



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**“Web-based floorplanning/placement
visualizer/analyzer”**

Κωνσταντίνος Μαντός

Επιβλέπων

Γεώργιος Σταμούλης

Συνεπιβλέπων

Αντώνιος Δαδαλιάρης

Λαμία, 2019

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ ΚΑΙ ΛΟΓΟΚΛΟΠΗΣ

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις⁽¹⁾, που προβλέπονται από τις διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί χωρίς να τα περικλείω σε εισαγωγικά και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία ή/και τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά, χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί παράθεση χωρίς εισαγωγικά, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια.
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

⁽¹⁾ “Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.”

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τους επιβλέπων και συνεπιβλέπων καθηγητές για την εμπιστοσύνη που μου έδειξαν, την υπομονή που έκαναν και για την πολύτιμη βοήθεια και καθοδήγηση κατά τη διάρκεια εκπόνησης της πτυχιακής μου εργασίας.

Θα ήθελα επίσης να απευθύνω τις ευχαριστίες μου στους γονείς μου, οι οποίοι στήριξαν τις σπουδές μου με διάφορους τρόπους, φροντίζοντας για την καλύτερη δυνατή μόρφωσή μου.

Τέλος, να ευχαριστήσω την αδερφή μου και τους φίλους μου που ήταν πάντα δίπλα μου για να με υποστηρίξουν και να με βοηθήσουν στις δύσκολες στιγμές.

ΠΕΡΙΛΗΨΗ

Το αντικείμενο που πραγματεύεται η παρούσα πτυχιακή εργασία είναι η δημιουργία μιας διαδικτυακής εφαρμογής με σκοπό την προσομοίωση και ανάλυση κυκλωμάτων των σταδίων του Floorplan και Placement της Φυσικής σχεδίασης Ολοκληρωμένων Κυκλωμάτων. Θα αναλυθούν στοιχεία σχετικά με τα Ολοκληρωμένα Κυκλώματα, τα στοιχεία από τα οποία αποτελούνται και την διαδικασία παραγωγής τους. Επίσης, θα γίνει ανάλυση των διαδικτυακών εφαρμογών, των σταδίων παραγωγής και των γλωσσών προγραμματισμού που χρησιμοποιήθηκαν. Τέλος, θα αναλυθούν τα χαρακτηριστικά και οι δυνατότητες του εργαλείου WEVIAN και θα παρουσιαστούν αποτελέσματα από πειράματα που διεξήχθησαν.

ABSTRACT

The subject of this thesis is the creation of a web application with the purpose to simulate and analyze circuits at Floorplan and Placement stages of Physical design of integrated circuits. Information on integrated circuits, their constituents and their production process will be analyzed. Information on web applications, production stages and programming languages used will also be analyzed. Finally, the features and capabilities of the WEVIAN tool will be analyzed and results from experiments performed will be presented.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ ΚΑΙ ΛΟΓΟΚΛΟΠΗΣ	2
ΕΥΧΑΡΙΣΤΙΕΣ.....	3
ΠΕΡΙΛΗΨΗ	4
ABSTRACT	5
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	6
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	7
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	9
ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ.....	10
ΚΕΦΑΛΑΙΟ 1.....	11
1.1 ΟΛΟΚΛΗΡΩΜΕΝΟ ΚΥΚΛΩΜΑ	11
1.2 ΣΧΕΔΙΑΣΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΚΥΚΛΩΜΑΤΩΝ	12
1.3 ΕΡΓΑΛΕΙΑ CAD ΣΤΗ ΦΥΣΙΚΗ ΣΧΕΔΙΑΣΗ.....	16
ΚΕΦΑΛΑΙΟ 2.....	17
2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ WORLD WIDE WEB	17
2.2 ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΦΑΡΜΟΓΕΣ.....	18
2.3 ΔΟΜΗ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ.....	20
2.4 ΚΑΤΑΣΚΕΥΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ WEBIAN	23
2.4.1 Γλώσσες προγραμματισμού	23
2.4.2 Βιβλιοθήκες.....	36
2.4.3 Τεχνικά χαρακτηριστικά.....	37
ΚΕΦΑΛΑΙΟ 3.....	39
3.1 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ	39
3.1.1 Floorplan.....	41
Προσομοίωση.....	41
Διαμοιρασμός.....	50
3.1.2 Placement	51
Προσομοίωση.....	51
Σύγκριση	58
Διαμοιρασμός.....	62
3.2 ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	63
3.2.1 Βασικά χαρακτηριστικά.....	63
3.2.2 Πειραματικά αποτελέσματα	65
Πειράματα Placement.....	65
Πειράματα Floorplan.....	73
Συμπέρασμα	75
ΚΕΦΑΛΑΙΟ 4.....	77
4.1 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	77
4.2 ΕΠΕΚΤΑΣΙΜΟΤΗΤΑ	77
ΒΙΒΛΙΟΓΡΑΦΙΑ	79

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Ροή σχεδίασης Ολοκληρωμένων Κυκλωμάτων. ^[5]	12
Εικόνα 2: Η πρώτη σελίδα της πρότασης του Tim Berners-Lee για το World Wide Web. ^[15]	17
Εικόνα 3: Βασική δομή μιας σελίδας HTML. ^[28]	24
Εικόνα 4: Γενική δομή των CSS κανόνων.....	26
Εικόνα 5: Τρόποι εισαγωγής CSS στην HTML.....	27
Εικόνα 6: Τρόποι εισαγωγής JavaScript στην HTML.....	29
Εικόνα 7: Κατανομή ιστοσελίδων μεταξύ των διαφορετικών εκδόσεων PHP. ^[35]	31
Εικόνα 8: Μεταφορά δεδομένων μεταξύ PHP και JavaScript χρησιμοποιώντας JSON. ^[36]	35
Εικόνα 9: Αρχική σελίδα.....	40
Εικόνα 10: Επιλογή μεταξύ των project.....	41
Εικόνα 11: Μενού επιλογών Floorplan Visualization & Analysis.....	41
Εικόνα 12: Αποθηκευμένα project στο μενού.....	42
Εικόνα 13: Τμήμα προσομοίωσης 2 διαφορετικών κυκλωμάτων.....	43
Εικόνα 14: Τμήμα προσομοίωσης και ενεργειών.....	43
Εικόνα 15: Επιλεγμένα nets σε διαφορετικούς χρωματισμούς.....	44
Εικόνα 16: Πίνακας πληροφοριών των block.....	44
Εικόνα 17: Resized blocks.....	45
Εικόνα 18: Υπολογισμός συνολικού μήκους καλωδίου.....	46
Εικόνα 19: Τετράγωνα blocks σε τυχαία τοποθεσία.....	47
Εικόνα 20: Υπολογισμός whitespace.....	47
Εικόνα 21: Επικαλυπτόμενα blocks.....	48
Εικόνα 22: Επεξεργασία θερμοκρασιών.....	49
Εικόνα 23: Δημιουργία και απεικόνιση θερμικού χάρτη.....	49
Εικόνα 24: Υπόμνημα προσομοίωσης.....	49
Εικόνα 25: Ειδοποίηση ολοκλήρωσης αποθήκευσης.....	50
Εικόνα 26: Μενού επιλογών Placement Visualization & Analysis.....	51
Εικόνα 27: Αποθηκευμένα project στο μενού.....	52
Εικόνα 28: Τμήμα προσομοίωσης 2 διαφορετικών κυκλωμάτων.....	52
Εικόνα 29: Τμήμα προσομοίωσης και ενεργειών.....	53
Εικόνα 30: Αριστερά με εμφάνιση σειρών. Δεξιά χωρίς εμφάνιση σειρών.....	54

Εικόνα 31: Υπολογισμός συνολικού μήκους καλωδίου.	55
Εικόνα 32: Συγκεντρωτικός πίνακας στοιχείων μετά το πέρας της ανάλυσης του κυκλώματος.....	56
Εικόνα 33: Overlap, overflow, pad και κινητά cell.	56
Εικόνα 34: Γράφημα θερμοκρασιών.....	57
Εικόνα 35: Γράφημα χάρτη συμφόρησης.	57
Εικόνα 36: Γράφημα χρονικών επιβαρύνσεων των λειτουργιών.	58
Εικόνα 37: Το πρώτο μέρος του πάνελ σύγκρισης.	59
Εικόνα 38: Το πρώτο μέρος του πάνελ σύγκρισης με 3 net highlighted.	59
Εικόνα 39: Το δεύτερο μέρος του πάνελ σύγκρισης.	60
Εικόνα 40: Το τρίτο μέρος του πάνελ σύγκρισης.	60
Εικόνα 41: Γράφημα on-the-fly Overlap 2 κυκλωμάτων στο τέταρτο μέρος του πάνελ σύγκρισης.....	61
Εικόνα 42: Γράφημα on-the-fly Overflow 2 κυκλωμάτων στο τέταρτο μέρος του πάνελ σύγκρισης.....	61
Εικόνα 43: Γράφημα on-the-fly Half-Perimeter 2 κυκλωμάτων στο τέταρτο μέρος του πάνελ σύγκρισης.	62

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Κατηγορίες Ολοκληρωμένων Κυκλωμάτων ως προς το πλήθος των τρανζίστορ.[3].....	11
Πίνακας 2: Πίνακας αληθείας της λογικής πύλης “AND”. [8]	14
Πίνακας 3: Τα HTML tags που χρησιμοποιήθηκαν στο εργαλείο WEVIAN.....	25
Πίνακας 4: Τα CSS properties που χρησιμοποιήθηκαν στο εργαλείο WEVIAN.	28
Πίνακας 5: Βασικές εντολές SQL.....	35
Πίνακας 6: Τεχνικά χαρακτηριστικά του Server.	37
Πίνακας 7: Εκδόσεις γλωσσών προγραμματισμού που χρησιμοποιήθηκαν.	38
Πίνακας 8: Εκδόσεις βιβλιοθηκών που χρησιμοποιήθηκαν.....	38
Πίνακας 9: Στοιχεία και Ενέργειες αποθηκευμένων προσομοιώσεων.	42
Πίνακας 10: Στοιχεία και Ενέργειες αποθηκευμένων προσομοιώσεων.	52
Πίνακας 11: Χαρακτηριστικά ISPD’04 benchmark circuits.[40].....	63
Πίνακας 12: Χαρακτηριστικά VLSI CAD Bookshelf.[41]	64
Πίνακας 13: Χαρακτηριστικά υπολογιστικού συστήματος.....	64

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 1: Μέσος χρόνος rendering προσομοίωσης από δείγμα 10 προσομοιώσεων ενός chip με 12752 cells συναρτήσει διαφορετικών μεθόδων indexing που χρησιμοποιήθηκαν από διαφορετικές εκδόσεις του εργαλείου WEVIAN.....	30
Διάγραμμα 2: Μέσος χρόνος overhead Overlap & Overflow συναρτήσει διαφορετικών browser.....	66
Διάγραμμα 3: Μέσος χρόνος overhead Thermal & Congestion Map συναρτήσει διαφορετικών browser.	67
Διάγραμμα 4: Μέσος χρόνος overhead Half-Perimeter Wire Length συναρτήσει διαφορετικών browser.	68
Διάγραμμα 5: Μέσος χρόνος overhead Congestion Map σε διαφορετικούς browser συναρτήσει διαφορετικών αριθμών bin.....	69
Διάγραμμα 6: Μέσος χρόνος overhead Overlap όλων των κυκλωμάτων σε διαφορετικούς browser.....	69
Διάγραμμα 7: Μέσος χρόνος overhead Overflow όλων των κυκλωμάτων σε διαφορετικούς browser.	70
Διάγραμμα 8: Μέσος χρόνος overhead Thermal Map όλων των κυκλωμάτων σε διαφορετικούς browser.	71
Διάγραμμα 9: Μέσος χρόνος overhead Congestion Map όλων των κυκλωμάτων σε διαφορετικούς browser.	71
Διάγραμμα 10: Μέσος χρόνος overhead Half-Perimeter Wire Length όλων των κυκλωμάτων σε διαφορετικούς browser.....	72
Διάγραμμα 11: Μέσος χρόνος overhead Visualization όλων των κυκλωμάτων σε διαφορετικούς browser.	73
Διάγραμμα 12: Μέσος χρόνος overhead Half-Perimeter Wire Length & Whitespace συναρτήσει διαφορετικών browser.....	74
Διάγραμμα 13: Μέσος χρόνος overhead Export to PNG σε διαφορετικούς browser συναρτήσει διαφορετικών αριθμών block.	75

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Ολοκληρωμένο Κύκλωμα

Τα πρώτα τρανζίστορ που εφευρέθηκαν χρησιμοποιούνταν το κάθε ένα ξεχωριστά και η ένωσή τους γινόταν με καλώδιο του οποίου η σύνδεση με τα τρανζίστορ ήταν χειροκίνητη. Τα Ολοκληρωμένα Κυκλώματα έλυσαν αυτό το πρόβλημα ενώνοντας πολλαπλά τρανζίστορ στο ίδιο κομμάτι ημιαγωγού υλικού που κατά κύριο λόγο είναι πυρίτιο. Η σύνδεση αυτών, επιτεύχθηκε με μικροσκοπικά καλώδια τα οποία είναι “χαραγμένα” στην επιφάνεια του ημιαγωγού υλικού. Το κομμάτι ημιαγωγού υλικού ονομάζεται wafer ενώ το σύνολο του Ολοκληρωμένου Κυκλώματος ονομάζεται chip. Πλέον εκτός από τα τρανζίστορ, πάνω σε ένα Ολοκληρωμένο Κύκλωμα περιλαμβάνονται αντιστάτες, πυκνωτές, επαγωγείς και δίοδοι.^[1]

Τα Ολοκληρωμένα Κυκλώματα αρχικά χρησιμοποιήθηκαν για στρατιωτικούς λόγους.^[2] Στην συνέχεια η τεχνολογία τους υιοθετήθηκε και από εμπορικές εταιρίες με αποτέλεσμα σήμερα να χρησιμοποιούνται ευρέως σε όλα τα ηλεκτρονικά αντικείμενα.^[1] Αυτό συμβαίνει διότι η τιμή ανά στοιχείο συνεχώς ελαττώνεται όσο αυξάνονται τα στοιχεία ανά κύκλωμα.

Τα Ολοκληρωμένα Κυκλώματα μπορεί να περιλαμβάνουν εκατομμύρια ή και δισεκατομμύρια τρανζίστορ πάνω στο chip. Διαχωρίζονται σε κατηγορίες ανάλογα με το πλήθος των τρανζίστορ που περιέχουν, όπως φαίνεται στον πίνακα 1.

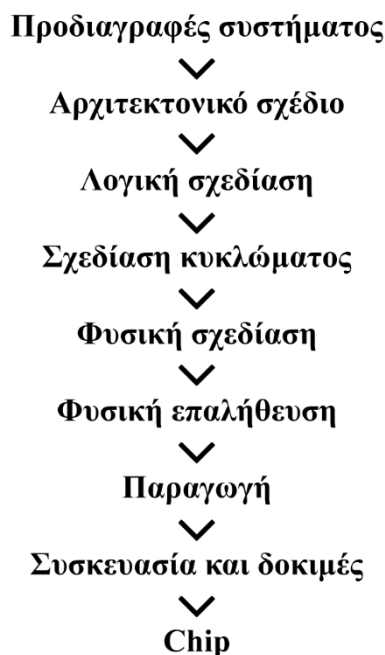
Όνομα	Σημασία	Πλήθος τρανζίστορ
SSI	Small-scale integration	1 με 100
MSI	Medium-scale integration	100 με 1.000
LSI	Large-scale integration	1.000 με 10.000
VLSI	Very large-scale integration	10.000 με 1.000.000
ULSI	Ultra large-scale integration	Άνω των 1.000.000

Πίνακας 1: Κατηγορίες Ολοκληρωμένων Κυκλωμάτων ως προς το πλήθος των τρανζίστορ.[3]

1.2 Σχεδίαση Ολοκληρωμένων Κυκλωμάτων

Η σχεδίαση Ολοκληρωμένων Κυκλωμάτων εξαρτάται από το μέγεθος του επιθυμητού κυκλώματος. Για μικρά Ολοκληρωμένα Κυκλώματα, η διαδικασία σχεδίασης μπορεί να γίνει ακόμη και από ένα άτομο. Όσο μεγαλώνει το μέγεθος και επομένως η πολυπλοκότητα που απαιτείται για την σχεδίαση τους τόσο αυξάνεται και ο αριθμός των ατόμων και των διεργασιών που πρέπει να ολοκληρωθούν πριν αυτά βγουν στην παραγωγή. Για σύνθετα Ολοκληρωμένα Κυκλώματα, όπως αυτά των κεντρικών μονάδων επεξεργασίας ηλεκτρονικών υπολογιστών, μέρη της σχεδίασης αυτοματοποιούνται από ειδικά εργαλεία τα οποία προγραμματίζονται από ομάδες σχεδιαστών κυκλωμάτων. Η διαδικασία σχεδίασης ακολουθεί μια προσέγγιση από καθοδηγούμενες ειδικές προσομοιώσεις ώστε να ανακαλυφθούν λάθη όσο δυνατόν πιο νωρίς.^[4]

Επομένως, για να υπάρξει καλύτερο αποτέλεσμα κατά την παραγωγή Ολοκληρωμένων Κυκλωμάτων ακολουθείται μια σειρά από βήματα που παρουσιάζονται στην Εικόνα 1. Τα βήματα αυτά διαμορφώνονται ανάλογα με τις συνθήκες και τις απαιτήσεις που μπορεί να υπάρχουν. Στις επόμενες παραγράφους θα εξηγηθεί συνοπτικά κάθε βήμα της ροής σχεδίασης.



Εικόνα 1: Ροή σχεδίασης Ολοκληρωμένων Κυκλωμάτων.^[5]

Στο στάδιο των προδιαγραφών συστήματος γίνεται η περιγραφή της επιθυμητής απόδοσης του συστήματος και επιλύονται ασυμβατότητες των συστημάτων που

προέρχονται από διαφορετικούς κατασκευαστές. Επιπρόσθετα, πραγματοποιείται η παραμετροποίηση όλων των εξαρτημάτων του συστήματος.^{[4][5]} Οι προδιαγραφές συστήματος δεν είναι ίδιες σε όλες τις παραγωγές. Η εταιρεία που ενδιαφέρεται για την παραγωγή κάποιων κυκλωμάτων είναι υπεύθυνη για τον ορισμό των προδιαγραφών.

Στο στάδιο του αρχιτεκτονικού σχεδιασμού γίνεται η περιγραφή των βασικών αρχών από τις οποίες διέπονται η κατασκευή, οι στόχοι και ο σκοπός του Ολοκληρωμένου Κυκλώματος. Οι αρχιτεκτονικές ομάδες συνεργάζονται με πολλές άλλες ομάδες και λαμβάνουν υπόψη τους όλες τις πληροφορίες που προέκυψαν από τη προδιαγραφή συστήματος για την δημιουργία του αρχιτεκτονικού σχεδίου. Έπειτα υπάρχει η διαδικασία της μικρο-αρχιτεκτονικής όπου βάση του αρχιτεκτονικού σχεδίου προσδιορίζονται οι μηχανισμοί και οι υποδομές για την υλοποίηση του κυκλώματος. Το αποτέλεσμα της μικρο-αρχιτεκτονικής θα είναι το σημείο εκκίνησης της σχεδίασης.^[5]

Στο στάδιο της λογικής σχεδίασης χρησιμοποιείται η άλγεβρα Boole για να περιγράψει τη λειτουργία του κυκλώματος. Η άλγεβρα Boole πήρε το όνομά της από τον Άγγλο αυτοδίδακτο μαθηματικό George Boole ο οποίος το 1854 δημοσίευσε βιβλίο που περιείχε την άλγεβρα Boole για πρώτη φορά, το *The Laws of Thought*.^[6] Η άλγεβρα Boole είναι κομμάτι της άλγεβρας όπου οι μεταβλητές έχουν δύο πιθανές τιμές, αληθές και ψευδές. Στα ηλεκτρονικά κυκλώματα αυτές οι τιμές είναι το 1 και το 0, όπου 1 είναι το αληθές και το κύκλωμα είναι κλειστό και 0 το ψευδές όπου το κύκλωμα είναι ανοιχτό.^[7] Ακόμη, στη λογική σχεδίαση χρησιμοποιούνται έννοιες όπως οι λογικές πύλες ή απλά πύλες. Οι λογικές πύλες είναι τα βασικά στοιχεία ενός ψηφιακού συστήματος και είναι κυκλώματα με ένα ή περισσότερα σήματα εισόδου αλλά μόνο ένα σήμα εξόδου. Συνδέοντας τις λογικές πύλες με διαφορετικούς τρόπους δημιουργούνται κυκλώματα που εκτελούν συγκεκριμένες λειτουργίες. Οι βασικοί τύποι πυλών είναι τρεις. Είναι η πύλη “AND” που δηλώνει σύζευξη, η πύλη “OR” που δηλώνει διάζευξη και η πύλη “NOT” για να δηλώσει άρνηση. Από το συνδυασμό των παραπάνω πυλών δημιουργούνται όλες οι άλλες πύλες και κατ’ επέκταση τα κυκλώματα. Οι λειτουργίες των λογικών πυλών γίνονται εύκολα κατανοητές με τη βοήθεια των πινάκων αληθείας. Πίνακας αληθείας είναι ένας πίνακας που περιλαμβάνει όλες τις πιθανές εισόδους και εξόδους από ένα λογικό κύκλωμα, δείχνει δηλαδή το αποτέλεσμα της εξόδου για κάθε δυνατή είσοδο.^[8]

Είσοδος 1	Είσοδος 2	Έξοδος
0	0	0
0	1	0
1	0	0
1	1	1

Πίνακας 2: Πίνακας αληθείας της λογικής πύλης “AND”. [8]

Στο στάδιο της σχεδίασης κυκλώματος αναπαρίσταται το κύκλωμα βάση της περιγραφής του στη λογική σχεδίαση, δηλαδή η άλγεβρα Boole αναπαρίσταται ως κύκλωμα λαμβάνοντας υπόψη τις προδιαγραφές συστήματος. Όλες οι αναπαραστάσεις γίνονται σχηματικά σε μορφή διαγραμμάτων όπου φαίνονται όλες οι συνδέσεις των στοιχείων του κυκλώματος. Το σύνολο των συνδέσεων αυτών ονομάζεται netlist.^{[7][9]}

Στο στάδιο της φυσικής σχεδίασης το netlist αναπαρίσταται γεωμετρικά πάνω στο layout, δηλαδή σε ένα προκαταρκτικό σχέδιο το οποίο έπειτα θα αποτελέσει τον χώρο του chip και ονομάζεται chip area.^[9] Το layout περιέχει όλα τα υποκυκλώματα συνδεδεμένα μεταξύ τους σε πολλά στρώματα. Οι συνδέσεις αναπαρίστανται αρχικά σαν γραμμές που ενώνουν όλα τα συμβαλλόμενα μέρη μεταξύ τους. Επειδή όμως το αποτέλεσμα είναι δυσνόητο και σύνθετο, το στάδιο της φυσικής σχεδίασης διασπάται σε μικρότερα στάδια. Από αυτά, τα σημαντικότερα είναι το “Floorplanning”, το “Placement” και το “Routing”.

Το Floorplanning αφορά κυρίως την τοποθέτηση των pin εισόδου/εξόδου και την τοποθέτηση των μεγάλων αντικειμένων πάνω στο layout, τέτοια αντικείμενα είναι για παράδειγμα οι πυρήνες.

Το Placement αφορά την χωροθέτηση των πυλών πάνω στο layout σε συγκεκριμένες θέσεις χωρίς να γίνεται επικάλυψη με άλλα στοιχεία ή υπερχειλίση.^[5] Οι πύλες σε αυτό στο Placement συνήθως αναφέρονται ως cell. Για να γίνει σωστά το Placement ακολουθούνται 3 βήματα. Το Global placement, το Legalization και το Detailed placement.

1. Το Global placement αφορά την αποδοτικότερη τοποθέτηση των cell πάνω στο chip ακόμα και αν πρέπει να παραβαίνουν κάποιους κανόνες όπως επικάλυψη με άλλα cell.

2. Το Legalization αφορά την επίλυση τέτοιων προβλημάτων, δηλαδή τα cell κατατάσσονται με τέτοιο τρόπο ώστε να αποφεύγονται οι επικαλύψεις ή υπερχειλίσεις καθώς και θέτουν τα cell σε σειρές πάνω στο chip.
3. Το Detailed placement αφορά την μετακίνηση κάποιων cell για να γίνει αποδοτικότερο το κύκλωμα.

Σκοπός των 3 βημάτων είναι τα cell να τοποθετηθούν κατάλληλα ώστε να γίνεται καλύτερη διαχείριση του χώρου και ελαχιστοποίηση του συνολικού μήκους του καλωδίου που θα χρησιμοποιηθεί για την ένωση των στοιχείων του netlist.

Το Routing αφορά την τοποθέτηση των καλωδίων που απαιτούνται για την ένωση των τοποθετημένων στοιχείων.^[9] Τα βήματα της φυσικής σχεδίασης γίνονται επαναληπτικά ώστε όλα τα σφάλματα να διορθωθούν.

Στο στάδιο της φυσικής επαλήθευσης το κύκλωμα περνάει μία σειρά από ελέγχους πριν βγει στην παραγωγή. Οι έλεγχοι γίνονται επαναληπτικά και περιλαμβάνουν επιπρόσθετες διορθώσεις σε κάθε επανάληψη χρησιμοποιώντας διάφορους τύπους ελέγχου.^[10]

Μετά το πέρας της φυσικής επαλήθευσης το κύκλωμα υλοποιείται χρησιμοποιώντας μια τεχνική που ονομάζεται φωτολιθογραφία. Στη φωτολιθογραφία πέρα από τα υπάρχοντα στρώματα και στοιχεία του κυκλώματος, προστίθενται, τροποποιούνται ή αφαιρούνται διάφορα υλικά. Κάθε ένα στρώμα διαχωρίζεται και μετατρέπεται σε φωτολιθογραφική μάσκα. Στη συνέχεια, ένα φύλλο πυριτίου επικαλύπτεται από μια φωτοευαίσθητη ουσία ώστε το φως να το αλλοιώνει.^[10] Έτσι, η μάσκα επιτρέπει να πέσει το φως μόνο στα σημεία που χρειάζεται και εκεί χημικά αλλοιώνεται, δημιουργώντας το πρώτο επίπεδο στο chip. Η διαδικασία συνεχίζεται με την επόμενη μάσκα. Με αυτό τον τρόπο, τα χαρακτηριστικά του chip ενσωματώνονται στο πυρίτιο, ανά επίπεδο.

Στο στάδιο της συσκευασίας και δοκιμών, τα παραχθέντα chip συσκευάζονται και ελέγχονται για ακόμη μια φορά οι προδιαγραφές συστήματος και η λειτουργικότητά τους. Τα πλήρως λειτουργικά Ολοκληρωμένα Κυκλώματα πλέον συσκευάζονται σε μεγαλύτερες συσκευασίες και είναι έτοιμα για την εξαγωγή στην αγορά.

1.3 Εργαλεία CAD στη Φυσική σχεδίαση

Εξαιτίας του γιγαντιαίου αριθμού των στοιχείων που περιλαμβάνει ένα chip και των πληροφοριών που απαιτούνται κατά την παραγωγή τους, η διαδικασία της φυσικής σχεδίασης δεν μπορεί να γίνει χωρίς την χρήση ηλεκτρονικών υπολογιστών. Όλα τα βήματα της ροής σχεδίασης χρειάζονται την μέγιστη απόδοση και αξιοπιστία των ηλεκτρονικών υπολογιστών και των ειδικών προγραμμάτων που αυτοί εκτελούν. Για παράδειγμα, πολλά υπο-στάδια της φυσικής σχεδίασης είναι πλήρως αυτοματοποιημένα κυρίως από Εργαλεία CAD.^[11]

Για να γίνει η αυτοματοποίηση της φυσικής σχεδίασης χρειάζεται εμπειρία και γνώση πάνω στους αλγόριθμους και στις δομές δεδομένων που θα χρησιμοποιηθούν. Ο στόχος είναι να βρεθούν οι ιδανικές θέσεις των στοιχείων στο chip και η συντομότερη διασύνδεση τους. Ο χώρος πάνω στο chip είναι υπερβολικά πολύτιμος, οι αλγόριθμοι τοποθέτησης πρέπει να είναι ικανοί να τον χρησιμοποιήσουν έτσι ώστε να ελαχιστοποιήσουν το κόστος. Επιπλέον, η καλύτερη τοποθέτηση συνεπάγεται και μεγιστοποίηση της απόδοσης του κυκλώματος. Οι αλγόριθμοι κατά την φυσική σχεδίαση, πέρα από την ελαχιστοποίηση του κόστους και μεγιστοποίηση της αποδοτικότητάς τους δεν πρέπει να ξεφεύγουν από τους κανόνες που έχουν τεθεί. Αποτελεσματικοί αλγόριθμοι σημαίνει συντομότερος χρόνος σχεδίασης αλλά και σχεδίαση που δεν είναι επιρρεπής σε λάθη.

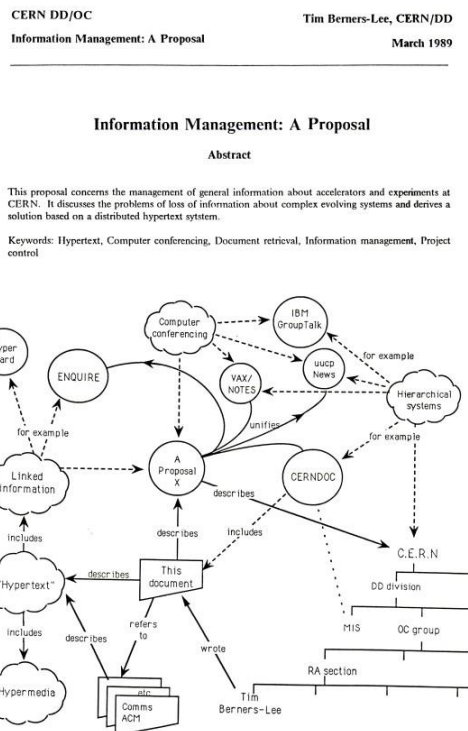
Υπάρχουν πολλά Εργαλεία CAD μέσω των οποίων δημιουργούνται και αξιοποιούνται οι αλγόριθμοι, αυτός ο τομέας είναι γνωστός ως Physical Design Automation. Τα εργαλεία αυτού του τομέα χωρίζονται σε τρεις κατηγορίες.^[9] Στη πρώτη κατηγορία, τα εργαλεία βοηθούν στην διαχείριση του layout. Για παράδειγμα, ο σχεδιαστής μπορεί να βάλει χειροκίνητα τρανζίστορ στο layout. Στη δεύτερη κατηγορία, τα εργαλεία βοηθούν στην αυτόματη εκτέλεση κάποιας λειτουργίας στο layout. Για παράδειγμα, εργαλεία placement και routing ανήκουν εδώ. Στη τρίτη κατηγορία, τα εργαλεία βοηθούν στον έλεγχο και επαλήθευση του κυκλώματος. Για παράδειγμα, τα εργαλεία DRC ελέγχουν αν τηρούνται οι κανόνες σχεδίασης που έχουν τεθεί. Είναι λογικό, τα περισσότερα εργαλεία να ανήκουν στην δεύτερη κατηγορία λόγω της μεγαλύτερης επίπτωσης που έχουν στην φυσική σχεδίαση. Για το λόγο αυτό πολλά εργαλεία παρέχουν λειτουργίες και από τις τρεις κατηγορίες.

ΚΕΦΑΛΑΙΟ 2

Υλοποίηση Διαδικτυακών Εφαρμογών

2.1 Ιστορική αναδρομή World Wide Web

Στις 12 Μαρτίου του 1989 έγινε η πρόταση από τον Tim Berners-Lee, βρετανό μηχανικό και επιστήμονα υπολογιστών, προς τους συναδέλφους του στο CERN για το World Wide Web.^[12] Η πρόταση αυτή δεν είχε μεγάλη απήχηση και αναγνώριση με αποτέλεσμα να ξεκινήσει ο ίδιος την δημιουργία του project. Ένα χρόνο μετά, το 1990, έκανε τη δεύτερη πρότασή του για το World Wide Web μαζί με τον συνάδελφό του, Robert Cailliau, η οποία αφορούσε συγκεκριμένα, ένα δίκτυο από hypertext documents τα οποία μπορούν να προβληθούν από ένα ειδικό λογισμικό που ονομάζεται browser. Επομένως, μέχρι το τέλος του 1990 η ομάδα του Tim Berners-Lee είχε αναπτύξει τον πρώτο web server και browser για να “ενσαρκώσει” την ιδέα του. Ωστόσο ο browser μπορούσε να εμφανίσει τις σελίδες μόνο σε εκείνον τον υπολογιστή.^{[13][14]}



Εικόνα 2: Η πρώτη σελίδα της πρότασης του Tim Berners-Lee για το World Wide Web.^[15]

Οι πρώτες σελίδες του site αφορούσαν το project αυτό καθ' αυτό. Εξηγούσαν την ιδέα του hypertext και λεπτομέρειες για την κατασκευή ενός web server. Το ίδιο έτος η Nicola Pellow γίνεται και αυτή μέλος στο project και κατασκευάζει έναν διαφορετικό browser για να μπορούν οι σελίδες να είναι μεταφέρσιμες και προσβάσιμες από όλους τους υπολογιστές.^{[13][14]} Το 1991 το World Wide Web βρίσκεται στη πρώτη ολοκληρωμένη έκδοση και δίνεται πρόσβαση στο διαδίκτυο σε άτομα εκτός του CERN.^[16] Στην έκδοση περιλαμβάνεται ο browser της Nicola Pellow, λογισμικό για τον web server και μια βιβλιοθήκη για τους προγραμματιστές.^[14]

Κάποιους μήνες αργότερα, τον Αύγουστο του 1991 το project ανακοινώνεται στο Internet newsgroups και τα νέα εξαπλώνονται αστραπιαία σε όλη τη Γη. Έπειτα, το project μεταφέρεται και στην Αμερική λόγω του ενδιαφέροντος για αυτό από τον φυσικό και προγραμματιστή Paul Kunz.^[17] Λίγα χρόνια μετά, το 1994, ο Tim Berners-Lee φεύγει από το CERN και πηγαίνει στο MIT για να συνεχίσει την ανάπτυξη του HTTP.^[14]

Οι χρήστες του παγκόσμιου ιστού αυξάνονται συνεχόμενα, από έναν το 1990 σε 10 εκατομμύρια στο τέλος του 1994. Προς το παρόν, όμως, δεν υπάρχουν μηχανές αναζήτησης και οι σελίδες είναι δυσεύρετες. Επομένως, εμφανίζονται indexing σελίδες που καταγράφουν πολλές διευθύνσεις σε λίστες.

Στο τέλος της δεκαετίας του 1990 ο παγκόσμιος ιστός χρησιμοποιείται και από εταιρείες για εμπορικούς λόγους. Μετά το 2002 ο παγκόσμιος ιστός σηματοδοτεί την εποχή της πληροφορίας, μηχανές αναζήτησης και εμπορικές διαφημίσεις κάνουν τα πρώτα τους βήματα. Εταιρείες τηλεπικοινωνιών και ειδησεογραφικά πρακτορεία δρουν μέσα από το παγκόσμιο ιστό.^[13] Τέλος, δημιουργούνται νέες ανάγκες και διαδικτυακές εφαρμογές έρχονται να τις καλύψουν. Η δημιουργία του Facebook το 2004, του YouTube το 2005 και πολλών άλλων διαδικτυακών εφαρμογών ενώνουν τους χρήστες διαδικτυακά χωρίς την ανάγκη φυσικής παρουσίας ή περιορισμούς απόστασης.

2.2 Διαδικτυακές εφαρμογές

Η κατηγορία των διαδικτυακών εφαρμογών περιλαμβάνει τις εφαρμογές που για την εκτέλεσή τους χρειάζονται πρόσβαση στο διαδίκτυο μέσα από έναν client που συνήθως είναι ο browser.^[18] Ωστόσο, δεν είναι αναγκαίος, πολλές εφαρμογές χρησιμοποιούν το διαδίκτυο χωρίς κάποιο browser, όπως οι desktop εφαρμογές. Γενικά, διαδικτυακή

εφαρμογή είναι η εφαρμογή ή ιστοσελίδα που αλληλοεπιδρά με το χρήστη, είναι δυναμική, συλλέγει δεδομένα και εμφανίζει πληροφορίες χρησιμοποιώντας το διαδίκτυο.^[19]

Οι διαδικτυακές εφαρμογές βασίζονται πάνω στο περιβάλλον client-server, κατά το οποίο ο client επικοινωνεί με το server αμφίδρομα για την μεταφορά ή αποθήκευση πληροφοριών, για παράδειγμα κατά την μεταφορά μιας εικόνας από έναν χρήστη σε έναν άλλο.^[20] Η εικόνα μεταφέρεται στο server, από εκεί μεταφέρεται στο δεύτερο χρήστη ο οποίος την λαμβάνει μέσα από το δικό του client, δηλαδή υπάρχει μεταφορά πληροφορίας από client στο server αλλά και από το server στο client.

Χάρη στη ραγδαία ανάπτυξη της τεχνολογίας υπάρχει ανάπτυξη και στην αρχιτεκτονική των διαδικτυακών εφαρμογών. Οι εφαρμογές δεν βασίζονται εξολοκλήρου στο concept του client-server, μπορούν να αλληλοεπιδρούν με το χρήστη, να διαχειρίζονται τα αρχεία του, να μεταφέρουν πληροφορίες και να αντιλαμβάνονται ακόμη και τις προσωπικές του απόψεις ή αρέσκεις. Το τελευταίο, μάλιστα, αποτελεί πεδίο της τεχνητής νοημοσύνης^[21] και έχει αποκτήσει ιδιαίτερο ενδιαφέρον τα τελευταία χρόνια. Με τις πρώτες υπερ-εφαρμογές να εμφανίζονται τη δεκαετία του 2010 και να θεωρούν ότι έχουν υλοποιήσει ένα είδος τεχνητής νοημοσύνης ικανής να επεξεργάζεται και να διαχειρίζεται γιγάντιο όγκο δεδομένων σε πραγματικό χρόνο.

Οι διαδικτυακές εφαρμογές χρησιμοποιούνται κυρίως για:

- Ηλεκτρονικό εμπόριο.^[22] Το μεγαλύτερο μέρος των διαδικτυακών εφαρμογών υπάρχει για την πώληση αγαθών ή υπηρεσιών ηλεκτρονικά. Οι πελάτες μπορούν σε ελάχιστο χρόνο να πραγματοποιήσουν τις αγορές τους διαδικτυακά χωρίς καθυστερήσεις. Στη σημερινή εποχή τα ηλεκτρονικά καταστήματα περιέχουν όλες τις λεπτομέρειες για κάθε ένα προϊόν χωρίς την ανάγκη για αναμονή σε σειρές ή χρονοβόρες μετακινήσεις μεταξύ καταστημάτων. Ηλεκτρονικές πληρωμές αλλά και επιστροφές γίνονται με ασφάλεια και αξιοπιστία. Πλατφόρμες όπως το eBay και το Amazon αποτελούν ηγέτιδες στο χώρο του ηλεκτρονικού εμπορίου καθώς επενδύουν συνεχώς στις διαδικασίες εξέλιξης, διαφήμισης, πώλησης, μεταφοράς, εξυπηρέτησης και εξόφλησης των προϊόντων. Στο ηλεκτρονικό εμπόριο ανήκουν και οι εφαρμογές πώλησης υπηρεσιών ή διαφημίσεων όπως κοινωνικά δίκτυα.
- Κυβερνητικές εφαρμογές.^[22] Αποπληρωμή των φόρων ή παραλαβή επιδομάτων γίνονται με αιτήσεις σε κυβερνητικές εφαρμογές. Μέσω των κυβερνητικών εφαρμογών

οι πολίτες μπορούν να εκδώσουν πιστοποιητικά, να συμπληρώσουν αιτήσεις ή φόρμες, να επικοινωνήσουν άμεσα με κρατικές υπηρεσίες ή να ενημερωθούν για θέματα που τους αφορούν χωρίς χρονικές καθυστερήσεις.

- Επιστημονικές εφαρμογές. Σε αυτή τη κατηγορία εμπίπτουν οι εφαρμογές που σκοπό έχουν να λύσουν ένα πρόβλημα ή να παράγουν πληροφορίες ανάλογα με τα εισαχθέντα δεδομένα. Παραδείγματα, είναι, διαδικτυακές εφαρμογές λεξικών, μετάφρασης κειμένων, εγκυκλοπαιδειών, μετατροπείς συναλλαγματικής ισοτιμίας νομισμάτων, εκπαιδευτικά εργαλεία κ.α.

2.3 Δομή διαδικτυακών εφαρμογών

Η κατασκευή μιας διαδικτυακής εφαρμογής δεν είναι απλή. Πρέπει να ακολουθηθεί μια σειρά από βήματα για να εξασφαλιστεί η μέγιστη σταθερότητα στην ανάπτυξή της και να ελαχιστοποιηθούν οι πιθανότητες αποτυχιών ή λαθών. Σε μια εμπορική διαδικτυακή εφαρμογή το παραμικρό λάθος μπορεί να έχει μεγάλο αντίκτυπο στον προϋπολογισμό της.

Τα κύρια βήματα που ακολουθούνται από τη πλειοψηφία των κατασκευαστών είναι 8 και χωρίζονται σε 4 στάδια.^[23]

Στάδιο 1: Η Ιδέα

1. Λήψη ιδέας. Το πιο σημαντικό κομμάτι για την δημιουργία μιας επιτυχημένης διαδικτυακής εφαρμογής είναι η ιδέα αυτή καθ' αυτή. Ποιος είναι ο λόγος και σε τι εξυπηρετεί αυτή η εφαρμογή είναι 2 βασικά ερωτήματα που πρέπει να απαντηθούν. Οι επιτυχείς ιδέες πρέπει να ικανοποιούν ανθρώπινες ανάγκες ή να δημιουργούν νέες. Η λήψη της αρχικής ιδέας μπορεί να αποδειχθεί ιδιαίτερα δύσκολη λόγω του εύρους των διαθέσιμων εφαρμογών.
2. Έρευνα αγοράς. Συλλογή πληροφοριών για την οπτική γωνία των πελατών και των αναγκών που έχουν. Οργάνωση όλων των διαθέσιμων πληροφοριών για την δημιουργία μιας στρατηγικής ικανής να πουλήσει το “προϊόν”. Η σημερινή έρευνα αγοράς δεν είναι απλά η αναζήτηση παρόμοιων εφαρμογών, είναι το σύνολο όλων εκείνων των ενεργειών για την εκτενή ανάλυση του πληθυσμού της αγοράς. Κατά την έρευνα αγοράς πρέπει να ληφθούν υπόψη:
 - Trends – Αναζήτηση στα charts των κορυφαίων εφαρμογών θα βοηθήσει στην εύρεση επιτυχών εφαρμογών του ίδιου τομέα.

- Search Engine Optimization – Οι τεχνικές SEO θα κάνουν την διαδικτυακή εφαρμογή ευδιάκριτη και εύκολα προσβάσιμη από τις μηχανές αναζήτησης.
 - Social Media – Έρευνα στα δίκτυα κοινωνικής δικτύωσης βοηθάει στην παρουσίαση της ιδέας και λήψη feedback.
3. Καθορισμός λειτουργικότητας. Αφού έχει γίνει λήψη της ιδέας και υπάρχει γνώση των αναγκών της αγοράς, το επόμενο βήμα είναι ο καθορισμός της θεωρητικής λειτουργικότητας της εφαρμογής. Πρέπει να υπάρχει μια λίστα με πιθανές λειτουργίες για την πρώτη έκδοση της εφαρμογής. Ωστόσο, πρέπει να σημειωθεί ότι ανάλογα με το πλήθος των features που υπάρχουν σε μια εφαρμογή, αυξάνεται ή μειώνεται ο χρόνος υλοποίησής τους. Περισσότερες λειτουργίες από ότι χρειάζονται οι χρήστες ενδέχεται να κάνει την εφαρμογή “βαριά” και μη-φιλική προς το χρήστη, κάτι που επηρεάζει άμεσα την αποδοχή από αυτούς.

Η πλειονότητα των διαδικτυακών εφαρμογών περιλαμβάνουν βασικές λειτουργίες όπως:

- Δημιουργία λογαριασμού
- Ανάκτηση κωδικού
- Αλλαγή κωδικού και email
- Σύστημα επικοινωνίας
- Δημιουργία επαφής
- Συγγραφή μηνύματος ή σχολίου
- Αξιολόγηση μηνύματος, σελίδας ή σχολίου
- Διαμοιρασμός σελίδας ή σχολίου
- Δυνατότητα για upload αρχεία
- Δυνατότητα για upload φωτογραφία προφίλ χρήστη
- Διαγραφή λογαριασμού

Στάδιο 2: Σχεδιασμός

4. Σχεδιασμός του Workflow. Σε αυτό το βήμα θα γίνει το πλάνο της εφαρμογής. Πώς πρέπει η εφαρμογή να καθοδηγεί το χρήστη από την εγγραφή του στο σύστημα, την συμπλήρωση του προφίλ του ή πληροφοριών που απαιτούνται μέχρι και το τελικό σημείο, για παράδειγμα την δημοσίευση ενός άρθρου ή σχολίου σε μια εφαρμογή αξιολόγησης τουριστικών αξιοθέατων. Το βήμα του σχεδιασμού του Workflow ουσιαστικά είναι η ίδια εφαρμογή μέσα από τα μάτια του χρήστη. Τα αποτελέσματα

από την έρευνα αγοράς παίζουν μεγάλο ρόλο στο σχεδιασμό του Workflow, δηλαδή τι κάνουν ίδιο ή διαφορετικό οι ανταγωνιστές και πώς αυτοί διαχειρίζονται τους χρήστες τους.

Στάδιο 3: Development

5. Αρχιτεκτονική και σχεδιασμός της Βάσης Δεδομένων. Η βάση δεδομένων είναι ο ψηφιακός τόπος που περιέχει όλα τα δυναμικά ψηφιακά δεδομένα μιας ιστοσελίδας ή διαδικτυακής εφαρμογής οργανωμένα σε κατηγορίες.^[24] Η βάση δεδομένων θα πρέπει να είναι αρχιτεκτονικά σωστή καθώς αποτελεί τα θεμέλια της εφαρμογής. Όπως μια κατοικία δεν μπορεί να αλλάξει θεμέλια έτσι και η βάση δεδομένων εάν χρησιμοποιηθεί είναι πολύ δύσκολο να αλλαχτεί αν κάτι δεν έγινε σωστά από την αρχή. Όλες οι πληροφορίες της εφαρμογής όπως λογαριασμοί, ονόματα, κάρτες, διευθύνσεις, ρυθμίσεις κ.α. αποθηκεύονται στη βάση δεδομένων. Για αυτό το λόγο πρέπει να είναι αρχιτεκτονικά ορθή ώστε στο μέλλον να μπορεί να επεκταθεί ή να τροποποιηθεί με ευκολία. Ο προγραμματιστής και δημιουργός της βάσης δεδομένων πρέπει να κατασκευάσει την βάση δεδομένων με γνώμονα παγκόσμια πρότυπα και τις καλύτερες διαθέσιμες πρακτικές. Τέλος, η βάση δεδομένων πρέπει να είναι κάθε στιγμή ασφαλής από ενδεχόμενες απειλές ή παράνομες προσβάσεις σε αυτή καθώς δεν κινδυνεύει μόνο η εφαρμογή αλλά και τα προσωπικά δεδομένα των χρηστών.
6. Κατασκευή Front End και Back End. Η κατασκευή των διαδικτυακών εφαρμογών χωρίζεται σε 2 μέρη. Το Front End μιλάει για την γενικότερη εμφάνιση της εφαρμογής και πώς φτάνει στα μάτια του χρήστη. Διάφορες γλώσσες προγραμματισμού, frameworks και εργαλεία βοηθούν τους προγραμματιστές να υλοποιήσουν το Front End της εφαρμογής. Αντιθέτως, το Back End αφορά την εσωτερική διαχείριση των δεδομένων. Μέσω γλωσσών προγραμματισμού επιτυγχάνεται η επικοινωνία της εφαρμογής με τη βάση δεδομένων. Είναι ουσιαστικά το νευρικό σύστημα της εφαρμογής και βασίζεται πάνω στο Front End.

Στάδιο 4: Εκκίνηση

7. Hosting. Όταν δημιουργηθεί η εφαρμογή πρέπει να ανέβει στο διαδίκτυο ώστε οι χρήστες να έχουν πρόσβαση. Αυτό γίνεται με 3 απλά βήματα:
 - Αγορά domain name
 - Αγορά χώρου

- Ανέβασμα των αρχείων της εφαρμογής στο χώρο
8. Εκκίνηση. Μπορεί να τελειώσει η κατασκευή της εφαρμογής αλλά δεν σημαίνει ότι τελειοποιήθηκε. Όλες οι εφαρμογές συνεχίζονται να αναπτύσσονται και να υλοποιούν νέα features που θα την κάνουν καλύτερη και φιλικότερη προς το χρήστη. Τέλος, απαραίτητη για μια εφαρμογή είναι η διαφήμιση και η εμφάνισή της στις μηχανές αναζήτησης.

2.4 Κατασκευή διαδικτυακής εφαρμογής WEVIAN

Σε αυτήν την ενότητα της πτυχιακής εργασίας επιχειρείται η ανάλυση των γλωσσών προγραμματισμού, των βιβλιοθηκών και άλλων βασικών τεχνικών χαρακτηριστικών της διαδικτυακής εφαρμογής WEVIAN.^[25]

Η διαδικτυακή εφαρμογή WEVIAN αφορά την ψηφιακή προσομοίωση και ανάλυση των υπο-σταδίων Floorplan και Placement του σταδίου της Φυσικής σχεδίασης στην ροή σχεδίασης Ολοκληρωμένων Κυκλωμάτων όπως αναπτύχθηκαν στο Κεφάλαιο 1, ενότητα 1.2. Το εργαλείο WEVIAN ανήκει στην κατηγορία των Επιστημονικών διαδικτυακών εφαρμογών όπως αναπτύχθηκε στο Κεφάλαιο 2, ενότητα 2.2. Σε επόμενο κεφάλαιο θα γίνει λεπτομερής παρουσίαση της εμφάνισης, λειτουργικότητας και πειραματικών αποτελεσμάτων της εφαρμογής.

2.4.1 Γλώσσες προγραμματισμού

Το εργαλείο WEVIAN δημιουργήθηκε χρησιμοποιώντας ένα σετ γλωσσών προγραμματισμού. Παρακάτω περιγράφονται οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για το Front End.

HTML

Η HTML (HyperText Markup Language) κάνει την πρώτη εμφάνισή της το 1990 από τον δημιουργό του World Wide Web, Tim-Berners Lee. Στην αρχική της υπόσταση η γλώσσα υποστήριζε 18 tags. Από τα 18 tags έχουν αλλαχτεί ή αφαιρεθεί τα 12 με μόνο 6 να έχουν τις ίδιες ιδιότητες από το 1990.^[26]

Όλες οι σελίδες που προβάλλονται από έναν browser σαν τελικό αποτέλεσμα παράγουν HTML. Η HTML δεν είναι γλώσσα προγραμματισμού αλλά γλώσσα markup, περιγράφει δηλαδή την δομή των σελίδων. Κάθε σελίδα αποτελείται από blocks που

ονομάζονται elements και αναπαριστώνται με tags. Κάθε tag επισημαίνει ένα κομμάτι κειμένου με μία ταμπέλα όπως “title”, “table”, “input” και περιλαμβάνει attributes τα οποία αλλάζουν τη συμπεριφορά του element ανάλογα.

Το μεγαλύτερο μέρος των tag εσωκλείονται στα μαθηματικά σύμβολα “μικρότερο” και “μεγαλύτερο”. Για παράδειγμα το tag “ <title> “ δηλώνει την αρχή επισήμανσης του τίτλου της σελίδας, ενώ το “ </title> “ δηλώνει το τέλος επισήμανσης του τίτλου της σελίδας. Όμως, υπάρχουν tags τα οποία δεν χρειάζεται να ανοίξουν, δεν χρειάζεται να κλείσουν ή δεν χρειάζεται να περιέχουν κάτι.^[27] Όλα τα tags μπορούν να περιλαμβάνουν αόριστο αριθμό από στοιχεία που ονομάζονται attributes. Κάθε attribute περιγράφεται από μια τιμή η οποία μπορεί να χρησιμοποιηθεί από άλλες γλώσσες προγραμματισμού ή να προσθέσει ιδιότητες στο element. Τα tags επίσης μπορούν να χρησιμοποιηθούν εμφωλευμένα δηλαδή το ένα μέσα στο άλλο. Όταν όμως γίνεται αυτό θα πρέπει να κλείνουν με την ίδια σειρά που άνοιξαν όπως φαίνεται στην Εικόνα 3.



Εικόνα 3: Βασική δομή μιας σελίδας HTML.^[28]

Ξεκινώντας από τα βασικότερα και συχνότερα elements, το “html” element υπάρχει σε κάθε HTML αρχείο και σηματοδοτεί την αρχή και το τέλος του. Στην αρχή του HTML αρχείου μετά το “html” element υπάρχει προαιρετικά το “head” element που περιλαμβάνει διάφορα άλλα elements που στο σύνολό τους αποτελούν το header της σελίδας. Τα πιθανά elements για το header είναι, το “title” για τον τίτλο της σελίδας, το “link” και το “script” για την εισαγωγή εξωτερικών αρχείων style sheet και client-side αρχείων JavaScript αντίστοιχα, το “style” για τον ορισμό του style της σελίδας, το “base” για τον ορισμό του βασικού URL και base target των σχετικών διευθύνσεων και το “meta” για τον ορισμό metadata.[29] Τα metadata ή μεταδεδομένα χρησιμοποιούνται από τους browser και τις μηχανές αναζήτησης για αλλαγή κωδικοποίησης χαρακτήρων, ορισμό περιγραφής σελίδας, λέξεων κλειδιών και συντακτών ή υπηρεσίες όπως αυτόματη ανανέωση κειμένου.^[30] Έπειτα μετά το header ξεκινάει ο κορμός ενός HTML αρχείου με το element “body”. Μέσα στο “body” μπορούν να υπάρχουν όλα τα υπόλοιπα tags.

Στον Πίνακα 3 σημειώνονται όλα τα tags τα οποία χρησιμοποιήθηκαν, καθώς και μια σύντομη περιγραφή.

TAGS	Περιγραφή
html, head, body	Δημιουργία βασικής δομής σελίδας.
meta, title, link, script	Δημιουργία header (μεταδεδομένα, τίτλος, εισαγωγή εξωτερικών αρχείων style sheet, εισαγωγή client-side αρχείων JavaScript).
div	Ορισμός ενός τμήματος στο αρχείο HTML.
a, input, button, img	Εισαγωγή συνδέσμου, πεδίου, κουμπιού, εικόνας.
hr, br	Εισαγωγή οριζόντιας γραμμής, νέας γραμμής.
b, center, u, i	Μορφοποίηση κειμένου (bold, center, underline, italics).
table, tr, th, td	Εισαγωγή πίνακα, σειράς πίνακα, επικεφαλίδας πίνακα, δεδομένων πίνακα.
select, option	Εισαγωγή μενού επιλογών, επιλογών.
ul, li	Εισαγωγή μη ταξινομημένης λίστας, στοιχείου λίστας.
Span	Inline ομαδοποίηση elements.
Form	Εισαγωγή φόρμας.

Πίνακας 3: Τα HTML tags που χρησιμοποιήθηκαν στο εργαλείο WEVIAN.


CSS

Το CSS (Cascading Style Sheets) προτάθηκε από τον Håkon Wium Lie το 1994 όταν ο ίδιος εργαζόταν στο CERN μαζί με τον δημιουργό του World Wide Web και της HTML, Tim-Berners Lee. Η πρώτη έκδοση του CSS εκδόθηκε 2 χρόνια αργότερα, από το World Wide Web Consortium, το 1996 έπειτα από τροποποιήσεις που έγιναν καθώς δεν ήταν η μόνη style sheet γλώσσα που προτάθηκε.^[31]

Το CSS δεν είναι γλώσσα προγραμματισμού αλλά γλώσσα style sheet, περιγράφει δηλαδή την παρουσίαση των HTML κειμένων σε διάφορες συσκευές. Όπως προαναφέρθηκε, κάθε HTML κείμενο αποτελείται από elements. Κάθε element μπορεί να αναπαρασταθεί διαφορετικά σε διαφορετικά μέσα ανάλογα με τις ιδιότητες που λαμβάνει από το CSS.

Τα αρχεία CSS περιέχουν έναν αόριστο αριθμό από rules, δηλαδή κανόνες που προσδίδουν ιδιότητες στα elements. Κάθε κανόνας αποτελείται από 2 μέρη, το πρώτο ονομάζεται selector και αντιστοιχεί στο όνομα του element στο οποίο αναφέρεται και το δεύτερο ονομάζεται declaration block ή block και σηματοδοτείται από τους χαρακτήρες “ { ” που δηλώνει την έναρξη του block και “ } ” για να δηλώσει το πέρας. Κάθε block αποτελείται από έναν αόριστο αριθμό από properties, όπου κάθε property έχει μία τιμή και διαχωρίζεται από το επόμενο property με semicolon. Η γενική δομή των CSS κανόνων φαίνεται στην Εικόνα 4.

declaration block



format: selector { property: value; }

example: body { color: black; }

Εικόνα 4: Γενική δομή των CSS κανόνων.

Υπάρχουν 3 διαφορετικοί τρόποι εισαγωγής CSS σε ένα αρχείο HTML (βλ. Εικόνα 5). Ο πρώτος είναι μέσω εξωτερικών αρχείων CSS και εισαγωγής τους από το tag “link” της HTML, ο δεύτερος είναι εσωτερική εισαγωγή των κανόνων CSS στο αρχείο HTML

μέσω του tag “style” και ο τρίτος είναι η inline εισαγωγή κανόνων CSS σε κάθε element ξεχωριστά ως τιμή του attribute “style”.

✧ **Εξωτερική εισαγωγή**

```
<link href="filename.css">
```

✧ **Εσωτερική εισαγωγή**

```
<style>
  body { color: black; }
</style>
```

✧ **Inline εισαγωγή**

```
<body style="color: black;">
  ...
</body>
```

Εικόνα 5: Τρόποι εισαγωγής CSS στην HTML.

Τα κύρια πλεονεκτήματα του CSS είναι:

- Η πιο σημαντική αφορμή για να χρησιμοποιήσει κανείς CSS είναι η δραματική αλλαγή της αισθητικής σε μία εφαρμογή. Με τη χρήση CSS, μια εφαρμογή μετατρέπεται από ένα απλό κείμενο σε φουτουριστικό, χρωματιστό, ευχάριστο και διαδραστικό.
- Τα τελευταία χρόνια οι εφαρμογές τείνουν να είναι όσο το δυνατό περισσότερο μινιμαλιστικές. Προσπαθούν να εμφανίζουν στο χρήστη αυτό που θέλει να δει εκείνη τη στιγμή χωρίς επιπλέον πληροφορίες. Αυτός ο τομέας ονομάζεται User Experience και ουσιαστικά αφορά τους τρόπους με τους οποίους μια εφαρμογή γίνεται περισσότερο φιλική προς το χρήστη. Animations, tooltips, popups, γραφικά και τεχνικές responsiveness είναι μερικά features του CSS που βελτιώνουν κατακόρυφα το User Experience.
- Επιπρόσθετα, μέσω του CSS είναι πολύ εύκολο να επιτευχθεί και να προσαρμοστεί η προσβασιμότητα. Για παράδειγμα, ένας χρήστης screen reader θα έχει μια διαφορετική εμπειρία από έναν χρήστη tablet. Το CSS είναι ικανό να διαχειριστεί διάφορες συσκευές και να εμφανίζει σε κάθε μια διαφορετικά αποτελέσματα ώστε να βελτιωθεί η εμπειρία του χρήστη.

Κατανοώντας τα παραπάνω, χρησιμοποιήθηκε CSS σε κάθε σελίδα του εργαλείου WEVIAN. Στο παρακάτω πίνακα αναγράφονται τα βασικότερα properties που χρησιμοποιήθηκαν.

Properties	Περιγραφή
background, background-color	Εισαγωγή background και επιλογή χρώματος background του element.
border, border-radius	Εισαγωγή γραμμής ορίου και ορισμός ακτίνας γωνιών του element.
margin, padding	Εισαγωγή χώρου γύρο από το element εκτός και εντός της γραμμής του ορίου αντίστοιχα.
position	Ορισμός της μεθόδου τοποθέτησης που χρησιμοποιείται από το element.
top, left	Ορισμός της κατακόρυφης και οριζόντιας θέσης αντίστοιχα του element.
transform	Μετασχηματισμός 2D ή 3D του element.
width, height	Ορισμός πλάτους και ύψους του element.

Πίνακας 4: Τα CSS properties που χρησιμοποιήθηκαν στο εργαλείο WEVIAN.

JavaScript

Το Front End κομμάτι του εργαλείου WEVIAN ολοκληρώνεται με την JavaScript. Η JavaScript αποτελεί τον εγκέφαλο της διαδικτυακής εφαρμογής WEVIAN. Είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε το 1995 από τον Brendan Eich όταν εργαζόταν για την Netscape Communications.^[32] Μπορεί να διαχειρίζεται την HTML και το CSS μιας σελίδας και μπορεί να την τροποποιήσει χωρίς να γίνει αλλαγή σελίδας ή refresh, μπορεί επίσης να επικοινωνήσει με το server για λήψη ή μετάδοση δεδομένων μέσω βιβλιοθηκών.

Υπάρχει μια γενική σύγχυση σχετικά με την JavaScript και την Java. Πρέπει να ξεκαθαριστεί ότι είναι 2 διαφορετικές γλώσσες. Από τη μία ο κώδικας της Java πρέπει να μεταγλωττιστεί πρώτα μέσω κάποιου compiler για να μπορεί να εκτελεστεί ενώ από την άλλη η JavaScript δεν χρειάζεται compiler και το συντακτικό της είναι πιο ευέλικτο.

Η JavaScript είναι γλώσσα προγραμματισμού, που σημαίνει ότι υποστηρίζει μεταβλητές, μεθόδους, κλάσεις, functions, δομές δεδομένων και objects. Μαζί με την HTML και το CSS αποτελούν το τρίπτυχο του Front End των διαδικτυακών εφαρμογών και χρησιμοποιούνται από το μεγαλύτερο μέρος των online ιστοσελίδων μέχρι και σήμερα.

Τρεις γενικές ιδέες αφορούν την JavaScript. Πρώτον, βασίζεται στα objects, δηλαδή δημιουργία αντικειμένων με ειδικά χαρακτηριστικά που όμως δεν βασίζονται αναγκαστικά σε κλάσεις και άρα δεν υπάρχει διαχωρισμός μεταξύ μιας κλάσης και ενός instance. Δεύτερον, είναι Client side γλώσσα, δηλαδή δεν απαιτείται κάποιος server ή compiler για την εκτέλεση του κώδικα και ο κώδικας είναι άμεσα διαθέσιμος στον χρήστη και μπορεί να τον τροποποιήσει, κάτι που μπορεί να αποδειχτεί επικίνδυνο σε εφαρμογές οι οποίες δεν έχουν πάρει τα απαραίτητα μέτρα προστασίας. Τρίτον, αποτελεί γλώσσα scripting από το γεγονός ότι δεν χρειάζεται compiler. Αυτό σημαίνει ότι ο browser εκτελεί τον κώδικα εκείνη τη στιγμή και τα errors που μπορεί να προκύψουν είναι εξαρτώμενα από τον εκάστοτε browser.^[33]

Υπάρχουν 3 τρόποι εισαγωγής JavaScript όπως φαίνονται στην Εικόνα 6, ο πρώτος είναι μέσω εξωτερικών αρχείων JavaScript και εισαγωγής από το tag “script” της HTML, ο δεύτερος είναι εσωτερικής εισαγωγής του κώδικα JavaScript στο ίδιο αρχείο HTML μέσω του tag “script” και ο τρίτος είναι η inline εισαγωγή κώδικα JavaScript σε κάθε element ξεχωριστά ως τιμή ορισμένων attribute όπως του “onclick”.

✧ Εξωτερική εισαγωγή

```
<script src="filename.js"></script>
```

✧ Εσωτερική εισαγωγή

```
<script>
  alert('Hello!');
</script>
```

✧ Inline εισαγωγή

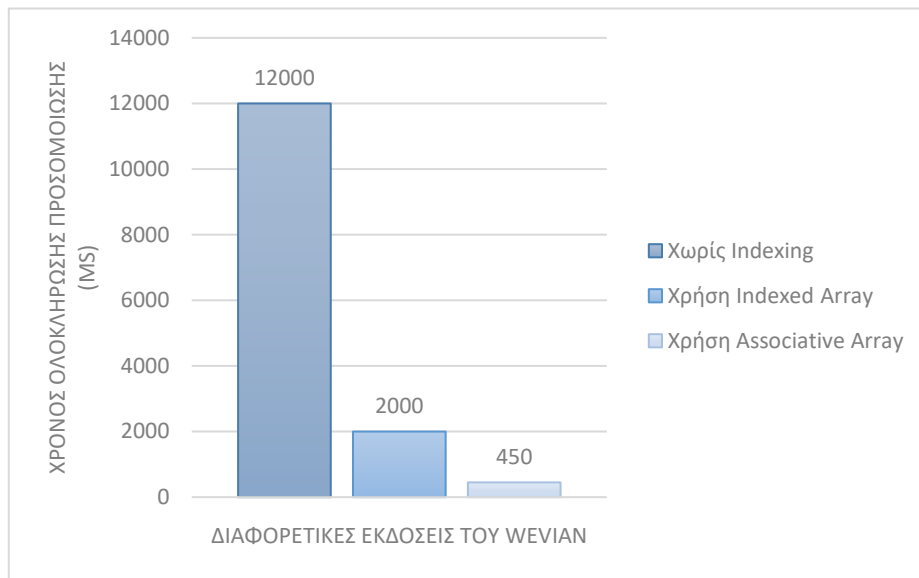
```
<body onclick="alert('Hello!');">
  ...
</body>
```

Εικόνα 6: Τρόποι εισαγωγής JavaScript στην HTML.

Στη διαδικτυακή εφαρμογή WEVIAN η JavaScript χρησιμοποιήθηκε ως επί το πλείστον για την διαχείριση του chip και των cell στις προσομοιώσεις του placement και

του floorplan. Ακόμη, χρησιμοποιήθηκε για την ικανότητα ασύγχρονης επικοινωνίας με τον server ώστε να αντλήσει τις πληροφορίες του chip για το εκάστοτε κύκλωμα.

Ωστόσο, η μεγαλύτερη πρόκληση στην δημιουργία του WEVIAN ήταν οι χρόνοι ολοκλήρωσης των προσομοιώσεων. Σταδιακά οι χρόνοι μειώθηκαν, έπειτα από ορθή χρήση της λογικής και των καλύτερων πρακτικών της JavaScript. Στην Εικόνα 7 φαίνονται οι σταδιακές μειώσεις στο χρόνο ολοκλήρωσης μιας προσομοίωσης ενός chip με περίπου 12000 cells σε σχέση με τις εκδόσεις της εφαρμογής.



Διάγραμμα 1: Μέσος χρόνος rendering προσομοίωσης από δείγμα 10 προσομοιώσεων ενός chip με 12752 cells συναρτήσει διαφορετικών μεθόδων indexing που χρησιμοποιήθηκαν από διαφορετικές εκδόσεις του εργαλείου WEVIAN.

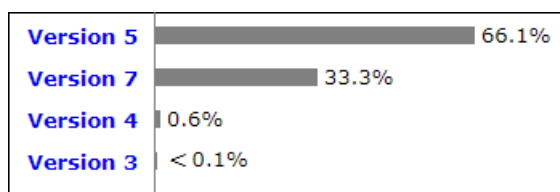
Από το παραπάνω διάγραμμα είναι εύκολο να αντιληφθεί κανείς ότι η σωστή χρήση μιας γλώσσας προγραμματισμού δύναται να αυξήσει την επίδοση και την ταχύτητα των διαδικτυακών εφαρμογών. Για το λόγο αυτό οι διαδικτυακές εφαρμογές πρέπει να περνάνε τακτικούς ελέγχους για την πιστοποίηση χρήσης καλών πρακτικών πριν και αφού δημοσιευτούν στο ευρύ κοινό.

Εν κατακλείδι, το Front End τμήμα της διαδικτυακής εφαρμογής WEVIAN αποτελεί το 90.3% του συνολικού project. Από αυτό, το 87.4% είναι JavaScript ενώ το 2.9% είναι CSS. Το υπόλοιπο 9.7% αποτελείται από την PHP και ένα αμελητέο ποσοστό της SQL. Στην συνέχεια αυτής της υπο-ενότητας θα αναπτυχθεί το Back End τμήμα.

PHP

Η PHP (Hypertext Preprocessor) όπως και η JavaScript είναι μια γλώσσα προγραμματισμού για διαδικτυακές εφαρμογές. Σχεδιάστηκε το 1995 από τον Rasmus Lerdorf για χρήση στην προσωπική του ιστοσελίδα χωρίς να έχει στόχο την δημιουργία μιας γλώσσας προγραμματισμού. Τρία χρόνια μετά οι Zeev Suraski και Andi Gutmans έχοντας επαναπροσδιορίσει τη γλώσσα εκδίδουν την PHP 3.^[34]

Η PHP αυτή τη στιγμή βρίσκεται στην 7^η έκδοσή της και χρησιμοποιείται ως βασική Back End γλώσσα προγραμματισμού από το μεγαλύτερο μέρος των online ιστοσελίδων. Πιο συγκεκριμένα, το 79% των συνολικών ιστοσελίδων για τις οποίες γνωρίζουμε την server-side γλώσσα προγραμματισμού χρησιμοποιεί PHP. Από αυτές μόνο το 33.3% χρησιμοποιεί την τελευταία έκδοση της PHP όπως φαίνεται στην Εικόνα 7.^[34]



Εικόνα 7: Κατανομή ιστοσελίδων μεταξύ των διαφορετικών εκδόσεων PHP.^[35]

Η διαδικτυακή εφαρμογή WEVIAN χρησιμοποιεί την PHP για 2 λόγους. Για τη δυνατότητα επικοινωνίας με τη βάση δεδομένων κατά την εισαγωγή ή εξαγωγή πληροφοριών και για το parsing των δεδομένων από τα αρχεία του εκάστοτε κυκλώματος.

Πέρα από την προσομοίωση κυκλωμάτων περιλαμβάνει και λειτουργίες ανάλυσης και εξαγωγής στατιστικών ενός κυκλώματος, όπως για παράδειγμα υπολογισμό του συνολικού μήκους του καλωδίου. Για να υλοποιηθούν οι επιπλέον λειτουργικότητες πρέπει να υπάρχουν και τα αντίστοιχα δεδομένα στα οποία βασίζονται. Τα δεδομένα για ένα κύκλωμα ανεβαίνουν στο server οργανωμένα σε μια ιδιαίτερη μορφή που ονομάζεται Bookshelf format.

Το Bookshelf format είναι ένας τρόπος οργάνωσης δεδομένων που χρησιμοποιείται στα ερευνητικά VLSI projects. Σύμφωνα με αυτό τα δεδομένα του κυκλώματος χωρίζονται σε 6 αρχεία κειμένου για το Placement και σε 3 αρχεία κειμένου για το Floorplan. Τα αρχεία αυτά είναι:

Placement:

1. Το αρχείο.aux σημειώνει τα αρχεία εισόδου σε Bookshelf format:

<name> : αρχείο.nodes αρχείο.nets αρχείο.wts αρχείο.pl αρχείο.scl

Όπου name το όνομα του format και δίπλα τα αρχεία του κυκλώματος.

2. Το αρχείο.nodes ορίζει το πλάτος, ύψος των node καθώς και αν είναι pad σε Bookshelf format. Αποτελείται από το header και το body.

Το header περιλαμβάνει:

NumNodes : <number>

NumTerminals : <number>

Όπου NumNodes είναι το πλήθος των κινητών cell και NumTerminals το πλήθος των pad.

Το body περιλαμβάνει για κάθε node:

<name> <width> <height> terminal

Όπου name το όνομα του node, width το πλάτος του node, height το ύψος του node και προαιρετικά τη λέξη terminal αν είναι pad.

3. Το αρχείο.nets ορίζει το σύνολο των nets σε Bookshelf format. Αποτελείται από το header και το body.

Το header περιλαμβάνει:

NumNets : <number>

NumPins : <number>

Όπου NumNets είναι το πλήθος των net group και NumPins το πλήθος των συνολικών συνδέσεων από κάθε net group.

Το body περιλαμβάνει για κάθε net group:

NetDegree : <links> <net_name>

<name> <io> : <x_offset> <y_offset>

<name> <io> : <x_offset> <y_offset>

Όπου links το σύνολο των συνδέσεων σε αυτό το net group, net_name προαιρετικά το όνομα του net group, name το όνομα του node, io για να ορίσει αν είναι είσοδος ή έξοδος με I να σημαίνει είσοδος και O έξοδος και x_offset και y_offset τα offset ως προς x και y άξονες.

4. Το αρχείο.wts ορίζει τα βάρη των node σε Bookshelf format:

<name> <number>

Όπου name το όνομα του node και number το βάρος.

5. Το αρχείο.pl ορίζει την τοποθεσία και προσανατολισμό των node καθώς και αν είναι pad σε Bookshelf format:

<name> <x_coordinate> <y_coordinate> : <orientation> /FIXED

Όπου name το όνομα του node, x_coordinate και y_coordinate οι συντεταγμένες του node πάνω στο chip, orientation ο προσανατολισμός του node και η λέξη /FIXED αν είναι pad.

6. Το αρχείο.scl ορίζει τις οριζόντιες γραμμές του κυκλώματος σε Bookshelf format. Αποτελείται από το header και το body.

Το header περιλαμβάνει:

NumRows : <number>

Όπου NumRows το πλήθος των σειρών.

Το body περιλαμβάνει για κάθε σειρά:

CoreRow Horizontal

Coordinate : <number>

Height : <number>

Sitewidth : <number>

Sitespacing : <number>

Siteorient : <orientation>

Sitesymmetry : <character>

SubrowOrigin : <number> Numsites : <number>

Όπου Coordinate η συντεταγμένη της γραμμής, Height το ύψος της γραμμής, Sidewidth συμπληρώνεται προαιρετικά αλλιώς ισοδυναμεί με το Sitespacing, Sitespacing η απόσταση μεταξύ των αρχών των γειτονικών site, Siteorient ο προσανατολισμός των node μέσα σε αυτή την γραμμή, Sitesymmetry συμπληρώνεται προαιρετικά αλλιώς αφήνεται κενό και το SubrowOrigin ορίζει ένα subrow και θέτει τιμή στο subrow origin ενώ το Numsites θέτει τη συντεταγμένη του subrow origin.

Floorplan:

1. Το αρχείο.blocks ορίζει το όνομα και άλλες προαιρετικές πληροφορίες για κάθε block και pad σε Bookshelf format. Επίσης, υπάρχουν 2 τύποι block, ο hard και ο soft. Αποτελείται από το header και το body.

Το header περιλαμβάνει:

NumSoftRectangularBlocks : <number>

NumHardRectilinearBlocks : <number>

NumTerminals : <number>

Όπου NumSoftRectangularBlocks είναι το πλήθος των soft block, NumHardRectilinearBlocks είναι το πλήθος των hard block και NumTerminals είναι το πλήθος των pad.

Το body περιλαμβάνει για κάθε soft block:

<name> softrectangular <area> <minAspectRatio> <maxAspectRatio>

Όπου name το όνομα του block, softrectangular για να ορίσει ότι είναι block τύπου soft, area το εμβαδό του block, minAspectRatio το ελάχιστο επιτρεπτό aspect ratio και maxAspectRatio το μέγιστο επιτρεπτό aspect ratio.

Το body περιλαμβάνει για κάθε hard block:

<name> hardrectilinear <totalVertexes> <vertex1> <vertex2>, ..., <vertexN>

Όπου name το όνομα του block, hardrectilinear για να ορίσει ότι είναι block τύπου hard, totalVertexes το πλήθος των vertices και vertex1-N η λίστα των vertices με φορά ρολογιού. Κάθε vertex αποτελείται από παρενθέσεις που περικλείουν τις συντεταγμένες του vertex χωρισμένες με κόμμα.

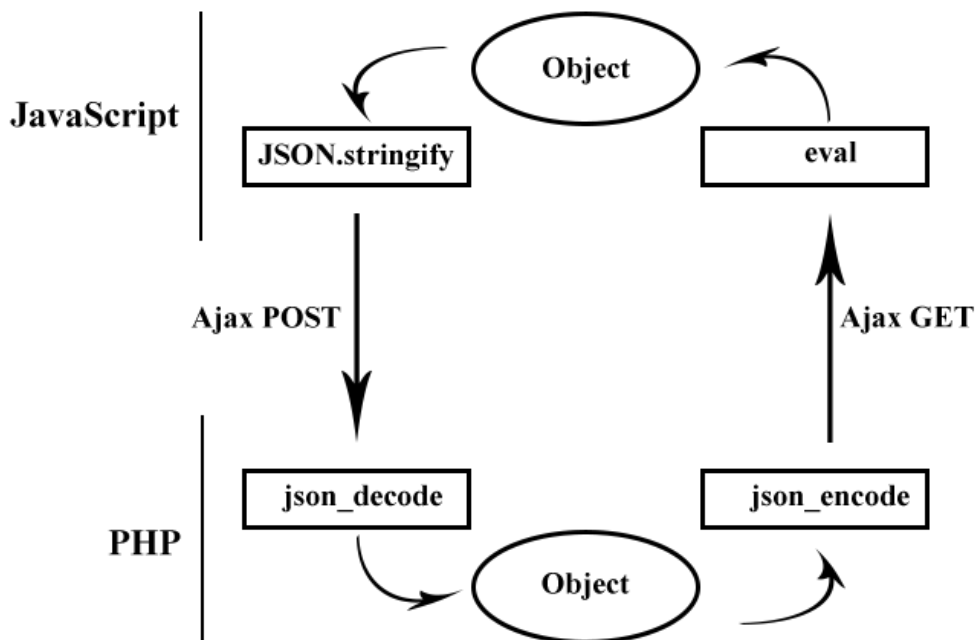
Το body περιλαμβάνει για κάθε pad:

<name> terminal

Όπου name το όνομα του pad και δίπλα η λέξη terminal.

2. Το αρχείο.nets ορίζει το σύνολο των nets σε Bookshelf format. Αντίστοιχο με το αρχείο.nets του Placement.
3. Το αρχείο.pl ορίζει την τοποθεσία και προσανατολισμό των node καθώς και αν είναι pad σε Bookshelf format. Αντίστοιχο με το αρχείο.pl του Placement.

Έτσι, αφού ολοκληρωθεί η ανάλυση, όλα τα δεδομένα από τα παραπάνω αρχεία βρίσκονται σε μορφή JSON και αποστέλλονται στην JavaScript. Η JSON (JavaScript Object Notation) είναι μια πολύ ελαφριά γλώσσα ανταλλαγής δεδομένων, ιδανική για μεταφορά και μετατροπή δεδομένων και χρησιμοποιήθηκε διότι είναι συμβατή με την JavaScript και την PHP.



Εικόνα 8: Μεταφορά δεδομένων μεταξύ PHP και JavaScript χρησιμοποιώντας JSON.^[36]

SOL

Η SQL (Structured Query Language) σχεδιάστηκε το 1974 από τους Donald D. Chamberlin και Raymond F. Boyce αλλά έκτοτε έχουν υπάρξει πολλές διαφορετικές εκδόσεις και αλλαγές που δεν θυμίζει τίποτα στην αρχική μορφή της.^[37] Η SQL είναι μια γλώσσα ερωταπαντήσεων που χρησιμοποιείται για την διαχείριση των βάσεων δεδομένων.

Οι κύριες λειτουργίες της είναι η προσπέλαση, δημιουργία, διαγραφή και τροποποίηση των βάσεων δεδομένων. Αυτό μπορεί να γίνει μέσω των 6 βασικών εντολών οι οποίες είναι αρκετές για να διαχειριστούν σχεδόν ολοκληρωτικά την βάση δεδομένων.

Εντολή	Παράδειγμα	Περιγραφή
SELECT	SELECT column FROM table_name;	Επιλογή δεδομένων.
INSERT	INSERT INTO table_name (column) VALUES (value);	Εισαγωγή δεδομένων.
UPDATE	UPDATE table_name SET column=value;	Τροποποίηση δεδομένων.
DELETE	DELETE FROM table_name;	Διαγραφή δεδομένων.
CREATE	CREATE DATABASE database_name;	Δημιουργία βάσης.
DROP	DROP DATABASE database_name;	Διαγραφή βάσης.

Πίνακας 5: Βασικές εντολές SQL.

Η SQL χρησιμοποιήθηκε για την αποθήκευση των στοιχείων των χρηστών καθώς και την αποθήκευση στοιχείων των project τους για μελλοντική χρήση.

2.4.2 Βιβλιοθήκες

Εκτός από τις γλώσσες προγραμματισμού στο εργαλείο WEVIAN χρησιμοποιήθηκαν διάφορες βιβλιοθήκες. Οι βιβλιοθήκες στις γλώσσες προγραμματισμού χρησιμοποιούνται για να αυτοματοποιήσουν ορισμένες λειτουργίες. Το βασικό τους χαρακτηριστικό είναι ότι είναι επαναχρησιμοποιήσιμες. Παρακάτω περιγράφονται όλες οι βιβλιοθήκες που χρησιμοποιήθηκαν.

jQuery

Η jQuery είναι μια βιβλιοθήκη γραμμένη σε JavaScript. Χρησιμοποιείται από την πλειοψηφία των ιστοσελίδων που χρησιμοποιούν JavaScript.^[38] Είναι γνωστή για το πλήθος των λειτουργιών που υποστηρίζει και για το μικρό της μέγεθος. Οι πιο γνωστές λειτουργίες της είναι η διαχείριση των HTML κειμένων, η δημιουργία και διαχείριση animation και συμβάντων καθώς και η Ajax. Η Ajax είναι ένα σύνολο τεχνικών που υλοποιούν την ασύγχρονη επικοινωνία μεταξύ του client και server στην JavaScript. Η jQuery είναι ο βασικός πυλώνας των περισσότερων διαδικτυακών εφαρμογών ή ιστοσελίδων που χρησιμοποιούν JavaScript.

jQuery UI

Ουσιαστικά πρόκειται για επέκταση της jQuery με περισσότερα χαρακτηριστικά. Το “UI” αναφέρεται στις αγγλικές λέξεις user interface, δηλαδή το πως ο χρήστης βλέπει μια εφαρμογή. Μερικές από τις λειτουργίες της είναι η δυνατότητα drag, drop, resize, select, sort των HTML element. Η jQuery UI χρησιμοποιήθηκε στο εργαλείο WEVIAN για την επεξεργασία του Front End και του user interface καθώς και στο Floorplan για την δυνατότητα drag των στοιχείων.

Chart.js

Το Chart.js είναι μια βιβλιοθήκη για την δημιουργία Chart, diagrams και animations. Χρησιμοποιήθηκε στο Placement για την δημιουργία στατιστικού report που μεταξύ άλλων περιλαμβάνει διαγράμματα για τους χρόνους εκτέλεσης και rendering των εικονικών chip.

html2canvas

Το html2canvas αποτελεί βιβλιοθήκη της JavaScript για την μετατροπή του HTML κειμένου ή μέρος αυτού σε canvas. Λόγω της μεγάλης πυκνότητας σε στοιχεία που εμφανίζονται στην οθόνη του χρήστη είναι τεχνικά δύσκολο να μπορέσει κανείς να τα αποθηκεύσει όλα μαζί ως εικόνα ή να τα εκτυπώσει. Για αυτό το λόγο το html2canvas αυτοματοποιεί την διαδικασία και παράγει εικόνες ή αρχεία από διάφορα μέρη του HTML κειμένου. Το html2canvas χρησιμοποιήθηκε για την εξαγωγή του chip ως εικόνα μορφής PNG.

dropzone.js

Η βιβλιοθήκη dropzone.js υπάρχει για να δημιουργεί και να διαχειρίζεται ζώνες που υποστηρίζουν drag'n'drop file uploads. Μια μικρή και εύχρηστη βιβλιοθήκη που χρησιμοποιήθηκε για να διαχειριστεί το ανέβασμα του αρχείου PL του χρήστη στο Placement.

Overlap.js

Η τελευταία βιβλιοθήκη είναι ικανή να συγκρίνει 2 HTML element για να δει αν επικαλύπτονται. Χρησιμοποιήθηκε στο Floorplan για τον έλεγχο επικάλυψης των στοιχείων ενώ ο χρήστης τα μετακινεί.

2.4.3 Τεχνικά χαρακτηριστικά

Η εφαρμογή χτίστηκε πάνω σε έναν υπολογιστή ικανό να διαχειριστεί την μεγάλη πυκνότητα σε στοιχεία και να παρέχει υπολογιστική ισχύ τόση ώστε όλα να γίνονται όσο το δυνατό γρηγορότερα. Επίσης κάθε γλώσσα προγραμματισμού και βιβλιοθήκης ήταν στην νεότερη έκδοση. Πιο συγκεκριμένα:

CPU	Intel i7 7700HQ
GPU	nVidia GeForce GTX 1050 Ti
RAM	16GB
SSD	512GB
OS	Windows 10

Πίνακας 6: Τεχνικά χαρακτηριστικά του Server.

HTML	HTML5
CSS	CSS3
JavaScript	ECMAScript6
PHP	7.3
SQL	MySQL 8

Πίνακας 7: Εκδόσεις γλωσσών προγραμματισμού που χρησιμοποιήθηκαν.

jQuery	3.4.1
jQuery UI	1.12.1
Chart.js	2.8
html2canvas	1
dropzone.js	5.2
Overlap.js	1

Πίνακας 8: Εκδόσεις βιβλιοθηκών που χρησιμοποιήθηκαν.

ΚΕΦΑΛΑΙΟ 3

WEVIAN

3.1 Λειτουργικότητα

Ακολουθεί μια παρουσίαση του εργαλείου WEVIAN, με έμφαση στα μέρη και τις λειτουργίες που υποστηρίζει. Η παρουσίαση είναι πλήρης και ακολουθεί τα βήματα ενός χρήστη που επισκέπτεται για πρώτη φορά την διαδικτυακή εφαρμογή, δηλαδή από την εγγραφή του μέχρι και την εμφάνιση της προσομοίωσης.

Αφού ο χρήστης πληκτρολογήσει την διεύθυνση της διαδικτυακής εφαρμογής στη μπάρα διευθύνσεων του τοπικού του browser εισέρχεται σε ένα περιβάλλον ασφαλούς σύνδεσης. Πρόκειται για μια σελίδα με 2 επιλογές, η μία είναι Register δηλαδή εγγραφή ενός νέου χρήστη και η άλλη Login δηλαδή είσοδος στο εργαλείο.

Εάν ο χρήστης δεν έχει στη διάθεσή του τα στοιχεία σύνδεσης που απαιτούνται για να κάνει είσοδο τότε θα πρέπει να κάνει εγγραφή. Η εγγραφή γίνεται με έναν πολύ απλό τρόπο. Ο χρήστης εισάγει το email του στο πεδίο "Email" της φόρμας εγγραφής που βρίσκεται στο δεξιό τμήμα του παραθύρου του browser και στη συνέχεια ακριβώς από κάτω εισάγει έναν επιθυμητό κωδικό στο πεδίο "Password". Αφού συμπληρώσει την φόρμα εγγραφής ο χρήστης θα πρέπει να πατήσει το κουμπί "Register" ώστε τα δεδομένα που εισήγαγε να αποσταλούν στο server και στη βάση δεδομένων κατ' επέκταση. Αφού ο server ανακατευθύνει τον χρήστη τότε είναι έτοιμος να συνδεθεί.

Θα πρέπει να σημειωθεί ότι κατά την αποθήκευση των στοιχείων χρήστη στο server χρησιμοποιείται αλγόριθμος κρυπτογράφησης του κωδικού χρήστη. Αυτό γίνεται μέσα από 3 βήματα

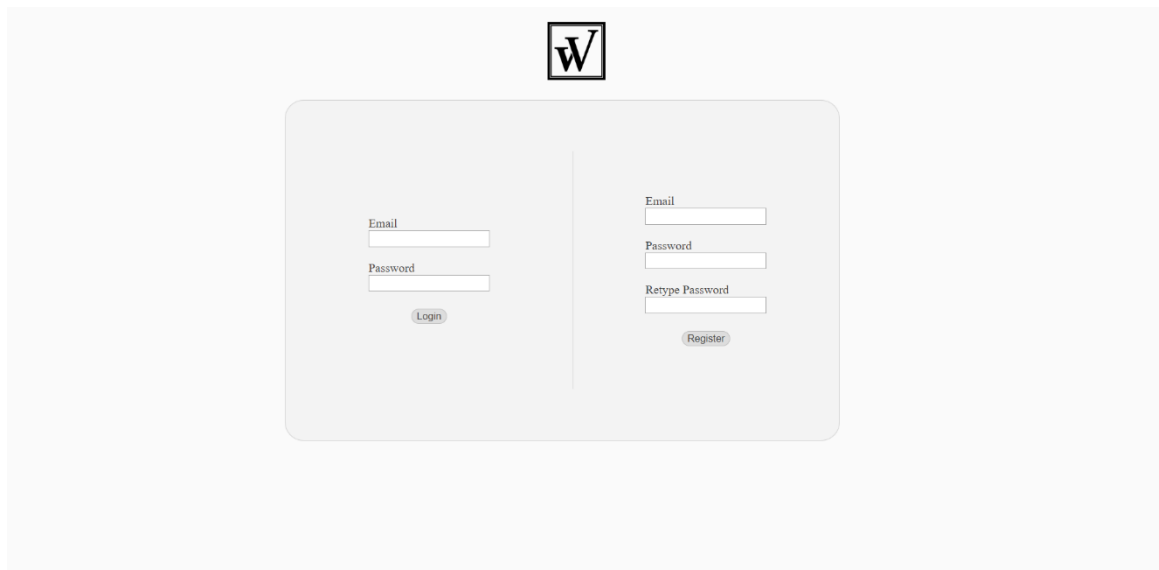
1. Ο server χαρτογραφεί το κωδικό μέσα από μια συνάρτηση hash και παράγει μια συμβολοσειρά σταθερού μεγέθους ανεξάρτητα από το μέγεθος σε χαρακτήρες του κωδικού χρήστη.
2. Στη συνέχεια προστίθενται τυχαίοι χαρακτήρες εμπρός ή πίσω από την παραχθείσα συμβολοσειρά, οι οποίοι ονομάζονται salt.

3. Ο server μεταφέρει την τελική συμβολοσειρά στη βάση δεδομένων και αποθηκεύεται σε αυτή τη μορφή αντί απλού κειμένου.

Έτσι, δεν υπάρχει πρόβλημα διάρρευσης των προσωπικών δεδομένων σε ενδεχόμενη παραβίαση της βάσης δεδομένων από τρίτους.

Εφόσον ο χρήστης έχει τα στοιχεία σύνδεσης, εισάγει στο αριστερό τμήμα του παραθύρου του browser το email του στο πεδίο “Email” της φόρμας σύνδεσης και στη συνέχεια ακριβώς από κάτω εισάγει το μυστικό κωδικό στο πεδίο “Password”. Αφού συμπληρώσει την φόρμα εγγραφής ο χρήστης θα πρέπει να πατήσει το κουμπί “Login” ώστε τα δεδομένα που εισήγαγε να αποσταλούν στο server για να γίνει επαλήθευση.

Αφού ο server λάβει τα στοιχεία του χρήστη, εκτελεί ένα query στη βάση δεδομένων για να επαληθεύσει ότι τα στοιχεία αντιστοιχούν σε κάποιον λογαριασμό. Επειδή όμως ο κωδικός στη βάση δεδομένων δεν είναι συγκρίσιμος με τον κωδικό που έχει ο server, διότι το ένα είναι hashed ενώ το άλλο είναι απλό κείμενο, θα πρέπει να μετατραπεί πρώτα ο κωδικός στην ίδια μορφή με αυτό της βάσης δεδομένων, έπειτα να αφαιρεθεί το salt και τέλος να γίνει σύγκριση των 2 κωδικών. Αν η σύγκριση είναι αληθής τότε ο κωδικός που εισήγαγε ο χρήστης ήταν σωστός, επομένως εισέρχεται στο εργαλείο, διαφορετικά εμφανίζεται ανάλογο ενημερωτικό μήνυμα.



Εικόνα 9: Αρχική σελίδα.

Ο χρήστης στη συνέχεια ανακατευθύνεται αυτόματα στην σελίδα επιλογής project. Όπως προαναφέρθηκε σε προηγούμενο κεφάλαιο, η παρούσα πτυχιακή εργασία θα

επικεντρωθεί στις προσομοιώσεις κυκλωμάτων στο Floorplan και Placement στάδιο της φυσικής σχεδίασης. Επομένως, η σελίδα επιλογής project αφορά την επιλογή Floorplan ή Placement όπως φαίνεται στην Εικόνα 10.



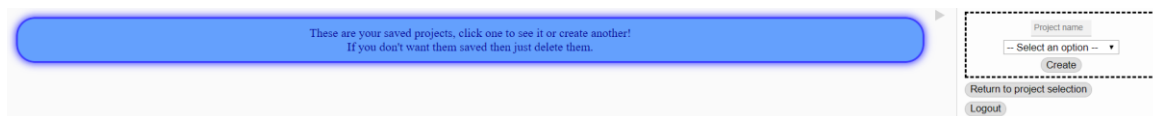
Εικόνα 10: Επιλογή μεταξύ των project.

3.1.1 Floorplan

Προσομοίωση

Κατά την επιλογή του Floorplan Visualization & Analysis ο χρήστης εισέρχεται σε ένα νέο μενού επιλογών. Στο μενού αυτό η οθόνη χωρίζεται σε 2 τμήματα. Στο μεγαλύτερο και αριστερό τμήμα ο χρήστης μπορεί να δει παλιές αποθηκευμένες προσομοιώσεις και να τις ξανατρέξει, βέβαια εάν δεν έχει ξανακάνει προσομοίωση το τμήμα αυτό παραμένει κενό. Στο μικρότερο τμήμα στα δεξιά της οθόνης του χρήστη βρίσκονται όλες οι διαθέσιμες ενέργειες, οι οποίες είναι:

1. Διαδικασία εκκίνησης νέας προσομοίωσης κυκλώματος.
2. Επιστροφή στην επιλογή project.
3. Αποσύνδεση λογαριασμού.



Εικόνα 11: Μενού επιλογών Floorplan Visualization & Analysis.

Κατά την εκκίνηση νέας προσομοίωσης ο χρήστης καλείται να εισάγει ένα όνομα στην προσομοίωσή του για αναγνωριστικούς λόγους. Στη συνέχεια επιλέγει το τύπο του κυκλώματος από ένα dropdown μενού. Τέλος, η εφαρμογή θα κάνει refresh ώστε η νέα προσομοίωση να αποθηκευτεί και να εμφανιστεί στη λίστα των αποθηκευμένων.



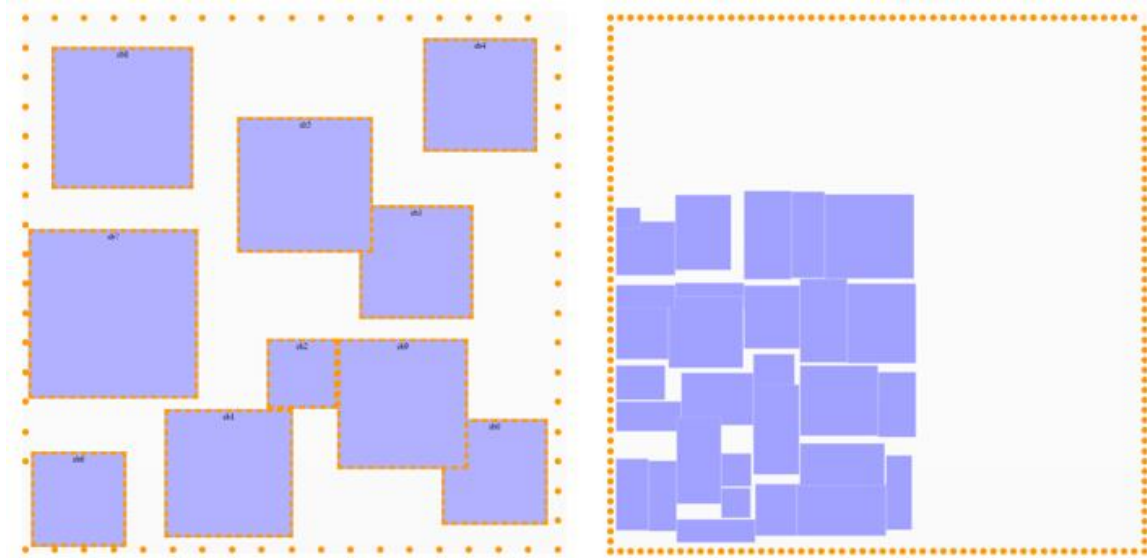
Εικόνα 12: Αποθηκευμένα project στο μενού.

Οι καρτέλες των αποθηκευμένων προσομοιώσεων περιλαμβάνουν τα στοιχεία τους και κάποιες ενέργειες, όπως αναφέρονται στον Πίνακα 9.

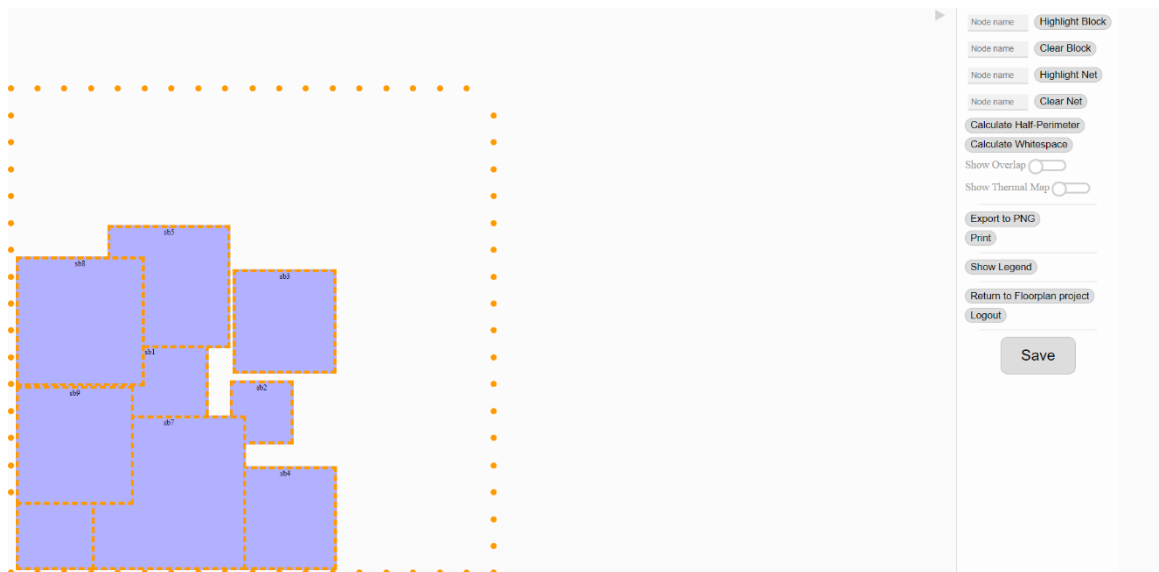
Στοιχεία	Ενέργειες
Όνομα αποθηκευμένης προσομοίωσης	Εκκίνηση προσομοίωσης κυκλώματος
Τύπος κυκλώματος	Σύνδεσμος διαμοιρασμού προσομοίωσης
Ημερομηνία αποθήκευσης	Διαγραφή αποθηκευμένης προσομοίωσης

Πίνακας 9: Στοιχεία και Ενέργειες αποθηκευμένων προσομοιώσεων.

Όταν ο χρήστης επιθυμεί μπορεί να εκκινήσει την προσομοίωση του κυκλώματος που τον ενδιαφέρει. Ο server θα αναλάβει το parsing όλων των στοιχείων για το εκάστοτε κύκλωμα και μέσα από τον αλγόριθμο προσομοίωσης θα εμφανιστεί στο παράθυρο του χρήστη μια νέα οθόνη. Όπως φαίνεται στις Εικόνες 13 και 14 η νέα οθόνη χωρίζεται σε 2 τμήματα. Αριστερά είναι το τμήμα προσομοίωσης και δεξιά το τμήμα ενεργειών.



Εικόνα 13: Τμήμα προσομοίωσης 2 διαφορετικών κυκλωμάτων.

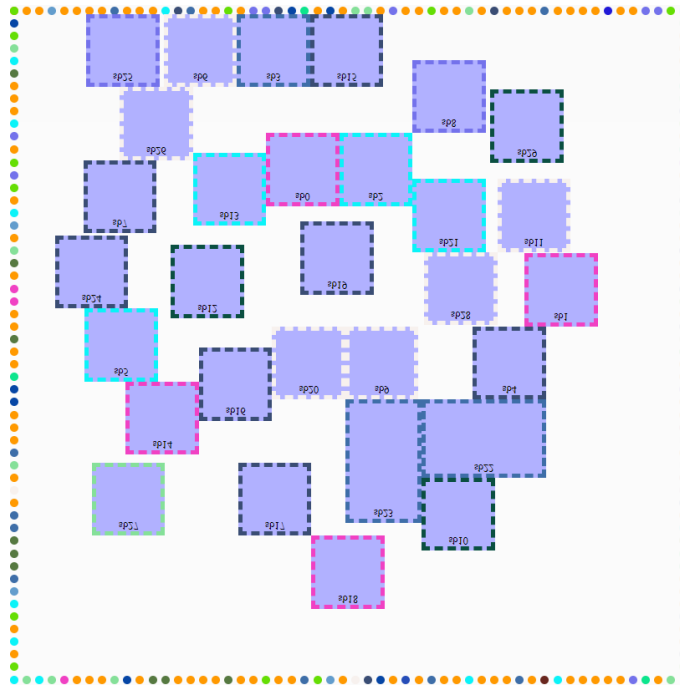


Εικόνα 14: Τμήμα προσομοίωσης και ενεργειών.

Ο αλγόριθμος προσομοίωσης αφού λάβει τα απαραίτητα δεδομένα από τον server σε μορφή JSON αναλαμβάνει σε επίπεδο χρήστη να κάνει render τα δεδομένα ως block προσαρμοσμένων διαστάσεων σε συγκεκριμένη τοποθεσία πάνω στο εικονικό chip. Ως πρώτο βήμα οριοθετεί το chip και κάνει render τα pads. Έπειτα, ταξινομεί τα blocks ως προς τις συντεταγμένες που ανήκουν από αριστερά προς τα δεξιά και από πάνω προς τα κάτω ώστε να είναι προσπελάσιμα για τις μετέπειτα λειτουργικότητες. Στη συνέχεια θεωρεί ένα tag div της HTML για κάθε block. Κάθε εικονικό block έχει δικό του ύψος, πλάτος και

συντεταγμένες για να ορίσουν την ακριβή τοποθεσία του πάνω στο chip. Αφού γίνουν render όλα τα blocks τότε έχει σχηματιστεί το τμήμα προσομοίωσης.

Το τμήμα προσομοίωσης είναι διαδραστικό με την έννοια ότι κάθε block είναι επιλέξιμο, draggable και resizable. Αν γίνει κλικ σε ένα block τότε αυτομάτως θα εμφανιστούν όλα τα block που ανήκουν στο ίδιο netlist με αυτό, δηλαδή οι μεταξύ συνδέσεις των στοιχείων του κυκλώματος. Για την καλύτερη αναγνωσιμότητα κάθε φορά τα nets έχουν τυχαία διαφορετικό χρώμα μέσα από ένα εύρος έντονων χρωμάτων.



Εικόνα 15: Επιλεγμένα nets σε διαφορετικούς χρωματισμούς.

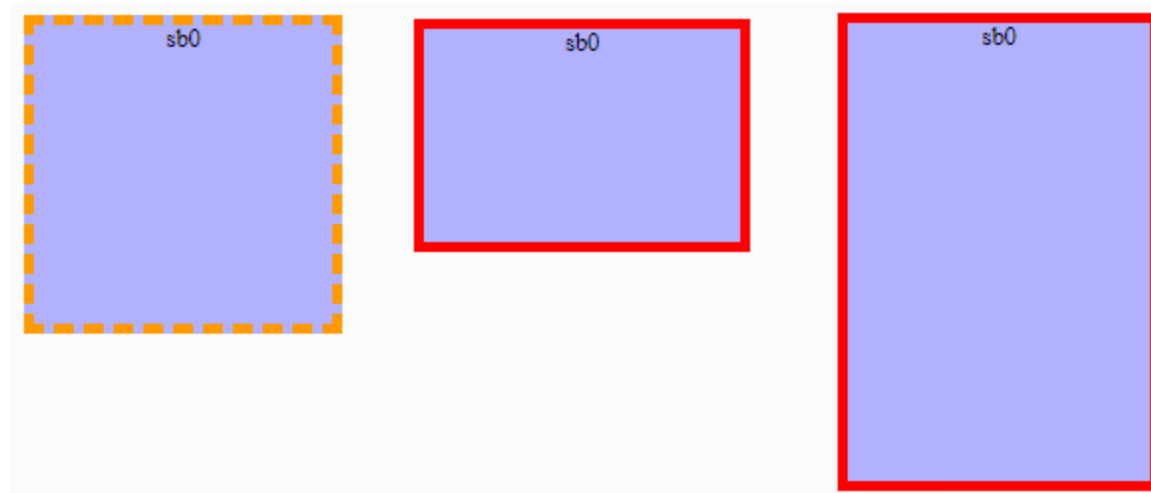
Επιπλέον, με ctrl+click σε ένα block τότε γίνεται highlight και εμφανίζονται σχετικές πληροφορίες στο τμήμα ενεργειών.

Name	X	Y	x
sb20	313	373	Delete row
sb23	263	373	Delete row
sb12	192	276	Delete row
sb17	192	371	Delete row

Εικόνα 16: Πίνακας πληροφοριών των block.

Τα blocks γίνονται drag για την αλλαγή της τοποθεσίας τους πάνω στο chip. Κατά το drag των block εκτελείται αλγόριθμος snapping για την τοποθέτηση των block με απόλυτη ακρίβεια και καλύτερη απόδοση. Ο αλγόριθμος snapping ελέγχει αν ένα block πλησιάζει κάποιο άλλο και μεταφέρει το block ακριβώς στα όρια του διπλανού με αποτέλεσμα την τέλεια επαφή και αποφυγή επέκτασης του whitespace.

Επίσης, στο διαδραστικό κομμάτι της προσομοίωσης τα block γίνονται resizable. Ο χρήστης μπορεί να αλλάξει το μέγεθος ενός block με 3 τρόπους. Επιλέγοντας μια γωνία για αύξηση του μεγέθους και από τις 2 πλευρές ταυτόχρονα, επιλέγοντας μια πλευρά του άξονα x για να αλλάξει το μέγεθος μόνο προς τα πάνω ή προς τα κάτω ή επιλέγοντας μια πλευρά του άξονα y για να αλλάξει το μέγεθος μόνο προς τα αριστερά ή προς τα δεξιά. Κατά την αλλαγή μεγέθους των block ο αλγόριθμος ελέγχει το aspect ratio και επιτρέπει την αλλαγή μόνο αν το aspect ratio εμπίπτει στα επιτρεπτά όρια και αν το εμβαδό παραμένει ίδιο, διαφορετικά το border του block χρωματίζεται κόκκινο ώστε ο χρήστης να αντιληφθεί το σφάλμα.



Εικόνα 17: Resized blocks.

Η υπόλοιπη λειτουργικότητα βρίσκεται στο τμήμα ενεργειών και αποτελεί το πιο σημαντικό κομμάτι του Floorplan.

Τα cells και nets μπορούν να γίνουν highlight εισάγοντας το όνομα ενός cell ή net group ή να καθαριστούν τα ήδη highlighted cells. Η επιλογή ενός cell ή net είναι αντίστοιχη με τη λειτουργία click και ctrl+click που προαναφέρθηκε. Επίσης, για κάθε cell που γίνεται highlight εμφανίζονται σχετικές πληροφορίες στο τμήμα ενεργειών (βλ. Εικόνα 16).

Η επόμενη λειτουργικότητα στο Floorplan Visualization & Analysis είναι από τις πιο σημαντικές και αφορά την καταγραφή του συνολικού μήκους καλωδίου. Ο υπολογισμός του συνολικού μήκους καλωδίου είναι ένας παγκόσμιος τρόπος μέτρησης της ποιότητας οποιουδήποτε placer.^[39]

Ο τρόπος υπολογισμού είναι απλός αλλά ιδιαίτερος. Για κάθε net group υπολογίζεται η μέγιστη και ελάχιστη συντεταγμένη (x,y) των block. Στη συνέχεια υπάρχει ένας μετρητής ο οποίος αυξάνεται σε κάθε net group κατά την πρόσθεση των διαφορών του μέγιστου X και του ελάχιστου X ανά net group με των διαφορών του μέγιστου Y και του ελάχιστου Y ανά net group. Αυτός ο τρόπος υπολογισμού ονομάζεται ημιπερίμετρος. Έτσι, στο τέλος ο μετρητής περιέχει την τιμή του συνολικού μήκους καλωδίου.

Calculate Half-Perimeter

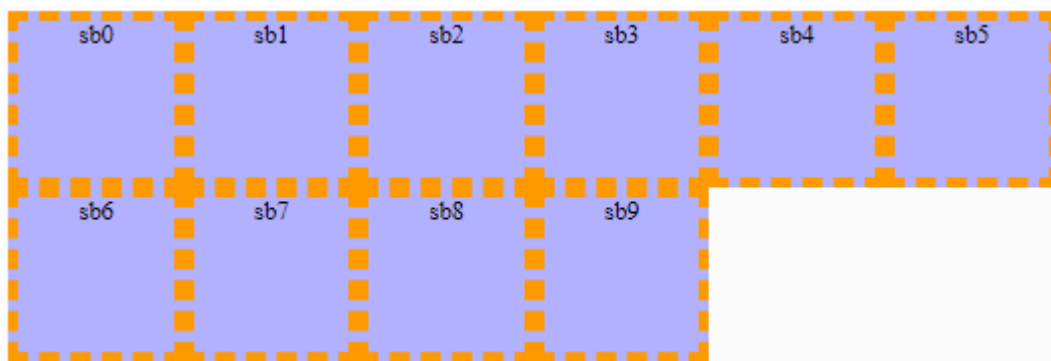
Half-Perimeter is 64158.

Εικόνα 18: Υπολογισμός συνολικού μήκους καλωδίου.

Το συνολικό whitespace μεταξύ των block μπορεί να υπολογιστεί πατώντας το αντίστοιχο κουμπί. Το whitespace υπολογίζεται με βάση τα παρακάτω βήματα:

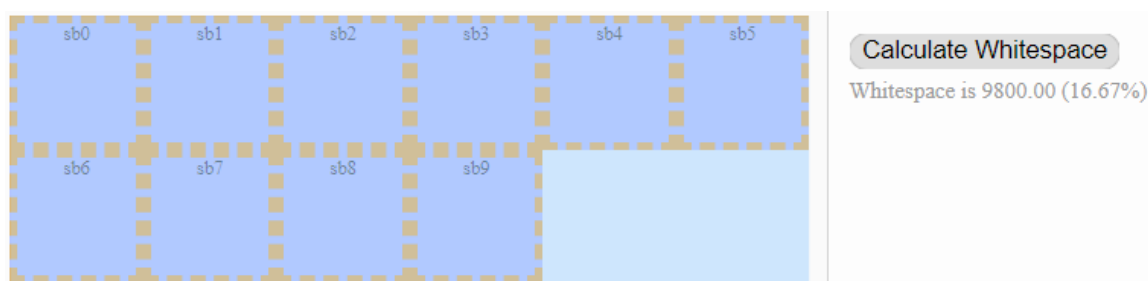
1. Υπολογίζονται οι ελάχιστες και μέγιστες συντεταγμένες από όλα τα κινητά block.
2. Από αυτές τις συντεταγμένες δημιουργείται ένα ορθογώνιο που περικλείει όλα τα κινητά block.
3. Υπολογίζεται το εμβαδό αυτού του ορθογωνίου.
4. Αφαιρούνται τα εμβαδά όλων των block από το εμβαδό του ορθογωνίου.
5. Αυτό που απομένει πλέον είναι το whitespace.

Το whitespace αφού υπολογιστεί εμφανίζεται σε απόλυτη τιμή και σε ποσοστό. Επίσης για 3 δευτερόλεπτα ο χρήστης μπορεί να δει το ορθογώνιο σχηματισμένο πάνω στην προσομοίωση ώστε να αντιληφθεί πιο γρήγορα ενδεχόμενα κενά. Για παράδειγμα, σε ένα κύκλωμα με 10 τετράγωνα blocks τα οποία τα 6 είναι διατεταγμένα το ένα δίπλα στο άλλο και τα υπόλοιπα 4 ακριβώς από κάτω τους έτσι ώστε όλα τα blocks να εφάπτονται με τους γείτονές τους τότε δημιουργείται ένα κενό ίσο με το πλάτος 2 block (βλ. Εικόνα 19).



Εικόνα 19: Τετράγωνα blocks σε τυχαία τοποθεσία.

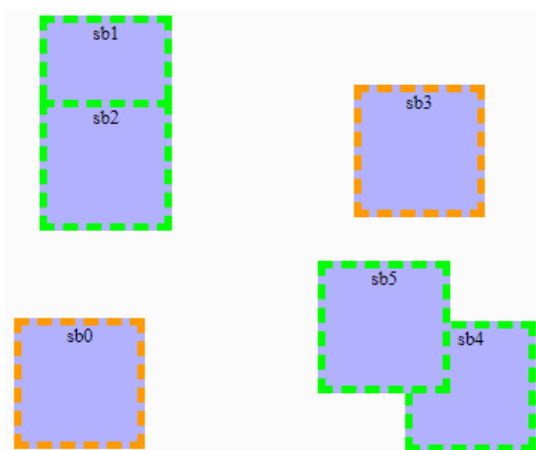
Κατά τον υπολογισμό του whitespace του παραπάνω παραδείγματος εμφανίζεται το ορθογώνιο που περικλείει όλα τα blocks χρωματίζοντας το κενό χώρο στο τμήμα προσομοίωσης, καθώς και η απόλυτη τιμή και το ποσοστό του whitespace στο τμήμα ενεργειών.



Εικόνα 20: Υπολογισμός whitespace.

Τα overlap μπορούν να εντοπιστούν και να γίνουν highlight βοηθώντας το χρήστη να δει τις επικαλύψεις των block. Κατά την εκτέλεση του αλγορίθμου εύρεσης overlap ελέγχονται όλα τα block με βάση τα παρακάτω βήματα:

1. Κάθε block ελέγχεται με όλα τα άλλα.
2. Για κάθε έλεγχο προσδιορίζεται πιο από τα 2 block βρίσκεται αριστερότερα.
3. Αν η δεξιά πλευρά του αριστερότερου block είναι μετά την αριστερή μεριά του δεξιότερου block τότε υπάρχει πιθανή επικάλυψη. Διαφορετικά δεν υπάρχει επικάλυψη.
4. Στη συνέχεια προσδιορίζεται πιο από τα 2 block βρίσκεται ανώτερα.
5. Υπάρχει σίγουρη επικάλυψη εφόσον το κάτω σημείο του ανώτερου block είναι μετά την πάνω μεριά του κατώτερου block. Διαφορετικά δεν υπάρχει επικάλυψη.

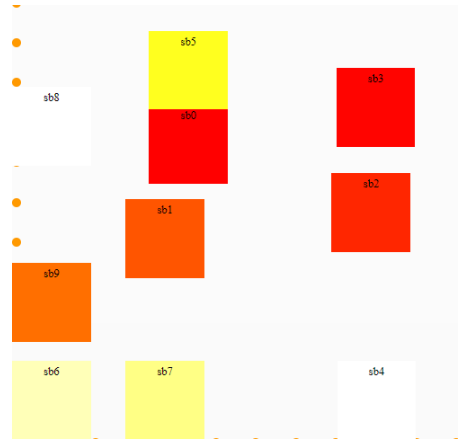


Εικόνα 21: Επικαλυπτόμενα blocks.

Ακόμη, το Floorplan περιλαμβάνει λειτουργία θερμικού χάρτη. Ο θερμικός χάρτης παράγεται από την ατομική θερμότητα του κάθε block. Οι πληροφορίες για τις θερμοότητες των block εισάγονται ή επεξεργάζονται on-the-fly από το εργαλείο. Ο θερμικός χάρτης αποτελείται από το προσομοιωμένο κύκλωμα χρωματισμένο στη κλίμακα άσπρο-κόκκινο, όπου άσπρο είναι η χαμηλότερη θερμοκρασία και κόκκινο η μέγιστη. Στη συνέχεια παρουσιάζεται η διαδικασία εισαγωγής θερμοκρασιών για κάθε block:

1. Επιλέγεται η ενέργεια “Show Thermal Map”.
2. Εμφανίζεται μήνυμα τύπου alert για την επεξεργασία των θερμοκρασιών.
3. Αν ο χρήστης δεν συμφωνήσει τότε εμφανίζεται αυτόματα ο θερμικός χάρτης με τις υπάρχουσες αποθηκευμένες θερμοκρασίες ή 0 σε περίπτωση που δεν έχουν τεθεί ποτέ.
4. Αν ο χρήστης συμφωνήσει τότε εμφανίζεται παράθυρο τύπου modal με περιεχόμενο έναν πίνακα 2 κολόνων και δυναμικών σειρών. Κάθε σειρά του πίνακα είναι 1 block για όσα block υπάρχουν. Η πρώτη κολόνα είναι το όνομα του block και η δεύτερη πεδίο εισαγωγής θερμοκρασίας (βλ. Εικόνα 22).
5. Όταν ο χρήστης τελειώσει την επεξεργασία αποθηκεύει τις θερμοκρασίες και εμφανίζει το θερμικό χάρτη. Οι θερμοκρασίες αποθηκεύονται όπως και το κύκλωμα, στο server.

Node Name	Temperature
sb0	<input type="text" value="30"/>
sb1	<input type="text" value="20"/>
sb2	<input type="text" value="25"/>
sb3	<input type="text" value="29"/>
sb4	<input type="text" value="0"/>
sb5	<input type="text" value="13"/>
sb6	<input type="text" value="4"/>
sb7	<input type="text" value="7"/>
sb8	<input type="text" value="0"/>
sb9	<input type="text" value="17"/>

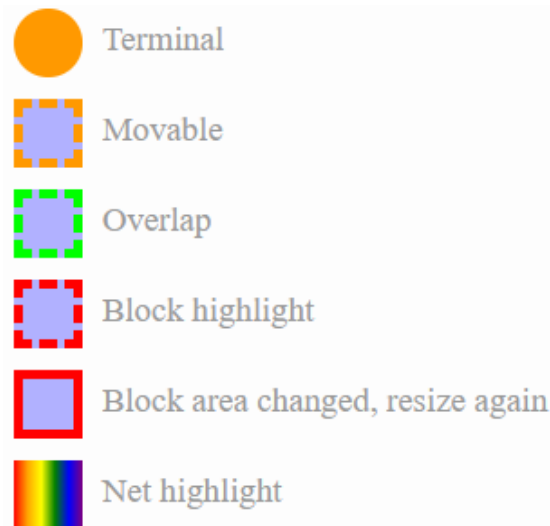


Εικόνα 22: Επεξεργασία θερμοκρασιών.

Εικόνα 23: Δημιουργία και απεικόνιση θερμικού χάρτη.

Το κύκλωμα μπορεί να εκτυπωθεί ή να εξαχθεί σε εικόνα μορφής PNG υψηλής ευκρίνειας. Αφού επιλεγθεί το κουμπί “Export to PNG” από το τμήμα ενεργειών, εξάγεται η προσομοίωση σε εικόνα όπως είναι εκείνη τη στιγμή χωρίς όμως να φαίνεται το τμήμα ενεργειών.

Η επεξήγηση των χρωμάτων που έχουν τα block φαίνεται στο υπόμνημα εφόσον πιεστεί το κουμπί “Show Legend”.



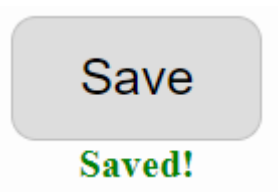
Εικόνα 24: Υπόμνημα προσομοίωσης.

Όπου συμπαγές πορτοκαλί είναι τα pad, μπλε με πορτοκαλί border τα κινητά block, μπλε με πράσινο border τα επικαλυπτόμενα block, μπλε με κόκκινο border τα highlighted

block, μπλε με solid κόκκινο border τα block με αλλαγμένο εμβαδό και τυχαία χρωματισμένα τα highlighted net.

Επιπρόσθετα, ο χρήστης μπορεί να γυρίσει στις αποθηκευμένες προσομοιώσεις ή να κάνει αποσύνδεση από το εργαλείο.

Τέλος, όλο το κύκλωμα αφού έχει επεξεργασθεί ή υπάρξουν μετακινήσεις, αλλαγές μεγέθους και τοποθεσίας των block μπορεί να αποθηκευτεί στο server και να συνεχιστεί η επεξεργασία του από το ίδιο σημείο. Με την επιλογή “Save” αποθηκεύεται η κατάσταση του κυκλώματος σε πραγματικό χρόνο ειδοποιώντας τον χρήστη κατά την ολοκλήρωση της ενέργειας.



Εικόνα 25: Ειδοποίηση ολοκλήρωσης αποθήκευσης.

Αυτή ήταν η διαδικασία προσομοίωσης ενός κυκλώματος στο Floorplan Visualization & Analysis, από την εγγραφή του χρήστη μέχρι και την παραγωγή των αποτελεσμάτων. Ωστόσο, υπάρχει μία ακόμα λειτουργικότητα του Floorplan, αυτή της διαμοίρασης κυκλωμάτων.

Διαμοιρασμός

Η τελευταία λειτουργία του Floorplan είναι η δυνατότητα διαμοιρασμού ενός κυκλώματος με άλλους χρήστες. Η λειτουργία αυτή μετασχηματίζει το εργαλείο από ερευνητικό εργαλείο σε πλατφόρμα. Η λειτουργία υλοποιείται μέσα από κοινόχρηστους συνδέσμους.

Οι κοινόχρηστοι σύνδεσμοι επιτρέπουν την αντιγραφή και διαμοιρασμό συνδέσμων με άλλους χρήστες τόσο εντός όσο και εκτός της πλατφόρμας, χωρίς να είναι αναγκαία η δημιουργία λογαριασμού για να επισκεφτεί κάποιος τον σύνδεσμο. Οι κοινόχρηστοι σύνδεσμοι είναι διευθύνσεις URL των αποθηκευμένων κυκλωμάτων. Όταν παράγεται ένας κοινόχρηστος σύνδεσμος τότε το κύκλωμα ορίζεται ως δημόσιο, δίνοντας πρόσβαση σε

άτομα με αυτό το URL να δούνε το κύκλωμα και να εκτελέσουν όλες τις λειτουργίες χωρίς, όμως, να έχουν πρόσβαση στην μετακίνηση και αλλαγή μεγέθους των block.

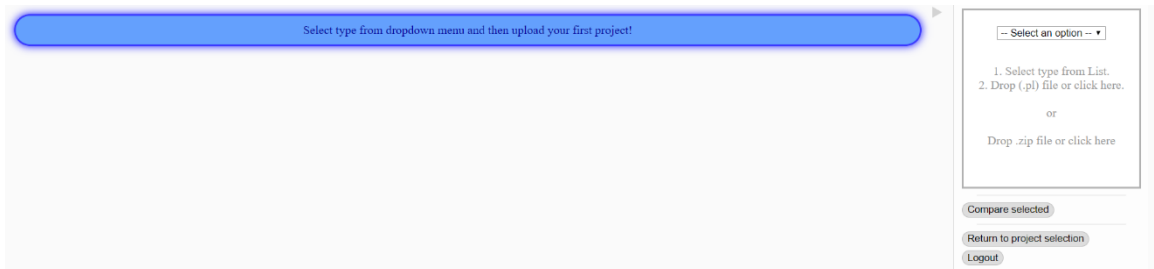
Ο χρήστης καθώς βρίσκεται στο κεντρικό μενού του Floorplan Visualization & Analysis επιλέγει το κουμπί “Share link” ενός κυκλώματος και αυτόματα δημιουργείται ένας κοινόχρηστος σύνδεσμος ο οποίος αποθηκεύεται στο clipboard. Κάνοντας paste το σύνδεσμο στη μπάρα διευθύνσεων του τοπικού browser η οθόνη χωρίζεται σε 2 τμήματα. Το αριστερό τμήμα περιέχει την προσομοίωση του κυκλώματος και το δεξιό όλες τις λειτουργίες που αναφέρθηκαν στην προηγούμενη ενότητα.

3.1.2 Placement

Προσομοίωση

Κατά την επιλογή του Placement Visualization & Analysis ο χρήστης εισέρχεται σε ένα νέο μενού επιλογών. Στο μενού αυτό η οθόνη χωρίζεται σε 2 τμήματα. Στο μεγαλύτερο και αριστερό τμήμα ο χρήστης μπορεί να δει παλιές αποθηκευμένες προσομοιώσεις και να τις ξανατρέξει, βέβαια εάν δεν έχει ξανακάνει προσομοίωση το τμήμα αυτό παραμένει κενό. Στο μικρότερο τμήμα στα δεξιά της οθόνης του χρήστη βρίσκονται όλες οι διαθέσιμες ενέργειες, οι οποίες είναι:

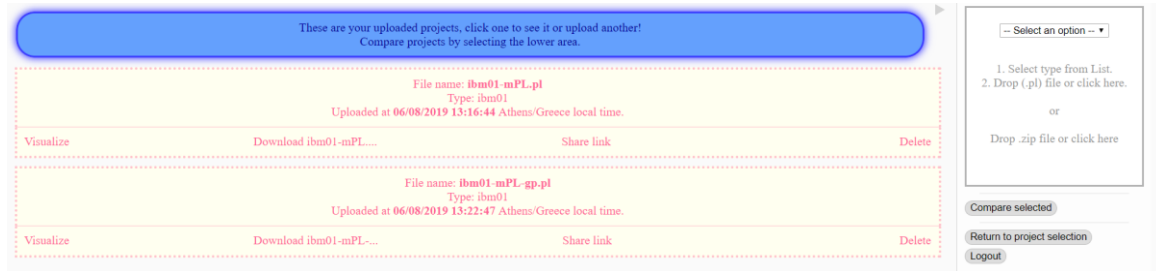
1. Διαδικασία εκκίνησης νέας προσομοίωσης κυκλώματος.
2. Σύγκριση 2 αποθηκευμένων προσομοιώσεων με αναλυτικά στοιχεία και στατιστικά για το καθένα.
3. Επιστροφή στην επιλογή project.
4. Αποσύνδεση λογαριασμού.



Εικόνα 26: Μενού επιλογών Placement Visualization & Analysis.

Κατά την εκκίνηση νέας προσομοίωσης ο χρήστης καλείται να επιλέξει το τύπο του κυκλώματος από ένα dropdown μενού. Στη συνέχεια θα πρέπει να ανεβάσει το .pl αρχείο

το οποίο περιέχει τις συντεταγμένες και τον προσανατολισμό των cell πάνω στο chip. Εναλλακτικά ο χρήστης μπορεί να ανεβάσει αρχείο .zip. Τέλος, η εφαρμογή θα ανεβάσει το αρχείο και θα κάνει refresh ώστε η νέα προσομοίωση να αποθηκευτεί και να εμφανιστεί στη λίστα των αποθηκευμένων.



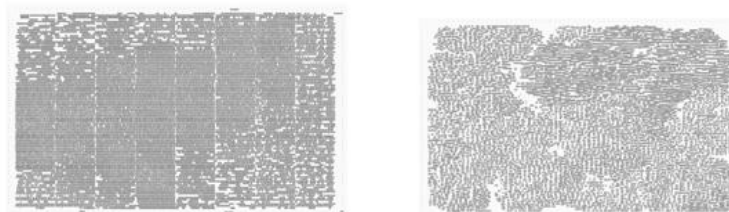
Εικόνα 27: Αποθηκευμένα project στο μενού.

Οι καρτέλες των αποθηκευμένων προσομοιώσεων περιλαμβάνουν τα στοιχεία τους και κάποιες ενέργειες, όπως αναφέρονται στον Πίνακα 10.

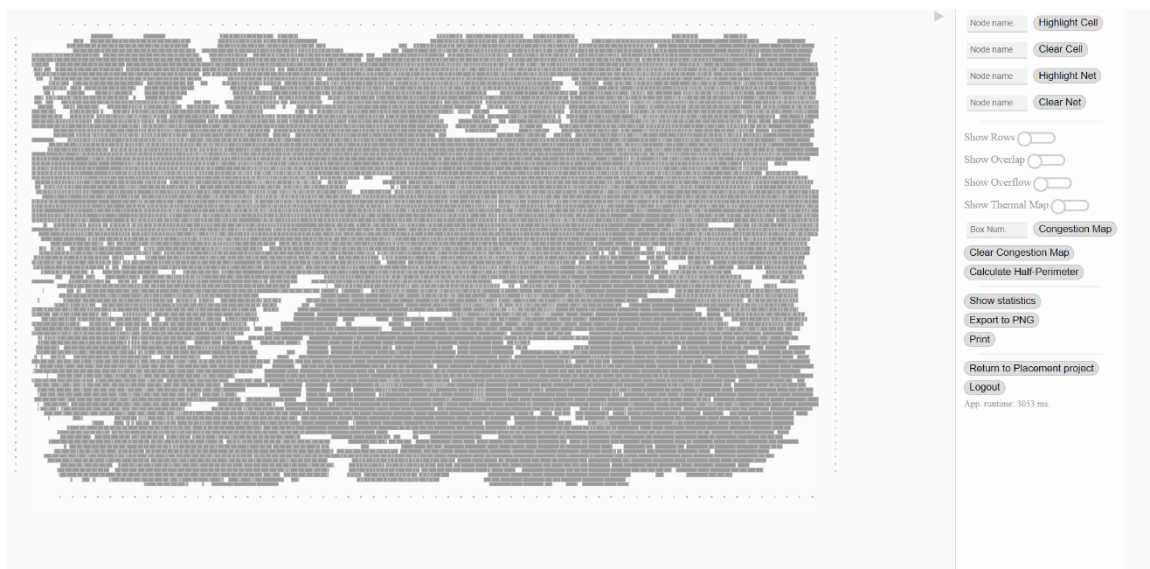
Στοιχεία	Ενέργειες
Όνομα αρχείου .pl	Εκκίνηση προσομοίωσης κυκλώματος
Τύπος κυκλώματος	Κατέβασμα αρχείου .pl
Ημερομηνία αποθήκευσης	Σύνδεσμος διαμοιρασμού προσομοίωσης
	Διαγραφή αποθηκευμένης προσομοίωσης

Πίνακας 10: Στοιχεία και Ενέργειες αποθηκευμένων προσομοιώσεων.

Όταν ο χρήστης επιθυμεί μπορεί να εκκινήσει την προσομοίωση του κυκλώματος που τον ενδιαφέρει. Ο server θα αναλάβει το parsing όλων των στοιχείων για το εκάστοτε κύκλωμα και μέσα από τον αλγόριθμο προσομοίωσης θα εμφανιστεί στο παράθυρο του χρήστη μια νέα οθόνη. Όπως φαίνεται στις Εικόνες 28 και 29 η νέα οθόνη χωρίζεται σε 2 τμήματα. Αριστερά είναι το τμήμα προσομοίωσης και δεξιά το τμήμα ενεργειών.



Εικόνα 28: Τμήμα προσομοίωσης 2 διαφορετικών κυκλωμάτων.



Εικόνα 29: Τμήμα προσομοίωσης και ενεργειών.

Ο αλγόριθμος προσομοίωσης αφού λάβει τα απαραίτητα δεδομένα από τον server σε μορφή JSON αναλαμβάνει σε επίπεδο χρήστη να κάνει render τα δεδομένα ως cell προσαρμοσμένων διαστάσεων σε συγκεκριμένη τοποθεσία πάνω στο εικονικό chip. Ως πρώτο βήμα οριοθετεί το chip και κάνει render τις σειρές του πάνω στις οποίες θα τοποθετηθούν τα cells. Έπειτα, ταξινομεί τα cell ως προς τις συντεταγμένες που ανήκουν από αριστερά προς τα δεξιά και από πάνω προς τα κάτω ώστε να είναι προσπελάσιμα για τις μετέπειτα λειτουργικότητες. Στη συνέχεια θεωρεί ένα tag div της HTML για κάθε cell. Κάθε εικονικό cell έχει δικό του ύψος, πλάτος και συντεταγμένες για να ορίσουν την ακριβή τοποθεσία του πάνω στο chip. Αφού γίνουν render όλα τα cells τότε έχει σχηματιστεί το τμήμα προσομοίωσης.

Το τμήμα προσομοίωσης είναι διαδραστικό με την έννοια ότι κάθε cell είναι επιλέξιμο. Αν ο χρήστης κάνει κλικ σε ένα cell τότε αυτομάτως θα εμφανιστούν όλα τα cell που ανήκουν στο ίδιο netlist με αυτό, δηλαδή οι μεταξύ συνδέσεις των στοιχείων του κυκλώματος. Πέρα από το απλό κλικ, αν ο χρήστης κάνει ctrl+click τότε γίνεται highlight το επιλεγμένο cell και εμφανίζονται σχετικές πληροφορίες.

Η υπόλοιπη λειτουργικότητα βρίσκεται στο τμήμα ενεργειών και αποτελεί το πιο σημαντικό κομμάτι του Placement.

Τα cells και nets μπορούν να γίνουν highlight εισάγοντας το όνομα ενός cell ή net group ή να καθαριστούν τα ήδη highlighted cells προκειμένου να αποκτήσετε καλύτερη εικόνα. Η λειτουργία αυτή σχεδιάστηκε και υλοποιήθηκε από τον Ν. Κούγκαλη.

Ο χρήστης μπορεί να εμφανίσει τις σειρές του κυκλώματος για καλύτερη εποπτεία των ορίων του chip.



Εικόνα 30: Αριστερά με εμφάνιση σειρών. Δεξιά χωρίς εμφάνιση σειρών.

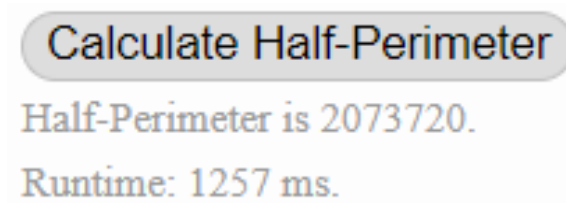
Τα overlap και/ή τα overflow μπορούν να εντοπιστούν και να γίνουν highlight βοηθώντας το χρήστη να δει τις επικαλύψεις των cell ή ποια cell βγαίνουν εκτός του μέγιστου ορίου του chip. Η λειτουργία του overlap σχεδιάστηκε και υλοποιήθηκε από τον Ν. Κούγκαλη ενώ η λειτουργία του overflow σχεδιάστηκε και υλοποιήθηκε από τον Γ. Δεμεσιώτη.

Ακόμη, ο Placement visualizer περιλαμβάνει ενέργειες για την χαρτογράφηση του κυκλώματος. Υποστηρίζονται 2 ειδών χαρτογραφήσεις. Η πρώτη είναι θερμικός χάρτης και η δεύτερη χάρτης συμφόρησης ή πυκνότητας. Ο θερμικός χάρτης παράγεται από την ατομική θερμότητα του κάθε cell. Οι πληροφορίες για τις θερμότητες των cell εξάγονται από το αρχείο .wts του κυκλώματος. Ο θερμικός χάρτης αποτελείται από το προσομοιωμένο κύκλωμα χρωματισμένο στη κλίμακα άσπρο-κόκκινο, όπου άσπρο είναι η χαμηλότερη θερμοκρασία και κόκκινο η μέγιστη. Η λειτουργία του θερμικού χάρτη σχεδιάστηκε και υλοποιήθηκε από τον Γ. Δεμεσιώτη. Στο χάρτη συμφόρησης το κύκλωμα διασπάται σε ίσα τετράγωνα τμήματα, που ονομάζονται bins, ανάλογα με το input του χρήστη και το προσομοιωμένο κύκλωμα χρωματίζεται στη κλίμακα άσπρο-μπλε, όπου άσπρο είναι η μηδενική συμφόρηση ενώ μπλε η μέγιστη. Η λειτουργία του χάρτη συμφόρησης σχεδιάστηκε και υλοποιήθηκε από τον Ν. Κούγκαλη.

Η επόμενη λειτουργικότητα στο Placement είναι ίσως η πιο σημαντική και αφορά την καταγραφή του συνολικού μήκους καλωδίου.

Ο τρόπος υπολογισμού είναι παρόμοιος με αυτόν του Floorplan. Για κάθε net group υπολογίζεται η μέγιστη και ελάχιστη συντεταγμένη (x,y) των cell. Στη συνέχεια υπάρχει

έναν μετρητή ο οποίος αυξάνεται σε κάθε net group κατά την πρόσθεση των διαφορών του μέγιστου X και του ελάχιστου X ανά net group με των διαφορών του μέγιστου Y και του ελάχιστου Y ανά net group. Έτσι, στο τέλος ο μετρητής περιέχει την τιμή του συνολικού μήκους καλωδίου.



Εικόνα 31: Υπολογισμός συνολικού μήκους καλωδίου.

Το εργαλείο WEVIAN πέρα από προσομοιωτής είναι και αναλυτής και αυτό διότι παράγει στατιστικά, γραφήματα και πληροφορίες για το εκάστοτε κύκλωμα. Όταν ο χρήστης πατήσει στο αντίστοιχο κουμπί θα εμφανιστεί ένα παράθυρο τύπου modal.

Η ανάλυση ξεκινάει με έναν πίνακα που περιέχει συγκεντρωμένα τα στοιχεία του κυκλώματος.

Η πρώτη στήλη περιέχει:

1. NumNodes: Το πλήθος των cell. Αφορά όλα τα cells.
2. NumTerminals: Το πλήθος των ακίνητων cell. Περιλαμβάνει τα pads που περικλείουν τα κινητά cells.
3. NumRows: Το πλήθος των σειρών του chip.
4. NumNets: Το πλήθος των net group. Κάθε NumNet περιέχει 1 net, δηλαδή οι μεταξύ συνδέσεις ενός cell με άλλα.
5. NumPins: Ο συνολικός αριθμός όλων των συνδέσεων από κάθε NumNet.
6. Half-Perimeter: Η τιμή του συνολικού μήκους καλωδίου.

Η δεύτερη στήλη περιέχει:

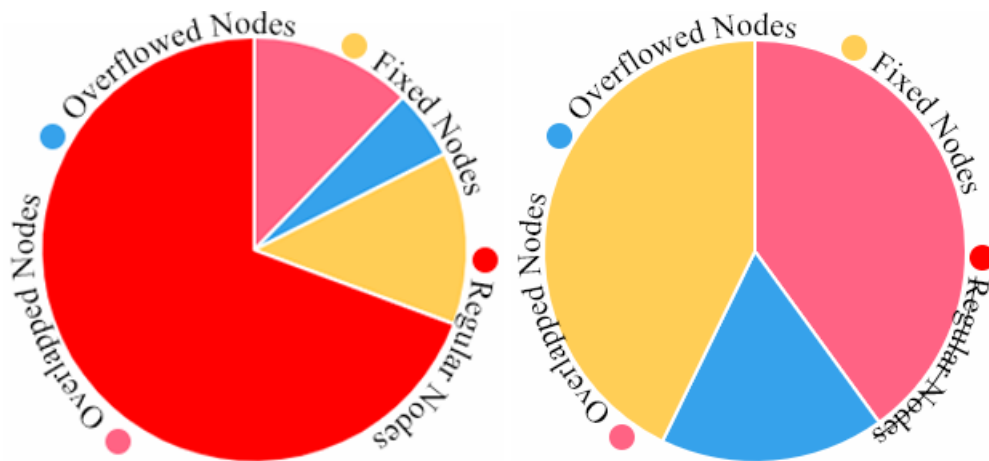
1. Overlaps: Το πλήθος των επικαλυπτόμενων cell.
2. Overflows: Το πλήθος των cell που είναι εκτός του ορίου του chip.
3. Min. Temperature: Η ελάχιστη θερμοκρασία που σημειώθηκε στον θερμικό χάρτη.
4. Max. Temperature: Η μέγιστη θερμοκρασία που σημειώθηκε στον θερμικό χάρτη.
5. Min. Area: Το ελάχιστο εμβαδό που σημειώθηκε στον χάρτη συμφόρησης.
6. Max. Area: Το μέγιστο εμβαδό που σημειώθηκε στο χάρτη συμφόρησης.

NumNodes	12752	230	Overlaps
NumTerminals	246	6	Overflows
NumRows	96	0	Min. Temperature
NumNets	14111	704	Max. Temperature
NumPins	50566	1728	Min. Area
Half-Perimeter	2403624.240060999	7872	Max. Area

Εικόνα 32: Συγκεντρωτικός πίνακας στοιχείων μετά το πέρας της ανάλυσης του κυκλώματος.

Τα γραφήματα που παράγονται είναι 4. Γράφημα πίτας, γράφημα γραμμής τύπου point style και 2 γραφήματα bar.

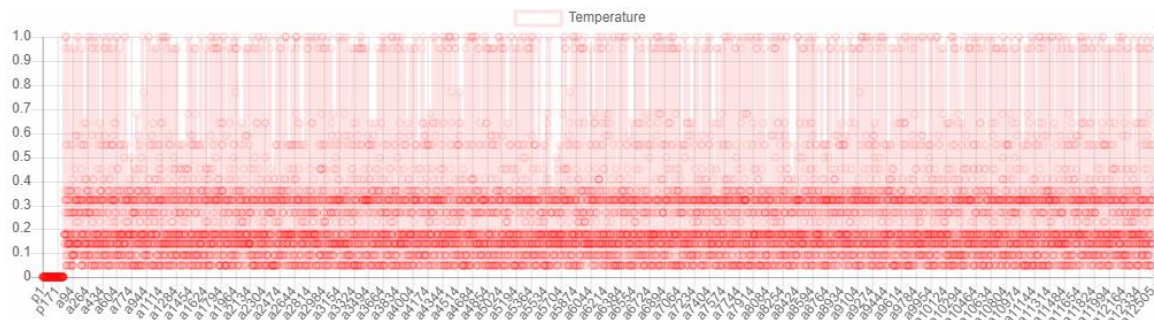
Το γράφημα πίτας αναπαριστά τα σύνολα των overlap, overflow, pad και κινητών cell. Εμφανίζονται σε αναλογίες πόσα από τα cells είναι overlap, overflow ή pad. Κάθε τμήμα έχει αντίστοιχα μοναδικό χρώμα. Επίσης, το γράφημα πίτας είναι διαδραστικό, ο χρήστης μπορεί να πατήσει στο υπόμνημα του γραφήματος ώστε να κρύψει το αντίστοιχο κομμάτι από το γράφημα ή να το ξαναεμφανίσει.



Εικόνα 33: Overlap, overflow, pad και κινητά cell.

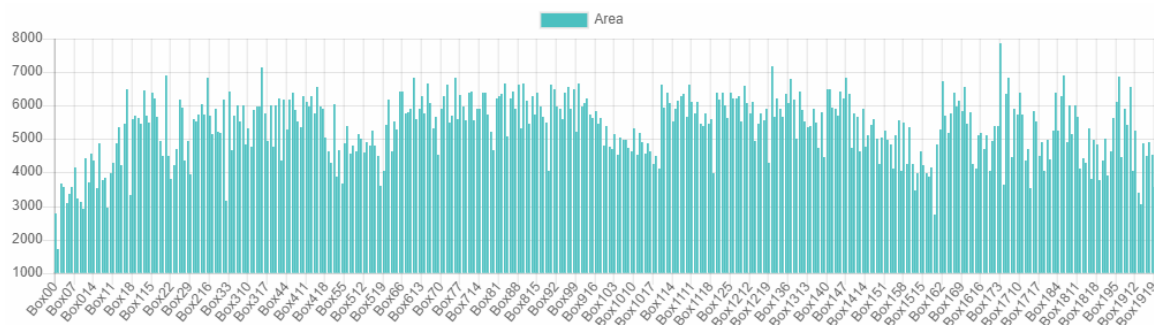
Το γράφημα γραμμής τύπου point style περιλαμβάνει όλα τα cells και τα κατατάσσει από το 0 έως το 1 με βάση την θερμοκρασία τους, έτσι προκύπτει ένα είδος θερμικού χάρτη και γίνεται εύκολα αντιληπτό σε ποιες θερμοκρασίες κυμαίνονται. Η διαφορά με το θερμικό χάρτη που προαναφέρθηκε είναι ότι σε αυτή την περίπτωση τα cell κατατάσσονται ανάλογα με την θερμοκρασία τους ενώ στην άλλη περίπτωση τα cell εμφανίζονται συναρτήσει της τοποθεσίας τους πάνω στο chip. Άξιο να σημειωθεί το γεγονός ότι τα pads

δεν λαμβάνουν μέρος στη δημιουργία του θερμικού χάρτη και επομένως βρίσκονται στην θερμοκρασία 0.



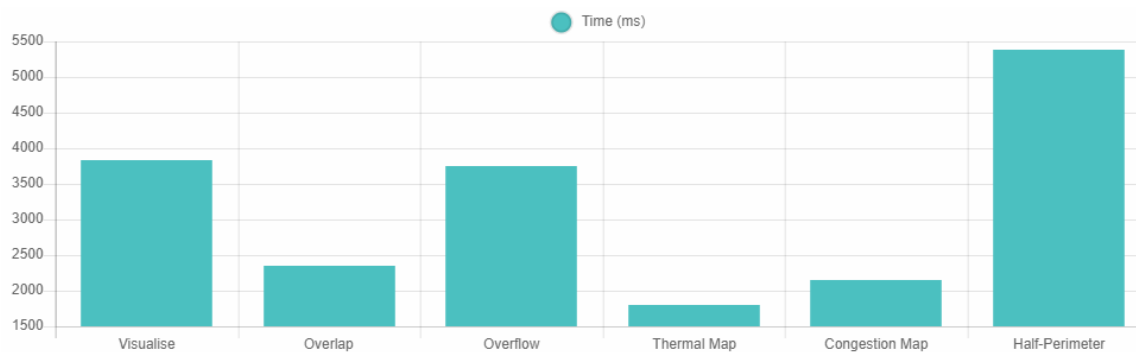
Εικόνα 34: Γράφημα θερμοκρασιών.

Το πρώτο εκ των 2 bar γράφημα αναπαριστά τα bin που δημιουργήθηκαν από τον χάρτη συμφόρησης συναρτήσει των εμβαδών τους. Η διαφορά με το χάρτη συμφόρησης που προαναφέρθηκε είναι ότι σε αυτή την περίπτωση τα bin εμφανίζονται σε μορφή μπαρών και είναι πιο ευδιάκριτος ο μέσος όρος των εμβαδών. Το παράδειγμα της Εικόνας 20 βασίζεται σε χάρτη συμφόρησης 400 bin.



Εικόνα 35: Γράφημα χάρτη συμφόρησης.

Το τελευταίο γράφημα αλλά και η τελευταία λειτουργία του αναλυτή του Placement είναι το bar γράφημα αναπαράστασης των χρόνων για τις διάφορες λειτουργίες του αναλυτή. Το σημαντικότερο κομμάτι ενός placement αλγορίθμου είναι να είναι γρήγορος, η επίτευξη της μέγιστης ταχύτητας βασίζεται πάνω στην αποδοτικότερη τοποθεσία των cell. Για αυτό το λόγο είναι πολύ σημαντικό να γνωρίζει ο ερευνητής πόσο είναι το overhead, δηλαδή η χρονική επιβάρυνση, του chip και να έχει την δυνατότητα να την συγκρίνει στο μέλλον με ένα νέο chip.



Εικόνα 36: Γράφημα χρονικών επιβαρύνσεων των λειτουργιών.

Τέλος, το κύκλωμα μπορεί να εκτυπωθεί ή να εξαχθεί σε εικόνα μορφής PNG (αυτή η λειτουργία σχεδιάστηκε και υλοποιήθηκε από τον Γ. Δεμεσιώτη), να γυρίσει στις αποθηκευμένες προσομοιώσεις ή να κάνει αποσύνδεση από το εργαλείο.

Αυτή ήταν η διαδικασία προσομοίωσης ενός κυκλώματος στο Placement Analysis & Visualization, από την εγγραφή του χρήστη μέχρι και την παραγωγή των αποτελεσμάτων. Ωστόσο, υπάρχουν 2 ακόμα λειτουργικότητες του Placement, αυτή της σύγκρισης και διαμοίρασης κυκλωμάτων.

Σύγκριση

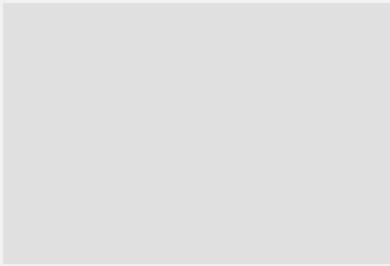
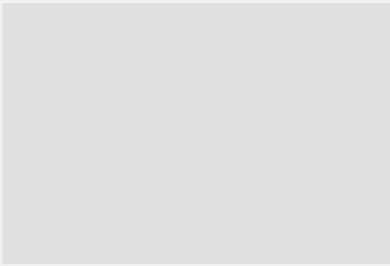
Κατά τη σύγκριση 2 αποθηκευμένων κυκλωμάτων ο χρήστης πρέπει να βρίσκεται στο μενού επιλογών του Placement Visualization & Analysis και να επιλέξει τα 2 αποθηκευμένα κυκλώματα που επιθυμεί. Για να επιλέξει ένα κύκλωμα πατάει σε κενό σημείο στο κάτω μέρος του αντίστοιχου κυκλώματος, το panel έπειτα θεωρείται επιλεγμένο και θα αλλάξει χρωματισμό. Η ίδια διαδικασία ακολουθείται και για να αποεπιλεγθεί. Πρέπει να σημειωθεί ότι μόνο 2 κυκλώματα ίδιου τύπου μπορούν να συγκριθούν, για παράδειγμα 2 κυκλώματα τύπου “ibm01”. Αφού γίνει επιλογή, ο χρήστης πρέπει να πιέσει το κουμπί “Compare selected” που βρίσκεται στο δεξιό τμήμα της οθόνης στο τμήμα ενεργειών. Στη συνέχεια αναλύονται και τα 2 κυκλώματα και παράγεται ένα παράθυρο τύπου modal με τα αποτελέσματα.

Τα αποτελέσματα της σύγκρισης χωρίζονται σε 4 μέρη. Το πρώτο μέρος αφορά γενικά χαρακτηριστικά και ένα μικρό διαδραστικό thumbnail των κυκλωμάτων. Το δεύτερο μέρος αφορά τα ειδικά χαρακτηριστικά του τύπου κυκλώματος. Το τρίτο μέρος περιέχει τα πλήθη των overlap, overflow και τα συνολικά μήκη καλωδίου. Το τέταρτο και τελευταίο μέρος

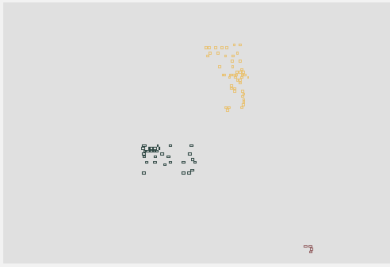

του πάνελ σύγκρισης περιλαμβάνει 1 γράφημα μπαρών το οποίο δείχνει αναλογικά τα στατιστικά του τρίτου μέρους.

Στο πρώτο μέρος ο χρήστης βλέπει έναν πίνακα που περιέχει τα παρακάτω δεδομένα:

1. Name: Τα ονόματα των αρχείων .pl που ανέβασε ως αναγνωριστικά του κάθε κυκλώματος.
2. Thumbnail: Ένα κενό thumbnail για το κάθε κύκλωμα, το hover του mouse πάνω από τα thumbnail κάνουν αυτόματα μεγέθυνση.
3. Actions: Ένα input στο οποίο ο χρήστης εισάγει το όνομα ενός cell και επιλέγει αν θέλει να κάνει highlight το cell ή το net του. Στη συνέχεια στα thumbnail των 2 κυκλωμάτων εμφανίζονται τα cells. Πατώντας το κουμπί clear καθαρίζει όλα τα highlighted cells ή nets.
4. Type: Ο τύπος των κυκλωμάτων.
5. Uploaded: Ημερομηνίες δημιουργίας των κυκλωμάτων.

Name	ibm01_PARABACUS_8threads_panos.pl	ibm01-mPL-gp.pl
Thumbnail		
Actions	Node name <input type="text"/> <input checked="" type="radio"/> Cell <input type="radio"/> Net <input type="button" value="Clear"/>	
Type	ibm01	
Uploaded	12/07/2019 15:23:42	12/07/2019 15:24:01

Εικόνα 37: Το πρώτο μέρος του πάνελ σύγκρισης.

Name	ibm01_PARABACUS_8threads_panos.pl	ibm01-mPL-gp.pl
Thumbnail		
Actions	a200 <input type="radio"/> Cell <input checked="" type="radio"/> Net <input type="button" value="Clear"/>	
Type	ibm01	
Uploaded	12/07/2019 15:23:42	12/07/2019 15:24:01

Εικόνα 38: Το πρώτο μέρος του πάνελ σύγκρισης με 3 net highlighted.

Στο δεύτερο μέρος ο χρήστης βλέπει έναν πίνακα που περιέχει τα παρακάτω δεδομένα:

1. Nodes: Το πλήθος των cell. Αφορά όλα τα cells.
2. Terminals: Το πλήθος των ακίνητων cell. Περιλαμβάνει τα pads που περικλείουν τα κινητά cells.
3. Nets: Το πλήθος των net group. Δηλαδή οι μεταξύ συνδέσεις ενός cell με άλλα.
4. Pins: Ο συνολικός αριθμός όλων των συνδέσεων από κάθε net.
5. Rows: Το πλήθος των σειρών του chip.

Nodes	12752
Terminals	246
Nets	14111
Pins	50566
Rows	96

Εικόνα 39: Το δεύτερο μέρος του πάνελ σύγκρισης.

Στο τρίτο μέρος ο χρήστης βλέπει έναν πίνακα που περιέχει τα παρακάτω δεδομένα:

1. Overlaps: Το πλήθος των επικαλυπτόμενων cell του κάθε κυκλώματος.
2. Overflows: Το πλήθος των cell του κάθε κυκλώματος που τοποθετήθηκαν εκτός των ορίων του εκάστοτε κυκλώματος.
3. Half-Perimeter: Η τιμή του συνολικού μήκους καλωδίου του κάθε κυκλώματος.

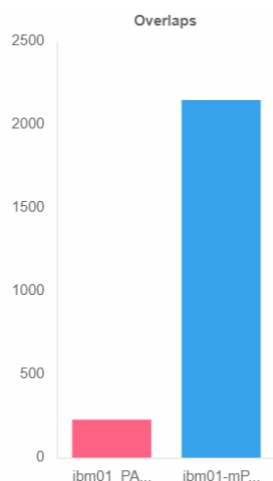
Overlaps	230	2150
Overflows	6	0
Half-Perimeter	2403624.240060999	2067949

Εικόνα 40: Το τρίτο μέρος του πάνελ σύγκρισης.

Στην Εικόνα 40 καταλαβαίνει κανείς την σημαντικότητα του μήκους καλωδίου. Στην πρώτη περίπτωση, το κύκλωμα θα χρειαστεί περίπου 2400000 μονάδες για το καλώδιό του ενώ το δεύτερο κύκλωμα είναι πολύ πιο αποδοτικό με μόλις λίγο περισσότερο από 2000000. Αυτό σημαίνει ότι όταν τα 2 αυτά κυκλώματα φτάσουν στη φάση της παραγωγής, το πρώτο θα χρειαστεί πολύ περισσότερο αγωγίμο καλώδιο από το δεύτερο επομένως περισσότερο κόστος και χειρότερη απόδοση καθώς το ρεύμα θα πρέπει να διανύσει περισσότερη απόσταση. Ωστόσο, και τα 2 αυτά κυκλώματα δεν θα έφταναν ποτέ στην φάση παραγωγής μιας και τα 2 περιέχουν επικαλυπτόμενα cells, αλλά είναι ένα καλό παράδειγμα που δείχνει την αξία στην διαχείριση του μήκους καλωδίου.

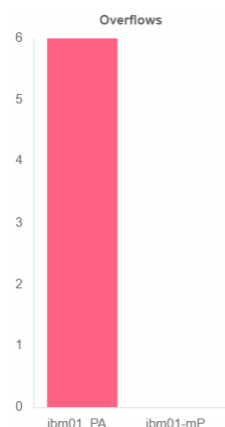
Στο τέταρτο μέρος ο χρήστης βλέπει ένα γράφημα. Το γράφημα αυτό εμφανίζεται on-the-fly ανάλογα σε πια γραμμή του πίνακα του τμήματος 3 κάνει hover με το mouse του ο χρήστης.

Αν ο χρήστης έχει το mouse του πάνω από την γραμμή 1 δηλαδή την γραμμή των Overlap τότε θα εμφανιστεί το γράφημα που δείχνει τα Overlaps και των 2 κυκλωμάτων.



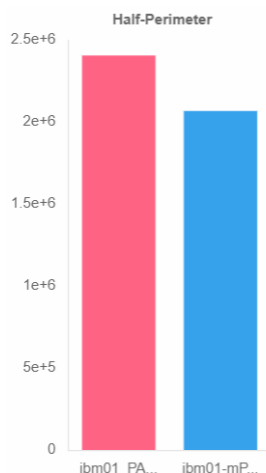
Εικόνα 41: Γράφημα on-the-fly Overlap 2 κυκλωμάτων στο τέταρτο μέρος του πάνελ σύγκρισης.

Αν ο χρήστης έχει το mouse του πάνω από την γραμμή 2 δηλαδή την γραμμή των Overflow τότε θα εμφανιστεί το γράφημα που δείχνει τα Overflows και των 2 κυκλωμάτων.



Εικόνα 42: Γράφημα on-the-fly Overflow 2 κυκλωμάτων στο τέταρτο μέρος του πάνελ σύγκρισης.

Αν ο χρήστης έχει το mouse του πάνω από την γραμμή 3 δηλαδή την γραμμή του Half-Perimeter τότε θα εμφανιστεί το γράφημα που δείχνει τα συνολικά μήκη καλωδίου και των 2 κυκλωμάτων.



Εικόνα 43: Γράφημα on-the-fly Half-Perimeter 2 κυκλωμάτων στο τέταρτο μέρος του πάνελ σύγκρισης.

Όπως φαίνεται από τις Εικόνες 41, 42 και 43 κάθε γράφημα του τέταρτου μέρους του πάνελ σύγκρισης έχει 2 άξονες, ο άξονας των x περιλαμβάνει το όνομα του αρχείου .pl του εκάστοτε κυκλώματος ως αναγνωριστικό ενώ ο άξονας των y περιλαμβάνει την τιμή της εκάστοτε μέτρησης (Overlap, Overflow ή Half-Perimeter Wire Length αντίστοιχα). Επειδή οι τιμές των Half-Perimeter Wire Length συνήθως είναι μεγάλες υπολογίζονται σε scientific μορφή.

Διαμοιρασμός

Η τελευταία λειτουργία του Placement είναι η δυνατότητα διαμοιρασμού ενός κυκλώματος με άλλους χρήστες. Όπως και στο Floorplan, η λειτουργία υλοποιείται μέσα από κοινόχρηστους συνδέσμους μετατρέποντας το εργαλείο σε πλατφόρμα.

Όταν παράγεται ένας κοινόχρηστος σύνδεσμος τότε το κύκλωμα ορίζεται ως δημόσιο, δίνοντας πρόσβαση σε άτομα με αυτό το URL να δούνε το κύκλωμα και να εκτελέσουν όλες τις λειτουργίες χωρίς, όμως, να έχουν πρόσβαση στο αρχείο .pl.

Ο χρήστης καθώς βρίσκεται στο κεντρικό μενού του Placement Visualization & Analysis επιλέγει το κουμπί “Share link” ενός κυκλώματος και αυτόματα δημιουργείται ένας κοινόχρηστος σύνδεσμος ο οποίος αποθηκεύεται στο clipboard. Κάνοντας paste το σύνδεσμο στη μπάρα διευθύνσεων του τοπικού browser η οθόνη χωρίζεται σε 2 τμήματα. Το αριστερό τμήμα περιέχει την προσομοίωση του κυκλώματος και το δεξιό όλες τις λειτουργίες που αναφέρθηκαν στην προηγούμενη ενότητα.

3.2 Πειραματικά αποτελέσματα

3.2.1 Βασικά χαρακτηριστικά

Ακολουθεί μια παρουσίαση και έπειτα ανάλυση των αποτελεσμάτων από διάφορα πειράματα που διεξήχθησαν στο Placement και στο Floorplan. Τα πειράματα για το Placement έγιναν με δεδομένα από τα ISPD'04 benchmark circuits^[40] των οποίων τα χαρακτηριστικά φαίνονται στο Πίνακα 11. Τα κυκλώματα έγιναν placed και legalized από τον αλγόριθμο Domocus.^[41] Τα πειράματα για το Floorplan έγιναν με δεδομένα από το VLSI CAD Bookshelf^[42] των οποίων τα χαρακτηριστικά φαίνονται στον Πίνακα 12. Τα πειράματα έγιναν σε μηχανήμα του οποίου τα χαρακτηριστικά φαίνονται στον Πίνακα 13.

Όνομα	Cells	Pads	Nets	Pins	Rows
ibm01	12506	246	14111	50566	96
ibm02	19342	259	19584	81199	109
ibm03	22853	283	27401	93573	121
ibm04	27220	287	31970	105589	136
ibm05	28146	1201	28446	126308	139
ibm06	32332	166	34826	128182	126
ibm07	45639	287	48117	175639	166
ibm08	51023	286	50513	204890	170
ibm09	53110	285	60902	222088	183
ibm10	68685	744	75196	297567	234
ibm11	70152	406	81454	280786	208
ibm12	70439	637	77240	317760	242
ibm13	83709	490	99666	357075	224
ibm14	147088	517	152772	546816	305
ibm15	161187	383	186608	715823	303
ibm16	182980	504	190048	778823	347
ibm17	184752	743	189581	860036	379
ibm18	210341	272	201920	819697	361

Πίνακας 11: Χαρακτηριστικά ISPD'04 benchmark circuits.[40]

GSRC – SOFT & HARD				
Όνομα	Blocks	Pads	Nets	Pins
n10	10	69	118	248
n30	30	212	349	723
n50	50	209	485	1050
n100	100	334	885	1873
n200	200	564	1585	3599
n300	300	569	1893	4358
MCNC – SOFT & HARD				
Όνομα	Blocks	Pads	Nets	Pins
ami33	33	42	123	522
ami49	49	22	408	953
apte	9	73	97	287
hp	11	45	83	309
xerox	10	2	203	698

Πίνακας 12: Χαρακτηριστικά VLSI CAD Bookshelf.[41]

CPU	Intel i7 7700HQ
GPU	nVidia GeForce GTX 1050 Ti
RAM	16GB
SSD	512GB
OS	Windows 10

Πίνακας 13: Χαρακτηριστικά υπολογιστικού συστήματος.

Όλα τα πειράματα έγιναν με βάση τον μέσο όρο 10 μετρήσεων για καλύτερη αξιοπιστία. Επειδή το εργαλείο WEVIAN βασίζεται κυρίως στην JavaScript, μια γλώσσα προγραμματισμού client side, το μηχάνημα και κατ' επέκταση ο browser έχουν βαρύτητα στην απόδοση του εργαλείου και των μετρήσεων. Για αυτό το λόγο οι μετρήσεις έγιναν σε 3 διαφορετικούς browsers. Οι browsers που χρησιμοποιήθηκαν ήταν ο Chrome, ο Firefox και ο Opera.

3.2.2 Πειραματικά αποτελέσματα

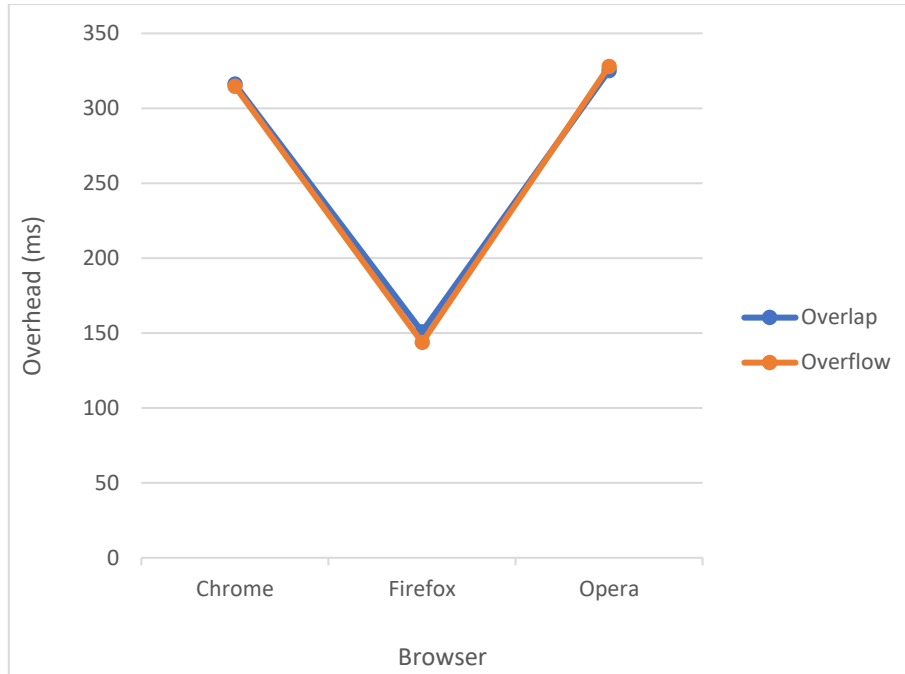
Πειράματα Placement

Στο Διάγραμμα 2 φαίνεται το overhead όπως καταγράφηκε από 2 βασικές λειτουργίες του Placement Analysis & Visualization συναρτήσεων των browser. Οι λειτουργίες Overlap και Overflow είναι από τις πιο σημαντικές διότι ελέγχουν αν έχει γίνει σωστά placed το κύκλωμα. Στο πείραμα αυτό χρησιμοποιήθηκε το κύκλωμα `ibm01`.

Το αποτέλεσμα δείχνει ότι στην καταγραφή Overlap ο Chrome και ο Opera απέχουν 9 μόλις millisecond, δηλαδή μια αμελητέα διαφορά και αυτό συμβαίνει διότι και οι 2 browser χρησιμοποιούν το Chromium engine επομένως οι υπολογισμοί και το πως αντιλαμβάνεται ο browser την JavaScript είναι σχεδόν ίδιοι, από την άλλη μεριά ο Firefox έχει αισθητή διαφορά στην καταγραφή των Overlap με περίπου 165 millisecond από τους άλλους 2 browser. Φυσικά όλες οι μετρήσεις είναι μέσα στα πλαίσια αποδεκτών χρόνων.

Στην καταγραφή του overhead για τα Overflows έχουμε αντίστοιχο αποτέλεσμα ίδιων αναλογιών. Και οι 3 browsers βρίσκονται περίπου στους ίδιους χρόνους με τον Firefox να έχει το μικρότερο overhead, η διαφορά με τους άλλους 2 browser είναι περίπου 175 millisecond.

Συμπερασματικά, η καταγραφή των Overlap και Overflow γίνεται client side μέσω της JavaScript επομένως παίζουν πολύ μεγάλο ρόλο τα τεχνικά χαρακτηριστικά του μηχανήματος και το πως χειρίζεται ο browser τις εντολές της JavaScript.



Διάγραμμα 2: Μέσος χρόνος overhead Overlap & Overflow συναρτήσει διαφορετικών browser.

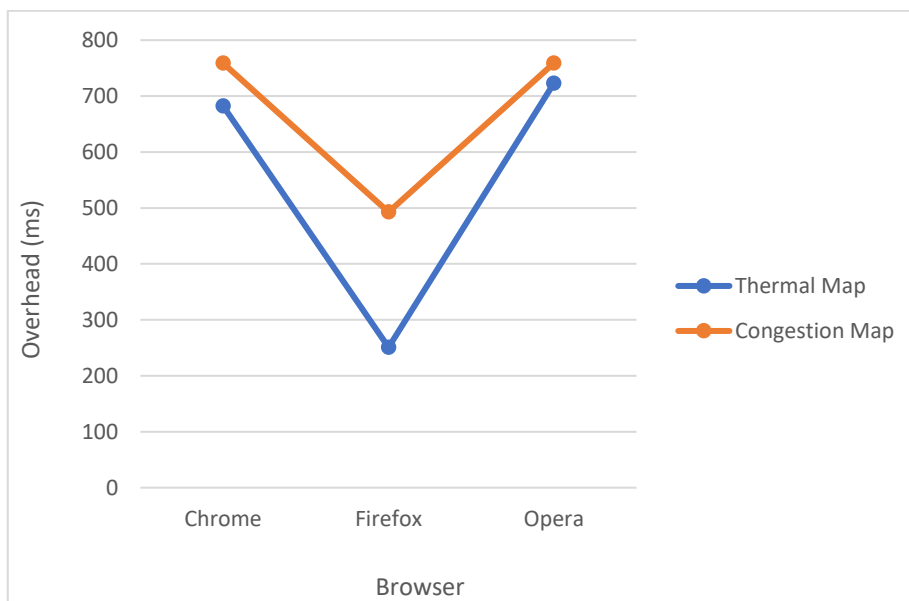
Στο Διάγραμμα 3 φαίνεται το overhead όπως καταγράφηκε από 2 λειτουργίες του Placement Analysis & Visualization συναρτήσει των browser. Οι λειτουργίες Thermal Map και Congestion Map κάνουν χαρτογράφηση του κυκλώματος ως προς τη θερμότητα και την πυκνότητα αντίστοιχα. Στο πείραμα αυτό χρησιμοποιήθηκε το κύκλωμα ibm02. Για το χάρτη πυκνότητας χρησιμοποιήθηκαν 100x100 bins.

Το αποτέλεσμα δείχνει ότι στην καταγραφή Thermal Map ο Chrome και ο Opera απέχουν ελάχιστα μεταξύ τους ωστόσο είναι και οι 2 πιο αργοί από τον Firefox, ο οποίος είναι ταχύτερος κατά περίπου 450 millisecond. Φυσικά όλες οι μετρήσεις είναι μέσα στα πλαίσια αποδεκτών χρόνων.

Στην καταγραφή του overhead για το Congestion Map 200x200 bin έχουμε αποτέλεσμα με σχεδόν ίδιους χρόνους από Chrome και Opera ωστόσο ο Firefox αν και είναι ο πιο γρήγορος από τους 3 με 250 millisecond προβάδισμα, είναι πιο αργός σε σχέση με την απόδοσή του στο Thermal Map.

Συμπερασματικά, φαίνεται πως ο Firefox κάνει καλύτερη διαχείριση πόρων κατά το rendering –χρωματισμένων– element. Ο υπολογισμός του Thermal και Congestion Map γίνεται client side μέσω της JavaScript επομένως παίζουν πολύ μεγάλο ρόλο τα τεχνικά

χαρακτηριστικά του μηχανήματος και το πως χειρίζεται ο browser τις εντολές της JavaScript.

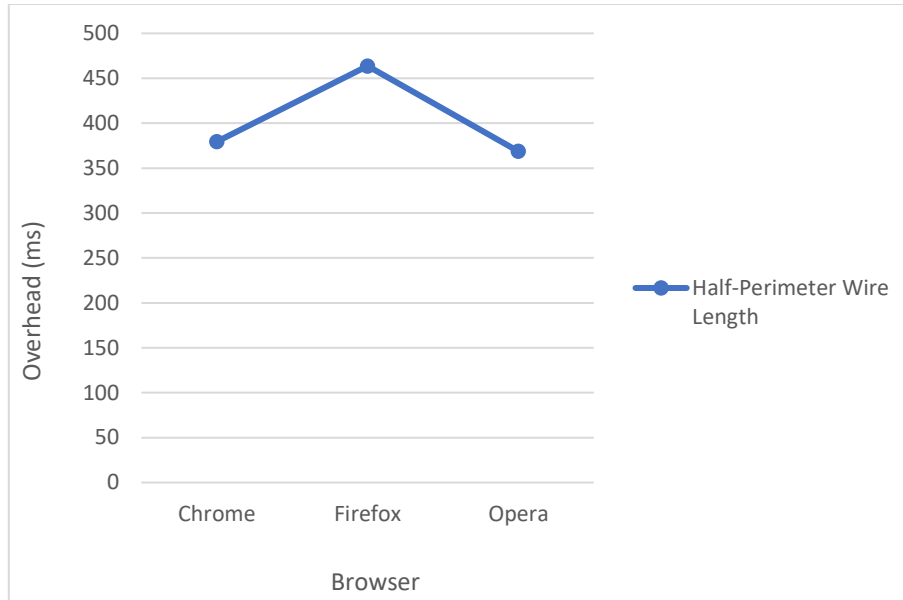


Διάγραμμα 3: Μέσος χρόνος overhead Thermal & Congestion Map συναρτήσει διαφορετικών browser.

Στο Διάγραμμα 4 φαίνεται το overhead όπως καταγράφηκε από το Half-Perimeter Wire Length συναρτήσει των browser. Οι λειτουργία του Half-Perimeter Wire Length κάνει υπολογισμό του συνολικού μήκους καλωδίου. Στο πείραμα αυτό χρησιμοποιήθηκε το κύκλωμα `ibm03`.

Το αποτέλεσμα δείχνει ότι στην καταγραφή του Half-Perimeter Wire Length ο Firefox είναι ο πιο αργός με απόκλιση περίπου 100 millisecond από τον Chrome ενώ ο Opera είναι ο ταχύτερος με διαφορά περίπου 10 millisecond από τον Chrome. Φυσικά όλες οι μετρήσεις είναι μέσα στα πλαίσια αποδεκτών χρόνων.

Κατά τον υπολογισμό του Half-Perimeter Wire Length χρησιμοποιούνται client side εμφωλευμένες εντολές επανάληψης. Για αυτό το λόγο βγαίνει το συμπέρασμα ότι ο Opera έχει την καλύτερη διαχείριση των επαναληπτικών εντολών της JavaScript για μεγάλα datasets.

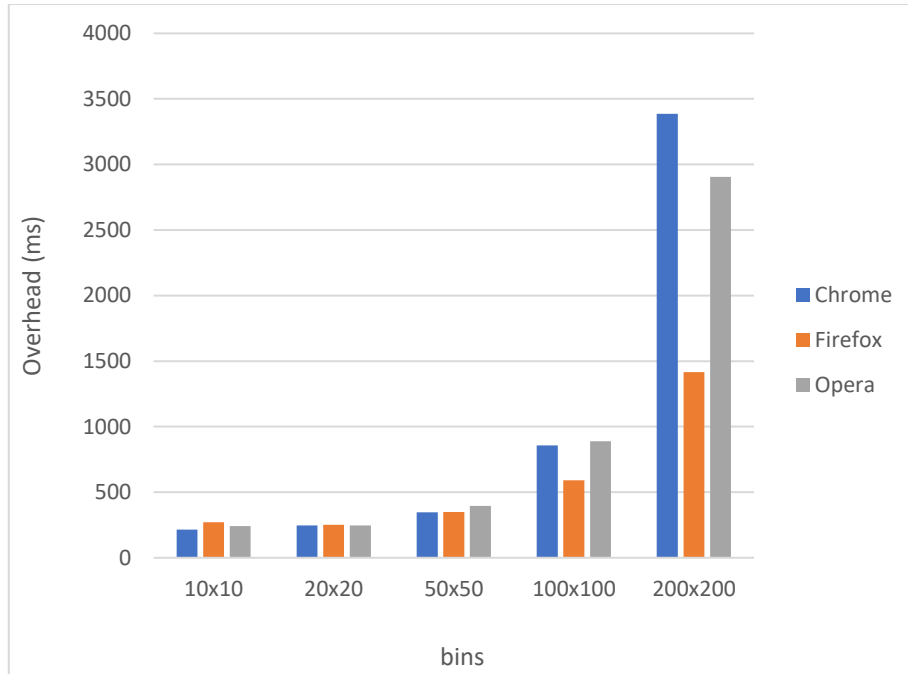


Διάγραμμα 4: Μέσος χρόνος overhead Half-Perimeter Wire Length συναρτήσει διαφορετικών browser.

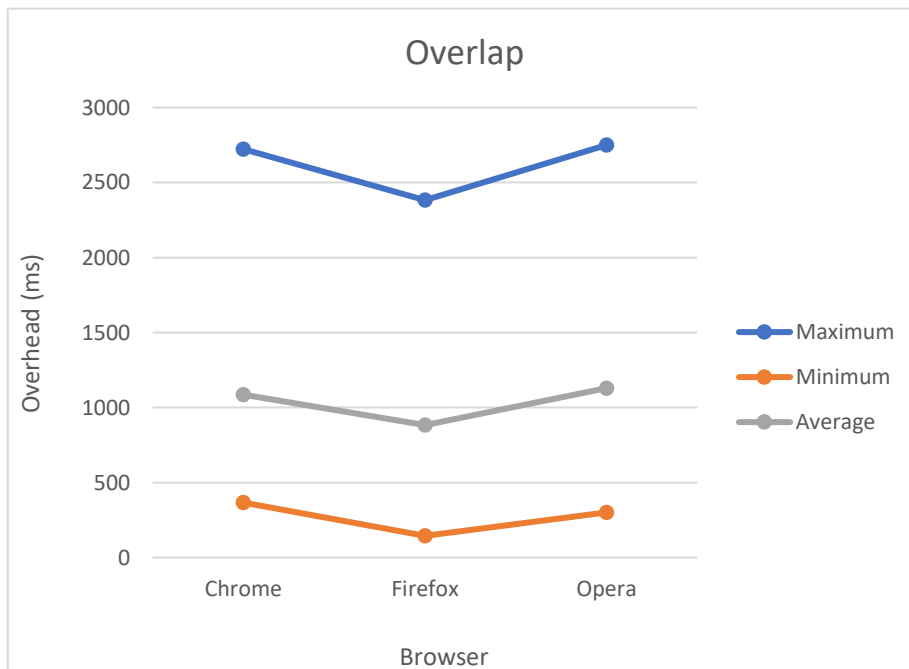
Στο Διάγραμμα 5 φαίνεται το overhead για διαφορετικά bins όπως καταγράφηκε από το Congestion Map του Placement συναρτήσει των browser. Ο χάρτης συμφόρησης χωρίζεται σε bins το πλήθος των οποίων ορίζει ο χρήστης. Στο πείραμα αυτό χρησιμοποιήθηκε το κύκλωμα ibm04.

Το αποτέλεσμα δείχνει ότι στην καταγραφή Congestion Map ο Chrome και ο Opera απέχουν λίγα millisecond μεταξύ τους σε οποιοδήποτε πλήθος bin, από την άλλη μεριά ο Firefox έχει αισθητή διαφορά στην καταγραφή του overhead με φυσιολογική ανοδική πορεία ανάλογα με το πλήθος.

Στα 10x10 bins όλοι οι browsers είναι στους ίδιους περίπου χρόνους διότι είναι πολύ λίγα τα elements που γίνονται rendering. Ωστόσο, όταν διπλασιάζονται, στα 20x20 bins οι χρόνοι αυξάνονται ελάχιστα. Τα overhead παραμένουν στις ίδιες αναλογίες και στα 50x50 bins. Η διαφορά ξεκινάει να γίνεται αισθητή από τα 100x100 bins και μετά καθώς ο Firefox είναι κατά 280 millisecond γρηγορότερος. Τέλος, για ακόμη μια φορά φαίνεται ότι ο Firefox είναι αρκετά ισχυρός στο rendering χρωματισμένων element καθώς σημειώνεται διαφορά 1400-1900 millisecond στα 200x200 bin από τους άλλους 2 browser.



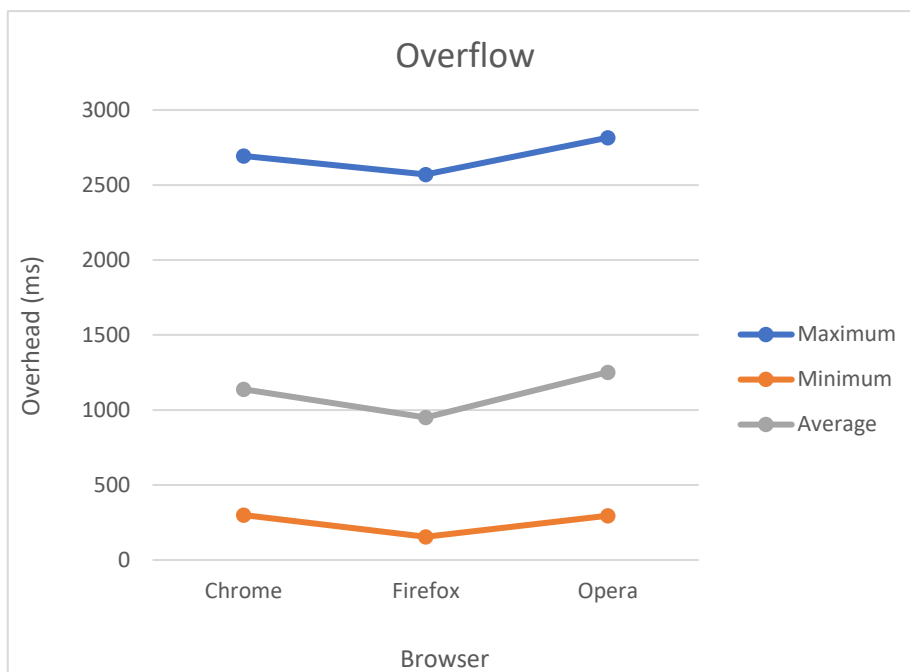
Διάγραμμα 5: Μέσος χρόνος overhead Congestion Map σε διαφορετικούς browser συναρτήσει διαφορετικών αριθμών bin.



Διάγραμμα 6: Μέσος χρόνος overhead Overlap όλων των κυκλωμάτων σε διαφορετικούς browser.

Στο Διάγραμμα 6 φαίνεται το overhead του Overlap στο Placement για όλα τα κυκλώματα συναρτήσει των browser. Τα κυκλώματα περιέχουν όλα ίδιο πλήθος Overlap για πιο αξιόπιστα αποτελέσματα.

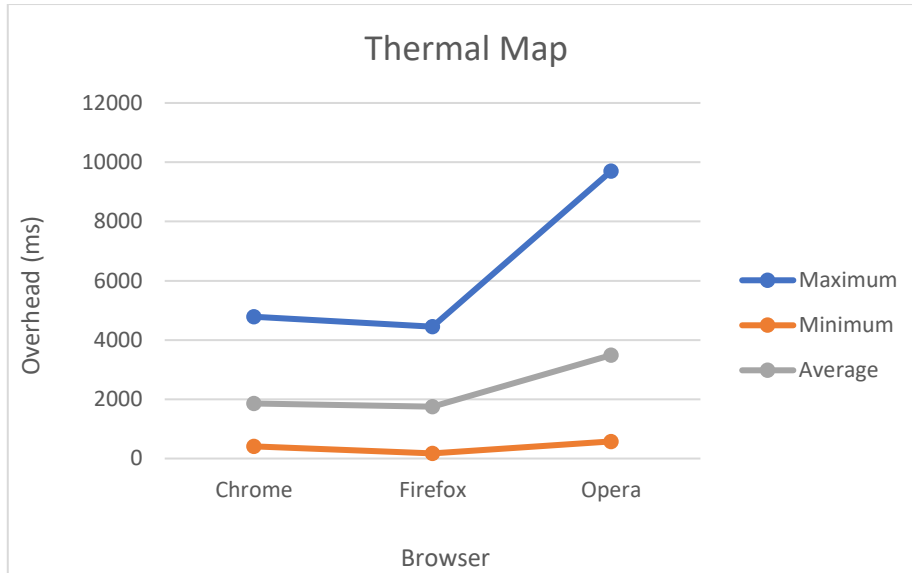
Στην εύρεση Overlap ο Firefox είναι ο πιο γρήγορος καθώς ο μέσος όρος για τα 18 κυκλώματα ibm ήταν ο χαμηλότερος με διαφορά περίπου 200 millisecond από τους άλλους browser. Το ελάχιστο overhead ήταν στα 144 millisecond στο ibm01 από τον Firefox ενώ το υψηλότερο ήταν στα 2750 στο ibm18 από τον Opera. Οι διαφορές του Chrome με τον Opera ήταν μικρές ανεξαρτήτου κυκλώματος.



Διάγραμμα 7: Μέσος χρόνος overhead Overflow όλων των κυκλωμάτων σε διαφορετικούς browser.

Στο Διάγραμμα 7 φαίνεται το overhead του Overflow στο Placement για όλα τα κυκλώματα συναρτήσει των browser. Τα κυκλώματα περιέχουν όλα ίδιο πλήθος Overflow για πιο αξιόπιστα αποτελέσματα.

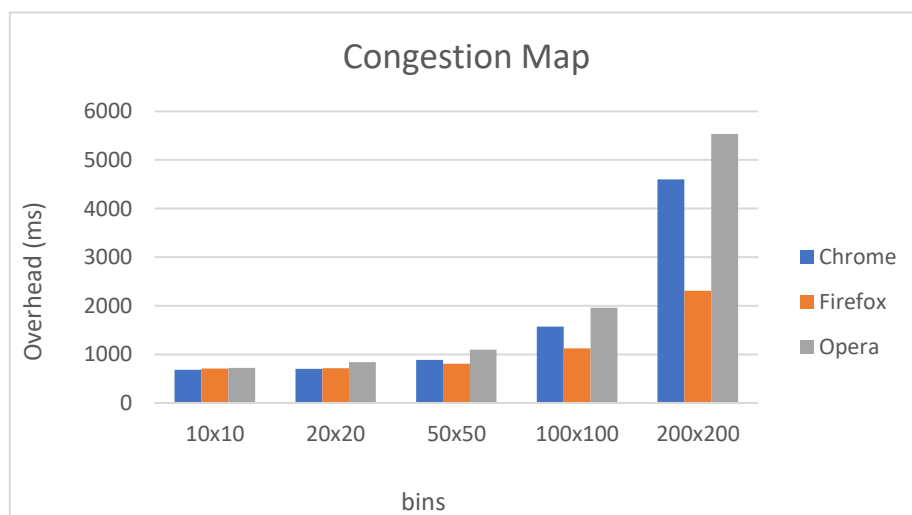
Στην εύρεση Overflow ο Firefox είναι ο πιο γρήγορος καθώς ο μέσος όρος για τα 18 κυκλώματα ibm ήταν ο χαμηλότερος με διαφορά περίπου 200-300 millisecond από τους άλλους browser. Το ελάχιστο overhead ήταν στα 155 millisecond στο ibm01 από τον Firefox ενώ το υψηλότερο ήταν στα 2815 στο ibm18 από τον Opera. Οι διαφορές του Chrome με τον Opera ήταν στο εύρος 100-200 millisecond ανεξαρτήτου κυκλώματος.



Διάγραμμα 8: Μέσος χρόνος overhead Thermal Map όλων των κυκλωμάτων σε διαφορετικούς browser.

Στο Διάγραμμα 8 φαίνεται το overhead του Thermal Map στο Placement για όλα τα κυκλώματα συναρτήσε των browser. Οι θερμοκρασίες του Thermal Map βασίζονται στα δεδομένα του αρχείου wts.

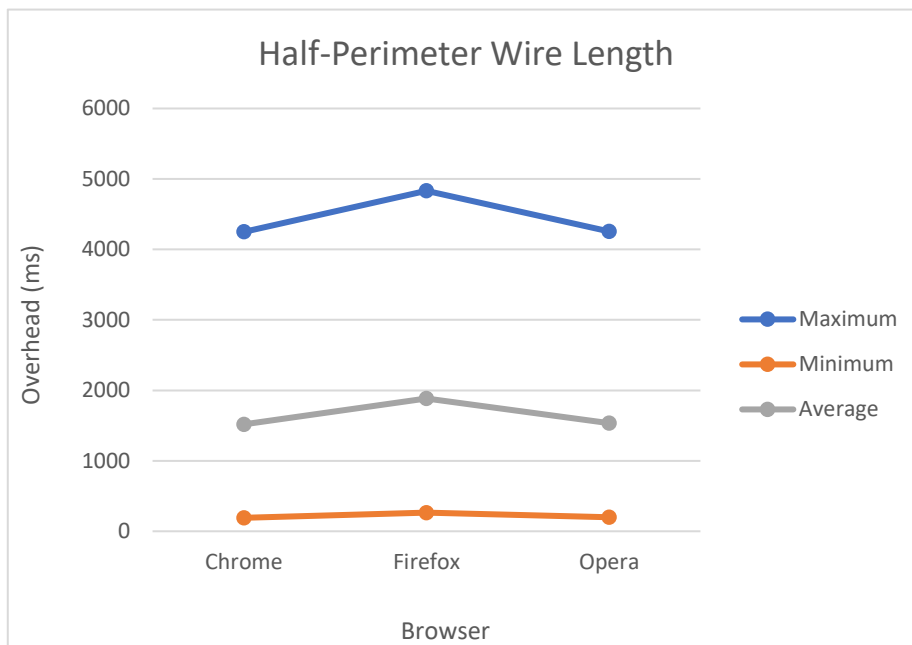
Στην θερμική χαρτογράφηση ο Firefox είναι ο πιο γρήγορος καθώς ο μέσος όρος για τα 18 κυκλώματα ibm ήταν ο χαμηλότερος με διαφορά περίπου 100 millisecond από τον Chrome και 1700 millisecond από τον Opera. Το ελάχιστο overhead ήταν στα 173 millisecond στο ibm01 από τον Firefox ενώ το υψηλότερο ήταν στα 9700 στο ibm18 από τον Opera. Οι διαφορές του Chrome με τον Firefox ήταν οριακές ενώ ο Opera είχε πολύ μεγάλη διαφορά από τους άλλους 2 ανεξαρτήτου κυκλώματος.



Διάγραμμα 9: Μέσος χρόνος overhead Congestion Map όλων των κυκλωμάτων σε διαφορετικούς browser.

Στο Διάγραμμα 9 φαίνεται το overhead του Congestion Map στο Placement για όλα τα κυκλώματα συναρτήσει διαφορετικών αριθμών bin για κάθε browser. Οι τιμές είναι οι μέσοι όροι για κάθε group bin αντίστοιχα.

Κατά την χαρτογράφηση πυκνότητας στα 10x10 bins ο Chrome είναι ο πιο γρήγορος με αμελητέα διαφορά από τους άλλους 2. Ανάλογα αποτελέσματα και στα 20x20 bins με μια διαφορά περίπου 130 millisecond του Opera από τους άλλους 2 browser που τον κάνει τον πιο αργό. Στα 50x50 bins ο Firefox γίνεται ο πιο γρήγορος browser αφήνοντας τον Opera 300 millisecond πίσω. Τέλος στα 100x100 και 200x200 bins ο Firefox ανοίγει την διαφορά και είναι κατά 400-800 millisecond πιο γρήγορος από τον Chrome και Opera αντίστοιχα. Στα 100x100 και στα 200x200 η διαφορά βρίσκεται στα 2300 και 2700 millisecond αντίστοιχα. Το ελάχιστο overhead ήταν στα 95 millisecond στα 10x10 bins από τον Chrome ενώ το υψηλότερο ήταν στα 15700 στα 200x200 bins από τον Opera.

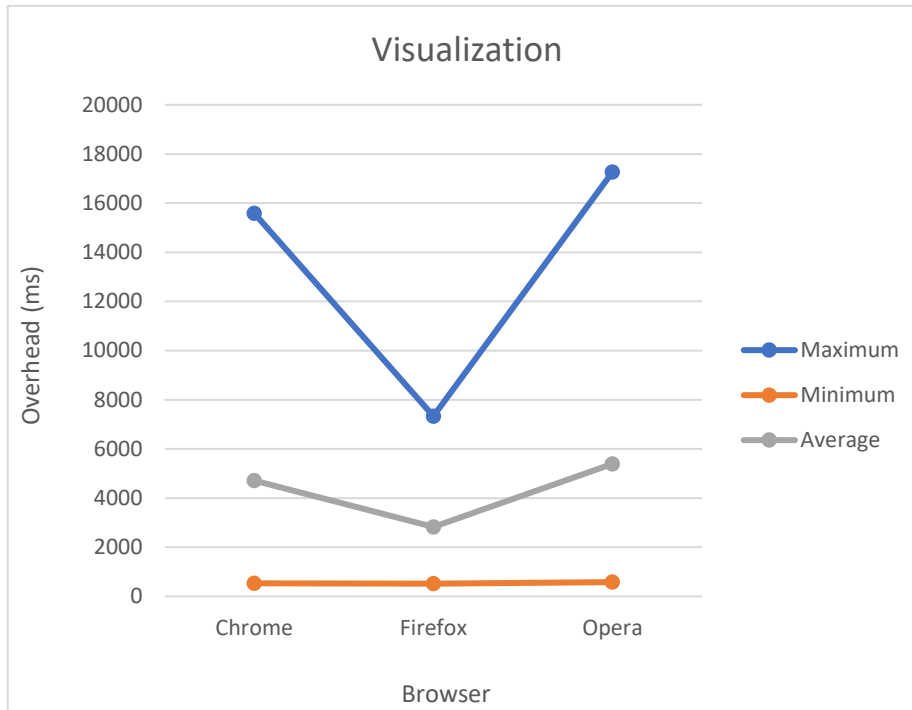


Διάγραμμα 10: Μέσος χρόνος overhead Half-Perimeter Wire Length όλων των κυκλωμάτων σε διαφορετικούς browser.

Στο Διάγραμμα 10 φαίνεται το overhead του Half-Perimeter Wire Length στο Placement για όλα τα κυκλώματα συναρτήσει των browser.

Στην καταγραφή του Half-Perimeter Wire Length ο Chrome είναι ο πιο γρήγορος καθώς ο μέσος όρος για τα 18 κυκλώματα ibm ήταν ο χαμηλότερος με διαφορά περίπου 15 millisecond από τον Opera και 300 millisecond από τον Firefox. Το ελάχιστο overhead

ήταν στα 191 millisecond στο ibm01 από τον Chrome ενώ το υψηλότερο ήταν στα 4831 στο ibm18 από τον Firefox. Οι διαφορές του Chrome με τον Opera ήταν οριακές ενώ ο Firefox είχε ορατή διαφορά από τους άλλους 2 περίπου 300 millisecond ανεξαρτήτου κυκλώματος.



Διάγραμμα 11: Μέσος χρόνος overhead Visualization όλων των κυκλωμάτων σε διαφορετικούς browser.

Στο Διάγραμμα 11 φαίνεται το overhead του γενικού Visualization για όλα τα κυκλώματα συναρτήσει των browser.

Ο Firefox είναι ο πιο γρήγορος καθώς ο μέσος όρος για τα 18 κυκλώματα ibm ήταν ο χαμηλότερος με μεγάλη διαφορά περίπου 2000-2400 millisecond από τους άλλους browser. Το ελάχιστο overhead ήταν στα 518 millisecond στο ibm01 από τον Firefox ενώ το υψηλότερο ήταν στα 17264 στο ibm18 από τον Opera. Οι διαφορές του Chrome με τον Opera ήταν μικρές ανεξαρτήτου κυκλώματος.

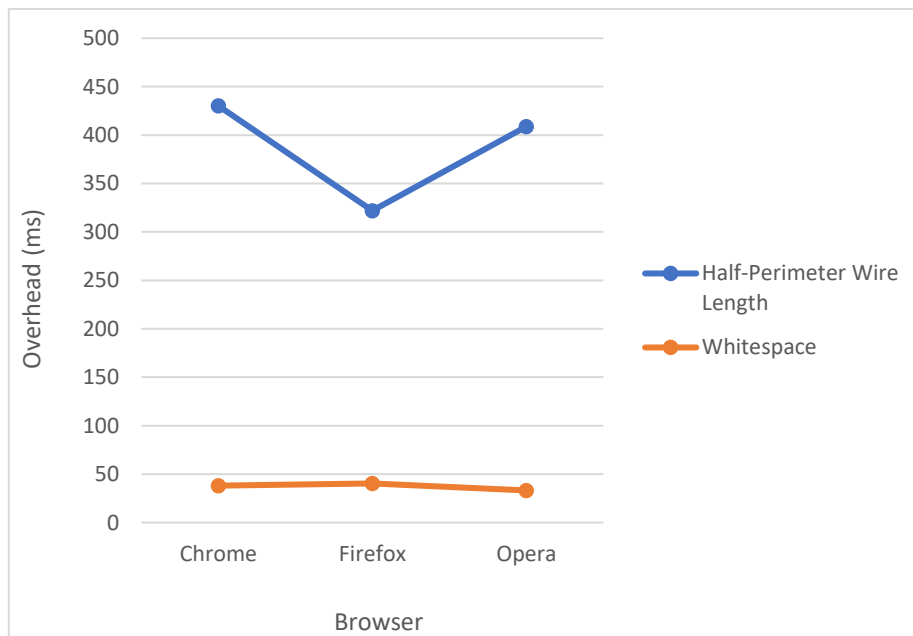
Πειράματα Floorplan

Στο Διάγραμμα 12 φαίνεται το overhead όπως καταγράφηκε από 2 βασικές λειτουργίες του Floorplan Analysis & Visualization συναρτήσει των browser. Οι λειτουργίες Half-Perimeter Wire Length και Whitespace είναι από τις πιο σημαντικές διότι δείχνουν αν

υπάρχει αχρειαστο καλώδιο ή αχρησιμοποίητος κενός χώρος. Στο πείραμα αυτό χρησιμοποιήθηκε το κύκλωμα GSRC SOFT n300.

Το αποτέλεσμα δείχνει ότι στην καταγραφή του Half-Perimeter Wire Length ο Firefox είναι ο πιο γρήγορος ενώ ο Chrome είναι ο πιο αργός με διαφορά περίπου 100 millisecond από τον Firefox, τέλος ο Opera βρίσκεται ενδιάμεσα από τους 2 με μικρή απόκλιση από τον Chrome. Φυσικά όλες οι μετρήσεις είναι μέσα στα πλαίσια αποδεκτών χρόνων. Αντίθετα από το Placement όπου ο Opera ήταν ο πιο γρήγορος σε αυτή την περίπτωση ο Firefox είναι ταχύτερος και κάτι τέτοιο σημαίνει ότι σε μικρότερα datasets ο τελευταίος είναι πιο αποδοτικός.

Αν και είναι το μεγαλύτερο κύκλωμα, με 300 στοιχεία, το overhead για το Whitespace σε όλους τους browser είναι ελάχιστο για την ανθρώπινη αντίληψη και περίπου στα 35 millisecond ενώ οι μεταξύ τους διαφορές είναι στο εύρος 2-5 millisecond.

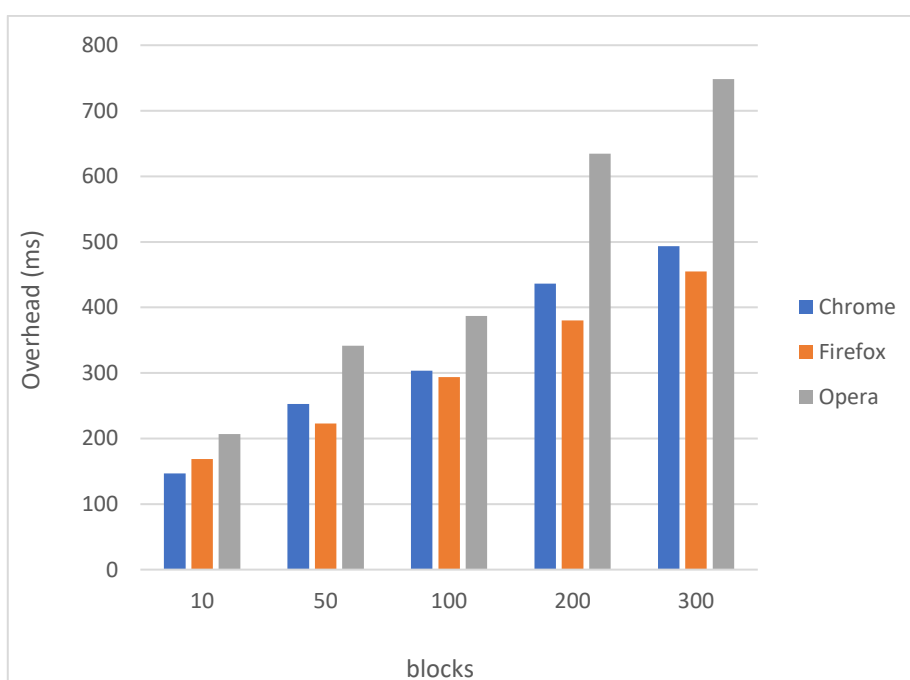


Διάγραμμα 12: Μέσος χρόνος overhead Half-Perimeter Wire Length & Whitespace συναρτήσει διαφορετικών browser.

Στο Διάγραμμα 13 φαίνεται το overhead για διαφορετικά πλήθη block όπως καταγράφηκε από το Export to PNG του Floorplan visualizer συναρτήσει των browser. Κατά την εξαγωγή ως εικόνα PNG, όλα τα blocks γίνονται render και ενσωματώνονται σε 1 εικόνα. Στο πείραμα αυτό χρησιμοποιήθηκαν τα κυκλώματα GSRC HARD n10, GSRC HARD n50, GSRC HARD n100, GSRC HARD n200 και GSRC HARD n300.

Το αποτέλεσμα δείχνει ότι στην καταγραφή Export to PNG ο Chrome και ο Firefox απέχουν λίγα millisecond μεταξύ τους σε οποιοδήποτε πλήθος block με την διαφορά να βρίσκεται στο εύρος των 9 με 56 millisecond. Από την άλλη μεριά ο Opera είναι ο πιο αργός και έχει αισθητή διαφορά στην καταγραφή του overhead με την διαφορά να φτάνει περίπου 250 millisecond από τους άλλους 2.

Από τα 10 blocks υπάρχουν αισθητές διαφορές στους χρόνους με τον Opera να είναι περίπου 38 millisecond πιο αργός. Όταν πενταπλασιάζονται, στα 50 blocks, οι χρόνοι του Chrome με τον Firefox είναι περίπου ίδιοι ενώ ο Opera μειώνει την απόδοσή του περίπου κατά 90 millisecond. Στα 100 blocks παρατηρείται το ίδιο μοτίβο και αντίστοιχα αποτελέσματα με τα 50 blocks. Ωστόσο, από τα 200 blocks και πάνω ο Opera έχει γίνει αρκετά αργός συγκριτικά με τους άλλους 2 browser οι οποίοι βρίσκονται στα ίδια περίπου νούμερα, με τον Firefox όμως να είναι ο πιο ταχύς.



Διάγραμμα 13: Μέσος χρόνος overhead Export to PNG σε διαφορετικούς browser συναρτήσει διαφορετικών αριθμών block.

Συμπέρασμα

Με βάση τα παραπάνω πειράματα overhead μπορεί κανείς να βγάλει το γενικό συμπέρασμα ότι τα χαρακτηριστικά του μηχανήματος έχουν τη μεγαλύτερη βαρύτητα στο πόσο γρήγορα θα εκτελεστούν οι ενέργειες του WEVIAN. Όμως ένα μικρό μέρος των αποτελεσμάτων αναλογεί και στο πόσο ελαφρύς είναι ένας browser ή πόσο καλή

διαχείριση πόρων μπορεί να κάνει. Όλα τα αποτελέσματα βρίσκονται σε λογικά και αποδεκτά χρονικά πλαίσια και αυτό κάνει το WEVIAN ένα αξιόπιστο, γρήγορο και αποτελεσματικό ερευνητικό εργαλείο.

ΚΕΦΑΛΑΙΟ 4

Συμπεράσματα και Μελλοντικές Επεκτάσεις

4.1 Συμπεράσματα

Είναι πλέον ευρέως αποδεκτό ότι η εποχή αυτή χαρακτηρίζεται από τη ραγδαία εξέλιξη στο χώρο των ηλεκτρονικών. Κάθε ηλεκτρονική συσκευή που χρησιμοποιείται φέρει μέσα της ένα κύκλωμα, κάποιες φορές είναι μεγάλο και περιέχει συμπυκνωμένα δισεκατομμύρια τρανζίστορ που δεν διακρίνονται με γυμνό μάτι και άλλες φορές είναι μικρό και ευδιάκριτο. Όμως, όλα αυτά τα κυκλώματα έχουν κοινά θεμέλια στον πολύπλοκο τρόπο παραγωγής τους.

Η ραγδαία αυτή εξέλιξη της τεχνολογίας μαζί με την εφεύρεση του διαδικτύου δημιούργησε περισσότερες ανάγκες. Η ανάπτυξη των υπολογιστών δεν θα μπορούσε να αφήσει ανεπηρέαστα τα λογισμικά που εκτελούνται. Δημιουργήθηκαν διάφορα είδη λογισμικών, από αυτά, κάποια απαιτούν διαδίκτυο και άλλα όχι. Οι διαδικτυακές εφαρμογές αποτελούν το μεγαλύτερο μέρος των εφαρμογών που υπάρχουν σήμερα, ωστόσο και αυτές εξελίσσονται συνεχώς. Για να γίνει αυτό, δημιουργούνται νέες γλώσσες προγραμματισμού ή ενημερώνονται οι παλιές με νέες εκδόσεις. Με αυτό τον τρόπο διευκολύνεται η δημιουργία νέων εφαρμογών και η συντήρηση των παλιών.

Η διαδικτυακή εφαρμογή WEVIAN χτίστηκε με τις νεότερες τεχνολογίες και αποτελεί ένα ερευνητικό εργαλείο το οποίο είναι γρήγορο, αξιόπιστο και αποτελεσματικό. Μέσα από την παρουσίαση των λειτουργιών και των πειραμάτων η εφαρμογή αποδείχθηκε πλήρως λειτουργική και χρήσιμη. Συγκριτικά με τις πρώτες εκδόσεις του εργαλείου παρατηρήθηκε δραματική αλλαγή στους χρόνους επιβάρυνσης των λειτουργιών καθώς και στο πλήθος των λειτουργιών.

4.2 Επεκτασιμότητα

Η διαδικτυακή εφαρμογή WEVIAN έχει ξεκάθαρες προοπτικές επέκτασης με νέες λειτουργικότητες. Καθώς οι αλγόριθμοι τοποθέτησης και legalization είναι επαναληπτικοί θα ήταν βοηθητικό να φαίνεται η προσομοίωση σε κάθε βήμα της επανάληψης έως ότου

αυτή ολοκληρωθεί. Ακόμη, περισσότερα διαγράμματα και στατιστικά, όπως η κατανάλωση μνήμης κατά την προσομοίωση, θα ήταν πολύ βοηθητικά και εύκολα να υλοποιηθούν. Επίσης, θα ήταν θετική η ύπαρξη αλγορίθμων για την δημιουργία αρχείων .pl και μετέπειτα προσομοίωσή τους. Τέλος, εκτός από το bookshelf, αποδοχή περισσότερων placement format θα μπορούσε να ενσωματώσει το εργαλείο σε περισσότερες αυτοματοποιημένες ροές σχεδιασμού.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Integrated circuit. (2019, 30 March). In Wikipedia. Ανακτήθηκε 30 Μαρτίου 2019, από https://en.wikipedia.org/wiki/Integrated_circuit
- [2] Kilby, J.S. (1976). Invention of the integrated circuit. IEEE Transactions on electron devices, 23(7), 648-654.
- [3] Ram, B. (2000). Computer Fundamentals: Architecture and Organization. Delhi: New Age International.
- [4] Circuit design. (2019, 2 April). In Wikipedia. Ανακτήθηκε 2 Απριλίου 2019, από https://en.wikipedia.org/wiki/Circuit_design
- [5] Integrated circuit design. (2019, 2 April). In Wikipedia. Ανακτήθηκε 2 Απριλίου 2019, από https://en.wikipedia.org/wiki/Integrated_circuit_design
- [6] George Boole. (2019, 23 July). In Wikipedia. Ανακτήθηκε 23 Ιουλίου 2019, από https://en.wikipedia.org/wiki/George_Boole
- [7] Godse, D.A. & Godse, A.P. (2008). Logic Design. Pune: Technical Publications.
- [8] Truth table. (2019, 5 May). In Wikipedia. Ανακτήθηκε 5 Μαΐου 2019, από https://en.wikipedia.org/wiki/Truth_table
- [9] Sherwani, N. (2002). Algorithms for VLSI physical design automation. New York: Kluwer Academic Publishers
- [10] Βουρόπουλος, Η., Γιολδάσης, Ν. & Σαράφης, Ι. (χ.η.). Επεξεργαστής Η κατασκευή του. Ανακτήθηκε από <http://www.it.uom.gr/project/mycomputer/cpu/info/construc.html>
- [11] Sarcar, M.M.M., Rao K.M. & Narayan, K.L. (2008). Computer Aided Design and Manufacturing. New Delhi: PHI Learning Pvt. Ltd.
- [12] Tim Berners-Lee. (2019, 19 May). In Wikipedia. Ανακτήθηκε 19 Μαΐου 2019, από https://en.wikipedia.org/wiki/Tim_Berners-Lee
- [13] History of the World Wide Web. (2019, 19 May). In Wikipedia. Ανακτήθηκε 19 Μαΐου 2019, από https://en.wikipedia.org/wiki/History_of_the_World_Wide_Web
- [14] A short history of the Web. (2019, 19 May). Ανακτήθηκε 19 Μαΐου 2019, από <https://home.cern/science/computing/birth-web/short-history-web>
- [15] Berners-Lee, T. (1989). Information Management: A Proposal. Ανακτήθηκε από <http://cds.cern.ch/record/369245/files/dd-89-001.pdf>

- [16] McCracken, H. (2014, 12 March). The Web at 25: Revisiting Tim Berners-Lee's Amazing Proposal. Ανακτήθηκε από <http://time.com/21039/tim-berners-lee-web-proposal-at-25/>
- [17] Paul Kunz. (2019, 19 May). In Wikipedia. Ανακτήθηκε 19 Μαΐου 2019, από https://en.wikipedia.org/wiki/Paul_Kunz
- [18] Nations, D. (2019, 17 March). What Exactly Is a Web Application?. Ανακτήθηκε από <https://www.lifewire.com/what-is-a-web-application-3486637>
- [19] Carti, E. (2019, 19 May). What is the difference between a web app and a website?. Ανακτήθηκε από <https://qr.ae/TWNWsX>
- [20] Web application. (2019, 19 May). In Wikipedia. Ανακτήθηκε 19 Μαΐου 2019, από https://en.wikipedia.org/wiki/Web_application
- [21] Τεχνητή νοημοσύνη. (2019, 19 May). In Wikipedia. Ανακτήθηκε 19 Μαΐου 2019, από https://el.wikipedia.org/wiki/Τεχνητή_νοημοσύνη
- [22] Tsapatsoulis, N. (2011). ΕΣΔ516: Τεχνολογίες Διαδικτύου. Ανακτήθηκε από <http://cis.cut.ac.cy/~nicolas.tsapatsoulis/cis516/lectures/IT07.pdf>
- [23] Johnston, J. (2019). How to build a web app: A beginner's guide (2019). Ανακτήθηκε από <https://www.budibase.com/blog/how-to-make-a-web-app/>
- [24] Database. (2019, 22 May). In Wikipedia. Ανακτήθηκε 22 Μαΐου 2019, από <https://en.wikipedia.org/wiki/Database>
- [25] WEVIAN. (χ.η.). Ανακτήθηκε 17 Αυγούστου 2019, από <https://wevian.xyz>
- [26] Rinehart, M. (2019, 1 July). HTML Tag History. Ανακτήθηκε από <http://www.martinrinehart.com/frontend-engineering/engineers/html/html-tag-history.html>
- [27] Index of the HTML 4 Elements. (2019, 1 July). Ανακτήθηκε 1 Ιουλίου 2019 από <https://www.w3.org/TR/1999/REC-html401-19991224/index/elements>
- [28] HTML.com: Study HTML and Learn to Code With Our Step-By-Step Guide. (χ.η.). Ανακτήθηκε από <https://html.com/>
- [29] HTML head tag. (2019, 1 July). Ανακτήθηκε 1 Ιουλίου 2019 από https://www.w3schools.com/tags/tag_head.asp
- [30] HTML meta tag. (2019, 1 July). Ανακτήθηκε 1 Ιουλίου 2019 από https://www.w3schools.com/tags/tag_meta.asp
- [31] Cascading Style Sheets. (2019, 1 July). In Wikipedia. Ανακτήθηκε 1 Ιουλίου 2019, από https://en.wikipedia.org/wiki/Cascading_Style_Sheets

- [32] Brendan Eich. (2019, 1 July). In Wikipedia. Ανακτήθηκε 1 Ιουλίου 2019, από https://en.wikipedia.org/wiki/Brendan_Eich
- [33] Pollock, J. (2010). JavaScript a Beginner's Guide. Toronto: McGraw Hill.
- [34] PHP. (2019, 1 July). In Wikipedia. Ανακτήθηκε 1 Ιουλίου 2019, από <https://en.wikipedia.org/wiki/PHP>
- [35] Usage statistics of PHP for websites. (2019, 1 July). Ανακτήθηκε 1 Ιουλίου 2019 από <https://w3techs.com/technologies/details/pl-php/all/all>
- [36] JSON to exchange objects between different languages. (2019, 1 July). Ανακτήθηκε 1 Ιουλίου 2019 από <https://www.scriptol.com/programming/json.php>
- [37] SQL. (2019, 1 July). In Wikipedia. Ανακτήθηκε 1 Ιουλίου 2019, από <https://en.wikipedia.org/wiki/SQL>
- [38] Usage statistics and market share of jQuery for websites. (2019, 1 July). Ανακτήθηκε 1 Ιουλίου 2019 από <https://w3techs.com/technologies/details/js-jquery/all/all>
- [39] Mantos, K., Koungalis, N., Demesiotis, G., Oikonomou, P., Dadaliaris, A.N., & Stamoulis, G.I. (2017). WEVIAN: A Web-based Placement Visualizer / Analyzer. PCI.
- [40] N. Viswanathan & C. N. Chu. FastPlace: efficient analytical placement using cell shifting, iterative local refinement, and a hybrid net model. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 24(5), 722-733, 2005. DOI: <http://dx.doi.org/10.1145/981066.981072>
- [41] P. Oikonomou, M. G. Koziri, A. N. Dadaliaris, T. Loukopoulos & I. G. Stamoulis. (2017). Domocus: Lock free parallel legalization in standard cell placement. In Modern Circuits and Systems Technologies (MOCASST), 2017 6th International Conference on (pp. 1-4). IEEE. DOI: <http://dx.doi.org/10.1109/MOCASST.2017.7937644>
- [42] A. E. Caldwell, A. B. Kahng & I. L. Markov. VLSI CAD Bookshelf. Ανακτήθηκε 27 Ιουλίου 2019 από <http://vlsicad.eecs.umich.edu/BK>