



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Υπολογιστικών Συστημάτων

Αλγόριθμοι Ελαχιστοποίησης Μη Πλήρως Ορισμένων
Λογικών Συναρτήσεων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του

Μάριου Ι. Καλαθά

Επιβλέπων Καθηγητής: Γ. Παπακωνσταντίνου

Αθήνα, Ιανουάριος 2011

.....
Μάριος Καλαθάς
Υποψήφιος Διδάκτωρ Ε.Μ.Π.

©Μάριος Καλαθάς, 2011
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματός της για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται στο συγγραφέα.

Οι απόψεις και τα συμπεράσματα που βρίσκονται σε αυτό το κείμενο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Στη Δήμητρα

Πρόλογος

Η παρούσα διδακτορική διατριβή, που εκπονήθηκε στο Εργαστήριο Υπολογιστικών Συστημάτων του Εθνικού Μετσόβιου Πολυτεχνείου, υπήρξε μια ευκαιρία να εξοικωθώ με τη σύγχρονη ερευνητική διαδικασία και να διευρύνω τους πνευματικούς μου ορίζοντες. Παράλληλα είχα τη δυνατότητα να διαπιστώσω τη συμβολή της επιστημονικής σκέψης και έρευνας στην επίλυση πολύπλοκων ζητημάτων, που βρίσκονται στην αιχμή της τεχνολογίας.

Ο τομέας της λογικής σχεδίασης, τον οποίο πραγματεύεται η παρούσα διατριβή, βρίσκει πλήθος πρακτικών εφαρμογών, καθώς τα ψηφιακά συστήματα έχουν κατακλύσει την καθημερινότητά μας. Επίσης παρουσιάζει και σημαντικό προσωπικό ενδιαφέρον και αισθάνομαι ευγνώμων για την ενασχόληση μου με το πεδίο αυτό.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Γεώργιο Παπακωνσταντίνου για την αδιάλειπτη συμπαράστασή του και την εμπιστοσύνη που έδειξε στο πρόσωπό μου. Οι ιδέες του, οι υποδείξεις του και η βοήθεια του υπήρξαν καθοριστικές για την ολοκλήρωση αυτής της διατριβής. Επίσης με το πάθος του για την έρευνα, το ήθος του και την προσωπική υποστήριξή του, αποτέλεσε υπόδειγμα και πολύτιμο αρωγό σε αυτήν την προσπάθεια.

Θεωρώ σημαντική εμπειρία την ενασχόλησή μου με την εκπαιδευτική διαδικασία και τις δραστηριότητες του Εργαστηρίου Υπολογιστικών Συστημάτων. Ήρθα, μάλιστα, σε επαφή με πολλούς ερευνητές, με τους οποίους είχα τη δυνατότητα να ανταλλάξω απόψεις για επιστημονικά ζητήματα, αλλά και να αναπτύξω διαπροσωπικές σχέσεις φιλίας. Ευχαριστώ όλους τους υποψήφιους διδάκτορες, τους καθηγητές και τις γραμματείς του Εργαστηρίου, και ιδιαίτερα τους Δ. Βουδούρη και Μ. Σαμσών, που ήταν τα μέλη της ερευνητικής μου ομάδας, με τους οποίους συνεργαστήκαμε άψογα.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου για την αμέριστη συμπαράστασή τους και την υποστήριξή τους, αλλά και τη σύντροφό μου που με βοήθησε να αντιμετωπίσω τις δύσκολες και κουραστικές στιγμές με χαμόγελο.

Περιεχόμενα

1	Εισαγωγή	1
1.1	Εκφράσεις ESOP	1
1.2	Εκφράσεις ESCT	4
1.3	Κβαντικοί υπολογιστές	6
2	Λογικές Συναρτήσεις και εκφράσεις ESOP	9
2.1	Λογικές συναρτήσεις πολλαπλών τιμών	9
2.2	Λογικές εκφράσεις XOR-αθροίσματος γινομένων	10
2.3	Αναπτύγματα και αναπαραστάσεις boolean συναρτήσεων	11
2.4	Συναρτήσεις πολλών εξόδων και χαρακτηριστική συνάρτηση	12
2.5	Μη πλήρως ορισμένες λογικές συναρτήσεις	13
3	Ευριστική ESOP ελαχιστοποίηση μη πλήρως ορισμένων λογικών συναρτήσεων	16
3.1	Αναπαράσταση των μη πλήρως ορισμένων συναρτήσεων	16
3.2	Αποσύνθεση	17
3.3	Παραγωγή των εκφράσεων MVESOP των φύλλων	18
3.4	Σύνθεση	19
3.5	Ο αλγόριθμος DCMIN	22
3.6	Ο αλγόριθμος QuickDCMIN	23
3.7	Πειραματικά αποτελέσματα	27
4	Ακριβείς εκφράσεις ESOP για μη πλήρως ορισμένες συναρτήσεις	31
4.1	Θεωρητικό υπόβαθρο	31
4.2	Αποθήκευση σε πίνακα των βαρών όλων των συναρτήσεων με 5 μεταβλητές εισόδου	35
4.3	Εύρεση μιας ακριβούς έκφρασης	36
4.4	Υλοποίηση	40
4.4.1	Σύγκριση με ακριβείς μεθόδους για πλήρως ορισμένες συναρτήσεις	41
4.4.2	Σύγκριση με ευριστικές μεθόδους για μη πλήρως ορισμένες συναρτήσεις	42
4.4.3	Σύγκριση με εξανηλητικές μεθόδους	43

4.4.4	Χρήση του Αλγορίθμου 10 για περισσότερες από 6 μεταβλητές	44
5	Εκφράσεις ESCT	45
5.1	Σύνθετοι Όροι και εκφράσεις ESCT	45
5.2	Περιορισμένοι σύνθετοι όροι	47
5.3	Πράξεις με κύτταρα Maitra	50
5.4	Αντιστρέψιμες Λογικές Πύλες	52
6	Ελαχιστοποίηση συναρτήσεων πολλών εξόδων	54
6.1	1η Προσέγγιση	54
6.2	2η Προσέγγιση	57
6.3	Πειραματικά αποτελέσματα	58
7	Ευριστική ESCT ελαχιστοποίηση μη πλήρως ορισμένων λογικών συναρτήσεων	60
7.1	Αναπαράσταση μη πλήρως ορισμένων συναρτήσεων	61
7.2	Αποσύνθεση	61
7.3	Παραγωγή των εκφράσεων των φύλλων	62
7.4	Σύνθεση	63
7.5	Ο αλγόριθμος ESCT-DC	68
7.6	Ο αλγόριθμος QuickESCT-DC	71
7.7	Πειραματικά αποτελέσματα	72
8	Κβαντικοί Υπολογιστές	78
8.1	Στοιχεία άλγεβρας	78
8.1.1	Ορθομοναδιαίοι πίνακες	78
8.1.2	Χώρος Hilbert	78
8.2	Βασικές έννοιες κβαντικών υπολογισμών	80
8.3	Ο αλγόριθμος του Grover	84
9	Ένας κβαντικός αλγόριθμος για την εύρεση ελάχιστων εκφράσεων αποκλειστικού-ή για μη πλήρως ορισμένες Boolean συναρτήσεις	88
9.1	Θεωρητικό Υπόβαθρο	88
9.1.1	ESOP και ESCT ελαχιστοποίηση	88
9.1.2	Ο αλγόριθμος του Grover	89
9.2	Αλγόριθμοι	90
9.2.1	Προηγούμενη δουλειά	90
9.2.2	DCQMin	91
10	Συμπεράσματα και μελλοντικές κατευθύνσεις	96

Κατάλογος Σχημάτων

1.1	Κυτταρική αρχιτεκτονική Maitra	8
3.1	Γεννήτορ δένδρο του Παραδείγματος 16	18
6.1	Κυτταρική δομή για την απεικόνιση της χαρακτηριστικής συνάρτησης (1η προσέγγιση).	55
6.2	Ανανεωμένη κυτταρική δομή για την απεικόνιση της χαρακτηριστικής συνάρτησης (1η προσέγγιση)	56
6.3	Κυτταρική δομή για την απεικόνιση της χαρακτηριστικής συνάρτησης (2η προσέγγιση).	58
7.1	Γεννήτορ δένδρο του Παραδείγματος 34	61
8.1	Η κατάσταση ενός τυχαίου qubit $ R\rangle$ απεικονιζόμενη σε μια σφαίρα Bloch. θ είναι η γωνία με τον άξονα ξ και ϕ η γωνία της προβολής του $ R\rangle$ στο επίπεδο xy με τον άξονα x . Το $ L\rangle$ είναι το qubit: $(0\rangle + 1\rangle)/2^{1/2}$	82
8.2	Διδιάστατη απεικόνιση ενός qubit	82
8.3	Ο αλγόριθμος του Grover	87
9.1	Κύκλωμα QMin-Oracle	90
9.2	Ιεραρχία του αλγορίθμου DCQmin	94
9.3	Ένας 2-qubit Comparator, που συγκρίνει το s_0s_1 με το t_0t_1 , υλοποιώντας τη συνάρτηση: $(s_1 \oplus t_1)t_1 \oplus (s_1 \oplus t_1)(s_0 \oplus t_0)t_0$	95
9.4	Κύκλωμα DCQmin-Oracle με παράδειγμα	95

Κατάλογος Πινάκων

1.1	Πλήθος κύβων ορισμένων κλάσεων συναρτήσεων	2
2.1	Μη πλήρως ορισμένη λογική συνάρτηση	14
3.1	Η συνάρτηση του Παραδείγματος 15	17
3.2	Παραγωγή των εκφράσεων MVESOP των φύλλων	19
3.3	Τύποι όταν μια υποσυνάρτηση είναι μηδέν	21
3.4	Τύποι όταν δύο υποσυναρτήσεις είναι μηδέν	21
3.5	Εχμετάλλευση των αδιάφορων όρων για εκφράσεις ESOP (1ος τρόπος λειτουργίας)	28
3.6	Εχμετάλλευση των αδιάφορων όρων για εκφράσεις ESOP (2ος τρόπος λειτουργίας)	29
3.7	Σύγκριση του QuickDCMIN με τον MINT, τον EXORCSIM-MV-2 και τον EXORCISM-MV-3 (1ος τρόπος λειτουργίας)	30
3.8	Σύγκριση του QuickDCMIN με τους MINT, EXORCSIM-MV-2 και EXORCISM-MV-3 (2ος τρόπος λειτουργίας)	30
4.1	Πίνακας RT για τη συνάρτηση του Παραδείγματος 21	32
4.2	Πίνακας RT_{DC} για τη συνάρτηση του Παραδείγματος 21	33
4.3	Πίνακας RT για το Παράδειγμα 25	39
4.4	Πίνακας RT_{DC}^1 για το Παράδειγμα 25	39
4.5	Βάρη των τυχαία παραγόμενων συναρτήσεων	41
4.6	Πειραματικά αποτελέσματα συγκρίνοντας τους Αλγόριθμους 9 και 10 με τους αλγόριθμους $min_tau(f)$ και <i>QuickDCMIN</i>	42
4.7	Πειραματικά αποτελέσματα συγκρίνοντας τους Αλγόριθμους 9 και 10 με τον Εξαντλητικό Αλγόριθμο	44
5.1	Πλήρες σύνολο συναρτήσεων για σύνθετους όρους	48
5.2	Συγχώνευση κελιών	50
5.3	Συγχώνευση όρων	51
5.4	Τυπικές αντιστρέψιμες λογικές πύλες	53
6.1	Μετροσυναρτήσεις πολλών εξόδων	59
7.1	Παραγωγή των εκφράσεων ESCT των φύλλων	63

7.2	Τύποι όταν μια υποσυνάρτηση είναι μηδέν ή ένα	66
7.3	Τύποι όταν δύο υποσυναρτήσεις είναι μηδέν ή ένα	67
7.4	Τύποι για τετριμμένες περιπτώσεις.	68
7.5	Εχμετάλλευση των αδιάφορων όρων για εκφράσεις ESCT (1ος τρόπος λειτουργίας)	74
7.6	Εχμετάλλευση των αδιάφορων όρων για εκφράσεις ESCT (2ος τρόπος λειτουργίας)	75
7.7	Σύγκριση του QuickESCT-DC (για εκφράσεις ESCT) με τον αλγόριθμο των Lee, Drechsler, Perkowski (1ος τρόπος λειτουργίας) .	76
7.8	Σύγκριση του QuickESCT-DC (για εκφράσεις ESCT) με τον αλγόριθμο των Lee, Drechsler, Perkowski (2ος τρόπος λειτουργίας) .	76

Δημοσιεύσεις

1. I. Drositis, T. Andronikos, M. Kalathas, G. Papakonstantinou, and N. Koziris. Optimal loop parallelization in n-dimensional index spaces. In *International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA'02*, Las Vegas, 2002.
2. T. Andronikos, M. Kalathas, F. M. Ciorba, P. Theodoropoulos, G. Papakonstantinou, and P. Tsanakas. Scheduling nested loops with the least number of processors. In *21st IAESTED International Conference on Applied Informatics*, pages 713-718, Innsbruck, Austria, 2003.
3. T. Andronikos, M. Kalathas, F. M. Ciorba, P. Theodoropoulos, and G. Papakonstantinou. An efficient scheduling of uniform dependence loops. In *The 6th Hellenic European Research on Computer Mathematics and its Applications Conference (HERCMA)*, Athens, Greece, 2003.
4. D. Voudouris, M. Kalathas, and G. Papakonstantinou. Decomposition of multi-output boolean functions. In *HERMIS Journal*, vol. 6, 2005, pages 154-161.
5. M. Kalathas, D. Voudouris, and G. Papakonstantinou. A heuristic algorithm to minimize esops for multiple-output incompletely specified functions. In *2006 ACM Great Lakes Symposium on VLSI - GLSVLSI'06*, pages 357-361, Philadelphia, PA, USA, April 30 - May 2, 2006.
6. M. Sampson, D. Voudouris, M. Kalathas and G. Papakonstantinou. A quantum algorithm for finding minimum exclusive-or expressions for incompletely specified Boolean functions. In *The 8th Hellenic European Research on Computer Mathematics and its Applications Conference (HERCMA)*, Athens, September 2007.
7. M. Kalathas, D. Voudouris and G. Papakonstantinou. ESCT minimization for incompletely specified functions. Accepted for publication in *Journal of Multiple-Valued Logic and Soft Computing*, 2010.
8. M. Sampson, M. Kalathas, D. Voudouris and G. Papakonstantinou. Exact ESOP expressions for incompletely specified functions. Submitted in *Integration, the VLSI Journal*, 2010.

Περίληψη

Στην παρούσα διδακτορική διατριβή εξετάζονται μέθοδοι ανάλυσης και βελτιστοποίησης λογικών εκφράσεων σε μορφή αποκλειστικού-ή αθροίσματος από γινόμενα (Exclusive-Or Sum of Products - ESOP) και αποκλειστικού-ή αθροίσματος από σύνθετους όρους (Exclusive-or Sum of Complex Terms - ESCT). Οι συναρτήσεις που μελετώνται περιέχουν αδιάφορους όρους (don't cares), πρόκειται δηλαδή για μη πλήρως ορισμένες συναρτήσεις (incompletely specified functions). Η βελτιστοποίησή τους έγκειται στην εύρεση ελάχιστων εκφράσεων, δηλαδή εκφράσεων με όσο το δυνατόν λιγότερους όρους.

Αξίζει να σημειωθεί ότι τα προβλήματα της ελαχιστοποίησης εκφράσεων ESOP και ESCT είναι από τη φύση τους εξαιρετικά δύσκολα, καθώς στη γενική μορφή τους είναι NP-hard. Η ύπαρξη αδιάφορων όρων δυσχεραίνει ακόμα περισσότερο την εύρεση ελάχιστων εκφράσεων.

Καταρχήν μας ενδιαφέρει η ελαχιστοποίηση εκφράσεων ESOP. Αρχικά προτείνεται ένας ευριστικός αλγόριθμος για την παραγωγή και την ελαχιστοποίηση τέτοιων εκφράσεων για τυχαίες μη πλήρως ορισμένες λογικές συναρτήσεις πολλών εξόδων. Αυτός ο αλγόριθμος χρησιμοποιεί αποσύνθεση συναρτήσεων και λογική πολλαπλών τιμών και η αποτελεσματικότητά του μετράται χρησιμοποιώντας συγκεκριμένες μετροσυναρτήσεις (benchmark functions). Όπως μπορούμε να δούμε από τα πειραματικά αποτελέσματα, παρουσιάζει καλύτερη απόδοση από προηγούμενους αλγόριθμους.

Στη συνέχεια της διατριβής, προτείνεται μια μέθοδος για την εύρεση μιας ακριβούς έκφρασης ESOP για τυχαίες μη πλήρως ορισμένες συναρτήσεις με 6 το πολύ μεταβλητές. Για να το επιτύχουμε αυτό, αποθηκεύουμε σε ένα συμπιεσμένο πίνακα το βάρος όλων των συναρτήσεων 5 μεταβλητών. Στην προσέγγισή μας χρησιμοποιούμε εκτεταμένα αυτόν τον πίνακα, ώστε να επιταχύνουμε το χρόνο εκτέλεσης. Με βάση όσα γνωρίζουμε, αυτός είναι ο πρώτος αλγόριθμος που αντιμετωπίζει αυτό το πρόβλημα.

Στη συνέχεια, παρουσιάζεται η δυνατότητα χρησιμοποίησης σύνθετων όρων (complex terms), για την παραγωγή εκφράσεων που είναι πιο συμπαγείς από τις αντίστοιχες ESOP (δηλαδή αποτελούνται από μικρότερο αριθμό όρων). Σε αυτή την περίπτωση δεν επιτρέπεται μόνο η πράξη AND μεταξύ δύο μεταβλητών, αλλά οποιαδήποτε λογική συνάρτηση. Οι εκφράσεις που παράγονται είναι οι προαναφερόμενες εκφράσεις ESCT. Ερευνάται, έπειτα, το πρόβλημα της αποσύνθεσης, απεικόνισης και ελαχιστοποίησης των λογικών συναρτήσεων πολλών εξόδων, σε μορφή εκφράσεων ESCT. Πρώτα παρουσιάζεται ένας μαθηματικός φορμαλισμός για τις λογικές συναρτήσεις μιας εξόδου και έπειτα επεκτείνεται για συναρτήσεις πολλών εξόδων.

Επίσης, παρουσιάζεται ένας καινούριος ευριστικός αλγόριθμος για την ελαχιστοποίηση εκφράσεων ESCT για μη πλήρως ορισμένες συναρτήσεις πολλών εξόδων. Ο αλγόριθμος αυτός αποτελεί επέκταση του προηγούμενου για τις εκφράσεις ESOP. Δίνονται πειραματικά αποτελέσματα, που υποδεικνύουν την ανωτερότητα

αυτού του αλγορίθμου, όταν συγκρίνεται με προηγούμενους, για μη πλήρως ορισμένες συναρτήσεις και την εκμετάλλευση των αδιάφορων όρων που επιτυγχάνεται.

Τέλος, η παρούσα διδακτορική διατριβή παρουσιάζει έναν χβαντικό αλγόριθμο για την εύρεση ελάχιστων εκφράσεων ESOP ή ESCT για τυχαίες μη πλήρως ορισμένες συναρτήσεις. Ο προτεινόμενος αλγόριθμος εκμεταλλεύεται τον έμφυτο μαζικό παραλληλισμό των χβαντικών κυκλωμάτων, ώστε να επιτύχει καλύτερη πολυπλοκότητα από τους συμβατικούς αλγόριθμους. Οι προτεινόμενες εκφράσεις αποκλειστικού-ή (XOR), όπως οι ESOP και οι ESCT, μπορούν να χρησιμοποιηθούν για να υλοποιήσουν μια τυχαία λογική συνάρτηση σε ένα αντιστρέψιμο ή ακόμα και σε ένα χβαντικό κύκλωμα.

Abstract

Various methods of analysis and optimization of logic expressions in the form of Exclusive-Or Sum of Products (ESOP) and Exclusive-or Sum of Complex Terms (ESCT) are presented in this PhD thesis. The functions considered contain don't care terms, namely they are incompletely specified functions. Their optimization concerns the finding of minimal expressions, by that means expressions with the least possible number of terms.

It should be highlighted that the problems of minimizing ESOP and ESCT expressions are by default extremely complex, since they are NP-hard. The existence of don't care terms complicates even more the finding of exact expressions.

Initially, we are interested in the minimization of ESOP expressions. First of all, we propose a heuristic algorithm for the generation and the minimization of such expressions for arbitrary multiple-output incompletely specified logic functions. This algorithm uses functional decomposition and multiple-valued logic and its efficiency is measured using standard benchmark functions. As we can observe from the experimental results, our algorithm presents with better performance compared to previous ones.

Afterwards, we propose a method which finds an exact ESOP expression for arbitrary incompletely specified functions of up to 6 variables. To achieve that, we store the weight of all functions of 5 variables in a compressed table. In our approach, we use this table extensively, in order to decrease the execution time. To the best of our knowledge, this is the first algorithm dealing with this problem.

Moreover, we present the possibility of using complex terms to generate more compact expressions (which are consisted of fewer terms) than the respective ESOP ones. In this case, not only the AND operation is permitted between two variables, but also any logic function. The produced expressions are the previously mentioned ESCT expressions. Additionally, we examine the problem of decomposition, mapping and minimization of multiple-output logic functions, in the form of ESCT expressions. Initially, a mathematical formulation for single output logic functions is presented, which is extended for multiple-output functions.

Furhermore, we propose a naive heuristic algorithm to minimize ESCT expressions for multiple-output incompletely specified functions. This algorithm is an extension of the previous one for ESOP expressions. The experimental results presented indicate both the superiority of our algorithm compared to previous ones for completely specified functions, as well as the exploitation of the presence of don't care terms achieved.

Finally, this PhD thesis presents a quantum algorithm to find exact ESOP or ESCT expressions for arbitrary incompletely specified functions. The proposed algorithm exploits the inherent massive parallelism of quantum circuits, in order to achieve better complexity than the conventional ones. The suggested XOR

expressions, such as ESOP and ESCT, might be used to implement an arbitrary logic function in a reversible, or even in a quantum circuit.

Κεφάλαιο 1

Εισαγωγή

1.1 Εκφράσεις ESOP

Όπως είναι γνωστό όλα τα λογικά κυκλώματα διέπονται από τους κανόνες της άλγεβρας του Boole. Η εξέλιξη της τεχνολογίας οδήγησε σε καλύτερα κυκλώματα (γρηγορότερα, μικρότερα και με λιγότερες θερμικές απώλειες). Όμως οι περιορισμοί που θέτει η ίδια η τεχνολογία αρχίζουν να γίνονται εμφανείς. Έτσι η έρευνα στον χώρο της λογικής σχεδίασης αποτελεί μια σημαντική ερευνητική περιοχή. Η συνεχής πρόοδος και οι αυξανόμενες απαιτήσεις στη σχεδίαση κυκλωμάτων πολύ μεγάλης κλίμακας ολοκλήρωσης (Very Large Scale Integration - VLSI) οδηγούν στην αποκλειστική χρήση εργαλείων CAD (Computer Aided Design) για την κατασκευή τέτοιων κυκλωμάτων.

Ένα από τα περισσότερα σημαντικά προβλήματα είναι η αποτελεσματική αποσύνθεση και ελαχιστοποίηση μιας λογικής συνάρτησης. Αυτή η συνάρτηση μπορεί να έχει πολλές εισόδους και πολλές εξόδους. Κάθε τέτοια συνάρτηση μπορεί να αποσυνθεθεί και να εκφραστεί με πολλούς διαφορετικούς τρόπους. Οι Ashenhurst [1], Curtis [7], Roth και Karp [64] έκαναν πρωτοποριακή δουλειά σε αυτό το πεδίο.

Για πολλά χρόνια η λογική σύνθεση βασιζόταν σε πύλες AND, OR και NOT. Τα περισσότερα εργαλεία λογικής σύνθεσης παράγουν εκφράσεις της μορφής αθροίσματος γινομένων (Sum Of Products - SOP) για μια τυχαία λογική συνάρτηση. (Εναλλακτικά ένα επίπεδο πυλών AND και ένα επίπεδο πυλών OR αντικαθίστανται από δύο επίπεδα πυλών NAND.) Οι εκφράσεις SOP έχουν μελετηθεί εκτενώς στο παρελθόν, καθώς διακρίνονται από την απλότητά τους και την ευκολία απεικόνισής τους σε αρχιτεκτονικές FPGA (Field Programmable Gate Array). Το πρόβλημα σε αυτές τις περιπτώσεις είναι η εύρεση των εκφράσεων με τους λιγότερους δυνατούς όρους, γιατί κάτι τέτοιο θα οδηγήσει στην κατασκευή ολοκληρωμένων κυκλωμάτων με όσο το δυνατόν μικρότερο μέγεθος.

Τα τελευταία χρόνια η έρευνα έχει στραφεί στις παρόμοιες εκφράσεις αποκλειστικού-ή αθροίσματος γινομένων (Exclusive-or Sum Of Products - ESOP). Παρόλο που η ελαχιστοποίηση των εκφράσεων ESOP είναι περισσότερο δυσχερές από ότι για τις αντίστοιχες SOP, εν τούτοις παρουσιάζουν ορισμένες ελκυστικές

ιδιότητες. Έτσι, για κάποια είδη εφαρμογών, όπως αριθμητικές, τηλεπικοινωνιακές ή εφαρμογές διόρθωσης λαθών, η χρήση πυλών XOR μπορεί να μειώσει την πολυπλοκότητα των λογικών κυκλωμάτων [73, 72]. Για μια συνάρτηση n μεταβλητών, το άνω φράγμα για τον αριθμό των κύβων των εκφράσεων ESOP είναι $29 \cdot 2^{n-7}$, $n > 6$ [21], συγκρινόμενο με το 2^{n-1} για τις εκφράσεις SOP. Στον Πίνακα 1.1 μπορούμε να δούμε μια σύγκριση σχετικά με τα άνω φράγματα των κύβων των SOP και των ESOP για μερικές συγκεκριμένες κλάσεις συναρτήσεων, όπως αυτά παρουσιάζονται στην αναφορά [71]. Μια ικανή συνθήκη μορφοποιήθηκε στις εργασίες [3, 15], που μας επιτρέπει να αποφασίσουμε αν μια δεδομένη συνάρτηση είναι δυνατόν να ελαχιστοποιηθεί περισσότερο αποτελεσματικά χρησιμοποιώντας XOR ελαχιστοποίηση.

Συναρτήσεις	SOP	ESOP
Τυχαίες	2^{n-1}	$29 \cdot 2^{n-7}, n > 6$
Συμμετρικές	2^{n-1}	$3^{n/2-1}$
Ισοτιμίας	2^{n-1}	n
n -bit Αθροιστές	$3 \cdot 2^{n+1} - 4n - 5$	$2^{n+1} - 1$
$x_1y_1 + \dots + x_ny_n$	n	$2^n - 1$

Πίνακας 1.1: Πλήθος κύβων ορισμένων κλάσεων συναρτήσεων

Οι εκφράσεις ESOP έχουν μελετηθεί σε έκταση στο παρελθόν. Πολλοί αλγόριθμοι και πολλές συστηματικές μέθοδοι έχουν προταθεί για τη δημιουργία τέτοιων εκφράσεων για τυχαίες λογικές συναρτήσεις. Επιπλέον, είναι συνήθως σημαντικό οι εκφράσεις που παράγονται να είναι όσο το δυνατόν μικρότερες, ώστε να οδηγούν σε μικρότερες υλοποιήσεις. Το πρόβλημα της ακριβούς ESOP ελαχιστοποίησης (δηλαδή της εύρεσης μιας έκφρασης ESOP με τον ελάχιστο δυνατό αριθμό κύβων) είναι εξαιρετικά δύσκολο. Οι πρακτικές λύσεις που έχουν προταθεί αφορούν συναρτήσεις μέχρι το πολύ 7 μεταβλητές [4, 53, 54, 70, 50, 21, 28, 87, 90, 26, 27]. Μια άλλη προσέγγιση είναι η ακριβής ελαχιστοποίηση συναρτήσεων οποιουδήποτε αριθμού εισόδων, αλλά με περιορισμούς στο βάρος της συνάρτησης, δηλαδή στον αριθμό των κύβων στην ελάχιστη αναπαράσταση [25, 50, 89, 91, 92]. Επιπλέον, πολλές ευριστικές προσεγγίσεις έχουν προταθεί, όπου δεν υπάρχει βεβαιότητα ότι η ευρισκόμενη λύση είναι η καλύτερη δυνατή [24, 73, 69, 71, 70, 85, 81, 105, 104, 78, 46, 61, 90, 91, 92, 88, 29, 84, 86, 94]. Ένας άλλος αλγόριθμος βρίσκει μια βέλτιστη ESOP έκφραση για μια πιθανά μερική (partial) boolean συνάρτηση [59], όπου το μέτρο της βελτιστότητας είναι ο αριθμός των literals. Εξαιτίας της δυσκολίας του προβλήματος της ESOP ελαχιστοποίησης, έχουν δημοσιευθεί πολλές ακριβείς και ευριστικές μέθοδοι για την ελαχιστοποίηση ορισμένων υποκλάσεων των ESOP εκφράσεων [74, 8, 11, 9, 14, 75, 102, 41, 12, 18, 16, 62, 2, 107].

Οι τεχνικές που χρησιμοποιούνται στους προτεινόμενους αλγόριθμους περιλαμβάνουν την αποσύνθεση της τυχαίας συνάρτησης στις υποσυναρτήσεις της [53, 54, 25, 21, 28, 89, 90, 91, 92, 88, 26, 27], την επαναληπτική εφαρμογή ποικίλων μετασχηματισμών κύβων σε ένα αρχικό κάλυμμα της συνάρτησης [24, 73, 8, 9, 46], την εφαρμογή διάφορων μετασχηματισμών στις συναρτήσεις που οδηγούν σε α-

πλούστερα προβλήματα ελαχιστοποίησης [69, 85, 29], την επίλυση γραμμικών συστημάτων [59], τον μετασχηματισμό τυποποιημένων κανονικών μορφών (canonical normal forms) [87, 84, 86], την εφαρμογή διαγραμμάτων ροής (flow charts) [102, 107], τη χρήση διάφορων τύπων διαγραμμάτων απόφασης (decision diagrams) [70, 74, 11, 14, 50, 75, 105, 104, 41, 2, 94], transeunt τριγώνων [16], γενετικών αλγορίθμων [12], look-up πινάκων [18], φασματικών μεθόδων [62], Sierpinski gaskets [61] κτλ. Μία καινοτόμος προσέγγιση είναι η χρήση εννοιών από τη θεωρία πληροφορίας, όπως η εντροπία Shannon, για να σχεδιαστεί μια στρατηγική βασισμένη στην εντροπία για την ESOP ελαχιστοποίηση [78].

Οι αριθμητικές εκφράσεις είναι μια εναλλακτική προσέγγιση για να περιγράψουμε τα λογικά κυκλώματα και μπορούν να θεωρηθούν σαν οι ακέραιες αντίστοιχες των εκφράσεων Reed-Muller, καθώς ορίζονται σε όρους της ίδιας βάσης, έχουν όμως μεταβλητές και τιμές συναρτήσεων που ερμηνεύονται ως ακέραιοι 0 και 1 αντί για λογικές τιμές [30]. Έχουν προταθεί κάποιες μέθοδοι [30, 83] για την βελτιστοποίηση αριθμητικών εκφράσεων.

Οι αλγόριθμοι στους οποίους αναφερθήκαμε παραπάνω έχουν σχεδιαστεί για να ελαχιστοποιούν πλήρως ορισμένες συναρτήσεις. Όμως οι περισσότερες πραγματικές εφαρμογές βασίζονται σε μη πλήρως ορισμένες συναρτήσεις, δηλαδή συναρτήσεις που περιέχουν αδιάφορους όρους. Η απεικόνιση μη πλήρως ορισμένων λογικών συναρτήσεων, μίας ή πολλών εξόδων, και η ελαχιστοποίηση των αντίστοιχων εκφράσεων ESOP είναι ακόμα μια ανοιχτή ερευνητική περιοχή. Η ύπαρξη των αδιάφορων όρων αυξάνει τη δυσκολία και την πολυπλοκότητα του προβλήματος, κάτι που καταδεικνύεται από το γεγονός ότι μόνο ευριστικοί αλγόριθμοι υπάρχουν για την ελαχιστοποίηση τέτοιου είδους συναρτήσεων.

Κάποιες προσπάθειες για την αναπαράσταση μη πλήρως ορισμένων συναρτήσεων σε συμπαγή μορφή, χρησιμοποιώντας διαφορετικές κατηγορίες διαγραμμάτων απόφασης, παρουσιάζονται στις εργασίες [31, 58, 57, 13, 19, 6, 44, 76, 17]. Το πρόβλημα της ESOP ελαχιστοποίησης τέτοιων συναρτήσεων διαπραγματεύεται στις εργασίες [56, 35, 81, 82, 55, 87, 32], όπου ορίζονται καινούριοι μετασχηματισμοί κύβων που εκμεταλλεύονται τους αδιάφορους όρους, με συνέπεια την περαιτέρω ελάττωση του μεγέθους των παραγόμενων εκφράσεων. Σε άλλες εργασίες παρουσιάζονται αλγόριθμοι ελαχιστοποίησης κάποιων υποκλάσεων των εκφράσεων ESOP [106, 10, 60].

Στην παρούσα διδακτορική διατριβή παρουσιάζεται ένας καινούριος ευριστικός αλγόριθμος για την ελαχιστοποίηση εκφράσεων ESOP για μη πλήρως ορισμένες συναρτήσεις πολλών εξόδων. Ένας αρχικός αλγόριθμος, ο DCMIN [32] χρησιμοποιεί αποσύνθεση συναρτήσεων στις απλούστερες υποσυναρτήσεις τους (που εξαρτώνται από λιγότερες μεταβλητές) για να παράγει σχεδόν ελάχιστες εκφράσεις για τέτοιες συναρτήσεις. Μια βολική αντιστοίχιση μιας Boolean συνάρτησης πολλών εξόδων με μια συνάρτηση μιας εξόδου, που όμως εξαρτάται επιπλέον από μια μεταβλητή πολλαπλών τιμών, μας παρέχει τη δυνατότητα να χειριστούμε συναρτήσεις πολλών εξόδων.

Όμως, ο DCMIN κρίνεται αναποτελεσματικός από την άποψη του χρόνου εκτέλεσής του. Επομένως, προτείνεται ένας βελτιωμένος αλγόριθμος, ο QuickDCMIN [32], ο οποίος βασίζεται στον προηγούμενο και δίνει καλύτερα πειραματικά αποτελέσματα από αυτόν. Ο QuickDCMIN χρησιμοποιεί μια απλοποιημένη παραλλαγή

του DCMIN σαν ένα μετασχηματισμό κύβων και τον εφαρμόζει επαναληπτικά σε τυχαία επιλεγμένους κύβους από το κάλυμμα της συνάρτησης, για να ελαχιστοποιήσει το μέγεθός του. Η αποτελεσματικότητά του μετράται χρησιμοποιώντας συγκεκριμένες μετροσυναρτήσεις από τη βιβλιογραφία. Όπως θα καταδειχθεί στα επόμενα από τα πειραματικά αποτελέσματα, παρουσιάζει καλύτερες επιδόσεις από προηγούμενους αλγόριθμους, στις περισσότερες περιπτώσεις. Έτσι, ο αλγόριθμός μας είναι ανώτερος από άλλους γνωστούς αλγόριθμους για την ESOP ελαχιστοποίηση μη πλήρως ορισμένων συναρτήσεων, όπως ο MINT [35], ο EXORCISM-MV-2 [81] και ο EXORCISM-MV-3 [82].

Έστερα, προτείνουμε μια μέθοδο για την εύρεση μιας ακριβούς έκφρασης ESOP για μια τυχαία μη πλήρως ορισμένη λογική συνάρτηση, με 6 το πολύ μεταβλητές εισόδου [65]. Με βάση όσα γνωρίζουμε, αυτή είναι η πρώτη προσπάθεια να αντιμετωπιστεί αυτό το πρόβλημα. Το βάρος των συναρτήσεων 5 μεταβλητών αποθηκεύεται σε ένα συμπιεσμένο πίνακα, που χρησιμοποιείται εκτεταμένα στην προσέγγισή μας, ώστε να επιταχυνθεί ο χρόνος εκτέλεσης. Οι μέθοδοι που έχουν παρουσιαστεί μέχρι τώρα στη βιβλιογραφία στο πεδίο της ESOP ελαχιστοποίησης, είτε αποθηκεύουν σε πίνακες τα βάρη όλων των συναρτήσεων 4 μεταβλητών [28] είτε χρησιμοποιούν κλάσεις ισοδυναμίας [21]. Οι τελευταίες μέθοδοι είναι πιο αργές από την μέθοδο που προτείνουμε και όλες οι άλλες μέθοδοι για ESOP ελαχιστοποίηση μπορούν να εκμεταλλευτούν την προσέγγισή μας.

1.2 Εκφράσεις ESCT

Στις εκφράσεις ESOP, οι όροι αποτελούνται από literals των μεταβλητών εισόδου, στα οποία εφαρμόζεται η λογική πράξη AND. (Τα literals θα εξηγηθούν παρακάτω, αλλά για την ώρα αρκεί ότι το literal μιας δυαδικής μεταβλητής x είναι είτε x είτε \bar{x} είτε 1.) Μία ενδιαφέρουσα επέκταση είναι εκφράσεις στις οποίες χρησιμοποιούμε πιο περίπλοκους όρους. Τέτοιες αναπαραστάσεις είναι οι εκφράσεις της μορφής αποκλειστικού-ή αθροίσματος από σύνθετους όρους (Exclusive-or Sum of Complex Terms - ESCT). Στους σύνθετους όρους οι επιτρεπόμενες πράξεις ανάμεσα στα literals των μεταβλητών εισόδου είναι όλες οι λογικές συναρτήσεις 2 εισόδων και 1 εξόδου.

Το μεγαλύτερο πλεονέκτημα των εκφράσεων ESCT σε σχέση με τις ESOP είναι ότι αποτελούνται από λιγότερους όρους, οδηγώντας έτσι σε μικρότερα και καλύτερα λογικά κυκλώματα. Επίσης οι εκφράσεις ESCT μπορούν να απεικονιστούν εύκολα [47] σε αρχιτεκτονικές που θεωρούνται αντιστρέψιμες (reversible) [5], οδηγώντας σε κυκλώματα με μικρότερες θερμικές απώλειες. Μπορούν επίσης να χρησιμοποιηθούν [47] σε κβαντικούς υπολογιστές [49, 67]. Το μειονέκτημα είναι ότι η αποτελεσματική αποσύνθεση και ελαχιστοποίηση των ESCT είναι περισσότερο δύσκολη από την ελαχιστοποίηση των ESOP. Άλλωστε το πρόβλημα της εύρεσης ελάχιστης έκφρασης ESOP για μια τυχαία λογική συνάρτηση ανήκει στην κατηγορία των NP-hard προβλημάτων, κατά συνέπεια δεν μπορεί να επιλυθεί σε πολυωνυμικό χρόνο. Το αντίστοιχο πρόβλημα της εύρεσης ελάχιστης έκφρασης ESCT είναι, θεωρητικά, τουλάχιστον το ίδιο δύσκολο.

Το πρόβλημα της αποσύνθεσης των λογικών συναρτήσεων και της εύρεσης

σχεδόν ελάχιστων εκφράσεων ESCT μελετήθηκε στο παρελθόν για συναρτήσεις μιας εξόδου [77, 68, 36, 37, 40, 80, 48, 47]. Στην αναφορά [101] παρουσιάστηκαν δύο αλγόριθμοι (που αφορούν λογικές συναρτήσεις μιας εξόδου) για την ελαχιστοποίηση του αριθμού των σύνθετων όρων σε μια έκφραση ESCT. Ο πρώτος εγγυάται την εύρεση ελάχιστων εκφράσεων για συναρτήσεις μέχρι 5 μεταβλητές. Ο δεύτερος αλγόριθμος εφαρμόζει επαναληπτικά τον πρώτο σε μια ομάδα από σύνθετους όρους, σαν ένα τελεστή μετασχηματισμού σύνθετων όρων, ώστε να μειωθεί το πλήθος τους. Αυτός ο αλγόριθμος βελτιώθηκε στην αναφορά [97] με την εισαγωγή των συγγενών σύνθετων όρων (relative complex terms) και των ισοδύναμων (equivalent) εκφράσεων ESCT. Ένας άλλος αλγόριθμος [99] βρίσκει ακριβείς εκφράσεις ESCT για συναρτήσεις μιας εξόδου που έχουν μέχρι 6 μεταβλητές εισόδου. Επίσης έχουν προταθεί ευριστικοί αλγόριθμοι [98] για την εκτίμηση της βέλτιστης (ή σχεδόν βέλτιστης) αναδιάταξης μεταβλητών (variable reordering) μιας λογικής συνάρτησης, ώστε να βελτιωθούν τα αποτελέσματα ελαχιστοποίησης των προαναφερόμενων αλγορίθμων [97, 101, 99].

Στην παρούσα διδακτορική διατριβή παρουσιάζεται ένας καινούριος μαθηματικός φορμαλισμός [96], που επιτρέπει την επέκταση των αλγορίθμων και των μεθόδων που παρουσιάζονται στις εργασίες [101, 97] για λογικές συναρτήσεις πολλών εξόδων. Εν τούτοις, τα περισσότερα προβλήματα βασίζονται σε μη πλήρως ορισμένες συναρτήσεις, οι οποίες περιέχουν αδιάφορους όρους. Η απεικόνιση των μη πλήρως ορισμένων συναρτήσεων, μίας ή πολλών εξόδων, και η ελαχιστοποίηση των αντίστοιχων εκφράσεων ESCT είναι ακόμα μια ανοικτή ερευνητική περιοχή.

Παρουσιάζουμε, λοιπόν, στη συνέχεια έναν πιο γενικευμένο αλγόριθμο [33], που χρησιμοποιεί τον προαναφερόμενο φορμαλισμό [96], ώστε να επιτύχει ευριστική ESCT ελαχιστοποίηση για μη πλήρως ορισμένες συναρτήσεις πολλών εξόδων. Όπως και στην περίπτωση των ESOP, ένας αρχικός αλγόριθμος, ο ESCT-DC (που αποτελεί επέκταση του DCMIN), παράγει σχεδόν ελάχιστες εκφράσεις για τέτοιες συναρτήσεις. Και ο ESCT-DC κρίνεται αναποτελεσματικός όσον αφορά το χρόνο εκτέλεσής του.

Έτσι, και σε αυτήν την περίπτωση, αναπτύσσουμε ένα βελτιωμένο αλγόριθμο, τον QuickESCT-DC (που αποτελεί επέκταση του QuickDCMIN), και είναι γρηγορότερος. Ο QuickESCT-DC χρησιμοποιεί μια απλοποιημένη παραλλαγή του ESCT-DC σαν έναν τελεστή μετασχηματισμού κύβων και τον εφαρμόζει επαναληπτικά σε τυχαία επιλεγμένους κύβους από το κάλυμμα της συνάρτησης, ώστε να ελαχιστοποιήσει το μέγεθός της. Η αποτελεσματικότητα του αλγορίθμου αξιολογείται χρησιμοποιώντας τυποποιημένες μετροσυναρτήσεις από τη βιβλιογραφία.

Τονίζουμε την αναλογία ανάμεσα στις δύο περιπτώσεις της ESOP και της ESCT ελαχιστοποίησης. Οι εκφράσεις ESOP είναι υποπεριπτώσεις των εκφράσεων ESCT, ο αλγόριθμος ESCT-DC επεκτείνει τον αλγόριθμο DCMIN και ο αλγόριθμος QuickESCT-DC επεκτείνει τον αλγόριθμο QuickDCMIN. Για αυτό, άλλωστε, οι ESCT-DC και QuickESCT-DC έχουν μια παράμετρο που μπορεί να καθορίσει αν θα παράγουν εκφράσεις ESOP ή ESCT. Στη πρώτη περίπτωση παράγουν ακριβώς τα ίδια αποτελέσματα με τους DCMIN και QuickDCMIN αντίστοιχα.

Με βάση όσα γνωρίζουμε, ο QuickESCT-DC είναι ο πρώτος αλγόριθμος στο πεδίο της ευριστικής ESCT ελαχιστοποίησης για μη πλήρως ορισμένες συναρτήσεις πολλών εξόδων. Σε άλλες αναφορές [37, 38] προτείνεται ένας αλγόριθμος

που παράγει εκφράσεις παρόμοιες με τις εκφράσεις ESCT, αλλά επιτρέπονται και ο τελεστής OR και ο τελεστής XOR ανάμεσα στους ESCT όρους. Αν και αυτές οι εκφράσεις είναι πιο γενικές από τις εκφράσεις ESCT, και θα περιμέναμε να δίνουν καλύτερα αποτελέσματα, ο QuickESCT-DC (που επιτρέπει μόνο τον τελεστή XOR) πετυχαίνει μεγαλύτερη μείωση του αριθμού των όρων.

1.3 Κβαντικοί υπολογιστές

Μέχρι τώρα, όλα τα κλασικά υπολογιστικά συστήματα χρησιμοποιούν δίτιμη λογική και στοιχεία δύο καταστάσεων. Υπάρχουν όμως περιορισμοί ως προς την απόδοση που μπορούμε να επιτύχουμε, περιορισμοί που οι ίδιοι οι φυσικοί νόμοι μας επιβάλλουν. Ο πιο αξιόλογος από αυτούς είναι η πεπερασμένη ταχύτητα του ίδιου του φωτός. Τα εξαρτήματα ενός κλασικού υπολογιστικού συστήματος ανταλλάσσουν πληροφορία μεταξύ τους με τη μορφή μεταφοράς ενέργειας. Η ταχύτητα της παραπάνω ανταλλαγής είναι (στην καλύτερη περίπτωση) ίση με την ταχύτητα του φωτός. Έτσι για να διανυθεί μια απόσταση 30 εκατοστών, στο κενό, απαιτείται χρόνος 1 νανοδευτερόλεπτο. Στον ίδιο χρόνο η πληροφορία μεταδίδεται 20 εκατοστά σε έναν μεταλλικό αγωγό. Είναι, λοιπόν, αναπόφευκτο ότι στην προσπάθειά μας να αυξήσουμε την ταχύτητα ενός κλασικού υπολογιστικού συστήματος θα πρέπει να μειώσουμε τις αποστάσεις των εξαρτημάτων, φτάνοντας ενδεχομένως σε υποατομικά επίπεδα. Όταν όμως γίνει κάτι τέτοιο τότε οι καταστάσεις των εξαρτημάτων θα κυβερνώνονται από την αρχή της απροσδιοριστίας του Heisenberg.

Επιπλέον η ίδια η τεχνολογία κατασκευής των ολοκληρωμένων κυκλωμάτων μας επιβάλλει περιορισμούς ως προς το ταβάνι της απόδοσης η οποία μπορεί να επιτευχθεί. Κάθε ολοκληρωμένο κύκλωμα αποβάλλει ενέργεια με τη μορφή θερμότητας. Η ενέργεια αυτή θα πρέπει να απομακρύνεται από το κύκλωμα. Όμως η ικανότητά μας να απομακρύνουμε την παραπάνω θερμότητα αυξάνεται με μικρότερο ρυθμό σε σχέση με το ρυθμό που αυξάνεται η δημιουργία της όσο το μέγεθος των ολοκληρωμένων κυκλωμάτων μειώνεται [43].

Η ιδέα της κατασκευής και χρήσης υπολογιστικών συστημάτων που θα χρησιμοποιούν φαινόμενα και νόμους της κβαντικής μηχανικής είναι πλέον ευρέως διαδεδομένη και πλήθος ερευνητικών προσπαθειών στρέφεται προς αυτήν την κατεύθυνση. Η αρχική ιδέα για την ανάπτυξη των κβαντικών υπολογιστών ανήκει στον Richard P. Feynman (Βραβείο Nobel Φυσικής, 1965). Παρά τις τεχνικές δυσκολίες στην υλοποίηση ενός ολοκληρωμένου κβαντικού υπολογιστή, τα κβαντικά κυκλώματα μελετώνται ήδη εκτεταμένα. Πολλοί αλγόριθμοι, ειδικά σχεδιασμένοι για κβαντικούς υπολογιστές, έχουν προταθεί, και μερικοί από αυτούς αποδεικνύουν ότι, σε ορισμένα είδη προβλημάτων, ένας κβαντικός υπολογιστής μπορεί να επιτύχει καλύτερη πολυπλοκότητα από ένα συμβατικό.

Οι τρεις αλγόριθμοι που αποτελούν, μέχρι τώρα, τη θεωρητική θεμελίωση των κβαντικών αλγορίθμων είναι οι αλγόριθμοι του Shor [79], του Grover [23] και του Κβαντικού Μετασχηματισμού Fourier [34].

Η μεγάλη έκρηξη στο πεδίο των κβαντικών υπολογιστών έλαβε χώρα το 1994, όταν ο Peter Shor δημοσίευσε ένα κβαντικό αλγόριθμο [79], ο οποίος μπορούσε σε πολυωνυμικό χρόνο να λύσει το πρόβλημα της παραγοντοποίησης ενός ακεραίου.

Στους κλασικούς υπολογιστές, η πολυπλοκότητα της επίλυσης του προβλήματος της παραγοντοποίησης ενός ακεραίου σε δύο πρώτους αριθμούς αυξάνεται εκθετικά ανάλογα με το πλήθος των ψηφίων. Η παραγοντοποίηση ενός αριθμού σε δύο πρώτους αριθμούς είναι ιδιαίτερα σημαντική για την κρυπτογραφία και ειδικότερα για τον αλγόριθμο RSA, έναν αλγόριθμο κρυπτογράφησης που χρησιμοποιείται ευρέως. Έτσι, ο αλγόριθμος του Shor καθιστά τον αλγόριθμο RSA και τους σχετικούς αλγόριθμους κρυπτογραφίας ξεπερασμένους.

Μετά τη δημοσίευση του αλγορίθμου του Shor, παρουσιάστηκε και ο αλγόριθμος του Lov Grover [23] για αναζήτηση σε μη δομημένες βάσεις δεδομένων. Αν και η λύση του Grover δεν πετυχαίνει τα εντυπωσιακά αποτελέσματα που πετυχαίνει ο αλγόριθμος του Shor σε σχέση με τους κλασικούς υπολογιστές, είναι αποδεδειγμένα πιο αποδοτικός από το βέλτιστο αλγόριθμο των κλασικών υπολογιστών για το συγκεκριμένο πρόβλημα.

Τέλος, ο Κβαντικός Μετασχηματισμός Fourier είναι πρακτικά η υλοποίηση του Διακριτού Μετασχηματισμού Fourier ως ένα κβαντικό κύκλωμα, και έχει πολλές εφαρμογές στους κβαντικούς αλγόριθμους, καθώς μας δίνει τη θεωρητική βάση για τη διαδικασία εκτίμησης φάσης και αποτελεί κλειδί για πολλούς σημαντικούς κβαντικούς αλγόριθμους.

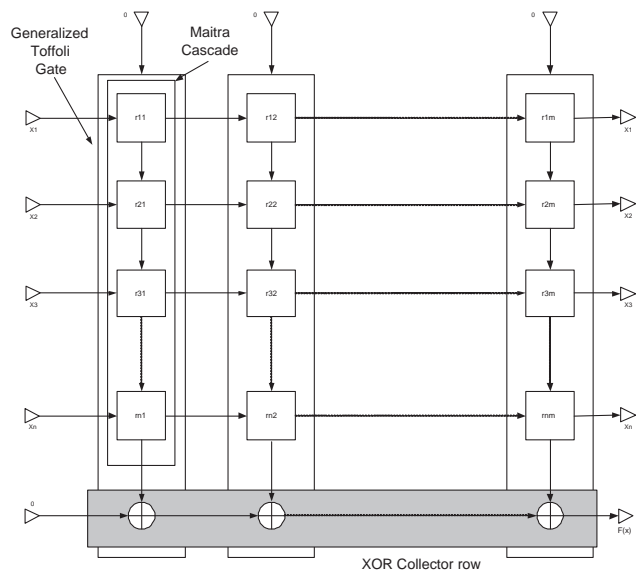
Όλοι αυτοί οι κβαντικοί αλγόριθμοι επιτυγχάνουν πολυπλοκότητες πολύ καλύτερες από τους αντίστοιχους συμβατικούς. Φαίνεται ότι οι κβαντικοί υπολογιστές είναι η απάντηση στην επίλυση κλασικών προβλημάτων με πολύ μεγάλη πολυπλοκότητα.

Στην συγκεκριμένη διδακτορική διατριβή θα ασχοληθούμε με την κβαντική επέκταση κάποιων από τους κλασικούς αλγόριθμους που αναπτύξαμε για τις εκφράσεις ESOP και ESCT. Από τους παραπάνω αλγόριθμους μας ενδιαφέρει κυρίως ο αλγόριθμος του Grover, ο οποίος αποτελεί και τη βάση για τον κβαντικό αλγόριθμο που θα παρουσιάσουμε.

Μια ενδιαφέρουσα ιδιότητα των εκφράσεων ESOP και ESCT είναι ότι μπορούν να απεικονιστούν απευθείας σε κυτταρικές αρχιτεκτονικές, όπως η κυτταρική αρχιτεκτονική Maitra (Εικόνα 1.1), η οποία έχει αποδειχθεί ότι είναι αντιστρέψιμη [47]. Επομένως, η έκφραση μιας boolean συνάρτησης σε μορφή ESOP ή ESCT καταλήγει σε ένα αντιστρέψιμο κύκλωμα και μπορεί να βοηθήσει στη σχεδίαση κβαντικών κυκλωμάτων [47].

Στην αναφορά [39] περιγράφεται ένας κβαντικός αλγόριθμος για την εύρεση εκφράσεων FPRM (Fixed Polarity Reed Muller) με μικρότερο αριθμό όρων από ένα προσδιορισμένο κατώφλι. Προτείνει την κατασκευή ενός εξειδικευμένου κβαντικού τελεστή (Oracle) που αποτιμά εκφράσεις FPRM. Χρησιμοποιεί έπειτα αυτό το Oracle σε συνδυασμό με τον αλγόριθμο του Grover για να βρει τις εκφράσεις FPRM με τα επιθυμητά χαρακτηριστικά. Στην αναφορά [66] περιγράφεται ένας κβαντικός αλγόριθμος (ο QMin) για την εύρεση εκφράσεων ESOP ή ESCT με μικρότερο αριθμό όρων από ένα προσδιορισμένο κατώφλι (και αν το κατώφλι είναι κατάλληλα επιλεγμένο, βρίσκει ελάχιστες εκφράσεις). Ο αλγόριθμος QMin χρησιμοποιεί ένα τροποποιημένο oracle για τον αλγόριθμο του Grover και προσδιορίζει ελάχιστες εκφράσεις ESOP ή ESCT για μια τυχαία πλήρως ορισμένη boolean συνάρτηση.

Σε αυτήν τη διατριβή επεκτείνεται ο προηγούμενα αναφερόμενος αλγόριθμος QMin [66] ώστε να προσδιορίσει ελάχιστες εκφράσεις ESOP ή ESCT για μη πλή-



Σχήμα 1.1: Κυτταρική αρχιτεκτονική Maitra

πως ορισμένες Boolean συναρτήσεις (ο QMin έχει σχεδιαστεί για πλήρως ορισμένες Boolean συναρτήσεις). Με βάση όσα γνωρίζουμε, αυτός είναι ο πρώτος χβαντικός αλγόριθμος που προσδιορίζει ελάχιστες εκφράσεις ESOP ή ESCT για μια τυχαία μη πλήρως ορισμένη Boolean συνάρτηση.

Κεφάλαιο 2

Λογικές Συναρτήσεις και εκφράσεις ESOP

2.1 Λογικές συναρτήσεις πολλαπλών τιμών

Ορισμός 1 Έστω X μια μεταβλητή που παίρνει τιμές από το σύνολο $V = \{0, \dots, v-1\}$ και $S \subseteq V$. Τότε X^S είναι *literal* της X τέτοιο ώστε $X^S = 1$ όταν $X \in S$ και $X^S = 0$ όταν $X \in V \setminus S$. Όταν $S = V$ τότε $X^S = 1$ και θεωρείται περιττό.

Παράδειγμα 1 Έστω $V = \{0, 1\}$. Τότε $\bar{x} = x^{\{0\}}$, $x = x^{\{1\}}$ και $1 = x^{\{0,1\}}$ είναι τα *literals* της δυαδικής μεταβλητής x .

Ας κάνουμε τη διάκριση μεταξύ των δυαδικών μεταβλητών και των μεταβλητών πολλαπλών τιμών. Στη συνέχεια της διδακτορικής διατριβής χρησιμοποιούμε κεφαλαία γράμματα για να αναπαραστήσουμε μεταβλητές πολλαπλών τιμών και πεζά γράμματα για τις δυαδικές.

Ορισμός 2 Έστω X^{S_1} και X^{S_2} δύο *literals* της μεταβλητής X . Τότε $X^{S_1} \oplus X^{S_2} \equiv X^{S_1 \oplus S_2} \equiv X^{(S_1 \cup S_2) \setminus (S_1 \cap S_2)}$

Ορισμός 3 Έστω $V_i = \{0, \dots, v_i - 1\}$, $i = 1, \dots, n$ και $B = \{0, 1\}$. Μια συνάρτηση εισόδων πολλαπλών τιμών, δυαδικής εξόδου $f(X_1, \dots, X_n)$ είναι μια απεικόνιση $f : V_1 \times \dots \times V_n \rightarrow B$. Αν $|V_i| = 2$, $i = 1, \dots, n$ τότε η f ονομάζεται *switching* ή *Boolean* συνάρτηση.

Στη συγκεκριμένη διδακτορική διατριβή μας απασχολούν απεικονίσεις της μορφής $f : \{0, 1\} \times \{0, 1\} \dots \times \{0, \dots, v-1\} \rightarrow \{0, 1\}$, δηλαδή απεικονίσεις που εξαρτώνται από μία μόνο μεταβλητή πολλαπλών τιμών, ενώ οι υπόλοιπες μεταβλητές είναι δυαδικές.

2.2 Λογικές εκφράσεις XOR-αθροίσματος γινομένων

Ορισμός 4 Έστω $X_1^{S_1}, \dots, X_n^{S_n}$ literals των μεταβλητών εισόδου της συνάρτησης f . Τότε $C = X_1^{S_1} \dots X_n^{S_n}$ ορίζεται ως κύβος μιας συνάρτησης f , η οποία εξαρτάται από τις μεταβλητές X_1, \dots, X_n . Αν $|S_i| = 1, i = 1, \dots, n$ τότε ο C είναι ελαχιστόρος (minterm) της f .

Παράδειγμα 2 Έστω f μια λογική συνάρτηση που εξαρτάται από τις δυαδικές μεταβλητές x_1, x_2, x_3, x_4 . Τότε οι $\bar{x}_1 x_3 x_4, x_2 x_4, \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4$ αποτελούν κύβους της f . Από αυτούς, μόνο ο τελευταίος είναι ελαχιστόρος της f .

Ορισμός 5 Η απόσταση δύο κύβων ορίζεται ως ο αριθμός των literals στα οποία διαφέρουν.

Παράδειγμα 3 Έστω f μια boolean συνάρτηση με 3 μεταβλητές εισόδου x_1, x_2, x_3 . Τότε οι κύβοι $x_1 \bar{x}_2 x_3$ και $x_1 \bar{x}_2 \bar{x}_3$ έχουν απόσταση 1, ενώ η απόσταση των κύβων $x_1 x_3$ και $\bar{x}_2 x_3$ είναι 2.

Ορισμός 6 Ένα MVESOP (Multiple-Valued Exclusive-or Sum Of Products) ή κάλυμμα είναι μια έκφραση της μορφής $\bigoplus C_i$, όπου C_i είναι κύβοι της συνάρτησης f . Αν όλες οι μεταβλητές εισόδου της f είναι δυαδικές, τότε μια έκφραση της παραπάνω μορφής ονομάζεται ESOP. Ένα MVESOP (ή ένα ESOP) αναπαριστά με μη μοναδικό τρόπο μια συνάρτηση f .

Παράδειγμα 4 Η έκφραση $X_1^{\{1,3\}} X_2^{\{0,2\}} X_3^{\{0,1\}} \oplus X_2^{\{1\}} \oplus X_1^{\{0,1,2\}} X_3^{\{2\}}$ είναι ένα MVESOP της συνάρτησης $f(X_1, X_2, X_3) : \{0, 1, 2, 3\} \times \{0, 1, 2\} \times \{0, 1, 2\} \rightarrow \{0, 1\}$.

Ορισμός 7 Ένα MVESOP μιας συνάρτησης f που έχει τον ελάχιστο αριθμό κύβων ανάμεσα σε όλα τα δυνατά MVESOP για αυτή τη συνάρτηση λέγεται ελάχιστο ή ακριβές MVESOP της f .

Ορισμός 8 Το μέγεθος $s(E)$ ενός MVESOP E είναι το πλήθος των κύβων σε αυτό. Το βάρος $w(f)$ μιας συνάρτησης f είναι το πλήθος των κύβων σε ένα ακριβές MVESOP της f .

Παράδειγμα 5 Η συνάρτηση ισοτιμίας 4 μεταβλητών ορίζεται από τη σχέση $f(x_1, x_2, x_3, x_4) = x_1 \oplus x_2 \oplus x_3 \oplus x_4$. Μία ESOP έκφραση της f είναι η $f = \bar{x}_1 \bar{x}_2 \bar{x}_4 \oplus \bar{x}_1 \bar{x}_2 x_4 \oplus \bar{x}_3 \oplus x_4 \oplus x_1 x_2$. Το βάρος της f είναι ίσο με 4, επομένως μία ακριβής ESOP έκφραση για την f είναι η σχέση ορισμού της.

Ορισμός 9 Έστω f μια συνάρτηση και $w(f)$ το βάρος της. Τότε ένα k -ισοδύναμο MVESOP της f είναι ένα MVESOP E με μέγεθος $s(E) = w(f) + k$.

Παράδειγμα 6 Αν $f = x_1$ είναι μια συνάρτηση που εξαρτάται από τις μεταβλητές x_1, x_2 , τότε ένα 1-ισοδύναμο ESOP της f είναι το $f = x_1 \bar{x}_2 \oplus x_1 x_2$.

2.3 Αναπτύγματα και αναπαραστάσεις boolean συναρτήσεων

Ορισμός 10 Έστω μια συνάρτηση f με μία τουλάχιστον δυαδική μεταβλητή x_1 . Τότε οι συναρτήσεις $f^0 = f(x_1 = 0, x_2, \dots, x_n)$, $f^1 = f(x_1 = 1, x_2, \dots, x_n)$ και $f^2 = f^0 \oplus f^1$ ορίζονται ως οι υποσυναρτήσεις της f .

Παράδειγμα 7 Έστω $f = x_1 \bar{x}_2 x_3 \oplus \bar{x}_1 \bar{x}_4 \oplus x_2 x_4$ μια λογική συνάρτηση 4 μεταβλητών. Τότε $f^0 = \bar{x}_4 \oplus x_2 x_4$, $f^1 = \bar{x}_2 x_3 \oplus x_2 x_4$ και $f^2 = \bar{x}_4 \oplus \bar{x}_2 x_3$.

Ορισμός 11 Μια συνάρτηση f με μία τουλάχιστον δυαδική μεταβλητή x_1 μπορεί να αναπτυχθεί ως εξής:

$$f = \bar{x}_1 f^0 \oplus x_1 f^1 \quad (2.1)$$

$$f = f^0 \oplus x_1 f^2 \quad (2.2)$$

$$f = f^1 \oplus \bar{x}_1 f^2 \quad (2.3)$$

Αυτά τα αναπτύγματα είναι γνωστά ως Shannon (Boole), θετικό Davio και αρνητικό Davio αντίστοιχα.

Ο υπολογισμός των υποσυναρτήσεων και η ανάπτυξη σύμφωνα με τα αναπτύγματα Shannon και Davio μπορεί να γίνει ως προς οποιαδήποτε δυαδική μεταβλητή εισόδου ή μπορεί να γίνει και αναδρομικά. Έτσι, προκειμένου για μια αρχική συνάρτηση f , που εξαρτάται από τουλάχιστον δύο δυαδικές μεταβλητές x_1 και x_2 , οι υποσυναρτήσεις π.χ. της υποσυνάρτησης f^0 συμβολίζονται ως f^{00} , f^{01} και f^{02} και ισχύει π.χ. $f^0 = \bar{x}_2 f^{00} \oplus x_2 f^{01}$.

Παράδειγμα 8 Για τη συνάρτηση f^1 του Παραδείγματος 7, αν υπολογίσουμε τις υποσυναρτήσεις της ως προς την μεταβλητή x_2 , έχουμε ότι $f^{10} = x_3$, $f^{11} = x_4$ και $f^{12} = x_3 \oplus x_4$.

Ορισμός 12 Η αναπαράσταση ελαχιστόρων (MT - MinTerm) μιας switching συνάρτησης f με n μεταβλητές εισόδου είναι ένα bit διάνυσμα μεγέθους 2^n , όπου το i -οστό bit είναι 1 αν ο i -οστός ελαχιστόρος της f είναι 1.

Πρακτικά, η MT αναπαράσταση μιας switching συνάρτησης είναι η δεξιά στήλη του πίνακα αληθείας της (που αναπαριστά την έξοδο της συνάρτησης), και το λιγότερο σημαντικό ψηφίο (Least Significant Bit - LSB) της MT αντιστοιχεί στην πρώτη σειρά της δεξιάς στήλης και το περισσότερο σημαντικό ψηφίο (Most Significant Bit - MSB) της MT αντιστοιχεί στην τελευταία σειρά της δεξιάς στήλης. Ο i -οστός ελαχιστόρος αναφέρεται στην i -οστή σειρά του πίνακα αληθείας.

Παράδειγμα 9 Έστω f η switching συνάρτηση του παραδείγματος 7. Τότε η αναπαράσταση ελαχιστόρων για τους τρεις κύβους της f είναι [0000 1100 0000 0000], [0000 0000 0101 0101], [1010 0000 1010 0000] αντίστοιχα. Η αναπαράσταση ελαχιστόρων για την f είναι [1010 1100 1111 0101], ή, περισσότερο συμπυκνωμένα, σε δεκαεξαδική μορφή, [acf5].

2.4 Συναρτήσεις πολλών εξόδων και χαρακτηριστική συνάρτηση

Ορισμός 13 Έστω $f : \{0, 1\}^n \rightarrow \{0, 1\}^k$ μια *switching* συνάρτηση n μεταβλητών με k εξόδους, και f_j η j -ιστή έξοδος της, $j \in \{0, \dots, k-1\}$. Η χαρακτηριστική συνάρτηση ψ της f ορίζεται ως $\psi : \{0, 1\}^n \times \{0, \dots, k-1\} \rightarrow \{0, 1\}$, όπου $\psi(x_1, \dots, x_n, j) = f_j(x_1, \dots, x_n)$.

Ορισμός 14 Έστω $f : \{0, 1\}^n \rightarrow \{0, 1\}^k$ μια *switching* συνάρτηση n μεταβλητών με k εξόδους και $\psi : \{0, 1\}^n \times \{0, \dots, k-1\} \rightarrow \{0, 1\}$ η χαρακτηριστική της συνάρτηση. Η αναπαράσταση ελαχιστόρων μεταβλητών πολλαπλών τιμών (MVMT - Multiple-Valued MinTerm) m της f ή της ψ είναι ένα bit διάνυσμα μεγέθους $2^{n+\lceil \lg(k) \rceil}$. Έστω $x_1^{a_1} \dots x_n^{a_n} X^S$ ένας κύβος της ψ , όπου $a_i \in \{0, 1\}$ και $S \subseteq \{0, \dots, k-1\}$. Έστω ότι το p_1 είναι ίσο με το δυαδικό αριθμό $\langle a_1, \dots, a_n \rangle$, $p_2 = 2^{\lceil \lg(k) \rceil}$ και O είναι ένα bit διάνυσμα μεγέθους p_2 που το i -ιστό του bit είναι 1 αν $i \in S$. Τότε, τα bits $[(p_1 \cdot p_2 + p_2 - 1) \dots (p_1 \cdot p_2)]$ της MVMT m είναι ίσα με το O . Αν $k = 2$ τότε το MVMT της f είναι ουσιαστικά το MT της f .

Ουσιαστικά η MVMT αναπαράσταση της f ή της ψ μπορεί να θεωρηθεί σαν η επικάλυψη των $2^{\lceil \lg_2 k \rceil}$ MT των συναρτήσεων n μεταβλητών g_i , όπου $g_i = f_i$ όταν $0 \leq i \leq k-1$ και $g_i = 0$ όταν $k \leq i \leq 2^{\lceil \lg_2 k \rceil} - 1$ και το LSB αντιστοιχεί στην g_0 .

Σημειώνουμε ότι όταν ο αριθμός των εξόδων δεν είναι δύναμη του 2, προσθέτουμε τον ελάχιστο δυνατό αριθμό μηδενικών που αντιστοιχούν σε εικονικές εξόδους, ώστε ο αριθμός των bits για κάθε ελαχιστόρο στην αναπαράσταση MVMT να είναι δύναμη του 2.

Παράδειγμα 10 Έστω $f : \{0, 1\}^4 \rightarrow \{0, 1\}^6$ μια συνάρτηση. Η MVMT της f μπορεί να θεωρηθεί σαν η παρεμβολή των 6 MT των συναρτήσεων 4 μεταβλητών f_i (που αντιστοιχούν στις εξόδους της f με $i = 0, \dots, 5$) και αντιστρόφως σύμφωνα με τους ορισμούς 13 και 14. Έστω $f_0 = [d137]$, $f_1 = [20ac]$, $f_2 = [f3e4]$, $f_3 = [89bd]$, $f_4 = [56af]$ και $f_5 = [dd33]$ (σε δεκαεξαδική μορφή). Περισσότερο συγκεκριμένα, σε δυαδική μορφή: $f_0 = [1101\ 0001\ 0011\ 0111]$, $f_1 = [0010\ 0000\ 1010\ 1100]$, $f_2 = [1111\ 0011\ 1110\ 0100]$, $f_3 = [1000\ 1001\ 1011\ 1101]$, $f_4 = [0101\ 0110\ 1010\ 1111]$ και $f_5 = [1101\ 1101\ 0011\ 0011]$.

Χρησιμοποιώντας τα τελευταία ψηφία των MT αναπαραστάσεων των f_5, f_4, f_3, f_2, f_1 και f_0 , όπου το LSB αντιστοιχεί στην πρώτη έξοδο, και γειρίζοντας με μηδενικά ώστε να χρησιμοποιήσουμε δεκαεξαδικούς αριθμούς, τα τελευταία δύο ψηφία της MVMT της f παράγονται. Επομένως: $(00)(111001) = [39]$. Όλα τα άλλα ψηφία παράγονται με τον ίδιο τρόπο. Τότε η MVMT αναπαράσταση της f (σε δεκαεξαδική μορφή) είναι $[2d350635\ 2830142d\ 1e043f29\ 1a1f3139]$.

Ορισμός 15 Έστω m η MVMT αναπαράσταση μιας συνάρτησης f με $2k$ bits. Τότε η MVMT αναπαράσταση m_0 της f^0 είναι τα k λιγότερο σημαντικά bits της m , η MVMT αναπαράσταση m_1 της f^1 είναι τα k περισσότερα σημαντικά bits της

m και η MVMT αναπαράσταση m_2 της f^2 είναι το bit προς bit αποκλειστικό-ή της m_0 και της m_1 .

Παράδειγμα 11 Για τη συνάρτηση f του παραδείγματος 10 η MVMT αναπαράσταση της f^1 είναι $[2d350635\ 2830142d]$ και της f^0 είναι $[1e043f29\ 1a1f3139]$. Υπολογίζοντας το bit προς bit αποκλειστικό-ή των δύο προηγούμενων αναπαραστάσεων βρίσκουμε ότι η MVMT αναπαράσταση της f^2 είναι $[3331391c\ 322f2514]$.

Ορισμός 16 Η positional notation (PN) αναπαράσταση ενός κύβου $C = x_1^{S_1} \dots x_n^{S_n} X^{S_{n+1}}$, $S_i \subseteq \{0, 1\}$, $i \in \{1, \dots, n\}$, $S_{n+1} \subseteq \{0, \dots, v-1\}$ είναι ένα bit διάνυσμα μεγέθους $2n + v$, όπου το bit $2 \cdot (k-1) + l$ είναι 1 αν $l \in S_k$.

Παράδειγμα 12 Έστω $C = x_1^{\{0\}} x_2^{\{1\}} x_3^{\{0,1\}} X^{\{0,2,3,5\}}$ ένας κύβος μιας συνάρτησης $f : \{0, 1\} \times \{0, 1\} \times \{0, 1\} \times \{0, 1, 2, 3, 4, 5\} \rightarrow \{0, 1\}$. Η PN αναπαράσταση για τον C είναι $[101101 \cdot 11 \cdot 10 \cdot 01]$.

2.5 Μη πλήρως ορισμένες λογικές συναρτήσεις

Ορισμός 17 Υπάρχουν πολλές περιπτώσεις όπου δεν ενδιαφερόμαστε για την τιμή εξόδου μιας συνάρτησης για μερικούς συγκεκριμένους συνδυασμούς τιμών των μεταβλητών εισόδου. Οι κύβοι που περιγράφονται από αυτά τα σύνολα τιμών των εισόδων ονομάζονται αδιάφοροι όροι. Οι τιμές εξόδου για τους αδιάφορους όρους είναι απροσδιόριστες.

Σημειώνουμε ότι κάθε κύβος του προηγούμενου ορισμού αντιστοιχεί σε έναν ή περισσότερους ελαχιστόρους της συνάρτησης. Το σύνολο αυτών των ελαχιστόρων αποτελεί τους αδιάφορους ελαχιστόρους της συνάρτησης, που ορίζονται στον επόμενο ορισμό.

Ορισμός 18 Έστω $V_i = \{0, \dots, v_i - 1\}$, $i = 1, \dots, n$ και $T = \{0, 1, d\}$. Μια μη πλήρως ορισμένη συνάρτηση, εισόδων πολλαπλών τιμών και δυαδικής εξόδου $f(X_1, \dots, X_n)$ είναι μια απεικόνιση $f : V_1 \times \dots \times V_n \rightarrow T$. Οι ελαχιστόροι $X_1^{S_1} \dots X_n^{S_n}$, $|S_i| = 1$, $i = 1, \dots, n$, για τους οποίους $f(X_1, \dots, X_n) = d$, είναι οι αδιάφοροι ελαχιστόροι της συνάρτησης.

Παράδειγμα 13 Ο Πίνακας 2.1 αναπαριστά μια λογική συνάρτηση 4 εισόδων και 1 εξόδου. Οι 4 είσοδοι αναπαριστούν ένα δεκαδικό ψηφίο χρησιμοποιώντας τον κώδικα BCD (Binary Coded Decimal), σύμφωνα με τον οποίο κάθε ψηφίο αναπαριστάται από τη δυαδική τιμή του. Η τιμή της συνάρτησης είναι 1 όταν η είσοδος συμβολίζει έναν άρτιο αριθμό. Επειδή δεν ορίζονται δεκαδικά ψηφία μεγαλύτερα από το 9, οι αντίστοιχοι συνδυασμοί αποτελούν αδιάφορους όρους.

Ορισμός 19 Έστω f μια μη πλήρως ορισμένη συνάρτηση δυαδικής εξόδου. Το σύνολο των ελαχιστόρων για τους οποίους η τιμή εξόδου είναι 1 ορίζεται ως το

x_1	x_2	x_3	x_4	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	d
1	0	1	1	d
1	1	0	0	d
1	1	0	1	d
1	1	1	0	d
1	1	1	1	d

Πίνακας 2.1: Μη πλήρως ορισμένη λογική συνάρτηση

ON-set της f . Το σύνολο των ελαχιστόρων για τους οποίους η τιμή εξόδου είναι 0 ορίζεται ως το *OFF-set* της f . Το σύνολο των ελαχιστόρων για τους οποίους η τιμή εξόδου είναι αδιάφορη ορίζεται ως το *DC-set* της f . Τα τρία παραπάνω οριζόμενα σύνολα είναι ξένα ανά δύο και η ένωση αυτών των συνόλων είναι το σύνολο όλων των ελαχιστόρων της f .

Παράδειγμα 14 Για την συνάρτηση του παραδείγματος 13 το σύνολο *ON-SET* είναι το $\{\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4, \bar{x}_1\bar{x}_2x_3\bar{x}_4, \bar{x}_1x_2\bar{x}_3\bar{x}_4, \bar{x}_1x_2x_3\bar{x}_4, x_1\bar{x}_2\bar{x}_3\bar{x}_4\}$, το σύνολο *OFF-SET* είναι το $\{\bar{x}_1\bar{x}_2\bar{x}_3x_4, \bar{x}_1\bar{x}_2x_3x_4, \bar{x}_1x_2\bar{x}_3x_4, \bar{x}_1x_2x_3x_4, x_1\bar{x}_2\bar{x}_3x_4\}$ και το σύνολο *DC-SET* είναι το $\{x_1\bar{x}_2x_3\bar{x}_4, x_1\bar{x}_2x_3x_4, x_1x_2\bar{x}_3\bar{x}_4, x_1x_2\bar{x}_3x_4, x_1x_2x_3\bar{x}_4, x_1x_2x_3x_4\}$.

Μπορούμε να επεκτείνουμε τους ορισμούς μας για τη χαρακτηριστική συνάρτηση και τις MT και MVMT αναπαραστάσεις για μη πλήρως ορισμένες συναρτήσεις.

Ορισμός 20 Η αναπαράσταση ελαχιστόρων (MT) μιας μη πλήρως ορισμένης *switching* συνάρτησης n μεταβλητών f είναι ένα διάνυσμα με μέγεθος 2^n , όπου το i -οστό ψηφίο είναι 1 αν ο i -οστός ελαχιστόρος είναι στο *ON-set* της f , το i -οστό ψηφίο είναι 0 αν ο i -οστός ελαχιστόρος είναι στο *OFF-set* της f και το i -οστό ψηφίο είναι d αν ο i -οστός ελαχιστόρος είναι στο *DC-set* της f .

Ορισμός 21 Έστω $f : \{0, 1\}^n \rightarrow \{0, 1, d\}^k$ μια μη πλήρως ορισμένη *switching* συνάρτηση n μεταβλητών με k εξόδους και f_j η j -οστή της έξοδος, $j \in \{0, \dots, k-1\}$. Η χαρακτηριστική συνάρτηση ψ της f ορίζεται ως $\psi : \{0, 1\}^n \times \{0, \dots, k-1\} \rightarrow \{0, 1, d\}$, όπου $\psi(x_1, \dots, x_n, j) = f_j(x_1, \dots, x_n)$.

Ορισμός 22 Έστω $f : \{0, 1\}^n \rightarrow \{0, 1, d\}^k$ μια μη πλήρως ορισμένη *switching* συνάρτηση n μεταβλητών με k εξόδους και $\psi : \{0, 1\}^n \times \{0, \dots, k-1\} \rightarrow \{0, 1, d\}$ η χαρακτηριστική της συνάρτηση. Η *MVMT* αναπαράσταση της f ή της ψ μπορεί να θεωρηθεί ως η επικάλυψη των $2^{\lceil \log_2 k \rceil}$ *MTs* των συναρτήσεων n μεταβλητών g_i , όπου $g_i = f_i$ όταν $0 \leq i \leq k-1$ και $g_i = 0$ όταν $k \leq i \leq 2^{\lceil \log_2 k \rceil} - 1$ και το *LSB* αντιστοιχεί στη g_0 .

Ορισμός 23 Έστω m η *MVMT* αναπαράσταση μιας μη πλήρως ορισμένης συνάρτησης f με $2k$ bits. Τότε η *MVMT* αναπαράσταση m_0 της f^0 είναι τα k λιγότερο σημαντικά bits της m , η *MVMT* αναπαράσταση m_1 της f^1 είναι τα k περισσότερα σημαντικά bits της m και η *MVMT* αναπαράσταση m_2 της f^2 είναι το bit προς bit αποκλειστικό-ή της m_0 και της m_1 , όπου όταν οποιοδήποτε από δύο αντίστοιχα bits της m_0 ή της m_1 είναι d , τότε το αντίστοιχο bit που παράγεται για την m_2 είναι d .

Κεφάλαιο 3

Ευριστική ESOP ελαχιστοποίηση μη πλήρως ορισμένων λογικών συναρτήσεων

Σε αυτό το κεφάλαιο παρουσιάζεται ένας ευριστικός αλγόριθμος για την παραγωγή και την ελαχιστοποίηση εκφράσεων ESOP για τυχαίες μη πλήρως ορισμένες Boolean συναρτήσεις πολλών εξόδων. Κάθε Boolean συνάρτηση f μετασχηματίζεται στη χαρακτηριστική της συνάρτηση ψ , η οποία έχει n δυαδικές μεταβλητές (όπου n είναι το πλήθος των εισόδων της f) και μια μεταβλητή πολλαπλών τιμών με τιμές από το σύνολο $S = \{0, \dots, k - 1\}$ (όπου k είναι το πλήθος των εξόδων της f). Ο προτεινόμενος αλγόριθμος εφαρμόζεται στη χαρακτηριστική συνάρτηση. Κάθε MVESOP της χαρακτηριστικής συνάρτησης αντιστοιχεί σε ένα ESOP της αντίστοιχης συνάρτησης πολλών εξόδων.

Η κύρια ιδέα πίσω από τον προτεινόμενο αλγόριθμο είναι η αποσύνθεση της χαρακτηριστικής συνάρτησης στις απλούστερες υποσυναρτήσεις της (οι οποίες εξαρτώνται από λιγότερες μεταβλητές), η κατασκευή των εκφράσεων MVESOP για αυτές τις υποσυναρτήσεις και η σύνθεση των εκφράσεων MVESOP για την αρχική συνάρτηση από τις εκφράσεις MVESOP των υποσυναρτήσεων. Αυτή η σύνθεση των εκφράσεων πραγματοποιείται χρησιμοποιώντας τα αναπτύγματα Shannon, θετικό Davio και αρνητικό Davio.

3.1 Αναπαράσταση των μη πλήρως ορισμένων συναρτήσεων

Μπορούμε να αναπαραστήσουμε κάθε μη πλήρως ορισμένη συνάρτηση $f : \{0, 1\}^n \times \{0, \dots, k - 1\} \rightarrow \{0, 1, d\}$ χρησιμοποιώντας την MVMT αναπαράσταση, σύμφωνα

με τον Ορισμό 22. Κάθε bit της συμβολοσειράς (εκτός από τα επιπλέον μηδενικά γεμίσματα) υποδεικνύει την τιμή της αρχικής συνάρτησης πολλών εξόδων για ένα συγκεκριμένο ελαχιστόρο και μια συγκεκριμένη έξοδο. Οι MVMT αναπαραστάσεις των υποσυναρτήσεων μιας συνάρτησης f μπορούν εύκολα να εξαχθούν από την MVMT αναπαράστασή της, σύμφωνα με τον Ορισμό 23.

x_1	x_2	x_3	f_0	f_1	f_2
0	0	0	0	d	d
0	0	1	1	0	d
0	1	0	1	1	0
0	1	1	d	1	d
1	0	0	0	1	d
1	0	1	d	1	d
1	1	0	0	1	0
1	1	1	d	d	1

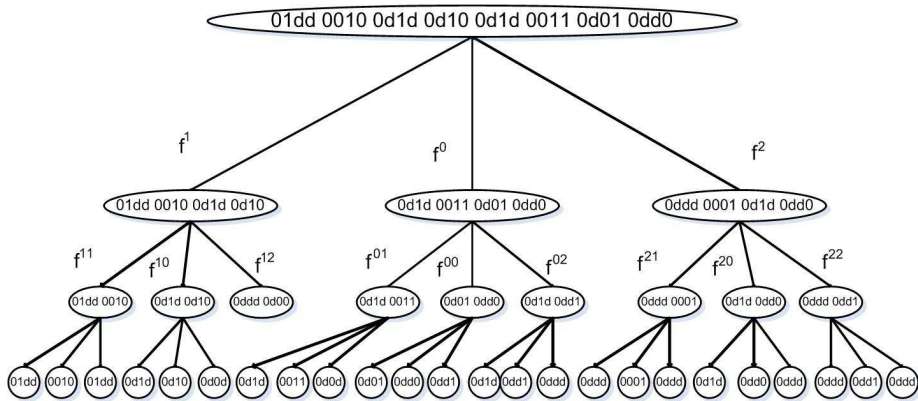
Πίνακας 3.1: Η συνάρτηση του Παραδείγματος 15

Παράδειγμα 15 Έστω f μια συνάρτηση με 3 εισόδους και 3 εξόδους, που ορίζεται από τον πίνακα αληθείας του Πίνακα 3.1. Η MVMT για αυτή τη συνάρτηση είναι $[01dd\ 0010\ 0d1d\ 0d10\ 0d1d\ 0011\ 0d01\ 0dd0]$. Οι MVMTs των υποσυναρτήσεων είναι $[0d1d\ 0011\ 0d01\ 0dd0]$, $[01dd\ 0010\ 0d1d\ 0d10]$ και $[0ddd\ 0001\ 0d1d\ 0dd0]$ για τις f^0 , f^1 και f^2 αντίστοιχα.

3.2 Αποσύνθεση

Κατά τη φάση αποσύνθεσης του αλγορίθμου παράγεται ένα τριαδικό δένδρο το πολύ n επιπέδων (όπου n είναι ο αριθμός των μεταβλητών εισόδου εκτός από την μεταβλητή πολλαπλών τιμών). Το δένδρο αυτό ονομάζεται γεννήτορ δένδρο. Η ρίζα του δένδρου αναπαριστά την αρχική συνάρτηση και τα παιδιά κάθε κόμβου αναπαριστούν τις 3 υποσυναρτήσεις του. Η αποσύνθεση για ένα συγκεκριμένο κόμβο σταματά όταν είτε η αντίστοιχη συνάρτηση είναι ίση με την μηδενική ή έχουν πραγματοποιηθεί n αποσυνθέσεις. Η συνάρτηση για κάθε κόμβο αναπαριστάται χρησιμοποιώντας την MVMT αναπαράσταση, όπως αναφέρθηκε στην προηγούμενη παράγραφο.

Θεωρούμε ότι μια συνάρτηση είναι ίση με την μηδενική συνάρτηση αν όλα τα bits της MVMT αναπαράστασής της είναι ίσα με 0 ή d . Σε αυτή την περίπτωση αυτοί οι συγκεκριμένοι αδιάφοροι όροι τίθενται ίσοι με 0. Ο λόγος για αυτή την επιλογή είναι ότι ένας όρος με 0 και d μπορεί να συνδυαστεί με έναν άλλο όρο για να μειώσει το βάρος. Όμως, αν θεωρηθεί σαν μη μηδενικός όρος, αυξάνει το βάρος κατά 1. Επομένως, η επίπτωση είναι η ίδια. Εν τούτοις, εκτιμάται ότι αυτή η επιλογή μπορεί να μειώσει ελαφρώς την ποιότητα της τελικής λύσης, αλλά επίσης απλοποιεί τον αλγόριθμο και μειώνει το χρόνο εκτέλεσής του.



Σχήμα 3.1: Γεννήτορ δένδρο του Παραδείγματος 16

Παράδειγμα 16 Στο σχήμα 3.1 μπορούμε να δούμε το γεννήτορ δένδρο για τη συνάρτηση του παραδείγματος 15. Κάθε υποσυνάρτηση αναπαριστάται από την MVMT αναπαράστασή της. Για κάθε κόμβο, το αριστερό παιδί αντιστοιχεί στην υποσυνάρτηση f^1 , το μεσαίο παιδί στην f^0 και το δεξί παιδί στην f^2 . Η υποσυνάρτηση f^{12} (που αναπαριστάται με τη συμβολοσειρά [0ddd 0d00]) είναι ίση με μηδέν, σύμφωνα με την υπόθεση που κάναμε.

3.3 Παραγωγή των εκφράσεων MVESOP των φύλλων

Όπως αναφέρθηκε προηγουμένως, η αποσύνθεση σταματά όταν ένας κόμβος στο γεννήτορ δένδρο αναπαριστά την μηδενική συνάρτηση. Σε αυτή την περίπτωση δεν παράγεται καμία έκφραση MVESOP για αυτό το φύλλο.

Στο τελευταίο επίπεδο του δένδρου, μετά από n αποσυνθέσεις, τα φύλλα αναπαριστούν literals της μεταβλητής πολλαπλών τιμών. Επομένως, οι εκφράσεις MVESOP για αυτά τα φύλλα έχουν βάρος 1 και ο τρόπος παραγωγής τους είναι σαφής.

Η MVMT αναπαράσταση που έχει ανατεθεί σε ένα φύλλο μπορεί να περιέχει ένα ή περισσότερα d bits. Κάθε πιθανός συνδυασμός από αναθέσεις για αυτά τα αδιάφορα bits οδηγεί σε μια έκφραση MVESOP. Αν υπάρχουν n bits με τιμή d , τότε υπάρχουν 2^n διαφορετικές εκφράσεις βάρους 1.

Παράδειγμα 17 Στον Πίνακα 3.2 μπορούμε να δούμε τις υποσυναρτήσεις που αντιστοιχούν στα φύλλα του τελευταίου επιπέδου στο δένδρο του σχήματος 3.1, τις MVMT αναπαραστάσεις τους και τις αντίστοιχες εκφράσεις MVESOP. Οι υποσυναρτήσεις f^{102} , f^{012} , f^{000} , f^{022} , f^{211} , f^{212} , f^{200} , f^{202} , f^{221} και f^{222} θεωρούνται ίσες με την μηδενική συνάρτηση και καμία έκφραση δεν παράγεται για αυτές.

Υποσυναρτήσεις	MVMT	MVESOP Εκφράσεις
f^{111}, f^{112}	[01dd]	$X^{\{2\}}, X^{\{0,2\}}, X^{\{1,2\}}, X^{\{0,1,2\}}$
f^{110}	[0010]	$X^{\{1\}}$
$f^{101}, f^{011}, f^{021}, f^{201}$	[0d1d]	$X^{\{1\}}, X^{\{0,1\}}, X^{\{1,2\}}, X^{\{0,1,2\}}$
f^{100}	[0d10]	$X^{\{1\}}, X^{\{1,2\}}$
f^{010}	[0011]	$X^{\{0,1\}}$
f^{001}	[0d01]	$X^{\{0\}}, X^{\{0,2\}}$
$f^{002}, f^{020}, f^{220}$	[0dd1]	$X^{\{0\}}, X^{\{0,1\}}, X^{\{0,2\}}, X^{\{0,1,2\}}$
f^{210}	[0001]	$X^{\{0\}}$

Πίνακας 3.2: Παραγωγή των εκφράσεων MVESOP των φύλλων

3.4 Σύνθεση

Κατά τη διάρκεια της φάσης σύνθεσης του αλγορίθμου οι εκφράσεις MVESOP των υποσυναρτήσεων συνδυάζονται για να παράγουν τις εκφράσεις MVESOP κάθε κόμβου προς τη ρίζα του δένδρου. Η παραγωγή των νέων εκφράσεων βασίζεται στα αναπτύγματα που ορίζονται στον Ορισμό 11 και πραγματοποιείται επαναληπτικά προς τη ρίζα του γεννήτορος δένδρου. Μπορούμε να διακρίνουμε τρεις διαφορετικές περιπτώσεις, ανάλογα με τον αριθμό των υποσυναρτήσεων που είναι ίσες με την μηδενική συνάρτηση.

Στην πρώτη περίπτωση, όλες οι υποσυναρτήσεις ενός συγκεκριμένου κόμβου είναι μη μηδενικές. Έστω n το επίπεδο ενός κόμβου (που αναπαριστά μια συνάρτηση f που εξαρτάται από $n + 1$ μεταβλητές) και k_0, k_1 και k_2 ο αριθμός των προϋπολογισμένων εκφράσεων για τις υποσυναρτήσεις f^0, f^1 και f^2 της f αντίστοιχα. Οι εκφράσεις από κάθε ζευγάρι υποσυναρτήσεων συνδυάζονται ώστε να παράγουν μια καινούρια έκφραση για την f χρησιμοποιώντας του τύπους $f = \bar{x}_n f^0 \oplus x_n f^1$, $f = f^0 \oplus x_n f^2$ και $f = f^1 \oplus \bar{x}_n f^2$. Ο συνολικός αριθμός των παραγόμενων εκφράσεων είναι $k_0 k_1 + k_0 k_2 + k_1 k_2$. Από αυτές τις εκφράσεις, όσες δεν επαληθεύουν την f , καθώς και όσες έχουν μεγαλύτερο μέγεθος από το ελάχιστο για όλες τις εκφράσεις της f (που εκτιμάται ότι είναι το βάρος της f), απορρίπτονται.

Ο λόγος που μερικές εκφράσεις δεν επαληθεύουν την f είναι ο ακόλουθος. Οι τιμές που ανατίθενται στις εξόδους που είναι ορισμένες ως αδιάφορες δεν είναι ανεξάρτητες ανάμεσα στις τρεις υποσυναρτήσεις. Αυτό σημαίνει ότι μια τυχαία επιλογή της τιμής για ένα συγκεκριμένο αδιάφορο ελαχιστόρο περιορίζει τις τιμές που μπορούν να ανατεθούν στις αντίστοιχες εξόδους των άλλων υποσυναρτήσεων. Αλλά η παραγωγή των εκφράσεων για κάθε υποσυνάρτηση γίνεται ανεξάρτητα στον αλγόριθμό μας. Ως αποτέλεσμα, ο συνδυασμός ενός ζευγαριού εκφράσεων δύο υποσυναρτήσεων μπορεί να συνεπάγεται έναν ασύμβατο συνδυασμό από αναθέσεις τιμών στις αδιάφορες εξόδους και οι παραγόμενες εκφράσεις μπορεί να μην επαληθεύουν τη γονική συνάρτηση. Επομένως, είναι απαραίτητο να επιβεβαιώσουμε ότι κάθε νέα έκφραση επαληθεύει την αντίστοιχη συνάρτηση, πριν γίνει αποδεκτή ως έγκυρη έκφραση, σε όλες τις περιπτώσεις που περιγράφονται προηγουμένως.

Πιο συγκεκριμένα, για δύο εκφράσεις g_a (της υποσυνάρτησης f^a με μέγεθος s_a) και g_b (της υποσυνάρτησης f^b με μέγεθος s_b) μια νέα έκφραση g παράγεται χρησιμοποιώντας τον γενικό τύπο $g = x^c g_a \oplus x^d g_b$, όπου $(a, b, c, d) \in \{(0, 1, \{0\}, \{1\}), (0, 2, \{0, 1\}, \{1\}), (1, 2, \{0, 1\}, \{0\})\}$. Αν υπάρχουν s κοινοί κύβοι στην g_a και στην g_b , τότε για κάθε κοινό κύβο t οι παραγόμενοι κύβοι $x^c \cdot t$ και $x^d \cdot t$ συγχωνεύονται σε έναν κύβο $x^{c \oplus d} \cdot t$. Το μέγεθος της τελικής έκφρασης MVESOP είναι $s_a + s_b - s$.

Παράδειγμα 18 Ας εξετάσουμε την f^{11} και τις υποσυναρτήσεις της στο παράδειγμα του Σχήματος 3.1. Η υποσυνάρτηση f^{110} έχει μόνο μια έκφραση: $X^{\{1\}}$. Οι εκφράσεις για την f^{111} είναι $X^{\{2\}}$, $X^{\{1,2\}}$, $X^{\{0,2\}}$, $X^{\{0,1,2\}}$ και οι εκφράσεις για την f^{112} είναι $X^{\{2\}}$, $X^{\{1,2\}}$, $X^{\{0,2\}}$, $X^{\{0,1,2\}}$.

Ο αριθμός των παραγόμενων εκφράσεων για την f^{11} είναι $1 \cdot 4 + 1 \cdot 4 + 4 \cdot 4 = 24$. Όπως έχει εξηγηθεί, δεν είναι απαραίτητο όλες αυτές οι εκφράσεις να ικανοποιούν την f^{11} , επομένως πρέπει να τις ελέγξουμε.

Σύμφωνα με το ανάπτυγμα Shannon ($f^{11} = x_3^{\{0\}} f^{110} \oplus x_3^{\{1\}} f^{111}$), 4 εκφράσεις παράγονται, οι οποίες ικανοποιούν την f^{11} : $x_3^{\{0\}} X^{\{1\}} \oplus x_3^{\{1\}} X^{\{2\}}$, $x_3^{\{0\}} X^{\{1\}} \oplus x_3^{\{1\}} X^{\{1,2\}}$, $x_3^{\{0\}} X^{\{1\}} \oplus x_3^{\{1\}} X^{\{0,2\}}$, $x_3^{\{0\}} X^{\{1\}} \oplus x_3^{\{1\}} X^{\{0,1,2\}}$.

Σύμφωνα με το ανάπτυγμα θετικό Davio ($f^{11} = f^{110} \oplus x_3^{\{1\}} f^{112}$), 4 νόμιμες εκφράσεις παράγονται: $X^{\{1\}} \oplus x_3^{\{1\}} X^{\{2\}}$, $X^{\{1\}} \oplus x_3^{\{1\}} X^{\{1,2\}}$, $X^{\{1\}} \oplus x_3^{\{1\}} X^{\{0,2\}}$, $X^{\{1\}} \oplus x_3^{\{1\}} X^{\{0,1,2\}}$.

Το ανάπτυγμα αρνητικό Davio ($f^{11} = f^{111} \oplus x_3^{\{0\}} f^{112}$) οδηγεί σε 16 εκφράσεις, αλλά μόνο 4 από αυτές επαληθεύουν την f^{11} : $X^{\{2\}} \oplus x_3^{\{0\}} X^{\{1,2\}}$, $X^{\{1,2\}} \oplus x_3^{\{0\}} X^{\{2\}}$, $X^{\{0,2\}} \oplus x_3^{\{0\}} X^{\{0,1,2\}}$, $X^{\{0,1,2\}} \oplus x_3^{\{0\}} X^{\{0,2\}}$. Οι άλλες 12 εκφράσεις, που δεν επαληθεύουν την f^{11} , απορρίπτονται, π.χ. η έκφραση $X^{\{2\}} \oplus x_3^{\{0\}} X^{\{2\}}$.

Η δεύτερη περίπτωση είναι όταν μία από τις υποσυναρτήσεις είναι μηδέν. Για τις πλήρως ορισμένες συναρτήσεις, όταν μία υποσυνάρτηση είναι μηδέν, τότε οι άλλες δύο υποσυναρτήσεις είναι ίσες και αρκεί να εξετάσουμε μόνο μία από αυτές για να παράγουμε τις κατάλληλες εκφράσεις [91]. Η παρουσία των αδιάφορων όρων στις μη πλήρως ορισμένες συναρτήσεις διαφοροποιεί τη στρατηγική, επειδή θεωρούμε ότι μια συνάρτηση που οι έξοδοί της είναι μηδέν ή αδιάφοροι είναι η μηδενική συνάρτηση. Στην πραγματικότητα, μια διαφορετική ανάθεση από τιμές στις αδιάφορες εξόδους θα οδηγούσε σε μια μη μηδενική συνάρτηση. Επομένως, πρέπει να εξετάσουμε και τις δύο άλλες υποσυναρτήσεις. Για παράδειγμα, ας θεωρήσουμε τη συνάρτηση f^{02} στο Σχήμα 3.1. Η υποσυνάρτηση f^{022} είναι $[0ddd]$ και θεωρείται μηδέν, αλλά οι δύο άλλες υποσυναρτήσεις δεν είναι ίσες (η f^{021} είναι $[0d1d]$ και η f^{020} είναι $[0dd1]$).

Ανάλογα με το ποια υποσυνάρτηση είναι μηδέν, οι προαναφερόμενοι τύποι στον Ορισμό 11 για τα ανάπτυγματα Shannon, θετικό και αρνητικό Davio αλλάζουν κατάλληλα αντικαθιστώντας την αντίστοιχη υποσυνάρτηση με 0, όπως φαίνεται στον Πίνακα 3.3, ώστε να λάβουμε υπ' όψιν και τις δύο (άνισες) υποσυναρτήσεις.

Έστω f^a , f^b , $a, b \in \{0, 1, 2\}$, $a \neq b$ οι δύο μη μηδενικές υποσυναρτήσεις και k_a , k_b ο αριθμός των προϋπολογισμένων εκφράσεων για τις f^a , f^b αντίστοιχα. Ο

Σύνθηξη	Τύποι
$f^0 = 0$	$f = x_n f^1, f = x_n f^2, f = f^1 \oplus \bar{x}_n f^2$
$f^1 = 0$	$f = \bar{x}_n f^0, f = \bar{x}_n f^2, f = f^0 \oplus x_n f^2$
$f^2 = 0$	$f = f^0, f = f^1, f = \bar{x}_n f^0 \oplus x_n f^1$

Πίνακας 3.3: Τύποι όταν μια υποσυνάρτηση είναι μηδέν

αριθμός των νέων εκφράσεων είναι $k_a + k_b + k_a k_b$. Όταν $f = x^i f^j$, $i \subseteq \{0, 1\}$, $j \in \{0, 1, 2\}$ το μέγεθος της νέας έκφρασης είναι ίσο με το μέγεθος της έκφρασης για την f^j . Όταν $f = x_n^i f^a \oplus x_n^j f^b$, $i, j \subseteq \{0, 1\}$, $i \neq j$, το μέγεθος των αντίστοιχων εκφράσεων είναι s_a και s_b και υπάρχουν s κοινοί κύβοι, το μέγεθος της τελικής έκφρασης είναι $s_a + s_b - s$. Προφανώς, οι κοινοί κύβοι συγχωνεύονται σε έναν πολλαπλασιαζόμενο με το κατάλληλο literal της μεταβλητής x_n .

Παράδειγμα 19 *Ας εξετάσουμε την f^{10} του παραδείγματος στο Σχήμα 3.1. Οι εκφράσεις για την f^{100} είναι $X^{\{1\}}$, $X^{\{1,2\}}$ και για την f^{101} είναι $X^{\{1\}}$, $X^{\{1,2\}}$, $X^{\{0,1\}}$, $X^{\{0,1,2\}}$. Η υποσυνάρτηση f^{102} είναι μηδέν και οι εκφράσεις για την f^{10} παράγονται χρησιμοποιώντας τους τύπους $f^{10} = f^{100}$, $f^{10} = f^{101}$, $f^{10} = x_3^{\{0\}} f^{100} \oplus x_3^{\{1\}} f^{101}$.*

Ο πρώτος τύπος οδηγεί στις εκφράσεις $X^{\{1\}}$ και $X^{\{1,2\}}$, οι οποίες επαληθεύουν την f^{10} . Οι ίδιες εκφράσεις βρίσκονται εφαρμόζοντας το δεύτερο τύπο. Δύο άλλες εκφράσεις, οι $X^{\{0,1\}}$ και $X^{\{0,1,2\}}$, που παράγονται χρησιμοποιώντας τη δεύτερη εξίσωση, δεν επαληθεύουν την f^{10} και απορρίπτονται.

Το μέγεθος των περισσότερων εκφράσεων που παράγονται από τον τρίτο τύπο είναι 2, επομένως απορρίπτονται. Όμως, σε δύο περιπτώσεις, υπάρχει ένας κοινός κύβος στις εκφράσεις που αντιστοιχούν στις υποσυναρτήσεις f^{100} και f^{101} και οι παραγόμενες εκφράσεις έχουν μέγεθος ίσο με 1. Αυτές οι εκφράσεις είναι οι $x_3^{\{0\}} X^{\{1\}} \oplus x_3^{\{1\}} X^{\{1\}} = X^{\{1\}}$ και $x_3^{\{0\}} X^{\{1,2\}} \oplus x_3^{\{1\}} X^{\{1,2\}} = X^{\{1,2\}}$. Συγκρίνοντας τις εκφράσεις με τις προηγούμενα υπολογιζόμενες, μπορούμε να δούμε ότι είναι οι ίδιες και τις κρατάμε μια φορά.

Στην τρίτη περίπτωση υπάρχουν ακριβώς δύο μηδενικές υποσυναρτήσεις. Για τον ίδιο λόγο που αναφέρθηκε παραπάνω, η τρίτη υποσυνάρτηση εξετάζεται. Οι τύποι των αναπτυγμάτων αλλάζουν, όπως δείχνει ο Πίνακας 3.4.

Συνθηξη	Τύποι
$f^0 = 0, f^1 = 0$	$f = x_n f^2, f = \bar{x}_n f^2$
$f^0 = 0, f^2 = 0$	$f = x_n f^1, f = f^1$
$f^1 = 0, f^2 = 0$	$f = \bar{x}_n f^0, f = f^0$

Πίνακας 3.4: Τύποι όταν δύο υποσυναρτήσεις είναι μηδέν

Έστω f^i , $i \in \{0, 1, 2\}$ η μη μηδενική υποσυνάρτηση και k ο αριθμός των ήδη υπολογισμένων εκφράσεων. Ο αριθμός των νέων MVESOP είναι $2k$. Για μία συγκεκριμένη έκφραση της f^i με μέγεθος s η νέα έκφραση της f έχει μέγεθος s .

Παράδειγμα 20 Ας εξετάσουμε την f^{22} του παραδείγματος στο Σχήμα 3.1. Οι υποσυναρτήσεις f^{221} και f^{222} είναι ίσες με το μηδέν. Οι εκφράσεις της f^{220} είναι οι $X^{\{0\}}$, $X^{\{0,2\}}$, $X^{\{0,1\}}$, $X^{\{0,1,2\}}$. Χρησιμοποιούμε τις εξισώσεις $f^{22} = x_3^{\{0\}} f^{220}$ και $f^{22} = f^{220}$ και οι παραγόμενες εκφράσεις για την f^{22} είναι οι $x_3^{\{0\}} X^{\{0\}}$, $x_3^{\{0\}} X^{\{0,2\}}$, $x_3^{\{0\}} X^{\{0,1\}}$, $x_3^{\{0\}} X^{\{0,1,2\}}$, $X^{\{0\}}$, $X^{\{0,2\}}$, $X^{\{0,1\}}$ και $X^{\{0,1,2\}}$. Όλες οι παραπάνω εκφράσεις επαληθεύουν την f^{22} .

Όπως είδαμε, για κάθε κόμβο, επιλέγουμε να κρατήσουμε μόνο τις εκφράσεις με το μικρότερο μέγεθος και απορρίπτουμε τις άλλες. Αυτό μπορεί να χειροτερέψει την ποιότητα της τελικής λύσης, αλλά βελτιώνει το χρόνο εκτέλεσης του αλγορίθμου.

3.5 Ο αλγόριθμος DCMIN

Βασίζομενοι στα προηγούμενα, περιγράφουμε στους Αλγόριθμους 1-3 τον αλγόριθμο DCMIN για την παραγωγή και την ελαχιστοποίηση εκφράσεων ESOP για τυχαίες μη πλήρως ορισμένες switching συναρτήσεις πολλών εξόδων. Ο αλγόριθμος ακολουθεί ακριβώς την άτυπη περιγραφή που δόθηκε στις Ενότητες 3.1-3.4 και είναι εύκολα κατανοητός.

Algorithm 1: DCMIN(input: MVMT formulation of f) returns: a list of MVESOP expressions of f

```

Tree: generator tree of  $f$ ;
Expr( $f$ ): list of MVESOP expressions of  $f$ ;
begin
  Tree=Decomposition( $f$ );
  Expr( $f$ )=Composition(Tree);
  return Expr( $f$ );
end

```

Ο αλγόριθμος DCMIN υλοποιήθηκε σε C++. Η MVMT αναπαράσταση χρησιμοποιήθηκε για να αναπαρασταθεί η αρχική συνάρτηση και κάθε υποσυνάρτηση. Αφ' ετέρου κάθε έκφραση MVESOP αναπαραστάθηκε σαν μια λίστα από κύβους σε PN αναπαράσταση, η οποία είναι περισσότερο συμπαγής. Μια συνάρτηση με n δυαδικές μεταβλητές και μια μεταβλητή k τιμών χρειάζεται $2^{n+\lceil \lg k \rceil}$ bits για να αναπαρασταθεί σε MVMT αναπαράσταση. Ένας κύβος από αυτή τη συνάρτηση χρειάζεται $2n + k$ bits για να αναπαρασταθεί σε PN αναπαράσταση. Το μειονέκτημα της PN είναι ότι μπορεί να αναπαραστήσει μόνο κύβους, αλλά όχι οποιαδήποτε τυχαία συνάρτηση, για την οποία μία λίστα από κύβους είναι απαραίτητη. Για έναν αρκετά μεγάλο αριθμό μεταβλητών και έναν αρκετά μικρό αριθμό κύβων, η PN αναπαράσταση είναι προτιμότερη από την MVMT αναπαράσταση. Επιπλέον, ο αλγόριθμός μας ξοδεύει ένα μεγάλο ποσοστό του χρόνου εκτέλεσής του συγκρίνοντας κύβους, έτσι η επιλογή της PN αναπαράστασης είναι λογική.

Algorithm 2: Decomposition(input: MVMT formulation of f) returns: a ternary generator tree for f

f^0, f^1, f^2 : subfunctions of f in MVMT formulation;
 $CurNode$: current node of ternary generator tree;
 $f^1Node, f^0Node, f^2Node$: left, middle and right child of $CurNode$ respectively;

```

begin
  Assign  $f$  to  $CurNode$ ;
  if  $f = 0$  then
    | return;
  end if
  if  $f$  is a function of 1 (multiple-valued) variable then
    | return;
  end if
  generate  $f^1$ ;
  generate  $f^0$ ;
  generate  $f^2$ ;
   $CurNode \rightarrow f^0Node = \text{Decomposition}(f^0)$ ;
   $CurNode \rightarrow f^1Node = \text{Decomposition}(f^1)$ ;
   $CurNode \rightarrow f^2Node = \text{Decomposition}(f^2)$ ;
end

```

Ο αλγόριθμος DCMIN δοκιμάστηκε με μερικές μετροσυναρτήσεις, αλλά αποδείχθηκε μη αποδοτικός όσον αφορά το χρόνο εκτέλεσης για μεγάλους αριθμούς μεταβλητών και αδιάφορων όρων. Η παρουσία των αδιάφορων όρων αυξάνει τον αριθμό των παραγόμενων εκφράσεων εκθετικά. Για παράδειγμα, ακόμα και για μια υποσυνάρτηση στο τελευταίο επίπεδο του γεννήτορος δένδρου, 10 αδιάφοροι όροι οδηγούν σε $2^{10} = 1024$ εκφράσεις.

Επομένως, αλλάξαμε ελαφρά τον αλγόριθμο ώστε να κρατάμε λιγότερες εκφράσεις σε κάθε κόμβο του τριαδικού γεννήτορος δένδρου. Έτσι, ο χρόνος εκτέλεσης του αλγορίθμου μειώνεται, αλλά η ποιότητα της λύσης χειροτερεύει. Αυτή η παραλλαγή του DCMIN, που ονομάζεται SimpleDCMIN, είναι πολύ χρήσιμη για να υλοποιήσουμε ένα βελτιωμένο αλγόριθμο, όπως εξηγείται στην επόμενη ενότητα. Ο αριθμός q των εκφράσεων ESOP σε κάθε κόμβο είναι μια παράμετρος του SimpleDCMIN, και ο αλγόριθμος δημιουργεί τόσες εκφράσεις ανά κόμβο όσες μπορεί, χρησιμοποιώντας τα αναπτύγματα που έχουμε ήδη περιγράψει, αλλά όχι περισσότερες από q εκφράσεις. Αν μπορούν να παραχθούν περισσότερες εκφράσεις, τότε οι q εκφράσεις επιλέγονται τυχαία.

3.6 Ο αλγόριθμος QuickDCMIN

Ένας νέος αλγόριθμος, βασισμένος στον DCMIN, σχεδιάστηκε και υλοποιήθηκε. Ο νέος ευριστικός αλγόριθμος ονομάστηκε QuickDCMIN και είναι πολύ γρηγορότερος από τον προηγούμενο. Ο QuickDCMIN είναι επαναληπτικός και

Algorithm 3: Composition(input: ternary generator tree of f) returns (a list of MVESOP expressions of f)

f^0, f^1, f^2 : subfunctions of f in MVMT formulation;
 $CurNode$: current node of ternary generator tree;
 $f^1Node, f^0Node, f^2Node$: left, middle and right child of $CurNode$ respectively;
 $Expr(f)$: list of MVESOP expressions of f ;
begin
 * f is the function assigned to $CurNode$ *\br/>
 if f^0Node exists **then**
 | $f^0 = CurNode \rightarrow f^0Node \rightarrow f$;
 | $Expr(f^0) = \text{Composition}(CurNode \rightarrow f^0Node)$;
 end if
 else if
 | $f^0 = 0$;
 end if
 if f^1Node exists **then**
 | $f^1 = CurNode \rightarrow f^1Node \rightarrow f$;
 | $Expr(f^1) = \text{Composition}(CurNode \rightarrow f^1Node)$;
 end if
 else if
 | $f^1 = 0$;
 end if
 if f^2Node exists **then**
 | $f^2 = CurNode \rightarrow f^2Node \rightarrow f$;
 | $Expr(f^2) = \text{Composition}(CurNode \rightarrow f^2Node)$;
 end if
 else if
 | $f^2 = 0$;
 end if
 if $f^0 \neq 0$ AND $f^1 \neq 0$ AND $f^2 \neq 0$ **then**
 | $Expr(f) =$ expressions produced using Shannon, positive and
 | negative Davio expansions and merging common cubes
 end if
 if ($f^0 = 0$ AND $f^1 = 0$) OR ($f^0 = 0$ AND $f^2 = 0$) OR ($f^1 = 0$ AND
 $f^2 = 0$) **then**
 | $Expr(f) =$ expressions produced using formulas of Table 3.4
 end if
 if $f^0 = 0$ OR $f^1 = 0$ OR $f^2 = 0$ **then**
 | $Expr(f) =$ expressions produced using formulas of Table 3.3 and
 | merging common cubes
 end if
 return $Expr(f)$;
end

χρησιμοποιεί τον SimpleDCMIN σαν ένα τελεστή μετασχηματισμού κύβων και ελαχιστοποίησης. Μια περιγραφή του βλέπουμε στον Αλγόριθμο 4.

Algorithm 4: QuickDCMIN(input: MVMT formulation of f) returns: a MVESOP expression of f

```

Tree: generator tree of  $f$ ;
Expr( $f$ ): a MVESOP expression of  $f$ ;
on-trms( $f$ ): set of terms in the ON-set of  $f$ ;
dc-trms( $f$ ): set of terms in the DC-set of  $f$ ;
begin
  generate on-trms set (initial cover) using EXORCISM-4 or
  SimpleDCMIN;
  generate dc-trms set from MVMT formulation;
  for  $i = 1$  to  $K$  do
    choose randomly  $N$  terms from on-trms;
    generate the function  $y$ , defined by these  $N$  terms and all dc-trms;
    Tree=Decomposition( $y$ );
    Expr( $y$ )=Composition(Tree); \* only  $q$  expressions produced *
    if size of Expr( $y$ )  $\leq N$  then
      remove the selected  $N$  terms from on-trms set;
      insert terms of Expr( $y$ ) into on-trms set;
    end if
  end for
  Expr( $f$ ) = expression produced from terms of on-trms;
  return Expr( $f$ );
end

```

Πολλοί ευριστικοί αλγόριθμοι για την ελαχιστοποίηση εκφράσεων ESOP (όπως ο QuickDCMIN) βασίζονται σε επαναληπτικούς μετασχηματισμούς κύβων [24, 46]. Ένα αρχικό κάλυμμα της συνάρτησης προς ελαχιστοποίηση υπολογίζεται και, στη συνέχεια, ένας αριθμός κύβων επιλέγονται και μετασχηματίζονται χρησιμοποιώντας ένα σύνολο κανόνων. Αυτοί οι μετασχηματισμοί εκτελούνται επαναληπτικά και οδηγούν τελικά σε μικρότερα καλύμματα. Αυτό το είδος των αλγορίθμων χαρακτηρίζεται από την μέθοδο υπολογισμού του αρχικού καλύμματος, από την μέθοδο επιλογής των κύβων σε κάθε επανάληψη και από τους κανόνες μετασχηματισμού που χρησιμοποιούνται για να ελαχιστοποιήσουμε τη συνάρτηση.

Ο QuickDCMIN χρησιμοποιεί σαν είσοδο ένα αρχικό κάλυμμα που παράγεται από ένα άλλο πρόγραμμα. Μπορούμε να διαλέξουμε οποιαδήποτε μέθοδο επιθυμούμε, αλλά σε αυτή την εργασία χρησιμοποιούμε είτε τον SimpleDCMIN ή έναν άλλο πολύ γνωστό αλγόριθμο ελαχιστοποίησης, τον EXORCISM-4 [24], για να αποκτήσουμε το αρχικό κάλυμμα. Ο SimpleDCMIN πραγματοποιεί μια αρχική μείωση στον αριθμό των όρων (η παράμετρος του αριθμού των παραγόμενων εκφράσεων ανά κόμβο τίθεται κατάλληλα ώστε να μειωθεί ο χρόνος εκτέλεσης σε λιγότερο από ένα τέταρτο της ώρας) και η έξοδος του δίνεται σαν είσοδος στον QuickESCT-DC. Όμως, ο EXORCISM-4 ελαχιστοποιεί πλήρως ορισμένες συναρ-

τήσεις. Επομένως, για μια μη πλήρως ορισμένη συνάρτηση, αναθέτουμε την τιμή 0 για κάθε αδιάφορη έξοδο. Το αποτέλεσμα δίνεται ως είσοδος στον EXORCISM-4. Εναλλακτικά, αναθέτουμε την τιμή 1 σε κάθε αδιάφορο όρο και δίνουμε το αποτέλεσμα ως είσοδο στον EXORCISM-4. Η έξοδος (και για τις δύο περιπτώσεις) είναι το αρχικό κάλυμμα της συνάρτησης που θέλουμε να ελαχιστοποιήσουμε. Μπορούμε να αποκαλούμε 0-κάλυμμα και 1-κάλυμμα αντίστοιχα τα δύο αρχικά καλύμματα για μελλοντική αναφορά. Όμως, στα επόμενα στάδια του αλγορίθμου, λαμβάνουμε επίσης υπ' όψιν το DC-set της συνάρτησης.

Το επόμενο βήμα είναι η μεθοδολογία για την επιλογή των κύβων σε κάθε επανάληψη του αλγορίθμου. Ένα σύνολο όρων (το οποίο ονομάζεται *CurrentTerms*) ανανεώνεται σε κάθε επανάληψη και αποτελεί πρακτικά την παραγόμενη έκφραση. Πριν από την πρώτη επανάληψη αυτό το σύνολο είναι ίσο με το ON-set. Όλοι οι όροι του DC-set επιλέγονται σε κάθε επανάληψη, προσθέτοντας ένα σταθερό αριθμό όρων από το σύνολο *CurrentTerms*. Αυτός ο αριθμός είναι μια παράμετρος του προγράμματος. Οι όροι επιλέγονται τυχαία, χρησιμοποιώντας μια γεννήτρια τυχαίων αριθμών. Επομένως, κάθε εκτέλεση του αλγορίθμου παράγει διαφορετικά αποτελέσματα.

Η λειτουργία των μετασχηματισμών βασίζεται στον SimpleDCMIN. Σε κάθε επανάληψη του QuickDCMIN, ο αλγόριθμος SimpleDCMIN εκτελείται, παίρνοντας σαν είσοδο τη συνάρτηση που ορίζεται από τους επιλεγμένους κύβους του συνόλου *CurrentTerms* και όλους τους ελαχιστόρους του DC-set. Αν το μέγεθος της παραγόμενης MVESOP έκφρασης είναι μικρότερο από ή ίσο με τον αριθμό των επιλεγμένων κύβων από το σύνολο *CurrentTerms*, τότε αυτοί οι κύβοι αντικαθίστανται από τους κύβους της έκφρασης. (Στην περίπτωση της ανισότητας, το μέγεθος της αρχικής συνάρτησης μειώνεται.) Αλλιώς, το σύνολο *CurrentTerms* παραμένει το ίδιο. Αυτή η διαδικασία επιβεβαιώνει ότι ο QuickDCMIN συγκλίνει, πιθανά προς μια ακριβή λύση της συνάρτησης εισόδου, μετά από έναν άπειρο αριθμό επαναλήψεων.

Υπάρχουν 2 τρόποι λειτουργίας του προγράμματος. Στον πρώτο τρόπο υπάρχει ένας σταθερός αριθμός επαναλήψεων που θα τρέξει το πρόγραμμα και έπειτα σταματάει και παίρνουμε το αποτέλεσμα. Στον δεύτερο τρόπο αν δεν υπάρξει καθόλου μείωση στο μέγεθος της συνάρτησης μετά από έναν συγκεκριμένο αριθμό επαναλήψεων, το πρόγραμμα σταματάει. Σε κάθε περίπτωση δίνουμε 2 επιπλέον παραμέτρους στο πρόγραμμα. Η μία καθορίζει τον τρόπο λειτουργίας του προγράμματος και η δεύτερη καθορίζει τον αριθμό των επαναλήψεων, που ερμηνεύεται ανάλογα με την πρώτη παράμετρο.

Η μοναδική διαφορά του SimpleDCMIN (που χρησιμοποιείται σε κάθε επανάληψη του QuickDCMIN) σε σχέση με τον DCMIN είναι ότι σε κάθε κόμβο του γεννήτορος δένδρου μόνο q το πολύ εκφράσεις MVESOP παράγονται. Αυτό βελτιώνει δραστικά την ταχύτητα του αλγορίθμου και επιτρέπει την επαναληπτική χρήση του για πολλές επαναλήψεις. Αν μπορούν να παραχθούν περισσότερες εκφράσεις, επιλέγονται q τυχαία, και αυτό είναι ένας πρόσθετος λόγος που κάθε εκτέλεση του προγράμματος παράγει διαφορετικά αποτελέσματα. Επιπλέον, οι λύσεις που παράγονται από τον QuickDCMIN φαίνεται ότι είναι καλύτερες από αυτές του αυθεντικού DCMIN, όσον αφορά το μέγεθος των παραγόμενων εκφράσεων, ακόμα και όταν $q = 1$.

3.7 Πειραματικά αποτελέσματα

Ο αλγόριθμος QuickDCMIN υλοποιήθηκε σε C++ και δοκιμάστηκε σε έναν υπολογιστή Core 2 Quad 2,4GHz με 4GB RAM που έτρεχε Linux. Όπως αναφέρεται στην Ενότητα 3.6, η τυχαία επιλογή των όρων βασίζεται σε μια γεννήτρια τυχαίων αριθμών. Επίσης, επιλέγουμε τυχαία τις εκφράσεις που θα κρατήσουμε σε κάθε κόμβο, αν παράγονται περισσότερες από τον αριθμό που δίνουμε ως παράμετρο. Σαν συνέπεια, κάθε εκτέλεση του προγράμματος παράγει διαφορετικά αποτελέσματα. Οι παράμετροι του προγράμματος είναι οι ακόλουθες: α) ο τρόπος που θα τρέξει το πρόγραμμα σύμφωνα με όσα αναφέρθηκαν στην Ενότητα 3.6, β) ο αριθμός των επαναλήψεων που ερμηνεύεται ανάλογα με την προηγούμενη παράμετρο, γ) ο αριθμός των όρων από το σύνολο *CurrentTerms* που επιλέγονται σε κάθε επανάληψη και δ) ο αριθμός των εκφράσεων που παράγονται για κάθε κόμβο του γεννητούρου δένδρου όταν εφαρμόζουμε τον μετασχηματισμό που βασίζεται στον SimpleESCT-DC.

Το πρόγραμμα δοκιμάστηκε με ένα σύνολο από πρότυπες μετροσυναρτήσεις. Στον πρώτο τρόπο λειτουργίας, όπου το πρόγραμμα τρέχει για έναν προαποφασισμένο αριθμό επαναλήψεων, το τρέξαμε για 100000 επαναλήψεις και 5 – 10 επιλεγόμενους όρους ανά επανάληψη. Ο αριθμός των παραγόμενων εκφράσεων για κάθε υποσυνάρτηση σε κάθε κόμβο τέθηκε σε 1 για να επιτρέψει περισσότερες επαναλήψεις σε λιγότερο χρόνο. Για κάθε σύνολο παραμέτρων τρέξαμε το πρόγραμμα αρκετές φορές και επιλέξαμε την καλύτερη περίπτωση. Προφανώς, ένας μεγαλύτερος αριθμός επαναλήψεων οδηγεί, συνήθως, σε μικρότερο μέγεθος των παραγόμενων εκφράσεων ESOP.

Εναλλακτικά, το πρόγραμμα δοκιμάστηκε στον δεύτερο τρόπο λειτουργίας, για 5000 επαναλήψεις χωρίς μείωση του μεγέθους των παραγόμενων εκφράσεων και για 5 – 10 επιλεγόμενους όρους ανά επανάληψη. Ο αριθμός των παραγόμενων εκφράσεων για κάθε υποσυνάρτηση σε κάθε κόμβο τέθηκε ξανά σε 1 και τρέξαμε το πρόγραμμα αρκετές φορές διαλέγοντας την καλύτερη περίπτωση. Επιλέξαμε 5000 επαναλήψεις για να επιτύχουμε μια ισορροπία ανάμεσα στο χρόνο εκτέλεσης και το μέγεθος της λύσης.

Καταρχήν θέλουμε να δείξουμε την ανωτερότητα της προσέγγισής μας επειδή εκμεταλλευόμαστε την παρουσία των αδιάφορων όρων. Έτσι, τροποποιούμε τα αρχεία PLA των πρότυπων μετροσυναρτήσεων για μη πλήρως ορισμένες συναρτήσεις [103], αντικαθιστώντας όλα τα σύμβολα που υποδεικνύουν ότι ένας όρος ανήκει στο DC-set, με τα σύμβολα για το OFF-set στην μια περίπτωση ή με τα σύμβολα για το ON-set στην άλλη περίπτωση. Με αυτόν τον τρόπο δημιουργούμε δύο διαφορετικές εκδόσεις αρχείων PLA που αντιστοιχούν στα προαναφερόμενα 0-κάλυμμα και 1-κάλυμμα. Εκτελέσαμε αλγόριθμους για πλήρως ορισμένες συναρτήσεις για αυτές τις δύο τροποποιήσεις των προτύπων μετροσυναρτήσεων, καθώς και τους αλγόριθμους μας για τις αρχικές μη πλήρως ορισμένες μετροσυναρτήσεις.

Συγκεκριμένα, χρησιμοποιήσαμε τους αλγόριθμους QuiXOR [93] και EXOR-CISM-4 για το 0-κάλυμμα και το 1-κάλυμμα των μετροσυναρτήσεων και τον αλγόριθμο QuickDCMIN για τις μη πλήρως ορισμένες μετροσυναρτήσεις. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 3.5 για τον πρώτο τρόπο λειτουργίας, και στον Πίνακα 3.6 για το δεύτερο τρόπο λειτουργίας. Όταν τρέξαμε τον αλ-

Συνάρτηση	QuiXOR	QuiXOR	EXOR- CISM-4	EXOR- CISM-4	Quick- DCMIN	
	dc=0	dc=1	dc=0	dc=1	με dc	Χρόνος(s)
alu2	67	97	67	97	40	1532.34
alu3	48	89	48	89	39	20.33
apla	45	46	45	46	36	1.67
b11	26	26	26	26	26	0
bench	27	44	28	44	18	0
bw	22	22	22	22	22	0
dk17	30	31	30	31	23	0
dk27	14	15	14	15	11	0
ex1010	633	701	633	701	422	1484.35
fout	53	52	53	52	41	0
inc	27	26	27	26	24	227.27
p1	57	-	57	56	44	59.81
p3	37	40	37	40	29	8.16

Πίνακας 3.5: Εχμετάλλευση των αδιάφορων όρων για εκφράσεις ESOP (1ος τρόπος λειτουργίας)

γόριθμο QuickDCMIN, χρησιμοποιήσαμε σαν αρχικά καλύμματα τις εξόδους από τον EXORCISM-4 (παίρνοντας ως είσοδο το 0-κάλυμμα και το 1-κάλυμμα αντίστοιχα) και τον SimpleDCMIN (παίρνοντας ως είσοδο την μη πλήρως ορισμένη μετροσυνάρτηση). Ο SimpleDCMIN παράγει σε ορισμένες περιπτώσεις καλύτερο αρχικό κάλυμμα από τον EXORCISM-4, επειδή εχμεταλλεύεται τους αδιάφορους όρους. Διαλέξαμε την καλύτερη από τις τρεις περιπτώσεις (χρησιμοποιώντας τα 3 διαφορετικά αρχικά καλύμματα) για κάθε μετροσυνάρτηση.

Κατά τον πρώτο τρόπο λειτουργίας, ο χρόνος εκτέλεσης που παρουσιάζεται αντιστοιχεί στο χρόνο μέχρι την τελευταία μείωση του αριθμού των όρων της συνάρτησης. Σε πολλές περιπτώσεις, ο χρόνος εκτέλεσης είναι πρακτικά ίσος με μηδέν. Η εξήγηση είναι ότι το αρχικό κάλυμμα (που παίρνουμε από το EXORCISM-4 ή το SimpleDCMIN) είναι αρκετά μικρό και δεν συμβαίνει περαιτέρω ελαχιστοποίηση. Στο δεύτερο τρόπο λειτουργίας δίνουμε τον συνολικό χρόνο που χρειάστηκε, που εκτιμούμε ότι είναι ένα πιο αντικειμενικό μέτρο του απαιτούμενου χρόνου. Όπως παρατηρούμε, ο αλγόριθμος QuickDCMIN εχμεταλλεύεται την παρουσία των αδιάφορων όρων, δίνοντας σε όλες τις περιπτώσεις καλύτερα αποτελέσματα συγκρινόμενος με τα αποτελέσματα για τις πλήρως ορισμένες τροποποιήσεις των μετροσυναρτήσεων.

Στον Πίνακα 3.7 συγκρίνουμε την ποιότητα (όπως υποδεικνύεται από το μέγεθος των εκφράσεων) των αποτελεσμάτων που πήραμε (με τον πρώτο τρόπο εκτέλεσης) με τον QuickDCMIN με τα αποτελέσματα τριών άλλων γνωστών αλγορίθμων που μπορέσαμε να βρούμε και που έχουν δώσει δημοσιευμένα αποτελέσματα για μη πλήρως ορισμένες συναρτήσεις. Οι αλγόριθμοι είναι ο MINT [35], ο EXORCISM-MV-2 [81] και ο EXORCISM-MV-3 [82]. Τα αποτελέσματα αυτών των αλγορίθμων τα βρίσκουμε στις αντίστοιχες δημοσιεύσεις [35, 81, 82]. Οι χρόνοι αφορούν διαφορετικά υπολογιστικά συστήματα, επίσης. Στον Πίνακα 3.8 παρουσιάζουμε τα αντίστοιχα αποτελέσματα χρησιμοποιώντας το δεύτερο τρόπο εκτέλεσης. Όπως

Συνάρτηση	QuiXOR	QuiXOR	EXOR-	EXOR-	Quick-	
	dc=0	dc=1	CISM-4	CISM-4	DCMIN	
	Μέγεθος	Μέγεθος	dc=0	dc=1	Μέγεθος	Χρόνος(s)
alu2	67	97	67	97	41	484.29
alu3	48	89	48	89	39	558.46
apla	45	46	45	46	37	55.27
b11	26	26	26	26	26	116.45
bench	27	44	28	44	18	9.3
bw	22	22	22	22	22	6.92
dk17	30	31	30	31	23	51.96
dk27	14	15	14	15	11	19.18
ex1010	633	701	633	701	423	142.65
fout	53	52	53	52	41	18.17
inc	27	26	27	26	24	102.64
p1	57	-	57	56	47	63.6
p3	37	40	37	40	31	47.35

Πίνακας 3.6: Εχμετάλλευση των αδιάφορων όρων για εκφράσεις ESOP (2ος τρόπος λειτουργίας)

είναι φανερό, η σύγκριση δείχνει ότι ο αλγόριθμός μας παράγει καλύτερα αποτελέσματα συγκρινόμενος με τους προηγούμενους, που αφορούν μη πλήρως ορισμένες συναρτήσεις. Το σύμβολο '-' σημαίνει ότι δεν υπάρχουν δημοσιευμένα αποτελέσματα στις αντίστοιχες δημοσιεύσεις για αυτές τις μετροσυναρτήσεις.

Συνάρτηση	MINT	EXORCISM -MV-2	EXORCISM -MV-3	Quick- DCMIN
alu2	47	-	-	40
alu3	41	-	-	39
apla	44	-	-	36
b11	-	-	-	26
bench	-	24	-	18
bw	-	22	22	22
dk17	30	-	-	23
dk27	14	-	-	11
ex1010	-	695	670	422
fout	-	48	-	41
inc	-	26	26	24
p1	-	53	-	44
p3	-	39	-	29

Πίνακας 3.7: Σύγκριση του QuickDCMIN με τον MINT, τον EXORCSIM-MV-2 και τον EXORCISM-MV-3 (1ος τρόπος λειτουργίας)

Συνάρτηση	MINT		EXORCISM -MV-2		EXORCISM -MV-3		QuickDCMIN (για ESOPs)	
	Μέγεθος	Χρόνος(s)	Μέγεθος	Χρόνος(s)	Μέγεθος	Χρόνος(s)	Μέγεθος	Χρόνος(s)
alu2	47	224	-	-	-	-	41	484.29
alu3	41	60	-	-	-	-	39	558.46
apla	44	210	-	-	-	-	37	55.27
b11	-	-	-	-	-	-	26	116.45
bench	-	-	24	15	-	-	18	9.3
bw	-	-	22	0.6	22	1.1	22	6.92
dk17	30	118	-	-	-	-	23	51.96
dk27	14	14	-	-	-	-	11	19.18
ex1010	-	-	695	16506.3	670	1141.5	423	142.65
fout	-	-	48	60	-	-	41	18.17
inc	-	-	26	1.7	26	1.7	24	102.64
p1	-	-	53	107	-	-	47	63.6
p3	-	-	39	50	-	-	31	47.35

Πίνακας 3.8: Σύγκριση του QuickDCMIN με τους MINT, EXORCSIM-MV-2 και EXORCISM-MV-3 (2ος τρόπος λειτουργίας)

Κεφάλαιο 4

Ακριβείς εκφράσεις ESOP για μη πλήρως ορισμένες συναρτήσεις

4.1 Θεωρητικό υπόβαθρο

Ορισμός 24 Ορίζουμε το G^n ως το σύνολο όλων των συναρτήσεων n μεταβλητών.

Ορισμός 25 Για μία συνάρτηση f , ορίζουμε ως f_{ON} τη συνάρτηση που αποτελείται μόνο από τους ελαχιστόρους του ON -set της f . Η f_{DC} αποτελείται μόνο από τους ελαχιστόρους του DC -set της f , οι οποίοι θεωρούνται ως το ON -set για την f_{DC} .

Παράδειγμα 21 Έστω f μια μη πλήρως ορισμένη συνάρτηση με 3 μεταβλητές εισόδου, της οποίας τα ON -set, OFF -set και DC -set είναι όπως ακολούθως, όπου αριθμούμε τους ελαχιστόρους στο δυαδικό και το δεκαδικό σύστημα:

$$\begin{aligned} ON\text{-set} &= \{\bar{x}_1\bar{x}_2\bar{x}_3, \bar{x}_1x_2\bar{x}_3, \bar{x}_1x_2x_3, x_1x_2\bar{x}_3\} = \{000, 010, 011, 110\} = \{0, 2, 3, 6\} \\ OFF\text{-set} &= \{\bar{x}_1\bar{x}_2x_3, x_1\bar{x}_2x_3\} = \{001, 101\} = \{1, 5\} \\ DC\text{-set} &= \{x_1\bar{x}_2\bar{x}_3, x_1x_2x_3\} = \{100, 111\} = \{4, 7\} \end{aligned}$$

Τότε, η f μπορεί να είναι οποιαδήποτε από τις ακόλουθες συναρτήσεις σε MT αναπαράσταση:

$$[01001101], [01011101], [11001101], [11011101] \text{ ή } [4d], [5d], [cd], [dd]$$

σε δεκαεξαδική μορφή, για όλους τους δυνατούς συνδυασμούς των αδιάφορων όρων.

$$\text{Έχουμε } f_{ON} = [01001101] = [4d] \text{ και } f_{DC} = [10010000] = [90].$$

Ορισμός 26 Η συμπίεσμένη συνάρτηση (*reduced function*) $red(f)$ μιας δεδομένης μη πλήρως ορισμένης ή πλήρως ορισμένης συνάρτησης f , είναι η συνάρτηση που παίρνουμε αν παραλείψουμε, στην *MT* αναπαράσταση, όλα τα *bits* που αντιστοιχούν στους ελαχιστόρους ενός συνόλου *DC*.

Σημειώνουμε ότι αν η f είναι μια μη πλήρως ορισμένη συνάρτηση, το σύνολο *DC* του προηγούμενου ορισμού είναι ταυτόσημο με το *DC-set* της f .

Παράδειγμα 22 Για τη συνάρτηση του Παραδείγματος 21 η συμπίεσμένη συνάρτηση της f είναι $red(f) = 101101$, η οποία υπολογίζεται αν παραλείψουμε τα *bits* 4 και 7, που αντιστοιχούν στο *DC-set*. Παρατηρούμε ότι η $red(f)$ είναι η ίδια για όλες τις συναρτήσεις f του Παραδείγματος 21, για όλους τους δυνατούς συνδυασμούς των αδιάφορων ελαχιστόρων.

Ορισμός 27 Ο συμπίεσμένος πίνακας (*reduced table*) RT_{DC} των μη πλήρως ορισμένων συναρτήσεων $f : f \in G^n$ με ένα κοινό σύνολο αδιάφορων όρων *DC*, ορίζεται ως ο πίνακας για τον οποίο $RT_{DC}(red(f)) = \min(w(f))$, για όλες τις f με την ίδια $red(f)$.

Παράδειγμα 23 Ο συμπίεσμένος πίνακας για το *DC-set* της συνάρτησης του Παραδείγματος 21 παρουσιάζεται στον Πίνακα 4.2. Μπορούμε να δούμε ότι ο πίνακας RT_{DC} περιέχει το βάρος 2 για τη συνάρτηση $red(f)$. Αυτό είναι αληθές, σύμφωνα με τον Ορισμό 26, και τον Πίνακα 4.1, ο οποίος δίνει το βάρος των συναρτήσεων 3 μεταβλητών εισόδου, αφού το 2 είναι το μικρότερο βάρος για όλες τις δυνατές συναρτήσεις ($[4d], [5d], [cd], [dd]$) με την ίδια $red(f)$.

Συνάρτηση f	Βάρος w
$[00] = [00000000]$	0
⋮	⋮
$[4d] = [01001101]$	3
⋮	⋮
$[5d] = [01011101]$	2
⋮	⋮
$[cd] = [11001101]$	2
⋮	⋮
$[dd] = [11011101]$	2
⋮	⋮

Πίνακας 4.1: Πίνακας *RT* για τη συνάρτηση του Παραδείγματος 21

Ορισμός 28 Η αντιπροσωπευτική συνάρτηση (*representative function*) $rep(f)$ μιας μη πλήρως ορισμένης ή πλήρως ορισμένης συνάρτησης, είναι αυτή (αυτές), ανάμεσα σε όλες τις συναρτήσεις f με την ίδια συμπίεσμένη συνάρτηση, με βάρος $w(f) = RT_{DC}(red(f))$.

$red(f)$	Βάρος w
[000000]	0
⋮	⋮
[101101]	2
⋮	⋮

Πίνακας 4.2: Πίνακας RT_{DC} για τη συνάρτηση του Παραδείγματος 21

Επίσης, αν η f είναι μη πλήρως ορισμένη, το σύνολο DC του προηγούμενου ορισμού είναι ταυτόσημο με το DC-set της f .

Παράδειγμα 24 Σύμφωνα με τον Ορισμό 28, η αντιπροσωπευτική συνάρτηση $rep(f)$, για τη συνάρτηση του Παραδείγματος 21, μπορεί να είναι οποιαδήποτε από τις συναρτήσεις $[5d]$, $[cd]$ και $[dd]$ με βάρος $w = 2$.

Οι περισσότεροι αλγόριθμοι ελαχιστοποίησης [21, 53, 28, 90, 26, 27, 89, 91, 93] βασίζονται στις ακόλουθες προφανείς σχέσεις:

$$f = \bar{x}_1(f^0 \oplus g) \oplus x_1(f^1 \oplus g) \oplus g = (f^0 \oplus g) \oplus x_1(f^2 \oplus g) \oplus \bar{x}_1 g = (f^1 \oplus g) \oplus \bar{x}_1(f^2 \oplus g) \oplus x_1 g \quad (4.1)$$

όπου $f \in G^n$, $g \in G^{n-1}$ και f^0, f^1, f^2 είναι οι υποσυναρτήσεις της f . Αν θεωρήσουμε όλες τις δυνατές συναρτήσεις g και βρούμε τα βάρη για τις $g, f^0 \oplus g, f^1 \oplus g, f^2 \oplus g$, τότε μπορούμε να βρούμε μια g τέτοια ώστε

$$\begin{aligned} &w(f^0 \oplus g) + w(f^1 \oplus g) + w(g) \text{ ή} \\ &w(f^0 \oplus g) + w(f^2 \oplus g) + w(g) \text{ ή} \\ &w(f^1 \oplus g) + w(f^2 \oplus g) + w(g) \end{aligned}$$

να είναι το ελάχιστο για όλες τις συναρτήσεις g , και, επομένως, μπορούμε να βρούμε μια ακριβή έκφραση ESOP για την f . Οι διάφορες προσεγγίσεις στη βιβλιογραφία διαφοροποιούνται στις τεχνικές που χρησιμοποιούνται ώστε να μειωθεί ο υπολογιστικός χρόνος, ο οποίος είναι απαγορευτικός.

Το κίνητρό μας ήταν να χρησιμοποιήσουμε την ίδια προσέγγιση που περιγράφηκε προηγουμένως, λαμβάνοντας υπ' όψιν ότι τώρα εξετάζουμε μη πλήρως ορισμένες συναρτήσεις. Οι ακόλουθες τεχνικές χρησιμοποιούνται ώστε να μειώσουμε τον υπολογιστικό χρόνο, λαμβάνοντας υπ' όψιν ότι ασχολούμαστε με μη πλήρως ορισμένες συναρτήσεις.

1. Η αποθήκευση σε ένα συμπιεσμένο πίνακα RT , που απαιτεί τον ελάχιστο χώρο μνήμης, του βάρους όλων των συναρτήσεων 5 μεταβλητών.
2. Η δημιουργία ενός άλλου πίνακα RT_{DC} (συμπιεσμένος πίνακας για ένα δεδομένο σύνολο αδιάφορων όρων DC).

Algorithm 5: Create the table RT for all functions $f \in G^5$

```

begin
  /*Initialize the entire table with zeros*/
  for  $f = 1$  to  $2^{32} - 1$  do
    |  $RT(f) = 0$ ;
  end for
  for all product terms  $p$  (total number of  $3^5 - 1$  product terms) do
    |  $RT(p) = 1$ ;
  end for
  /*The following loop gradually allocates the weight for each function.
  Initially only product terms have weight 1.*/
  for  $i = 2$  to  $8$  do
    | for each  $f$  of table  $RT$ , for which  $f \neq 0$  and  $RT(f) = i - 1$  do
      | | for each product term  $p$  do
          | | | /*The following XOR-sum corresponds to a function. If
          | | | this function has a weight entry 0 it means that it is not a
          | | | product term and thus has weight equal to  $i$  or 1 for the
          | | | case of 8.*/
          | | | if  $RT(f \oplus p) = 0$  then
          | | | |  $RT(f \oplus p) = (i < 8 ? i : 1)$ ; exit;
          | | | end if
          | | end for
        | end for
      end for
    end for
  end
end

```

Algorithm 6: Create table RT_{DC}

```

begin
  set each element of  $RT_{DC}$  to the value 9;
  for each function  $f \in G^5$  do
    | if  $RT_{DC}(red(f)) > RT(f)$  then
    | |  $RT_{DC}(red(f)) = RT(f)$ ; exit;
    | end if
  end for
end

```

4.2 Αποθήκευση σε πίνακα των βαρών όλων των συναρτήσεων με 5 μεταβλητές εισόδου

Ο αριθμός των Boolean συναρτήσεων 5 μεταβλητών είναι $2^{2^5} = 4294967296$ και το βάρος τους είναι $w \leq 9$ [21]. Επομένως, απαιτούνται 4 Gbytes μνήμης για να αποθηκεύσουμε έναν πίνακα T που περιέχει τα βάρη τους. Προσπαθώντας να μειώσουμε τον απαιτούμενο χώρο, δημιουργήσαμε έναν πίνακα RT , ο οποίος χρησιμοποιεί μόνο 3 bits για το βάρος κάθε συνάρτησης. Άρα, ο συνολικός απαιτούμενος χώρος για τον RT είναι $4294967296 \times 3/8 = 1610612736$ bytes. Το πρόβλημα είναι ότι πρέπει να αναπαραστήσουμε τα βάρη w με $0 \leq w \leq 9$ χρησιμοποιώντας μόνο 3 bits. Προφανώς, το βάρος $w = 0$ είναι αληθές μόνο για τη συνάρτηση $f = 0$. Επομένως, μπορούμε να χρησιμοποιήσουμε την τιμή 0 για να δηλώσουμε το βάρος 9. Επιπλέον, αν αποφασίσουμε να αποθηκεύσουμε τα βάρη 1 και 8 ως την τιμή 1, τότε χρειάζεται να αναπαραστήσουμε ακριβώς 8 διαφορετικές τιμές, κάτι που μπορεί να γίνει με 3 bits. Το μόνο ζήτημα που απομένει είναι να διαφοροποιηθούμε ανάμεσα στα βάρη 1 και 8. Αυτό μπορεί να επιτευχθεί εύκολα, αφού μπορούμε να χρησιμοποιήσουμε τον Αλγόριθμο 7 για να βρούμε αν μια συνάρτηση είναι ένα γινόμενο, που σημαίνει ότι έχει βάρος $w = 1$.

Algorithm 7: Find if a function f has weight $w = 1$ or 8

```
begin
  g=f; w=1; for i=1 to 4 do
    if  $g^1 = 0$  then
      |  $g = g^0$ ;
    end if
    else if  $g^0 = 0$  then
      |  $g = g^1$ ;
    end if
    else if  $g^1 = g^0$  then
      |  $g = g^1$ ;
    end if
    else
      |  $w = 8$ ; exit;
    end if
  exit;
end for
end
exit;
```

Για να δημιουργήσουμε τον πίνακα RT , πρέπει να εκτελέσουμε τον Αλγόριθμο 5.

Τέλος, για να βρούμε το βάρος w μιας συνάρτησης, χρησιμοποιώντας τον πίνακα RT , πρέπει να χρησιμοποιήσουμε τον Αλγόριθμο 8.

Algorithm 8: Find the weight of function f (for $n \leq 5$)

```
begin
  if  $f = 0$  then
    |  $w = 0$ ; exit;
  end if
   $w = RT(f)$ ;
  if  $w = 0$  then
    |  $w = 9$ ; exit;
  end if
  if  $w = 1$  then
    | execute algorithm 3 to find if  $w$  is 1 or 8
  end if
end
```

Με βάση όσα γνωρίζουμε, αυτή είναι η πρώτη προσπάθεια να αποθηκεύσουμε σε πίνακα το βάρος όλων των συναρτήσεων 5 μεταβλητών. Μέχρι τώρα, οι αλγόριθμοι ESOP ελαχιστοποίησης χρησιμοποιούν πίνακες για τα βάρη όλων των συναρτήσεων 4 μεταβλητών [28] ή χρησιμοποιούν τις κλάσεις ισοδυναμίας των συναρτήσεων 5 μεταβλητών [21]. Στην πρώτη περίπτωση, ένα περισσότερο επίπεδο αναδρομής ή επανάληψης πρέπει να εκτελεστεί και στη δεύτερη ένας αλγόριθμος πρέπει να εκτελεστεί για να βρει την κλάση ισοδυναμίας της δεδομένης συνάρτησης 5 μεταβλητών. Επομένως, και οι δύο προσεγγίσεις καταναλώνουν περισσότερο χρόνο από αυτή που προτείνουμε εδώ. Αφού η εύρεση του βάρους μιας συνάρτησης 4 ή 5 μεταβλητών γίνεται επαναληπτικά στους γνωστούς αλγόριθμους ελαχιστοποίησης [21, 28, 27], είναι φανερό ότι ακόμη και μια μικρή μείωση στο χρόνο εύρεσης του βάρους, θα μειώσει δραστικά το συνολικό χρόνο εκτέλεσης του αλγορίθμου. Η προτεινόμενη μέθοδός μας αποθήκευσης σε πίνακα είναι πρακτικά μια άμεση πρόσβαση σε look-up table (LUT) και μπορεί εύκολα να χρησιμοποιηθεί σε όλους τους άλλους γνωστούς αλγόριθμους ελαχιστοποίησης για να τους κάνει γρηγορότερους.

4.3 Εύρεση μιας ακριβούς έκφρασης

Ας θεωρήσουμε μια μη πλήρως ορισμένη συνάρτηση n μεταβλητών με δεδομένα ON-set, OFF-set και DC-set.

Ορισμός 29 Ορίζεται ως F το σύνολο όλων των δυνατών πλήρως ορισμένων συναρτήσεων $f \in G^n$, αν αναθέσουμε όλους τους δυνατούς συνδυασμούς τιμών 0 ή 1 στους ελαχιστόρους του DC-set μιας μη πλήρως ορισμένης συνάρτησης, διατηρώντας, στην αναπαράσταση ελαχιστόρων της f , όλα τα άλλα bits όπως αυτά της μη πλήρως ορισμένης συνάρτησης.

Ορισμός 30 Μια ακριβής έκφραση μιας μη πλήρως ορισμένης συνάρτησης, ορίζεται ως μια ακριβής έκφραση μιας συνάρτησης $\phi \in F$, τέτοιας ώστε $w(\phi) \leq w(f)$ για όλες τις $f \in F$.

Λήμμα 1 Αν $h \in G^n$ είναι μια δεδομένη συνάρτηση, τότε $red(f \oplus h)$, $rep(f \oplus h)$ και $RT_{DC}(red(f \oplus h))$ είναι τα ίδια για οποιαδήποτε $f \in F$ και θεωρώντας ως σύνολο DC το δεδομένο DC-set της μη πλήρως ορισμένης συνάρτησης.

Απόδειξη

Το Λήμμα 1 είναι μια συνέπεια των ορισμών 26, 27, 28.

Έστω μια έκφραση ESOP μιας συνάρτησης $f \in F$:

$$f = x_1(f^1 \oplus h) \oplus \bar{x}_1(f^0 \oplus h) \oplus h = x_1 f^1 \oplus \bar{x}_1 f^0 \quad (4.2)$$

όπου $f^1 \oplus h$, $f^0 \oplus h$, h είναι συναρτήσεις $n - 1$ μεταβλητών. Για να βρούμε την f με το ελάχιστο βάρος (δηλαδή μια ακριβή έκφραση της μη πλήρως ορισμένης συνάρτησης), πρέπει να θεωρήσουμε όλες τις δυνατές f^1 , f^0 και h για όλες τις συναρτήσεις $f \in F$ και $h \in G^{n-1}$, να υπολογίσουμε τα βάρη $w(f^1 \oplus h)$, $w(f^0 \oplus h)$, $w(h)$ και να βρούμε αυτές τις f^1 , f^0 , h για τις οποίες το

$$w(f^1 \oplus h) + w(f^0 \oplus h) + w(h) \quad (4.3)$$

να είναι ελάχιστο.

Για μια δεδομένη h και ένα δεδομένο σύνολο DC,

$$w(f^1 \oplus h) \geq RT_{DC^1}(red(f^1 \oplus h)), w(f^0 \oplus h) \geq RT_{DC^0}(red(f^0 \oplus h)) \quad (4.4)$$

σύμφωνα με τον Ορισμό 27, για κάθε συνάρτηση $f = (x_1 f^1 \oplus x_0 f^0) \in F$.

Στις Σχέσεις 4.4, το $DC^1 - set$ και το $DC^0 - set$ προκύπτουν από το $DC - set$ ως εξής: Το $DC^1 - set$ περιέχει έναν ελαχιστόρο $x_2^* \cdots x_n^*$ για κάθε ελαχιστόρο $x_1 x_2^* \cdots x_n^*$ του $DC - set$ με $x_i^* = x_i, \bar{x}_i$ για $2 \leq i \leq n$. Το $DC^0 - set$ περιέχει έναν ελαχιστόρο $x_2^* \cdots x_n^*$ για κάθε ελαχιστόρο $\bar{x}_1 x_2^* \cdots x_n^*$ του $DC - set$ με $x_i^* = x_i, \bar{x}_i$ για $2 \leq i \leq n$. Οι $RT_{DC^1}(red(f^1 \oplus h))$ και $RT_{DC^0}(red(f^0 \oplus h))$ είναι οι ίδιοι για όλες τις συναρτήσεις $f \in F$ και h είναι μια δεδομένη συνάρτηση τέτοια ώστε $h \in G^{n-1}$ (σύμφωνα με το Λήμμα 1). Άρα, το ελάχιστο της Εξίσωσης 4.3 είναι το ελάχιστο της

$$RT_{DC^1}(red(f^1 \oplus h)) + RT_{DC^0}(red(f^0 \oplus h)) + RT(h) \quad (4.5)$$

Αφού, οι $RT_{DC^1}(red(f^1 \oplus h))$ και $RT_{DC^0}(red(f^0 \oplus h))$ είναι οι ίδιοι για όλες τις συναρτήσεις $f \in F$ και δεδομένη h , μπορούμε να επιλέξουμε η f να είναι μια συνάρτηση με μηδενικά, στην MT αναπαράσταση της, για όλους τους ελαχιστόρους του αντίστοιχου συνόλου DC, δηλαδή η συνάρτηση που οι ελαχιστόροι της είναι αυτοί του ON-set, η οποία είναι η συνάρτηση f_{ON} . Μπορούμε να θεωρήσουμε όλες τις δυνατές συναρτήσεις $h \in G^{n-1}$ και να βρούμε ποια (ποιες) ελαχιστοποιεί (ελαχιστοποιούν) την έκφραση

$$RT_{DC^1}(red(f_{ON}^1 \oplus h)) + RT_{DC^0}(red(f_{ON}^0 \oplus h)) + RT(h) \quad (4.6)$$

(Αλγόριθμος 9).

Μια ακριβής έκφραση της αρχικής μη πλήρως ορισμένης συνάρτησης (δεδομένων των ON-set, OFF-set και DC-set της), είναι:

$$x_1 \text{rep}(\text{red}(f_{ON}^1 \oplus h)) \oplus \bar{x}_1 \text{rep}(\text{red}(f_{ON}^0 \oplus h)) \oplus h \quad (4.7)$$

σύμφωνα με τον Ορισμό 28.

Οι αντιπροσωπευτικές συναρτήσεις για ένα δεδομένο σύνολο DC , μπορούν να αποθηκευτούν κατά την παραγωγή του πίνακα RT_{DC} .

Παράδειγμα 25 Έστω f η συνάρτηση του Παραδείγματος 21. Μια ακριβής έκφραση της f είναι η $\phi = \bar{x}_1 x_2 x_3 \oplus \bar{x}_3$. Ας δούμε πώς μπορούμε να τη βρούμε χρησιμοποιώντας την προηγούμενη θεωρία.

$$\phi = x_1 \phi^1 \oplus \bar{x}_1 \phi^0 \oplus h$$

Έχουμε:

$$f_{ON}^1 = [0100] = [4]$$

$$f_{ON}^0 = [1101] = [d]$$

$$DC^1 - \text{set} = \{\bar{x}_2 \bar{x}_3, x_2 x_3\} = [1001] = [9]$$

$$DC^0 - \text{set} = \emptyset = [0000] = [0]$$

Εξετάζοντας εξαντλητικά όλες τις συναρτήσεις h 2 μεταβλητών, θα εξετάσουμε τη συνάρτηση $h = \bar{x}_3 = [0101] = [5]$. Για $h = \bar{x}_3$:

$$\begin{aligned} w(\phi^1) &= RT_{DC^1}(\text{red}(f_{ON}^1 \oplus h)) = RT_{DC^1}(\text{red}([0100] \oplus [0101])) = \\ &= RT_{DC^1} \text{red}([0001]) = RT_{DC^1}([00]) = 0 \end{aligned}$$

(όπου αφαιρούμε τα bits 0, 3 επειδή αντιστοιχούν στους αδιάφορους ελαχιστόρους) σύμφωνα με τους Πίνακες 4.3 και 4.4. Αυτοί οι πίνακες είναι οι αντίστοιχοι RT και RT_{DC} για τη συνάρτηση του παραδείγματος.

$$\phi^1 = \text{rep}(\text{red}(f_{ON}^1 \oplus h)) = \text{rep}(00) = 0$$

Κατά τον ίδιο τρόπο,

$$\begin{aligned} w(\phi^0) &= RT_{DC^0}(\text{red}(f_{ON}^0 \oplus h)) = \\ &= RT_{DC^0}(\text{red}([1101] \oplus [0101])) = RT_{DC^0}(\text{red}([1000])) = \\ &= RT([1000]) = 1 \end{aligned}$$

και

$$\phi^0 = \text{rep}(\text{red}(f_{ON}^0 \oplus h)) = \text{rep}([1000]) = x_2 x_3$$

. Επομένως,

$$w(f) = w(\phi^1) + w(\phi^0) + w(h) = 0 + 1 + 1 = 2$$

f	w	έκφραση
0000	0	0
0001	1	$\bar{x}_2\bar{x}_3$
0010	1	\bar{x}_2x_3
0011	1	\bar{x}_2
0100	1	$x_2\bar{x}_3$
0101	1	\bar{x}_3
0110	2	$x_2\bar{x}_3 \oplus \bar{x}_2x_3$
0111	2	$x_2\bar{x}_3 \oplus \bar{x}_2$
1000	1	x_2x_3
1001	2	$x_2x_3 \oplus \bar{x}_2\bar{x}_3$
1010	1	x_3
1011	2	$x_2x_3 \oplus \bar{x}_2$
1100	1	x_2
1101	2	$x_2 \oplus \bar{x}_2\bar{x}_3$
1110	2	$x_2 \oplus \bar{x}_2x_3$
1111	1	1

Πίνακας 4.3: Πίνακας RT για το Παράδειγμα 25

$red(f)$	w	$rep(f)$
00	0	0
01	1	$\bar{x}_2x_3, \bar{x}_2, x_3$
10	1	$x_2\bar{x}_3, \bar{x}_3, x_2$
11	1	1

Πίνακας 4.4: Πίνακας RT_{DC}^1 για το Παράδειγμα 25

και μια ακριβής έκφραση είναι η

$$f = x_1 \cdot 0 \oplus \bar{x}_1x_2x_3 \oplus \bar{x}_3$$

Ο Αλγόριθμος 6 δημιουργεί τον πίνακα RT_{DC} για ένα δεδομένο σύνολο DC θεωρώντας όλες τις συναρτήσεις 5 μεταβλητών. Χρησιμοποιώντας τον πίνακα RT_{DC} , μπορούμε να χρησιμοποιήσουμε οποιονδήποτε αλγόριθμο ελαχιστοποίησης στη βιβλιογραφία (για πλήρως ορισμένες συναρτήσεις), για να βρούμε την ακριβή έκφραση ESOP μιας μη πλήρως ορισμένης συνάρτησης 6 μεταβλητών. Υπάρχουν ορισμένοι περιορισμοί που πρέπει να λάβουμε υπ' όψιν μας. Ο ένας είναι να χρησιμοποιήσουμε μόνο το ανάπτυγμα Shannon και όχι τα αναπτύγματα Davio σε αυτούς τους αλγόριθμους. Αυτό γίνεται επειδή αν και τα DC-set για την f^1 και την f^0 είναι ανεξάρτητα, το DC-set για την f^2 εξαρτάται από αυτά για την f^1 και την f^0 , κάτι το οποίο περιπλέκει τον αλγόριθμο για να καταγράψουμε για ποιους ελαχιστόρους έχει ήδη επιλεγεί η τιμή τους σε κάθε βήμα. Σαν παράδειγμα θεωρήστε τον πρώτο αλγόριθμο της αναφοράς [21] ή τον αλγόριθμο min-esop της αναφοράς [28] που είναι παρόμοιοι. Μπορούν να μετασχηματιστούν για μη πλήρως ορισμένες συναρ-

τήσεις 6 μεταβλητών, όπως υποδεικνύεται στον Αλγόριθμο 9, όπου RT_{DC^1} είναι ο συμπιεσμένος πίνακας της $f^1 \in G^{n-1}$ με το $DC^1 - set$ ως σύνολο DC και ο RT_{DC^0} είναι ο συμπιεσμένος πίνακας της $f^0 \in G^{n-1}$ με το $DC^0 - set$ ως σύνολο DC .

Algorithm 9: Find an exact expression of an incompletely specified function f

```

begin
  w = 15; for each h ∈ G5 do
    | g1 = RTDC1(red(fON1 ⊕ h)) ; g0 = RTDC0(red(fON0 ⊕ h)) ;
    | g2 = RT(h); g = g1 + g0 + g2; if g is less than w then
    | | w = g;
    | end if
  end for
end

```

Σημειώνεται ότι ο Αλγόριθμος 9 είναι εξαντλητικός και εγγυάται ότι θα βρεθεί μια ακριβής έκφραση.

Εν τούτοις, η μέθοδός μας μπορεί να συνδυαστεί με άλλους γνωστούς αλγόριθμους της βιβλιογραφίας για ακριβή ελαχιστοποίηση, όπως ο *min_tau(f)* της αναφοράς [28]. Σε αυτήν την περίπτωση πρέπει να χρησιμοποιήσουμε μόνο το ανάπτυγμα Shannon (και όχι τα Davio). Αυτό, καθώς επίσης και το γεγονός ότι δεν έχουμε αδιάφορους ελαχιστόρους, έχει ως συνέπεια ότι τα Λήμματα 4 και 5 της αναφοράς [28] δεν είναι έγκυρα, και επομένως τα άνω όρια (δείτε την Ενότητα 4 της αναφοράς [28]) δεν είναι έγκυρα. Επιλέξαμε να χρησιμοποιήσουμε ένα χαλαρότερο άνω όριο, δηλαδή το $\kappa(f) = \text{minimum}(9, s - 2)$, όπου το 9 είναι το μέγιστο βάρος οποιασδήποτε συνάρτησης 5 μεταβλητών και s είναι το τρέχον βάρος της συνάρτησης 6 μεταβλητών που πρόκειται να ελαχιστοποιηθεί. Επιπλέον, σταματάμε την αναδρομή στις συναρτήσεις 5 μεταβλητών, όπου χρησιμοποιούμε τους RT_{DC_0} και RT_{DC_1} για τα βάρη της $f^1 \oplus h$ και της $f^0 \oplus h$ αντίστοιχα και τον RT για το βάρος της h (Έκφραση 4.5). Αυτός ο τροποποιημένος αλγόριθμος θα λέγεται Αλγόριθμος 10.

4.4 Υλοποίηση

Η προτεινόμενη μέθοδος υλοποιήθηκε χρησιμοποιώντας έναν Intel Core 2 Duo T7500 CPU @ 2.20 GHz και 2 GB RAM.

Ο Αλγόριθμος 5 απαιτήσε περίπου 24 ώρες για να δημιουργήσει τον πίνακα RT , αλλά αυτό έγινε μόνο μια φορά. Ο Αλγόριθμος 6 απαιτήσε περίπου 2 ώρες, για να δημιουργήσει και τον πίνακα RT_{DC^1} και τον πίνακα RT_{DC^0} για ένα δεδομένο $DC - set$. Αφού έχουμε δημιουργήσει αυτούς τους πίνακες, μετά μπορούμε να τους χρησιμοποιήσουμε για κάθε συνάρτηση 6 μεταβλητών εισόδου με το ίδιο $DC - set$. Ο Αλγόριθμος 9 απαιτεί περίπου 130 λεπτά για να βρει μια ακριβή έκφραση μιας μη πλήρως ορισμένης συνάρτησης 6 μεταβλητών εισόδου, ανεξάρτητα

Algorithm 10: Find an exact expression of an incompletely specified function modifying $\mathit{min_tau}(f)$

begin

It is the $\mathit{min_tau}(f)$ algorithm [28] with the following modifications:

1. Use only the Shannon expansion
2. Take as upper bound $\kappa(f) = \mathit{minimum}(9, s - 2)$
3. Stop recursion to 5-variable functions
4. Use the RT_{DC^1} , RT_{DC^0} for the weights of $f^1 \oplus h$ and $f^0 \oplus h$ respectively and RT for the weight of h (Expression 4.5)

end

από τον αριθμό των αδιάφορων όρων ή το βάρος της συνάρτησης που πρόκειται να ελαχιστοποιηθεί. Εν τούτοις, η μέθοδός μας μπορεί να συνδυαστεί με άλλους γνωστούς αλγόριθμους της βιβλιογραφίας, όπως ο $\mathit{min_tau}(f)$ της αναφοράς [28], όπως εξηγήθηκε προηγουμένως (Αλγόριθμος 10).

Καθώς, με βάση όσα γνωρίζουμε, δεν υπάρχουν δημοσιευμένα αποτελέσματα για ακριβείς λύσεις μη πλήρως ορισμένων συναρτήσεων 6 μεταβλητών, μπορούμε να χρησιμοποιήσουμε τυχαία παραγόμενες μη πλήρως ορισμένες συναρτήσεις, ώστε να δοκιμάσουμε την μέθοδό μας. Επομένως, δημιουργήσαμε 100 τυχαία παραγόμενες συναρτήσεις 6 μεταβλητών, με βάρη ανάμεσα σε 6 και 10, όπως φαίνεται στον Πίνακα 4.5. Εκτελέσαμε 4 είδη πειραμάτων, που επεξηγούνται περισσότερο στις επόμενες ενότητες.

Βάρος	Ποσοστό
6	4
7	22
8	44
9	28
10	2
	100

Πίνακας 4.5: Βάρη των τυχαία παραγόμενων συναρτήσεων

4.4.1 Σύγκριση με ακριβείς μεθόδους για πλήρως ορισμένες συναρτήσεις

Το πρώτο πείραμα ήταν να ελέγξουμε την αποτελεσματικότητα όσον αφορά τον υπολογιστικό χρόνο, καθώς και την ποιότητα των αποτελεσμάτων της προσέγγισής μας, όταν συγκρίνεται με ακριβείς μεθόδους για πλήρως ορισμένες συναρτήσεις. Για αυτόν το σκοπό χρησιμοποιήσαμε τον Αλγόριθμο $\mathit{min_tau}(f)$ [28], καθώς

και τους Αλγόριθμους 9, 10 για τρία διαφορετικά $DC - set$, το $DC_1 - set$ με 12 ελαχιστόρους, το $DC_2 - set$ με 16 ελαχιστόρους και το $DC_3 - set$ με 18 ελαχιστόρους. Στον Πίνακα 4.6 μπορούμε να δούμε τα αποτελέσματα. Το σύνολο των τυχαίων συναρτήσεων που χρησιμοποιήσαμε είχε ένα μέσο βάρος ίσο με 8.02.

	Αλγόριθμος 9	Αλγόριθμος 10	Αλγόριθμος $min_tau(f)$	Αλγόριθμος $QuickDCMIN$
Μέσο βάρος των συναρτήσεων χρησιμοποιώντας το $DC_1 - set$	7.2	7.2	8.02	7.62
Μέσο βάρος των συναρτήσεων χρησιμοποιώντας το $DC_2 - set$	7.00	7.00	8.02	7.58
Μέσο βάρος των συναρτήσεων χρησιμοποιώντας το $DC_3 - set$	6.82	6.82	8.02	7.49
Μέσος υπολογιστικός χρόνος σε secs χρησιμοποιώντας το $DC_1 - set$	7800 (+7200)	16.43 (+7200)	0.017	2.00
Μέσος υπολογιστικός χρόνος σε secs χρησιμοποιώντας το $DC_2 - set$	7800 (7200)	15.11 (+7200)	0.017	2.00
Μέσος υπολογιστικός χρόνος σε secs χρησιμοποιώντας το $DC_3 - set$	7800 (7200)	9.70 (+7200)	0.017	2.00

Πίνακας 4.6: Πειραματικά αποτελέσματα συγκρίνοντας τους Αλγόριθμους 9 και 10 με τους αλγόριθμους $min_tau(f)$ και $QuickDCMIN$

Όπως μπορούμε να δούμε από τον Πίνακα 4.6:

1. Αναφορικά με την ποιότητα των αποτελεσμάτων, οι Αλγόριθμοι 9 και 10 έχουν καλύτερη επίδοση από τον $min_tau(f)$, όπως αναμενόταν, επειδή ο $min_tau(f)$ δεν παίρνει υπ' όψιν του τους αδιάφορους ελαχιστόρους.
2. Αναφορικά με τον υπολογιστικό χρόνο, ο $min_tau(f)$ δίνει καλύτερους χρόνους εκτέλεσης. Παρ' όλα αυτά, πρέπει να σημειωθεί ότι οι Αλγόριθμοι 9 και 10 απαιτούν περίπου 2 ώρες για να παράξουν τους πίνακες RT_{DC} , όπως βλέπουμε στον Πίνακα 4.6. Για την ελαχιστοποίηση πολλών συναρτήσεων που έχουν το ίδιο DC-set, αυτή η παραγωγή πρέπει να γίνει μια φορά.

4.4.2 Σύγκριση με ευριστικές μεθόδους για μη πλήρως ορισμένες συναρτήσεις

Το δεύτερο πείραμα ήταν να συγκρίνουμε την προσέγγισή μας, σε υπολογιστικό χρόνο, καθώς και στην ποιότητα των αποτελεσμάτων, με ευριστικές μεθόδους για μη πλήρως ορισμένες συναρτήσεις. Για αυτό χρησιμοποιήσαμε τον αλγόριθμο $QuickDCMIN$. Στον Πίνακα 4.6 μπορούμε να δούμε τα αποτελέσματα. Όπως μπορούμε να δούμε:

1. Αναφορικά με την ποιότητα των αποτελεσμάτων, οι Αλγόριθμοι 9 και 10 έχουν καλύτερη επίδοση από τον αλγόριθμο $QuickDCMIN$.
2. Αναφορικά με τον υπολογιστικό χρόνο, οι Αλγόριθμοι 9 και 10 είναι χειρότεροι από τον αλγόριθμο $QuickDCMIN$.

4.4.3 Σύγκριση με εξαντλητικές μεθόδους

Το τρίτο πείραμα ήταν να συγκρίνουμε την προσέγγισή μας με εξαντλητικές μεθόδους, π.χ. με μεθόδους για την εύρεση ακριβών λύσεων για πλήρως ορισμένες συναρτήσεις, λαμβάνοντας επίσης υπ' όψιν όλους τους δυνατούς συνδυασμούς των ελαχιστόρων του DC-set. Αν το DC-set έχει m ελαχιστόρους, τότε πρέπει να τρέξουμε το πρόγραμμα για ακριβείς λύσεις πλήρων ορισμένων συναρτήσεων 2^m φορές, για όλους τους δυνατούς συνδυασμούς αδιάφορων όρων. Για αυτόν το σκοπό, χρησιμοποιήσαμε εξαντλητικά τον $\min_tau(f)$ 2^m φορές. Θα καλούμε αυτόν τον αλγόριθμο 'Εξαντλητικό Αλγόριθμο'. Σημειώνεται ότι εξαιτίας υπολογιστικών χρονικών περιορισμών, χρησιμοποιήσαμε σε αυτήν την περίπτωση ένα σύνολο από 10 τυχαία παραγόμενες συναρτήσεις. Στον Πίνακα 4.7 μπορούμε να δούμε ότι:

1. Αναφορικά με την ποιότητα των αποτελεσμάτων, όλοι οι αλγόριθμοι δίνουν τα ίδια αποτελέσματα, αφού δίνουν ακριβείς λύσεις (που δεν συμπεριλαμβάνονται στον Πίνακα 4.7).
2. Αναφορικά με τον υπολογιστικό χρόνο, ο Εξαντλητικός Αλγόριθμος απαιτεί μη αποδεκτούς χρόνους εκτέλεσης για έναν μεγάλο αριθμό από αδιάφορους ελαχιστόρους στο DC-set (μεγαλύτερο από ή ίσο με 16). Πρακτικά, ο υπολογιστικός χρόνος, για τον Εξαντλητικό Αλγόριθμο, αυξάνεται εκθετικά με την αύξηση του αριθμού των ελαχιστόρων στο DC-set. Παρατηρήσαμε επίσης ότι ο υπολογιστικός χρόνος, επίσης για τον Εξαντλητικό Αλγόριθμο, αυξάνεται σχεδόν γραμμικά με το βάρος της συνάρτησης, για το ίδιο DC-set. Επομένως, ακόμα και για μικρά βάρη των συναρτήσεων, ο κυριαρχών παράγοντας είναι ο αριθμός των αδιάφορων ελαχιστόρων. Για αυτόν τον λόγο, επιλέξαμε τα βάρη των παραγόμενων συναρτήσεων να μην είναι μεγάλα, προκειμένου να έχουμε μια δίκαια σύγκριση του Εξαντλητικού Αλγορίθμου με τους Αλγόριθμους 9 και 10. Ο Αλγόριθμος 9 απαιτεί σταθερό υπολογιστικό χρόνο, ανεξάρτητα από τα βάρη των συναρτήσεων που πρόκειται να ελαχιστοποιηθούν ή τον αριθμό των αδιάφορων ελαχιστόρων. Ο Αλγόριθμος 10 απαιτεί υπολογιστικό χρόνο που δεν εξαρτάται από τα βάρη των συναρτήσεων που πρόκειται να ελαχιστοποιηθούν ή τον αριθμό των αδιάφορων ελαχιστόρων, όπως μπορούμε να δούμε από τους Πίνακες 4.6 και 4.7. Όπως φαίνεται στον Πίνακα 4.7, οι Αλγόριθμοι 9 και 10 έχουν καλύτερη απόδοση από τον Εξαντλητικό Αλγόριθμο, ο οποίος δεν μπορεί να εφαρμοστεί πρακτικά για έναν μεγάλο αριθμό αδιάφορων ελαχιστόρων.

Εικάζεται ότι το άνω όριο είναι χαμηλότερο από αυτό που χρησιμοποιήσαμε, δηλαδή το $\kappa(f) = \text{minimum}(9, s - 2)$, κάτι το οποίο θα οδηγήσει σε δραστική μείωση του υπολογιστικού χρόνου. Χρησιμοποιώντας εμπειρικά ως άνω όριο το $5 \leq \kappa(f) = s - 1 - g \leq 9$ με $g = \text{maximum}(w(f^1), w(f^0))$, πήραμε τα ίδια πειραματικά αποτελέσματα, αλλά με μια επιτάχυνση τουλάχιστον μιας τάξης μεγέθους στον υπολογιστικό χρόνο. Η επιβάρυνση για να δημιουργήσουμε τους πίνακες RT_{DC} δεν υπολογίζεται σε αυτήν τη σύγκριση.

	Αλγόριθμος 9	Αλγόριθμος 10	Εξαντλητικός Αλγόριθμος
Μέσος υπολογιστικός χρόνος σε secs χρησιμοποιώντας το $DC_1 - set$	7800 (+7200)	11.02 (+7200)	1137.80
Μέσος υπολογιστικός χρόνος σε secs χρησιμοποιώντας το $DC_2 - set$	7800 (+7200)	17.146 (+7200)	25122.20
Μέσος υπολογιστικός χρόνος σε secs χρησιμοποιώντας το $DC_3 - set$	7800 (+7200)	7.41 (+7200)	98632.47

Πίνακας 4.7: Πειραματικά αποτελέσματα συγκρίνοντας τους Αλγόριθμους 9 και 10 με τον Εξαντλητικό Αλγόριθμο

4.4.4 Χρήση του Αλγορίθμου 10 για περισσότερες από 6 μεταβλητές

Σαν ένα τελικό πείραμα χρησιμοποιήσαμε τον Αλγόριθμο 10 για συναρτήσεις με περισσότερες από 6 μεταβλητές εισόδου, αποσυνθέτοντάς τις χρησιμοποιώντας το ανάπτυγμα Shannon (αν και θα μπορούσαμε να χρησιμοποιήσουμε τα αναπτύγματα Davio επίσης), μέχρι το επίπεδο των 6 μεταβλητών. Χρησιμοποιήσαμε τον Αλγόριθμο 10 για να υπολογίσουμε τις ακριβείς λύσεις σε αυτό το επίπεδο και να εκτιμήσουμε το βάρος σε κάθε κόμβο του δένδρου, προσθέτοντας τα βάρη των υποσυναρτήσεων του κόμβου. Για αυτόν το σκοπό, χρησιμοποιήσαμε την μετροσυνάρτηση bench [32, 81] με 6 εισόδους και 8 εξόδους. Η χαρακτηριστική της συνάρτηση, καθώς και το αντίστοιχο DC-set δίνονται παρακάτω.

$$f_{ON} = [a004000800c000800000204400500000a801c000000001009000400000020020000300001005000000000000000e430004000800046000140000400808a00400]$$

$$f_{DC} = [43c321e5dd33bf6ffffbdd397f2f776f07fa3fad7d3f68f76d5ebabdfdf4fbd3af7cafb7ebfaf7ff4975ebcf6ff1bcfb e2fed5bff90aceaaabfa2d77f74bf3eb]$$

Το βάρος που βρέθηκε με τον Αλγόριθμο 10 και την αποσύνθεση ήταν 24. Χρειάστηκε περίπου $8 \cdot 2 = 16$ ώρες για να δημιουργήσουμε τους πίνακες DC στα φύλλα του γεννήτορος δένδρου. Οι Αλγόριθμοι QuickDCMIN [32] και EXORSISM-MV-2 [81] δίνουν βάρη 18 και 24 αντίστοιχα. Επομένως ο Αλγόριθμος 10 έδωσε χειρότερα αποτελέσματα από τον QuickDCMIN, αλλά τα ίδια αποτελέσματα με τον EXORSISM-MV-2 για αυτήν τη συνάρτηση.

Κεφάλαιο 5

Εκφράσεις ESCT

5.1 Σύνθετοι Όροι και εκφράσεις ESCT

Ορισμός 31 Έστω $f : \{0, 1\}^n \rightarrow \{0, 1\}^k$ μια *switching* συνάρτηση n μεταβλητών με k εξόδους και $\psi : \{0, 1\}^n \times \{0, \dots, k-1\} \rightarrow \{0, 1\}$ η χαρακτηριστική της συνάρτηση. Έστω $x_i, i = 1, \dots, n$ οι δυαδικές μεταβλητές και $Y \in \{0, \dots, k-1\}$ η μεταβλητή πολλαπλών τιμών της ψ , Y^S ($S \subseteq \{0, \dots, k-1\}$) ένα *literal* της Y και $R_i, i = 1, \dots, n$ τυχαίες λογικές συναρτήσεις δύο εισόδων και μιας εξόδου. Τότε $U = R_n(x_n, R_{n-1}(x_{n-1}, R_{n-2}(x_{n-2}, \dots, R_1(x_1, Y^S) \dots)))$ ορίζεται ως ένας σύνθετος όρος (*complex term*) ή όρος *Maitra* (*Maitra term*), ο οποίος εξαρτάται από τις μεταβλητές Y, x_1, \dots, x_n . Οι συναρτήσεις R_i ονομάζονται *ESCT* κυτταρικές συναρτήσεις (*ESCT cell functions*) του όρου.

Προκειμένου για boolean συναρτήσεις μιας εξόδου, ο ορισμός των όρων *Maitra* τροποποιείται ως εξής:

Ορισμός 32 Έστω $f : \{0, 1\}^n \rightarrow \{0, 1\}$ μια συνάρτηση n μεταβλητών με 1 έξοδο, που εξαρτάται από τις δυαδικές μεταβλητές $x_i, i = 1, \dots, n$, y μια δυαδική τιμή (σταθερή είσοδος) και $R_i, i = 1, \dots, n$ τυχαίες λογικές συναρτήσεις δύο εισόδων και μιας εξόδου. Τότε $U = R_n(x_n, R_{n-1}(x_{n-1}, R_{n-2}(x_{n-2}, \dots, R_1(x_1, y) \dots)))$ ορίζεται ως ένας σύνθετος όρος ή όρος *Maitra* που εξαρτάται από τις μεταβλητές x_1, \dots, x_n . Οι συναρτήσεις R_i ονομάζονται *ESCT* κυτταρικές συναρτήσεις του όρου.

Οι *ESCT* κυτταρικές συναρτήσεις αντιστοιχούν στα κύτταρα *Maitra* της κυτταρικής αρχιτεκτονικής *Maitra*, που μπορούμε να δούμε στην εικόνα 1.1. Τα κύτταρα *Maitra* έχουν 2 εισόδους και 1 έξοδο και υλοποιούν τις λογικές συναρτήσεις 2 εισόδων και 1 εξόδου.

Παράδειγμα 26 Έστω $f : \{0, 1\}^5 \rightarrow \{0, 1\}^4$. Η έκφραση $U = ((Y^{\{0,2\}} + x_1)\bar{x}_2 \oplus x_3)x_4 + x_5$ είναι ένας όρος *Maitra* της συνάρτησης f .

Παράδειγμα 27 Έστω $f : \{0, 1\}^5 \rightarrow \{0, 1\}$. Η έκφραση $U = (x_1\bar{x}_2 \oplus x_3)x_4 + x_5$ αποτελεί έναν όρο *Maitra* που εξαρτάται από τις μεταβλητές x_1, \dots, x_5 .

Ένας κύβος είναι μια ειδική περίπτωση ενός σύνθετου όρου, όπου οι συναρτήσεις $R_i(x, y)$ μπορούν να είναι της μορφής $xy, \bar{x}y, x\bar{y}, \bar{x}\bar{y}, x, y, \bar{x}, \bar{y}, 0, 1$. Όταν επιτρέπονται μόνο οι πρώτες 4 περιπτώσεις τότε ο κύβος είναι ελαχιστόρος.

Ορισμός 33 Μία έκφραση XOR-αθροίσματος από σύνθετους όρους (*Exclusive or Sum of Complex Terms - ESCT*) ή έκφραση Maitra E μιας boolean συνάρτησης f ορίζεται ως

$$E = \sum_{i=1}^m \oplus M_i \quad (5.1)$$

όπου m είναι ο αριθμός των σύνθετων όρων M_i . Σε κάθε σύνθετο όρο οι μεταβλητές εμφανίζονται με την ίδια σειρά.

Κάθε συνάρτηση $f : \{0, 1\}^n \rightarrow \{0, 1\}^k$ (και η αντίστοιχη χαρακτηριστική της συνάρτηση ψ) μπορεί να εκφραστεί σαν μια έκφραση ESCT με πολλούς διαφορετικούς τρόπους. Στα επόμενα, όταν αναφερόμαστε σε μια έκφραση ESCT της f ή της ψ , θα είναι ταυτόσημο. Επίσης, για ευκολία, δε θα κάνουμε τη διάκριση ανάμεσα σε ESOP και MVESOP, και θα μιλάμε πάντα για ESOP.

Στον παραπάνω ορισμό αν οι σύνθετοι όροι είναι κύβοι, τότε πρόκειται για μία έκφραση ESOP. Δηλαδή οι εκφράσεις ESOP είναι υποπεριπτώσεις των εκφράσεων ESCT (δεδομένου ότι οι κύβοι είναι υποπεριπτώσεις των όρων Maitra).

Παράδειγμα 28 Η έκφραση $f(x_1, x_2, x_3, x_4) = \{(x_1 \oplus x_2)x_3 + x_4\} \oplus \{x_1 + x_2 + x_3 + x_4\}$ είναι μια ESCT έκφραση για μια συνάρτηση f με 4 μεταβλητές εισόδου.

Ορισμός 34 Μία έκφραση ESCT μιας συνάρτησης f που έχει τον ελάχιστο αριθμό σύνθετων όρων ανάμεσα σε όλες τις δυνατές εκφράσεις ESCT για αυτή τη συνάρτηση λέγεται ελάχιστη ή ακριβής έκφραση ESCT της f .

Ορισμός 35 Το μέγεθος $s(E)$ μιας έκφρασης ESCT E είναι το πλήθος των σύνθετων όρων σε αυτή. Το βάρος $w(f)$ μιας συνάρτησης f είναι το πλήθος των σύνθετων όρων σε μια ακριβή έκφραση ESCT της f .

Όταν μιλάμε για το βάρος μιας συνάρτησης είναι σημαντικό να κάνουμε τη διάκριση ανάμεσα στις εκφράσεις ESOP και ESCT. Συνήθως είναι φανερό από τα συμφοραζόμενα για το τι είδους εκφράσεις πρόκειται. Για οποιαδήποτε συνάρτηση, όταν διαπραγματευόμαστε εκφράσεις ESCT το βάρος της είναι μικρότερο ή ίσο από την περίπτωση που μας ενδιαφέρουν εκφράσεις ESOP. Αυτό είναι λογικό, αφού οι εκφράσεις ESOP αποτελούν υποπερίπτωση των εκφράσεων ESCT.

Έχει αποδειχθεί [101] ότι το συμπλήρωμα ενός σύνθετου όρου είναι επίσης ένας σύνθετος όρος, επίσης ότι ισχύει για μια λογική συνάρτηση f : $w(f) = w(\bar{f})$.

Ορισμός 36 Μία συνάρτηση λέγεται υλοποιήσιμη ως ένας όρος Maitra ή υλοποιήσιμη ως cascade (*cascade realizable*) όταν έχει βάρος 1.

Έχει αποδειχθεί [42] ότι όλες οι συναρτήσεις με 2 μεταβλητές εισόδου είναι υλοποιήσιμες ως ένας όρος Maitra.

Ορισμός 37 Εκτός από τα κλασικά αναπτύγματα Shannon, θετικό Davio και αρνητικό Davio, που παρουσιάστηκαν στον ορισμό 11, μία λογική συνάρτηση f , με μία τουλάχιστον δυαδική μεταβλητή x και υποσυναρτήσεις τις f^0, f^1, f^2 όσον αφορά αυτή την μεταβλητή, μπορεί να αναπτυχθεί επίσης ως εξής:

$$f = (x + f^2) \oplus (x \oplus f^1) \quad (5.2)$$

$$f = (x + f^0) \oplus (x\overline{f^1}) \quad (5.3)$$

$$f = (x\overline{f^2}) \oplus (x \oplus f^0) \quad (5.4)$$

$$f = (x + \overline{f^0}) \oplus (\overline{x} + \overline{f^1}) \quad (5.5)$$

$$f = (\overline{x} + \overline{f^2}) \oplus \overline{f^0} \quad (5.6)$$

$$f = (x + \overline{f^2}) \oplus \overline{f^1} \quad (5.7)$$

$$f = (\overline{x}\overline{f^2}) \oplus (x \oplus \overline{f^1}) \quad (5.8)$$

$$f = (\overline{x}f^0) \oplus (\overline{x} + f^1) \quad (5.9)$$

$$f = (\overline{x} + f^2) \oplus (x \oplus \overline{f^0}) \quad (5.10)$$

Θα αποδείξουμε ότι τα παραπάνω αναπτύγματα είναι ισοδύναμα με τον ανάπτυγμα Shannon.

Για τη σχέση 5.2 ισχύει: $f = [x + f^2] \oplus [x \oplus f^1] = \overline{[x\overline{f^2}]} \oplus [x \oplus f^1] = \overline{[x(f^1 \oplus \overline{f^0})]} \oplus [x \oplus f^1] = \overline{x}f^0 \oplus \overline{x}f^1 \oplus \overline{x} \oplus f^1 = \overline{x}f^0 \oplus xf^1$

Για τη σχέση 5.3 ισχύει: $f = (x + f^0) \oplus (x\overline{f^1}) = \overline{x}f^0 \oplus xf^1 \oplus 1 = \overline{x}f^0 \oplus xf^1 \oplus (x \oplus \overline{x}) = \overline{x}f^0 \oplus xf^1$

Για τη σχέση 5.4 ισχύει: $f = \{x\overline{f^2}\} \oplus [x \oplus f^0] = x(f^0 \oplus \overline{f^1}) \oplus x \oplus f^0 = \overline{x}f^0 \oplus xf^1$

Οι σχέσεις 5.5, 5.6, 5.7 προέρχονται από τα αναπτύγματα Shannon, θετικό και αρνητικό Davio όταν αντιστρέψουμε κάθε όρο του αθροίσματος XOR. Οι σχέσεις 5.8, 5.9, 5.10 προέρχονται από τις σχέσεις 5.2, 5.3, 5.4 αντίστοιχα όταν αντιστρέψουμε κάθε όρο του αθροίσματος XOR.

Σημειώνουμε ότι τα παραπάνω αναπτύγματα υλοποιούν συναρτήσεις σε μορφή εκφράσεων ESCT.

5.2 Περιορισμένοι σύνθετοι όροι

Στους ορισμούς 31 και 32 των όρων Maitra οι συναρτήσεις R_i μπορεί να είναι οποιεσδήποτε από τις 16 λογικές συναρτήσεις 2 εισόδων και 1 εξόδου. Όμως έχει αποδειχθεί [45] ότι ένα σύνολο από 6 συναρτήσεις είναι αρκετό για να αναπαραστήσουμε οποιονδήποτε όρο Maitra. Ένα τέτοιο σύνολο ονομάζεται πλήρες σύνολο. Υπάρχουν πολλά ισοδύναμα τέτοια σύνολα [52]. Στα επόμενα θα υιοθετήσουμε, χωρίς περιορισμό της γενικότητας, το σύνολο συναρτήσεων που φαίνεται στον Πίνακα 5.1.

Ορισμός 38 Οι σύνθετοι όροι, για τους οποίους οι συναρτήσεις R_i που τους ορίζουν ανήκουν σε ένα πλήρες σύνολο, ονομάζονται περιορισμένοι σύνθετοι όροι (*restricted complex terms*) ή περιορισμένοι όροι Maitra (*restricted Maitra terms*).

Δείκτης	$G_i(x, y)$
1	$x + y$
2	$\bar{x} + y$
3	$\bar{x}y$
4	xy
5	$x \oplus y$
6	y

Πίνακας 5.1: Πλήρες σύνολο συναρτήσεων για σύνθετους όρους

Από εδώ και στο εξής, για λόγους απλότητας, όταν μιλάμε για σύνθετους όρους θα εννοούμε περιορισμένους σύνθετους όρους.

Για τις λογικές συναρτήσεις μιας εξόδου, υιοθετούμε την υπόθεση ότι η σταθερή είσοδος y κάθε σύνθετου όρου (που αναφέρεται στον Ορισμό 32) είναι η σταθερά 0. Επομένως, ένας σύνθετος όρος χαρακτηρίζεται από τις ESCT κυτταρικές συναρτήσεις του, σε κάθε μία από τις οποίες αντιστοιχεί ένας δείκτης σύμφωνα με τον πίνακα 5.1.

Ορισμός 39 Ένας σύνθετος όρος, που εξαρτάται μόνο από δυαδικές μεταβλητές, αναπαρίστανται ως μια ακολουθία από δείκτες, που αντιστοιχούν στις ESCT κυτταρικές συναρτήσεις του (με βάση την αντιστοίχιση του Πίνακα 5.1).

Παράδειγμα 29 Η συνάρτηση $f(x_1, x_2, x_3, x_4) = (x_1 \oplus x_2)x_3 + x_4$ έχει βάρος 1 και αναπαριστάται ως εξής: (1541) ή χρησιμοποιώντας δυαδικά ψηφία: (001101100001). Θεωρούμε ότι $x_1 = x_1 + y$, όπου $y = 0$, για αυτό ο πρώτος δείκτης είναι 1.

Ορισμός 40 Ένας σύνθετος όρος που εξαρτάται από ορισμένες δυαδικές μεταβλητές και ακριβώς μια μεταβλητή πολλαπλών τιμών ορίζεται ως ένας όρος πολλαπλών τιμών (*mv-term*).

Ένας *mv-term* μπορεί να αναπαρασταθεί χρησιμοποιώντας την MVMT αναπαράσταση. Εναλλακτικά χρησιμοποιούμε την παρακάτω αναπαράσταση.

Ορισμός 41 Ένας *mv-term* μπορεί να αναπαρασταθεί σαν ένας σύνθετος όρος που εξαρτάται μόνο από δυαδικές μεταβλητές (για ακολουθία από ESCT κυτταρικές συναρτήσεις, την οποία θα ονομάζουμε *2v-term*) με πρόθεμα μια αριθμητική τιμή που αντιστοιχεί στην μεταβλητή πολλαπλών τιμών (που θα την ονομάζουμε *mv-var*) μέσα σε άγκιστρα. Δηλαδή η αναπαράσταση του *mv-term* έχει την μορφή $\{mv - var\}2v - term$. Η *mv - var* προκύπτει από το *literal* Y^S της μεταβλητής πολλαπλών τιμών του σύνθετου όρου, αν αντιστοιχίσουμε κάθε πιθανή τιμή της μεταβλητής σε ένα *bit* ενός δυαδικού αριθμού και κάθε *bit* είναι 1 αν η αντίστοιχη τιμή ανήκει στο σύνολο S , αλλιώς είναι 0, με το *MSB* του δυαδικού αριθμού να αντιστοιχεί στην τιμή 0.

Παράδειγμα 30 Έστω $f : \{0, 1\}^3 \rightarrow \{0, 1, 2\}$ και ο σύνθετος όρος $Y^{\{1,2\}}x_1\bar{x}_2x_3$. Ο αντίστοιχος *mv-term* είναι $\{3\}434$ και έχει *2v-term* 434 και *mv - var* 3.

Δύο mn_term είναι ίσοι όταν έχουν ίσες mn_var και ίσους $2n_term$. Δύο mn_term είναι συμπληρωματικοί αν οι mn_var τους είναι συμπληρωματικές και οι $2n_term$ τους είναι συμπληρωματικοί.

Για να αναπαραστήσουμε έναν σύνθετο όρο πολλαπλών τιμών σε MVMT αναπαράσταση ακολουθούμε τα εξής βήματα: Πρώτα, εξετάζουμε το literal Y^S της μεταβλητής πολλαπλών τιμών. Σύμφωνα με τα στοιχεία του συνόλου S , θέτουμε τα αντίστοιχα bits στο MVMT της Y^S σε 1, με το LSB να αντιστοιχεί στην τιμή 0 της μεταβλητής. Προσθέτουμε μηδενικά γεμίματος, αν είναι απαραίτητο, έτσι ώστε το μήκος του MVMT να είναι δύναμη του δύο. Για παράδειγμα, αν έχουμε μια μεταβλητή Y 5 τιμών και το literal είναι $Y^{\{0,1,4\}}$, το MVMT του είναι [0001 0011].

Ακολούθως, κατασκευάζουμε το MVMT της $R_1(x_1, Y^S)$. Έστω m το MVMT της Y^S . Το παραγόμενο MVMT είναι η συνένωση δύο συμβολοσειρών m_l και m_r . Το μήκος κάθε μιας είναι ίσο με το μήκος του m . Η συνάρτηση R_1 μπορεί να είναι οποιαδήποτε από τις 6 ESCT κυτταρικές συναρτήσεις στο προαναφερόμενο σύνολο, και υπάρχουν 6 περιπτώσεις αντίστοιχα:

- αν $R_1 = G_1 = x_1 + Y^S$, τότε όλα τα bits του m_l (εξαιρώντας τα μηδενικά γεμίματος) είναι 1 και το m_r είναι το ίδιο με το m .
- αν $R_1 = G_2 = \bar{x}_1 + Y^S$, τότε το m_l είναι το ίδιο με το m και όλα τα bits του m_r (εξαιρώντας τα μηδενικά γεμίματος) είναι 1.
- αν $R_1 = G_3 = \bar{x}_1 Y^S$, τότε όλα τα bits του m_l είναι 0 και το m_r είναι το ίδιο με το m .
- αν $R_1 = G_4 = x_1 Y^S$, τότε το m_l είναι το ίδιο με το m και όλα τα bits του m_r είναι 0.
- αν $R_1 = G_5 = x_1 \oplus Y^S = x_1 \bar{Y}^S + \bar{x}_1 Y^S$, τότε τα bits του m_l είναι τα συμπληρώματα των αντίστοιχων bits του m (εξαιρώντας τα μηδενικά γεμίματος) και το m_r είναι το ίδιο με το m .
- αν $R_1 = G_6 = Y^S$, τότε το m_l και το m_r είναι τα ίδια με το m .

Αν ο όρος Maitra εξαρτάται από n δυαδικές μεταβλητές, επαναλαμβάνουμε το δεύτερο βήμα $n-1$ φορές, ώστε να κατασκευάσουμε το MVMT μετά την εφαρμογή των ESCT κυτταρικών συναρτήσεων G_2, \dots, G_n αντίστοιχα.

Τα παραπάνω θα εξηγηθούν με το ακόλουθο παράδειγμα.

Παράδειγμα 31 Έστω μια μεταβλητή Y 5 τιμών και το literal $Y^S = Y^{\{0,1,4\}}$. Το MVMT του είναι [0001 0011]. Υπάρχουν 3 μηδενικά γεμίματος.

Το MVMT του $G_1(x_1, Y^S)$ είναι [0001 1111 0001 0011].

Το MVMT του $G_2(x_1, Y^S)$ είναι [0001 0011 0001 1111].

Το MVMT του $G_3(x_1, Y^S)$ είναι [0000 0000 0001 0011].

Το MVMT του $G_4(x_1, Y^S)$ είναι [0001 0011 0000 0000].

Το MVMT του $G_5(x_1, Y^S)$ είναι [0000 1100 0001 0011].

Το MVMT του $G_6(x_1, Y^S)$ είναι [0001 0011 0001 0011].

Συγχώνευση N κυττάρων με δείκτες r_i	r
Περιττός αριθμός με δείκτη 1 και οποιοσδήποτε αριθμός με δείκτη 3	1
Περιττός αριθμός με δείκτη 2 και οποιοσδήποτε αριθμός με δείκτη 4	2
Άρτιος αριθμός με δείκτη 1 και οποιοσδήποτε αριθμός με δείκτη 3	3
Άρτιος αριθμός με δείκτη 2 και οποιοσδήποτε αριθμός με δείκτη 4	4
Περιττός αριθμός με δείκτη 5 και οποιοσδήποτε αριθμός με δείκτη 6	5
Άρτιος αριθμός με δείκτη 5 και οποιοσδήποτε αριθμός με δείκτη 6	6

Πίνακας 5.2: Συγχώνευση κελιών

5.3 Πράξεις με κύτταρα Maitra

Στην αναφορά [101] μπορούμε να βρούμε θεωρήματα, λήμματα και αξιώματα σχετικά με πράξεις ανάμεσα σε κύτταρα Maitra και τις αποδείξεις τους.

Λήμμα 2 (Συγχώνευση δύο κελιών). Η σχέση $G_{r_1}(x, y_1) \oplus_{G_{r_2}}(x, y_2) = G_r(x, y_1 \oplus y_2)$ όπου G_i είναι ESCT κυτταρικές συναρτήσεις, x είναι μια δυαδική μεταβλητή και $y_1 \neq y_2$, $y_1 \neq \bar{y}_2$ ισχύει αν και μόνο αν: $(r_1, r_2, r) = (1, 1, 3), (1, 3, 1), (3, 3, 3), (2, 2, 4), (2, 4, 2), (4, 4, 4), (5, 5, 6), (5, 6, 5), (6, 6, 6)$.

Το παραπάνω λήμμα μπορεί να αποδειχτεί πολύ εύκολα εξαντλητικά. Είναι σημαντικό να αναφέρουμε ότι το παραπάνω λήμμα ισχύει ακόμα και αν τα y_1, y_2 είναι mv-terms.

Λήμμα 3 (Συγχώνευση N κυττάρων.) Η σχέση $\sum \oplus_{F_{r_i}}(x, y_i) = F_r(x, \sum \oplus y_i)$, $y_i \neq y_k$, $y_i \neq \bar{y}_k, \forall k \neq i$ ισχύει σύμφωνα με τον Πίνακα 5.2.

Τα προηγούμενα λήμματα δηλώνουν ότι μπορούμε να συγχωνεύσουμε οποιονδήποτε αριθμό από κύτταρα Maitra τύπου 1 ή 3 σε ένα μόνο κύτταρο τύπου 1 ή 3. Το ίδιο ισχύει για τις ομάδες κυττάρων με δείκτες 2 ή 4 και 5 ή 6. Επομένως τα λήμματα υποδεικνύουν ότι υπάρχουν 3 διαφορετικά σύνολα από κύτταρα (κλάσεις κυττάρων), που είναι τα σύνολα $\{1, 3\}$, $\{2, 4\}$ και $\{5, 6\}$. Οποιοσδήποτε αριθμός από κύτταρα που ανήκουν στην ίδια κλάση συγχωνεύονται σε ένα κύτταρο της ίδιας κλάσης.

Θεώρημα 1 (Συμπληρωματικός σύνθετος όρος.) Η συμπληρωματική συνάρτηση ενός σύνθετου όρου είναι επίσης ένας σύνθετος όρος. Στον συμπληρωματικό σύνθετο όρο, όλα τα κύτταρα που ανήκουν στην κλάση $\{1, 3\}$ ή στην κλάση $\{2, 4\}$ αντικαθίσταται από τα άλλα της ίδια κλάσης. Τα κύτταρα που ανήκουν στην κλάση $\{5, 6\}$ παραμένουν τα ίδια στον συμπληρωματικό σύνθετο όρο.

Αξίωμα 1 Μια switching συνάρτηση και η συμπληρωματική της έχουν το ίδιο βάρος.

Θεώρημα 2 Αν $G_r(x, y^1)$, $G_q(x, y^2)$, και $G_g(x, y^3)$ είναι 3 κύτταρα Maitra, με $y^1, y^2, y^3 = y, \bar{y}$, τότε η εξίσωση:

$$G_r(x, y^1) \oplus G_q(x, y^2) = G_g(x, y^3) \quad (5.11)$$

ισχύει σύμφωνα με τον Πίνακα 5.3.

$y^1 = y$	$y^2 = y$	$y^3 = y$
r	q	g
1	4	5
1	5	4
3	4	6
3	6	4
4	5	1
4	6	3
$y^1 = y$	$y^2 = y$	$y^3 = \bar{y}$
r	q	g
1	2	6
1	6	4
2	3	5
2	5	1
2	6	3
3	5	4
$y^1 = y$	$y^2 = \bar{y}$	$y^3 = y$
r	q	g
1	4	6
1	6	2
2	1	5
2	3	6
2	5	3
2	6	1
$y^1 = y$	$y^2 = \bar{y}$	$y^3 = y$
r	q	g
3	4	5
3	5	2
5	1	2
5	4	3
6	3	2
6	4	1

Πίνακας 5.3: Συγχώνευση όρων.

5.4 Αντιστρέψιμες Λογικές Πύλες

Η επιστήμη της σύνθεσης λογικών κυκλωμάτων βασισμένων σε αντιστρέψιμες πύλες (reversible logic gates) έχει βασιστεί στην αρχή των von Neumann-Landauer [20], η οποία υποστηρίζει ότι οι κοινές μη αντιστρέψιμες λογικές πύλες, μοιραία, χάνουν ενέργεια απλά και μόνο επειδή κάποιες από τις δυνατές καταστάσεις εξόδου εξαφανίζονται. Χονδρικά η ενέργεια που χάνεται για κάθε έξοδο αντιστοιχεί στην ενέργεια του σήματος και οφείλεται στο λεγόμενο θερμικό θόρυβο (thermal noise). Η παραπάνω απώλεια ενέργειας αντιστοιχεί τόσο στην απώλεια πληροφορίας όσο και στην απώλεια λόγω τεχνολογικών παραγόντων (ατέλειες κατασκευής λογικών πυλών κτλ). Για κάθε bit πληροφορίας που χάνεται λόγω της μη αντιστρεψιμότητας παράγονται $KT \ln 2$ Joule ενέργειας που διαχέεται στο περιβάλλον (K είναι η σταθερά του Boltzmann και T είναι η θερμοκρασία λειτουργίας). Αντίθετα, λογικά κυκλώματα που συντίθενται από αντιστρέψιμες λογικές πύλες μπορούν να ξαναχρησιμοποιήσουν ένα κλάσμα της ενέργειας που χάνεται από το θερμικό θόρυβο και οφείλεται στην απώλεια της πληροφορίας. Μάλιστα, θεωρητικά, το κλάσμα αυτό μπορεί να φτάσει το 100%. Η άλλη παράμετρος απώλειας ενέργειας που οφείλεται σε τεχνολογικούς παράγοντες (κατασκευαστικές ατέλειες) αναμένεται να μειώνεται όσο η τεχνολογία βελτιώνεται. Παρόλα αυτά, τουλάχιστον μέχρι τώρα, η δεύτερη παράμετρος απώλειας ενέργειας είναι σημαντικά μεγαλύτερη από την πρώτη (τουλάχιστον στις επικρατέστερες τεχνολογίες κατασκευής ολοκληρωμένων κυκλωμάτων όπως στα κυκλώματα CMOS - Complementary Metal-Oxide Semiconductor). Κατά συνέπεια η σύνθεση κυκλωμάτων χρησιμοποιώντας αντιστρέψιμες πύλες δεν προσφέρει κάποιο ουσιαστικό πλεονέκτημα.

Είναι προφανές από τα παραπάνω ότι οι αντιστρέψιμες λογικές πύλες θα γίνονται όλο και πιο ελκυστικές για τη σύνθεση λογικών κυκλωμάτων λόγω της έμφυτης αυτής ιδιότητάς τους να μην χάνουν ενέργεια λόγω απώλειας πληροφορίας. Η σύνθεση δηλαδή αντιστρέψιμων κυκλωμάτων θα βοηθήσει στην περαιτέρω μείωση των απωλειών ενέργειας. Όπως επίσης θα φανεί στο κεφάλαιο για τους κβαντικούς αλγόριθμους, όλες οι κβαντικές πύλες είναι αντιστρέψιμες και όλες οι αντιστρέψιμες πύλες μπορούν να αποτελέσουν κβαντικούς τελεστές. Οποιαδήποτε, λοιπόν, μέθοδος σύνθεσης κυκλωμάτων που στηρίζεται σε αντιστρέψιμες πύλες, εκτός της μειωμένης κατανάλωσης σε ενέργεια, προσφέρει και τη (θεωρητική) δυνατότητα της απευθείας σύνθεσης και του αντίστοιχου κβαντικού κυκλώματος.

Οι περισσότερες από τις κλασικές λογικές πύλες που χρησιμοποιούνται για τη σύνθεση λογικών συναρτήσεων είναι μη αντιστρέψιμες (irreversible logic gates). Τέτοιες πύλες είναι οι γνωστές μας AND και OR. Παρ' όλα αυτά υπάρχουν και λογικές πύλες που είναι αντιστρέψιμες, όπως η πύλη XOR, όταν επεκταθεί να έχει δύο εισόδους και δύο εξόδους. Μια λογική πύλη λέγεται αντιστρέψιμη όταν αντιστοιχεί με μοναδικό τρόπο κάθε διάνυσμα εισόδου σε ένα και μοναδικό διάνυσμα εξόδου και αντίστροφα (1 - 1 απεικόνιση). Το παραπάνω γεγονός υπονοεί, επίσης, ότι οι αντιστρέψιμες πύλες έχουν τον ίδιο αριθμό εισόδων και εξόδων. Ισχύει ακόμα ότι ένα λογικό κύκλωμα που συντίθεται από αντιστρέψιμες λογικές πύλες είναι και το ίδιο αντιστρέψιμο.

Στη συνέχεια κάνουμε μια σύντομη αναφορά στις πιο γνωστές στοιχειώδεις αντιστρέψιμες λογικές πύλες της βιβλιογραφίας, οι οποίες χρησιμοποιούνται για τη

Πύλη	Είσοδοι/Έξοδοι	Μαθηματική περιγραφή
Μίας εισόδου μιας εξόδου.		
Ταυτότητας (Identity)	a/b	$b = a$
Αντιστροφής (NOT)	a/b	$b = \bar{a}$
Δύο εισόδων δύο εξόδων.		
Αποκλειστικό ή (XOR - CNOT)	$a, b/c, d$	$c = a, d = a \oplus b$
Feynman	$a, b/c, d$	$c = a, d = a + b$
Τριών εισόδων τριών εξόδων.		
Fredkin (Controlled swap)	$a, b, c/d, e, f$	Αν $c = 0$: $d = a, e = b, f = c$, Αν $c = 1$: $d = b, e = a, f = c$
Toffoli (CC-NOT)	$a, b, c/d, e, f$	$d = a, e = b,$ Αν $a, b \neq 1$: $c = c.$ Αν $a = b = 1$: $f = \bar{c}.$ Εναλλακτικά: $d = a, e = b, f = c \oplus ab$

Πίνακας 5.4: Τυπικές αντιστρέψιμες λογικές πύλες

σύνθεση αντιστρέψιμων λογικών κυκλωμάτων (Πίνακας 5.4).

Ανάμεσα στις παραπάνω είναι γνωστό ότι η πύλη Toffoli είναι καθολική (universal), δηλαδή μπορεί να χρησιμοποιηθεί από μόνη της για την υλοποίηση οποιουδήποτε λογικού κυκλώματος.

Η πύλη Toffoli ($\{a_1, \dots, a_n\} \rightarrow \{b_1, \dots, b_n\}$) μπορεί να επεκταθεί ώστε να ορίζεται για τη γενική περίπτωση των n εισόδων n εξόδων ως:

$$\left\{ \begin{array}{l} b_1 = a_1 \\ b_2 = a_2 \\ \dots \\ b_{n-1} = a_{n-1} \\ b_n = a_n \oplus a_1 a_2 \dots a_{n-1} \end{array} \right\}$$

Περαιτέρω γενίκευση (Γενικευμένη πύλη Toffoli) της παραπάνω πύλης παρουσιάστηκε στις εργασίες [95, 63, 47] και ο τυπικός της ορισμός ακολουθεί.

Ορισμός 42 Μια $k * k$ γενικευμένη πύλη Toffoli ($k * k$ Generalized Toffoli gate) ορίζεται ως:

$$\left\{ \begin{array}{l} b_1 = a_1 \\ b_2 = a_2 \\ \dots \\ b_{n-1} = a_{n-1} \\ b_n = a_n \oplus f(a_1 a_2 \dots a_{n-1}) \end{array} \right\}$$

, όπου A_i, P_i είναι οι είσοδοι και οι έξοδοι αντίστοιχα της πύλης και $f(a_1 a_2 \dots a_{n-1})$ είναι μια τυχαία συνάρτηση $n - 1$ μεταβλητών.

Στην εργασία [47] αποδεικνύεται ότι η κυτταρική αρχιτεκτονική Maitra (και κατά συνέπεια οποιαδήποτε ESCT έκφραση) αντιστοιχείται σε μια συστοιχία από γενικευμένες πύλες Toffoli. Άρα κάθε έκφραση ESCT μπορεί να απεικονιστεί ως αντιστρέψιμο κύκλωμα αφού αποτελείται από αντιστρέψιμες πύλες.

Κεφάλαιο 6

Ελαχιστοποίηση συναρτήσεων πολλών εξόδων

Σε αυτό το κεφάλαιο παρουσιάζουμε 2 διαφορετικούς φορμαλισμούς για την ελαχιστοποίηση συναρτήσεων πολλών εξόδων. Σε αυτούς η συνάρτηση πολλών εξόδων μετασχηματίζεται σε μια καινούρια λογική συνάρτηση (θα την ονομάζουμε χαρακτηριστική συνάρτηση), που είναι η συνάρτηση που θα ελαχιστοποιηθεί τελικά. Από τις ελάχιστες (ή σχεδόν ελάχιστες) εκφράσεις της χαρακτηριστικής συνάρτησης, μπορούμε εύκολα να αποκτήσουμε τις αντίστοιχες ελάχιστες (ή σχεδόν ελάχιστες) εκφράσεις για την αρχική συνάρτηση πολλών εξόδων. Σε αυτό το κεφάλαιο θα αναπαριστούμε την αρχική συνάρτηση πολλών εξόδων ως f και την αντίστοιχη χαρακτηριστική συνάρτηση ως g . Θα αναλύσουμε την διαδικασία που χρησιμοποιείται για να δημιουργήσουμε τη χαρακτηριστική συνάρτηση (που είναι διαφορετική σε κάθε προσέγγιση) και θα παρουσιάσουμε την προτεινόμενη αρχιτεκτονική (όπου είναι απαραίτητο).

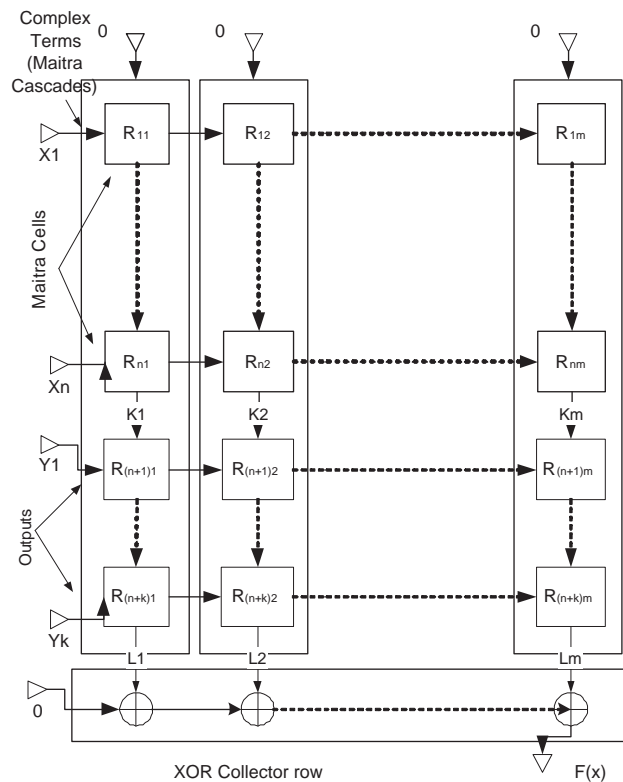
6.1 1η Προσέγγιση

Έστω $f(x_1, \dots, x_n) = \{f_0, f_1, \dots, f_{m-1}\}$ μια λογική συνάρτηση n εισόδων και m εξόδων που πρόκειται να ελαχιστοποιηθεί ($f_i, i = 0, \dots, (m-1)$ είναι οι έξοδοι της f). Σε αυτή την προσέγγιση η χαρακτηριστική της συνάρτηση g είναι μια λογική συνάρτηση με $n + \lceil \log(m) \rceil$ εισόδους και μια έξοδο που ορίζεται ως:

$$g(x_1, \dots, x_n, y_1, \dots, y_{\lceil \log(m) \rceil}) = f_i, i = \sum_{j=0}^{\lceil \log(m) \rceil - 1} (y_j 2^j) \quad (6.1)$$

Μια κυκλωματική αναπαράσταση αυτής της αρχιτεκτονικής φαίνεται στην Εικόνα 6.1:

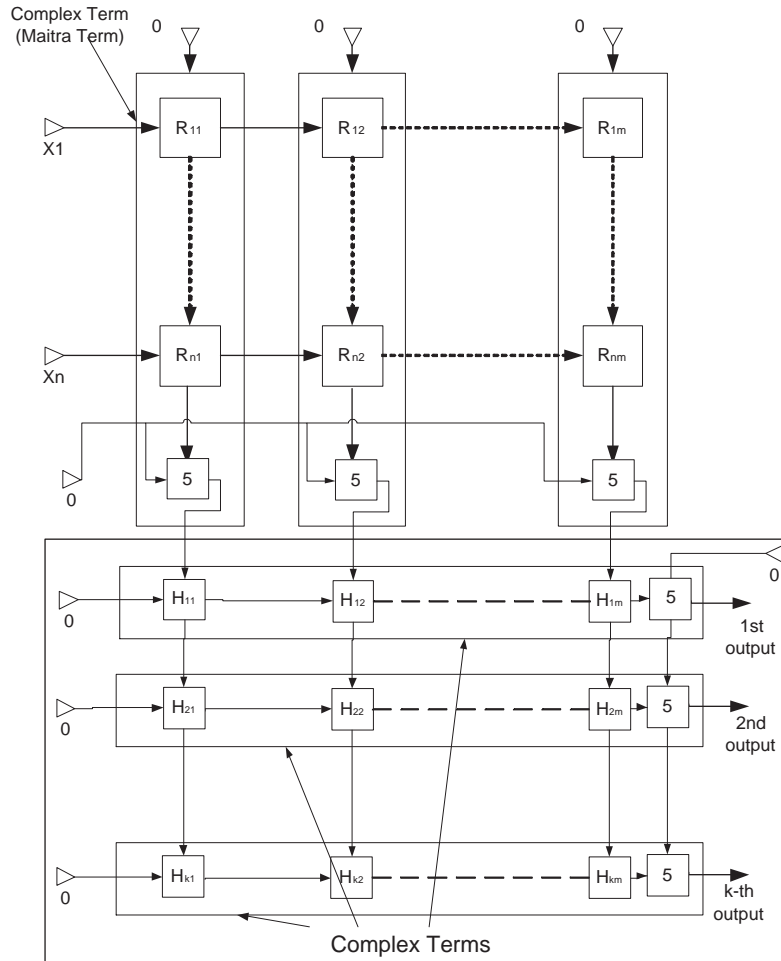
Για να αποσυνθέσουμε και να ελαχιστοποιήσουμε την χαρακτηριστική συνάρτηση g , χρησιμοποιούμε οποιονδήποτε αλγόριθμο ελαχιστοποίησης συναρτήσεων μιας εξόδου, αφού η g είναι όντως μια συνάρτηση μιας εξόδου, που έχει όμως περισσότερες μεταβλητές εισόδου, συγκρινόμενη με την αρχική συνάρτηση πολλών εξόδων. Από τις ελάχιστες εκφράσεις ESCT της g μπορούμε πολύ εύκολα να παράγουμε τις ελάχιστες εκφράσεις ESCT για την κάθε έξοδο f_i της αρχικής συνάρτησης, θέτοντας τις μεταβλητές $y_1, \dots, y_{\lceil \log(m) \rceil}$ στην αντίστοιχη τιμή σύμφωνα με την Εξίσωση 6.1.



Σχήμα 6.1: Κυτταρική δομή για την απεικόνιση της χαρακτηριστικής συνάρτησης (1η προσέγγιση).

Η παραπάνω αρχιτεκτονική έχει ένα μικρό μειονέκτημα. Μπορούμε να έχουμε μόνο μια από τις m πιθανές εξόδους σε κάθε δεδομένη χρονική στιγμή (δεν μπορούμε να έχουμε όλες τις πιθανές εξόδους την ίδια στιγμή). Για να ξεπεράσουμε αυτόν τον περιορισμό μπορούμε να μετασχηματίσουμε την παραπάνω αρχιτεκτονική σε αυτήν της Εικόνας 6.2. Αυτός ο μετασχηματισμός δεν προσθέτει επιπλέον στήλες στην αρχιτεκτονική και είναι πολύ απλός. Έχει δείχθει προηγουμένως ότι για να παράγουμε μια συγκεκριμένη έξοδο (έστω την έξοδο z) από την αρχιτεκτονική

της Εικόνας 6.1, πρέπει να θέσουμε τις εισόδους $y_1, \dots, y_{\lceil \log(m) \rceil}$ στην κατάλληλη σταθερή τιμή. Είναι εύκολο να αποδείξουμε ότι αν οι είσοδοι $y_1, \dots, y_{\lceil \log(m) \rceil}$ τεθούν σε μια συγκεκριμένη τιμή, τότε η έξοδος της L_i στην Εικόνα 6.2 θα είναι $0, 1, K_i$ ή \bar{K}_i . Διακρίνουμε τις παρακάτω περιπτώσεις:



Σχήμα 6.2: Ανανεωμένη κυτταρική δομή για την απεικόνιση της χαρακτηριστικής συνάρτησης (1η προσέγγιση)

- Αν $L_i = 0$ ή $L_i = 1$ τότε αυτή η στήλη δεν χρησιμοποιείται για τη έξοδο i . Επομένως το κύτταρο H_{iz} θα έχει δείκτη 6. Αν $L_i = 1$ τότε η έξοδος πρέπει να αντιστραφεί, οπότε αλλάζουμε την αρχική είσοδο της οριζόντιας γραμμής z από 0 σε 1 και αντίστροφα (ούτως ώστε να αντιστρέψουμε την έξοδο).

- Αν $L_i = K_i$ ή $L_i = \overline{K_i}$ τότε το κύτταρο H_{iz} θα έχει δείκτη 5 (ώστε να συνδυάσουμε την έξοδο L_i αυτής της στήλης με αυτές των υπολοίπων στηλών για να παράγουμε την έξοδο). Αν επιπλέον $L_i = \overline{K_i}$ αντιστρέφουμε επίσης την αρχική είσοδο της οριζόντιας γραμμής (όπως ακριβώς στην προηγούμενη περίπτωση).

Παράδειγμα 32 Έστω $f(x_1, x_2, x_3) = \{[12], [34], [23]\}$ μια λογική συνάρτηση 3 εισόδων και 3 εξόδων (όπου κάθε έξοδος εκφράζεται σε MT αναπαράσταση). Η χαρακτηριστική της συνάρτηση θα είναι: $g(x_1, x_2, x_3, y_1, y_2) = [00342312]$. Μια ελάχιστη έκφραση ESCT για την g είναι: $(12334) \oplus (11343) \oplus (62236) \oplus (21563)$. Η έκφραση για την $f_0 = [12]$ θα παραχθεί αν θέσουμε $(y_1, y_2) = (0, 0)$ και θα είναι: $(622) \oplus (215)$. Ομοίως για τη την $f_1 = [34]$ θα είναι $(123) \oplus (622)$ και για την $f_2 = [23]$ θα είναι: $(113) \oplus (215)$.

6.2 2η Προσέγγιση

Σε αυτή την περίπτωση η χαρακτηριστική συνάρτηση παράγεται ως η παρεμβολή των MT αναπαραστάσεων των εξόδων της f , σύμφωνα με τον ορισμό 14. Σε αυτή την περίπτωση η g έχει τόσες δυαδικές μεταβλητές όσες η f και μια επιπλέον μεταβλητή πολλών τιμών.

Οι εκφράσεις που θα παραχθούν από την αποσύνθεση και την ελαχιστοποίηση της g θα αποτελούνται από m -term. Όλοι οι 2 v -term (από τους αντίστοιχους m -term) της τελικής έκφρασης χρησιμοποιούνται για να παράγουμε τις ελάχιστες εκφράσεις για κάθε έξοδο (ενώ στην προηγούμενη περίπτωση μερικοί όροι χρησιμοποιήθηκαν). Η αρχική σταθερή είσοδος είναι διαφορετική (για κάποιες εισόδους μπορεί να είναι 1 και για άλλες 0). Αυτό δηλώνεται με την μεταβλητή πολλών τιμών.

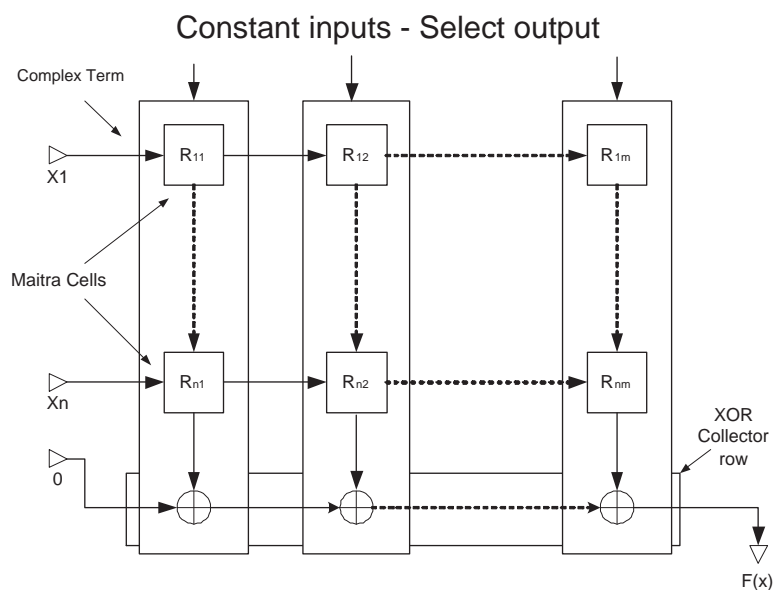
Το προτεινόμενο κύκλωμα φαίνεται στην Εικόνα 6.3.

Σε αυτή την αρχιτεκτονική δεν μπορούμε να παράγουμε κάθε έξοδο την ίδια χρονική στιγμή.

Για την αποσύνθεση και την ελαχιστοποίηση της χαρακτηριστικής συνάρτησης μπορούμε εύκολα να επεκτείνουμε (κατά έναν ευριστικό τρόπο) τον αλγόριθμο ελαχιστοποίησης συναρτήσεων μιας εξόδου Min1 [101].

Ακολουθεί ένα παράδειγμα για τη διαδικασία που χρησιμοποιούμε για να αποκτήσουμε κάθε έξοδο από την τελική έκφραση μιας χαρακτηριστικής συνάρτησης, όπως περιγράφηκε παραπάνω.

Παράδειγμα 33 Έστω $f = (x_1, x_2, x_3) = \{f_0, f_1, f_2\} = \{[23], [34], [12]\}$ και η χαρακτηριστική της συνάρτηση $g = [00360251]$. Μια ελάχιστη λύση για την g είναι: $g = (\{6\}436) \oplus (\{3\}336) \oplus (\{2\}343) \oplus (\{3\}233)$. Το πρώτο m -term $(\{6\}436)$ έχει 2 v -term (436) και m -var 6 και είναι ίσο με $Y^S x_1 \bar{x}_2$ (Y^S είναι η αρχική είσοδος του σύνθετου όρου με $S = \{0, 1\}$). Για τις εξόδους f_0, f_1 η αρχική είσοδος θα είναι $y = 1$ και η συνάρτηση που αναπαριστάται από τον 2 v -term είναι η [22] (ο σύνθετος όρος (436) έχει αρχική είσοδο 1 για αυτές τις εξόδους επειδή το περισσότερο σημαντικό και το επόμενο προς τα δεξιά bit του m -var=6 είναι 1). Για την έξοδο f_2 η συνάρτηση που αναπαριστάται από τον 2 v -term είναι η [00]



Σχήμα 6.3: Κυτταρική δομή για την απεικόνιση της χαρακτηριστικής συνάρτησης (2η προσέγγιση).

(επειδή το λιγότερο σημαντικό bit του $mv-var=6$ είναι 0). Ερμηνεύοντας ανάλογα τους άλλους $mv-term$, έχουμε τα ακόλουθα αποτελέσματα για τις εξόδους:

- f_0 : (436) με είσοδο 1, (336) με είσοδο 0, (343) με είσοδο 0, (233) με είσοδο 0.
- f_1 : (436) με είσοδο 1, (336) με είσοδο 1, (343) με είσοδο 1, (233) με είσοδο 1.
- f_2 : (436) με είσοδο 0, (336) με είσοδο 1, (343) με είσοδο 0, (233) με είσοδο 1.

6.3 Πειραματικά αποτελέσματα

Σε αυτό το κεφάλαιο παρουσιάστηκαν δύο διαφορετικοί αλγόριθμοι για την αποσύνθεση και την ελαχιστοποίηση λογικών συναρτήσεων πολλών εξόδων. Ο πρώτος υλοποιήθηκε σε C++ και ελέγχθηκε με μερικές μετροσυναρτήσεις. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 6.1 και συγκρίνονται με εκείνα των αλγορίθμων που παρουσιάζονται στο [37] (αλγόριθμος του Lee) και στο [80] (Minict) (στους οποίους χρησιμοποιούνται παρόμοιες αρχιτεκτονικές με αυτές που προτείνονται σε αυτή την εργασία).

Συνάρτηση	Αλγόριθμος του Lee	Minict (Song)	1η προσέγγιση
rd53	9	9	7
rd73	19	25	19
rd84	-	46	34
con1	-	8	8
f51m	21	30	19

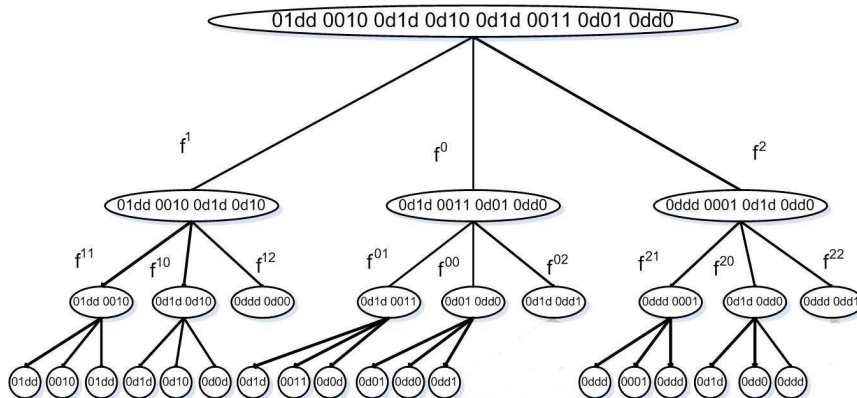
Πίνακας 6.1: Μετροσυναρτήσεις πολλών εξόδων

Κεφάλαιο 7

Ευριστική ESCT ελαχιστοποίηση μη πλήρως ορισμένων λογικών συναρτήσεων

Σε αυτή την ενότητα παρουσιάζεται ένας ευριστικός αλγόριθμος για την παραγωγή και την ελαχιστοποίηση εκφράσεων ESCT για τυχαίες μη πλήρως ορισμένες Boolean συναρτήσεις πολλών εξόδων. Ο αλγόριθμος αποτελεί επέκταση του αλγορίθμου QuickDCMIN που περιγράφηκε στο Κεφάλαιο 3 για την ευριστική ESOP ελαχιστοποίηση. Μάλιστα, καθώς οι εκφράσεις ESOP είναι υποπερίπτωση των εκφράσεων ESCT, θέτοντας κατάλληλα μια παράμετρο, ο νέος αλγόριθμος μπορεί να παράγει και εκφράσεις ESOP. Σε αυτή την περίπτωση θα δουλέψει ακριβώς όπως ο QuickDCMIN. Στην περιγραφή που ακολουθεί επικεντρωνόμαστε στην παραγωγή και ελαχιστοποίηση εκφράσεων ESCT, και, κυρίως, στις διαφορές από την ESOP ελαχιστοποίηση, ώστε να αποφύγουμε περιττές επαναλήψεις.

Κάθε συνάρτηση πολλών εξόδων f μετασχηματίζεται στην χαρακτηριστική της συνάρτηση ψ , στην οποία εφαρμόζεται ο προτεινόμενος αλγόριθμος, όπως στην περίπτωση των εκφράσεων ESOP. Η κύρια ιδέα είναι η ίδια, δηλαδή η αποσύνθεση της χαρακτηριστικής συνάρτησης στις απλούστερες υποσυναρτήσεις της, η κατασκευή των εκφράσεων ESCT για αυτές τις υποσυναρτήσεις και η σύνθεση των εκφράσεων ESCT για την αρχική συνάρτηση από τις εκφράσεις ESCT των υποσυναρτήσεων. Αυτή η αποσύνθεση πραγματοποιείται χρησιμοποιώντας, εκτός από τα κλασικά αναπτύγματα Shannon, θετικό Davio και αρνητικό Davio (Ορισμός 11) και τα 9 επιπλέον αναπτύγματα που παρουσιάστηκαν για τις εκφράσεις ESCT (Ορισμός 37).



Σχήμα 7.1: Γεννήτωρ δένδρο του Παραδείγματος 34

7.1 Αναπαράσταση μη πλήρως ορισμένων συναρτήσεων

Η αναπαράσταση των μη πλήρως ορισμένων συναρτήσεων γίνεται χρησιμοποιώντας την MVMT αναπαράστασή τους, ακριβώς όπως περιγράφεται στην Ενότητα 3.1.

7.2 Αποσύνθεση

Και στην περίπτωση της ευριστικής ελαχιστοποίησης εκφράσεων ESCT παράγεται ένα τριαδικό γεννήτωρ δένδρο, όπου η ρίζα του δένδρου αναπαριστά την αρχική συνάρτηση και τα παιδιά κάθε κόμβου αναπαριστούν τις 3 υποσυναρτήσεις του. Η αποσύνθεση για ένα συγκεκριμένο κόμβο σταματά όταν είτε η συσχετισμένη υποσυνάρτησή του είναι ίση με τη σταθερά μηδέν ή τη σταθερά ένα ή όταν έχουν πραγματοποιηθεί n αποσυνθέσεις (σε αντίθεση με την ESOP περίπτωση, όπου όταν η συσχετισμένη της υποσυνάρτηση είναι ίση με ένα συνεχίζεται η αποσύνθεση).

Υποθέτουμε ότι μία συνάρτηση είναι ίση με την σταθερά μηδέν, αν όλα τα bits της MVMT αναπαράστασής της είναι ίσα με 0 ή d . Σε αυτή την περίπτωση αυτοί οι συγκεκριμένοι αδιάφοροι όροι τίθενται ίσοι με 0. Ομοίως, υποθέτουμε ότι μια συνάρτηση είναι ίση με τη σταθερά ένα αν όλα τα bits της MVMT αναπαράστασής της (εξαιρώντας τα μηδενικά γεισίματα που αντιστοιχούν σε εικονικές μεταβλητές) είναι ίσα με 1 ή d . Για τη δικαιολόγηση αυτών των υποθέσεων ισχύουν τα προαναφερόμενα στην Ενότητα 3.2.

Παράδειγμα 34 Στην Εικόνα 7.1 μπορούμε να δούμε το γεννήτωρ δένδρο της συνάρτησης του παραδείγματος 15, θεωρώντας ότι παράγουμε εκφράσεις ESCT. Κάθε υποσυνάρτηση αναπαριστάται από την MVMT αναπαράστασή της. Για κάθε συνάρτηση, το αριστερό παιδί αντιστοιχεί στην υποσυνάρτηση f^1 , το μεσαίο παιδί

στην f^0 και το δεξί παιδί στην f^2 . Η υποσυνάρτηση f^{12} (που αναπαριστάται από τη συμβολοσειρά [0ddd 0d00]) είναι ίση με μηδέν, σύμφωνα με την υπόθεση που κάναμε. Ομοίως, οι υποσυναρτήσεις f^{02} και f^{22} (που αναπαριστώνται από τις συμβολοσειρές [0d1d 0dd1] και [0ddd 0dd1] αντίστοιχα) είναι ίσες με ένα, λαμβάνοντας υπ' όψιν ότι τα μηδενικά είναι μηδενικά γεμίσιματος και δεν αντιστοιχούν σε εξόδους της συνάρτησης.

7.3 Παραγωγή των εκφράσεων των φύλλων

Όπως αναφέρθηκε προηγουμένως, στην ESCT περίπτωση, η αποσύνθεση σταματά όταν ένας κόμβος του γεννήτορος δένδρου αναπαριστά τη σταθερά μηδέν ή τη σταθερά ένα. Σε αυτή την περίπτωση καμία έκφραση ESCT δεν παράγεται για αυτό το φύλλο.

Στο τελευταίο επίπεδο του δένδρου, ύστερα από n αποσυνθέσεις, τα φύλλα αναπαριστούν literals της μεταβλητής πολλαπλών τιμών. Επομένως, οι εκφράσεις ESCT για αυτά τα φύλλα έχουν βάρος 1 και η παραγωγή τους είναι άμεση. Κάθε έκφραση αποτελείται από ένα *mv_term* της μορφής $\{mv_var\}$, όπου *mv_var* είναι η τιμή της μεταβλητής πολλαπλών τιμών.

Οι MVMT αναπαραστάσεις που αντιστοιχούν σε ένα φύλλο μπορεί να περιέχουν ένα ή περισσότερα αδιάφορα bits. Κάθε πιθανός συνδυασμός από αναθέσεις για αυτά τα αδιάφορα bits οδηγεί σε μια έκφραση ESCT, όπως και στην περίπτωση των εκφράσεων ESOP. Αν υπάρχουν n bits με τιμή d , τότε υπάρχουν 2^n διαφορετικές εκφράσεις με βάρος ένα.

Παράδειγμα 35 Στον Πίνακα 7.1 μπορούμε να δούμε τις υποσυναρτήσεις που αναφέρονται στα φύλλα του τελευταίου επιπέδου του δένδρου της Εικόνας 7.1, τις MVMT αναπαραστάσεις τους και τις αντίστοιχες εκφράσεις ESCT. Οι υποσυναρτήσεις f^{102} , f^{012} , f^{000} , f^{211} , f^{212} , f^{200} και f^{202} θεωρούνται ίσες με τη σταθερά μηδέν και οι υποσυναρτήσεις f^{111} , f^{112} , f^{101} , f^{011} , f^{002} και f^{201} θεωρούνται ίσες με τη σταθερά ένα. Και στις δύο περιπτώσεις δεν παράγεται καμία έκφραση για τα φύλλα.

Ας εξετάσουμε την υποσυνάρτηση f^{001} . Η MVMT αναπαράστασή της είναι [0d01] (το λιγότερο σημαντικό bit αντιστοιχεί στην πρώτη έξοδο και το περισσότερο σημαντικό bit αντιστοιχεί στην τελευταία, εξαιρώντας τα μηδενικά γεμίσιματος), που αναπαριστά ένα literal. Υπάρχουν 3 έξοδοι και η πρώτη έξοδος είναι 1, η δεύτερη είναι 0 και η τρίτη είναι d . Κάθε δυνατός συνδυασμός για τις αδιάφορες εξόδους οδηγεί σε δύο συμβολοσειρές, οι οποίες είναι οι 001 και 101. Επομένως, οι δυνατές *mv_var* που παράγονται από αυτές τις συμβολοσειρές είναι οι 4 και 5. Οι εκφράσεις που παράγονται για την f^{001} είναι οι {4} και {5} (αγνοώντας το πρώτο μηδενικό γεμίσιματος και διαβάζοντας τη συμβολοσειρά με ανάποδη σειρά).

Σε MVMT αναπαράσταση, οι πιθανές εκφράσεις που παράγονται για την f^{001} είναι οι [0001] και [0101] (προσθέτοντας τα μηδενικά γεμίσιματος).

Υποσυναρτήσεις	MVMT	Εκφράσεις ESCT
f^{110}	[0010]	{2}
f^{100}	[0d10]	{2}, {3}
f^{010}	[0011]	{6}
f^{001}	[0d01]	{4}, {5}
f^{210}	[0001]	{4}

Πίνακας 7.1: Παραγωγή των εκφράσεων ESCT των φύλλων.

7.4 Σύνθεση

Κατά τη διάρκεια της φάσης σύνθεσης του αλγορίθμου οι εκφράσεις ESCT των υποσυναρτήσεων συνδυάζονται για να παράγουν τις εκφράσεις ESCT κάθε κόμβου, όπως γίνεται και στις εκφράσεις ESOP. Η παραγωγή των νέων εκφράσεων βασίζεται στα αναπτύγματα που ορίζονται στους ορισμούς 11 και 37 και πραγματοποιείται επαναληπτικά προς τη ρίζα του γεννήτορος δένδρου. Σημειώνουμε τη χρήση 12 αναπτυγμάτων στην ESCT περίπτωση, αντί για τα 3 κλασικά για την ESOP περίπτωση.

Παρατηρούμε ότι χρειάζεται να υπολογίσουμε το συμπλήρωμα μερικών εκφράσεων, όταν χρησιμοποιούμε τα αναπτύγματα του ορισμού 37. Αυτό το επιτυγχάνουμε υπολογίζοντας το συμπλήρωμα ενός μόνο τυχαία επιλεγμένου όρου της έκφρασης, κρατώντας τους άλλους όρους अपαράλλακτους. (Βεβαίως, θα μπορούσαμε να υπολογίσουμε το συμπλήρωμα οποιουδήποτε περιττού αριθμού όρων, αλλά αυτή η επιλογή θα οδηγούσε σε αυξημένο χρόνο εκτέλεσης του αλγορίθμου.) Επίσης, συγχωνεύουμε 2 όρους σε 1, όπου αυτό είναι δυνατό. Μπορούμε να διακρίνουμε και εδώ 3 διαφορετικές περιπτώσεις, ανάλογα με τον αριθμό των υποσυναρτήσεων που είναι ίσες με την σταθερά μηδέν ή ένα.

Στην πρώτη περίπτωση, όλες οι υποσυναρτήσεις ενός συγκεκριμένου κόμβου δεν είναι σταθερές μηδέν ή ένα. Έστω n το επίπεδο ενός κόμβου (ο οποίος αναπαριστά μια συνάρτηση f με $n + 1$ μεταβλητές) και k_0, k_1 και k_2 ο αριθμός των ήδη υπολογισμένων εκφράσεων για τις υποσυναρτήσεις f^0, f^1 και f^2 της f αντίστοιχα. Οι εκφράσεις για κάθε ζευγάρι υποσυναρτήσεων συνδυάζονται ώστε να παράγουν καινούριες εκφράσεις για την f χρησιμοποιώντας και τα 12 ορισμένα αναπτύγματα. Ο συνολικός αριθμός των παραγόμενων εκφράσεων είναι $4(k_0k_1 + k_0k_2 + k_1k_2)$. Από αυτές τις εκφράσεις, αυτές οι οποίες δεν επαληθεύουν την f , καθώς επίσης και αυτές με μέγεθος μεγαλύτερο από το ελάχιστο για όλες τις προηγούμενα υπολογιζόμενες εκφράσεις της f (που εκτιμάται ως το βάρος της f), απορρίπτονται. Φυσικά, ο λόγος που κάποιες από τις παραγόμενες εκφράσεις δεν επαληθεύουν την f είναι ο ίδιος που έχει εξηγηθεί στην Ενότητα 3.4.

Πιο συγκεκριμένα, για δυο εκφράσεις g_a (της υποσυνάρτησης f^a με μέγεθος s_a) και g_b (της υποσυνάρτησης f^b με μέγεθος s_b) μια καινούρια έκφραση g παράγεται χρησιμοποιώντας τα αναπτύγματα που ορίστηκαν παραπάνω. Αν δύο όροι είναι ίσοι ή συμπληρωματικοί, τότε μπορούν να συγχωνευθούν σε έναν όρο. Έτσι, αν υπάρχουν s_1 κοινοί όροι και s_2 συμπληρωματικοί όροι στις g_a και g_b , τότε το μέγεθος της τελικής έκφρασης ESCT είναι $s_a + s_b - s_1 - s_2$.

Τα παραπάνω θα διευκρινιστούν περαιτέρω με το επόμενο παράδειγμα.

Παράδειγμα 36 Έστω ότι (χρησιμοποιώντας τον αλγόριθμο που περιγράψαμε προηγουμένως) έχουμε τις επόμενες εκφράσεις *ESCT* για τις υποσυναρτήσεις της συνάρτησης f του παραδείγματος της Εικόνας 7.1 στο τρίτο επίπεδο του γεννητόρος δένδρου. Υπάρχουν 32 εκφράσεις για την υποσυνάρτηση f^0 , με βάρος 2, οι οποίες είναι: $\{1\}32 \oplus \{2\}21, \{1\}32 \oplus \{5\}51, \{1\}32 \oplus \{3\}21, \{1\}32 \oplus \{4\}51, \{1\}34 \oplus \{5\}41, \{1\}34 \oplus \{2\}51, \{1\}34 \oplus \{4\}41, \{1\}34 \oplus \{3\}51, \{6\}12 \oplus \{2\}23, \{6\}12 \oplus \{5\}53, \{6\}12 \oplus \{3\}23, \{6\}12 \oplus \{4\}53, \{6\}14 \oplus \{5\}43, \{6\}14 \oplus \{2\}53, \{6\}14 \oplus \{4\}43, \{6\}14 \oplus \{3\}53, \{1\}62 \oplus \{2\}21, \{1\}62 \oplus \{5\}51, \{1\}62 \oplus \{3\}21, \{1\}62 \oplus \{4\}51, \{1\}64 \oplus \{5\}41, \{1\}64 \oplus \{2\}51, \{1\}64 \oplus \{4\}41, \{1\}64 \oplus \{3\}51, \{6\}62 \oplus \{2\}23, \{6\}62 \oplus \{5\}53, \{6\}62 \oplus \{3\}23, \{6\}62 \oplus \{4\}53, \{6\}64 \oplus \{5\}43, \{6\}64 \oplus \{2\}53, \{6\}64 \oplus \{4\}43, \{6\}64 \oplus \{3\}53.$

Η υποσυνάρτηση f^1 έχει μία μόνο έκφραση, η οποία είναι: $\{2\}16$ (ή $[0111\ 0010\ 0111\ 0010]$ σε *MVMT* αναπαράσταση).

Τέλος, οι δύο εκφράσεις για την υποσυνάρτηση f^2 είναι: $\{3\}15, \{3\}65$ (ή $[0000\ 0001\ 0111\ 0110]$ και $[0001\ 0001\ 0110\ 0110]$ σε *MVMT* αναπαράσταση αντίστοιχα).

Ο αριθμός των παραγόμενων εκφράσεων για την f είναι $4 \cdot (32 \cdot 1 + 32 \cdot 2 + 1 \cdot 2) = 4 \cdot (32 + 64 + 2) = 4 \cdot 98 = 392$. Όπως έχει εξηγηθεί, δεν είναι απαραίτητο όλες οι παραπάνω εκφράσεις να επαληθεύουν την f και πρέπει να τις ελέγξουμε. Επίσης ενδιαφερόμαστε να ελαχιστοποιήσουμε τον αριθμό των όρων στις παραγόμενες εκφράσεις.

Το βάρος της f^0 είναι 2 και το βάρος της f^1 και της f^2 είναι 1. Άρα, αν συνδυάσουμε τις εκφράσεις για την f^1 και την f^2 , θα βρούμε εκφράσεις με βάρος 2. Οι τύποι που μας ενδιαφέρουν (που συνδυάζουν τις f^1 και f^2) και οι αντίστοιχες παραγόμενες εκφράσεις μέσα σε παρενθέσεις είναι:

1. $f = f^1 \oplus \bar{x}_n f^2$ ($\{2\}166 \oplus \{3\}153, \{2\}166 \oplus \{3\}653$)
2. $f = (x_n + f^2) \oplus (x_n \oplus f^1)$ ($\{2\}165 \oplus \{3\}151, \{2\}165 \oplus \{3\}651$)
3. $f = (x_n + \overline{f^2}) \oplus \overline{f^1}$ ($\{5\}366 \oplus \{4\}351, \{5\}366 \oplus \{4\}651$)
4. $f = (\bar{x}_n \overline{f^2}) \oplus (x_n \oplus \overline{f^1})$ ($\{5\}365 \oplus \{4\}353, \{5\}365 \oplus \{4\}653$)

Σε *MVMT* αναπαράσταση οι εκφράσεις είναι:

$$\begin{aligned} \alpha) f &= f^1 \oplus \bar{x}_n f^2 \\ &[0111\ 0010\ 0111\ 0010\ 0111\ 0010\ 0111\ 0010] \oplus \\ &[0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 0111\ 0110] \text{ και} \\ &[0111\ 0010\ 0111\ 0010\ 0111\ 0010\ 0111\ 0010] \oplus \\ &[0000\ 0000\ 0000\ 0000\ 0001\ 0001\ 0110\ 0110] \\ \beta) f &= (x_n + f^2) \oplus (x_n \oplus f^1) \\ &[0000\ 0101\ 0000\ 0101\ 0111\ 0010\ 0111\ 0010] \oplus \\ &[0111\ 0111\ 0111\ 0111\ 0000\ 0001\ 0111\ 0110] \text{ και} \\ &[0000\ 0101\ 0000\ 0101\ 0111\ 0010\ 0111\ 0010] \oplus \\ &[0111\ 0111\ 0111\ 0111\ 0001\ 0001\ 0110\ 0110] \end{aligned}$$

$$\begin{aligned} \gamma) f &= (x_n + \overline{f^2}) \oplus \overline{f^1} \\ &[0000\ 0101\ 0000\ 0101\ 0000\ 0101\ 0000\ 0101] \oplus \\ &[0111\ 0111\ 0111\ 0111\ 0111\ 0110\ 0000\ 0001] \text{ και} \\ &[0000\ 0101\ 0000\ 0101\ 0000\ 0101\ 0000\ 0101] \oplus \\ &[0111\ 0111\ 0111\ 0111\ 0110\ 0110\ 0001\ 0001] \\ \delta) f &= (\overline{x_n} \overline{f^2}) \oplus (x_n \oplus \overline{f^1}) \\ &[0111\ 0010\ 0111\ 0010\ 0000\ 0101\ 0000\ 0101] \oplus \\ &[0000\ 0000\ 0000\ 0000\ 0111\ 0110\ 0000\ 0001] \text{ και} \\ &[0111\ 0010\ 0111\ 0010\ 0000\ 0101\ 0000\ 0101] \oplus \\ &[0000\ 0000\ 0000\ 0000\ 0110\ 0110\ 0001\ 0001] \end{aligned}$$

Αν ελέγξουμε τις παραπάνω εκφράσεις, θα προσδιορίσουμε ότι μόνο η δεύτερη έκφραση κάθε υποπερίπτωσης επαληθεύει την f . Επομένως, οι έγκυρες εκφράσεις είναι οι ακόλουθες 4: $\{2\}166 \oplus \{3\}653$, $\{2\}165 \oplus \{3\}651$, $\{5\}366 \oplus \{4\}651$, $\{5\}365 \oplus \{4\}653$.

Σε *MVMT* αναπαράσταση οι έγκυρες εκφράσεις είναι:

$$\begin{aligned} \alpha) &[0111\ 0010\ 0111\ 0010\ 0111\ 0010\ 0111\ 0010] \oplus \\ &[0000\ 0000\ 0000\ 0000\ 0001\ 0001\ 0110\ 0110], \\ \beta) &[0000\ 0101\ 0000\ 0101\ 0111\ 0010\ 0111\ 0010] \oplus \\ &[0111\ 0111\ 0111\ 0111\ 0001\ 0001\ 0110\ 0110], \\ \gamma) &[0000\ 0101\ 0000\ 0101\ 0000\ 0101\ 0000\ 0101] \oplus \\ &[0111\ 0111\ 0111\ 0111\ 0110\ 0110\ 0001\ 0001] \text{ και} \\ \delta) &[0111\ 0010\ 0111\ 0010\ 0000\ 0101\ 0000\ 0101] \oplus \\ &[0000\ 0000\ 0000\ 0000\ 0110\ 0110\ 0001\ 0001] \end{aligned}$$

Αν συνδυάσουμε την f^0 με την f^1 ή την f^0 με την f^2 οι παραγόμενες εκφράσεις θα έχουν βάρος 3 (και θα απορριφθούν), εκτός αν υπάρχουν ίσοι ή συμπληρωματικοί όροι ανάμεσα στις εκφράσεις για την f^0 και την f^1 ή την f^0 και την f^2 αντίστοιχα, που μπορούν να συγχωνευθούν.

Οι όροι των εκφράσεων για την f^1 και την f^2 είναι οι $\{2\}16$, $\{3\}15$ και $\{3\}65$ και τα συμπληρώματά τους είναι οι $\{5\}36$, $\{4\}35$ και $\{4\}65$. Κανένας από αυτούς τους 6 όρους δεν υπάρχει στις εκφράσεις για την υποσυνάρτηση f^0 , επομένως δεν μπορεί να παραχθεί καμία επιπλέον έκφραση για την f με βάρος 2.

Η δεύτερη περίπτωση είναι όταν ακριβώς μία από τις υποσυναρτήσεις είναι μηδέν ή ένα. Για τις πλήρως ορισμένες συναρτήσεις, όταν μια υποσυνάρτηση είναι μηδέν ή ένα, τότε οι άλλες δύο υποσυναρτήσεις είναι ίσες ή συμπληρωματικές αντίστοιχα και χρειάζεται να εξετάσουμε μόνο την μία από αυτές για να παράγουμε τις κατάλληλες εκφράσεις [99]. Αυτό όμως δεν ισχύει στις μη πλήρως ορισμένες συναρτήσεις, για τους λόγους που εξηγήσαμε στην Ενότητα 3.4.

Ανάλογα με το ποια υποσυνάρτηση είναι μηδέν ή ένα, οι προαναφερόμενοι τύποι στους Ορισμούς 11 και 37 αλλάζουν κατάλληλα, αντικαθιστώντας την αντίστοιχη υποσυνάρτηση με 0 ή 1, όπως μπορούμε να δούμε στον Πίνακα 7.2, λαμβάνοντας υπ' όψιν και τις δύο άλλες υποσυναρτήσεις.

Έστω f^a , f^b , $a, b \in \{0, 1, 2\}$, $a \neq b$ οι δύο υποσυναρτήσεις που δεν είναι μηδέν ούτε ένα και k_a , k_b ο αριθμός των προϋπολογισμένων εκφράσεων για τις f^a , f^b αντίστοιχα. Ο αριθμός των καινούριων εκφράσεων είναι $k_a + k_b + 4k_a k_b$. Επίσης, οι κοινοί ή συμπληρωματικοί όροι συγχωνεύονται σε έναν όρο.

Συνθήκη	Τύποι
$f^0 = 0$	$f = x_n f^1, f = x_n f^2,$ $f = f^1 \oplus \bar{x}_n f^2, f = (x_n + f^2) \oplus (x_n \oplus f^1),$ $f = (x_n + f^2) \oplus f^1, f = \bar{x}_n f^2 \oplus (x_n \oplus f^1)$
$f^1 = 0$	$f = \bar{x}_n f^0, f = \bar{x}_n f^2,$ $f = f^0 \oplus x_n f^2, f = x_n f^2 \oplus (x_n \oplus f^0),$ $f = (\bar{x}_n + f^2) \oplus f^0, f = (\bar{x}_n + f^2) \oplus (x_n \oplus f^0)$
$f^2 = 0$	$f = f^0, f = f^1,$ $f = \bar{x}_n f^0 \oplus x_n f^1, f = (x_n + f^0) \oplus x_n f^1,$ $f = (x_n + f^0) \oplus (\bar{x}_n + f^1), f = \bar{x}_n f^0 \oplus (\bar{x}_n + f^1)$
$f^0 = 1$	$f = \bar{x}_n + f^1, f = \bar{x}_n + f^2,$ $f = \bar{x}_n f^2 \oplus f^1, f = (x_n + f^2) \oplus (x_n \oplus f^1),$ $f = (x_n + f^2) \oplus f^1, f = \bar{x}_n f^2 \oplus (x_n \oplus f^1)$
$f^1 = 1$	$f = x_n + f^0, f = x_n + f^2,$ $f = x_n f^2 \oplus f^0, f = x_n f^2 \oplus (x_n \oplus f^0),$ $f = (\bar{x}_n + f^2) \oplus f^0, f = (\bar{x}_n + f^2) \oplus (x_n \oplus f^0)$
$f^2 = 1$	$f = x_n \oplus f^0, f = x_n \oplus f^1,$ $f = \bar{x}_n f^0 \oplus x_n f^1, f = (x_n + f^0) \oplus x_n f^1,$ $f = (x_n + f^0) \oplus (\bar{x}_n + f^1), f = \bar{x}_n f^0 \oplus (\bar{x}_n + f^1)$

Πίνακας 7.2: Τύποι όταν μια υποσυνάρτηση είναι μηδέν ή ένα

Όταν χρησιμοποιούμε έναν τύπο της μορφής $f = G_i(x_n, F^a)$, $F^a = f^a$ ή $F^a = \bar{f}^a$, G_i είναι μια κυτταρική ESCT συνάρτηση με $i \in \{1, 2, 3, 4, 5, 6\}$, $a \in \{0, 1, 2\}$, το μέγεθος της καινούριας έκφρασης είναι ίσο με το μέγεθος της έκφρασης για την F^a .

Όταν χρησιμοποιούμε έναν τύπο της μορφής $f = G_i(x_n, F^a) \oplus G_j(x_n, F^b)$, $F^a = f^a$ ή $F^a = \bar{f}^a$, $F^b = f^b$ ή $F^b = \bar{f}^b$, G_i και G_j είναι κυτταρικές ESCT συναρτήσεις με $i, j \in \{1, 2, 3, 4, 5, 6\}$, $a, b \in \{0, 1, 2\}$, για δύο εκφράσεις g_a (της F^a με μέγεθος s_a) και g_b (της F^b με μέγεθος s_b), με s_1 κοινούς και s_2 συμπληρωματικούς όρους ανάμεσα στην g_a και στην g_b , παράγεται μια καινούρια έκφραση g με μέγεθος $s_a + s_b - s_1 - s_2$.

Παράδειγμα 37 Χρησιμοποιώντας τον αλγόριθμο που περιγράφουμε, έχουμε τις ακόλουθες εκφράσεις για τις υποσυναρτήσεις της f^1 από το παράδειγμα της Εικόνας 7.1. Η υποσυνάρτηση f^{10} έχει 4 εκφράσεις (αγνοώντας τα γεμίσματα με μηδενικά): $\{2\}1, \{2\}6, \{3\}1, \{3\}6$.

Η υποσυνάρτηση f^{11} έχει 2 εκφράσεις: $\{2\}1, \{2\}5$.

Τέλος, η υποσυνάρτηση f^{12} θεωρείται ίση με την σταθερά μηδέν.

Θα χρησιμοποιήσουμε τους τύπους από τον πίνακα 7.2 για την περίπτωση που $f^2 = 0$.

Ο πρώτος τύπος ($f^1 = f^{10}$) παράγει 4 εκφράσεις: $\{2\}16, \{2\}66, \{3\}16, \{3\}66$.

Αλλά μόνο η πρώτη έκφραση ($\{2\}16$) επαληθεύει την f^1 .

Ο δεύτερος τύπος ($f^1 = f^{11}$) οδηγεί σε δύο εκφράσεις: $\{2\}16, \{2\}56$.

Συνθήκη	Τύποι
$f^0 = 0, f^2 = 0$	$f = f^1, f = x_n f^1$
$f^1 = 0, f^2 = 0$	$f = f^0, f = \bar{x}_n f^0$
$f^0 = 1, f^2 = 1$	$f = \bar{x}_n + f^1, f = x_n \oplus \bar{f}^1$
$f^1 = 1, f^2 = 1$	$f = x_n + f^0, f = x_n \oplus f^0$
$f^0 = 0, f^2 = 1$	$f = x_n f^1, f = x_n \oplus \bar{f}^1$
$f^0 = 1, f^2 = 0$	$f = \bar{x}_n + f^1, f = f^1$
$f^1 = 0, f^2 = 1$	$f = \bar{x}_n f^0, f = x_n \oplus f^0$
$f^1 = 1, f^2 = 0$	$f = x_n + f^0, f = f^0$

Πίνακας 7.3: Τύποι όταν δύο υποσυναρτήσεις είναι μηδέν ή ένα

Ξανά, μόνο η πρώτη επαληθεύει την f^1 . Επιπλέον, αυτή η έκφραση έχει ήδη υπολογιστεί από τον πρώτο τύπο.

Μέχρι τώρα, έχουμε μόνο 1 νόμιμη έκφραση με βάρος 1. Όπως μπορούμε να δούμε οι 4 άλλοι τύποι του Πίνακα 7.2 παράγουν εκφράσεις μεγέθους 2, εκτός αν υπάρχουν κοινοί ή συμπληρωματικοί όροι ανάμεσα στις εκφράσεις της f^{10} και της f^{11} . Οι όροι των εκφράσεων της f^{11} είναι οι $\{2\}1$ και $\{2\}5$. Τα συμπληρώματά τους είναι οι $\{5\}3$ και $\{5\}5$ αντίστοιχα. Από αυτούς τους 4 όρους, μόνο ο $\{2\}1$ υπάρχει στις εκφράσεις της f^{10} . Επομένως, συνδυάζουμε τις εκφράσεις της f^{10} και της f^{11} που περιέχουν τον όρο $\{2\}1$. Οι 4 προαναφερόμενοι τύποι και οι παραγόμενες εκφράσεις μέσα σε παρενθέσεις είναι:

1. $f^1 = \bar{x}_n f^{10} \oplus x_n f^{11}$ ($\{2\}13 \oplus \{2\}14 = \{2\}16$)
2. $f^1 = (x_n + f^{10}) \oplus x_n \bar{f}^{11}$ ($\{2\}11 \oplus \{5\}34 = \{2\}16$)
3. $f^1 = (x_n + \bar{f}^{10}) \oplus (\bar{x}_n + \bar{f}^{11})$ ($\{5\}31 \oplus \{5\}32 = \{2\}16$)
4. $f^1 = \bar{x}_n \bar{f}^{10} \oplus (\bar{x}_n + f^{11})$ ($\{5\}33 \oplus \{2\}12 = \{2\}16$)

Επομένως η παραγόμενη έκφραση για την f^1 είναι μόνο μία: $\{2\}16$.

Όπως παρατηρούμε, όλοι οι τύποι οδηγούν στις ίδιες εκφράσεις, που είναι ήδη γνωστές από τους δύο άλλους τύπους.

Στην τρίτη περίπτωση, υπάρχουν ακριβώς 2 υποσυναρτήσεις, που είναι ίσες με μηδέν ή ένα. Τότε εξετάζουμε την τρίτη υποσυνάρτηση. Οι τύποι των αναπτυγμάτων αλλάζουν, όπως υποδεικνύει ο Πίνακας 7.3.

Έστω f^i , $i \in \{0, 1, 2\}$ η υποσυνάρτηση που δεν είναι μηδέν ή ένα και k ο αριθμός των ήδη υπολογισμένων εκφράσεων. Ο αριθμός των καινούριων ESCT είναι $2k$. Για μία έκφραση της f^i με μέγεθος s οι καινούριες εκφράσεις της f έχουν μέγεθος s .

Παράδειγμα 38 Ας εξετάσουμε την f^{21} του παραδείγματος της Εικόνας 7.1. Οι υποσυναρτήσεις f^{211} και f^{212} είναι ίσες με μηδέν. Η υποσυνάρτηση f^{210} έχει μόνο 1 έκφραση: $\{4\}$.

Χρησιμοποιούμε τους τύπους $f^{21} = f^{210}$ και $f^{21} = \bar{x}_n f^{210}$ και οι παραγόμενες εκφράσεις είναι οι: $\{4\}6$, $\{4\}3$.

Συνθήκη	Τύποι
$f^0 = 0, f^1 = 1, f^2 = 0$	$f = x_n$
$f^0 = 0, f^1 = 1, f^2 = 1$	$f = x_n$
$f^0 = 1, f^1 = 0, f^2 = 0$	$f = \bar{x}_n$
$f^0 = 1, f^1 = 0, f^2 = 1$	$f = \bar{x}_n$

Πίνακας 7.4: Τύποι για τετριμμένες περιπτώσεις.

Και οι δύο παραπάνω εκφράσεις επαληθεύουν την f^{21} .

Τέλος, σημειώνουμε ότι στον Πίνακα 7.3 δεν υπάρχουν τύποι για τις περιπτώσεις $f^0 = 0, f^1 = 0$ και $f^0 = 1, f^1 = 1$. Αυτοί οι συνδυασμοί είναι αδύνατον να υπάρξουν στο γεννήτωρ δένδρο, αλλιώς η γονική συνάρτηση θα ήταν $f = 0$ ή $f = 1$ αντίστοιχα, επομένως, η αποσύνθεση θα σταματούσε στο επίπεδο του πατέρα.

Επιπλέον, δεν υπάρχουν τύποι για τις περιπτώσεις $f^0 = 0, f^1 = 1$ και $f^0 = 1, f^1 = 0$. Μπορούμε να παρατηρήσουμε ότι και στις δύο περιπτώσεις τα bits της MVMT αναπαράστασης της μιας υποσυνάρτησης (που είναι μηδέν) είναι 0 και d και της άλλης υποσυνάρτησης (που είναι ένα) είναι 1 και d . Επομένως, τα παραγόμενα bits της MVMT αναπαράστασης της f^2 είναι είτε 1 είτε d . Αν όλα τα παραγόμενα bits είναι d τότε θεωρούμε ότι $f^2 = 0$, αλλιώς $f^2 = 1$. Σε αυτές τις τετριμμένες περιπτώσεις χρησιμοποιούμε μόνο έναν τετριμμένο τύπο, που δεν εξαρτάται από καμία υποσυνάρτηση, όπως μπορούμε να δούμε στον Πίνακα 7.4.

7.5 Ο αλγόριθμος ESCT-DC

Βασιζόμενοι στα προηγούμενα, περιγράφουμε στους Αλγόριθμους 11-13 τον αλγόριθμο ESCT-DC για την παραγωγή και την ελαχιστοποίηση εκφράσεων ESCT για τυχαίες μη πλήρως ορισμένες συναρτήσεις πολλών εξόδων. Ο αλγόριθμος ESCT-DC είναι ο αντίστοιχος του DCMIN για εκφράσεις ESOP και μπορεί να δουλέψει όπως αυτός με την κατάλληλη παράμετρο.

Algorithm 11. (ESCT-DC(input: MVMT formulation of f , flag))

```

Tree: generator tree of  $f$ ;
Expr( $f$ ): list of ESCT or ESOP expressions of  $f$ ;
begin
    Tree=Decomposition( $f$ , flag);
    Expr( $f$ )=Composition(Tree, flag);
    return Expr( $f$ );
end

```

Algorithm 12. (Decomposition(input: MVMT formulation of f , flag))

```

 $f_0, f_1, f_2$ : subfunctions of  $f$  in MVMT formulation;
CurNode: current node of ternary generator tree;

```

f_1 Node, f_0 Node, f_2 Node: left, middle and right child of CurNode respectively;

```

begin
  Assign  $f$  to CurNode;
  if ( $(f = 0)$ ) then
    return;
  endif
  if ( $(f = 1)$  AND ( $flag = ESCT$ )) then
    return;
  endif
  if ( $f$  is a function of 1 (multiple-valued) variable) then
    return;
  endif
  generate  $f_1$ ;
  generate  $f_0$ ;
  generate  $f_2$ ;
  CurNode→  $f_0$ Node = Decomposition( $f_0$ , flag);
  CurNode→  $f_1$ Node = Decomposition( $f_1$ , flag);
  CurNode→  $f_2$ Node = Decomposition( $f_2$ , flag);
end

```

Algorithm 13. (Composition(input: ternary generator tree of f , flag))

f_0, f_1, f_2 : subfunctions of f in MVMT formulation;
CurNode: current node of ternary generator tree;
 f_1 Node, f_0 Node, f_2 Node: left, middle and right child of CurNode respectively;
Expr(f): list of ESCT or ESOP expressions of f ;

```

begin
  \*  $f$  is the function assigned to CurNode *\
  if ( $flag = ESCT$ ) then
    if ( $(f_0 \neq 0$  AND  $f_0 \neq 1)$ ) then
       $f_0 = \text{CurNode} \rightarrow f_0\text{Node} \rightarrow f$ ;
      Expr( $f_0$ ) = Composition(CurNode→  $f_0$ Node, flag);
    endif
    if ( $(f_1 \neq 0$  AND  $f_1 \neq 1)$ ) then
       $f_1 = \text{CurNode} \rightarrow f_1\text{Node} \rightarrow f$ ;
      Expr( $f_1$ ) = Composition(CurNode→  $f_1$ Node, flag);
    endif
    if ( $(f_2 \neq 0$  AND  $f_2 \neq 1)$ ) then
       $f_2 = \text{CurNode} \rightarrow f_2\text{Node} \rightarrow f$ ;
      Expr( $f_2$ ) = Composition(CurNode→  $f_2$ Node, flag);
    endif
    if ( $((f_0 = 0$  AND  $f_1 = 1)$  OR  $(f_0 = 1$  AND  $f_1 = 0))$ ) then
      Expr( $f$ ) = expressions produced using formulas of Table 7.4;
    endif
    if ( $((f_0 = 0$  AND  $f_2 = 0)$  OR  $(f_1 = 0$  AND  $f_2 = 0)$  OR  $(f_0 = 1$  AND  $f_2 = 1)$  OR  $(f_1 = 1$  AND  $f_2 = 1)$  OR  $(f_0 = 0$  AND  $f_2 = 1)$  OR  $(f_0 = 1$ 

```

```

AND  $f_2 = 0$ ) OR ( $f_1 = 0$  AND  $f_2 = 1$ ) OR ( $f_1 = 1$  AND  $f_2 = 0$ )) then
    Expr( $f$ ) = expressions produced using formulas of Table 7.3;
endif
if (( $f_0 = 0$  OR  $f_1 = 0$  OR  $f_2 = 0$  OR  $f_0 = 1$  OR  $f_1 = 1$  OR
 $f_2 = 1$ )) then
    Expr( $f$ ) = expressions produced using formulas of Table 7.2
    and merging common and complement terms;
endif
if (( $f_0 \neq 0$  AND  $f_0 \neq 1$  AND  $f_1 \neq 0$  AND  $f_1 \neq 1$  AND  $f_2 \neq 0$  AND
 $f_2 \neq 1$ )) then
    Expr( $f$ ) = expressions produced using expansions from definitions
    11 and 37 and merging common and complement terms;
endif
endif
else
    if ( $f_0 \neq 0$ ) then
         $f_0 = \text{CurNode} \rightarrow f_0 \text{Node} \rightarrow f$ ;
        Expr( $f_0$ ) = Composition(CurNode  $\rightarrow$   $f_0$ Node, flag));
    endif
    if ( $f_1 \neq 0$ ) then
         $f_1 = \text{CurNode} \rightarrow f_1 \text{Node} \rightarrow f$ ;
        Expr( $f_1$ ) = Composition(CurNode  $\rightarrow$   $f_1$ Node, flag));
    endif
    if ( $f_2 \neq 0$ ) then
         $f_2 = \text{CurNode} \rightarrow f_2 \text{Node} \rightarrow f$ ;
        Expr( $f_2$ ) = Composition(CurNode  $\rightarrow$   $f_2$ Node, flag));
    endif
    if ( $f_0 \neq 0$  AND  $f_1 \neq 0$  AND  $f_2 \neq 0$ ) then
        Expr( $f$ ) = expressions produced using Shannon, positive and
        negative Davio expansions and merging common cubes;
    endif
    if (( $f_0 = 0$  AND  $f_1 = 0$ ) OR ( $f_0 = 0$  AND  $f_2 = 0$ ) OR
    ( $f_1 = 0$  AND  $f_2 = 0$ )) then
        Expr( $f$ ) = expressions produced using formulas of Table 3.4;
    endif
    if ( $f_0 = 0$  OR  $f_1 = 0$  OR  $f_2 = 0$ ) then
        Expr( $f$ ) = expressions produced using formulas of Table 3.3 and
        merging common cubes;
    endif
endelse
return Expr( $f$ );
end

```

Ο αλγόριθμος ESCT-DC υλοποιήθηκε σε C++. Χρησιμοποιήθηκε η MVMT αναπαράσταση για να αναπαραστήσουμε την αρχική συνάρτηση και κάθε υποσυνάρτηση. Όμως κάθε έκφραση ESCT αναπαριστάται σαν μια λίστα από *mv_term*,

που είναι περισσότερο συμπαγείς. Μια συνάρτηση με n δυαδικές μεταβλητές και μια μεταβλητή k τιμών χρειάζεται $2^{n+\lceil \lg k \rceil}$ bits για να αναπαρασταθεί σε MVMT αναπαράσταση. Ένας `mv_term` αυτής της συνάρτησης χρειάζεται $3n + k$ bits για να αναπαρασταθεί ($3n$ bits για τον `2v_term` και k bits για την `mv_var`). Το μειονέκτημα των `mv_term` είναι ότι μπορούν να αναπαραστήσουν μόνο συναρτήσεις υλοποιήσιμες ως cascade, αλλά όχι οποιαδήποτε τυχαία συνάρτηση, για την οποία μια λίστα από `mv_term` είναι απαραίτητη. Για έναν αρκετά μεγάλο αριθμό από μεταβλητές και έναν αρκετά μικρό αριθμό από σύνθετους όρους, η αναπαράσταση με `mv_term` είναι προτιμότερη από την MVMT αναπαράσταση. Επιπλέον, ο αλγόριθμός μας ξοδεύει ένα μεγάλο ποσοστό του χρόνου εκτέλεσής του συγκρίνοντας σύνθετους όρους, άρα η επιλογή της αναπαράστασης με `mv_term` είναι λογική.

Ο αλγόριθμος ESCT-DC ελέγχθηκε με ορισμένες μετροσυναρτήσεις, αλλά αποδείχθηκε επίσης χρονοβόρος όσον αφορά το χρόνο εκτέλεσης για έναν μεγάλο αριθμό μεταβλητών και αδιάφορων όρων. Επομένως, αλλάξαμε ελαφρά τον αλγόριθμο ώστε να κρατάμε λιγότερες εκφράσεις σε κάθε κόμβο του τριαδικού γεννήτορος δένδρου, όπως και στην περίπτωση του DCMIN. Έτσι, υπολογίζουμε το πολύ q εκφράσεις σε κάθε κόμβο. Αν μπορούν να παραχθούν περισσότερες εκφράσεις, τότε οι q εκφράσεις επιλέγονται τυχαία. Αυτός ο τροποποιημένος αλγόριθμος ονομάζεται SimpleESCT-DC σε αναλογία με τον SimpleDCMIN.

7.6 Ο αλγόριθμος QuickESCT-DC

Ένας καινούριος αλγόριθμος, βασισμένος στον ESCT-DC, σχεδιάστηκε και υλοποιήθηκε. Ο νέος ευριστικός αλγόριθμος ονομάστηκε QuickESCT-DC (ανάλογα με τον QuickDCMIN για εκφράσεις ESOP) και είναι πολύ γρηγορότερος από τον προηγούμενο. Ο QuickESCT-DC χρησιμοποιεί τον SimpleESCT-DC σαν ένα τελεστή μετασχηματισμού κύβων και ελαχιστοποίησης. Μπορούμε να δούμε μια περιγραφή του στον Αλγόριθμο 14.

Algorithm 14. (QuickESCT-DC(input: MVMT formulation of f , flag))

```

Tree: generator tree of  $f$ ;
Expr( $f$ ): an ESCT or ESOP expression of  $f$ 
on-trms( $f$ ): set of terms in the ON-set of  $f$ ;
dc-trms( $f$ ): set of terms in the DC-set of  $f$ ;
begin
    generate on-trms set (initial cover) using EXORCISM-4 or SimpleESCT-DC;
    generate dc-trms set from MVMT formulation;
    for  $i = 1$  to  $K$ 
        choose randomly  $N$  terms from on-trms;
        generate the function  $y$ , defined by these  $N$  terms and all dc-trms;
        Tree=Decomposition( $y$ , flag);
        Expr( $y$ )=Composition(Tree, flag); \* only  $q$  expressions per node produced *
    if (size of Expr( $y$ )  $\leq N$ ) then
        remove the selected  $N$  terms from on-trms set;
        insert terms of Expr( $y$ ) into on-trms set;

```

```

        endif
    endfor
    Expr(f) = expression produced from terms of on-trms;
    return Expr(f);
end

```

Ο QuickESCT-DC χρησιμοποιεί επίσης σαν είσοδο ένα αρχικό κάλυμμα που παράγεται είτε από τον SimpleESCT-DC ή από τον EXORCISM-4 [24], όπως ακριβώς περιγράφεται στην Ενότητα 3.6. Επίσης το σύνολο *CurrentTerms* ανα-νεώνεται σε κάθε επανάληψη με τον ίδιο τρόπο. Και εδώ, όλοι οι όροι του DC-set επιλέγονται σε κάθε επανάληψη, προσθέτοντας ένα σταθερό αριθμό όρων από το σύνολο *CurrentTerms*. Τέλος, ο τελεστής μετασχηματισμού είναι βασισμένος στον SimpleESCT-DC, όπως ο τελεστής μετασχηματισμού του QuickDCMIN βασίζεται στον SimpleDCMIN.

Θυμίζουμε ότι ο αλγόριθμος QuickESCT-DC μπορεί να χρησιμοποιηθεί και για εκφράσεις ESCT και για εκφράσεις ESOP. Όταν ασχολούμαστε με εκφράσεις ESOP, μόνο οι κυτταρικοί δείκτες 3, 4 και 6 επιτρέπονται.

7.7 Πειραματικά αποτελέσματα

Ο αλγόριθμος QuickESCT-DC υλοποιήθηκε σε C++ και δοκιμάστηκε σε έναν υπολογιστή Core 2 Quad 2,4GHz με 4GB RAM που έτρεχε Linux. Όπως αναφέρεται στην Ενότητα 7.6, η τυχαία επιλογή των όρων βασίζεται σε μια γεννήτρια τυχαίων αριθμών. Επίσης, όταν χρειαζόμαστε το συμπλήρωμα μιας έκφρασης, επιλέγουμε τυχαία τον όρο του οποίου το συμπλήρωμα θα υπολογίσουμε. Ακόμα όταν μπορούν να παραχθούν πολλές εκφράσεις σε κάθε κόμβο επιλέγουμε τυχαία ποιες θα κρατήσουμε. Σαν συνέπεια, κάθε εκτέλεση του προγράμματος παράγει διαφορετικά αποτελέσματα. Επίσης μπορεί να τρέξει με δύο τρόπους: είτε με ένα δεδομένο αριθμό επαναλήψεων, είτε με ένα αριθμό επαναλήψεων χωρίς μείωση του βάρους της συνάρτησης πριν σταματήσει. Οι παράμετροι του προγράμματος, σε αναλογία με τον QuickDCMIN, είναι οι ακόλουθες: α) ο τρόπος λειτουργίας του προγράμματος, β) ο αριθμός των επαναλήψεων που ερμηνεύεται ανάλογα με την πρώτη παράμετρο, γ) ο αριθμός των όρων από το σύνολο *CurrentTerms* που επιλέγονται σε κάθε επανάληψη, δ) ο αριθμός των εκφράσεων που παράγονται για κάθε κόμβο του γεννητορος δένδρου όταν εφαρμόζουμε τον μετασχηματισμό που βασίζεται στον SimpleESCT-DC και ε) μια σημαία (flag) που υποδεικνύει αν θα παραχθούν εκφράσεις ESOP ή ESCT.

Το πρόγραμμα δοκιμάστηκε στον πρώτο τρόπο λειτουργίας με ένα σύνολο από πρότυπες μετροσυναρτήσεις για 100000 επαναλήψεις και 5 – 10 επιλεγόμενους όρους ανά επανάληψη. Ο αριθμός των παραγόμενων εκφράσεων για κάθε υποσυνάρτηση σε κάθε επανάληψη τέθηκε σε 1 για να επιτρέψει περισσότερες επαναλήψεις σε λιγότερο χρόνο. Για κάθε σύνολο παραμέτρων τρέξαμε το πρόγραμμα αρκετές φορές και επιλέξαμε την καλύτερη περίπτωση.

Εναλλακτικά, το πρόγραμμα δοκιμάστηκε στο δεύτερο τρόπο λειτουργίας με ένα σύνολο από πρότυπες μετροσυναρτήσεις για 5000 επαναλήψεις χωρίς μείωση

του μεγέθους των παραγόμενων εκφράσεων και για 5 – 10 επιλεγμένους όρους ανά επανάληψη. Ο αριθμός των παραγόμενων εκφράσεων για κάθε υποσυνάρτηση σε κάθε επανάληψη τέθηκε σε 1 επίσης και επιλέξαμε την καλύτερη περίπτωση έπειτα από διαφορετικές εκτελέσεις.

Καταρχήν, για να δείξουμε την ανωτερότητα της προσέγγισής μας επειδή εκμεταλλευόμαστε την παρουσία των αδιάφορων όρων, χρησιμοποιήσαμε τα προαναφερόμενα 0-κάλυμμα και 1-κάλυμμα. Εκτελέσαμε αλγόριθμους για πλήρως ορισμένες συναρτήσεις για αυτές τις δύο τροποποιήσεις των προτύπων μετροσυναρτήσεων, καθώς και τους αλγόριθμους μας για τις αρχικές μη πλήρως ορισμένες μετροσυναρτήσεις. Συγκεκριμένα, χρησιμοποιήσαμε τους αλγόριθμους Min2 [101] και QuickESCT-DC για τις πλήρως ορισμένες τροποποιήσεις των μετροσυναρτήσεων και τον QuickESCT-DC για τις πρότυπες μετροσυναρτήσεις, όπως φαίνεται στον Πίνακα 7.5 με τον πρώτο τρόπο λειτουργίας, και στον Πίνακα 7.6 με το δεύτερο τρόπο λειτουργίας.

Στην περίπτωση της 3ης στήλης των πινάκων (QuickESCT-DC $dc=0$) χρησιμοποιήσαμε δύο διαφορετικές εναλλακτικές ως αρχικό κάλυμμα: το πρώτο παρήχθη από τον EXORCISM-4 με το 0-κάλυμμα ως είσοδο, και το δεύτερο παρήχθη από τον SimpleESCT-DC με το 0-κάλυμμα ως είσοδο (σε αυτήν την περίπτωση ο SimpleESCT-DC προσπάθησε να ελαχιστοποιήσει μια πλήρως ορισμένη συνάρτηση). Θέλαμε να ελαχιστοποιήσουμε το 0-κάλυμμα και διαλέξαμε την καλύτερη περίπτωση, χρησιμοποιώντας τα δύο διαφορετικά αρχικά καλύμματα, για κάθε μετροσυνάρτηση.

Ομοίως, στην περίπτωση της 4ης στήλης των πινάκων (QuickESCT-DC $dc=1$) χρησιμοποιήσαμε δύο εναλλακτικές για το αρχικό κάλυμμα, επίσης: το πρώτο παρήχθη από τον EXORCISM-4 με το 1-κάλυμμα ως είσοδο, και το δεύτερο παρήχθη από τον SimpleESCT-DC με το 1-κάλυμμα ως είσοδο. Θέλαμε να ελαχιστοποιήσουμε το 1-κάλυμμα και διαλέξαμε την καλύτερη περίπτωση, χρησιμοποιώντας τα δύο διαφορετικά αρχικά καλύμματα.

Τέλος, στην περίπτωση της 5ης στήλης των πινάκων (QuickESCT-DC με dc) χρησιμοποιήσαμε τρεις εναλλακτικές ως αρχικό κάλυμμα: δύο αρχικά καλύμματα παρήχθησαν από τον EXORCISM-4 με το 0-κάλυμμα και το 1-κάλυμμα ως είσοδο αντίστοιχα και το τελευταίο παρήχθη από τον SimpleESCT-DC με την αρχική μη πλήρως ορισμένη μετροσυνάρτηση ως είσοδο. Ο σκοπός μας ήταν η ελαχιστοποίηση της μη πλήρως ορισμένης συνάρτησης και, προφανώς, διαλέξαμε την καλύτερη περίπτωση χρησιμοποιώντας τα τρία διαφορετικά αρχικά καλύμματα, επίσης.

Η ύπαρξη των αδιάφορων όρων επιτρέπει στον QuickESCT-DC να ελαχιστοποιήσει τις μετροσυναρτήσεις περισσότερο από τις πλήρως ορισμένες εναλλακτικές τους (0-κάλυμμα και 1-κάλυμμα). Αναφερόμενοι στις πλήρως ορισμένες τροποποιήσεις των μετροσυναρτήσεων (όταν $dc = 0$ και $dc = 1$) τα αποτελέσματά μας είναι καλύτερα από τα αποτελέσματα του Min2, επειδή ξεκινούμε με καλύτερα αρχικά καλύμματα, ενώ ο Min2 ξεκινά με την MVMT αναπαράσταση. Στην περίπτωση που ο Min2 θα μπορούσε να πάρει ως είσοδο τα ίδια αρχικά καλύμματα θα μπορούσε πιθανόν να παράγει καλύτερα αποτελέσματα.

Επειδή οι παραπάνω εναλλακτικές (όταν $dc = 0$ και $dc = 1$) φαίνονται απλοϊκές, ένας περισσότερο εξεζητημένος αλγόριθμος υλοποιήθηκε για να ελέγξουμε

Συνάρτηση	Min2	Min2	Quick-ESCT-DC		Quick-ESCT-DC		Quick-ESCT-DC	
	dc=0	dc=1	dc=0	dc=1	dc=0	dc=1	με dc	
	Μέγεθος	Μέγεθος	Μέγεθος	Χρόνος(s)	Μέγεθος	Χρόνος(s)	Μέγεθος	Χρόνος(s)
alu2	68	160	67	0	96	2.84	40	1992.84
alu3	47	195	48	0	88	0.5	39	712.94
apla	55	67	45	0	45	2.73	33	331.24
b11	35	36	26	0	26	0	26	0
bench	30	45	27	27.11	39	511.98	15	19.00
bw	64	74	22	0	22	0	22	0
dk17	37	44	30	0	30	0	20	63.86
dk27	16	27	14	0	14	0	10	0.34
ex1010	1070	1141	633	0	699	13429.13	375	18968.94
fout	74	67	47	1150.00	46	1389.88	33	1316.76
inc	29	33	27	0	26	0	23	1030.83
p1	81	117	57	0	55	0.42	42	21749.21
p3	51	57	37	0	37	3221.29	28	124.44

Πίνακας 7.5: Εκμετάλλευση των αδιάφορων όρων για εκφράσεις ESCT (1ος τρόπος λειτουργίας)

την επίδοση του αλγορίθμου μας, ως ακολούθως: Ξεκινώντας από μια μη πλήρως ορισμένη συνάρτηση πολλών εξόδων, δημιουργούμε το τριαδικό της γεννήτωρ δένδρο (σύμφωνα με την μεθοδολογία που περιγράψαμε στην εργασία). Σε κάθε κόμβο του δένδρου ελέγχουμε αν η συνάρτησή του μπορεί να μετατραπεί σε μια σταθερά μηδέν ή σταθερά ένα, λαμβάνοντας υπ' όψιν και το ON-set και το DC-set της. Αν γίνεται, τότε την μετατρέπουμε σε μια σταθερή συνάρτηση (προσπαθούμε να δημιουργήσουμε όσο περισσότερες συναρτήσεις μηδέν μπορούμε και έπειτα όσο περισσότερες συναρτήσεις ένα) θέτοντας κατάλληλες τιμές στους ελαχιστόρους του DC-set της, και έπειτα ενημερώνοντας, ανάλογα, όλες τις συναρτήσεις του γεννήτορος δένδρου (θυμηθείτε ότι στο γεννήτωρ δένδρο, οι συναρτήσεις συνδέονται μεταξύ τους, αφού για παράδειγμα $f^2 = f^1 \oplus f^0$). Όταν φθάσουμε στα κάτω φύλλα του γεννήτορος δένδρου, θέτουμε οποιουδήποτε εναπομείναντες ελαχιστόρους σε 0 και δημιουργούμε μια πλήρως ορισμένη Boolean συνάρτηση. Η διαίσθηση πίσω από αυτήν την προσέγγιση είναι ότι οι συναρτήσεις με πολλές σταθερές μηδέν ή ένα υποσυναρτήσεις στο γεννήτωρ δένδρο τους συνήθως καταλήγουν σε μικρότερες ελάχιστες υλοποιήσεις. Αυτή η συνάρτηση τότε ελαχιστοποιείται χρησιμοποιώντας τον αλγόριθμο Min2-EW7Min [100].

Ο αλγόριθμος QuickESCT-DC, συγκρινόμενος με την προηγούμενη προσέγγιση, απέδωσε μια βελτιστοποίηση 55,5% στα αποτελέσματα. Έτσι, πιστεύουμε ότι ο QuickESCT-DC παρουσιάζει πολύ καλύτερη επίδοση ακόμα και από αυτήν την προσέγγιση, όσον αφορά το μέγεθος των παραγόμενων εκφράσεων.

Στους Πίνακες 7.7 (πρώτος τρόπος λειτουργίας) και 7.8 (δεύτερος τρόπος λειτουργίας) συγκρίναμε τον αλγόριθμο QuickESCT-DC με τον αλγόριθμο των Lee, Drechsler και Perkowski [37, 38]. Σημειώνεται ότι οι χρόνοι αφορούν διαφορετικά υπολογιστικά συστήματα. Ο τελευταίος αλγόριθμος προτείνει μια γενικότερη αρχιτεκτονική, όπου επιτρέπονται και ο τελεστής OR και ο τελεστής XOR ανάμεσα στους όρους Maitra. Παρά τη γενικότητα αυτού του αλγορίθμου, ο QuickESCT-

Συνάρτηση	Min2	Min2	Quick-ESCT-DC		Quick-ESCT-DC		Quick-ESCT-DC	
	dc=0	dc=1	dc=0	dc=1	dc=0	dc=1	με dc	με dc
	Μέγεθος	Μέγεθος	Μέγεθος	Χρόνος(s)	Μέγεθος	Χρόνος(s)	Μέγεθος	Χρόνος(s)
alu2	68	160	67	792.47	96	901.76	41	2000.45
alu3	47	195	48	774.84	88	894.28	40	435.84
apla	55	67	45	867.72	45	905.33	37	69.99
b11	35	36	26	191.27	26	196.01	26	230.93
bench	30	45	27	65.53	40	245.11	15	26.85
bw	64	74	22	18.94	22	18.94	22	91514.51
dk17	37	44	30	821.69	30	859.43	22	19.14
dk27	16	27	14	392.95	14	398.14	10	12.12
ex1010	1070	1141	633	910.06	700	906.52	380	335.49
fout	74	67	52	97.24	52	55.71	35	229.98
inc	29	33	27	103.48	26	107.82	24	280.93
p1	81	117	57	234.65	54	406.9	43	9245.93
p3	51	57	37	228.2	38	379.65	29	316.25

Πίνακας 7.6: Εχμετάλλευση των αδιάφορων όρων για εκφράσεις ESCT (2ος τρόπος λειτουργίας)

DC, ενώ είναι πιο αργός, δίνει καλύτερα πειραματικά αποτελέσματα για όλες τις πρότυπες μετροσυναρτήσεις που χρησιμοποιήσαμε.

Μια ενδιαφέρουσα προοπτική είναι η σύγκριση με κυτταρικές αρχιτεκτονικές Maitra όπου οι σύνθετοι όροι συλλέγονται από τον τελεστή OR, και όχι από τον τελεστή XOR. Δοκιμάζοντας έναν αριθμό από συναρτήσεις με μια μέθοδο σύνθεσης πινάκων Maitra με σειρά συλλεκτών OR (OR collector row) [51] και με τον SimpleESCT-DC, επιτύχαμε μια βελτίωση 23,8% στην περίπτωση των ESCT (πίνακες Maitra με σειρά συλλεκτών XOR - XOR collector row).

Ο αλγόριθμός μας αφορά μη πλήρως ορισμένες συναρτήσεις. Αφού ξεκινά με ένα αρχικό κάλυμμα χρησιμοποιώντας οποιονδήποτε αλγόριθμο για πλήρως ορισμένες συναρτήσεις, δεν έχει κανένα νόημα να συγκρίνεται με αλγόριθμους για πλήρως ορισμένες συναρτήσεις και δεν ισχυριζόμαστε ότι είναι ένας καλύτερος αλγόριθμος ελαχιστοποίησης για αυτή την κατηγορία συναρτήσεων.

Αυτός ο αλγόριθμος φαίνεται να απαιτεί μη αμελητέο χρόνο σε κάποιες περιπτώσεις (π.χ. για τη συνάρτηση bw), που εξαρτάται από τον αριθμό των επαναλήψεων (χωρίς μείωση στο μέγεθος της συνάρτησης πριν σταματήσει το πρόγραμμα). Όμως, μπορούμε να ορίσουμε αυτή την παράμετρο ώστε να επιτύχουμε ένα συμβιβασμό ανάμεσα στο χρόνο εκτέλεσης και την ποιότητα των αποτελεσμάτων. Μια μικρή μείωση στην ποιότητα των αποτελεσμάτων (διατηρώντας τα καλύτερα από αυτά που παίρνουμε από άλλες μεθόδους στη βιβλιογραφία) μειώνει δραστικά το χρόνο εκτέλεσης. (Για παράδειγμα, τρέξαμε την μετροσυνάρτηση alu2, θέτοντας την προαναφερόμενη παράμετρο σε 1000 επαναλήψεις. Μια έκφραση με 46 όρους παρήχθη σε 134.79 δευτερόλεπτα, αντί για 41 όρους σε 2000.45 δευτερόλεπτα με 5000 επαναλήψεις.) Επιπλέον, αυτή η τεχνική επιλύει το πρόβλημα της κλίμακας.

Επιλέξαμε, στο δεύτερο τρόπο λειτουργίας, να κρατήσουμε την παράμετρο των επαναλήψεων σε μια σχετικά μεγάλη τιμή (5000 επαναλήψεις) για όλα τα παραδείγματα, ώστε να επιτύχουμε μικρότερα μεγέθη για τις εκφράσεις. Ειδικά, για

Συνάρτηση	Lee, Drechsler, Perkowski	QuickESCT-DC
alu2	66	40
alu3	48	39
apla	59	33
b11	33	26
bench	-	15
bw	65	22
dk17	36	20
dk27	15	10
ex1010	-	375
fout	-	33
inc	36	23
p1	-	42
p3	-	28

Πίνακας 7.7: Σύγκριση του QuickESCT-DC (για εκφράσεις ESCT) με τον αλγόριθμο των Lee, Drechsler, Perkowski (1ος τρόπος λειτουργίας)

Συνάρτηση	Lee, Drechsler, Perkowski		QuickESCT-DC	
	Μέγεθος	Χρόνος(s)	Μέγεθος	Χρόνος(s)
alu2	66	1.93	41	2000.45
alu3	48	0.82	40	435.84
apla	59	0.99	37	69.99
b11	33	0.27	26	230.93
bench	-	-	15	26.85
bw	65	0.51	22	91514.51
dk17	36	0.74	22	19.14
dk27	15	0.71	10	12.12
ex1010	-	-	380	335.49
fout	-	-	35	229.98
inc	36	0.29	24	280.93
p1	-	-	43	9245.93
p3	-	-	29	316.25

Πίνακας 7.8: Σύγκριση του QuickESCT-DC (για εκφράσεις ESCT) με τον αλγόριθμο των Lee, Drechsler, Perkowski (2ος τρόπος λειτουργίας)

τη συνάρτηση bw υποθέτουμε ότι δεν μπορεί να απλοποιηθεί περισσότερο από το αρχικό της κάλυμμα (με μέγεθος 22). Επομένως, δεν ήταν απαραίτητο να έχουμε 5000 επαναλήψεις που απαιτούν τόσο πολύ χρόνο εκτέλεσης.

Κεφάλαιο 8

Κβαντικοί Υπολογιστές

8.1 Στοιχεία άλγεβρας

8.1.1 Ορθομοναδιαίοι πίνακες

Ορισμός 43 Έστω A ένας πίνακας διαστάσεων $n \times m$. Ο πίνακας B διαστάσεων $m \times n$ είναι ο ερμιτιανός συζυγής του A και συμβολίζεται ως A^+ , αν για όλα τα στοιχεία $a_{ij}, b_{ji} \in \mathbb{C}$ των πινάκων A και B αντίστοιχα, με $1 \leq i \leq n$ και $1 \leq j \leq m$ ισχύει: $b_{ji} = a_{ij}^*$.

Δηλαδή ο πίνακας B είναι ο ερμιτιανός συζυγής του A αν οι στήλες του A είναι γραμμές του B και τα στοιχεία του B είναι τα συζυγή μιγαδικά των αντίστοιχων στοιχείων του A .

Παράδειγμα 39 Έστω $A = \begin{bmatrix} 2 & i & 3+i \\ 5-i & -i & 0 \end{bmatrix}$. Τότε ο ερμιτιανός συζυγής πίνακας B του A είναι ο $B = A^+ = \begin{bmatrix} 2 & 5+i \\ -i & i \\ 3-i & 0 \end{bmatrix}$.

Ορισμός 44 Ένας πίνακας A είναι ορθομοναδιαίος αν ο ερμιτιανός συζυγής του είναι ίσος με τον αντίστροφό του, δηλαδή αν ισχύει: $AA^+ = A^+A = I$.

8.1.2 Χώρος Hilbert

Ορισμός 45 Ο χώρος Hilbert \mathcal{H} είναι ένας διανυσματικός χώρος πάνω στο πεδίο των μιγαδικών αριθμών \mathbb{C} , δηλαδή ένας διανυσματικός μιγαδικός χώρος. Τα στοιχεία που αποτελούν τον χώρο Hilbert είναι ένα σύνολο διανυσμάτων $\{|x_n\rangle\}$ με τις παρακάτω ιδιότητες:

1. Για κάθε $|a\rangle, |b\rangle \in \mathcal{H}$ υπάρχει ένα μοναδικό $|c\rangle \in \mathcal{H}$ το οποίο ονομάζεται άθροισμα των $|a\rangle$ και $|b\rangle$ και συμβολίζεται ως $|c\rangle = |a\rangle + |b\rangle$.
2. Για κάθε $|a\rangle, |b\rangle \in \mathcal{H}$ ισχύει $|a\rangle + |b\rangle = |b\rangle + |a\rangle$.

3. Για κάθε $|a\rangle, |b\rangle, |c\rangle \in \mathcal{H}$ ισχύει $(|a\rangle + |b\rangle) + |c\rangle = |a\rangle + (|b\rangle + |c\rangle)$.
4. Υπάρχει το μηδενικό διάνυσμα $|0\rangle \in \mathcal{H}$ με την ιδιότητα $|a\rangle + |0\rangle = |a\rangle$ για κάθε $|a\rangle \in \mathcal{H}$.
5. Για κάθε $|a\rangle \in \mathcal{H}$ υπάρχει μοναδικό $|-a\rangle \in \mathcal{H}$, που ονομάζεται αντίστροφο διάνυσμα του $|a\rangle$, ώστε να ισχύει $|a\rangle + |-a\rangle = |0\rangle$.
6. Για κάθε $|a\rangle, |b\rangle \in \mathcal{H}$ και για κάθε $\mu \in \mathcal{C}$ ισχύει $\mu(|a\rangle + |b\rangle) = \mu|a\rangle + \mu|b\rangle$.
7. Για κάθε $|a\rangle \in \mathcal{H}$ και για κάθε $\mu, \nu \in \mathcal{C}$ ισχύει $(\mu + \nu)|a\rangle = \mu|a\rangle + \nu|a\rangle$.
8. Για κάθε $|a\rangle \in \mathcal{H}$ και για κάθε $\mu, \nu \in \mathcal{C}$ ισχύει $\mu\nu|a\rangle = \mu(\nu|a\rangle)$.
9. Για κάθε $|a\rangle \in \mathcal{H}$ ισχύει $1|a\rangle = |a\rangle$.

Ορισμός 46 Για οποιαδήποτε διανύσματα $|a\rangle, |b\rangle \in \mathcal{H}$ ορίζεται το εσωτερικό τους γινόμενο στο χώρο Hilbert και συμβολίζεται ως $\langle a|b\rangle = (|a\rangle, |b\rangle)$. Το εσωτερικό γινόμενο είναι ένας μιγαδικός αριθμός και έχει τις εξής ιδιότητες:

1. Για κάθε $|a\rangle, |b\rangle \in \mathcal{H}$ και για κάθε $\lambda \in \mathcal{C}$ ισχύει $\langle \lambda a|b\rangle = \lambda \langle a|b\rangle$.
2. Για κάθε $|a\rangle, |b\rangle \in \mathcal{H}$ και για κάθε $\lambda \in \mathcal{C}$ ισχύει $\langle \lambda a|a\rangle = \lambda^* \langle a|a\rangle$.
3. Για κάθε $|a\rangle, |b\rangle, |c\rangle \in \mathcal{H}$ ισχύει $\langle a|(|b\rangle + |c\rangle) = \langle a|b\rangle + \langle a|c\rangle = \langle a|b\rangle + \langle a|c\rangle$.
4. Για κάθε $|a\rangle, |b\rangle, |c\rangle \in \mathcal{H}$ ισχύει $\langle a|(|b\rangle + |c\rangle) = \langle a|b\rangle + \langle a|c\rangle = \langle a|b\rangle + \langle a|c\rangle$.
5. Για κάθε $|a\rangle, |b\rangle \in \mathcal{H}$ ισχύει $\langle a|b\rangle = \langle b|a\rangle^*$.

Ορισμός 47 Το μέτρο (norm) ενός διανύσματος $|a\rangle \in \mathcal{H}$ ορίζεται ως $\|a\| = \sqrt{\langle a|a\rangle}$.

Ορισμός 48 Ένα σύνολο διανυσμάτων $\{|q_n\rangle\}$ στο χώρο Hilbert αποτελεί ένα ορθοκανονικό σύστημα διανυσμάτων (orthonormal system) αν ισχύει $\langle q_n|q_m\rangle = \delta_{nm} = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases}$.

Ορισμός 49 Ένα ορθοκανονικό σύστημα διανυσμάτων $\{|q_n\rangle\}$ στο χώρο Hilbert είναι πλήρες, όταν κάθε τυχαίο διάνυσμα $|\psi\rangle$ του χώρου μπορεί να εκφραστεί ως γραμμικός συνδυασμός των διανυσμάτων αυτών, δηλαδή $|\psi\rangle = \sum_n a_n |q_n\rangle$, όπου $a_n \in \mathcal{C}$.

Ένα πλήρες ορθοκανονικό σύστημα n διανυσμάτων μπορεί να αποτελέσει τη βάση ενός χώρου Hilbert n διαστάσεων.

Ορισμός 50 Έστω $|\psi\rangle, |\phi\rangle \in \mathcal{H}$ και \hat{A}, \hat{B} τελεστές του χώρου Hilbert. Ισχύουν οι εξής ιδιότητες:

1. $\hat{A}(a|\psi\rangle + b|\phi\rangle) = a\hat{A}|\psi\rangle + b\hat{A}|\phi\rangle$
2. $(\hat{A} + \hat{B})|\psi\rangle = \hat{A}|\psi\rangle + \hat{B}|\psi\rangle$
3. $(\hat{A} \cdot \hat{B})|\psi\rangle = \hat{A} \cdot (\hat{B}|\psi\rangle)$
4. Υπάρχει ο μοναδιαίος τελεστής \hat{I} ώστε $\hat{I}|\phi\rangle = |\phi\rangle$.
5. Υπάρχει ο μηδενικός τελεστής $\hat{0}$ ώστε $\hat{0}|\phi\rangle = 0$.
6. Γενικά ισχύει $\hat{A} \cdot \hat{B} \neq \hat{B} \cdot \hat{A}$.

8.2 Βασικές έννοιες κβαντικών υπολογισμών

Σε ένα κλασικό υπολογιστικό σύστημα, η βασική μονάδα πληροφορίας είναι το bit. Ένα bit μπορεί να πάρει δύο διαφορετικές τιμές: 0 ή 1.

Στους κβαντικούς υπολογιστές (και στους αντίστοιχους κβαντικούς υπολογισμούς) η βασική μονάδα πληροφορίας είναι το qubit. Το qubit έχει (όπως και το κλασικό bit) δύο βασικές καταστάσεις που συμβολίζονται με $|0\rangle$ και $|1\rangle$. Οι καταστάσεις αυτές αντιστοιχούν στις καταστάσεις 0 και 1 του κλασικού bit. Ένα qubit όμως μπορεί να βρίσκεται και ταυτόχρονα στις δύο αυτές καταστάσεις. Αυτή είναι και η θεμελιώδης διαφορά του qubit με το bit.

Γενικότερα ένα κβαντικό σύστημα που απεικονίζεται από ένα qubit μπορεί να βρίσκεται σε άπειρες διαφορετικές καταστάσεις (παρόλο που οι βασικές καταστάσεις είναι μόνο δύο). Σύμφωνα με τις αρχές της κβαντομηχανικής, η κατάσταση ενός κβαντικού φυσικού συστήματος δύο καταστάσεων παριστάνεται με ένα διάνυσμα σε ένα ιδιαίτερο διανυσματικό χώρο που ονομάζεται χώρος Hilbert και είναι ένας μιγαδικός διανυσματικός χώρος. Η κατάσταση ενός κβαντικού συστήματος περιγράφεται από ένα διάνυσμα κατάστασης της μορφής: $|R\rangle = a|0\rangle + b|1\rangle$. Οι συντελεστές a, b ορίζουν πλάτη πιθανότητας και το τετράγωνο του κάθε συντελεστή δίνει την πιθανότητα το qubit που απεικονίζει το παραπάνω κβαντικό σύστημα να είναι στην κατάσταση $|0\rangle$ και $|1\rangle$ αντίστοιχα. Προφανώς πρέπει να ισχύει: $a^2 + b^2 = 1$.

Πολλές φορές είναι εύκολο να απεικονίσουμε τα qubits με μορφή πινάκων. Έτσι το παραπάνω qubit θα γραφτεί: $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ και άρα: $|R\rangle = a|0\rangle + b|1\rangle = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$. Γενικά, όπως και θα δούμε και στη συνέχεια, η χρήση των πινάκων διευκολύνει ιδιαίτερα τους υπολογισμούς και θα χρησιμοποιηθεί εκτενώς.

Τα παραπάνω συνοψίζονται στον επόμενο ορισμό.

Ορισμός 51 Έστω ότι θεωρούμε ότι $\{|0\rangle, |1\rangle\}$ είναι η ορθοκανονική βάση του διδιάστατου χώρου Hilbert. Ένα qubit ($|q\rangle$) είναι ένα κανονικοποιημένο διάνυσμα στο χώρο Hilbert δύο διαστάσεων, και

$$|q\rangle = a|0\rangle + b|1\rangle = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

όπου $|a|^2 + |b|^2 = 1$.

Από τα παραπάνω είναι προφανές ότι ένα κβαντικό σύστημα δεν είναι ντετερμινιστικό. Δηλαδή δεν μπορούμε με βεβαιότητα (στις περισσότερες περιπτώσεις) να προβλέψουμε σε ποια πραγματική κατάσταση βρίσκεται το σύστημα. Το κβαντικό σύστημα θα καταλήξει σε μια από τις βασικές του καταστάσεις όταν επιχειρήσουμε να μετρήσουμε την κατάσταση του (όταν δηλαδή θα προσπαθήσουμε να επέμβουμε σε αυτό).

Συνοψίζοντας τα προηγούμενα, ένα κβαντικό σύστημα που εκφράζεται από ένα qubit μπορεί να βρίσκεται ταυτόχρονα και στις δύο βασικές του καταστάσεις. Η πιθανότητα να βρίσκεται στην κατάσταση $|0\rangle$ (δηλαδή όταν θα κάνουμε τη μέτρηση να το βρούμε στην κατάσταση $|0\rangle$) είναι $|a|^2$. Αντίστοιχα η πιθανότητα να βρίσκεται στην κατάσταση $|1\rangle$ (δηλαδή όταν θα κάνουμε τη μέτρηση να το βρούμε στην κατάσταση $|1\rangle$) είναι $|b|^2$. Όταν $|a|^2, |b|^2 \neq 0$ τότε το κβαντικό μας σύστημα λέμε ότι βρίσκεται σε κατάσταση υπέρθεσης (superposition state). Όταν το σύστημα δεν βρίσκεται σε κατάσταση υπέρθεσης τότε μπορούμε με απόλυτη βεβαιότητα να προβλέψουμε σε ποια από τις δύο βασικές του καταστάσεις βρίσκεται.

Εν γένει οι συντελεστές a, b είναι μιγαδικοί αριθμοί για το λόγο αυτό ένα διάνυσμα κατάστασης απεικονίζεται σε μια σφαίρα Bloch. Αν θεωρήσουμε ένα qubit της μορφής: $|R\rangle = e^{i\gamma}[\cos \frac{\theta}{2}|0\rangle + e^{i\phi} \sin \frac{\theta}{2}|1\rangle] = a|0\rangle + b|1\rangle$ τότε η απεικόνισή του σε μια σφαίρα Bloch (έχει ακτίνα ίση με 1) φαίνεται στην Εικόνα 8.1. Στην απεικόνιση αυτή $e^{i\gamma}$ είναι μια γενική διαφορά φάσης η οποία δεν είναι παρατηρήσιμη και συνήθως την αγνοούμε.

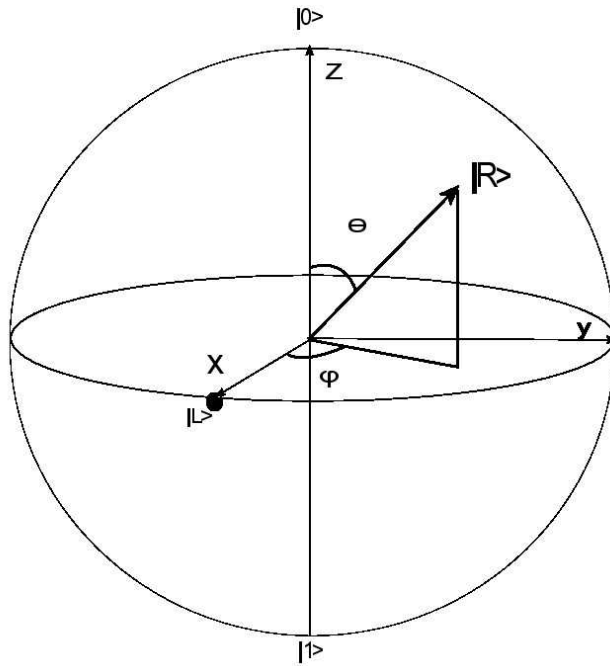
Πολλές φορές όμως, για λόγους απλότητας, χρησιμοποιούμε διδιάστατη απεικόνιση όπως στην Εικόνα 8.2. Στην περίπτωση αυτή οι συντελεστές a, b θεωρούνται ως πραγματικοί αριθμοί. Άλλωστε αρκετοί από τους κβαντικούς τελεστές (κβαντικές πύλες) δεν ενεργούν καθόλου στη γωνία φάσης ϕ , οπότε μπορούμε, συνήθως, να την αγνοούμε.

Βεβαίως τα παραπάνω μπορούν να γενικευθούν για κβαντικά συστήματα n βασικών καταστάσεων. Στην περίπτωση αυτή η κατάσταση ενός τέτοιου συστήματος θα ορίζεται ως: $|R\rangle = a_1|T_1\rangle + a_2|T_2\rangle + \dots + a_n|T_n\rangle$ όπου οι καταστάσεις $|T_i\rangle$ για $i = 1, \dots, n$ ορίζουν ορθομοναδιαίο σύστημα και είναι ανεξάρτητες μεταξύ τους. Επιπλέον θα ισχύει: $|a_1|^2 + |a_2|^2 + \dots + |a_n|^2 = 1$. Ο χώρος Hilbert που απαιτείται για να απεικονιστεί ένα τέτοιο σύστημα είναι n διαστάσεων.

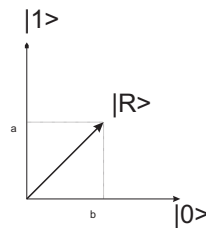
Όπως και στα κλασικά υπολογιστικά συστήματα, έτσι και εδώ, μπορούμε να ορίσουμε κβαντικούς καταχωρητές οι οποίοι μπορούν να αποτελούνται από n qubits. Ένας κβαντικός καταχωρητής (όπως και ένας κλασικός) μπορεί να απεικονίσει 2^n διαφορετικούς αριθμούς. Όμως ένας κβαντικός καταχωρητής που βρίσκεται σε υπέρθεση μπορεί να απεικονίσει όλους αυτούς τους αριθμούς ταυτόχρονα.

Αντίστοιχα με την υπέρθεση, υπάρχει και ένα άλλο ενδιαφέρον φαινόμενο, το φαινόμενο της κβαντικής διεμπλοκής (entanglement). Το φαινόμενο αυτό μπορεί να εμφανιστεί σε ένα σύστημα που αποτελείται από περισσότερα από 1 qubits. Δύο qubits που βρίσκονται σε κβαντική διεμπλοκή επηρεάζουν άμεσα το ένα την κβαντική κατάσταση του άλλου. Το παραπάνω θα φανεί καλύτερα με το επόμενο παράδειγμα.

Παράδειγμα 40 Έστω ένας κβαντικός καταχωρητής που αποτελείται από δύο qubits. Προφανώς υπάρχουν τέσσερις βασικές καταστάσεις για τον παραπάνω κατα-



Σχήμα 8.1: Η κατάσταση ενός τυχαίου qubit $|R\rangle$ απεικονιζόμενη σε μια σφαίρα Bloch. θ είναι η γωνία με τον άξονα z και ϕ η γωνία της προβολής του $|R\rangle$ στο επίπεδο xy με τον άξονα x . Το $|L\rangle$ είναι το qubit: $(|0\rangle + |1\rangle)/2^{1/2}$.



Σχήμα 8.2: Διδιάστατη απεικόνιση ενός qubit

χωρητή: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Οι καταστάσεις αυτές αντιστοιχούν σε κάθε δυνατό συνδυασμό των βασικών καταστάσεων των δύο qubits του καταχωρητή. Έστω ότι ο καταχωρητής αυτός έχει το παρακάτω διάνυσμα κατάστασης: $\frac{\sqrt{2}}{2}|00\rangle + \frac{\sqrt{2}}{2}|11\rangle$. Αυτό σημαίνει ότι ο παραπάνω καταχωρητής μπορεί να βρίσκεται στις καταστάσεις $|00\rangle, |11\rangle$ με ίδια πιθανότητα 50%. Στην περίπτωση αυτή τα δύο qubits του καταχωρητή είναι σε κατάσταση κβαντικής διεμπλοκής, διότι δεν μπορούμε να μετρήσουμε την κατάσταση του ενός, χωρίς να επηρεάσουμε και την κατάσταση του άλλου. Πράγματι, αν μετρήσουμε το πρώτο qubit, τότε θα το βρούμε με ίση πιθανότητα είτε στην κατάσταση $|0\rangle$ είτε στην κατάσταση $|1\rangle$. Όμως στην πρώτη περίπτωση και το δεύτερο qubit θα βρεθεί, σίγουρα, στην κατάσταση $|0\rangle$ ενώ στην δεύτερη περίπτωση στην κατάσταση $|1\rangle$. Δηλαδή στο σύστημά μας η κατάσταση του ενός qubit επηρεάζει άμεσα την κατάσταση του άλλου.

Ένας κβαντικός τελεστής (ή κβαντική πύλη) είναι οποιοσδήποτε μαθηματικός μετασχηματισμός μπορεί να ασκηθεί σε ένα κβαντικό σύστημα (που αποτελείται από qubits και ενδεχομένως από κβαντικούς καταχωρητές) και να μετατρέψει, ίσως, την κατάστασή του. Δηλαδή οι κβαντικές πύλες είναι τελεστές του χώρου Hilbert και αυτό που κάνουν είναι να περιστρέφουν το διάνυσμα κατάστασης ενός κβαντικού συστήματος μέσα στο χώρο Hilbert. Τους κβαντικούς τελεστές τους περιγράφουμε με τη βοήθεια πινάκων $n \times n$ όπου n είναι ο αριθμός των εισόδων και των εξόδων του τελεστή.

Βασικές προϋποθέσεις για να είναι ένα τελεστής του χώρου Hilbert κβαντική πύλη, είναι να μη μεταβάλλει το μήκος διανύσματος κατάστασης και να τηρεί τη χρονική συμμετρία των κβαντικών συστημάτων. Οι τελεστές με αυτές τις ιδιότητες ονομάζονται ορθομοναδιαίοι και περιγράφονται από ορθομοναδιαίους πίνακες. Η χρονική συμμετρία σημαίνει ότι αν με μία κβαντική πύλη G που αντιπροσωπεύει τον τελεστή G αλλάξουμε την κατάσταση ενός κβαντικού καταχωρητή από $|q_{R1}\rangle$ σε $|q_{R2}\rangle$, τότε πρέπει να δράσουμε στην κατάσταση $|q_{R2}\rangle$ με την ίδια πύλη για να πάρουμε την κατάσταση $|q_{R1}\rangle$:

$$\begin{aligned} G|q_{R1}\rangle &= |q_{R2}\rangle \\ G|q_{R2}\rangle &= |q_{R1}\rangle \end{aligned}$$

Το παραπάνω βέβαια γεγονός υποδηλώνει ότι όλες οι κβαντικές πύλες πρέπει να είναι αντιστρέψιμες. Επίσης αυτό μας δείχνει ότι οποιαδήποτε αντιστρέψιμη πύλη μπορεί να γίνει και κβαντικός τελεστής.

Οι πιο γνωστές κβαντικές πύλες είναι:

- Η κβαντική πύλη αδρανείας (μία είσοδος-έξοδος). Είναι, ουσιαστικά, η αντιστρέψιμη πύλη αδρανείας που παρουσιάστηκε σε άλλο κεφάλαιο. Περιγράφεται από τον πίνακα: $U = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.
- Η κβαντική πύλη αντιστροφής (μία είσοδος-έξοδος). Είναι, ουσιαστικά, η αντιστρέψιμη πύλη αντιστροφής που παρουσιάστηκε σε άλλο κεφάλαιο. Περιγράφεται από τον πίνακα: $U = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

- Η κβαντική πύλη Walsh-Hadamard (μία είσοδος-έξοδος). Η πιο διάσημη από τις κβαντικές πύλες. Δρώντας σε ένα σύστημα που βρίσκεται στη βασική κατάσταση $|0\rangle$, το φέρνει σε κατάσταση υπέρθεσης με όλες τις δυνατές καταστάσεις να μπορούν να παρουσιαστούν ισοπίθانا. Περιγράφεται από τον πίνακα: $U = H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

- Η κβαντική πύλη μετατόπισης φάσης (μία είσοδος-έξοδος). Η πύλη αυτή αλλάζει τη γωνία φάσης ϕ ενός qubit. Περιγράφεται από τον πίνακα: $U = \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi\phi} \end{bmatrix}$.

- Η κβαντική πύλη CNOT (δύο είσοδοι-έξοδοι). Είναι, ουσιαστικά, η αντιστρέψιμη πύλη CNOT που παρουσιάστηκε σε άλλο κεφάλαιο. Περιγράφεται από τον πίνακα: $CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

- Η κβαντική πύλη Toffoli (τρεις είσοδοι-έξοδοι). Αποτελεί, ουσιαστικά, την αντιστρέψιμη πύλη Toffoli που παρουσιάστηκε σε άλλο κεφάλαιο. Περιγράφεται από τον πίνακα: $G_{Toffoli} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$.

Ένας κβαντικός υπολογισμός είναι μια ακολουθία εφαρμογών κβαντικών πυλών σε κάποιο κβαντικό καταχωρητή. Συνήθως ένας κβαντικός υπολογισμός τερματίζεται με τη μέτρηση (παρατήρηση) κάποιου από τους κβαντικούς καταχωρητές του συστήματος. Στην περίπτωση αυτή το σύστημα θα καταρρεύσει σε μια από τις βασικές του καταστάσεις.

8.3 Ο αλγόριθμος του Grover

Ο κβαντικός αλγόριθμος που θα μας απασχολήσει στη συγκεκριμένη διδακτορική διατριβή είναι ο αλγόριθμος του Grover. Ο αλγόριθμος που θα προτείνουμε βασίζεται σε αυτόν. Στην ενότητα αυτή δίνουμε μια σύντομη περιγραφή του.

Ο Lov Grover απέδειξε [23] ότι ένας κβαντικός υπολογιστής μπορεί να μας βοηθήσει να βρούμε ένα στοιχείο σε μία μη δομημένη βάση δεδομένων N στοιχείων, αν την ερευνήσουμε \sqrt{N} φορές. Η βελτίωση που παρέχει αυτή η μέθοδος είναι ιδιαίτερα σημαντική, γιατί σε έναν κλασικό υπολογιστή, στην καλύτερη περίπτωση, το στοιχείο μπορεί να βρεθεί στην πρώτη προσπάθεια αλλά στη χειρότερη

περίπτωση το στοιχείο μπορεί να βρεθεί μετά από N προσπάθειες. Κατά μέσο όρο όμως γίνονται $N/2$ αναζητήσεις.

Αναλυτικότερα το πρόβλημα που επιλύει ο αλγόριθμος του Grover περιγράφεται ως εξής: Έστω μία μη δομημένη βάση δεδομένων που περιέχει N στοιχεία. Κάθε στοιχείο της βάσης δεδομένων έχει έναν αριθμό από το 0 έως το $N - 1$. Το στοιχείο που αντιστοιχεί στον αριθμό k συμβολίζεται με x_k . Χωρίς περιορισμό της γενικότητας μπορούμε να θεωρήσουμε:

$$N = 2^n \text{ για } n = 1, 2, 3, \dots$$

(Αν έχουμε λιγότερα στοιχεία μπορούμε να προσθέσουμε εμείς όσα χρειάζονται για να φτάσουμε στον επιθυμητό αριθμό).

Κάθε στοιχείο αντιστοιχεί σε μία από τις βασικές καταστάσεις ενός κβαντικού καταχωρητή που περιλαμβάνει n qubits.

Θεωρούμε επίσης ότι διαθέτουμε ένα σύστημα το οποίο μπορεί να αναγνωρίσει αν κάποιο στοιχείο είναι αυτό που ζητάμε ή όχι. Το σύστημα αυτό ονομάζεται Oracle και είναι μία λογική συνάρτηση f . Έτσι αν το στοιχείο που ψάχνουμε είναι το x_i , τότε έχουμε:

$$U = \begin{cases} 1, & \text{αν } x = x_i \\ 0, & \text{αν } x \neq x_i \end{cases}$$

Αν και δεν παρουσιάζονται οι πύλες που υλοποιούν το κβαντικό Oracle, όταν αυτό δράσει σε έναν κβαντικό καταχωρητή που βρίσκεται στην κατάσταση $|xy\rangle$, τότε: $|xy\rangle = |x\rangle|y\rangle \xrightarrow{O} |x\rangle|f(x) \oplus y\rangle$ όπου το $|y\rangle$ είναι το qubit του Oracle.

Κατά την αναζήτηση της βάσης δεδομένων, το qubit του Oracle τίθεται στη βασική κατάσταση $|1\rangle$ και στη συνέχεια δρα σε αυτό μία κβαντική πύλη Hadamard, η οποία έχει σαν αποτέλεσμα:

$$|1\rangle \xrightarrow{O} \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Αν το κβαντικό Oracle δράσει σε ένα τυχαίο στοιχείο της μη δομημένης βάσης δεδομένων που αναπαρίσταται από την κατάσταση του κβαντικού καταχωρητή, το αποτέλεσμα που προκύπτει είναι:

$$|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \xrightarrow{O} |x\rangle f(x) \oplus \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Ισοδύναμα η δράση του κβαντικού Oracle μπορεί να γραφεί (λόγω της μη μεταβολής του qubit του Oracle):

$$|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) = \begin{cases} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right), & \text{αν } x \text{ δεν είναι το στοιχείο που ψάχνουμε} \\ -|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right), & \text{αν } x \text{ είναι το στοιχείο που ψάχνουμε} \end{cases}$$

ή

$$|x\rangle = \begin{cases} |x\rangle, & \text{αν } x \text{ δεν είναι το στοιχείο που ψάχνουμε} \\ -|x\rangle, & \text{αν } x \text{ είναι το στοιχείο που ψάχνουμε} \end{cases}$$

$$\text{Πιο συνοπτικά: } |x\rangle \xrightarrow{O} (-1)^{f(x)} |x\rangle.$$

Συμπερασματικά, το κβαντικό Oracle δρα στις βασικές καταστάσεις $|x\rangle$ που αντιστοιχούν σε στοιχεία της μη δομημένης βάσης δεδομένων και έχει σαν αποτέλεσμα:

- Αν η βασική κατάσταση δεν αντιστοιχεί σε κατάσταση που ψάχνουμε, την αφήνει όπως ήταν.

- Αν η βασική κατάσταση αντιστοιχεί σε κατάσταση που ψάχνουμε, αλλάζει το πρόσημο της.

Το κβαντικό Oracle είναι και αυτό ένας τελεστής του χώρου Hilbert. Αν το στοιχείο που ψάχνουμε αντιστοιχεί στην βασική κατάσταση $|x_i\rangle$, ο τελεστής του κβαντικού Oracle είναι: $\hat{O} = \hat{I} - 2|x_i\rangle\langle x_i|$, όπου η πράξη X συμβολίζει το ταυστικό γινόμενο.

Για να ερευνήσουμε τη μη δομημένη βάση δεδομένων που περιέχει N στοιχεία με έναν κβαντικό υπολογιστή, αντιστοιχίζουμε κάθε ένα από τα στοιχεία με μία από τις βασικές καταστάσεις του κβαντικού καταχωρητή που περιλαμβάνει n qubits. Για να το πετύχουμε αυτό, θέτουμε τον κβαντικό καταχωρητή σε μία κατάσταση $|s\rangle$, η οποία είναι υπέρθεση όλων των βασικών καταστάσεων με το ίδιο πλάτος πιθανότητας:

$$|s\rangle = \frac{1}{\sqrt{N}}|0\rangle + \frac{1}{\sqrt{N}}|1\rangle + \dots + \frac{1}{\sqrt{N}}|N-1\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle.$$

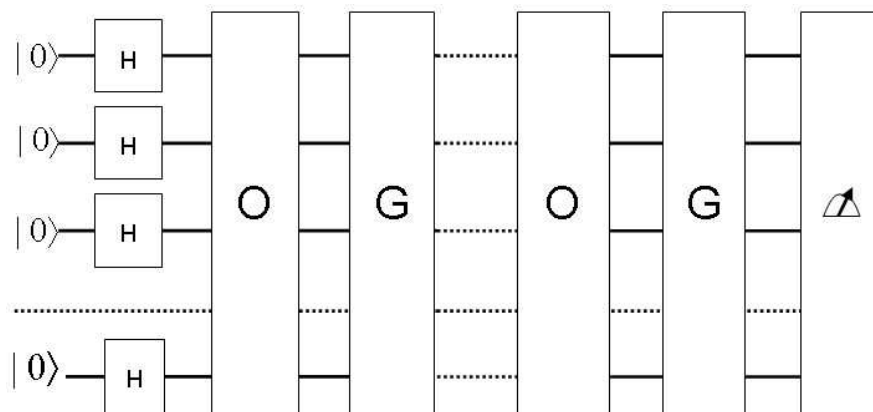
Ο τελεστής του Grover είναι:

$$\hat{G} = -(\hat{I} - 2|s\rangle\langle s|) = 2(sXs) - \hat{I}.$$

Ο αλγόριθμος του Grover είναι μία διαδοχική εφαρμογή των τελεστών \hat{O} και \hat{G} στον κβαντικό καταχωρητή περίπου $[(\frac{\pi}{4}\sqrt{N})] - 0.5$ φορές. Τα βήματα του αλγορίθμου είναι:

1. Θέτουμε όλα τα qubits του κβαντικού καταχωρητή σε $|0\rangle$, δηλαδή θέτουμε τον κβαντικό καταχωρητή στην κατάσταση $|000\dots 0\rangle$. Στη συνέχεια δρούμε σε κάθε qubit του καταχωρητή με μία πύλη Hadamard και έτσι αυτός μεταβαίνει στην κατάσταση: $|s\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |x_j\rangle$. Η κατάσταση $|s\rangle$ είναι η υπέρθεση των N βασικών καταστάσεων. Κάθε βασική κατάσταση αντιστοιχεί σε ένα στοιχείο της μη δομημένης βάσης δεδομένων και έστω ότι αναζητείται το στοιχείο που αντιστοιχεί στην κατάσταση $|x_i\rangle$.
2. Δρούμε στον καταχωρητή με τον τελεστή: $\hat{O} = \hat{I} - 2|x_i\rangle\langle x_i|$
3. Δρούμε στον κβαντικό καταχωρητή με τον τελεστή: $\hat{G} = -(\hat{I} - 2|s\rangle\langle s|) = 2(sXs) - \hat{I}$
4. Αν ο αριθμός των επαναλήψεων είναι μεγαλύτερος ή περίπου ίσος με $[(\frac{\pi}{4}\sqrt{N})] - 0.5$, τότε προχωράμε στο επόμενο βήμα. Αν ο αριθμός των επαναλήψεων είναι μικρότερος, τότε αυξάνεται ο αριθμός των επαναλήψεων κατά 1 και ο αλγόριθμος επαναλαμβάνεται από το προηγούμενο βήμα.
5. Τέλος, μετράμε την κατάσταση του κβαντικού καταχωρητή και αν το ζητούμενο στοιχείο βρίσκεται στη βάση τότε η κατάσταση του καταχωρητή αντιστοιχεί στο στοιχείο αυτό, διαφορετικά αντιστοιχεί σε κάποια τυχαία κατάσταση.

Σχηματικά ο παραπάνω αλγόριθμος παρουσιάζεται στην Εικόνα 8.3.



Σχήμα 8.3: Ο αλγόριθμος του Grover

Κεφάλαιο 9

Ένας κβαντικός αλγόριθμος για την εύρεση ελάχιστων εκφράσεων αποκλειστικού-ή για μη πλήρως ορισμένες Boolean συναρτήσεις

9.1 Θεωρητικό Υπόβαθρο

9.1.1 ESOP και ESCT ελαχιστοποίηση

Όπως δηλώθηκε στην εισαγωγή, η εύρεση ελάχιστων εκφράσεων ESOP ή ESCT για μια τυχαία Boolean συνάρτηση είναι ένα πολύ δύσκολο πρόβλημα.

Σε αυτήν την ενότητα, παρουσιάζουμε το θεωρητικό υπόβαθρο που θα μας βοηθήσει να κατασκευάσουμε αλγόριθμους που βρίσκουν ελάχιστες εκφράσεις ESOP ή ESCT για μια τυχαία Boolean συνάρτηση.

Μια τυχαία πλήρως ορισμένη Boolean συνάρτηση μπορεί πάντα να γραφεί ως εξής:

$$f(x_1, \dots, x_n) = \bar{x}_n g_1 \oplus x_n g_2 \oplus g_3 \quad (9.1)$$

Αυτό γίνεται επειδή μπορούμε να εκφράσουμε τις υποσυναρτήσεις f^0, f^1 της f όπως ακολούθως: $f^0 = g_1 \oplus g_3, f^1 = g_2 \oplus g_3$ και g_1, g_2, g_3 είναι Boolean συναρτήσεις των μεταβλητών x_1, x_2, \dots, x_{n-1} . Σύμφωνα με το ανάπτυγμα Shannon ισχύει: $f = x_n f^1 \oplus \bar{x}_n f^0 = x_n (g_2 \oplus g_3) \oplus \bar{x}_n (g_1 \oplus g_3) = x_n g_2 \oplus \bar{x}_n g_1 \oplus (x_n \oplus \bar{x}_n) g_3 = x_n g_2 \oplus \bar{x}_n g_1 \oplus g_3$.

Στις αναφορές [21, 28] αποδεικνύεται ότι κάθε ελάχιστη έκφραση ESOP γράφεται στην μορφή της Εξίσωσης 9.1. Στην αναφορά [66] αποδεικνύεται επίσης ότι κάθε ελάχιστη έκφραση ESCT γράφεται στην ίδια μορφή. Η μοναδική διαφορά είναι ότι στην πρώτη περίπτωση οι συναρτήσεις g_1, g_2, g_3 πρέπει να εκφραστούν ως ελάχιστες εκφράσεις ESOP, ενώ στη δεύτερη πρέπει να είναι ελάχιστες εκφράσεις ESCT.

Και στις δύο περιπτώσεις, αν βρούμε τις κατάλληλες g_1, g_2, g_3 συναρτήσεις (και τις ελάχιστες τους εκφράσεις ESOP ή ESCT) μπορούμε να βρούμε τις ελάχιστες εκφράσεις ESOP ή ESCT για τη συνάρτηση εισόδου μας. Αφού $g_1 = f^0 \oplus g_3$ και $g_2 = f^1 \oplus g_3$, η δυσκολία είναι να βρούμε τη g_3 συνάρτηση για την οποία η f είναι ελάχιστη. Για να τη βρούμε πρέπει να ελέγξουμε κάθε δυνατή g_3 συνάρτηση.

Η προηγούμενη παρατήρηση μας δίνει το επόμενο θεώρημα.

Θεώρημα 3 *Κάθε ελάχιστη έκφραση ESOP ή ESCT για μια πλήρως ορισμένη Boolean συνάρτηση f n μεταβλητών μπορεί να βρεθεί συγχωνεύοντας κάθε δυνατή συνάρτηση g $n - 1$ μεταβλητών με τις υποσυναρτήσεις f^0, f^1 της f .*

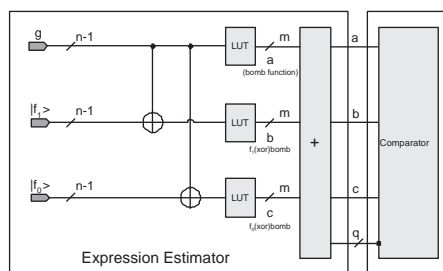
Το Θεώρημα 3 δηλώνει ότι μπορούμε να συγχωνεύσουμε οποιαδήποτε δυνατή g συνάρτηση (συμπεριλαμβάνοντας τις σταθερές συναρτήσεις 0 και 1) με τις f^0, f^1 και θα αποκτήσουμε τουλάχιστον μια ελάχιστη έκφραση ESOP (ESCT) για τη συνάρτηση f στην μορφή: $x(f^1 \oplus g) \oplus \bar{x}(f^0 \oplus g) \oplus g$. Η μοναδική διαφορά είναι ότι στην περίπτωση των εκφράσεων ESCT πρέπει να σταματήσουμε την αναδρομή στο επίπεδο των 2 μεταβλητών (καθώς κάθε μη σταθερή συνάρτηση 2 μεταβλητών έχει ESCT βάρος ίσο με 1), αντί για το επίπεδο της 1 μεταβλητής στην ESOP περίπτωση. Επιπλέον, η σταθερή συνάρτηση 1 δεν προσθέτει στο ESCT βάρος, αν και το κάνει στην ESOP περίπτωση [99].

Στην περίπτωση των μη πλήρως ορισμένων συναρτήσεων, μπορούμε να εκμεταλλευτούμε την προηγούμενη θεωρητική κατασκευή για να βρούμε ελάχιστες εκφράσεις ESOP ή ESCT, αν και το πρόβλημα γίνεται πολύ πιο δύσκολο. Κατ' ουσίαν, μια μη πλήρως ορισμένη συνάρτηση είναι ένα σύνολο από πλήρως ορισμένες συναρτήσεις (τοποθετώντας τους αδιάφορους όρους είτε στο ON-set είτε στο OFF-set της συνάρτησης με όλους τους δυνατούς συνδυασμούς). Επομένως, ένας απλοϊκός αλγόριθμος για την εύρεση ελάχιστων εκφράσεων ESOP ή ESCT θα ήταν να παράγουμε κάθε δυνατή πλήρως ορισμένη Boolean συνάρτηση που προκύπτει από την μη πλήρως ορισμένη συνάρτηση και να ελαχιστοποιήσουμε κάθε μια από αυτές. Στο τέλος θα κρατούσαμε τα καλύτερα αποτελέσματα. Αυτός ο αλγόριθμος είναι βεβαίως intractable για ένα συμβατικό υπολογιστή (ειδικά αν η συνάρτηση εισόδου μας περιέχει πολλούς αδιάφορους όρους), αλλά όπως θα δούμε στις επόμενες ενότητες είναι tractable για έναν κβαντικό υπολογιστή.

9.1.2 Ο αλγόριθμος του Grover

Στην αναφορά [23], ο L. Grover παρουσίασε έναν κβαντικό αλγόριθμο για την εύρεση ενός συγκεκριμένου στοιχείου σε μια μη ταξινομημένη βάση δεδομένων σε $O(\sqrt{N})$ βήματα (N είναι ο αριθμός των στοιχείων στη βάση δεδομένων). Αυτό το αποτέλεσμα είναι πολύ καλύτερο από το συμβατικό ανάλογο του ($O(N)$). Η αρχική

κατάσταση της βάσης δεδομένων είναι η υπέρθεση όλων των δυνατών N στοιχείων. Αυτό επιτυγχάνεται χρησιμοποιώντας την πύλη Walsh-Hadamard. Σε κάθε βήμα, ο αλγόριθμος του Grover αυξάνει το πλάτος των σημειωμένων καταστάσεων (των καταστάσεων των στοιχείων που αναζητούμε) και μειώνει εκείνο των μη σημειωμένων κατά $O(1/\sqrt{N})$. Μετά από $O(\sqrt{N})$ βήματα η πιθανότητα των σημειωμένων καταστάσεων θα είναι σχεδόν 1 και εκείνη των μη σημειωμένων καταστάσεων θα είναι σχεδόν 0. Σε αυτό το σημείο, εκτελούμε μια μέτρηση και βρίσκουμε ένα από τα στοιχεία που αναζητούμε. Η πρακτική συμπεριφορά του αλγορίθμου του Grover εξαρτάται από ένα συγκεκριμένο κβαντικό τελεστή, που λέγεται Oracle. Αυτός ο τελεστής αποφασίζει ποιες καταστάσεις είναι οι σημειωμένες και ποιες όχι.



Σχήμα 9.1: Κύκλωμα QMin-Oracle

9.2 Αλγόριθμοι

9.2.1 Προηγούμενη δουλειά

Ο αλγόριθμος του Grover μπορεί να ειπωθεί ως ένα γενικό πλαίσιο για την επίλυση πολλών δύσκολων προβλημάτων, για συμβατικούς υπολογιστές. Αυτό μπορεί να γίνει κατασκευάζοντας έναν κατάλληλο τελεστή Oracle.

Στην αναφορά [39] οι Li, Thornton και Perkowski παρουσίασαν έναν κβαντικό αλγόριθμο βασισμένο σε αυτόν του Grover, που θα μπορούσε να βρει εκφράσεις FPRM (Fixed Polarity Reed Muller) για μια συγκεκριμένη πλήρως ορισμένη Boolean συνάρτηση, με τον αριθμό των όρων μικρότερο από ένα συγκεκριμένο κατώφλι. Ο αλγόριθμός τους είναι πρακτικά ο αλγόριθμος του Grover που χρησιμοποιεί έναν ειδικό τελεστή Oracle. Αυτός ο τελεστής Oracle υπολογίζει όλες τις δυνατές εκφράσεις FPRM για την Boolean συνάρτηση εισόδου και έπειτα προτιμά εκείνες τις εκφράσεις που έχουν αριθμό από FPRM όρους μικρότερο από ένα συγκεκριμένο κατώφλι. Στο τέλος, ο αλγόριθμος του Grover θα καταλήξει σε μια από αυτές τις εκφράσεις. Ξαναρυθμίζοντας συνεχώς το κατώφλι με τον καινούριο ελάχιστο αριθμό από FPRM όρους και ξανατρέχοντας τον αλγόριθμο του Grover, κάποιος μπορεί να βρει ελάχιστες εκφράσεις FPRM για μια τυχαία πλήρως ορισμένη Boolean συνάρτηση.

Στην αναφορά [66], προτείνεται μια διαφορετική υλοποίηση του Oracle, ώστε να βρούμε ελάχιστες εκφράσεις ESOP ή ESCT για μια τυχαία πλήρως ορισμένη

Boolean συνάρτηση. Σε αυτήν την περίπτωση, το Oracle τροποποιείται με την ίδια αρχή όπως προηγουμένως. Αποτελείται από δύο διαφορετικά τμήματα: Τον Expression-Estimator και τον Comparator. Το πρώτο τμήμα υπολογίζει παράλληλα όλες τις δυνατές εκφράσεις ESOP ή ESCT και προσδιορίζει τον αριθμό των όρων (σύνθετων ή γινομένων) σε κάθε μια (συνάρτηση κόστους). Ο Comparator είναι ο ίδιος όπως στην αναφορά [39]. Αυτός ο αλγόριθμος προσδιορίζει επίσης τις εκφράσεις που έχουν αριθμό από όρους (γινόμενα ή σύνθετους) μικρότερο από ή ίσο με το συγκεκριμένο κατώφλι. Ο προτεινόμενος κβαντικός αλγόριθμος ονομάστηκε QMin.

9.2.2 DCQMin

Όλοι οι προαναφερόμενοι αλγόριθμοι είναι σχεδιασμένοι για να ελαχιστοποιούν πλήρως ορισμένες Boolean συναρτήσεις. Ο προτεινόμενος αλγόριθμός μας, που ονομάζεται DCQMin, είναι ουσιαστικά ο αλγόριθμος QMin, αλλά ξανασχεδιασμένος για να ελαχιστοποιεί μη πλήρως ορισμένες Boolean συναρτήσεις.

Τα στάδια του DCQMin παρουσιάζονται στην Εικόνα 9.2. Όπως μπορούμε να παρατηρήσουμε, ο DCQMin είναι ουσιαστικά ο αλγόριθμος του Grover, με τροποποιημένο Oracle.

Ο DCQMin αποτελείται από δύο διακριτά συστατικά. Το πρώτο, που ονομάζεται Expr-Estimator, είναι αυτό που υλοποιεί το Θεώρημα 3 και παράγει εκφράσεις ESOP ή ESCT για τη συνάρτηση εισόδου μας. Το δεύτερο είναι ο Comparator που συγκρίνει τον αριθμό των όρων (σύνθετων ή γινομένων) κάθε παραγόμενης έκφρασης με ένα δεδομένο κατώφλι. Επιστρέφει 1, δηλαδή δηλώνει μια σημειωμένη κατάσταση, αν αυτός ο αριθμός είναι μικρότερος ή ίσος με το κατώφλι και 0 αλλιώς, δηλώνοντας έτσι μια μη σημειωμένη κατάσταση.

Μπορούμε να δούμε το συστατικό Expr-Estimator στην Εικόνα 9.1. Αποτελείται από τρεις κύριους διαύλους. Ο πρώτος αρχικοποιείται από τις πύλες Walsh-Hadamard της Εικόνας 9.2 (παραμένουν έξω από τον τελεστή Oracle και μέσα στο τμήμα DCQ της Εικόνας 9.4) και αντιστοιχεί στη g συνάρτηση του Θεωρήματος 3. Προφανώς στο διάυλο αυτό δημιουργούνται σε υπέρθεση οι MT αναπαραστάσεις όλων των δυνατών συναρτήσεων $n - 1$ μεταβλητών. Ο δεύτερος και ο τρίτος αρχικοποιούνται από την αναπαράσταση ελαχιστόρων των υποσυναρτήσεων f^1 και f^0 της συνάρτησης εισόδου μας, αντίστοιχα. Οι πύλες CNOT παράγουν το bit προς bit XOR άθροισμα της g συνάρτησης (πρώτος διάυλος) με τις f^1 και f^0 αντίστοιχα, παράγοντας έτσι τις συναρτήσεις $g \oplus f^1$ και $g \oplus f^0$ του Θεωρήματος 3. Σύμφωνα με το Θεώρημα αυτό αν ελέγξουμε όλες τις δυνατές g συναρτήσεις, θα βρούμε όλες τις ελάχιστες ESOP ή ESCT εκφράσεις της, δεδομένου ότι γνωρίζουμε τα ESOP ή ESCT βάρη τους αντίστοιχα.

Είναι σημαντικό να σημειώσουμε ότι η συνάρτηση εισόδου στο κβαντικό κύκλωμα QMin είναι η αναπαράσταση ελαχιστόρων. Επομένως, είναι εύκολο να παράγουμε τις αναπαραστάσεις ελαχιστόρων των υποσυναρτήσεών της.

Οι τελεστές LUT που ακολουθούν, δίνουν το βάρος των συναρτήσεων εισόδου τους και είναι στην πραγματικότητα αναδρομικά στιγμιότυπα του μεγαλύτερου κυκλώματος Expr-Estimator. Για τις εκφράσεις ESOP και τις συναρτήσεις 1 μεταβλητής, το κύκλωμα LUT φαίνεται στην Εικόνα 9.2 (κάτω επίπεδο, δεξιότερο

τιμήμα). Για τις εκφράσεις ESCT και τις συναρτήσεις 2 μεταβλητών, το κύκλωμα LUT φαίνεται στην Εικόνα 9.2 (κάτω επίπεδο, αριστερότερο τμήμα). Για όλες τις άλλες περιπτώσεις το κύκλωμα LUT είναι στην πραγματικότητα το κύκλωμα Expr-Estimator, που ορίζεται αναδρομικά.

Σε αυτό το σημείο, ο αριθμός των όρων για κάθε δυνατή έκφραση ESOP ή ESCT των συναρτήσεων g , $g \oplus f^1$, $g \oplus f^0$ υπολογίζεται. Σύμφωνα με το Θεώρημα 3, ο αριθμός των όρων σε μια έκφραση της αρχικής μας συνάρτησης είναι το άθροισμά τους. Επομένως, χρησιμοποιούμε έναν χβαντικό αθροιστή για να εκτελέσουμε αυτήν την εργασία. Ένας τέτοιος χβαντικός αθροιστής παρουσιάζεται στην αναφορά [22].

Το άλλο συστατικό του DCQMin-Oracle είναι ο Comparator. Ο Comparator που χρησιμοποιείται έχει παρουσιαστεί στην αναφορά [39]. Ένας 2-qubit Comparator φαίνεται στην Εικόνα 9.3. Θα έπρεπε να σημειωθεί ότι ο Comparator χρησιμοποιείται μόνο μια φορά και όχι μέσα στην αναδρομή που εκτελείται από τον Expression-Estimator.

Πρέπει να σημειωθεί ότι η έξοδος q του τελεστή Oracle του DCQMin στην Εικόνα 9.1 είναι η είσοδος στον τελεστή Comparator. Οι έξοδοι a, b, c αντιστοιχούν στις g , $g \oplus f^1$, $g \oplus f^0$ συναρτήσεις του Θεωρήματος 3 και θεωρούνται έξοδοι του κυκλώματός μας χβαντικής ελαχιστοποίησης (Εικόνα 9.2). Αυτές οι έξοδοι απλά διαδίδονται από τον Comparator.

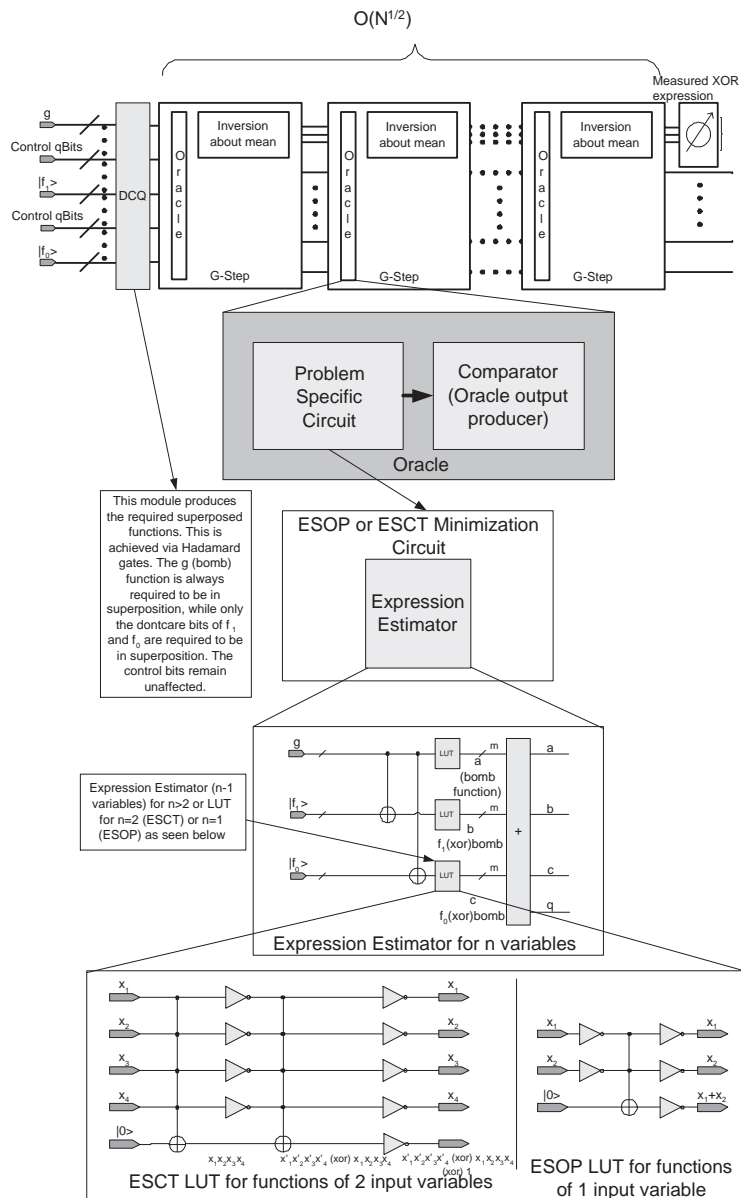
Ο πραγματικός αλγόριθμος χβαντικής ελαχιστοποίησης είναι, βέβαια, ο αλγόριθμος του Grover, χρησιμοποιώντας το κύκλωμα Oracle του DCQMin που περιγράφηκε προηγουμένως ως ο εξειδικευμένος τελεστής Oracle. Παίρνει ως είσοδο (Εικόνα 9.2) μια τυχαία switching συνάρτηση f και ένα κατώφλι. Οι έξοδοι του είναι οι a, b, c, q έξοδοι του κυκλώματος Oracle. Στο τέλος της εκτέλεσης, εκείνες οι εκφράσεις της f που έχουν αριθμό όρων (είτε σύνθετων είτε γινομένων, ανάλογα με το είδος των εκφράσεων που αναζητούμε) μικρότερο ή ίσο με το κατώφλι, θα έχουν, όλες μαζί αθροιστικά, πιθανότητα σχεδόν 1 (σημειωμένες καταστάσεις), ενώ όλες οι άλλες θα έχουν αθροιστικά πιθανότητα σχεδόν 0 (μη σημειωμένες καταστάσεις). Όταν μετρούμε μια από τις εξόδους του (για παράδειγμα την έξοδο a), όλες οι έξοδοι (και επομένως οι διαδιδόμενες αντίστοιχες είσοδοι) θα καταρρεύσουν σε αυτές που αντιστοιχούν στην μία από τις σημειωμένες καταστάσεις. Το βάρος της έκφρασης εξόδου θα δίνεται από το q , ενώ η πραγματική έκφραση μπορεί να ξανακατασκευαστεί από τις εξόδους a, b, c σύμφωνα με το Θεώρημα 3.

Το τμήμα DCQ της Εικόνας 9.2, παίρνει ως είσοδο την μη πλήρως ορισμένη Boolean συνάρτηση και παράγει το συσχετισμένο της σύνολο των πλήρως ορισμένων Boolean συναρτήσεων, όπως περιγράφεται στην Υποενότητα 9.1.1. Όλες αυτές οι συναρτήσεις είναι σε υπέρθεση και οδηγούνται στο κύριο χβαντικό κύκλωμα. Ο DCQMin χρησιμοποιεί τον αλγόριθμο του Grover για να βρει ποιες από αυτές τις συναρτήσεις έχουν μια λύση με τον ελάχιστο δυνατό αριθμό γινομένων (περίπτωση ESOP) ή σύνθετων όρων (περίπτωση ESCT) και επιστρέφει μια από αυτές.

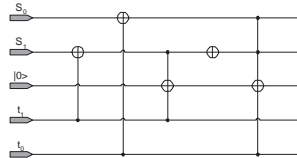
Μια μη πλήρως ορισμένη Boolean συνάρτηση μπορεί να περιγραφεί ως μια Boolean συνάρτηση όπου κάποιοι ελαχιστόροι είναι είτε 0 ή 1. Αυτό σημαίνει ότι δεν μπορούμε, αρχικά, να αποφασίσουμε την πραγματική τιμή για αυτούς τους ελαχιστόρους. Στα χβαντικά κυκλώματα αυτό είναι ισοδύναμο με το να έχουμε ένα

qubit να είναι είτε στην κατάσταση $|0\rangle$ ή στην κατάσταση $|1\rangle$, με ίση πιθανότητα. Η είσοδος στον DCQ είναι η αναπαράσταση ελαχιστόρων της μη πλήρως ορισμένης Boolean συνάρτησης, όπου κάθε qubit αναπαριστά την παρουσία ή την απουσία του αντίστοιχου ελαχιστόρου του. Κάποια από αυτά τα qubits αναπαριστούν αδιάφορους ελαχιστόρους. Όταν θέτουμε αυτά τα qubits σε υπέρθεση των βασικών τους καταστάσεων είναι ισοδύναμο με το να δηλώνουμε τους αντίστοιχους ελαχιστόρους τους ως αδιάφορους ελαχιστόρους. Αυτό κάνει πραγματικά το τμήμα DCQ. Προσθέτει μια πύλη Walsh-Hadamard για κάθε τέτοιο qubit, θέτοντας έτσι αυτά τα qubits σε υπέρθεση των βασικών τους καταστάσεων.

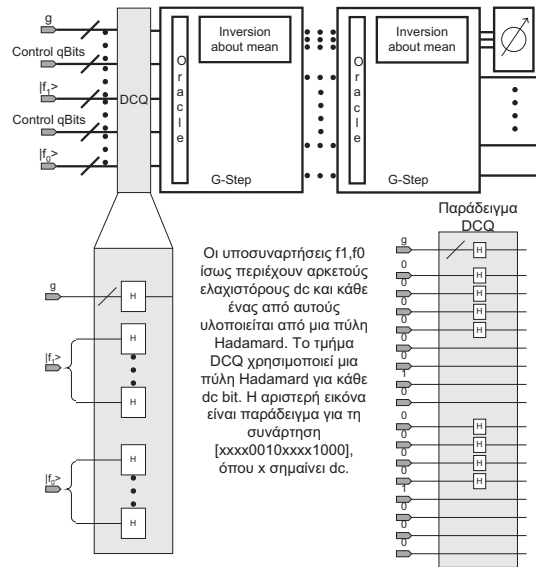
Ένα παράδειγμα του τμήματος DCQ για τη συνάρτηση 4 μεταβλητών [ddd0010ddd0100] φαίνεται στην Εικόνα 9.4.



Σχήμα 9.2: Ιεραρχία του αλγορίθμου DCQmin



Σχήμα 9.3: Ένας 2-qubit Comparator, που συγκρίνει το s_0s_1 με το t_0t_1 , υλοποιώντας τη συνάρτηση: $(s_1 \oplus t_1)t_1 \oplus (s_1 \oplus t_1)(s_0 \oplus t_0)t_0$



Σχήμα 9.4: Κύκλωμα DCQmin-Oracle με παράδειγμα

Κεφάλαιο 10

Συμπεράσματα και μελλοντικές κατευθύνσεις

Η κύρια συνεισφορά αυτής της διδακτορικής διατριβής είναι δύο καινούριοι ευριστικοί αλγόριθμοι για την ελαχιστοποίηση εκφράσεων ESOP και ESCT για μη πλήρως ορισμένες συναρτήσεις πολλών εξόδων, καθώς επίσης και η τροποποίηση της συμβατικής αποσύνθεσης και σύνθεσης μιας συνάρτησης, ώστε να προσαρμοστεί στην τριαδική λογική $(\{0, 1, d\})$. Σημειώνουμε ότι ο αλγόριθμος για τις εκφράσεις ESCT είναι αρκετά γενικός, ώστε, με τη χρήση της κατάλληλης σημαίας, να μπορεί να παράγει και εκφράσεις ESOP.

Ο νέοι αλγόριθμοι συγκρίθηκαν με άλλους προηγούμενα δημοσιευμένους αλγόριθμους στο χώρο της ESOP και ESCT ελαχιστοποίησης για μη πλήρως ορισμένες συναρτήσεις. Όπως υποδεικνύουν τα πειραματικά αποτελέσματα που παρουσιάστηκαν παραπάνω, οι αλγόριθμοι που προτείνονται είναι αποδοτικότεροι από τους άλλους στις περισσότερες περιπτώσεις, χρησιμοποιώντας ως βάση σύγκρισης το μέγεθος των εκφράσεων.

Επιπλέον, ο καινούριοι αλγόριθμοι οδηγούν σε καλύτερα αποτελέσματα, αν λάβουμε υπ' όψιν μας τους αδιάφορους όρους, συγκρινόμενοι με τα αποτελέσματα για τις ίδιες μετροσυναρτήσεις αν θέσουμε όλους τους αδιάφορους όρους ίσους με '0' ή '1' (χρησιμοποιώντας αλγόριθμους για πλήρως ορισμένες συναρτήσεις). Αυτό υποδεικνύει ότι οι προτεινόμενοι αλγόριθμοι εκμεταλλεύονται την παρουσία των αδιάφορων όρων (επιλέγοντας κάποιους από αυτούς σαν '0' και άλλους σαν '1') ώστε να ελαχιστοποιήσουν παραπέρα κάθε μετροσυνάρτηση.

Η μελλοντική δουλειά θα περιλαμβάνει τη βελτιστοποίηση των αλγορίθμων, χρησιμοποιώντας διαφορετικές τεχνικές στην επιλογή των όρων (ειδικότερα των αδιάφορων όρων, π.χ. επιλέγοντας τυχαία κάποιους από αυτούς ώστε να μειώσουμε το χρόνο εκτέλεσης ή επιλέγοντας αδιάφορους όρους που 'ταιριάζουν' με τους επιλεγμένους όρους από το σύνολο *CurrentTerms* ώστε να μειώσουμε το μέγεθος των εκφράσεων), ή προσπαθώντας να αποφύγουμε πιθανά τοπικά ελάχιστα κατά τη διάρκεια της διαδικασίας ελαχιστοποίησης, χρησιμοποιώντας τεχνικές όπως το annealing (π.χ. επιτρέποντας περιοδικά να αυξάνουμε το μέγεθος της έκ-

φρασης, ώστε να πάρουμε καλύτερα αποτελέσματα σε επόμενες επαναλήψεις). Μια ενδιαφέρουσα προοπτική είναι να ψάξουμε στο χώρο των πιθανών αναδιατάξεων μεταβλητών ώστε να επιτύχουμε καλύτερη ελαχιστοποίηση.

Προτείνουμε επίσης μια μέθοδο που βρίσκει ακριβείς λύσεις για τυχαίες μη πλήρως ορισμένες συναρτήσεις 6 μεταβλητών. Με βάση όσα γνωρίζουμε, αυτή είναι η πρώτη προσπάθεια για το παραπάνω πρόβλημα. Αν και η μέθοδός μας είναι απαιτητική υπολογιστικά, είναι tractable και μπορεί να δώσει πρακτικές λύσεις. Η μέθοδος μπορεί να επεκταθεί εύκολα σε συναρτήσεις πολλών εξόδων και, σαν μια ευριστική μέθοδος, για συναρτήσεις με περισσότερες από 6 μεταβλητές. Η μελλοντική δουλειά μας σκοπεύει στην επέκταση της προσέγγισής μας σε εκφράσεις ESCT και στην ανάπτυξη περισσότερο αποτελεσματικών αλγορίθμων για τη δημιουργία των πινάκων RT_{DC} .

Τέλος, όσον αφορά τους κβαντικούς αλγόριθμους, ο αλγόριθμος QMin [66] επεκτάθηκε για την ελαχιστοποίηση μη πλήρως ορισμένων Boolean συναρτήσεων. Ο DCQMin είναι ένας κβαντικός αλγόριθμος που παίρνει σαν είσοδο μια τυχαία μη πλήρως ορισμένη Boolean συνάρτηση και προσδιορίζει τις εκφράσεις της ESOP ή ESCT με αριθμό όρων (σύνθετων ή γινομένων) μικρότερο από ένα προσδιορισμένο κατώφλι. Είναι φανερό ότι εκτελώντας επαναληπτικά τον DCQMin και ενημερώνοντας κατάλληλα το κατώφλι, μπορούμε να βρούμε ελάχιστες εκφράσεις για μια συγκεκριμένη συνάρτηση. Μια αρχική εκτίμηση για το κατώφλι μπορεί να αποκτηθεί από συμβατικούς ευριστικούς αλγόριθμους ελαχιστοποίησης, όπως ο EXORCISM-4 [46], ο QuiXor [93] ή ο QuickDCMIN [32] (για την ESOP περίπτωση) ή ο EMin1 [97] (για την ESCT περίπτωση).

Η μελλοντική δουλειά θα εστιάσει στην επέκταση αυτού του αλγορίθμου, ώστε να αντιμετωπίσει το πρόβλημα της ελαχιστοποίησης λογικών συναρτήσεων πολλών εξόδων ή στην κβαντική επέκταση του αλγορίθμου που παρουσιάστηκε για την εύρεση ακριβών εκφράσεων ESOP, θεωρώντας πάντα μη πλήρως ορισμένες συναρτήσεις ώστε να εχμεταλλευτούμε την κβαντική παραλληλία.

Bibliography

- [1] R. L. Ashenurst. The decomposition of switching functions. *Ann. Computation Lab. Harvard University*, 29:74–116, 1959.
- [2] B. Becker, R. Drechsler, and M. Theobald. Minimization of 2-level and/xor expressions using ordered kronecker functional decision diagrams.
- [3] T. Bengtsson and E. Dubrova. A sufficient condition for detection of xor-type logic. In *NORCHIP'01*, pages 271–278, November 2001.
- [4] G. Bioul, M. Davio, and J. P. Deschamps. Minimization of ring-sum expansions of boolean functions. *Philips Res. Repts*, 28:17–36, 1973.
- [5] C. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.
- [6] M. Chrzanowska-Jeske, Y. Xu, and M. Perkowski. Logic synthesis for a regular layout. *VLSI Design*, 10(1):35–55, 1999.
- [7] H. A. Curtis. A new approach to the design of switching circuits. *Princeton, N.J. Van Nostrand*, 1962.
- [8] D. Debnath and T. Sasao. Grmin: A heuristic simplification algorithm for generalized reed-muller expressions. In *Proceedings of Asia and South Pacific Conference on Design Automation*, pages 341–347, August-September 1995.
- [9] D. Debnath and T. Sasao. Grmin2: A heuristic simplification algorithm for generalized reed-muller expressions. *IEE Proceedings - Computers and Digital Techniques*, 1996.
- [10] D. Debnath and T. Sasao. Exact minimization of fprms for incompletely specified functions. In *Proc. 4th Int. Workshop on Applications of Reed-Muller Expansion in Circuit Design*, pages 253–264, 1999.
- [11] R. Drechsler and B. Becker. Sympathy: Fast exact minimization of fixed polarity reed-muller expressions for symmetric functions. In *IEEE European Design And Test Conference*, pages 91–97, Paris, March 1995.

- [12] R. Drechsler, B. Becker, and N. Drechsler. Genetic algorithm for minimization of fixed polarity reed-muller expressions. *Computers and Digital Techniques, IEE Proceedings*, 147(5):349–353, October 2000.
- [13] R. Drechsler and N. Gockel. Minimization of bdds by evolutionary algorithms. In *International Workshop on Logic Synthesis (IWLS'97)*, Lake Tahoe, 1997.
- [14] R. Drechsler, M. Theobald, and B. Becker. Fast ofdd-based minimization of fixed polarity reed-muller expressions. *IEEE Transactions on Computers*, 45(11):1294–1299, November 1996.
- [15] E. Dubrova and T. Bengtsson. An algorithm for detecting xor-type logic. In *5th International Workshop of Applications of the Reed Muller Expansion in Circuit Design*, pages 271–276, August 2001.
- [16] G. W. Dueck, D. Maslov, J. T. Butler, V. P. Shmerko, and S. N. Yanuskevich. A method to find the best mixed polarity reed-muller expression using transeunt triangle. In *5th International Workshop on Applications of Reed-Muller Expansion in Circuit Design (RM)*, pages 82–93, Starkville, MS, August 2001.
- [17] V. Dvorak. Time- and space-efficient evaluation of sparse boolean functions in embedded software. In *14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, pages 178–185, Los Alamitos, US, 2007.
- [18] B. J. Falkowski and C. Chang. Minimization of k-variable-mixed-polarity reed-muller expansions. *VLSI Design*, 11(4):311–320, 2000.
- [19] C. Files, R. Drechsler, and M. A. Perkowski. Functional decomposition of mvl functions using multi-valued decision diagrams. In *Proc. ISMVL'97*, pages 27–32, 1997.
- [20] Michael P. Frank. Introduction to reversible computing: motivation, progress, and challenges. In *CF '05: Proceedings of the 2nd conference on Computing frontiers*, pages 385–390, New York, NY, USA, 2005. ACM.
- [21] A. Gaidukov. Algorithm to derive minimum esop for 6-variable function. In *5th IWBP*, 2002.
- [22] Phil Gossett. Quantum carry-save arithmetic, 1998.
- [23] L.K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th Annual ACM Symposium on Theory of Computation*, pages 212–219, 1996.
- [24] M. Helliwell and M. Perkowski. A fast algorithm to minimize multi-output mixed-polarity generalized reed-muller forms. In *25th ACM/IEEE Conference on Design Automation*, pages 427–432, 1988.

- [25] T. Hirayama and Y. Nishitani. A simplification algorithm of and-exor expressions for multiple-output functions. In *2nd Workshop on Applications of Reed-Muller Expansion in Circuit Design*, pages 88–93, Chiba City, Japan, 1995.
- [26] T. Hirayama and Y. Nishitani. Efficient search methods for obtaining exact minimum and-exor expressions. In *Third IEEE International Workshop on Electronic Design, Test and Applications (DELTA'06)*, January 2006.
- [27] T. Hirayama and Y. Nishitani. Exact minimization of and-exor expressions of practical benchmark functions. In *Journal of Circuits, Systems, and Computers (JCSC)*, October 2007.
- [28] T. Hirayama, Y. Nishitani, and T. Sato. A faster algorithm of minimizing and-exor expressions. In *2002 Asia-Pacific Conference on Circuits and Systems*, volume 2, pages 293–298, 2002.
- [29] T. Hirayama, M. Takahashi, and Y. Nishitani. Simplification of exclusive-or sum-of-products expressions through function transformation. In *IEEE Asia Pacific Conference on Circuits and Systems, 2006 (APCCAS 2006)*, pages 1480–1483, Singapore, December 2006.
- [30] D. Jankovic, R. S. Stankovic, and C. Moraga. Arithmetic expressions optimisation using dual polarity property. *Serbian Journal of Electrical Engineering*, 1(1):71–80, November 2003.
- [31] G. Jennings. Symbolic incompletely specified functions for correct evaluation in the presence of indeterminate input values. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences (HICSS'95)*, 1995.
- [32] M. Kalathas, D. Voudouris, and G. Papakonstantinou. A heuristic algorithm to minimize esops for multiple-output incompletely specified functions. In *2006 ACM Great Lakes Symposium on VLSI - GLSVLSI '06*, pages 357–361, Philadelphia, PA, USA, April 30 - May 2 2006.
- [33] M. Kalathas, D. Voudouris, and G. Papakonstantinou. Esct minimization for incompletely specified functions. *Accepted for publication in Journal of Multiple-Valued Logic and Soft Computing*, 2010.
- [34] D. E. Knuth. *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*. Addison-Wesley, 1981.
- [35] T. Kozłowski, E. L. Dagless, and J. M. Saul. An enhanced algorithm for the minimization of exclusive-or sum-of-products for incompletely specified functions. In *1995 IEEE Intrn. Conf. on Computer Design (ICDD'95)*, page 244, 1995.

- [36] G. Lee. Logic synthesis for cellular architecture fpga using bdd. In *ASP-DAC 97*, pages 253–258, January 1997.
- [37] G. Lee and R. Drechsler. Etd-based synthesis of term-based fpgas for incompletely specified boolean functions. In *ASP-DAC 98*, pages 75–80, Yokohama, Japan, February 1998.
- [38] G. Lee, R. Drechsler, and M.A. Perkowski. Etd-based synthesis of two-dimensional cellular arrays for multi-output incompletely specified boolean functions. *Computers and Digital Techniques, IEE Proceedings*, 146(6):302–308, November 1999.
- [39] L. Li, M. Thornton, and M. Perkowski. A quantum cad accelerator based on grover’s algorithm for finding the minimum fixed polarity reed-muller form. In *Proc. of the ISMVL’06*, volume 00, pages 17–20, May 2006.
- [40] P. Lindgren, R. Drechsler, and B. Becker. Look-up table fpga synthesis from minimized multi-valued pseudo kronecker expressions. In *ISMVL 98*.
- [41] P. Lindgren, R. Drechsler, and B. Becker. Improved minimization methods of pseudo kronecker expressions for multiple output functions. In *Int’l Symp. Circ. and Systems*, pages 187–190, 1998.
- [42] K. K. Maitra. Cascaded switching networks of two-input flexible cells. *IRE Trans. Electron. Comput.*, EC-11:136–143, 1962.
- [43] Dan C. Marinescu and Gabriela M. Marinescu. *Approaching Quantum Computing*. Prentice Hall, 2004.
- [44] M. Matsuura and T. Sasao. Representation of incompletely specified switching functions using pseudo-kronecker decision diagrams. In *Reed-Muller 2001*, pages 27–33, Starkville, Mississippi, USA, August 2001.
- [45] R. C. Minnick. Cutpoint cellular logic. *IEEE Trans. Electron. Comput.*, EC-13:685–698, December 1964.
- [46] A. Mishchenko and M. Perkowski. Fast heuristic minimization of exclusive sums-of-products. In *5th International Workshop on Applications of the Reed Muller Expansion in Circuit Design*, August 2001.
- [47] A. Mishchenko and M. Perkowski. Logic synthesis of reversible wave cascades. In *International Workshop on Logic and Synthesis 2002*, New Orleans, Louisiana, June 4-7 2002.
- [48] M. Perkowski N. Song. A new approach to and/or/exor factorization for regular arrays. In *Proc. 1998 Euromicro*, pages 269–276, Vasteras, Sweden, August 25-27 1998.
- [49] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.

- [50] H. Ochi. An exact minimization algorithm of and-exor expressions using encoded mrcf. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E79-A(12):2131–2133, December 1996.
- [51] G. Papakonstantinou. A synthesis method for cutpoint cellular arrays. *IEEE Transactions on Computers*, C-21(12):1286–1292, December 1972.
- [52] G. Papakonstantinou. Cascade transformation. *IEEE Transactions on Computers*, 25(1):93–95, January 1976.
- [53] G. Papakonstantinou. Minimization of modulo-2 sum of products. *IEEE Trans. on Computers*, 28(2):163–167, 1979.
- [54] G. Papakonstantinou. Minimal modulo-2 expressions of switching functions with five variables. *Intrn. Journal of Electr.*, 50(3):211–214, 1981.
- [55] L. Parrilla, J. Ortega, and A. Lloris. Nondeterministic and-exor minimization by using rewrite rules and simulated annealing. *Computers and Digital Techniques, IEE Proceedings*, 146(1):1–8, January 1999.
- [56] M. Perkowski and M. Chrzanowska-Jeske. An exact algorithm to minimize mixed-radix exclusive sums of products for incompletely specified boolean functions. *IEEE International Symposium on Circuits and Systems, 1990*, 2:1652–1655, May 1990.
- [57] M. A. Perkowski. A new representation of strongly unspecified switching functions and its application to multi-level and/or/exor synthesis. In *IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion*, pages 143–151, 1995.
- [58] M. A. Perkowski, M. Chrzanowska-Jeske, A. Sarabi, and I. Schafer. Multi-level logic synthesis based on kronecker decision diagrams and boolean ternary decision diagrams for incompletely specified functions. *VLSI Design*, 3(3-4):301–313, 1995.
- [59] G. R. Pogosyan and I. G. Rosenberg. Gaussian eliminations in $gf(2)$ to construct an optimal esop expression for a partial boolean function. Technical report, International Christian University, December 2002.
- [60] D. Popel. Fixed polarity reed-muller minimization of incompletely specified boolean functions based on information estimations on decision trees. In *IAPR International Conference on Pattern Recognition and Information Processing (PRIP'99)*, pages 197–206, Minsk, Belarus, 1999.
- [61] D. V. Popel and A. Dani. Sierpinski gaskets for logic functions representation. *CoRR*, cs.LO/020624, 2002.
- [62] P. Porwik. Efficient calculation of the reed-muller form by means of the walsh transform. *Int. J. Appl. Math. Comput. Sci.*, 12(4):571–579, 2002.

- [63] Yvan Van Rentergem, Alexis De Vos, and Koen De Keyser. Six synthesis methods for reversible logic. *Open Systems & Information Dynamics*, 14(1):91–116, 2007.
- [64] J. P. Roth and R. M. Karp. Minimization over boolean graphs. *IBM Journal*, pages 227–238, April 1962.
- [65] M. Sampson, M. Kalathas, D. Voudouris, and G. Papakonstantinou. Exact esop expressions for incompletely specified functions. *Submitted in Integration, the VLSI Journal*, 2010.
- [66] M. Sampson, D. Voudouris, and G. Papakonstantinou. A quantum algorithm for finding minimum exclusive-or expressions. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI '07)*, pages 416–421, 2007.
- [67] M. Sampson, D. Voudouris, and G. Papakonstantinou. A quantum algorithm for finding minimum exclusive-or expressions for multi-output incompletely specified boolean functions. In *The 2008 International Conference on Computer Design, CDES'08*, Las Vegas, 2008.
- [68] A. Sarabi, N. Song, M. Chrzanowska-Jeske, and M. Perkowski. A comprehensive approach to logic synthesis and physical design for two-dimensional logic arrays. In *DAC 1994*, pages 321–326, 1994.
- [69] T. Sasao. A transformation of multiple-valued input two-valued output functions and its application to simplification of exclusive-or sum-of-products expressions. In *21st International Symposium on Multiple-Valued Logic*, pages 270–279, 1991.
- [70] T. Sasao. An exact minimization of and-exor expressions using reduced covering functions. In *Synthesis and Simulation Meeting and International Interchange*, pages 374–383, October 1993.
- [71] T. Sasao. Exmin2: A simplification algorithm for exclusive-or sum-of-products expressions for multiple-valued input two-valued output functions. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 12(5):621–632, May 1993.
- [72] T. Sasao. *Switching theory for logic synthesis*. Kluwer Academic Publishers, 1999.
- [73] T. Sasao and P. Besslich. On the complexity of mod-2 sum pla's. *IEEE Transactions on Computers*, 39(2):262–266, February 1990.
- [74] T. Sasao and D. Debnath. On a minimization of generalized reed-muller expressions. *IPSJ SIG Notes*, 94(93):115–120, October 1994.
- [75] T. Sasao and D. Debnath. Generalized reed-muller expressions: Complexity and an exact minimization algorithm (special section on vlsi design

and cad algorithms). *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 79(12):2123–2130, December 1996.

- [76] T. Sasao and Munehiro Matsuura. Bdd representation for incompletely specified multiple-output logic functions and its applications to functional decomposition. In *DAC2005*, June 2005.
- [77] I. Schaefer, M. Perkowski, and H. Wu. Multilevel logic synthesis for cellular fpgas based on orthogonal expansions. In *IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, pages 42–51, Hanburg, Germany, September 1993.
- [78] V. Shmerko, D. Popel, R. Stankovic, V. Cheushev, and S. Yanushkevich. And/exor minimization of switching functions based on information-theoretical approach. *Facta Universitatis Journal, Series: Electronics and Energetics (Yugoslavia)*, 13(1):11–25, 2000.
- [79] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *SIAM J. Computing* 26, pages 1484–1509, 1997.
- [80] N. Song and M. Perkowski. Minimization of exclusive sums of multi-valued complex terms for logic cell arrays. In *ISMVL 98*, page 32.
- [81] N. Song and M. A. Perkowski. Minimization of exclusive sum-of-products expressions for multiple-valued input, incompletely specified functions. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 15(4), April 1996.
- [82] N. Song and M. A. Perkowski. New fast approach to approximate esop minimization for incompletely specified multi-output functions. In *IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, pages 61–72, 1997.
- [83] R. S. Stankovic, M. G. Karpovsky, and J. T. Astola. Reduction of the number of coefficients in arithmetic expressions by autocorrelation functions.
- [84] B. Steinbach. Adjacency graph of the snf as source of information. In *Proceedings of the 7th International Workshop on Boolean Problems*, pages 19–28, Freiberg, September 2006.
- [85] B. Steinbach and G. Kempe. Minimization of and-exor expressions. In *IFIP WG 10.5 - Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, pages 20–26, Hamburg, 1993.
- [86] B. Steinbach and A. De Vos. The shape of the snf as a source of information. In *Proceedings of the 8th International Workshop on Boolean Problems*, volume 8, pages 127–136, 2008.

- [87] B. Steinbach and A. Zakrevskij. New approaches for minimization of esops, 2002.
- [88] S. Stergiou, K. Daskalakis, and G. Papakonstantinou. A fast and efficient heuristic esop minimization algorithm. In *GLSVLSI*, April 2004.
- [89] S. Stergiou and G. Papakonstantinou. An efficient algorithm for exact esop minimization. In *The 2002 Int. Conf. on VLSI*, June 2002.
- [90] S. Stergiou and G. Papakonstantinou. Towards a general novel exact esop minimization methodology. In *6th Intrn. Workshop on Appl. of the Reed Muller Expansion in Circuit Design*, March 2003.
- [91] S. Stergiou and G. Papakonstantinou. Exact minimization of esop expressions with less than eight product terms. *Journal of Circuits, Systems, and Computers*, 13(1):1–15, 2004.
- [92] S. Stergiou, D. Voudouris, and G. Papakonstantinou. Exact and heuristic mvesop minimization algorithms. *IEICE Trans. on Fundamentals*, E87-A(1), January 2004.
- [93] S. Stergiou, D. Voudouris, and G. Papakonstantinou. Multiple-value exclusive-or sum-of-products minimization algorithms. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E87-A(5):1226–1234, May 2004.
- [94] M. A. Thornton, B. Q. Vu, and R. Drechsler. Odd/even cube covering for minimizing esop circuits.
- [95] A. De Vos and Y. Van Rentergem. Synthesis of reversible circuits. In *Proceedings of the 14-th International Workshop on Logic and Synthesis*, pages 101–108, 2005.
- [96] D. Voudouris, M. Kalathas, and G. Papakonstantinou. Decomposition of multi-output boolean functions. *HERMIS Journal*, 6:154–161, 2005.
- [97] D. Voudouris and G. Papakonstantinou. Maitra cascade minimization. In *6th IWSBP*, page 209, 2004.
- [98] D. Voudouris, M. Sampson, and G. Papakonstantinou. Variable reordering for reversible wave cascades. In *HERCMA 2007*, Athens, Greece, September 2007.
- [99] D. Voudouris, M. Sampson, and G. Papakonstantinou. Exact esct minimization for functions of up to six input variables. *Integration, The VLSI Journal (Elsevier)*, 41(1):87–105, January 2008.
- [100] D. Voudouris, M. Sampson, and G. Papakonstantinou. Finding minimal esct expressions for boolean functions with weight up to 7. In *2008 Intrn. Conf. on Computer Design (CDES08)*, Las Vegas, Nevada, USA, July 14-17 2008.

- [101] D. Voudouris, S. Stergiou, and G. Papakonstantinou. Minimization of reversible wave cascades. *IEICE Trans. on Fundamentals*, E88-A(4):1015–1023, April 2005.
- [102] Haomin Wu, M.A. Perkowski, Xiaoqiang Zeng, and Nan Zhuang. Generalized partially-mixed-polarity reed-muller expansion and its fast computation. In *IEEE Transactions on Computers*, volume 45 of 9, pages 1084–1088, September 1996.
- [103] S. Yang. Logic synthesis and optimization benchmarks user guide version 3.0. Technical report, Microelectronics Center of North Carolina, 1991.
- [104] Y. Ye and K. Roy. Efficient synthesis of and/xor networks. In *Asia and South Pacific Design Automation Conference 1997 (ASP-DAC '97)*, pages 539–544, Chiba, Japan, January 1997.
- [105] Y. Ye and K. Roy. A graph-based synthesis algorithm for and/xor networks. In *Design Automation Conference*, pages 107–112, 1997.
- [106] X. Zeng, M. Perkowski, K. Dill, and A. Sarabi. Approximate minimization of generalized reed-muller forms. In *Proc. Reed-Muller'95*, pages 221–230, 1995.
- [107] X. Zeng, M. Perkowski, H. Wu, and A. Sarabi. A new exact algorithm for highly testable generalized partially-mixed-polarity reed-muller forms.