (Intellectual properties) και χρησιμοποίησή τους σε σύγχρονα SoC (System on Chip).
- Χρησιμοποίηση του ίδιου testbench, το οποίο δημιουργήθηκε για το αρχικό σύγχρονο κύκλωμα.

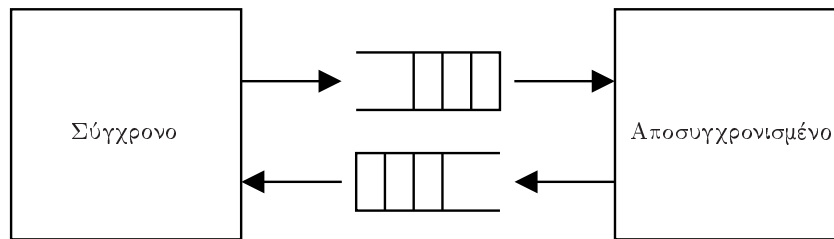
Στις περιπτώσεις αυτές, η επικοινωνία αφορά μεταφορά δεδομένων, τόσο από το αποσυγχρονισμένο κύκλωμα προς στο σύγχρονο τμήμα, όσο και το αντίστροφο. Το κύριο πρόβλημα είναι ότι, ενώ η συχνότητα ενός σύγχρονου τμήματος είναι σταθερή, η συχνότητα του αποσυγχρονισμένου δεν είναι, καθώς εξαρτάται από τις συνθήκες μεταβλητότητας.

Υπάρχουν τρεις τρόποι για να επιτευχθεί αυτή η επικοινωνία. Ο πρώτος τρόπος είναι να χρησιμοποιηθούν FIFOs για την μεταφορά των δεδομένων. Ο δεύτερος είναι να

δημιουργείται ένα σήμα ρολογιού από το ίδιο το αποσυγχρονισμένο κύκλωμα, το οποίο δίνεται και στο σύγχρονο. Τέλος, ο τρίτος τρόπος είναι να παρέχεται το ίδιο εξωτερικό ρολόι και στο αποσυγχρονισμένο κύκλωμα, στο οποίο πρέπει να μεταφράζεται με κάποιο τρόπο σε σήματα χειραψίας.

3.3.1 Επικοινωνία με τη Χρήση FIFOs

Στο Σχήμα 3.8 φαίνεται ο γενικός τρόπος επικοινωνίας ενός σύγχρονου κυκλώματος με ένα αποσυγχρονισμένο, χρησιμοποιώντας FIFOs και προς τις δύο κατευθύνσεις της ροής δεδομένων. Η μέθοδος αυτή επιτρέπει στα δύο κυκλώματα να λειτουργούν σε ανεξάρτητη συχνότητα. Να σημειωθεί ότι αυτού του είδους η επικοινωνία χρειάζεται κάποιον έλεγχο ροής: όταν γεμίζει κάποια FIFO πρέπει να αναστέλλεται η λειτουργία του κυκλώματος που την τροφοδοτεί με δεδομένα, ενώ αντίστοιχα όταν αδειάζει πρέπει να αναστέλλεται η λειτουργία του κυκλώματος που λαμβάνει δεδομένα από αυτή.

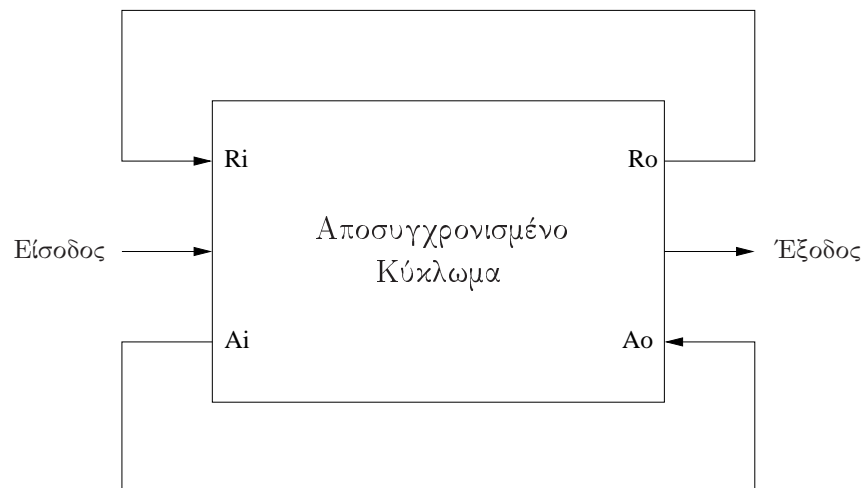


Σχήμα 3.8: Επικοινωνία σύγχρονου και αποσυγχρονισμένου κυκλώματος με τη χρήση FIFOs.

3.3.2 Συγχρονισμός Μέσω του Αποσυγχρονισμένου Κυκλώματος

Σύμφωνα με αυτή την προσέγγιση τα εξωτερικά σήματα χειραψίας του αποσυγχρονισμένου κυκλώματος ενώνονται μεταξύ τους και από αυτά προκύπτει ένα σήμα ρολογιού, το οποίο δίνεται στα σύγχρονα τμήματα. Με αυτόν τον τρόπο η καθολική συχνότητα λειτουργίας καθορίζεται αυτόματα από το αποσυγχρονισμένο κύκλωμα και οποιαδήποτε μεταβολή τάσης ή θερμοκρασίας τη μεταβάλλει αντίστοιχα.

Στο Σχήμα 3.9 φαίνεται ο τρόπος με τον οποίο πρέπει να γίνει η ένωση των σημάτων χειραψίας στο αποσυγχρονισμένο κύκλωμα. Αν οι ασύγχρονοι ελεγκτές που χρησιμοποιούνται είναι τετραφασικοί, τότε το σήμα ρολογιού προκύπτει απευθείας από το R_0 ή το A_1 . Αντίθετα αν είναι διφασικοί, πρέπει να χρησιμοποιηθεί ένας ανιχνευτής ακμής για το R_0 ή το A_1 , ο οποίος σε κάθε θετική ή αρνητική ακμή παράγει ένα θετικό παλμό ρολογιού. Το πλάτος του παλμού αυτού μπορεί να καθορίζεται π.χ. από ένα στοιχείο καθυστέρησης.



Σχήμα 3.9: Τρόπος σύνδεσης σημάτων χειραψίας σε ένα αποσυγχρονισμένο κύκλωμα. Από τα R_o ή A_i μπορεί να γίνει εξαγωγή ρολογιού για τα σύγχρονα τμήματα.

Το ρολόι, το οποίο προκύπτει από αυτήν την διαδικασία, μπορεί να χρειάζεται αλλαγή φάσης, ώστε να ικανοποιεί τους χρονικούς περιορισμούς των σύγχρονων τμημάτων. Ένα τέτοιο παράδειγμα είναι η περίπτωση στην οποία το αποσυγχρονισμένο τμήμα δίνει δεδομένα σε κάποιο σύγχρονο, το οποίο έχει στις εισόδους του συνδυαστική λογική. Σε αυτή την περίπτωση, το ρολόι προκύπτει από το R_o , με χρονική μετατόπιση, η οποία αντιστοιχεί στο κρίσιμότερο μονοπάτι αυτής της συνδυαστικής λογικής. Αυτό μπορεί να επιτευχθεί με τη χρήση ενός αντίστοιχου στοιχείου καθυστέρησης.

3.3.3 Συγχρονισμός Μέσω Εξωτερικού Ρολογιού

Στην περίπτωση αυτή, πρέπει να υπάρχει ένας μηχανισμός, ο οποίος να μεταφράζει το εξωτερικό ρολόι σε σήματα χειραψίας του αποσυγχρονισμένου κυκλώματος. Με αυτόν τον τρόπο, το αποσυγχρονισμένο κύκλωμα λειτουργεί σε σταθερή συχνότητα, η οποία αντιστοιχεί στη συχνότητα του ρολογιού.

Ο τρόπος δημιουργίας ενός τέτοιου είδους συγχρονισμού εξετάζεται στο Παράρτημα Γ'.5.

Με τον συγκεκριμένο τρόπο συγχρονισμού, το αποσυγχρονισμένο κύκλωμα, παρόλο που λειτουργεί σε σταθερή συχνότητα, διατηρεί ορισμένα από τα πλεονεκτήματα της μεθόδου του αποσυγχρονισμού. Αυτά είναι:

- Χαμηλότερες ηλεκτρομαγνητικές εκπομπές από το αρχικό σύγχρονο κύκλωμα.
- Δυνατότητα μείωσης της τάσης λειτουργίας του, όσο επιτρέπεται από τη δεδομένη συχνότητα του ρολογιού. Με αυτόν τον τρόπο επιτυγχάνεται μείωση της κατανάλωσης ενέργειας. Να σημειωθεί ότι η χαμηλότερη επιτρεπτή τάση λειτουργίας

μπορεί να διαπιστωθεί από τον έλεγχο των σημάτων output request και input acknowledge.

3.4 Testbench

Μία από τις βασικές μεθόδους για τον έλεγχο της ορθότητας των σύγχρονων κυκλωμάτων, είναι η προσομοίωση της λειτουργίας τους με κάποιο testbench. Αυτό ισχύει και στην περίπτωση των αποσυγχρονισμένων κυκλωμάτων. Μάλιστα, για την προσομοίωση ενός αποσυγχρονισμένου κυκλώματος μπορεί να χρησιμοποιηθεί το ίδιο testbench, το οποίο χρησιμοποιείται για τον έλεγχο του αντίστοιχου σύγχρονου, με ορισμένες μικρές μετατροπές. Με αυτόν τον τρόπο μπορούμε να προσομοιώσουμε τα δύο κυκλώματα στο ίδιο ακριβώς σενάριο λειτουργίας, να συγκρίνουμε τη λειτουργία τους και να διαπιστώσουμε αν τελικά είναι ισοδύναμα ή όχι. Αν για οποιοδήποτε λόγο δεν λειτουργεί σωστά το αποσυγχρονισμένο κύκλωμα, μπορούμε να αναζητήσουμε το σημείο του σφάλματος, βρίσκοντας το χρονικό σημείο στο οποίο η λειτουργία μεταξύ των δύο κυκλωμάτων αρχίζει να παρεκκλίνει.

3.4.1 Μετατροπές

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.6.1.

3.4.2 Μέθοδος Επαλήθευσης

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.6.2.

3.4.3 Προβληματικές Περιπτώσεις

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.6.3.

3.5 Υλοποίηση Στοιχείων C

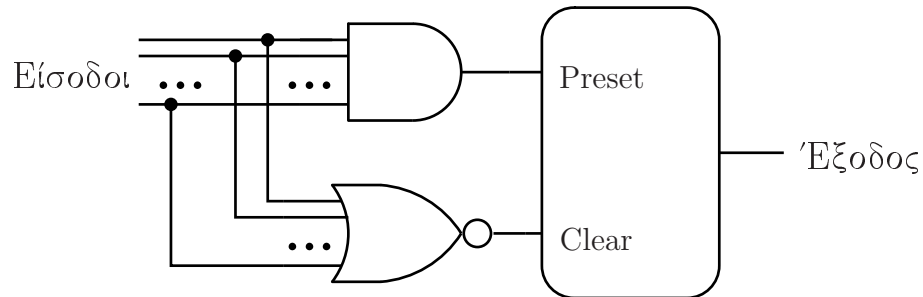
Ένα από τα βασικά δομικά στοιχεία ενός ασύγχρονου δικτύου ελέγχου είναι τα στοιχεία C. Για τον λόγο αυτό πρέπει να δίνεται ιδιαίτερη προσοχή στην υλοποίησή τους, ώστε να επιτυγχάνεται το καλύτερο δυνατό αποτέλεσμα, όσον αφορά τον τομέα την ταχύτητας και του εμβαδού.

Γενικά, ένα στοιχείο C με n εισόδους υλοποιεί την ακόλουθη εξίσωση Boole:

$$C_n = (a_1 a_2 \dots a_n) + C'_n (a_1 + a_2 + \dots + a_n) \quad (3.1)$$

Οι όροι a_1 έως a_n συμβολίζουν τις εισόδους του στοιχείου C, ενώ ο όρος C'_n συμβολίζει την προηγούμενη κατάσταση του. Από αυτήν την εξίσωση προκύπτει ότι η υλοποίηση

κάθε στοιχείου C μπορεί να αναχθεί σε μια υλοποίηση ενός κυκλώματος με δύο βασικά σήματα, τα **Preset** και **Clear**, όπως φαίνεται και στο Σχήμα 3.10. Όταν όλοι οι είσοδοι είναι 0, ενεργοποιείται το σήμα **Clear** του εσωτερικού κυκλώματος, οπότε η έξοδος γίνεται 0. Αντίθετα όταν όλες οι είσοδοι είναι 1, ενεργοποιείται το σήμα **Preset**, οπότε η έξοδος γίνεται 1.



Σχήμα 3.10: Γενική εικόνα υλοποίησης στοιχείου C.

Αναλυτικά ο τρόπος δημιουργίας των στοιχείων C εξετάζεται στο Παράρτημα Γ'.7.

3.6 Χρονική Βελτιστοποίηση του Ασύγχρονου Δικτύου Ελέγχου

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.8.

3.7 Ομαδοποίηση

Η ομαδοποίηση είναι η διαδικασία κατά την οποία οι πύλες του κυκλώματος ανατίθενται σε ξεχωριστές ομάδες αποσυγχρονισμού. Κάθε τέτοια ομάδα θα ελέγχεται από ένα διαφορετικό ζεύγος ασύγχρονων ελεγκτών για τους μανταλωτές τύπου Αφέντη και Σκλάβου, όπως εξηγήσαμε στην Ενότητα 2.2.2.

Στην Ενότητα αυτή εξετάζουμε τρεις διαφορετικές μεθόδους ομαδοποίησης: την χειροκίνητη ομαδοποίηση, τον αλγόριθμο “χαμηλής προσπάθειας” και τον αλγόριθμο “μεσαίας προσπάθειας”. Αυτές αντιστοιχούν στις εναλλακτικές μεθόδους που υποστηρίζονται από τα εργαλεία αποσυγχρονισμού *drdesync* και *Nanosync*, τα οποία χρησιμοποιήθηκαν στην παρούσα εργασία. Συγκεκριμένα, το *drdesync* υποστηρίζει χειροκίνητη ομαδοποίηση και τον αλγόριθμο “χαμηλής προσπάθειας”, ενώ το *Nanosync* υποστηρίζει και τις τρεις μεθόδους. Να σημειωθεί ότι οι ονομασίες των αλγορίθμων “χαμηλής” και “μεσαίας προσπάθειας” χρησιμοποιούνται με αυτόν τον τρόπο από το *Nanosync*, και τις διατηρούμε και στην παρούσα εργασία.

3.7.1 Καθολική Λογική

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.9.1.

3.7.2 Χειροκίνητη Ομαδοποίηση

Η μέθοδος αυτή δεν περιλαμβάνει κάποιον αλγόριθμο όπως οι άλλοι. Αντίθετα, ο ίδιος ο σχεδιαστής είναι υπεύθυνος, γνωρίζοντας την δομή του κυκλώματος, να αναθέσει τον καθένα από τους καταχωρητές του κυκλώματος σε ξεχωριστές ομάδες αποσυγχρονισμού.

Ο τρόπος που πραγματοποιείται η χειροκίνητη ομαδοποίηση, εξαρτάται από το εργαλείο αποσυγχρονισμού που χρησιμοποιείται. Στην περίπτωση του *drdesync* αυτό πρέπει να γίνει με κάποιο εργαλείο λογικής σύνθεσης, όπως το *Synopsys Design Compiler*. Συγκεκριμένα θα πρέπει το κύκλωμα να μετατραπεί σε μια μορφή δύο επιπέδων ιεραρχίας, στην οποία το ανώτατο τμήμα περιλαμβάνει επίπεδα υποτιμήματα. Το *drdesync* αναθέτει κάθε τέτοιο υποτιμήμα σε διαφορετική ομάδα αποσυγχρονισμού [1].

Ο τρόπος με τον οποίο πραγματοποιείται η χειροκίνητη ομαδοποίηση στο *Nanosync* εξετάζεται στο Παράρτημα Γ'.9.2.

3.7.3 Αλγόριθμος "Χαμηλής Προσπάθειας"

Ο αλγόριθμος "χαμηλής προσπάθειας" υποστηρίζεται τόσο από το *drdesync* όσο και από το *Nanosync*, και παρουσιάζεται αναλυτικά στο [1]. Ο συγκεκριμένος αλγόριθμος αντιμετωπίζει το πρόβλημα της ομαδοποίησης με αντίστροφο τρόπο, όχι δηλαδή με γνώμονα τους καταχωρητές αλλά με γνώμονα τις συνδυαστικές πύλες.

Τα βήματα που εκτελούνται είναι τα εξής:

1. Ομαδοποιούνται όλες οι συνδυαστικές πύλες, οι οποίες είναι συνδεδεμένες μεταξύ τους. Με αυτόν τον τρόπο αναγνωρίζονται τα συνδυαστικά τμήματα του κυκλώματος, τα οποία είναι ανεξάρτητα το ένα από το άλλο. Από τα τμήματα αυτά προκύπτουν οι ομάδες αποσυγχρονισμού του κυκλώματος.
2. Σε κάθε ομάδα προστίθενται οι καταχωρητές, οι οποίοι οδηγούνται από τα αντίστοιχα συνδυαστικά τμήματα.
3. Όσοι καταχωρητές απομένουν, ανατίθενται σε μια ξεχωριστή ομάδα.

3.7.4 Αλγόριθμος "Μεσαίας Προσπάθειας"

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.9.3.

3.7.5 Κόκκος Αποσυγχρονισμού

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.9.4.

3.8 Σύγκριση Διφασικών και Τετραφασικών Ασύγχρονων Ελεγκτών

Στην Ενότητα αυτή παρουσιάζεται μια γενική σύγκριση μεταξύ διφασικών και τετραφασικών ελεγκτών, κυρίως όσον αφορά τη χρήση τους στη μέθοδο του αποσυγχρονισμού. Να σημειωθεί ότι δεν συγκρίνονται αποκλειστικά οι ελεγκτές που παρουσιάστηκαν στις Ενότητες 2.4.3 και 2.4.4, αλλά οι δύο γενικότερες κατηγορίες μεταξύ τους.

Το κύριο πλεονέκτημα των τετραφασικών ελεγκτών, σε σχέση με τους διφασικούς, είναι ότι συνήθως είναι απλούστεροι στην υλοποίηση [9]. Αυτό έχει ως αποτέλεσμα να αποτελούνται από λιγότερες πύλες, με προφανή οφέλη σε εμβαδόν και ταχύτητα.

Το πλεονέκτημα σε εμβαδόν των τετραφασικών ελεγκτών δεν είναι τόσο σημαντικό στον αποσυγχρονισμό, αφού όπως θα δούμε στην Ενότητα 7.1.4, η κυρίως επιβάρυνση σε εμβαδόν οφείλεται στην αντικατάσταση των καταχωρητών από μανταλωτές και όχι στην εισαγωγή ελεγκτών. Ακόμα και όταν αυξάνεται το πλήθος των ομάδων αποσυγχρονισμού, αυτό που προκαλεί την περαιτέρω αύξηση σε εμβαδόν δεν είναι το πλήθος των ελεγκτών, το οποίο αυξάνεται γραμμικά, αλλά το πλήθος των στοιχείων καθυστέρησης, το οποίο μπορεί να αυξάνεται μέχρι και τετραγωνικά (βλέπε Σχέση 2.1).

Όσον αφορά την σύγκριση ταχύτητας μεταξύ των δύο κατηγοριών, οι τετραφασικοί μπορεί να έχουν γενικά απλούστερη υλοποίηση, αλλά ο αριθμός των μεταβάσεων που απαιτούν για κάθε χειραψία είναι διπλάσιος από αυτόν των διφασικών. Αυτό διότι, κάθε φορά, θα πρέπει τα σήματα Αίτησης και Αναγνώρισης να ξαναγυρίσουν στην αρχική τους κατάσταση (επιστροφή στο 0), πριν ξεκινήσει η επόμενη ανταλλαγή δεδομένων. Αυτό έχει αρνητική επίπτωση και στην κατανάλωση ενέργειας.

Το γεγονός της περιττής επιστροφής στο 0 αποκτά ιδιαίτερη σημασία, αν λάβουμε υπόψη και τις καθυστερήσεις των διασυνδέσεων. Κατά την τοποθέτηση και διασύνδεση, οι ελεγκτές μπορεί να τοποθετηθούν μακριά ο ένας από τον άλλο, με αποτέλεσμα τη δημιουργία μακρινών διασυνδέσεων σημάτων για τα Αίτησης και Αναγνώρισης. Οπότε ο χρόνος της επιστροφής στο 0, ενώ ιδανικά θα έπρεπε να είναι ελάχιστος, λόγω και της χρήσης ασύμμετρων στοιχείων καθυστέρησης (βλέπε Ενότητα 2.5.3), μπορεί να γίνει ιδιαίτερα σημαντικός.

Το συμπέρασμα είναι ότι οι διφασικοί ελεγκτές είναι γενικά προτιμότεροι από τους τετραφασικούς, από άποψη ταχύτητας και κατανάλωσης ενέργειας. Για τον λόγο αυτό χρησιμοποιήθηκαν στις περιπτώσεις αποσυγχρονισμού των κυκλωμάτων IND1b και IND2 (βλέπε Κεφάλαιο 6).

Κεφάλαιο 4

Ροή Αποσυγχρονισμού

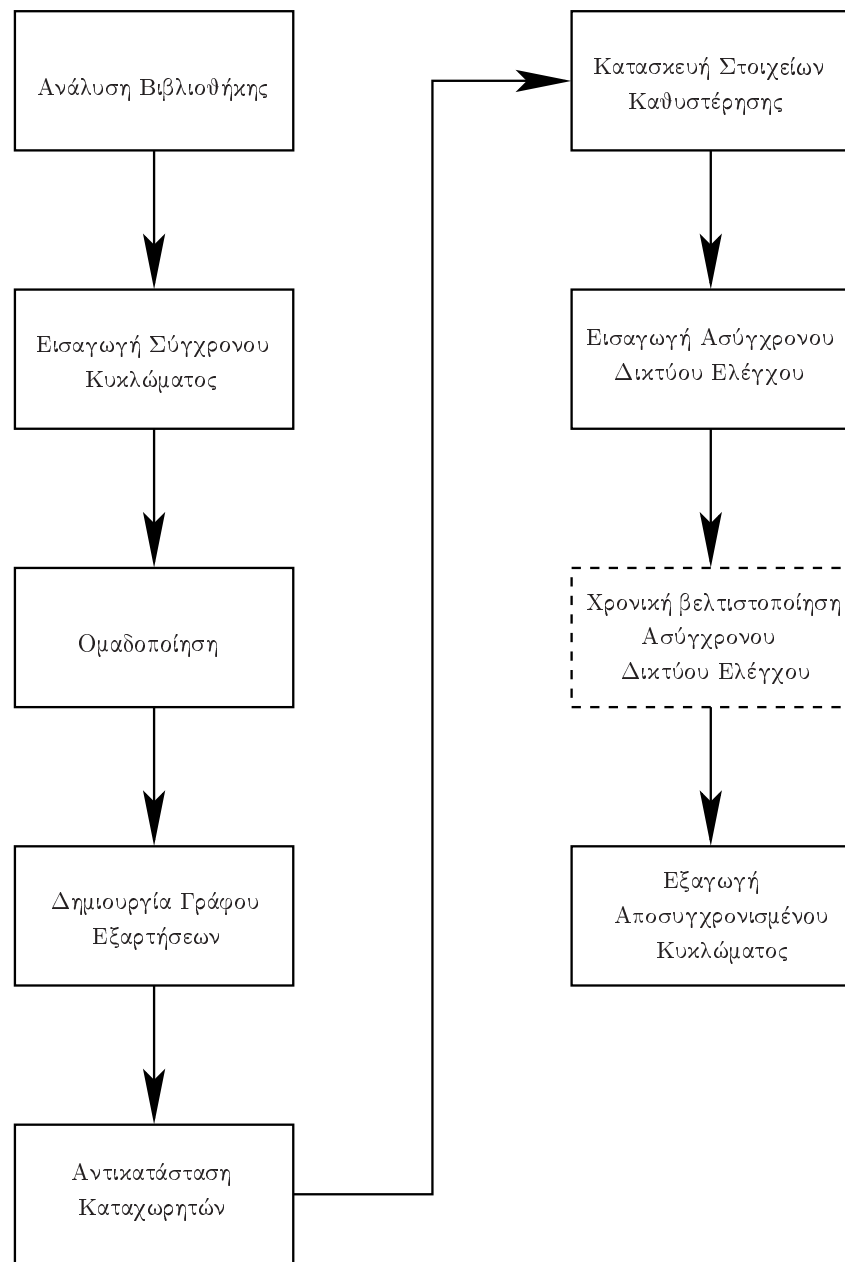
Στα προηγούμενα κεφάλαια, παρουσιάσαμε τις θεμελιώδεις έννοιες και ορισμένα ζητήματα του αποσυγχρονισμού, και αναλύσαμε τα προαπαιτούμενα για την εφαρμογή της μεθόδου σε κυκλώματα βιομηχανικής πολυπλοκότητας. Στο παρών κεφάλαιο παρουσιάζουμε λεπτομερώς τα βήματα της ροής του, με τη σειρά που αυτά εκτελούνται. Συγκεκριμένα, εξετάζουμε τα βήματα που εκτελούν δύο συγκεκριμένα εργαλεία αποσυγχρονισμού, το *drdesync* και το *Nanosync*.

Η ανάπτυξη του *drdesync* έγινε από τον Νικόλαο Ανδρίκο, στα πλαίσια της Μεταπτυχιακής του εργασίας [1]. Το συγκεκριμένο εργαλείο χρησιμοποιεί μόνο τους τετραφασικούς ελεγκτές που παρουσιάστηκαν στην Ενότητα 2.4.3. Για τον λόγο αυτό χρησιμοποιεί στοιχεία καθυστέρησης ασύμμετρου τύπου. Όσον αφορά το *Nanosync*, η ανάπτυξή του γίνεται από την εταιρία Nanochronous Logic. Χρησιμοποιεί, έως τώρα τους διφασικούς ελεγκτές που παρουσιάστηκαν στην Ενότητα 2.4.4, καθώς και στοιχεία καθυστέρησης συμμετρικού τύπου.

4.1 Ανάλυση Βιβλιοθήκης Τεχνολογίας

Το πρώτο βήμα που πρέπει να γίνει, είναι η ανάλυση της βιβλιοθήκης τεχνολογίας που χρησιμοποιεί το σύγχρονο κύκλωμα. Η ανάλυση αυτή έχει τους εξής στόχους:

1. Να εντοπίσει τα διάφορα είδη διαθέσιμων καταχωρητών και αντίστοιχων μανταλωτών. Αν για κάποιους καταχωρητές δεν υπάρχουν αντίστοιχοι μανταλωτές, τότε θα πρέπει να χρησιμοποιηθούν επιπλέον κάποιες συνδυαστικές πύλες, όπως για παράδειγμα δείξαμε στο Σχήμα 2.3.
2. Να κατασκευάσει ασύγχρονους ελεγκτές από τις κατάλληλες πύλες της βιβλιοθήκης.
3. Να εντοπίσει πύλες που χρειάζονται για την κατασκευή στοιχείων C.



Σχήμα 4.1: Διάγραμμα ροής αποσυγχρονισμού. Το βήμα της “χρονικής βελτιστοποίησης” είναι προαιρετικό.

4. Να εντοπίσει πύλες, οι οποίες μπορούν να χρησιμοποιηθούν για την κατασκευή των στοιχείων καθυστέρησης.

drdesync

Στην περίπτωση του *drdesync*, η διαδικασία ανάλυσης της βιβλιοθήκης δεν είναι αυτοματοποιημένη και απαιτεί χειροκίνητες ρυθμίσεις από τον σχεδιαστή. Συγκεκριμένα,

με τη βοήθεια κάποιου script, αναλύονται τα αρχεία *.lib* της βιβλιοθήκης και παράγεται ένα αρχείο *gatefile*. Το αρχείο αυτό περιέχει πληροφορίες για όλες τις πύλες της βιβλιοθήκης, ενώ για κάθε καταχωρητή περιέχει τα ονόματα των δύο μανταλωτών που μπορούν να τον αντικαταστήσουν. Επίσης, παράγεται μια λίστα με τους καταχωρητές, των οποίων η αντικατάσταση απαιτεί τη χρήση επιπλέον συνδυαστικών πυλών. Ο τρόπος αντικατάστασης αυτών των καταχωρητών θα πρέπει να καθορισθεί από τον ίδιο τον σχεδιαστή.

Η τεχνολογική αντιστοίχιση των ασύγχρονων ελεγκτών θα πρέπει να γίνει και αυτή από τον σχεδιαστή χειροκίνητα, όπως επίσης και η κατασκευή των στοιχείων καθυστέρησης, καθώς και των στοιχείων C. Για την κατασκευή των τελευταίων απαιτείται η χρήση ενός εργαλείου λογικής σύνθεσης, όπως είναι λ.χ. το *Synopsys Design Compiler*. Όσον αφορά τα στοιχεία καθυστέρησης, αυτά κατασκευάζονται σε διάφορα μεγέθη για μελλοντική χρήση, με πύλες AND. Για να βρεθεί η καθυστέρηση τους, χρησιμοποιείται κάποιο εργαλείο ΣΧΑ, όπως λ.χ. το *Synopsys PrimeTime*.

Nanosync

Το *Nanosync*, σε αντίθεση με το *drdesync*, αναλύει τα αρχεία *.lib* της βιβλιοθήκης τεχνολογίας με αυτόματο τρόπο. Επίσης, εντοπίζει αυτόματα τις πύλες που απαιτούνται για την σωστή αντικατάσταση των καταχωρητών της βιβλιοθήκης, καθώς και τις πύλες που μπορούν να χρησιμοποιηθούν για τη δημιουργία των ασύγχρονων ελεγκτών, των στοιχείων C και των στοιχείων καθυστέρησης.

4.2 Εισαγωγή Σύγχρονου Κυκλώματος

Σε αυτό το βήμα, το σύγχρονο κύκλωμα εισάγεται στο εργαλείο αποσυγχρονισμού. Το κύκλωμα αυτό πρέπει να είναι σε μορφή επιπέδου πυλών, η οποία θα έχει προκύψει έπειτα από τις φάσεις της σύνθεσης και του DFT. Τόσο το *drdesync* όσο και το *Nanosync*, υποστηρίζουν κυκλώματα σε γλώσσα περιγραφής Verilog.

4.3 Ομαδοποίηση

Η διαδικασία ομαδοποίησης των κυκλωμάτων περιγράφεται αναλυτικά στην Ενότητα 3.7. Στην παρούσα ενότητα θα εξετάσουμε τον τρόπο με τον οποίο υλοποιείται από τα εργαλεία *drdesync* και *Nanosync*.

drdesync

Στο *drdesync*, υπάρχουν δύο επιλογές:

1. Να γίνει χειροκίνητη ομαδοποίηση.
2. Να γίνει αυτοματοποιημένη ομαδοποίηση με τον αλγόριθμο “χαμηλής προσπάθειας”.

Στην πρώτη περίπτωση, όπως περιγράψαμε στην Ενότητα 3.7.2, θα πρέπει να δοθεί στο *drdesync* μία μορφή του κυκλώματος με δύο επίπεδα ιεραρχίας. Σε αυτή τη μορφή, το ανώτατο τμήμα περιλαμβάνει μόνο επίπεδα υποτιμήματα, τα οποία θα αποτελούν και τις ομάδες αποσυγχρονισμού του κυκλώματος.

Στην δεύτερη περίπτωση εκτελείται ο αλγόριθμος “χαμηλής προσπάθειας” της Ενότητας 3.7.3. Για να έχει αυτός ο αλγόριθμος όσο το δυνατόν καλύτερο αποτέλεσμα χρησιμοποιούνται δύο επιπλέον ευριστικά:

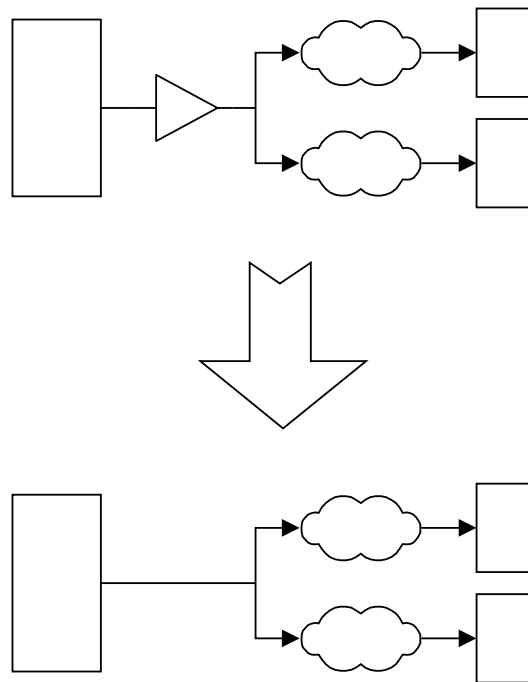
- Καθαρισμός λογικής: Ο αλγόριθμος “χαμηλής προσπάθειας” βασίζεται στην εύρεση των λογικών διασυνδέσεων. Για να ανταποκρίνονται όμως σε πραγματικές εξαρτήσεις δεδομένων, θα πρέπει να περιλαμβάνεται στο κύκλωμα μόνο “καθαρή” λογική, χωρίς την παρουσία ενισχυτών ή ζευγών από αντιστροφείς. Οπότε, πριν την ομαδοποίηση, το *drdesync* ανιχνεύει και αφαιρεί αυτές τις πύλες από το κύκλωμα. Με αυτόν τον τρόπο αντιμετωπίζονται οι περιπτώσεις, στις οποίες κάποιος αντιστροφέας ή ενισχυτής τροφοδοτείται από κάποιον καταχωρητή και τροφοδοτεί δύο ή περισσότερα συνδυαστικά σύννεφα, όπως φαίνεται στο Σχήμα 4.2.
- Ανίχνευση διάυλων δεδομένων: Το *drdesync*, χρησιμοποιώντας τα ονόματα των καταχωρητών, ανιχνεύει και ομαδοποιεί τα σήματα, τα οποία ανήκουν στον ίδιο διάυλο δεδομένων.

Τέλος, δίνεται στον σχεδιαστή και η δυνατότητα να προσδιορίσει διασυνδέσεις, οι οποίες δεν πρέπει να ληφθούν υπόψη κατά τη διαδικασία της ομαδοποίησης.

Nanosync

Στο *Nanosync*, ο σχεδιαστής θα πρέπει αρχικά να προσδιορίσει την καθολική λογική, η οποία, όπως είδαμε στην Ενότητα 3.7.1), δεν συμμετέχει στην διαδικασία της ομαδοποίησης. Στη συνέχεια, παρέχονται στον σχεδιαστή οι ακόλουθες επιλογές:

- Να ομαδοποιήσει το κύκλωμα χειροκίνητα.
- Να πραγματοποιήσει αυτοματοποιημένη ομαδοποίηση με τον αλγόριθμο “χαμηλής προσπάθειας”.
- Να πραγματοποιήσει αυτοματοποιημένη ομαδοποίηση με τον αλγόριθμο “μεσαίας προσπάθειας”.



Σχήμα 4.2: Περίπτωση καθαρισμού λογικής.

- Ένα μέρος του κυκλώματος να ομαδοποιηθεί χειροκίνητα και το υπόλοιπο με κάποιον από τους αλγόριθμους αυτοματοποιημένης ομαδοποίησης.

Η χειροκίνητη ομαδοποίηση καθώς και οι αλγόριθμοι “χαμηλής” και “μεσαίας προσπάθειας” παρουσιάστηκαν στην Ενότητα 3.7. Επίσης, δίνεται η δυνατότητα στον σχεδιαστή να αλλάξει το τελικό αποτέλεσμα, μετακινώντας πύλες σε διαφορετικές ομάδες.

4.4 Δημιουργία Γράφου Εξαρτήσεων των Ομάδων

Μετά και την ολοκλήρωση της ομαδοποίησης, μπορεί να δημιουργηθεί ο γράφος εξαρτήσεων των ομάδων. Ο κόμβοι του αντιστοιχούν στις ομάδες αποσυγχρονισμού του κυκλώματος και οι ακμές του στις μεταξύ τους εξαρτήσεις δεδομένων. Για παράδειγμα, στο Σχήμα 2.4 δείξαμε τον γράφο εξαρτήσεων, ο οποίος προκύπτει από την ομαδοποίηση του κυκλώματος που φαίνεται στο Σχήμα 2.1.

Στο *drdesync*, το βήμα αυτό γίνεται μετά την αντικατάσταση των καταχωρητών από μανταλωτές.

4.5 Αντικατάσταση Καταχωρητών από Μανταλωτές

Κάθε καταχωρητής του κυκλώματος αντικαθίσταται από ένα ζεύγος μανταλωτών. Ο τρόπος αυτής της αντικατάστασης, για όλους τους καταχωρητές της βιβλιοθήκης, έχει

καθοριστεί από πριν, στο βήμα της Ενότητας 4.1.

4.6 Κατασκευή Στοιχείων Καθυστερήσης

Για κάθε ακμή του γράφου εξαρτήσεων των ομάδων θα δημιουργηθεί ένα αντίστοιχο στοιχείο καθυστέρησης. Κάθε τέτοιο στοιχείο προκύπτει από το κρίσιμο μονοπάτι της αντίστοιχης συνδυαστικής λογικής, όπως περιγράψαμε στην Ενότητα 2.5.

drdesync

Το *drdesync* υπολογίζει τις καθυστερήσεις των κρίσιμων μονοπατιών, χρησιμοποιώντας ένα εξωτερικό εργαλείο ΣΧΑ, όπως λ.χ. το *Synopsys PrimeTime*. Οι τιμές των καθυστερήσεων αυτών χρησιμοποιούνται για την επιλογή των αντίστοιχων προκατασκευασμένων στοιχείων καθυστέρησης. Επίσης η χρονική ανάλυση στα στοιχεία καθυστέρησης έχει γίνει από πριν, στο βήμα της Ενότητας 4.1.

Nanosync

Το *Nanosync* υπολογίζει τις καθυστερήσεις των κρίσιμων μονοπατιών, χρησιμοποιώντας δικό του εσωτερικό χρονικό αναλυτή. Από τις καθυστερήσεις αυτές κατασκευάζονται αντίστοιχα στοιχεία καθυστέρησης.

4.7 Εισαγωγή Ασύγχρονου Δικτύου Ελέγχου

Για κάθε ομάδα αποσυγχρονισμού εισάγεται ένας ασύγχρονος ελεγκτής, ο οποίος ελέγχει τους μανταλωτές. Επίσης, εισάγονται τα κατάλληλα στοιχεία C, τα οποία συνδέουν τους ελεγκτές μεταξύ τους, καθώς και με τα στοιχεία καθυστέρησης.

4.8 Χρονική Βελτιστοποίηση του Ασύγχρονου Δικτύου Ελέγχου

Το βήμα αυτό εκτελείται μόνο από το *Nanosync*. Ο στόχος είναι η βελτιστοποίηση της ταχύτητας του αποσυγχρονισμένου κυκλώματος.

4.9 Εξαγωγή Αποσυγχρονισμένου Κυκλώματος

Το τελικό βήμα είναι η εξαγωγή του αποσυγχρονισμένου κυκλώματος. Όπως και στην περίπτωση του σύγχρονου, το αποσυγχρονισμένο κύκλωμα είναι σε μορφή επιπέδου πυλών.

Κεφάλαιο 5

Συνεισφορά στην Ανάπτυξη του *Nanosync*

Στο παρών κεφάλαιο περιγράφεται η συνεισφορά της εργασίας στην ανάπτυξη του εργαλείου αποσυγχρονισμού *Nanosync*. Συγκεκριμένα, στην Ενότητα 5.1 περιγράφεται η ανάπτυξη κώδικα, η οποία έγινε για την υποστήριξη ορισμένων λειτουργιών του. Στην Ενότητα 5.2 περιγράφεται η αποσφαλμάτωση του *Nanosync*, η οποία έγινε κατά τη διάρκεια της εργασίας αποσυγχρονισμού των βιομηχανικών κυκλωμάτων IND1b και IND2.

5.1 Ανάπτυξη Κώδικα

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.10.1.

5.2 Αποσφαλμάτωση

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.10.1.

Κεφάλαιο 6

Τρία Βιομηχανικά Κυκλώματα για Αποσυγχρονισμό

Στα προηγούμενα κεφάλαια, εξετάσαμε αναλυτικά όλα τα βήματα του αποσυγχρονισμού, καθώς και διάφορα θέματα που σχετίζονται με αυτόν. Στο παρών κεφάλαιο, θα δούμε πώς ο αποσυγχρονισμός εφαρμόστηκε στην περίπτωση τριών βιομηχανικών κυκλωμάτων. Επίσης, εξετάζουμε τα προβλήματα που παρουσιάστηκαν και πώς αυτά αντιμετωπίστηκαν.

Τα τρία κυκλώματα με τα οποία ασχοληθήκαμε, είναι τα εξής:

- IND1a
- IND1b
- IND2

Τα δύο πρώτα κυκλώματα είναι επεξεργαστές, οι οποίοι υλοποιούν δύο διαφορετικές εκδόσεις της αρχιτεκτονικής IND1. Η αρχιτεκτονική αυτή αναπτύσσεται από την εταιρία X_IND1_company_X. Το τρίτο κύκλωμα είναι ένας μικροελεγκτής, ο οποίος υλοποιήθηκε από την εταιρία X_Company_X.

Η εργασία αποσυγχρονισμού και των τριών κυκλωμάτων πραγματοποιήθηκε σε συνεργασία με την εταιρία X_Company_X.

6.1 Επεξεργαστής IND1a

6.1.1 Σύντομη Περιγραφή του Κυκλώματος

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.11.1.

6.1.2 Αποσυγχρονισμός και Επαλήθευση του IND1a

Η εργασία αποσυγχρονισμού του IND1a πραγματοποιήθηκε σε τρεις φάσεις. Οι φάσεις αυτές αφορούν αντίστοιχα τρεις χρονικές περιόδους και τρία διαφορετικά πρόσωπα. Είναι οι ακόλουθες:

- Πρώτη φάση: Αυτή πραγματοποιήθηκε από τον Νικόλαο Ανδρίκο, μεταπτυχιακό φοιτητή του Πανεπιστημίου Κρήτης και του Ιδρύματος Τεχνολογίας και Έρευνας. Η εργασία αυτή έγινε στα πλαίσια της μεταπτυχιακής του εργασίας.
- Δεύτερη φάση: Αυτή πραγματοποιήθηκε από τον X_Student_X, μεταπτυχιακό φοιτητή του X_University_X.
- Τρίτη φάση: Αυτή πραγματοποιήθηκε από τον συγγραφέα, μεταπτυχιακό φοιτητή του Πανεπιστημίου Κρήτης και του Ιδρύματος Τεχνολογίας και Έρευνας.

Πρώτη Φάση

Σε αυτή τη φάση χρησιμοποιήθηκε μια πρώιμη έκδοση του εργαλείου αποσυγχρονισμού *drdesync*. Η έκδοση αυτή δεν υποστήριζε χειροκίνητη ομαδοποίηση, αλλά μόνο αυτοματοποιημένη, με την χρήση του αλγορίθμου χαμηλής προσπάθειας. Επειδή όμως οι βαθμίδες του IND1 χρησιμοποιούν προώθηση (βλέπε Σχήμα Γ'.18), το αποτέλεσμα ήταν να δημιουργηθεί μια αποσυγχρονισμένη έκδοση του IND1a με μία μόνο ομάδα αποσυγχρονισμού.

Επίσης, στη φάση αυτή δεν υπήρχε πρόσβαση σε κανένα testbench του IND1a, οπότε δεν μπορούσε να γίνει έλεγχος ορθότητας του αποσυγχρονισμένου κυκλώματος. Έγιναν όμως κάποιες μετρήσεις για εμβαδόν, τόσο στη σύγχρονη όσο και στην αποσυγχρονισμένη έκδοση.

Δεύτερη Φάση

Στη δεύτερη φάση χρησιμοποιήθηκε μια νεότερη έκδοση του εργαλείου *drdesync*. Η έκδοση αυτή υποστήριζε επιπλέον, εκτός από τον αλγόριθμο “χαμηλής προσπάθειας”, και χειροκίνητη ομαδοποίηση.

Δημιουργήθηκαν, εκ νέου, τρεις αποσυγχρονισμένες εκδόσεις του IND1a. Οι εκδόσεις αυτές αποτελούνταν, αντίστοιχα, από μία, 10 και 33 ομάδες αποσυγχρονισμού. Η έκδοση με τη μία ομάδα αποσυγχρονισμού ήταν διαφορετική από την αντίστοιχη έκδοση της πρώτης φάσης, και δημιουργήθηκε για έλεγχο ορθότητας. Όσον αφορά την ομαδοποίηση των άλλων εκδόσεων, αυτή έγινε με τυχαίο τρόπο και με μοναδικό κριτήριο το μέγεθος των ομάδων, το οποίο έπρεπε να είναι παρόμοιο.

Επίσης, στη φάση αυτή δόθηκε πρόσβαση σε ένα testbench του IND1a, το οποίο χρησιμοποιούσε το δειγματοληπτικό πρόγραμμα Dhrystone [12]. Η σύνδεση των αποσυγχρονισμένων εκδόσεων με τα σύγχρονα τμήματα του testbench έγινε με τη μέθοδο που περιγράφεται στην Ενότητα 3.3.2. Με αυτόν τον τρόπο, τα σήματα χειραψίας του αποσυγχρονισμένου IND1a τροφοδοτούσαν με ρολόι τα υπόλοιπα σύγχρονα τμήματα.

Από την εκτέλεση των προσομοιώσεων με το testbench, διαπιστώθηκε η μη ορθή λειτουργία των αποσυγχρονισμένων κυκλωμάτων. Η ανάλυση των αιτιών έγινε στην τρίτη φάση.

Τρίτη Φάση – Τελική Επαλήθευση

Στην τρίτη φάση δεν δημιουργήθηκε κάποια καινούρια αποσυγχρονισμένη έκδοση του IND1a. Αντίθετα, έγινε προσομοίωση των αποσυγχρονισμένων εκδόσεων των προηγούμενων φάσεων. Η εκτέλεση των προσομοιώσεων έγινε με το *Mentor ModelSim SE*, ενώ για την εξαγωγή των αρχείων SDF (Standard Delay Format) χρησιμοποιήθηκε το *Synopsys PrimeTime*.

Αρχικά, έγινε έλεγχος των προσομοιώσεων της δεύτερης φάσης. Σε αυτές διαπιστώθηκαν ορισμένα προβλήματα, τα οποία σχετίζονταν με τα αρχεία SDF. Τα προβλήματα αυτά ήταν τα ακόλουθα:

- Η εξαγωγή του SDF αρχείου για την έκδοση με τη μία ομάδα αποσυγχρονισμού είχε γίνει λανθασμένα. Συγκεκριμένα, είχε γίνει εξαγωγή μόνο για τη συνδυαστική λογική και τους μανταλωτές του κυκλώματος. Το αποτέλεσμα ήταν να μην συμπεριλαμβάνεται καμία πληροφορία χρονισμού για το ασύγχρονο δίκτυο ελέγχου.
- Οι προσομοιώσεις των κυκλωμάτων της δεύτερης φάσης γίνονταν στο *Mentor ModelSim*, με την εντολή *vsim*. Σε όλες τις περιπτώσεις γινόταν χρήση του ορίσματος *+notimingchecks*, το οποίο απενεργοποιεί τον έλεγχο για τις χρονικές παραβιάσεις τύπου *setup* και *hold* [6]. Η αφαίρεση του συγκεκριμένου ορίσματος είχε ως αποτέλεσμα την εμφάνιση μηνυμάτων λάθους.

Εκτός από τον έλεγχο των προσομοιώσεων της δεύτερης φάσης έγιναν και καινούριες, με ορισμένες αλλαγές. Οι αλλαγές αυτές ήταν οι εξής:

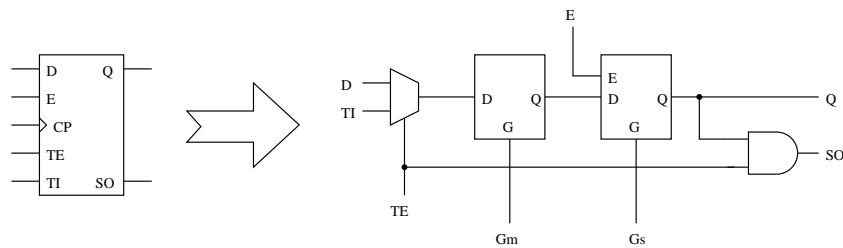
1. **Αλλαγή του τρόπου σύνδεσης των κυκλωμάτων:** Ο τρόπος σύνδεσης των αποσυγχρονισμένων κυκλωμάτων με τα σύγχρονα τμήματα του testbench άλλαξε και χρησιμοποιήθηκε η διεπαφή, η οποία περιγράφεται στην Ενότητα 3.3.3. Ο στόχος αυτής της αλλαγής ήταν να υπάρχει η δυνατότητα καθορισμού της περιόδου λειτουργίας, με την αλλαγή της συχνότητας του ρολογιού.

2. **Αύξηση του μεγέθους των στοιχείων καθυστέρησης:** Σε όλα τα κυκλώματα αυξήθηκε το μέγεθος των στοιχείων καθυστέρησης. Ο σκοπός αυτής της αλλαγής ήταν να αντιμετωπιστεί κάποιο πιθανό σφάλμα του *drdesync*, το οποίο θα είχε συμβεί κατά τον υπολογισμό τους.

3. **Αλλαγή του τρόπου αρχικοποίησης:** Βλέπε Ενότητα 3.2.

Από την εκτέλεση των προσομοιώσεων διαπιστώθηκε ότι τα αποσυγχρονισμένα κυκλώματα είχαν συγκεκριμένα προβλήματα. Αυτά ήταν τα εξής:

- **Λανθασμένος τρόπος αντικατάστασης καταχωρητών:** Το *drdesync* αντικαθιστούσε με λάθος τρόπο τους καταχωρητές, οι οποίοι είχαν είσοδο *enable*. Συγκεκριμένα, εισήγαγε στη θέση τους δύο μανταλωτές, εκ των οποίων ο σκλάβος είχε *enable*. Ο συγκεκριμένος τρόπος αντικατάστασης φαίνεται στο Σχήμα 6.1. Το πρόβλημα οφειλόταν στη διαφορετική λειτουργία του μανταλωτή της συγκεκριμένης βιβλιοθήκης τεχνολογίας: όταν το *enable* ήταν 0, η έξοδος του γινόταν z.
- **Χαμένες έξοδοι:** Τόσο στην αποσυγχρονισμένη έκδοση της πρώτης φάσης, όσο και στην αποσυγχρονισμένη έκδοση με τη μία ομάδα αποσυγχρονισμού της δεύτερης φάσης, έλειπαν κάποιες έξοδοι.
- **Λανθασμένος χειρισμός της φραγής ρολογιού:** Βλέπε Παράρτημα Γ'3.



Σχήμα 6.1: Λανθασμένος τρόπος αντικατάστασης των καταχωρητών με *enable*.

Εξαιτίας των προηγούμενων προβλημάτων, έπρεπε να γίνει εκ νέου αποσυγχρονισμός του IND1a. Αντί αυτού τελικά πραγματοποιήθηκε αποσυγχρονισμός του νέου IND1b, ο οποίος παρουσιάζεται στην επόμενη Ενότητα. Ο μοναδικός λόγος ήταν η νεότερη κατασκευαστική διεργασία του IND1b, η οποία ήταν στα 90nm. Το κύκλωμα αυτό ήταν πλέον διαθέσιμο από την X_Company_X και παρουσίαζε μεγαλύτερο ενδιαφέρον από άποψη αποτελεσμάτων.

6.2 Επεξεργαστής IND1b

6.2.1 Σύντομη Περιγραφή του Κυκλώματος

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.12.1.

6.2.2 Αποσυγχρονισμός του IND1b

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.12.2.

6.2.3 Προσομοίωση και Επαλήθευση

Στην ενότητα αυτή περιγράφεται η μέθοδος, η οποία χρησιμοποιήθηκε για τον έλεγχο της ορθότητας των αποσυγχρονισμένων εκδόσεων του IND1b. Αυτή ήταν η προσομοίωση των κυκλωμάτων με τη χρήση δεδομένου testbench. Δυστυχώς, δεν ήταν εφικτός κάποιος έλεγχος μέσω δοκιμής επαλήθευσης (formal verification). Ο λόγος είναι ότι δεν υπάρχει μέχρι σήμερα ολοκληρωμένο εργαλείο, το οποίο να υποστηρίζει δοκιμή επαλήθευσης σε αποσυγχρονισμένα κυκλώματα.

Η εκτέλεση των προσομοιώσεων έγινε με το *Mentor ModelSim SE*.

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.12.3.

6.3 Μικροελεγκτής IND2

6.3.1 Σύντομη Περιγραφή του Κυκλώματος

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.13.1.

6.3.2 Αποσυγχρονισμός του IND2

Ο αποσυγχρονισμός του IND2 έγινε με το εργαλείο *Nanosync*. Συγκεκριμένα, χρησιμοποιήθηκε μια νεότερη έκδοση από αυτήν του IND1b, η οποία υποστήριζε τον αλγόριθμο “μεσαίας προσπάθειας”, καθώς και το βήμα της χρονικής βελτιστοποίησης (βλέπε Ενότητες 4.3 και 4.8 αντίστοιχα).

Εκτός από το *Nanosync*, χρησιμοποιήθηκαν τα εργαλεία *Synopsys Design Compiler* και *Synopsys PrimeTime*. Το πρώτο χρησιμοποιήθηκε στις περιπτώσεις αλλαγής της ιεραρχίας, ενώ το δεύτερο στις περιπτώσεις εξαγωγής αρχείων SDF.

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.13.2.

6.3.3 Προσομοίωση και Επαλήθευση

Ο έλεγχος της ορθότητας των αποσυγχρονισμένων εκδόσεων έγινε με την προσομοίωση σε testbench, όπως και στην περίπτωση του IND1. Για την εκτέλεση των προσο-

μοιώσεων δεν χρησιμοποιήθηκε το *Mentor ModelSim* αλλά το περιβάλλον *VerilogXL - Simvision* της εταιρίας Cadence. Η αιτία ήταν το testbench που χρησιμοποιήθηκε και τα έτοιμα scripts, τα οποία αυτό περιελάμβανε. Τα scripts αυτά, έκαναν μεταγλώττιση και προσομοίωση, χρησιμοποιώντας τα συγκεκριμένα εργαλεία.

Το θέμα αυτό εξετάζεται στο Παράρτημα Γ'.13.3.

6.4 Περίληψη

Στο παρόν κεφάλαιο εξετάσαμε τη διαδικασία αποσυγχρονισμού τριών βιομηχανικών κυκλωμάτων. Η προσπάθεια αποσυγχρονισμού του επεξεργαστή IND1a φανέρωσε σημαντικά ζητήματα, τα οποία δεν είχαν εξεταστεί έως τώρα, όπως για παράδειγμα αυτό της αντιμετώπισης της φραγής ρολογιού. Η επίλυση των ζητημάτων αυτών υπήρξε σημαντική για τον επιτυχή αποσυγχρονισμό του επόμενου κυκλώματος, που ήταν ο επεξεργαστής IND1b. Όσον αφορά τον αποσυγχρονισμό του μικροελεγκτή IND2, αντιμετωπίστηκαν και εκεί ορισμένα προβλήματα. Κάποια από αυτά λύθηκαν, όπως λ.χ. το πρόβλημα με την χρονική απόκλιση σημάτων ρολογιού, ενώ κάποια άλλα δεν λύθηκαν, εξαιτίας του περιορισμένου διαθέσιμου χρόνου που υπήρχε.

Κεφάλαιο 7

Πειράματα και Αποτελέσματα

Σε αυτό το κεφάλαιο παρουσιάζονται οι μετρήσεις, οι οποίες έγιναν στις αποσυγχρονισμένες εκδόσεις του επεξεργαστή IND1b και του μικροελεγκτή IND2. Συγκεκριμένα, στην περίπτωση του IND1b, εξετάζεται το αποτέλεσμα του αποσυγχρονισμού, όσον αφορά την περίοδο λειτουργίας, την κατανάλωση, την ηλεκτρομαγνητική εκπομπή και το εμβαδόν. Στην περίπτωση του IND2, δεν κατέστη δυνατή η προσομοίωση των αποσυγχρονισμένων εκδόσεων, οπότε εξετάζεται μόνο η επιβάρυνση σε εμβαδόν.

7.1 Επεξεργαστής IND1b

Σε ορισμένες από τις περιπτώσεις που ακολουθούν, τα αποτελέσματα αφορούν εκδόσεις κυκλωμάτων, στις οποίες έγινε τοποθέτηση και διασύνδεση μέσω του εργαλείου *Cadence Encounter*. Τα scripts που χρησιμοποιήθηκαν για τον σκοπό αυτό, δίνονται στο Παράρτημα Α'5.

7.1.1 Περίοδος Λειτουργίας

Είναι γνωστό ότι η περίοδος κάθε κυκλώματος εξαρτάται από τρεις παράγοντες: την διεργασία, την τάση και την θερμοκρασία. Αυτό ισχύει τόσο στα σύγχρονα, όσο και στα αποσυγχρονισμένα κυκλώματα.

Στην περίπτωση ενός σύγχρονου κυκλώματος, η περίοδος του ρολογιού θα πρέπει να είναι τέτοια, ώστε να εγγυάται τη σωστή λειτουργία του, ακόμη και στις πιο ακραίες συνθήκες. Για τον λόγο αυτό, ισούται με την περίοδο λειτουργίας του κυκλώματος, η οποία αντιστοιχεί στη συνθήκη χαρακτηρισμού WC (Worst Case – Χείριστη Περίπτωση) της βιβλιοθήκης τεχνολογίας.

Αντίθετα, ένα αποσυγχρονισμένο κύκλωμα έχει τη δυνατότητα να προσαρμόζει την περίοδο λειτουργίας του, ανάλογα με τις υπάρχουσες συνθήκες. Οπότε, στις περισσότερες περιπτώσεις, θα λειτουργεί σε συνθήκες, οι οποίες θα αντιστοιχούν κοντά στη

συνθήκη χαρακτηρισμού TYP (Typical Case – Τυπική Περίπτωση) της βιβλιοθήκης τεχνολογίας. Παρόλα αυτά, το ασύγχρονο δίκτυο ελέγχου προσθέτει μια επιπλέον χρονική επιβάρυνση λόγω του συγχρονισμού μεταξύ των ελεγκτών. Το αποτέλεσμα είναι ότι, στις ίδιες συνθήκες, το αποσυγχρονισμένο κύκλωμα μπορεί να έχει μεγαλύτερη περίοδο από το σύγχρονο.

Οι μετρήσεις που αφορούν την περίοδο λειτουργίας του επεξεργαστή IND1b παρουσιάζονται στο Παράρτημα Γ'.14.1.

7.1.2 Κατανάλωση

Οι μετρήσεις που αφορούν την κατανάλωση του επεξεργαστή IND1b παρουσιάζονται στο Παράρτημα Γ'.14.2.

7.1.3 Ηλεκτρομαγνητική Εκπομπή

Η κατανάλωση ενέργειας των αποσυγχρονισμένων κυκλωμάτων είναι κατανομημένη στον χρόνο, σε αντίθεση με την κατανάλωση των σύγχρονων κυκλωμάτων, η οποία συσσωρεύεται σε κάθε θετική και αρνητική ακμή του ρολογιού. Για το λόγο αυτό τα αποσυγχρονισμένα κυκλώματα μπορούν να παρουσιάσουν χαμηλότερες τιμές ηλεκτρομαγνητικής παρεμβολής (ElectroMagnetic Interference – EMI).

Οι μετρήσεις που αφορούν την ηλεκτρομαγνητική εκπομπή του επεξεργαστή IND1b παρουσιάζονται στο Παράρτημα Γ'.14.3.

7.1.4 Εμβαδόν

Το βασικό μειονέκτημα των αποσυγχρονισμένων κυκλωμάτων, σε σχέση με τα σύγχρονα, είναι το μεγαλύτερο εμβαδόν. Τα βήματα αποσυγχρονισμού τα οποία είναι υπεύθυνα για την αύξηση εμβαδού είναι τα εξής:

1. Αντικατάσταση των καταχωρητών από ένα ζεύγος μανταλωτών.
2. Εισαγωγή στοιχείων καθυστέρησης.
3. Εισαγωγή ελεγκτών.

Η επιβάρυνση, η οποία προκαλείται στο πρώτο βήμα, είναι σταθερή για ένα κύκλωμα και ανεξάρτητη από την ομαδοποίηση. Αντίθετα, αυτό δεν ισχύει στα άλλα δύο βήματα.

Οι μετρήσεις που αφορούν το εμβαδόν του επεξεργαστή IND1b παρουσιάζονται στο Παράρτημα Γ'.14.4.

7.2 Μικροελεγκτής IND2

Όπως αναφέρθηκε στην Ενότητα 6.3.3, δεν κατέστη δυνατή η προσομοίωση των α-ποσυγχρονισμένων εκδόσεων του μικροελεγκτή IND2. Για το λόγο αυτό εξετάστηκε μόνο η επιβάρυνση σε εμβαδόν.

7.2.1 Εμβαδόν

Οι μετρήσεις που αφορούν το εμβαδόν του μικροελεγκτή IND2 παρουσιάζονται στο Παράρτημα Γ'.15.1.

Κεφάλαιο 8

Συμπεράσματα και Μελλοντική Εργασία

Η παρούσα εργασία έδειξε ότι ο αποσυγχρονισμός μπορεί να εφαρμοστεί με επιτυχία ακόμα και σε βιομηχανικά κυκλώματα αυξημένης πολυπλοκότητας. Τα ζητήματα τα οποία παρουσιάστηκαν, αντιμετωπίστηκαν με τον κατάλληλο τρόπο, είτε αυτά αφορούσαν ιδιαίτερα χαρακτηριστικά και τεχνικές σχεδίασης των κυκλωμάτων, είτε ιδιαιτερότητες της αυτοματοποιημένης ροής. Τέλος, τα αποτελέσματα των μετρήσεων επιβεβαίωσαν ότι και στα συγκεκριμένα κυκλώματα βιομηχανικής κλίμακας ισχύουν τα γνωστά πλεονεκτήματα και μειονεκτήματα της μεθόδου του αποσυγχρονισμού.

Από το σύνολο των προβλημάτων που προέκυψαν, δύο από αυτά έμειναν χωρίς κάποιο τρόπο αντιμετώπισης. Και τα δύο αφορούσαν τον μικροελεγκτή IND2 και ήταν τα εξής:

- Η μεγάλη επιβάρυνση εμβαδού στο αποσυγχρονισμένο κύκλωμα (βλέπε Ενότητα 7.2.1), η οποία οφείλεται στο μέγεθος των κρίσιμων μονοπατιών.
- Το πρόβλημα με το μεταβλητό ρολόι του testbench (βλέπε Ενότητα Γ'.6.3 και Ενότητα 6.3.3).

Τα παραπάνω προβλήματα δεν λύθηκαν, εξαιτίας της σύντομης χρονικής περιόδου, η οποία ήταν διαθέσιμη για την ενασχόληση με το συγκεκριμένο κύκλωμα.

Η μελλοντική εργασία περιλαμβάνει εξέταση ανάλογων περιπτώσεων, με αυτές του κυκλώματος και του testbench του IND2. Αυτές οι περιπτώσεις αφορούν κυκλώματα με σχετικά μεγάλα κρίσιμα μονοπάτια και testbench με μεταβλητό ρολόι. Τέλος, άλλα θέματα τα οποία επιδέχονται περισσότερη εξέταση είναι:

- αποτελεσματικότεροι αλγόριθμοι ομαδοποίησης,
- μέθοδοι formal verification για αποσυγχρονισμένα κυκλώματα,

- φραγή ρολογιού σε επίπεδο χειραψιών και όχι μανταλωτών,
- αλγόριθμοι τροποποίησης του μήκους των στοιχείων καθυστέρησης για την μείωση των ηλεκτρομαγνητικών εκπομπών.

Βιβλιογραφία

- [1] N. Andrikos, “A fully-automated desynchronization flow for synchronous circuits,” Master’s Thesis, Computer Science Department - University of Crete, Feb. 2006.
- [2] I. Blunno, J. Cortadella, A. Kondratyev, L. Lavagno, K. Lwin, and C. Sotiriou, “Handshake protocols for de-synchronization,” In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 149–158, 2004.
- [3] J. Cortadella, A. Kondratyev, L. Lavagno, and C. Sotiriou, “De-synchronization: synthesis of asynchronous circuits from synchronous specifications,” *IEEE Transactions on Computer-Aided Design*, vol. 25, no. 10, pp. 1904–1921, Oct. 2006.
- [4] S. B. Furber and P. Day, “Four-phase micropipeline latch control circuits,” *IEEE Transactions on VLSI Systems*, 4(2):247–253, June 1996.
- [5] P. L. Guernic, J.-P. Talpin, and J.-C. L. Lann, “Polychrony for system design,” *Journal of Circuits, Systems and Computers*, Apr. 2003.
- [6] Mentor Graphics, ModelSim SE Command Reference.
- [7] David E. Muller and W. S. Bartky, “A theory of asynchronous circuits,” In *Proceedings of an International Symposium on the Theory of Switching*, pages 204–243, Harvard University Press, April 1959.
- [8] D. Panyasak, G. Sicard, M. Renaudin, “A current shaping methodology for lowering EM disturbances in asynchronous circuits,” *Microelectronics journal* 35, June-2004, pp. 531–540.
- [9] J. Sparsø and S. Furber, “Principles of Asynchronous Circuit Design,” Kluwer Academic Publishers, 2001.
- [10] I.E. Sutherland, “Micropipelines,” *Communications of the ACM*, 32(6):720–738, June 1989.

- [11] Synopsys, Power Compiler User Guide, Chapter 7.
- [12] Reinhold P. Weicker, “Dhrystone: A Synthetic Systems Programming Benchmark,” *Communications of the ACM* 27, 10 (Oct. 1984), 1013–1030.