



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΛΥΣΗ ΠΡΟΤΥΠΩΝ ΚΥΚΛΩΜΑΤΙΚΩΝ
ΔΙΑΤΑΞΕΩΝ ΧΩΡΟΘΕΤΗΣΗΣ ΜΕ ΧΡΗΣΗ
ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ PANDAS

ΒΙΓΛΑΣ ΠΑΝΑΓΙΩΤΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΔΑΔΑΛΙΑΡΗΣ ΑΝΤΩΝΙΟΣ

ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

ΣΥΝΕΠΙΒΛΕΠΟΥΣΑ

ΚΟΖΥΡΗ ΜΑΡΙΑ

ΕΠΙΚΟΥΡΗ ΚΑΘΗΓΗΤΡΙΑ

Λαμία, Μάρτιος 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΛΥΣΗ ΠΡΟΤΥΠΩΝ ΚΥΚΛΩΜΑΤΙΚΩΝ
ΔΙΑΤΑΞΕΩΝ ΧΩΡΟΘΕΤΗΣΗΣ ΜΕ ΧΡΗΣΗ
ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ PANDAS

ΒΙΓΛΑΣ ΠΑΝΑΓΙΩΤΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΔΑΔΑΛΙΑΡΗΣ ΑΝΤΩΝΙΟΣ

ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

ΣΥΝΕΠΙΒΛΕΠΟΥΣΑ

ΚΟΖΥΡΗ ΜΑΡΙΑ

ΕΠΙΚΟΥΡΗ ΚΑΘΗΓΗΤΡΙΑ

Λαμία, Μάρτιος 2022



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

DATA ANALYSIS OF STANDARDIZED
PLACEMENT CIRCUITS USING THE PANDAS
LIBRARY

VIGLAS PANAGIOTIS

FINAL THESIS

ADVISOR

DADALIARIS ADONIOS

PROFESSOR

CO ADVISOR

KOZIRI MARIA

PROFESSOR

Lamia, March 2022

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της 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, 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.

Ημερομηνία: 03/03/2022

Ο Δηλών

Βίγλας Παναγιώτης

(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

Αισθάνομαι την ανάγκη να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Δαδαλιάρη Αντώνιο, καθώς και την συνεπιβλέπουσα κυρία Κοζύρη Μαρία, για την καθοδήγηση, τις πολύτιμες συμβουλές και υποδείξεις σε όλη τη διάρκεια εκπόνησης της πτυχιακής μου εργασίας.

Θα ήθελα επίσης να ευχαριστήσω την οικογένεια μου, για τη στήριξη τους όλα αυτά τα χρόνια.

ΠΕΡΙΛΗΨΗ

Σκοπό της παρούσας εργασίας αποτελεί η ανάδειξη της βιβλιοθήκης `pandas`, η οποία ανήκει στη γλώσσα προγραμματισμού `Python`, ως προς την ευκολία χρήσης αλλά και απόδοσή της, στην ανάλυση μεγάλου όγκου δεδομένων. Πιο συγκεκριμένα, το υλικό, που βρίσκεται στη διάθεσή μας, περιγράφει ολοκληρωμένα κυκλώματα, τα οποία δίνονται σε `bookshelf` μορφή. Για την ανάλυση τους, υλοποιήθηκαν οι αρμόζουσες συναρτήσεις κάνοντας χρήση των βασικών εργαλείων της γλώσσας `Python` αλλά και της `pandas`. Ωστόσο η πλειοψηφία αυτών, πραγματοποιήθηκε με τη συνδρομή των `DataFrames`, που προσφέρει η προαναφερθείσα βιβλιοθήκη. Όσον αφορά τη σειρά των διαδικασιών, που ακολουθήθηκαν, πρώτα δημιουργήθηκε ένας `parser` με τη χρήση της `Python` και στην συνέχεια, τα αντλούμενα δεδομένα μεταφέρθηκαν σε `DataFrames`. Πάνω σε αυτά υλοποιήθηκαν διάφορες συναρτήσεις, που αφορούν τόσο την διαχείριση και μεταποίηση, όσο και την απεικόνιση των δεδομένων. Τέλος, συμπεριλαμβάνονται και συναρτήσεις που έχουν πραγματωθεί και με τους δύο τρόπους, με σκοπό να συγκριθεί ο χρόνος εκτέλεσης σε κάθε μία από τις περιπτώσεις.

Λέξεις κλειδιά: `Python`, `pandas`, `matplotlib`, βιβλιοθήκη, ανάλυση δεδομένων, ολοκληρωμένο κύκλωμα, `bookshelf`

ABSTRACT

The purpose of this thesis, is to highlight one of Python's libraries, which is called pandas, by showing how easily we can write code with it and how good performance it has, while analyzing big amount of data. More specifically, the dataset that we analyzed, contains information of Integrated Circuits and has been given to us in bookshelf format. In order to extract the results, we implemented the proper functions, with the basic tools of Python, but also with the use of DataFrames, a data structure of pandas library. As far as the sequence of the implementation, first of all, we created a parser function, with the classic method, and later on the parsed data were converted to DataFrames. With the use of them, we created even more functions, for data modification, manipulation and visualization. In the end, we present the functions that are implemented both ways, in order to compare the execution times, in each case.

Keywords: Python, pandas, matplotlib, library, Data Analysis, circuit, bookshelf format

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	III
<u>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ.....</u>	1
1.1 ΧΡΗΣΙΜΟΤΗΤΑ ΑΝΑΛΥΣΗΣ ΔΕΔΟΜΕΝΩΝ	1
1.2 ΔΟΜΗ ΚΕΦΑΛΑΙΩΝ.....	2
<u>ΚΕΦΑΛΑΙΟ 2 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΓΥΡΩ ΑΠΟ ΤΟ ΚΥΚΛΩΜΑ.....</u>	3
2.1 ΨΗΦΙΑΚΕΣ ΛΟΓΙΚΕΣ ΠΥΛΕΣ	3
2.2 ΟΛΟΚΛΗΡΩΜΕΝΟ ΚΥΚΛΩΜΑ	4
2.3 ΦΥΣΙΚΗ ΣΧΕΔΙΑΣΗ	7
2.4 ΧΩΡΟΘΕΤΗΣΗ.....	8
2.5 ΜΟΝΤΕΛΟΠΟΙΗΣΗ	9
2.6 BOOKSHELF ΜΟΡΦΗ	11
<u>ΚΕΦΑΛΑΙΟ 3 ΡΥΘΜΟΝ ΚΑΙ ΒΙΒΛΙΟΘΗΚΗ PANDAS</u>	18
3.1 ΡΥΘΜΟΝ.....	18
3.1.1 ΠΡΟΕΛΕΥΣΗ.....	18
3.1.2 ΟΡΙΣΜΟΣ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	18
3.1.3 ΔΗΜΟΦΙΛΕΣΤΕΡΟΙ ΤΟΜΕΙΣ ΧΡΗΣΗΣ ΡΥΘΜΟΝ	20
3.2 ΒΙΒΛΙΟΘΗΚΗ PANDAS	21
3.2.1 ΠΡΟΕΛΕΥΣΗ.....	21
3.2.2 ΟΡΙΣΜΟΣ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	21
3.2.3 DATAFRAMES	22
3.2.4 ΘΕΤΙΚΑ ΤΗΣ PANDAS	23
3.2.5 ΑΡΝΗΤΙΚΑ ΤΗΣ PANDAS.....	25
3.3 ΒΙΒΛΙΟΘΗΚΗ MATPLOTLIB	25
<u>ΚΕΦΑΛΑΙΟ 4 ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΥΛΟΠΟΙΗΜΕΝΟΥ ΚΩΔΙΚΑ</u>	26
4.1 ΣΥΝΑΡΤΗΣΗ ΕΠΑΛΗΘΕΥΣΗΣ ΑΡΧΕΙΩΝ.....	26
4.2 ΣΥΝΑΡΤΗΣΗ PARSER	27
4.3 ΔΗΜΙΟΥΡΓΙΑ DATAFRAMES.....	29
4.3.1 ΠΕΡΙΓΡΑΦΗ ΤΡΟΠΟΥ ΔΗΜΙΟΥΡΓΙΑΣ ΤΩΝ DATAFRAME.....	29
4.3.2 DATAFRAME ΤΩΝ ΚΟΜΒΩΝ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ	29
4.3.3 DATAFRAME ΤΩΝ NET ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ.....	31
4.3.4 DATAFRAME ΤΩΝ ΣΕΙΡΩΝ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ	32

4.3.5 DATAFRAME ΟΛΟΚΛΗΡΟΥ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ	34
4.3.6 ΑΠΟΘΗΚΕΥΣΗ ΤΩΝ DATAFRAME ΣΕ ΑΡΧΕΙΟ	34
4.4 ΥΛΟΠΟΙΗΜΕΝΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΩΝ DATAFRAME	36
4.4.1 ΕΙΣΑΓΩΓΗ	36
4.4.2 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΟΥΣ ΚΟΜΒΟΥΣ.....	37
4.4.3 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΑ NET	37
4.4.4 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΙΣ ΣΕΙΡΕΣ.....	38
4.4.5 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΟ ΚΥΚΛΩΜΑ.....	39
4.4.6 ΓΡΑΦΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΜΕ ΧΡΗΣΗ ΤΗΣ MATPLOTLIB.....	40
4.5 ΧΡΟΝΟΙ ΕΚΤΕΛΕΣΗΣ	43
4.5.1 ΓΡΑΦΗΜΑΤΑ ΣΥΓΚΡΙΣΗΣ ΚΟΙΝΩΝ ΣΥΝΑΡΤΗΣΕΩΝ.....	44
4.5.2 ΧΡΟΝΟΙ ΕΚΤΕΛΕΣΗΣ ΟΛΩΝ ΤΩΝ ΣΥΝΑΡΤΗΣΕΩΝ.....	50
4.6 ΨΕΥΔΟΚΩΔΙΚΕΣ	51
4.6.1 ΕΙΣΑΓΩΓΗ	51
4.6.2 ΑΛΓΟΡΙΘΜΟΣ TETRIS-LIKE LEGALIZATION.....	51
4.6.3 ΑΛΓΟΡΙΘΜΟΣ DETAILED PLACEMENT	52
<u>ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ.....</u>	<u>54</u>
<u>ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ</u>	<u>56</u>

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικόνα 1: Γραφική απεικόνιση τοπι	5
Εικόνα 2: Εσωτερικό Ολοκληρωμένου Κυκλώματος.....	6
Εικόνα 3: Φυσική Μορφή ενός Ολοκληρωμένου Κυκλώματος, σειράς 7400	6
Εικόνα 4: Αναπαράσταση κυκλώματος.....	10
Εικόνα 5: Περιεχόμενα .aux αρχείου.....	11
Εικόνα 6: Περιεχόμενα .nodes αρχείου	12
Εικόνα 7: Γραφική Αναπαράσταση Συντεταγμένων	13
Εικόνα 8: Περιεχόμενα .pl αρχείου	13
Εικόνα 9: Αναπαράσταση Net σε μορφή γράφου	15
Εικόνα 10: Περιεχόμενα .nets αρχείου.....	15
Εικόνα 11: Περιεχόμενα .scl αρχείου	16
Εικόνα 12: Περιεχόμενα .wts αρχείου	17
Εικόνα 13: Σήμα της Python.....	18
Εικόνα 14: Σήμα της pandas.....	21
Εικόνα 15: Απεικόνιση ενός DataFrame.....	23
Εικόνα 16: Δημιουργία νέας στήλης	24
Εικόνα 17: Απεικόνιση της νέας στήλης.....	24
Εικόνα 18: Σήμα της matplotlib	25
Εικόνα 19: Μήνυμα επιτυχημένης επαλήθευσης	26
Εικόνα 20: Μήνυμα μη επιτυχημένης επαλήθευσης.....	26
Εικόνα 21: Δημιουργία DataFrame Κόμβων	29
Εικόνα 22: Απεικόνιση DataFrame Κόμβων	30
Εικόνα 23: Δημιουργία DataFrame Net.....	31
Εικόνα 24: Υπολογισμός καλωδίου & μεγέθους των Net	31
Εικόνα 25: Απεικόνιση DataFrame Net.....	32
Εικόνα 26: Δημιουργία DataFrame Σειρών.....	32
Εικόνα 27: Απεικόνιση DataFrame Σειρών (μέρος 1)	33
Εικόνα 28: Απεικόνιση DataFrame Σειρών (μέρος 2)	33
Εικόνα 29: Απεικόνιση DataFrame Κυκλώματος.....	34
Εικόνα 30: Αποθήκευση 4 DataFrame σε Excel αρχείο	34
Εικόνα 31: Απόσπασμα από το Nodes Excel αρχείο	35
Εικόνα 32: Απόσπασμα από το Nets Excel αρχείο	35
Εικόνα 33: Απόσπασμα από το Rows Excel αρχείο	36
Εικόνα 34: Απόσπασμα από το Design Excel αρχείο.....	36
Εικόνα 35: Αποτελέσματα των συναρτήσεων των κόμβων	37
Εικόνα 36: Αποτελέσματα των συναρτήσεων των Net.....	38
Εικόνα 37: Αποτελέσματα των συναρτήσεων των Σειρών.....	39
Εικόνα 38: Υπολογισμός καλωδίου Κυκλώματος	39
Εικόνα 39: Αριθμός κόμβων, με συγκεκριμένο μέγεθος	40
Εικόνα 40: Αριθμός Net, με συγκεκριμένο μέγεθος	41
Εικόνα 41: Αριθμός Net, με συγκεκριμένο πλήθος κόμβων.....	41

Εικόνα 42: Αριθμός σειρών, με συγκεκριμένο πλήθος κόμβων	42
Εικόνα 43: Αριθμός σειρών, με συγκεκριμένη πυκνότητα.....	42
Εικόνα 44: Απεικόνιση χωρητικότητας των Σειρών	43
Εικόνα 45: ibm01 Σύγκριση Χρόνων - Μέρος 1	44
Εικόνα 46: ibm01 Σύγκριση Χρόνων - Μέρος 2	45
Εικόνα 47: ibm01 Σύγκριση Χρόνων - Μέρος 3	45
Εικόνα 48: ibm05 Σύγκριση Χρόνων - Μέρος 1	46
Εικόνα 49: ibm05 Σύγκριση Χρόνων - Μέρος 2	46
Εικόνα 50: ibm05 Σύγκριση Χρόνων - Μέρος 3	47
Εικόνα 51: ibm18 Σύγκριση Χρόνων - Μέρος 1	47
Εικόνα 52: ibm18 Σύγκριση Χρόνων - Μέρος 2	48
Εικόνα 53: ibm18 Σύγκριση Χρόνων - Μέρος 3	48
Εικόνα 54: Χρόνοι Εκτέλεσης κοινών συναρτήσεων	49
Εικόνα 55: Χρόνοι εκτέλεσης συναρτήσεων - Μέρος 1.....	50
Εικόνα 56: Χρόνοι εκτέλεσης συναρτήσεων - Μέρος 2.....	50
Εικόνα 57: Χρόνοι εκτέλεσης συναρτήσεων - Μέρος 3.....	50
Εικόνα 58: Tetris-like Legalization ψευδοκώδικας.....	52
Εικόνα 59: Detailed Placement ψευδοκώδικας.....	53

ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ

Πίνακας 1: Πίνακες Αληθείας.....	4
----------------------------------	---

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 ΧΡΗΣΙΜΟΤΗΤΑ ΑΝΑΛΥΣΗΣ ΔΕΔΟΜΕΝΩΝ

Τα τελευταία χρόνια, σε παγκόσμιο επίπεδο, έχει πραγματοποιηθεί άνευ προηγουμένου, ραγδαία αύξηση του όγκου δεδομένων, που παράγονται, αναλύονται και αποθηκεύονται σε καθημερινή βάση, σε όλους τους τομείς. Κομβικό ρόλο σε αυτό, διαδραματίζει η πλέον εύκολη πρόσβαση, που έχει ο μέσος άνθρωπος στο διαδίκτυο, αλλά και το πλήθος των συνδεδεμένων συσκευών σε αυτό. Σε αντίθεση με το παρελθόν, όπου γινόταν μόνο χρήση υπολογιστών για τέτοιους σκοπούς, στη νέα εποχή υπάρχει πληθώρα συσκευών, που έχουν την δυνατότητα σύνδεσης στο διαδίκτυο, όπως είναι οι κινητές αλλά και όλων των ειδών οι έξυπνες συσκευές. Η εκθετική αυτή αύξηση των παραγόμενων δεδομένων, σε τόσο σύντομο χρονικό διάστημα, έχει καταστήσει αναγκαία την ψηφιοποίηση τους, σε όλον τον βιομηχανικό και μη κόσμο.

Η ανάλυση των δεδομένων αυτών, μπορεί να χρησιμοποιηθεί τόσο για απλές καθημερινές ενέργειες, όσο και στον επιχειρηματικό κόσμο. Όταν η χρήση αυτών αφορά τον δεύτερο, τότε η εύρεση νέων τρόπων για γρήγορη ανάλυση και εξαγωγή αποτελεσμάτων, είναι πρωταρχικός στόχος προκειμένου οι επιχειρήσεις να αυξήσουν τα κέρδη τους, να μείνουν ανταγωνιστικές στον χώρο, αλλά και να είναι προετοιμασμένες να προσαρμοστούν γρήγορα, στις μελλοντικές αλλαγές. Όλα αυτά επιτυγχάνονται από την αυτοματοποιημένη ανάλυση δεδομένων, καθώς με την χρήση των παραγόμενων αποτελεσμάτων τους, μπορούν να εξαχθούν νέες πληροφορίες, να βρεθούν νέες καινοτόμες ιδέες αλλά και να μειωθούν αισθητά οι χρόνοι εκτέλεσης μιας ενέργειας, που στο παρελθόν θα ήταν πιο χρονοβόρες και πιο αβέβαιες ως προς το αποτέλεσμα τους. Τρανό παράδειγμα της αξιοποίησης των εξαγόμενων αποτελεσμάτων από την ανάλυση δεδομένων, αποτελεί η χρήση αυτών από διαφημιστικούς φορείς, οι οποίοι σύμφωνα με τα δεδομένα, που αντλούν από τους χρήστες των μέσων

κοινωνικής δικτύωσης, προωθούν στοχευμένες διαφημίσεις στον καθένα από αυτούς, ανάλογα με τα ενδιαφέροντα τους, ώστε να αυξήσουν την ανταπόκρισή τους.

Η μορφή των δεδομένων αυτών μπορεί να είναι δομημένη αλλά και μη και συναρτήσει με τον όγκο τους, τα καθιστά αδύνατα να αναλυθούν χρησιμοποιώντας τις παραδοσιακές μεθόδους. Ως απόρροια αυτού, η εύρεση νέων αποτελεσματικών μέσων για την ψηφιακή ανάλυση αυτών, είναι ένας συνεχής αγώνας. Οι περισσότερες γλώσσες προγραμματισμού, εμπλουτίζουν συχνά το οικοσύστημα τους, με τέτοιου είδους εργαλεία και μόνιμος σκοπός τους είναι να τα καταστήσουν όλο και πιο γρήγορα και αποτελεσματικά. Ένα απ' τα εργαλεία αυτά, είναι και η βιβλιοθήκη της γλώσσας Python, που ονομάζεται pandas.

1.2 ΔΟΜΗ ΚΕΦΑΛΑΙΩΝ

Στο κεφάλαιο 2, παραθέτονται οι βασικές έννοιες σχετικά με το κύκλωμα, τις τεχνικές που εφαρμόζονται πάνω σε αυτό αλλά και η δομή των αρχείων, που το περιγράφουν.

Στο κεφάλαιο 3, γίνεται αναφορά στη γλώσσα προγραμματισμού Python και στα χαρακτηριστικά αυτής, ενώ εν συνεχεία γίνεται εισαγωγή σε κάποιες βιβλιοθήκες, που προσφέρει. Πιο συγκεκριμένα αναλύεται η pandas και πραγματοποιείται σύντομη αναφορά στην matplotlib.

Στο κεφάλαιο 4, περιγράφεται ο κώδικας, που πραγματώθηκε, παραθέτονται οι υλοποιημένες συναρτήσεις, τα αποτελέσματα και οι χρόνοι εκτέλεσης αυτών ενώ τέλος, παρουσιάζονται και δύο ψευδοκώδικες, που πραγματοποιούν και πιο σύνθετες λειτουργίες, πάνω σε ένα κύκλωμα.

Τέλος, στο κεφάλαιο 5, εξάγονται τα συμπεράσματα όλων των παραπάνω και πιο συγκεκριμένα, δίνεται μεγαλύτερη βαρύτητα στην βιβλιοθήκη pandas και στη χρησιμότητα αυτής, σε περιπτώσεις ανάγκης ανάλυσης μεγάλου όγκου δεδομένων.

ΚΕΦΑΛΑΙΟ 2

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΓΥΡΩ ΑΠΟ ΤΟ ΚΥΚΛΩΜΑ









2.1 ΨΗΦΙΑΚΕΣ ΛΟΓΙΚΕΣ ΠΥΛΕΣ

Θεμελιώδεις μονάδες των ψηφιακών κυκλωμάτων αποτελούν οι λογικές πύλες, οι οποίες πήραν την ονομασία τους από τις λογικές μεταβλητές της Άλγεβρας Boole.

Κάθε πύλη έχει μία ή περισσότερες εισόδους και αποκλειστικά μία έξοδο, που καθορίζεται από τις τιμές των εκάστοτε εισόδων, οι οποίες ανεξάρτητα από τη φύση τους, η τιμή τους περιορίζεται αποκλειστικά στο 0 ή 1. Κύριος σκοπός των πυλών είναι η επίλυση ενός προβλήματος στα ψηφιακά συστήματα, στα οποία υπάρχουν παραπάνω από μία εισοδοι. Η κάθε πύλη έχει τον δικό της πίνακα αληθείας ο οποίος απαρτίζεται από όλες τις πιθανές τιμές των εισόδων και το αποτέλεσμα αυτών (τιμή εξόδου).

Συνήθη χρήση των λογικών πυλών αποτελεί ο συνδυασμός αυτών, προκειμένου να δημιουργηθεί ένα σύνθετο ηλεκτρονικό κύκλωμα, το οποίο και θα αναλυθεί περαιτέρω στην επόμενη ενότητα.

Οι βασικότερες λογικές πύλες είναι οι AND, OR και NOT, ενώ επίσης υπάρχουν και οι BUFFER, NAND, NOR, XOR, XNOR. Τόσο οι γραφικές απεικονίσεις όσο και οι πίνακες αληθείας αυτών, παρουσιάζονται στον παρακάτω πίνακα:

ΕΙΣΟΔΟΙ		NOT	BUFFER	AND	NAND	OR	NOR	XOR	XNOR
									
X	Y	X'	X	$X * Y$	$(X * Y)'$	$X + Y$	$(X + Y)'$	$X (+) Y$	$(X (+) Y)'$
0	0	1	0	0	1	0	1	0	1
0	1	1	0	0	1	1	0	1	0
1	0	0	1	0	1	1	0	1	0
1	1	0	1	1	0	1	0	0	1

Πίνακας 1: Πίνακες Αληθείας

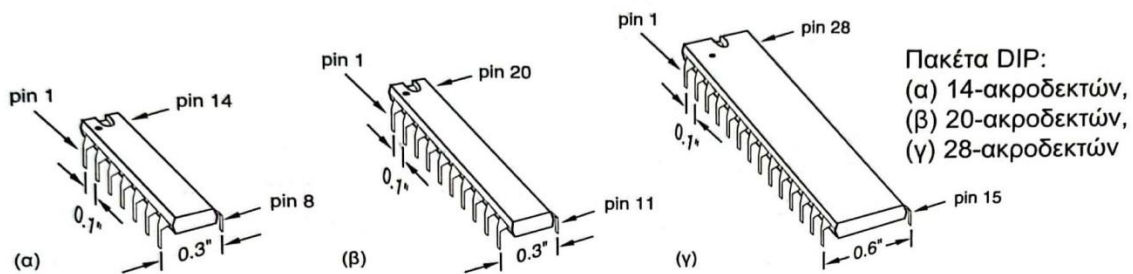
2.2 ΟΛΟΚΛΗΡΩΜΕΝΟ ΚΥΚΛΩΜΑ

Ως ολοκληρωμένο κύκλωμα ή IC (= Integrated Circuit) ορίζεται η ένωση λογικών πυλών σε ένα κύκλωμα, η οποία πραγματώνεται κατά κύριο λόγο πάνω σε φύλλα ημιαγωγών, συνήθως πυριτίου. Τα φύλλα αυτά είναι γνωστά και με τον όρο τσιπς (chips), έναν όρο, που ως κατ' επέκταση, χρησιμοποιούμε για να αναφερθούμε και στο κύκλωμα στο σύνολο του, ενώ ταυτόχρονα, σε περίπτωση που το τσιπ είναι της κλίμακας των μικρομέτρων, μπορεί να ονομαστεί και μικροτσιπ.

Το τσιπ με τη σειρά του εγκαθίσταται είτε σε πλαστική θήκη είτε σε κεραμική και στη συνέχεια, οι ακροδέκτες του συνενώνονται στους εξωτερικούς ακροδέκτες της θήκης αυτής, περατώνοντας με αυτόν τον τρόπο τη διαδικασία κατασκευής του ολοκληρωμένου κυκλώματος.

Με γνώμονα το πλήθος των λογικών πυλών, τα IC διακρίνονται στις τέσσερις παρακάτω βασικές κατηγορίες:

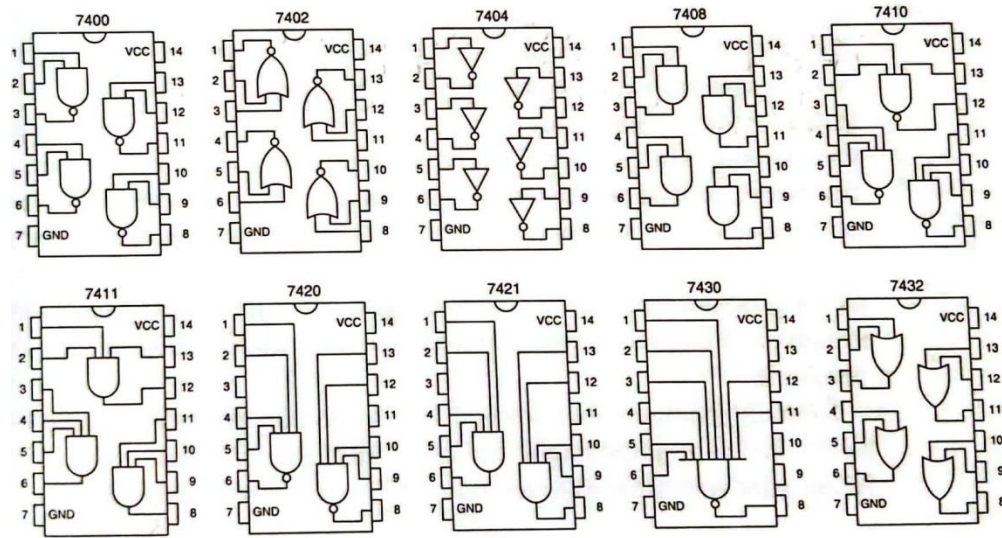
- **Μικρή κλίμακα ολοκλήρωσης (Small Scale Integration [SSI]):** Η θήκη αυτής της κλίμακας εμπεριέχει συνήθως ένα πλήθος πυλών, το οποίο δεν υπερβαίνει τις δέκα, ενώ ταυτόχρονα οριοθετείται από το πλήθος των ακροδεκτών, που απαρτίζουν το ολοκληρωμένο κύκλωμα.
- **Μεσαία κλίμακα ολοκλήρωσης (Medium Scale Integration [MSI]):** Στη συγκεκριμένη περίπτωση, η κάθε θήκη συντίθεται από ένα εύρος δέκα έως χιλίων πυλών.
- **Μεγάλη κλίμακα ολοκλήρωσης (Large Scale Integration [LSI]):** Στην κλίμακα αυτή, ο αριθμός των πυλών ανάγεται σε χιλιάδες.
- **Πολύ μεγάλη κλίμακα ολοκλήρωσης (Very Large Scale Integration [VLSI]):** Τέλος, το πλήθος των πυλών αυτής της κατηγορίας, ξεπερνάει κατά εκατοντάδες χιλιάδες αυτό των LSI κυκλωμάτων.



Εικόνα 1: Γραφική απεικόνιση τσιπ

(από το βιβλίο Wakerly, J. F. Ψηφιακή Σχεδίαση: Αρχές & Πρακτικές, σελίδα 15)

Διαγράμματα ακροδεκτών μερικών ολοκληρωμένων κυκλωμάτων SSI της σειράς 7400.



Εικόνα 2: Εσωτερικό Ολοκληρωμένου Κυκλώματος

(από το βιβλίο Wakerly, J. F. Ψηφιακή Σχεδίαση: Αρχές & Πρακτικές, σελίδα 15)



Εικόνα 3: Φυσική Μορφή ενός Ολοκληρωμένου Κυκλώματος, σειράς 7400

(ανακτήθηκε από: <https://www.electroschematics.com/7400-datasheet>)

2.3 ΦΥΣΙΚΗ ΣΧΕΔΙΑΣΗ

Κατά τη σχεδίαση ενός κυκλώματος, μετατρέπουμε την περιγραφή των λειτουργιών που πραγματοποιεί, σε προδιαγραφές που καθορίζουν τις λειτουργίες του αντίστοιχου κυκλώματος πάνω στα φύλλα ημιαγωγών.

Η προαναφερθείσα διαδικασία, όταν σχετίζεται με κυκλώματα της κλίμακας VLSI, τα οποία αποτελούνται από εκατομμύρια πύλες και τρανζίστορ, αποτελεί μία πολύ χρονοβόρα και δύσκολη δουλειά, γι' αυτό και ο έλεγχος και η επαλήθευση της ορθής λειτουργίας τους, είναι αδύνατον να γίνει χειρωνακτικά. Ως απόρροια αυτού, η προσομοίωση τους σε υπολογιστή με τη χρήση ειδικών εργαλείων, τα οποία ονομάζονται CAD (Computer Aided Design), κρίνεται αναγκαία. Με τα CAD αναπαρίστανται οι λειτουργίες του κυκλώματος, προκειμένου να αυτοματοποιηθούν οι διαδικασίες της σχεδίασης.

Για την υλοποίηση ενός λειτουργικού και ολοκληρωμένου κυκλώματος της κατηγορίας VLSI ακολουθείται μία ροή ενεργειών σχεδίασης (Design Flow), της οποίας οι επιμέρους διεργασίες θα περιγραφούν πιο κάτω.

Αρχικά, εισάγεται το σχέδιο, που αφορά τη διάταξη του κυκλώματος, σε υπολογιστικό περιβάλλον, ώστε να δημιουργηθεί η κατάλληλη βάση δεδομένων, η οποία αναλύει τις λειτουργίες του. Τα εργαλεία CAD προσφέρουν πληθώρα ειδών IC, που μπορεί κανείς να επιλέξει να προσομοιώσει, σύμφωνα με την ενέργεια που θέλει να επιτελέσει. Ανάλογα με την επιλογή που θα γίνει, προσφέρεται το κατάλληλο λογισμικό για την πραγμάτωση του φυσικού κυκλώματος.

Στη συνέχεια, με τη συνδρομή του λογισμικού, που μας χορηγήθηκε προηγουμένως, διεξάγεται έλεγχος που αφορά την ορθή λειτουργία του κυκλώματος ενώ επιπλέον υπάρχει η δυνατότητα δοκιμών για τυχόν αλλαγές, που θεωρούμε ότι αν εφαρμοστούν στο κύκλωμα, πιθανώς να το κάνουν είτε πιο γρήγορο, είτε πιο οικονομικό ως προς την κατασκευή του ή και άλλα τυχόν πιθανά αποτελέσματα, που ενδεχομένως να επιθυμούμε.

Τέλος, αφού επαληθεύσουμε ότι οι αλλαγές που διεξήχθησαν, επιφέρουν την επιθυμητή συμπεριφορά, προβαίνουμε στην φυσική κατασκευή του κυκλώματος.

2.4 ΧΩΡΟΘΕΤΗΣΗ

Θεμέλιο λίθο, για την φυσική σχεδίαση ενός ολοκληρωμένου κυκλώματος, αποτελεί η χωροθέτηση, η οποία σχετίζεται με το εσωτερικό του τσιπ (**chip area**) και την εγκατάσταση των βασικών δομικών στοιχείων (**κελιών**), που αφορούν την σχεδίαση, μέσα σε αυτό. Σκοπό της τακτοποίησης αυτής, αποτελεί η συρρίκνωση τόσο του μήκους του καλωδίου, που απαιτείται για την διασύνδεση των κελιών, όσο και οποιασδήποτε άλλης ποσότητας, η οποία είναι άμεσα συνυφασμένη με το συνολικό αποτέλεσμα και την αξία του κυκλώματος, όπως για παράδειγμα ο χρονοισμός του.

Η διαδικασία της χωροθέτησης αποτελείται από τρία διαφορετικά στάδια, τα οποία είναι: **α) η καθολική χωροθέτηση, β) η νομιμοποίηση και γ) η λεπτομερής χωροθέτηση.**

Το πρώτο βήμα που είναι η καθολική χωροθέτηση, ασχολείται με τα κελιά και την ακριβή τους θέση. Η τοποθέτηση των κελιών αυτών πραγματοποιείται, όχι κατά προσέγγιση αλλά με απόλυτα ορθά προσεγγμένους υπολογισμούς, ώστε το μέγεθος του καλωδίου, που αφορά την διεπαφή, να υποστεί συρρίκνωση. Ωστόσο, παρά τους υπολογισμούς αυτούς, κάποιες φορές, δεν αποκλείεται να διαπιστωθεί ότι τα κελιά αλληλεπικαλύπτουν το ένα το άλλο.

Το δεύτερο βήμα της χωροθέτησης, στο οποίο αναφερθήκαμε, είναι η νομιμοποίηση, η οποία συμβάλλει στην αντιμετώπιση της επικάλυψης των κελιών, στην οποία έγινε νύξη παραπάνω. Συγκεκριμένα, βοηθάει στην εξάλειψη του προβλήματος και τοποθετεί τα κελιά, με τρόπο τέτοιο, ώστε να μην επισκιάζει το ένα το άλλο, αλλά να έχει το καθένα τον δικό του αποκλειστικό χώρο.

Η λεπτομερής χωροθέτηση αποτελεί το τρίτο και τελευταίο βήμα και έπεται των παραπάνω βημάτων. Πιο αναλυτικά, συνεισφέρει στην βελτιστοποίηση του

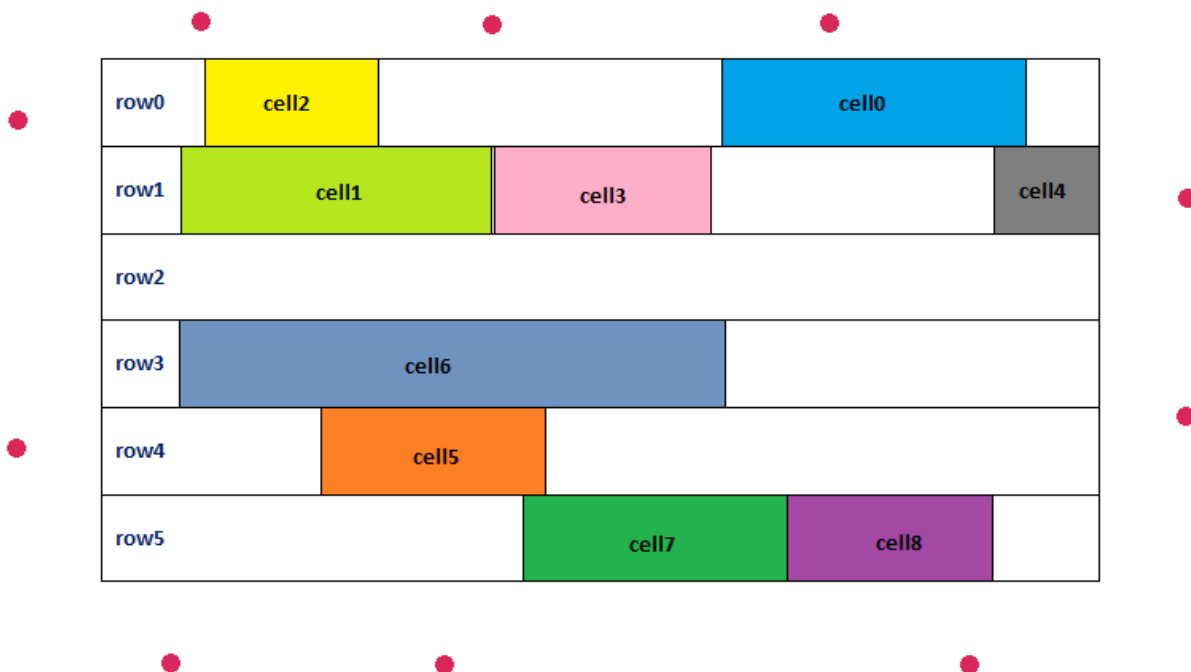
αποτελέσματος, που θα διεξαχθεί, διενεργώντας τροποποιήσεις, οι οποίες είναι μικρές και συνοπτικές και ενδέχεται να αφορούν τόσο τα κελιά και την διάταξη τους, όσο και τον επαναπροσδιορισμό τους στις εκάστοτε σειρές.

2.5 ΜΟΝΤΕΛΟΠΟΙΗΣΗ

Στην παρούσα εργασία, η μοντελοποίηση των κυκλωμάτων, που αναλύονται, εμφανίζει τα παρακάτω χαρακτηριστικά:

- Η **περιοχή του τσιπ** απεικονίζεται ως ένα ορθογώνιο παραλληλόγραμμο σχήμα, το οποίο εμπεριέχει ένα στατικό πλήθος σειρών ίδιου σχήματος, οι οποίες έχουν σταθερό ύψος αλλά και μήκος.
- Καθένα από τα δομικά στοιχεία, με τη σειρά τους έχουν όμοιο σχήμα με αυτό των σειρών, με μόνη διαφορά το μεταβλητό τους μήκος. Το ύψος παραμένει σταθερό και ίσο με αυτό των σειρών, ώστε να μπορούν να "κουμπώνουν" μέσα σε αυτές (standard cell design). Τα στοιχεία αυτά υλοποιούν μια λειτουργικότητα, όπως αυτή μιας πύλης AND ή OR ή NOT ή ακόμη και συνδυασμό πυλών όπως μία πύλη AND, ακολουθούμενη από μία πύλη OR, αλλά και άλλες αρκετά πιο σύνθετες λειτουργίες.
- Τα **Input/Output pins** (I/O pins ή terminals) έχουν αμετάβλητη θέση και είναι εγκαταστημένα στον περιφερειακό χώρο του chip, ενώ λόγω των μοναδιαίων διαστάσεων τους, απεικονίζονται και ως κουκίδες.

Η παραπάνω μοντελοποίηση έχει την εξής μορφή:



Εικόνα 4: Αναπαράσταση κυκλώματος

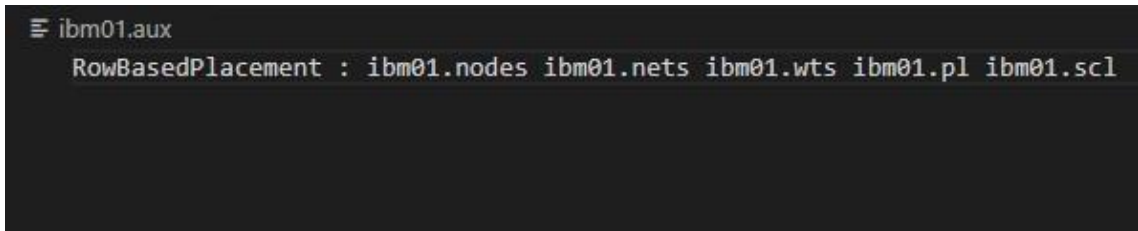
2.6 BOOKSHELF ΜΟΡΦΗ

Η ανωτέρω μοντελοποίηση, παρέχεται σε μορφή Bookshelf (= Bookshelf format), προκειμένου να εκτιμηθούν τα αποτελέσματά της, με τη χρήση κατάλληλων προγραμματιστικών εργαλείων. Η μορφή αυτή ασχολείται με την περιγραφή της αναπαράστασης του κυκλώματος και αποτελείται από έξι αρχεία:

α) **design_name.aux**, β) **design_name.nodes**, γ) **design_name.pl**,
δ) **design_name.nets**, ε) **design_name.scl** και στ) **design_name.wts**

Πιο αναλυτικά:

1. **design_name.aux**: Στο εσωτερικό του αρχείου αυτού, εμπεριέχεται μια λίστα ονομάτων, η οποία αφορά τα αρχεία που περιγράφουν ολόκληρο το κύκλωμα. Επί της ουσίας, περιέχει τις ονομασίες των επόμενων πέντε αρχείων, που θα αναλυθούν στη συνέχεια.



```
≡ ibm01.aux
RowBasedPlacement : ibm01.nodes ibm01.nets ibm01.wts ibm01.pl ibm01.scl
```

Εικόνα 5: Περιεχόμενα .aux αρχείου

2. **design_name.nodes**: Στο αρχείο αυτό, περιέχεται τόσο η γεωμετρική περιγραφή (δηλαδή πλάτος και ύψος) των κελιών και των Input/Output pins, όσο και το πλήθος τους. Και τα δύο μαζί, ανεξάρτητα από το είδος τους, ονομάζονται κόμβοι (=nodes). Η ονομασία αυτή προήλθε από τα **Net**, στα οποία και υπάγονται, καθότι η γραφική τους αναπαράσταση είναι όμοια με αυτή ενός γράφου.

```
ibm01.nodes
UCLA nodes 1.0
# Created      : Wed Oct  1 23:12:33 CDT 2003
# User        : nataraj@saturn.ee.iastate.edu (Natarajan Viswanathan)
# Platform    : SunOS 5.8 sparc SUNW,Sun-Fire-880

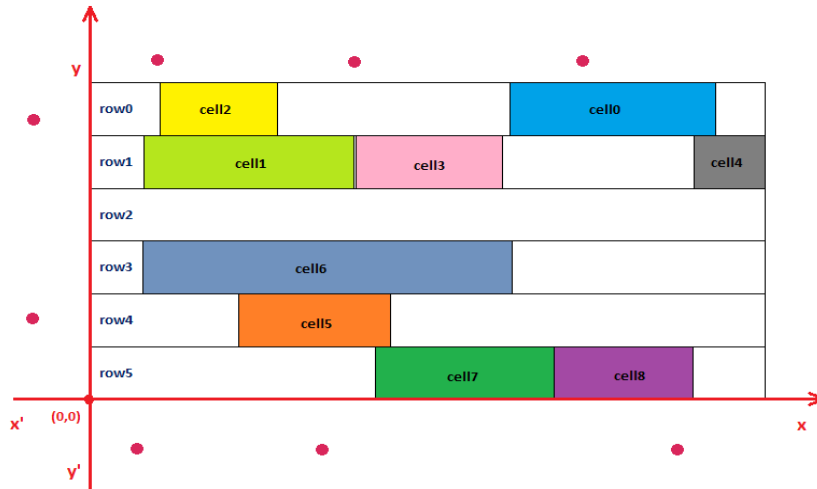
NumNodes :      12752  [Συνολικός αριθμός κόμβων (κόμβος = κελί ή I/O pin, I/O pin = terminal)]
NumTerminals :      246  [Συνολικός αριθμός I/O pins]
  a0         16         16  (Όνομα κόμβου, πλάτος, ύψος)
  a1         14         16
  a2          8         16
  a3          6         16
  a4          6         16
  a5          8         16
  .....
  .....
  .....

  p241        1         1 terminal (Όνομα κόμβου, πλάτος, ύψος, είδος = I/O pin)
  p242        1         1 terminal
  p243        1         1 terminal
  p244        1         1 terminal
  p245        1         1 terminal
  p246        1         1 terminal
```

Εικόνα 6: Περιεχόμενα .nodes αρχείου

3. **design_name.pl:** Στο συγκεκριμένο αρχείο, καταγράφονται οι συντεταγμένες (τοποθεσίες πάνω στο chip και περιμετρικά από αυτό) των κελιών, καθώς και των I/O pins.

Ως αρχή των αξόνων [= σημείο (0,0)], έχουμε θεωρήσει την κάτω αριστερή γωνία του chip (Lower Left Corner). Εξαιτίας αυτού, μπορεί να παρατηρηθούν αρνητικές τιμές σε ορισμένες συντεταγμένες, μιας και αυτές αντιστοιχούν στα terminals.



Εικόνα 7: Γραφική Αναπαράσταση Συντεταγμένων

```

ibm01.pl
UCLA pl 1.0
# Created
# Platform

a0 1402 688 : S
a1 31 80 : S
a2 433 1424 : S
a3 1414 944 : S
a4 952 1024 : N
a5 880 992 : N

.....
.....
.....

p241 1582 1404 : FW
p242 1582 1428 : FW
p243 1582 1452 : FW
p244 1582 1476 : FW
p245 1582 1500 : FW
p246 1582 1524 : FW

```

Εικόνα 8: Περιεχόμενα .pl αρχείου

4. **design_name.nets:** Στην περίπτωση αυτή, γίνεται περιγραφή των net του κυκλώματος και του συνολικού πλήθους αυτών.

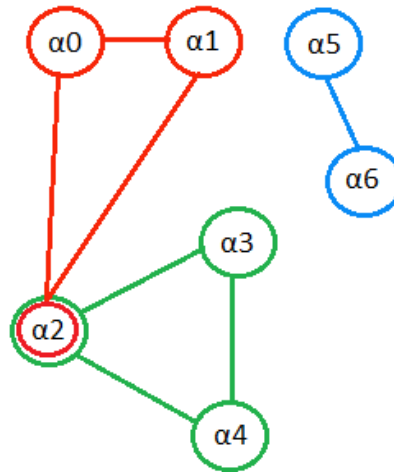
Ως **Net**, ορίζεται ένα σύνολο από κόμβους (κελιά & I/O pins) όπου **όλοι** συνδέονται μεταξύ τους. Η σύνδεση τους υλοποιείται με τη χρήση καλωδίου και το συνολικό μήκος αυτού μπορεί να υπολογισθεί, κατά προσέγγιση, ως η ημι-περίμετρος του νοητού παραλληλογράμμου, μέσα στο οποίο περικλείονται (**Half-Perimeter Wire-Length, HPWL**).

Ως απόρροια αυτού, διεξάγεται άθροιση, η οποία σχετίζεται με όλα τα επιμέρους μήκη καλωδίων των προαναφερθέντων net, με σκοπό να υπολογιστεί το συνολικό μήκος τους, που αφορά την ολική διασύνδεση της σχεδίασης.

Το εκάστοτε net, χαρακτηρίζεται από το δικό του NetDegree, δηλαδή το πλήθος των κόμβων, από το οποίο απαρτίζεται.

Επιπροσθέτως, όλοι οι κόμβοι μπορούν να ανήκουν σε περισσότερα από ένα net, οπότε συνακόλουθα, δύο ή περισσότερα nets μπορούν να αποτελούνται από τουλάχιστον έναν κοινό κόμβο, ενώ δεν αποκλείεται να μην έχουν και κανένα κοινό.

Κόμβοι: α0, α1, α2, α3, α4, α5, α6
Nets: net0, net1, net2



Εικόνα 9: Αναπαράσταση Net σε μορφή γράφου

```
ibm01.nets
UCLA nets 1.0
# Created      : Wed Oct  1 23:12:33 CDT 2003
# User        : nataraj@saturn.ee.iastate.edu (Natarajan Viswanathan)
# Platform    : SunOS 5.8 sparc SUNW,Sun-Fire-880

NumNets : 14111 (Συνολικός αριθμός Net του κυκλώματος)
NumPins : 50566
NetDegree : 2 (Αριθμός κόμβων του Net)
  p198 B (Όνομα 1ου κόμβου που ανήκει στο net, παράβλεψη)
  a8117 B (Όνομα 2ου κόμβου που ανήκει στο net, παράβλεψη)
NetDegree : 2 (Επόμενο NetDegree => αρχή επόμενου Net)
  a3044 B
  p98 B
NetDegree : 2
  a3767 B
  p245 B
NetDegree : 2
  a4992 B
  p231 B
NetDegree : 2
  a6837 B
  p9 B
.....
.....
```

Εικόνα 10: Περιεχόμενα .nets αρχείου

5. **design_name.scl:** Το σχετικό αυτό αρχείο, πραγματοποιεί περιγραφή, η οποία αφορά αποκλειστικά το εσωτερικό του chip. Πιο συγκεκριμένα, περιγράφει τόσο το πλήθος των σειρών, από τις οποίες συντίθεται, όσο και την γεωμετρία αυτών.

Όσον αφορά αυτές, δίνονται η συντεταγμένη y της κάτω αριστερής γωνίας, το ύψος και η συντεταγμένη x της κάτω αριστερής και δεξιάς γωνίας.

```
ibm01.scl
UCLA scl 1.0
# Created      : Wed Oct  1 23:12:33 CDT 2003
# User        : nataraj@saturn.ee.iastate.edu (Natarajan Viswanathan)
# Platform    : SunOS 5.8 sparc SUNW,Sun-Fire-880

NumRows : 96 (Συνολικός αριθμός σειρών στο εσωτερικό του chip)

CoreRow Horizontal (Αρχή περιγραφής σειράς)
Coordinate :      0 (Συντεταγμένη y κάτω αριστερής γωνίας της σειράς)
Height     :     16 (Υψος σειράς)
Sitewidth  :      1
Sitespacing :     1
Siteorient :      N
Sitesymmetry :     Y
SubrowOrigin :      0 Numsites :    1550 (Συντεταγμένη x κάτω αριστερής γωνίας,
End (Τέλος περιγραφής σειράς)          Συντεταγμένη x κάτω δεξιάς γωνίας,
                                         της σειράς)
CoreRow Horizontal
Coordinate :     16
Height     :     16
Sitewidth  :      1
Sitespacing :     1
Siteorient :     FS
Sitesymmetry :     Y
SubrowOrigin :      0 Numsites :    1550
End

.....
.....
```

Εικόνα 11: Περιεχόμενα .scl αρχείου

6. **design_name.wts**: Το παρόν αποτελεί και το τελευταίο αρχείο του bookshelf format, το οποίο ωστόσο δεν χρησιμοποιήθηκε στην εργασία αυτή, καθώς δεν ήταν αναγκαία η χρήση του.

```
ibm01.wts
UCLA wts 1.0
# Created      : Wed Oct  1 23:12:33 CDT 2003
# User         : nataraj@saturn.ee.iastate.edu (Natarajan Viswanathan)
# Platform    : SunOS 5.8 sparc SUNW,Sun-Fire-880

.....
.....

p241          0
p242          0
p243          0
p244          0
p245          0
p246          0
  a0          256
  a1          224
  a2          128
  a3           96
  a4           96
  a5          128

.....
.....
```

Εικόνα 12: Περιεχόμενα .wts αρχείου

ΚΕΦΑΛΑΙΟ 3

PYTHON ΚΑΙ ΒΙΒΛΙΟΘΗΚΗ PANDAS

3.1 PYTHON

3.1.1 ΠΡΟΕΛΕΥΣΗ

Δημιουργήθηκε από τον Guido Van Rossum και κυκλοφόρησε για πρώτη φορά το 1991. Το όνομά της προήλθε από τους κωμικούς Monty Python, ως φόρο τιμής.



Εικόνα 13: Σήμα της Python (πηγή: <https://www.python.org/community/logos/>)

3.1.2 ΟΡΙΣΜΟΣ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Η Python ορίζεται ως μία υψηλού επιπέδου, διερμηνευόμενη (Interpreted) γλώσσα προγραμματισμού, δυναμικής σημασιολογίας, με γενική χρήση η οποία είναι ικανή να πραγματοποιήσει μεγάλο εύρος προγραμματιστικών προτύπων. Επιπλέον, ενσωματώνει δυνατότητες αντικειμενοστραφούς (Object Oriented), διαδικαστικού (Procedural), προστακτικού (Imperative) και συναρτησιακού (Functional) προγραμματισμού.

Χαρακτηρίζεται δυναμική (Dynamic), διότι οι τύποι των μεταβλητών δεν χρειάζεται να αρχικοποιηθούν, αλλά αντιθέτως ορίζονται δυναμικά κατά την εκτέλεση του προγράμματος. Επιπρόσθετα, διαθέτει ένα αυτόματο σύστημα συλλογής απορριμμάτων, το οποίο αποδεδμεύει τον προγραμματιστή από την ανάγκη αυτόβουλης διαχείρισης μνήμης.

Όντας γλώσσα υψηλού επιπέδου (High-Level), το συντακτικό της είναι ευανάγνωστο και ευκόλως κατανοητό, καθότι είναι πολύ κοντινό με αυτό της φυσικής ανθρώπινης γλώσσας. Τα χαρακτηριστικά της αυτά, την έχουν ξεχωρίσει και αποτελούν έναν από τους κυριότερους λόγους της διαδεδομένης χρήσης της, μιας και επιτρέπει την εύκολη και γρήγορη ανάπτυξη εφαρμογών αλλά και την συντήρηση αυτών. Όπως γίνεται αντιληπτό, τα προαναφερθέντα γνωρίσματα της Python, συμβάλλουν βελτίωση της παραγωγικότητας των χρηστών, επιτρέποντάς στους χρήστες να υλοποιούν λειτουργίες, πολύ πιο γρήγορα και σε λιγότερες γραμμές κώδικα, απ' ότι σε άλλες γλώσσες προγραμματισμού, όπως για παράδειγμα η Java και η C++.

Προηγουμένως, η Python χαρακτηρίστηκε ως διερμηνευόμενη γλώσσα, καθόσον χρησιμοποιεί διερμηνευτή και όχι μεταγλωττιστή, για τον έλεγχο της ορθότητας του κώδικα και στη συνέχεια της εκτέλεσής του. Πιο συγκεκριμένα, σε αντίθεση με τη χρήση μεταγλωττιστή, ο έλεγχος πραγματοποιείται σειρά-σειρά και μεταφράζεται σε κώδικα μηχανής. Σε περίπτωση σφάλματος, τότε σταματάει η εκτέλεση του προγράμματος και παρουσιάζεται το κατάλληλο μήνυμα, ενώ αν δεν εμφανιστεί κάποιο σφάλμα τότε το πρόγραμμα εκτελείται κανονικά, αλλά δεν παράγεται εκτελέσιμο αρχείο.

Παράλληλα με όλα τα παραπάνω, είναι μια γλώσσα ανοιχτού κώδικα (Open-Source), επιτρέποντας έτσι τη δωρεάν λήψη και χρήση αυτής σε οποιαδήποτε εφαρμογή, ενώ ταυτόχρονα παρέχει τη δυνατότητα αδέσμευτης τροποποίησης και αναδιανομής της.

Τέλος, η χρήση και εγκατάστασή της είναι εφικτή σε όλα τα λειτουργικά συστήματα και συγχρόνως δύναται να επικοινωνήσει με την πλειονότητα των σύγχρονων γλωσσών προγραμματισμού, κάνοντας χρήση των κατάλληλων διασυνδέσεων.

3.1.3 ΔΗΜΟΦΙΛΕΣΤΕΡΟΙ ΤΟΜΕΙΣ ΧΡΗΣΗΣ ΡΥΤΗΟΝ

Η ευελιξία που προσφέρει σε συνάφεια με τις δυνατότητες της, χάρη στην πληθώρα των Framework και βιβλιοθηκών που κατέχει, επιτρέπουν την πραγμάτωση πολύ απλών έως και πολύ περίπλοκων εφαρμογών. Οι πιο διαδεδομένοι τομείς στους οποίους τη συναντάμε είναι αυτοί της Μηχανικής Μάθησης (Machine Learning) και της Ανάλυσης Δεδομένων (Data Analysis), όπου η τελευταία υπάγεται στην Επιστήμη Δεδομένων (Data Science).

3.2 ΒΙΒΛΙΟΘΗΚΗ pandas

3.2.1 ΠΡΟΕΛΕΥΣΗ

Η βιβλιοθήκη pandas άρχισε να αναπτύσσεται το 2008, το 2009 έγινε Open Source και εκδόθηκε για πρώτη φορά το 2012 από την εταιρεία AQR Capital Management. Το όνομά της προέρχεται από τον όρο Panel Data.



Εικόνα 14: Σήμα της pandas (πηγή: <https://pandas.pydata.org/static/img/pandas.svg>)

3.2.2 ΟΡΙΣΜΟΣ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Όπως κάθε γλώσσα προγραμματισμού, έτσι και η Python αποτελείται από μία πληθώρα βιβλιοθηκών. Ως βιβλιοθήκη, ορίζεται μία συλλογή λειτουργιών και μεθόδων, η οποία παρέχει τη δυνατότητα στο άτομο, που την χειρίζεται, να χρησιμοποιεί τις παροχές που προσφέρονται. Με αυτόν τον τρόπο, εξοικονομεί χρόνο στην ανάπτυξη ενός προγράμματος, καθώς δεν ξεκινάει κάθε φορά τον κώδικα του από την αρχή. Μία απ' αυτές τις βιβλιοθήκες είναι και η pandas.

Βασικά χαρακτηριστικά της είναι η δυσδιάστατη δομή δεδομένων DataFrame και η παραπλήσιά της δομή Series, με τη διαφορά ότι η τελευταία είναι μονοδιάστατη.

Επιπλέον, είναι ικανή στην ανάγνωση δεδομένων από διάφορους τύπους αρχείων όπως CSV, Microsoft Excel, αρχεία κειμένου, SQL βάσεις δεδομένων και

HDF5. Αντίστοιχα, εκτός από την ανάγνωση, το ίδιο ικανή είναι και στην εγγραφή δεδομένων στα προαναφερθέντα αρχεία.

Επίσης, ένα σύνηθες πρόβλημα, που παρουσιάζεται κατά την συγγραφή κώδικα, είναι η διαχείριση των ελλιπών ή χαμένων δεδομένων. Προκειμένου να λυθεί το πρόβλημα αυτό, η βιβλιοθήκη δίνει σε αυτά την τιμή NaN ή None διευκολύνοντας έτσι τον εντοπισμό τους για περαιτέρω διαχείριση.

Επιπροσθέτως, όσον αφορά τα οργανωμένα αλλά και τα μη καλά οργανωμένα δεδομένα, είναι αρκετά ευέλικτη ως προς την αναδιαμόρφωση και οργάνωση τους. Αυτό επιτυγχάνεται παρέχοντας τη δυνατότητα εύκολης εισαγωγής αλλά και διαγραφής στηλών, γρήγορης συγχώνευσης και προσθήκης δεδομένων και τέλος, με την πολύ ικανή συνάρτηση **groupby**, ομαδοποιεί τα δεδομένα σε διάφορες κατηγορίες και συγχρόνως εφαρμόζει επιπλέον συναρτήσεις πάνω σε αυτές .

3.2.3 DATAFRAMES

Όπως έχει επισημανθεί και ανωτέρω, τα DataFrames αποτελούν μία δυσδιάστατη δομή δεδομένων, η οποία διακρίνεται λόγω της ικανότητας της να διαχειρίζεται και να παραμετροποιεί δεδομένα γρήγορα και αποτελεσματικά, ανεξάρτητα από τον τύπο τους, ενώ ταυτόχρονα παρέχει και ενσωματωμένη αρίθμηση (indexing) των γραμμών.

Η απεικόνιση αυτών είναι η ευρέως διαδεδομένη μορφή ενός πίνακα. Το γεγονός αυτό τα καθιστά τα πλέον ιδανικά για την προβολή δεδομένων, καθόσον η μορφοποίηση αυτή είναι ευανάγνωστη και κατανοητή στο ανθρώπινο μάτι.

	Node_name	Width	Height	Type	Row_number	Coordinate_x_min	Coordinate_y_min	Coordinate_x_max	Coordinate_y_max
0	a1	14	10	Non_Terminal	row7	10	170	24	180
1	a2	8	10	Non_Terminal	row7	25	170	33	180
2	a3	6	10	Non_Terminal	row3	50	130	56	140
3	a4	6	10	Non_Terminal	row3	60	130	66	140
4	a5	8	10	Non_Terminal	row3	70	130	78	140
5	a6	8	10	Non_Terminal	row3	80	130	88	140
6	a7	8	10	Non_Terminal	row2	0	120	8	130
7	a8	8	10	Non_Terminal	row2	15	120	23	130
8	a9	12	10	Non_Terminal	row2	20	120	32	130
9	a10	42	10	Non_Terminal	row0	0	100	42	110
10	a11	8	10	Non_Terminal	row2	40	120	48	130
11	a12	12	10	Non_Terminal	row0	50	100	62	110
12	a13	8	10	Non_Terminal	row1	10	110	18	120
13	a14	6	10	Non_Terminal	row1	20	110	26	120
14	a15	14	10	Non_Terminal	row1	30	110	44	120
15	a16	8	10	Non_Terminal	row1	50	110	58	120
16	a17	8	10	Non_Terminal	row4	80	140	88	150
17	a18	24	10	Non_Terminal	row3	20	130	44	140
18	a19	4	10	Non_Terminal	row7	50	170	54	180
19	a20	6	10	Non_Terminal	row4	65	140	71	150
20	a21	2	10	Non_Terminal	row7	70	170	72	180
21	a22	44	10	Non_Terminal	row4	40	140	84	150
22	a23	44	10	Non_Terminal	row6	10	160	54	170
23	a24	14	10	Non_Terminal	row5	0	150	14	160
24	a25	24	10	Non_Terminal	row5	20	150	44	160
25	a26	44	10	Non_Terminal	row5	50	150	94	160

Εικόνα 15: Απεικόνιση ενός DataFrame

Όπως θα αναφερθεί και στο επόμενο κεφάλαιο, η χρήση της συγκεκριμένης δομής συνέβαλε στην υλοποίηση της πλειοψηφίας των συναρτήσεων, που αφορούσαν την ανάλυση δεδομένων και την εξαγωγή συμπερασμάτων βάση αυτής.

3.2.4 ΘΕΤΙΚΑ ΤΗΣ pandas

Όπως και παραπάνω στην περιγραφή των χαρακτηριστικών της, γίνεται ευκόλως κατανοητός ο λόγος, για τον οποίο έχει ξεχωρίσει η παρούσα βιβλιοθήκη στον τομέα ανάλυσης και διαχείρισης.

Η παρούσα βιβλιοθήκη, πέρα από τα θετικά της που απορρέουν από την παραπάνω περιγραφή των χαρακτηριστικών της, διαθέτει πολλά ακόμα προτερήματα. Ένα από τα πιο σπουδαία είναι η δυνατότητα της να εκτελεί περίπλοκες και σύνθετες διαδικασίες σε πολύ μικρό πλήθος εντολών. Ως απόδειξη αυτού, παραθέτεται ένα πολύ σύντομο παράδειγμα:

Με την εκτέλεση της μίας γραμμής κώδικα, που φαίνεται στην κάτωθι φωτογραφία, το DataFrame της **εικόνας 15**, αποκτά μία νέα στήλη (Size), η οποία

δημιουργείται ως αποτέλεσμα του γινομένου των στηλών Width και Height για την κάθε εγγραφή ξεχωριστά.

```
dataframe_a['Size'] = dataframe_a['Width'] * dataframe_a['Height']
```

Εικόνα 16: Δημιουργία νέας στήλης

	Node_name	Width	Height	Type	Row_number	Coordinate_x_min	Coordinate_y_min	Coordinate_x_max	Coordinate_y_max	Size
0	a1	14	10	Non_Terminal	row7	10	170	24	180	140
1	a2	8	10	Non_Terminal	row7	25	170	33	180	80
2	a3	6	10	Non_Terminal	row3	50	130	56	140	60
3	a4	6	10	Non_Terminal	row3	60	130	66	140	60
4	a5	8	10	Non_Terminal	row3	70	130	78	140	80
5	a6	8	10	Non_Terminal	row3	80	130	88	140	80
6	a7	8	10	Non_Terminal	row2	0	120	8	130	80
7	a8	8	10	Non_Terminal	row2	15	120	23	130	80
8	a9	12	10	Non_Terminal	row2	20	120	32	130	120
9	a10	42	10	Non_Terminal	row0	0	100	42	110	420
10	a11	8	10	Non_Terminal	row2	40	120	48	130	80
11	a12	12	10	Non_Terminal	row0	50	100	62	110	120
12	a13	8	10	Non_Terminal	row1	10	110	18	120	80
13	a14	6	10	Non_Terminal	row1	20	110	26	120	60
14	a15	14	10	Non_Terminal	row1	30	110	44	120	140
15	a16	8	10	Non_Terminal	row1	50	110	58	120	80
16	a17	8	10	Non_Terminal	row4	80	140	88	150	80
17	a18	24	10	Non_Terminal	row3	20	130	44	140	240
18	a19	4	10	Non_Terminal	row7	50	170	54	180	40
19	a20	6	10	Non_Terminal	row4	65	140	71	150	60
20	a21	2	10	Non_Terminal	row7	70	170	72	180	20
21	a22	44	10	Non_Terminal	row4	40	140	84	150	440
22	a23	44	10	Non_Terminal	row6	10	160	54	170	440
23	a24	14	10	Non_Terminal	row5	0	150	14	160	140
24	a25	24	10	Non_Terminal	row5	20	150	44	160	240
25	a26	44	10	Non_Terminal	row5	50	150	94	160	440

Εικόνα 17: Απεικόνιση της νέας στήλης

Το παραπάνω παράδειγμα αποτελεί μια σύντομη και απλή επίδειξη των ικανοτήτων της ενώ ταυτόχρονα, μπορεί να πραγματοποιήσει και αρκετά πιο σύνθετες λειτουργίες, τηρώντας παρά ταύτα το πολύ μικρό πλήθος εντολών.

Αξιοσημείωτο επίσης αποτελεί το γεγονός ότι όλα όσα έχουν προαναφερθεί για αυτήν την βιβλιοθήκη, μπορούν να εκτελεστούν σε σύντομο, για τον όγκο των δεδομένων, χρονικό διάστημα ενώ ταυτόχρονα, η χρήση της γλώσσας Cython ή C, σε συγκεκριμένα κομμάτια κώδικα, συμβάλλει στην επίτευξη του παραπάνω γεγονότος. Τέλος, μίας και υπάγεται στην οικογένεια βιβλιοθηκών της Python, αυτό της

επιτρέπει να αξιοποιήσει τις δυνατότητες αυτών. Μία από αυτές τις βιβλιοθήκες η οποία χρησιμοποιήθηκε και στην παρούσα εργασία είναι η Matplotlib.

3.2.5 ΑΡΝΗΤΙΚΑ ΤΗΣ pandas

Ένας από τους λόγους, που θα απέτρεπε κάποιον να κάνει χρήση αυτής της βιβλιοθήκης, θα ήταν αν τα δεδομένα του είχαν τρισδιάστατη μορφή. Ενώ στην περίπτωση των δύο διαστάσεων θεωρείται από τις καταλληλότερες, στην περίπτωση των τριών φαίνεται να αδυνατεί να ανταπεξέλθει αποτελεσματικά.

3.3 ΒΙΒΛΙΟΘΗΚΗ MATPLOTLIB

Η Matplotlib αποτελεί άλλη μία βιβλιοθήκη της Python, η οποία έχει τη δυνατότητα να δημιουργεί στατικές, κινούμενες και διαδραστικές γραφικές απεικονίσεις.

Στην συγκεκριμένη εργασία, σε συνεργασία με τα DataFrames παρήχθησαν γραφικές αναπαραστάσεις των αποτελεσμάτων, που προέκυψαν από την ανάλυση δεδομένων, που έγινε πάνω στα δοσμένα κυκλώματα.



Εικόνα 18: Σήμα της matplotlib (πηγή: <https://matplotlib.org/stable/gallery/misc/logos2.html>)

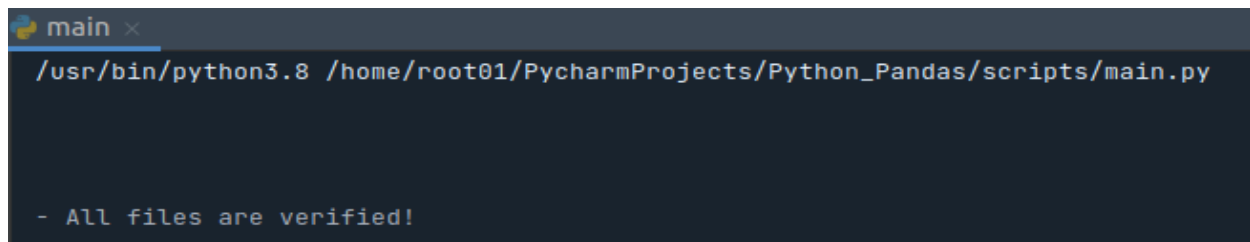
ΚΕΦΑΛΑΙΟ 4

ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΥΛΟΠΟΙΗΜΕΝΟΥ ΚΩΔΙΚΑ

4.1 ΣΥΝΑΡΤΗΣΗ ΕΠΑΛΗΘΕΥΣΗΣ ΑΡΧΕΙΩΝ

Αρχικά, δημιουργήθηκε η συνάρτηση `verify_files`, η οποία κάνει χρήση του αρχείου `.aux`, του οποίου η χρησιμότητα αναφέρθηκε και σε προηγούμενο κεφάλαιο, με σκοπό να ελέγξει είτε αν υπάρχουν όλα τα απαραίτητα αρχεία της `bookshelf` μορφής, είτε αν αυτά δεν είναι κενά και τέλος, αν είναι περισσότερα από ότι θα έπρεπε.

Αν όλα τα απαραίτητα αρχεία υπάρχουν και δεν είναι κενά τότε εμφανίζεται το κατάλληλο μήνυμα και έπειτα αρχίζει η διαδικασία `parsing` αυτών:

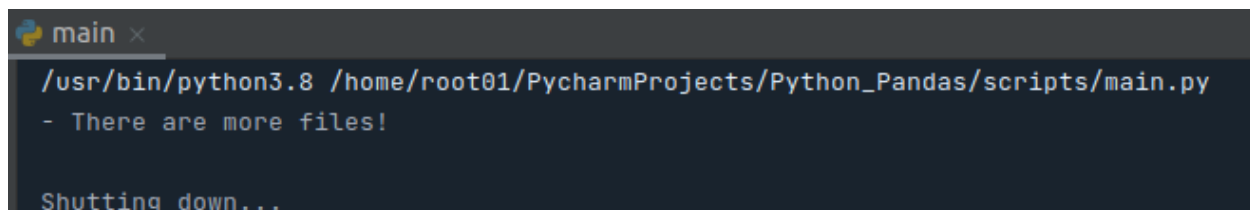


```
main x
/usr/bin/python3.8 /home/root01/PycharmProjects/Python_Pandas/scripts/main.py

- All files are verified!
```

Εικόνα 19: Μήνυμα επιτυχημένης επαλήθευσης

Σε αντιστοιχη περίπτωση, αν τα αρχεία δεν είναι όπως θα έπρεπε, εμφανίζεται το αντίστοιχο μήνυμα και μετά το πρόγραμμα παύει να εκτελείται:



```
main x
/usr/bin/python3.8 /home/root01/PycharmProjects/Python_Pandas/scripts/main.py
- There are more files!

Shutting down...
```

Εικόνα 20: Μήνυμα μη επιτυχημένης επαλήθευσης

4.2 ΣΥΝΑΡΤΗΣΗ PARSER

Αφού ο έλεγχος είναι επιτυχής, ξεκινάει το διαπέρασμα των αρχείων. Στο τέλος της συνάρτησης αυτής, θα έχουν δημιουργηθεί τρεις λίστες, οι οποίες θα έχουν όλους τους κόμβους, nets και σειρές του κυκλώματος, με όλα τα δεδομένα, που τα περιγράφουν, καθώς και τις σχέσεις μεταξύ τους. Επίσης, δημιουργείται και ένα Design αντικείμενο, το οποίο περιγράφει ολόκληρο το κύκλωμα.

Η συνάρτηση parser ξεκινάει διαπερνώντας το αρχείο **.nodes**, από το οποίο αντλώνται τα ονόματα των κόμβων, μαζί με το ύψος, πλάτος και τον τύπο τους. Κατά τη διάρκεια αυτής της διαδικασίας, απομονώνονται κάθε φορά πληροφορίες για τον εκάστοτε κόμβο και δημιουργείται ένα στιγμιότυπο της κλάσης Node για αυτόν, το οποίο ενημερώνεται με τα στοιχεία, που μόλις αναγνώστηκαν, από το αρχείο. Αφού ολοκληρωθούν οι παραπάνω ενέργειες, προστίθεται το τρέχον στιγμιότυπο στη λίστα των κόμβων και επαναλαμβάνεται η ίδια διαδικασία για κάθε κόμβο του κυκλώματος.

Εν συνεχεία, γίνεται ανάγνωση του **.pl** αρχείου, από το οποίο προκύπτουν πληροφορίες, που αφορούν τις συντεταγμένες x, y των κόμβων, που δημιουργήθηκαν προηγουμένως και επακόλουθα γίνεται ενημέρωση της λίστας τους, προσθέτοντας τα παραπάνω στοιχεία, στον αντίστοιχο κάθε φορά κόμβο.

Σειρά έχει το **.nets** αρχείο, στο οποίο αναγράφεται το πλήθος των net του κυκλώματος, καθώς και πληροφορίες για αυτά, όπως ο αριθμός των κόμβων, από τους οποίους απαρτίζονται, αλλά και τα ονόματα αυτών. Σύμφωνα με αυτά, δημιουργούνται στιγμιότυπα της κλάσης Net, ακολουθώντας την ίδια σειρά ενεργειών, όπως αναφέρθηκε και προηγουμένως στο αρχείο **.nodes**, μόνο που αυτή τη φορά εισάγεται το όνομα του εκάστοτε net, το πλήθος των κόμβων αλλά και μία λίστα αυτών. Επιπλέον, με τους κατάλληλους υπολογισμούς βρίσκονται οι εσωτερικοί και οι εξωτερικοί κόμβοι των Net και με την χρήση των δευτέρων, βρίσκουμε το μέγεθος του Net αλλά και το απαιτούμενο μήκος καλωδίου, που αυτό χρειάζεται. Αφού γίνουν όλοι οι υπολογισμοί για το καθένα και ενημερωθεί το τρέχον στιγμιότυπο, τότε εισάγεται και στην αντίστοιχη λίστα, που έχει δημιουργηθεί για αυτά.

Με την γραφή του απαιτούμενου κώδικα, γίνεται αντιστοιχία των net, με τα στιγμιότυπα της κλάσης Node που περιέχονται μέσα στη λίστα των κόμβων, με σκοπό να δημιουργηθεί και εκεί αντίστοιχη λίστα για τον κάθε κόμβο ξεχωριστά, στην οποία θα περιέχονται τα ονόματα των Net στα οποία ανήκει.

Τέλος, γίνεται ανάγνωση του αρχείου `.scl`, από το οποίο αντλούνται οι απαραίτητες πληροφορίες, που αφορούν τις σειρές και οι οποίες είναι τα ονόματά τους και οι συντεταγμένες των τεσσάρων γωνιών τους. Αντίστοιχα, όπως έγινε και στις προηγούμενες περιπτώσεις, δημιουργούνται στιγμιότυπα της κλάσης Row, τα οποία αφού ενημερωθούν, εισάγονται στη λίστα των σειρών του κυκλώματος. Ομοίως με παραπάνω, με τους κατάλληλους υπολογισμούς, βρίσκονται οι κόμβοι και τα net που υπάγονται στην κάθε σειρά, ενημερώνοντας έτσι τα στιγμιότυπα των σειρών, της τρέχουσας λίστας. Ταυτόχρονα, το ίδιο γίνεται και για τα στιγμιότυπα των nodes και των net, των υπόλοιπων δύο λιστών, στα οποία προσθέτουμε επιπλέον πληροφορίες, που αφορούν τις σειρές και συγκεκριμένα, σε ποιες ανήκει ο κάθε κόμβος και το κάθε net.

Αφού ολοκληρωθούν όλα τα ανωτέρω, δημιουργείται και ένα στιγμιότυπο της κλάσης Design, που περιγράφει το κύκλωμα και πιο αναλυτικά, τόσο το ύψος και πλάτος του, όσο και τα πλήθη των κόμβων, των I/O pins, των net και των σειρών, ξεχωριστά. Ακόμη, περιλαμβάνεται η συνολική έκταση, που αυτό καταλαμβάνει, αλλά και ο αριθμός, που αντικατοπτρίζει το εμβαδόν, το οποίο έχει καταλειφθεί απ' τους κόμβους. Επίσης, περιέχει το συνολικό μήκος καλωδίου, που απαιτείται για τη διασύνδεση όλου του κυκλώματος, αλλά και την πυκνότητα αυτού, σε ποσοστό επί τις εκατό.

Πραγματοποιώντας όλες τις παραπάνω διαδικασίες, θα έχει ολοκληρωθεί η δημιουργία των τριών λιστών, στις οποίες υπάρχει πλήρης περιγραφή όλου του κυκλώματος. Η πληροφορία αυτών είναι περισσότερη από αυτή, που δινόταν στα αρχικά αρχεία και προήλθε από την επιπλέον ανάλυση δεδομένων, που υλοποιήθηκε.

4.3 ΔΗΜΙΟΥΡΓΙΑ DATAFRAMES

4.3.1 ΠΕΡΙΓΡΑΦΗ ΤΡΟΠΟΥ ΔΗΜΙΟΥΡΓΙΑΣ ΤΩΝ DATAFRAME

Απ' τις τρεις λίστες που αναφέρθηκαν προηγουμένως, μεταφέρονται τα δεδομένα σε DataFrames, ώστε να υλοποιηθούν πάνω σε αυτά επιπλέον συναρτήσεις και να αξιοποιηθεί η ευκολία, αλλά και η ταχύτητα, που παρέχουν.

Στη συνέχεια, θα παρατεθούν φωτογραφίες από κομμάτια κώδικα, που δείχνουν, με πόσο μικρό πλήθος εντολών μεταφέρθηκαν όλες οι πληροφορίες από τις τρεις λίστες στα αντίστοιχα τρία DataFrames, αλλά και φωτογραφίες με την τελική μορφή αυτών.

4.3.2 DATAFRAME ΤΩΝ ΚΟΜΒΩΝ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ

Ακολουθεί κομμάτι κώδικα μετατροπής της λίστας των κόμβων σε DataFrame:

```
def create_nodes_df(node_list):
    nodes_df = pd.DataFrame.from_records([node.to_dict() for node in node_list])
    nodes_df['Size'] = nodes_df["Width"] * nodes_df["Height"]

    nodes_df.loc[nodes_df['Type'] == 'Terminal', 'Coordinate_x_max'] = (
        nodes_df['Coordinate_x_min'])
    nodes_df.loc[nodes_df['Type'] == 'Non_Terminal', 'Coordinate_x_max'] = (
        nodes_df['Coordinate_x_min'] + nodes_df['Width'])

    nodes_df.loc[nodes_df['Type'] == 'Terminal', 'Coordinate_y_max'] = (
        nodes_df['Coordinate_y_min'])
    nodes_df.loc[nodes_df['Type'] == 'Non_Terminal', 'Coordinate_y_max'] = (
        nodes_df['Coordinate_y_min'] + nodes_df['Height'])

    nodes_df = nodes_df.astype({"Coordinate_x_max": int,
                                "Coordinate_y_max": int})

    return nodes_df
```

Εικόνα 21: Δημιουργία DataFrame Κόμβων

Με την εκτέλεση της πρώτης γραμμής της συνάρτησης `create_nodes_df`, μετατρέπονται όλα τα δεδομένα, που υπήρχαν στη λίστα `nodes`, σε `DataFrame`. Στην πορεία, με τη βοήθεια του παραγόμενου `DataFrame`, εκτελούμε επιπλέον υπολογισμούς για να προκύψουν νέα δεδομένα, όπως το μέγεθος των κόμβων αλλά και οι μέγιστες τιμές των συντεταγμένων `x`, `y` σύμφωνα με τον τύπο τους.

Στον κώδικα παρατηρείται, πως στους κόμβους τύπου `terminal`, αναθέεται η ίδια τιμή τόσο στη μέγιστη όσο και στην ελάχιστη συντεταγμένη `x` και `y`. Αυτό προκύπτει διότι, όπως έχει αναφερθεί και παραπάνω, η γραφική αναπαράστασή τους είναι κουκίδα.

Η επόμενη φωτογραφία, είναι το αποτέλεσμα εκτέλεσης της παραπάνω συνάρτησης:

```

Display Nodes Dataframe:
   Node_name  Width  Height  Type Row_number  Nets  Coordinate_x_min  Coordinate_y_min  Size  Coordinate_x_max  Coordinate_y_max
0          a0     16     16  Non_Terminal  row43  [net12275, net12931]  1402           688  256  1418           704
1          a1     14     16  Non_Terminal  row5   [net12128, net12129]  31            80  224    45            96
2          a2      8     16  Non_Terminal  row89  [net6908, net8139, net13798]  433  1424  128    441           1440
3          a3      6     16  Non_Terminal  row59  [net11207, net11239, net11329, net13292]  1414  944  96    1420           960
4          a4      6     16  Non_Terminal  row64  [net2528, net14102]  952  1024  96    958           1040
...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
12747      p242     1     1  Terminal  None  [net88]  1582  1428  1  1582           1428
12748      p243     1     1  Terminal  None  [net187]  1582  1452  1  1582           1452
12749      p244     1     1  Terminal  None  [net66]  1582  1476  1  1582           1476
12750      p245     1     1  Terminal  None  [net2]  1582  1500  1  1582           1500
12751      p246     1     1  Terminal  None  [net73]  1582  1524  1  1582           1524

[12752 rows x 11 columns]

```

Εικόνα 22: Απεικόνιση DataFrame Κόμβων

4.3.3 DATAFRAME ΤΩΝ NET ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ

Ακολουθεί κομμάτι κώδικα μετατροπής της λίστας των Net σε DataFrame:

```
def create_nets_df(net_list, nodes_df):
    nets_df = pd.DataFrame.from_records([net.to_dict() for net in net_list])

    calculate_net_hpw(nets_df)
    calculate_net_size(nets_df)
    nets_df = nets_df.astype({"Half_Perimeter_Wirelength": int, "Net_Size": int})

    return nets_df
```

Εικόνα 23: Δημιουργία DataFrame Net

```
def calculate_net_hpw(nets_df):
    nets_df['Half_Perimeter_Wirelength'] = ((nets_df['x_max'] - nets_df['x_min'])
                                             + (nets_df['y_max'] - nets_df['y_min']))

def calculate_net_size(nets_df):
    nets_df['Net_Size'] = ((nets_df['x_max'] - nets_df['x_min'])
                           * (nets_df['y_max'] - nets_df['y_min']))
```

Εικόνα 24: Υπολογισμός καλωδίου & μεγέθους των Net

Όμοια, με την μετατροπή της λίστας των κόμβων σε DataFrame και στην περίπτωση αυτή, στην πρώτη γραμμή της συνάρτησης create_nets_df, η λίστα των Net μετατρέπεται σε DataFrame, ενώ στη συνέχεια μετράται το καλώδιο διασύνδεσης για το εκάστοτε Net και το μέγεθος τους. Το τελευταίο, υπολογίζεται σύμφωνα με το εμβαδόν, που προκύπτει από τις συντεταγμένες των τεσσάρων γωνιών του.

Η παρακάτω φωτογραφία, αποτελεί το εξαγόμενο αποτέλεσμα εκτέλεσης της παραπάνω συνάρτησης:

```
Display Nets DataFrame:
   Net_name  Nodes  Rows Internal_nodes External_nodes  x_min  x_max  y_min  y_max  Half_Perimeter_Wirelength  Net_Size
0    net0  [p198, a8117]  [row13]  []  [a8117, p198]  821  1582  208  368  921  121760
1    net1  [a3044, p98]  [row38]  []  [a3044, p98]  943  1220  -33  624  934  181989
2    net2  [a3767, p245]  [row1]  []  [p245, a3767]  1084  1582  16  1500  1982  739032
3    net3  [a4992, p231]  [row18]  []  [a4992, p231]  1386  1582  288  1163  1071  171500
4    net4  [a6837, p9]  [row18]  []  [p9, a6837]  252  870  288  1569  1899  791658
...
14106 net14106  [a2512, a1914]  [row63]  []  [a1914, a2512]  1067  1081  1008  1024  30  224
14107 net14107  [a5047, a1252]  [row63]  []  [a5047, a1252]  1081  1095  1008  1024  30  224
14108 net14108  [a3372, a11879]  [row65]  []  [a3372, a11879]  1055  1069  1040  1056  30  224
14109 net14109  [a2906, a7857]  [row60]  []  [a2906, a7857]  694  708  960  976  30  224
14110 net14110  [a2263, a12324]  [row23, row24]  []  [a12324, a2263]  335  398  368  400  95  2616

[14111 rows x 11 columns]
```

Εικόνα 25: Απεικόνιση DataFrame Net

4.3.4 DATAFRAME ΤΩΝ ΣΕΙΡΩΝ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ

Στη συνέχεια παρατίθεται κομμάτι κώδικα μετατροπής της λίστας των σειρών σε DataFrame:

```
def create_rows_df(row_list, nodes_df):
    rows_df = pd.DataFrame.from_records([row.to_dict() for row in row_list])

    rows_df['Width'] = rows_df['Coordinate_x_max'] - rows_df['Coordinate_x_min']
    rows_df['Height'] = rows_df['Coordinate_y_max'] - rows_df['Coordinate_y_min']
    rows_df['Row_area'] = rows_df['Width'] * rows_df['Height']

    row_density(nodes_df, rows_df)
    rows_df = rows_df.astype({"Nodes_area": int})

    return rows_df
```

Εικόνα 26: Δημιουργία DataFrame Σειρών

Όπως και στις 2 προηγούμενες περιπτώσεις, στην πρώτη γραμμή δημιουργείται το DataFrame των σειρών και έπειτα, βρίσκεται το πλάτος και ύψος αυτών αλλά και η περιοχή, που καταλαμβάνει κάθε μία από αυτές μέσα στο τσιπ. Τέλος, εντοπίζοντας τους κόμβους, που είναι τοποθετημένοι στην κάθε γραμμή, βρίσκεται και η πυκνότητα αυτής.

Οι ακόλουθες δύο φωτογραφίες, είναι το αποτέλεσμα εκτέλεσης της παραπάνω συνάρτησης:

```

Display Rows Dataframe:
  Row_name  Cells  Nets  Coordinate_x_min  Coordinate_x_max  Coordinate_y_min  Coordinate_y_max
0  row0  [a878, a1238, a3427, a3728, a4132, a4244, a468...  [net12569, net12024, net12026, net12531, net44...  0  1550  0  16
1  row1  [a98, a292, a427, a754, a768, a882, a1495, a15...  [net13410, net12235, net12236, net651, net8210...  0  1550  16  32
2  row2  [a37, a190, a342, a437, a496, a658, a1013, a11...  [net1985, net3413, net12485, net12486, net644, ...  0  1550  32  48
3  row3  [a205, a271, a291, a351, a430, a534, a859, a91...  [net3416, net13420, net150, net13001, net319, ...  0  1550  48  64
4  row4  [a87, a462, a587, a690, a739, a838, a1070, a11...  [net1985, net12401, net1986, net12486, net644, ...  0  1550  64  80
...  ...  ...  ...  ...  ...  ...  ...
91  row91  [a99, a274, a600, a449, a666, a669, a1054, a11...  [net10050, net9969, net9973, net4514, net9974, ...  0  1550  1456  1472
92  row92  [a132, a138, a191, a408, a414, a424, a448, a76...  [net10050, net9972, net9973, net10057, net4515...  0  1550  1472  1488
93  row93  [a71, a120, a235, a494, a618, a645, a789, a871...  [net9965, net10050, net9968, net9972, net10057...  0  1550  1488  1504
94  row94  [a366, a373, a375, a388, a612, a853, a983, a12...  [net13903, net9965, net9966, net9967, net7196, ...  0  1550  1504  1520
95  row95  [a876, a982, a1163, a1181, a1297, a1437, a1607...  [net9965, net9966, net9967, net9968, net5358, ...  0  1550  1520  1536

[96 rows x 13 columns]

```

Εικόνα 27: Απεικόνιση DataFrame Σειρών (μέρος 1)

```

Width  Height  Row_area  Nodes_area  Density(%)
1550   16      24800      7392      29.806452
1550   16      24800     13088     52.774194
1550   16      24800     17728     71.483871
1550   16      24800     21120     85.161290
1550   16      24800     20992     84.645161
...    ...    ...    ...    ...
1550   16      24800     23360     94.193548
1550   16      24800     21504     86.709677
1550   16      24800     20032     80.774194
1550   16      24800     17472     70.451613
1550   16      24800      9280     37.419355

```

Εικόνα 28: Απεικόνιση DataFrame Σειρών (μέρος 2)

4.3.5 DATAFRAME ΟΛΟΚΛΗΡΟΥ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ

Αφού έχουν δημιουργηθεί τα τρία ανωτέρω DataFrames, με την χρήση αυτών αντλούμε τα απαραίτητα δεδομένα, ώστε να φτιαχτεί και το αντίστοιχο, για ολόκληρο το κύκλωμα.

Στην ερχόμενη φωτογραφία, παραθέεται το αποτέλεσμα από την χρήση των τριών DataFrame, για την παραγωγή ενός τέταρτου, βάση αυτών:

```
Display Designs DataFrame:
```

	Number_of_cells	Number_of_terminals	Number_of_nets	Number_of_rows	Width	Height	Total_Area	Total_Cell_Area	Half_Perimeter_Wirelength	Density(%)
0	12752	246	14111	96	1550	1536	2380800	2142166	2073455	89.976731

Εικόνα 29: Απεικόνιση DataFrame Κυκλώματος

4.3.6 ΑΠΟΘΗΚΕΥΣΗ ΤΩΝ DATAFRAME ΣΕ ΑΡΧΕΙΟ

Όλα τα παραπάνω μπορούν πολύ εύκολα, να αποθηκευτούν σε ένα excel αρχείο, ή σε οποιαδήποτε άλλη μορφή, προκειμένου να γίνει η ανάγνωσή τους και σε ένα πιο οικείο περιβάλλον, για το ευρύ κοινό.

```
with pd.ExcelWriter('output.xlsx') as writer:  
    nodes_df.to_excel(writer, sheet_name='nodes_df')  
    nets_df.to_excel(writer, sheet_name='nets_df')  
    rows_df.to_excel(writer, sheet_name='rows_df')  
    design_df.to_excel(writer, sheet_name='design_df')
```

Εικόνα 30: Αποθήκευση 4 DataFrame σε Excel αρχείο

Κάτωθι, παρουσιάζεται το αποτέλεσμα της εκτέλεσης του κώδικα της Εικόνας 30:

	A	B	C	D	E	F	G	H	I	J	K	L
1		Node_name	Width	Height	Type	Row_number	Nets	Coordinate_x_min	Coordinate_y_min	Size	Coordinate_x_max	Coordinate_y_max
2	0	a0	16	16	Non_Terminal	row43	['net12275','net12931']	1402	688	256	1418	704
3	1	a1	14	16	Non_Terminal	row5	['net12128','net12129']	31	80	224	45	96
4	2	a2	8	16	Non_Terminal	row89	['net6908','net8139','net13	433	1424	128	441	1440
5	3	a3	6	16	Non_Terminal	row59	['net11207','net11239','net	1414	944	96	1420	960
6	4	a4	6	16	Non_Terminal	row64	['net2528','net14102']	952	1024	96	958	1040
7	5	a5	8	16	Non_Terminal	row62	['net2665','net2843','net34	880	992	128	888	1008
8	6	a6	8	16	Non_Terminal	row77	['net2454','net8728','net89	1068	1232	128	1076	1248
9	7	a7	8	16	Non_Terminal	row65	['net5315','net5316','net69	182	1040	128	190	1056
10	8	a8	8	16	Non_Terminal	row49	['net2818','net2935','net29	984	128	128	992	800
11	9	a9	12	16	Non_Terminal	row13	['net11898','net11990','net	484	208	192	496	224
12	10	a10	42	16	Non_Terminal	row20	['net846','net936','net939	540	320	672	582	336
13	11	a11	8	16	Non_Terminal	row58	['net4241','net4466','net44	476	928	128	484	944
14	12	a12	12	16	Non_Terminal	row86	['net4819','net5848','net59	278	1376	192	290	1392
15	13	a13	8	16	Non_Terminal	row82	['net10300','net10346','net	920	1312	128	928	1328
16	14	a14	6	16	Non_Terminal	row61	['net4391','net4486','net11	268	976	96	274	992
17	15	a15	14	16	Non_Terminal	row83	['net4514','net4516','net47	90	1328	224	104	1344
18	16	a16	8	16	Non_Terminal	row49	['net4917','net7264','net76	66	784	128	74	800
19	17	a17	8	16	Non_Terminal	row78	['net4317','net4320','net45	115	1248	128	123	1264
20	18	a18	24	16	Non_Terminal	row43	['net346','net1725','net172	1138	688	384	1162	704
21	19	a19	4	16	Non_Terminal	row21	['net13465','net13466']	772	336	64	776	352
22	20	a20	6	16	Non_Terminal	row40	['net10549','net13919']	1283	640	96	1289	656

Εικόνα 31: Απόσπασμα από το Nodes Excel αρχείο

1	A	B	C	D	E	F	G	H	I	J	K	L	M
	Net_name	Nodes	Rows	Internal_nodes	External_nodes	x_min	x_max	y_min	y_max	Half_Perimeter	Wirelength	Net_Size	Num_of_nodes
6550	6548	net6548	['a3336','a5515']	['row71','row70']	[]	['a5515','a3336']	282	296	1120	1152	46	448	2
6551	6549	net6549	['a9375','a11713']	['row69','row73']	['a9375','a7589']	['a11713','a12094','a	184	304	1040	1184	264	17280	5
6552	6550	net6550	['a11234','a3368']	['row52','row75']	['a11234','a11122']	['a1559','a3368','a11	86	188	800	1216	518	42432	6
6553	6551	net6551	['a5285','a492']	['row71','row70']	[]	['a5285','a492']	296	304	1120	1152	40	256	2
6554	6552	net6552	['a842','a3620']	['row63','row62']	[]	['a3620','a842']	30	40	992	1024	42	320	2
6555	6553	net6553	['a8083','a5464']	['row62']	[]	['a5464','a8083']	36	48	992	1008	28	192	2
6556	6554	net6554	['a2759','a4448','ε	['row76']	[]	['a3991','a4448','a27	370	406	1216	1232	52	576	5
6557	6555	net6555	['a8386','a1257','ε	['row72','row73']	[]	['a1257','a8386','a83	288	335	1152	1184	79	1504	3
6558	6556	net6556	['a2042','a8320']	['row72']	[]	['a8320','a2042']	323	371	1152	1168	64	768	2
6559	6557	net6557	['a8245','a11166']	['row74']	[]	['a8245','a11166']	276	284	1184	1200	24	128	2
6560	6558	net6558	['a5132','a6553']	['row48','row59']	[]	['a6553','a5132']	232	268	768	960	228	6912	2
6561	6559	net6559	['a12296','a8906']	['row57']	[]	['a12296','a8906']	64	84	912	928	36	320	2
6562	6560	net6560	['a9010','a9876','ε	['row52','row54']	['a7489']	['a9876','a9010']	338	406	832	880	116	3264	3
6563	6561	net6561	['a717','a9756']	['row74']	[]	['a717','a9756']	369	379	1184	1200	26	160	2
6564	6562	net6562	['a721','a5443','aε	['row53','row62']	['a721','a11512','a	['a5443','a2991','a31	364	446	528	1008	562	39360	8
6565	6563	net6563	['a3998','a520','a	['row57','row61']	['a3998','a7066','a	['a520','a426','a8596	324	414	880	992	202	10080	9
6566	6564	net6564	['a3410','a8666','ε	['row75','row74']	[]	['a10411','a3410','a8	396	432	1184	1216	68	1152	5
6567	6565	net6565	['a4637','a5168','ε	['row65','row64']	[]	['a4637','a5174','a51	44	70	1024	1056	58	832	3
6568	6566	net6566	['a3702','a5249']	['row53']	[]	['a5249','a3702']	221	249	848	864	44	448	2
6569	6567	net6567	['a8161','a3547','ε	['row51','row48']	[]	['a8161','a3547','a15	343	358	768	832	79	960	3
6570	6568	net6568	['a7678','a8073']	['row54']	[]	['a8073','a7678']	60	84	864	880	40	384	2

Εικόνα 32: Απόσπασμα από το Nets Excel αρχείο

	A	B	C	D	E	F	G	H	I	J	K	L	M
		Row_name	Cells	Nets	Coordinate_x_min	Coordinate_x_max	Coordinate_y_min	Coordinate_y_max	Width	Height	Row_area	Nodes_area	Density(%)
2	0	row0	['a878', 'a1238', 'a3	['net1291', 'net12211'	0	1550	0	16	1550	16	24800	7392	29.80645161
3	1	row1	['a90', 'a292', 'a427	['net1295', 'net11963'	0	1550	16	32	1550	16	24800	13088	52.77419355
4	2	row2	['a37', 'a190', 'a342	['net12209', 'net1291'	0	1550	32	48	1550	16	24800	17728	71.48387097
5	3	row3	['a205', 'a271', 'a25	['net11959', 'net1288'	0	1550	48	64	1550	16	24800	21120	85.16129032
6	4	row4	['a87', 'a462', 'a587	['net12883', 'net2048'	0	1550	64	80	1550	16	24800	20992	84.64516129
7	5	row5	['a1', 'a263', 'a316',	['net11957', 'net2047'	0	1550	80	96	1550	16	24800	21824	88
8	6	row6	['a109', 'a273', 'a27	['net12883', 'net1314'	0	1550	96	112	1550	16	24800	23008	92.77419355
9	7	row7	['a23', 'a257', 'a261	['net1543', 'net11959'	0	1550	112	128	1550	16	24800	22144	89.29032258
10	8	row8	['a72', 'a153', 'a485	['net11959', 'net1313'	0	1550	128	144	1550	16	24800	20064	80.90322581
11	9	row9	['a328', 'a521', 'a54	['net1284', 'net12883',	0	1550	144	160	1550	16	24800	22688	91.48387097
12	10	row10	['a158', 'a549', 'a66	['net1543', 'net12883'	0	1550	160	176	1550	16	24800	20224	81.5483871
13	11	row11	['a267', 'a339', 'a46	['net1284', 'net12883',	0	1550	176	192	1550	16	24800	22112	89.16129032
14	12	row12	['a47', 'a516', 'a812	['net12460', 'net285',	0	1550	192	208	1550	16	24800	22816	92
15	13	row13	['a9', 'a26', 'a141',	['net536', 'net285', 'ne	0	1550	208	224	1550	16	24800	20608	83.09677419
16	14	row14	['a25', 'a338', 'a410	['net12043', 'net537',	0	1550	224	240	1550	16	24800	20960	84.51612903
17	15	row15	['a32', 'a88', 'a209',	['net5071', 'net537', 'r	0	1550	240	256	1550	16	24800	23264	93.80645161
18	16	row16	['a282', 'a399', 'a44	['net12043', 'net5072'	0	1550	256	272	1550	16	24800	22912	92.38709677
19	17	row17	['a346', 'a774', 'a85	['net2383', 'net1969',	0	1550	272	288	1550	16	24800	23232	93.67741935
20	18	row18	['a104', 'a119', 'a27	['net536', 'net2383', 'r	0	1550	288	304	1550	16	24800	24096	97.16129032
21	19	row19	['a113', 'a216', 'a35	['net1964', 'net286', 'r	0	1550	304	320	1550	16	24800	22720	91.61290323
22	20	row20	['a10', 'a61', 'a124',	['net1963', 'net2384',	0	1550	320	336	1550	16	24800	23040	92.90322581

Εικόνα 33: Απόσπασμα από το Rows Excel αρχείο

	A	B	C	D	E	F	G	H	I	J	K	
		Number_of_cells	Number_of_terminals	Number_of_nets	Number_of_rows	Width	Height	Total_Area	Total_Cell_Area	Half_Perimeter	Wirelength	Density(%)
1	0	12752	246	14111	96	1550	1536	2380800	2142166		2073455	89.97673051

Εικόνα 34: Απόσπασμα από το Design Excel αρχείο

Απ' όλα τα παραπάνω, εύκολα ανάγεται το συμπέρασμα ότι τα DataFrames αποτελούν μία ευανάγνωστη δομή ενώ ταυτόχρονα, είναι ικανά να επιτυγχάνουν την διεκπεραίωση πλήθους ενεργειών με ελάχιστες γραμμές κώδικα.

4.4 ΥΛΟΠΟΙΗΜΕΝΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΩΝ DATAFRAME

4.4.1 ΕΙΣΑΓΩΓΗ

Αφού έχει ολοκληρωθεί η δημιουργία των τεσσάρων DataFrame, με τη αποκλειστική χρήση αυτών, υλοποιήθηκε ένα πλήθος συναρτήσεων για την εξαγωγή διαφόρων αποτελεσμάτων, που αφορούν το κύκλωμα. Οι συναρτήσεις αυτές χωρίζονται σε πέντε κατηγορίες, όπου κάθε μία ξεχωριστά αφορά τους κόμβους, τα net, τις σειρές, το κύκλωμα και τέλος, συναρτήσεις γραφικής απεικόνισης στατιστικών σχετικά με το τελευταίο.

4.4.2 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΟΥΣ ΚΟΜΒΟΥΣ

Με την χρήση του DataFrame, που περιγράφει τους κόμβους, υλοποιήθηκαν οι κατάλληλες πέντε συναρτήσεις, όπου βρίσκουν το πλήθος των κελιών αλλά και το μεγαλύτερο, μικρότερο και μέσο μέγεθος αυτών. Με την εύρεση των παραπάνω, ακολουθεί εμφάνιση των κελιών, που έχουν μέγεθος ίσο με αυτό του μεγαλύτερου αλλά και του μικρότερου αντίστοιχα, στην κάθε συνάρτηση ξεχωριστά. Τέλος, για τα I/O pins βρέθηκε το πλήθος τους, ενώ όσον αφορά τα μεγέθη τους είναι σε όλα μοναδιαία.

Στη συνέχεια εμφανίζονται το εξαγόμενα αποτελέσματα των συναρτήσεων:

```
Non Terminal nodes: 12506
Maximum Non Terminal Node size = 704
- Non Terminal Node(s) with max size: ['a22', 'a23', 'a26']

Minimum Non Terminal Node size = 32
- Non Terminal Node(s) with min size: ['a21', 'a31', 'a34']

Mean size of Non Terminal Node(s): 171.27
Terminals nodes: 246
```

Εικόνα 35: Αποτελέσματα των συναρτήσεων των κόμβων

4.4.3 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΑ NET

Οι συναρτήσεις, που αφορούν τα Net, πραγματοποιήθηκαν με αντίστοιχο τρόπο με αυτό των κόμβων, με τη μόνη διαφορά ότι αυτά, όσον αφορά το μέγεθός τους χωρίστηκαν σε δύο κατηγορίες, δηλαδή σύμφωνα τόσο με το πλήθος των κόμβων που ανήκουν στο καθένα όσο και με το μέγεθός τους ως εμβαδόν. Συνολικά υλοποιήθηκαν επτά συναρτήσεις, οι οποίες υπολογίζουν το συνολικό πλήθος των Net και σύμφωνα

με την κατηγοριοποίηση, που αναφέρθηκε προηγουμένως, βρίσκεται το μεγαλύτερο, το μικρότερο και το μέσο μέγεθος αυτών, σύμφωνα με τον αριθμό των κόμβων, από τους οποίους απαρτίζονται. Στην κάθε περίπτωση εμφανίζονται και τα Net που έχουν αυτό το πλήθος.

Αντιστοίχως, δημιουργήθηκαν και οι άλλες τρεις συναρτήσεις, που εντοπίζουν το μικρότερο, μεγαλύτερο αλλά και μέσο μέγεθος του καθενός από αυτά και παραθέτονται πάλι τα ονόματα των net, που έχουν το αντίστοιχο μέγεθος.

Ακολουθώς, εμφανίζονται οι υπολογισμοί των συναρτήσεων:

```
Number of nets: 14111

Maximum number of cells in a net: 42
- Biggest net(s): ['net4469']

Minimum number of cells in a net: 2
- Smallest net(s): ['net0', 'net1', 'net2', 'net3']

Mean number of cell(s) on each net: 3.58

Maximum Net Size: 1543580
- Biggest net(s): ['net177']

Minimum Net Size: 96
- Smallest net(s): ['net3300', 'net3301', 'net3302']

Mean Net Size: 13994.122528523847
```

Εικόνα 36: Αποτελέσματα των συναρτήσεων των Net

4.4.4 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΙΣ ΣΕΙΡΕΣ

Όπως και προηγουμένως, δημιουργήθηκαν τέσσερις συναρτήσεις, που βρίσκουν το πλήθος των σειρών του κυκλώματος αλλά και το μεγαλύτερο, μικρότερο και μέσο μέγεθος αυτών, σύμφωνα με το πλήθος κόμβων, που είναι τοποθετημένοι μέσα σε

αυτές. Εκτός όμως από αυτά, εντοπίζονται και τα ονόματα των σειρών των δύο πρώτων κατηγοριών.

Εν συνεχεία, εμφανίζονται τα αποτελέσματα των προαναφερθέντων συναρτήσεων:

```
Number of rows: 96

Maximum number of cells in a row: 204
- Biggest row(s): ['row59']

Minimum number of cells in a row: 23
- Smallest row(s): ['row0']

Mean number of cells on each row: 130.27
```

Εικόνα 37: Αποτελέσματα των συναρτήσεων των Σειρών

4.4.5 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΟ ΚΥΚΛΩΜΑ

Τέλος, υλοποιήθηκαν και δύο συναρτήσεις, που αφορούν το κύκλωμα στο σύνολό του, οι οποίες υπολογίζουν το συνολικό μήκος καλωδίου αλλά και την πυκνότητά του, όπως φαίνεται και παρακάτω.

Ακολουθώς, παραθέεται ο κώδικας της συνάρτησης υπολογισμού του καλωδίου διασύνδεσης:

```
def design_df_half_perimeter_wirelength(nets_df):
    design_hpw = nets_df['Half_Perimeter_Wirelength'].sum()

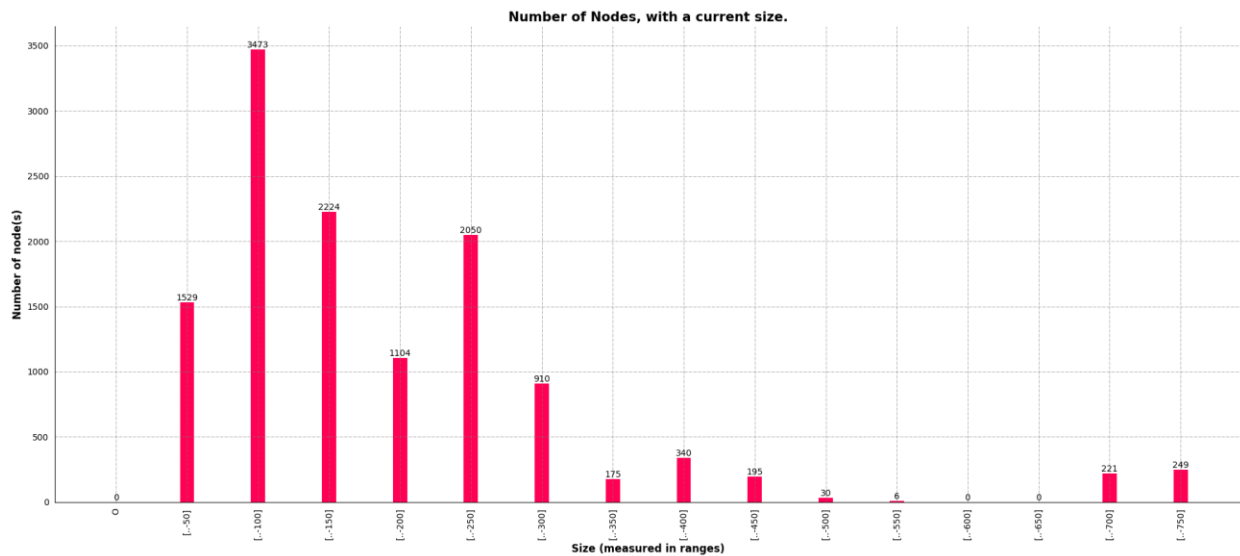
    return design_hpw
```

Εικόνα 38: Υπολογισμός καλωδίου Κυκλώματος

4.4.6 ΓΡΑΦΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ ΜΕ ΧΡΗΣΗ ΤΗΣ MATPLOTLIB

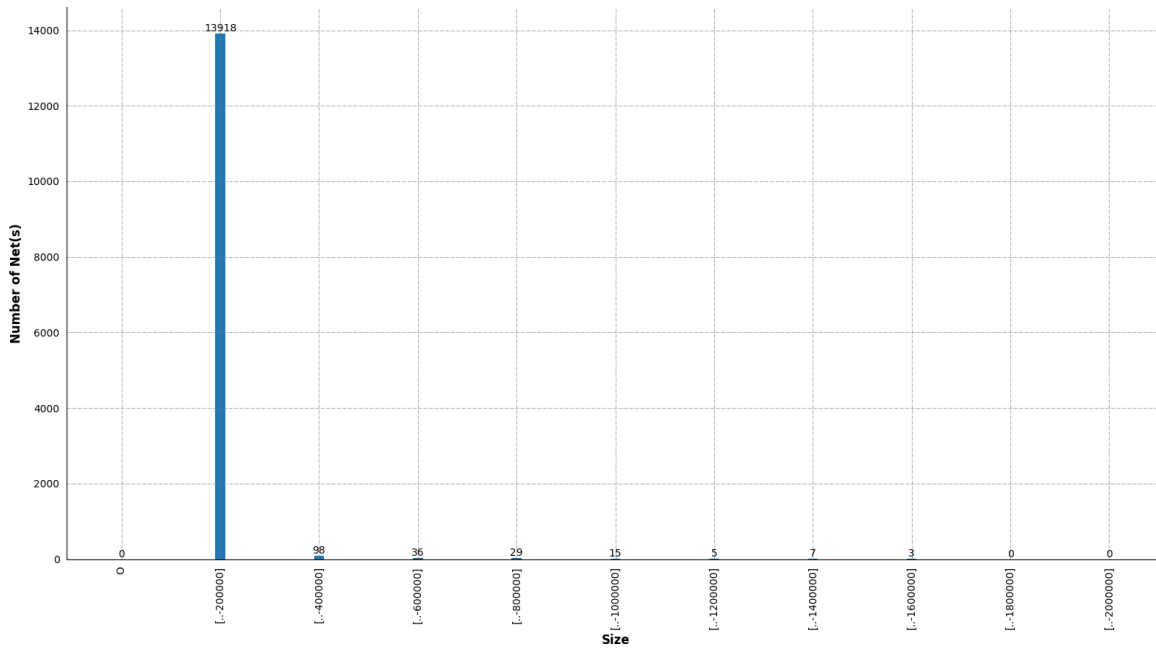
Λόγω του μεγέθους των κυκλωμάτων, που αναλύονται, τα αποτελέσματα έχουν ομαδοποιηθεί ως προς το ζητούμενο κάθε φορά μέγεθος, προκειμένου οι γραφικές απεικονίσεις να είναι ευανάγνωστες.

Η πρώτη συνάρτηση παρουσιάζει τα μεγέθη των κόμβων συναρτήσει του πλήθους, που έχουν το αντίστοιχο μέγεθος:



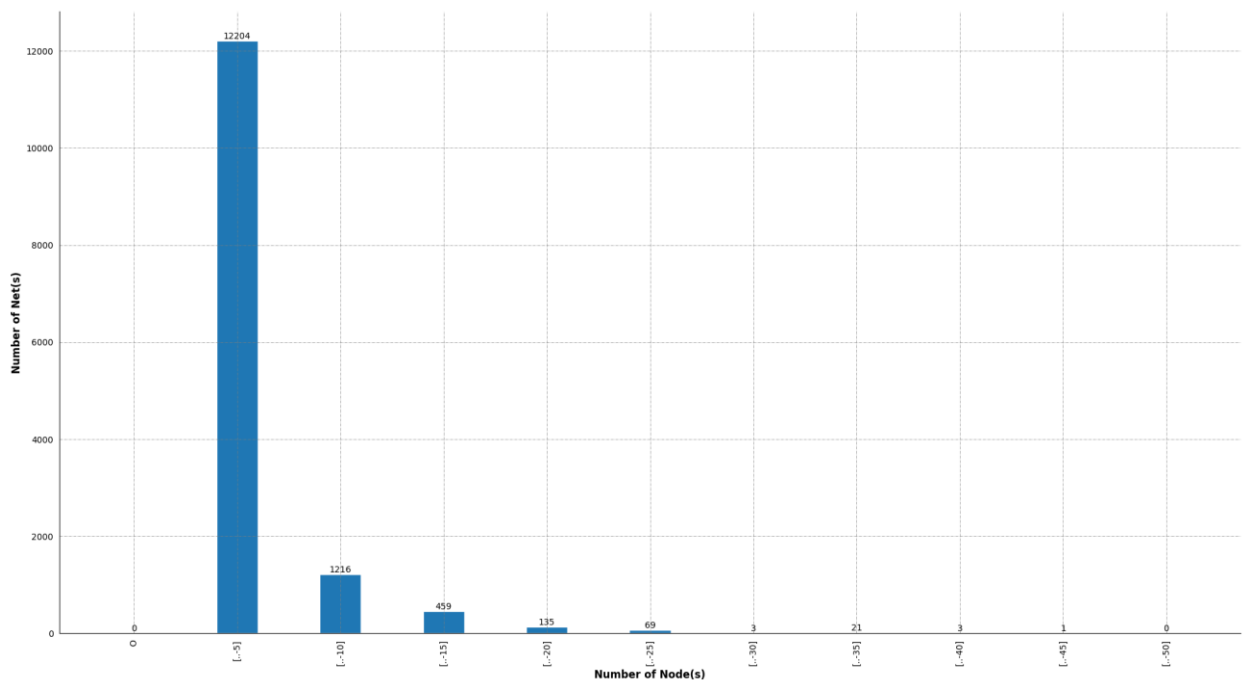
Εικόνα 39: Αριθμός κόμβων, με συγκεκριμένο μέγεθος

Η δεύτερη έχει το ίδιο αποτέλεσμα με την πρώτη, αλλά αυτή τη φορά τα μεγέθη αφορούν τα Net:



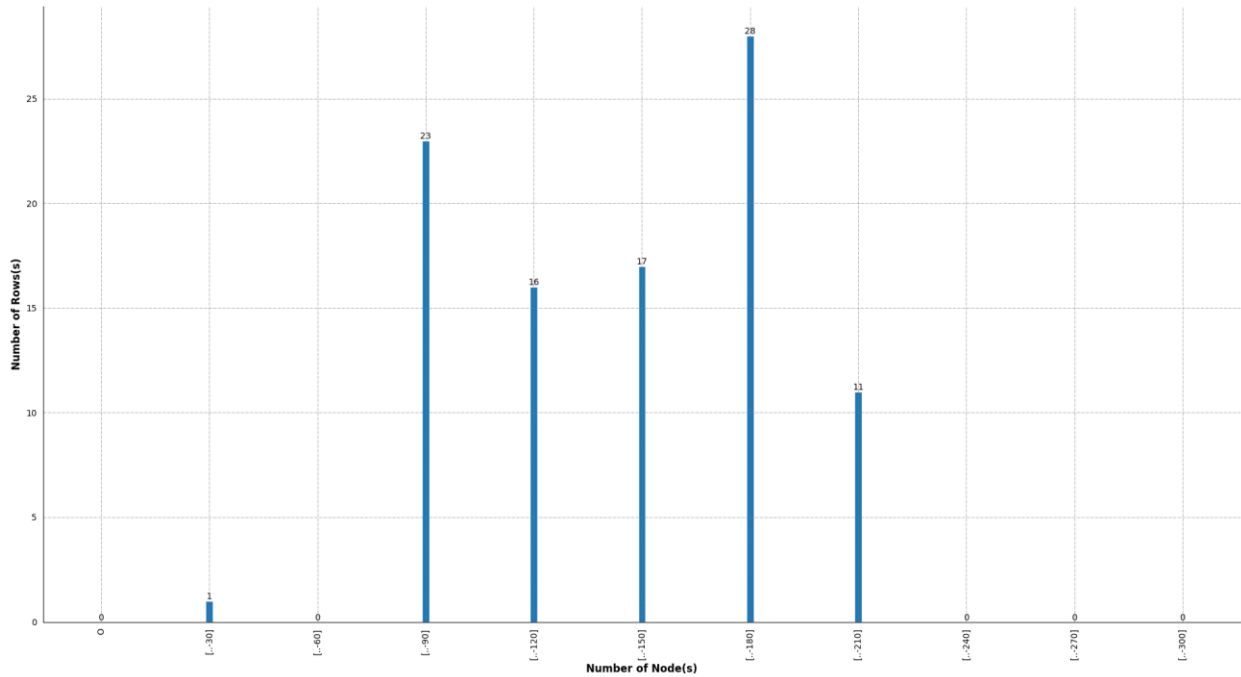
Εικόνα 40: Αριθμός Net, με συγκεκριμένο μέγεθος

Η τρίτη, εμφανίζει το πλήθος των κόμβων και αριθμό Net που έχουν το αντίστοιχο πλήθος:



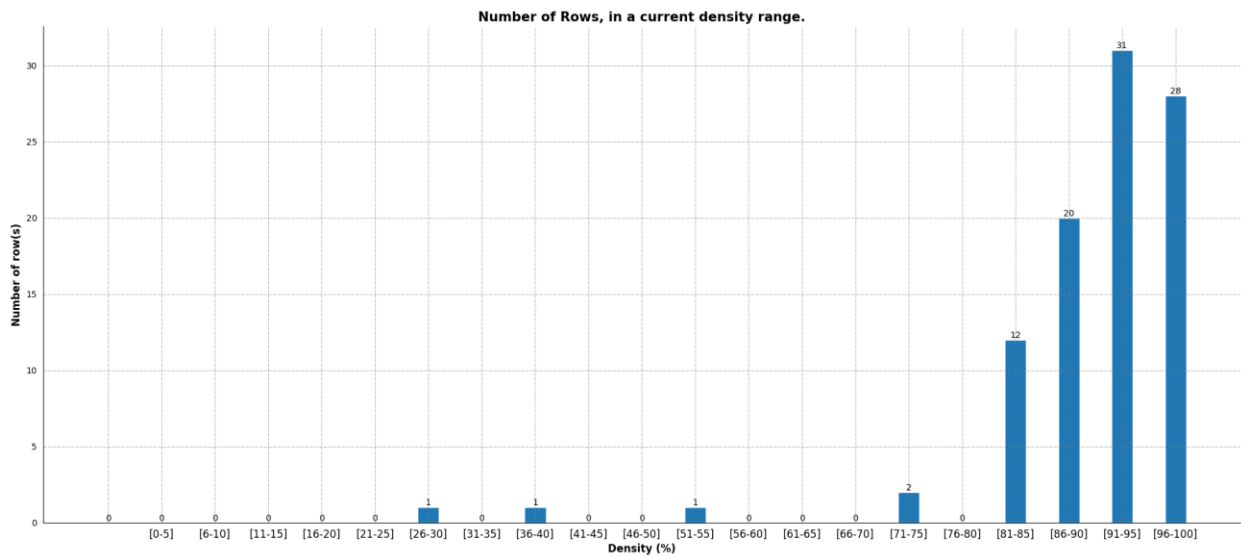
Εικόνα 41: Αριθμός Net, με συγκεκριμένο πλήθος κόμβων

Η τέταρτη, απεικονίζει πλήθη σειρών, συναρτήσει του πλήθους των κόμβων από τους οποίους αυτές αποτελούνται:



Εικόνα 42: Αριθμός σειρών, με συγκεκριμένο πλήθος κόμβων

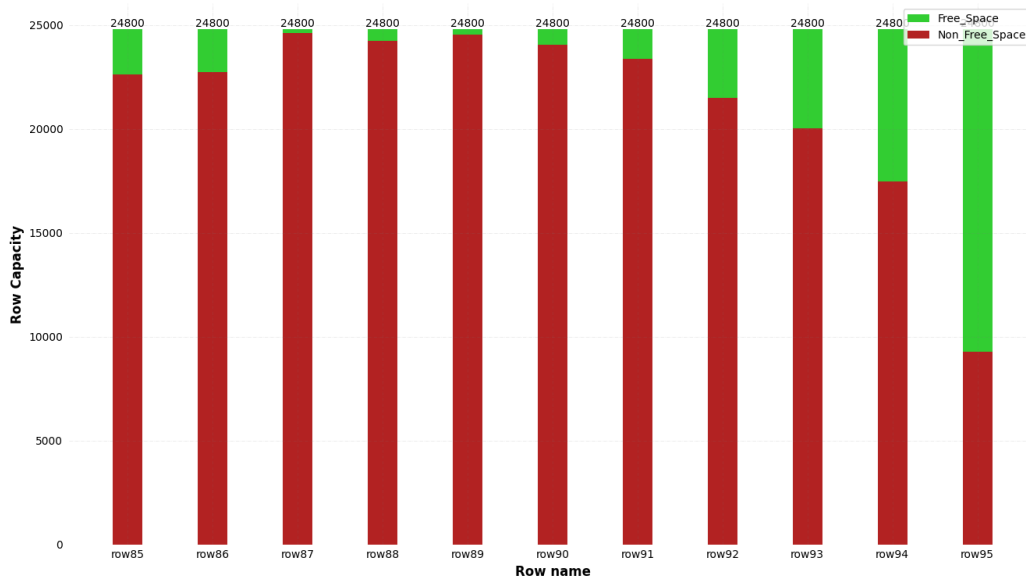
Η πέμπτη, αναπαριστά πλήθος σειρών συναρτήσει των πυκνοτήτων τους:



Εικόνα 43: Αριθμός σειρών, με συγκεκριμένη πυκνότητα

Και τέλος η έκτη, παρουσιάζει την χωρητικότητα της κάθε σειράς. Στη περίπτωση αυτή, ο χρήστης, που εκτελεί τον κώδικα, έχει την επιλογή να δει μέχρι και 15 συνεχόμενες σειρές, από το σύνολό τους. Με πράσινο χρώμα παρουσιάζεται ο ελεύθερος χώρος και με κόκκινο ο δεσμευμένος.

Στην παρακάτω φωτογραφία, εμφανίζονται οι σειρές 85 με 95 του τρέχοντος κυκλώματος:



Εικόνα 44: Απεικόνιση χωρητικότητας των Σειρών

4.5 ΧΡΟΝΟΙ ΕΚΤΕΛΕΣΗΣ

Οι παρακάτω χρόνοι μετρήθηκαν με τη χρήση φορητού υπολογιστή, με επεξεργαστή Intel core I7- 7700HQ, 16GB μνήμη RAM ταχύτητας 2400MHz και τα δεδομένα ήταν αποθηκευμένα σε SSD.

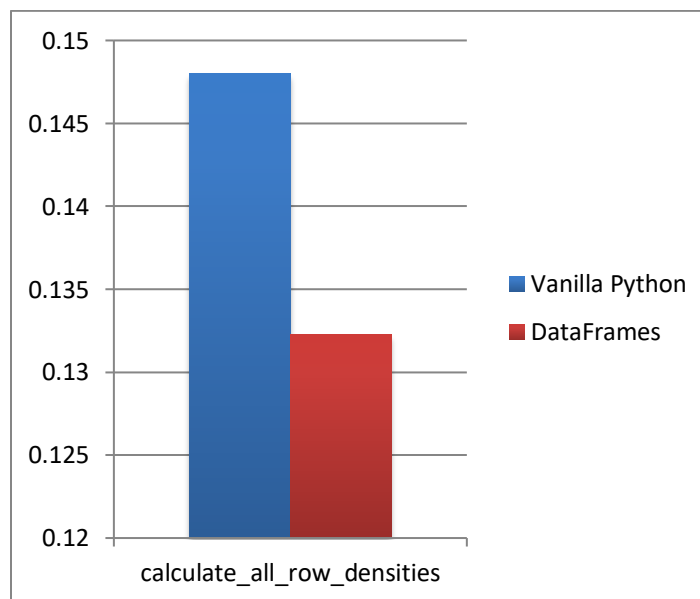
Τα κυκλώματα, πάνω στα οποία έγιναν οι μετρήσεις, είναι διαφόρων μεγεθών. Το **ibm01** είναι το πιο μικρό κύκλωμα, το **ibm05** ένα ενδιάμεσο και τέλος το **ibm18**, το πιο μεγάλο, που είχαμε στη διάθεσή μας.

- **ibm01** (1.9MB): Αποτελείται από 12.752 κόμβους, απ' τους οποίους οι 246 είναι I/O pins, 14.111 nets και 96 σειρές.
- **ibm05** (4.5MB): Αποτελείται από 29.347 κόμβους, απ' τους οποίους οι 1201 είναι I/O pins , 28.446 nets και 139 σειρές.
- **ibm18**(30.8MB): Αποτελείται από 210.613 κόμβους, απ' τους οποίους οι 272 είναι I/O pins , 201.920 nets και 361 σειρές.

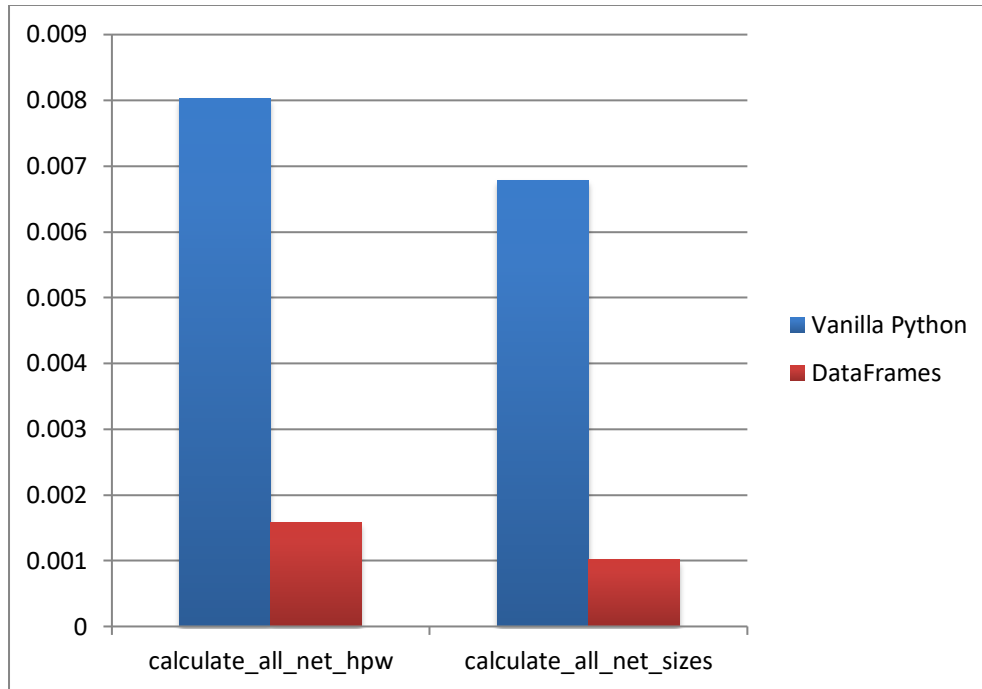
4.5.1 ΓΡΑΦΗΜΑΤΑ ΣΥΓΚΡΙΣΗΣ ΚΟΙΝΩΝ ΣΥΝΑΡΤΗΣΕΩΝ

Ακολουθώς, παραθέτονται σε γράφημα οι χρόνοι εκτέλεσης των συναρτήσεων, που έχουν υλοποιηθεί και με DataFrames αλλά και χωρίς, για τα τρία αυτά κυκλώματα. Όλοι οι χρόνοι είναι μετρημένοι σε δευτερόλεπτα.

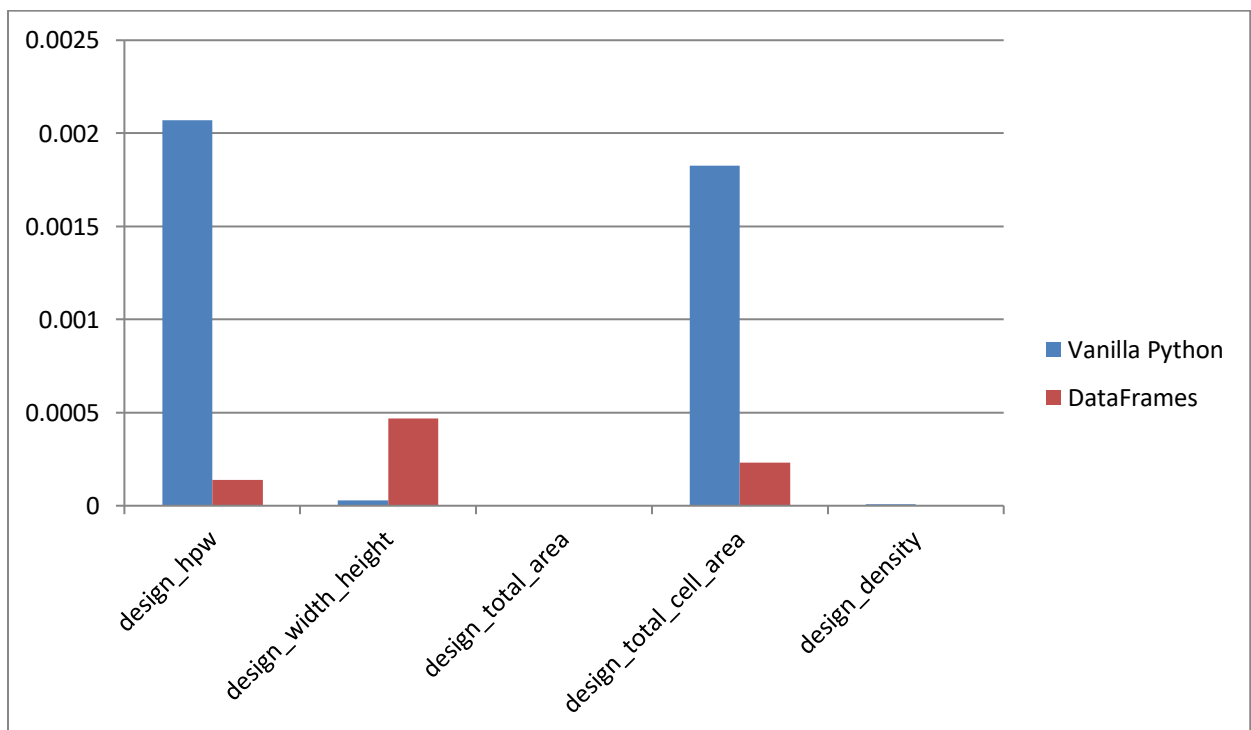
- **ibm01:**



Εικόνα 45: ibm01 Σύγκριση Χρόνων - Μέρος 1

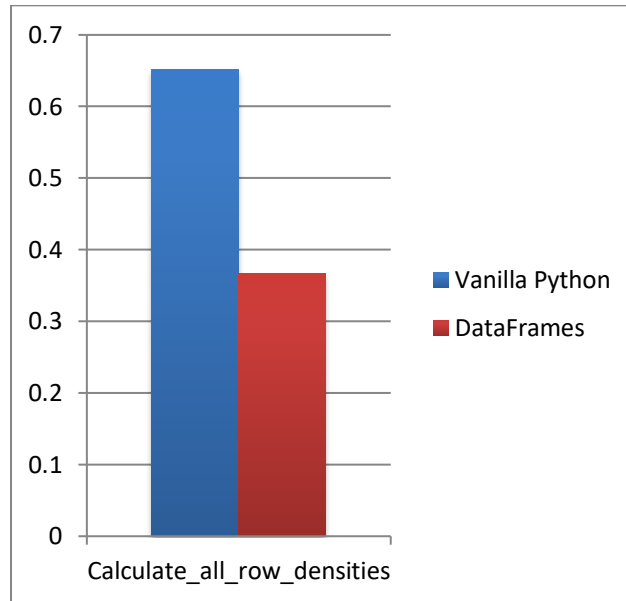


Εικόνα 46: ibm01 Σύγκριση Χρόνων - Μέρος 2

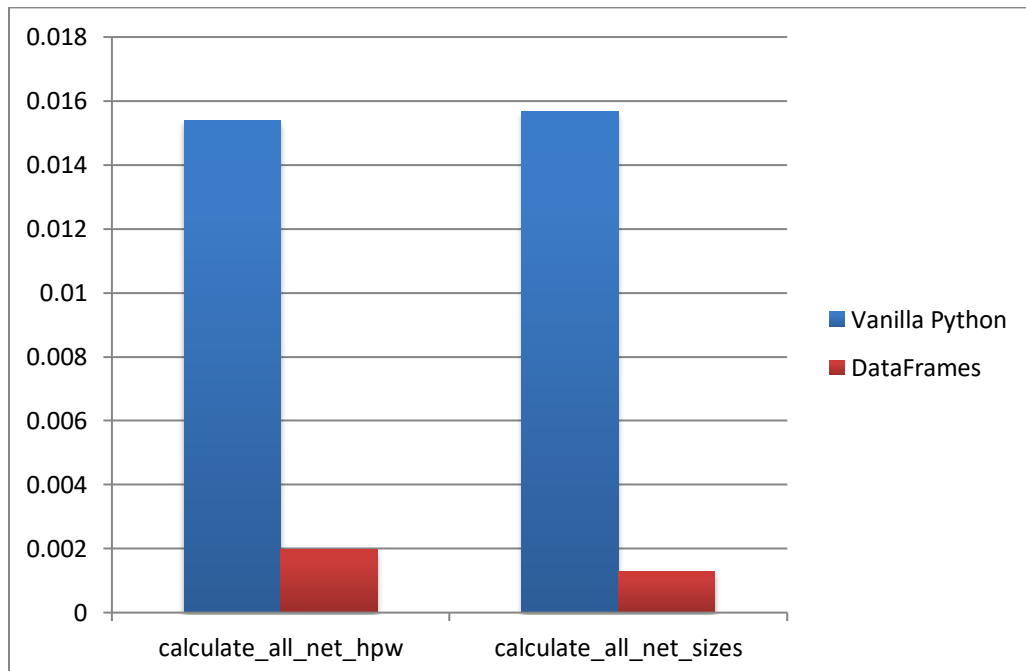


Εικόνα 47: ibm01 Σύγκριση Χρόνων - Μέρος 3

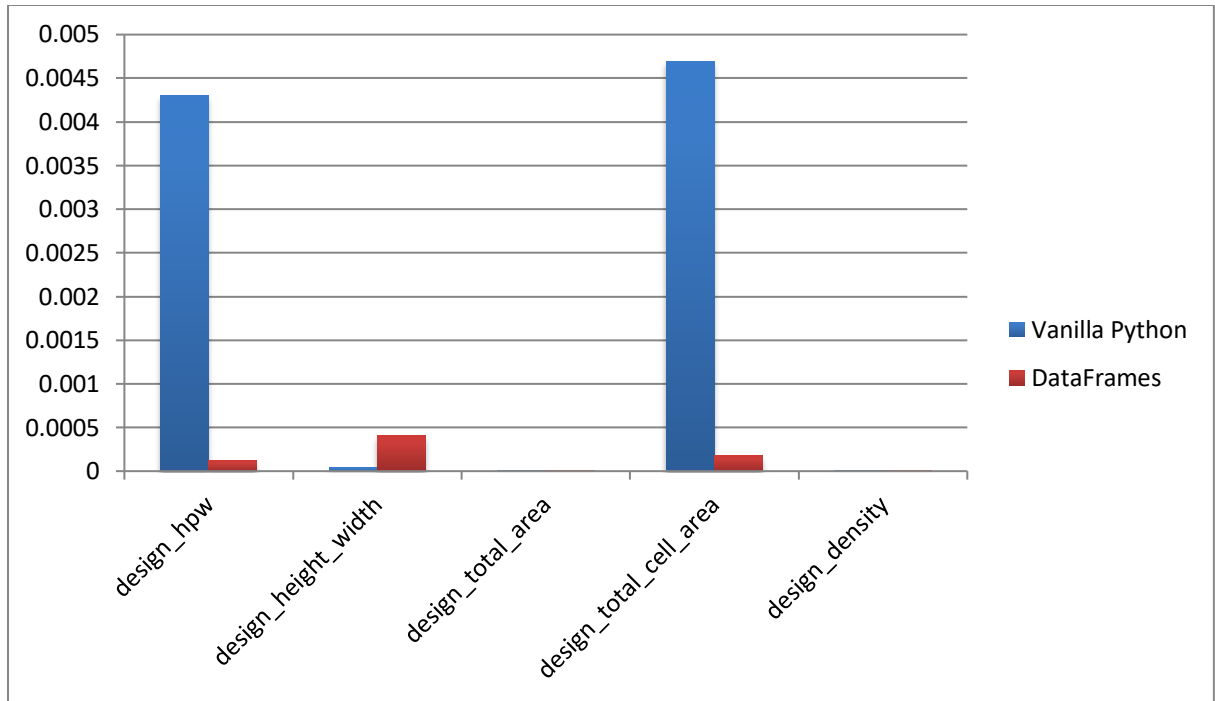
- **ibm05:**



Εικόνα 48: ibm05 Σύγκριση Χρόνων - Μέρος 1

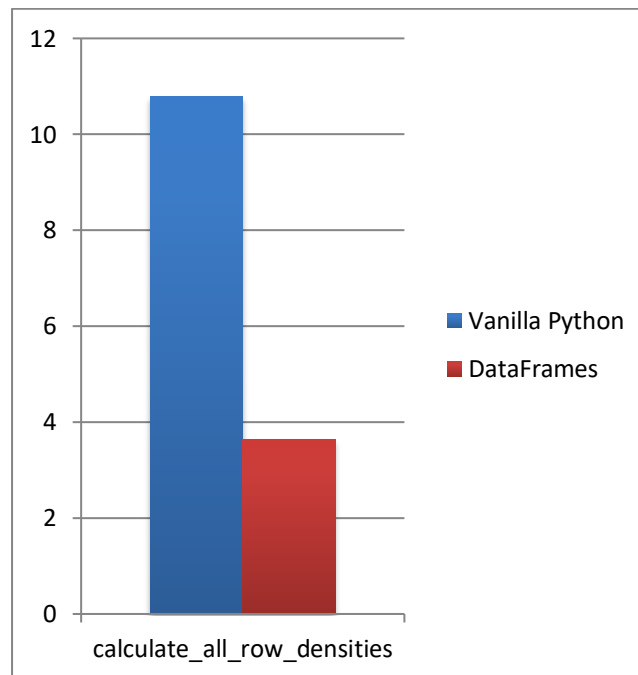


Εικόνα 49: ibm05 Σύγκριση Χρόνων - Μέρος 2

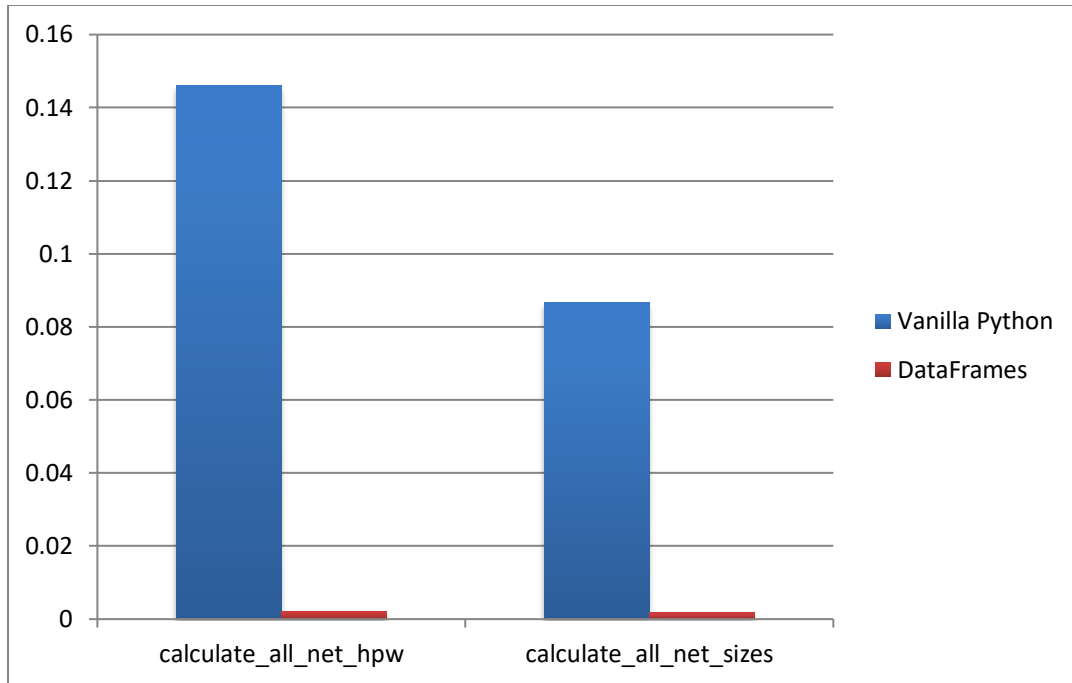


Εικόνα 50: ibm05 Σύγκριση Χρόνων - Μέρος 3

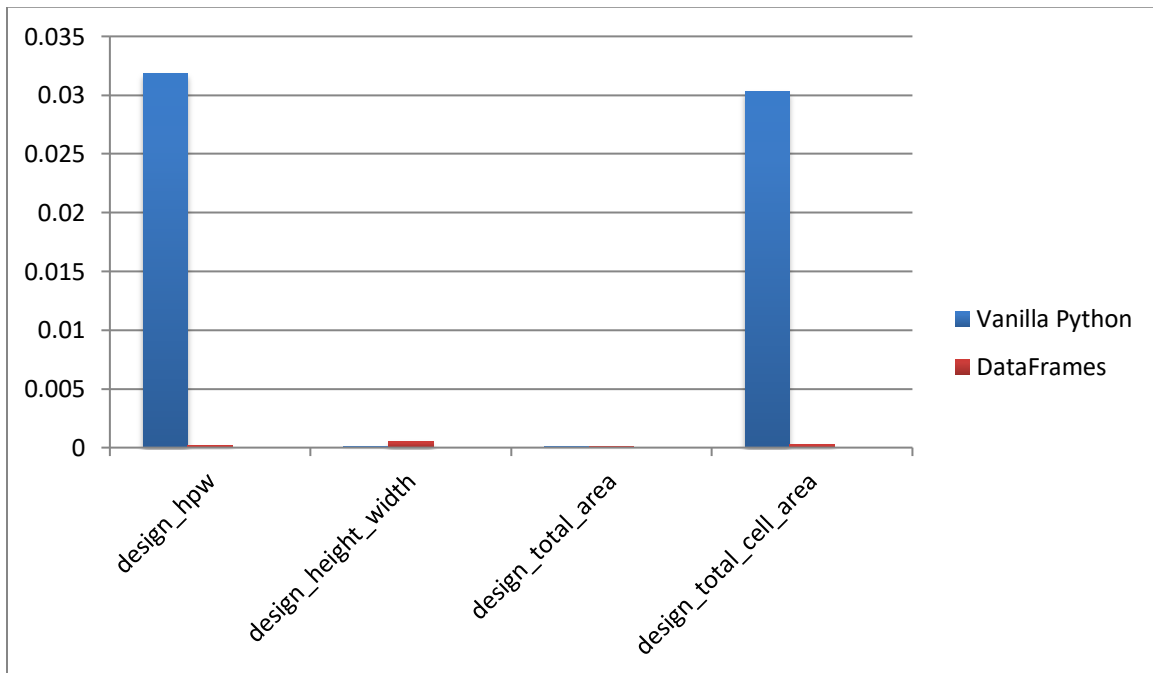
- **ibm18:**



Εικόνα 51: ibm18 Σύγκριση Χρόνων - Μέρος 1



Εικόνα 52: ibm18 Σύγκριση Χρόνων - Μέρος 2



Εικόνα 53: ibm18 Σύγκριση Χρόνων - Μέρος 3

Ακολουθούν οι ακριβείς χρόνοι μέτρησης, των προηγούμενων γραφικών αναπαραστάσεων, για τα τρία κυκλώματα:

Συναρτήσεις	Υλοποίηση	ibm18	ibm05	ibm01
Calculate_all_row_densities	Vanilla Python	10.783736 sec	0.6515546 sec	0.147983 sec
Calculate_all_row_densities_df	DataFrames	3.644311 sec	0.3675228 sec	0.132254 sec
calculate_all_net_hpw	Vanilla Python	0.146047 sec	0.0154014 sec	0.008036 sec
calculate_all_net_sizes	Vanilla Python	0.086624 sec	0.0156618 sec	0.006783 sec
calculate_all_net_hpw_df	DataFrames	0.002186 sec	0.0019596 sec	0.001579 sec
calculate_all_net_sizes_df	DataFrames	0.001788 sec	0.0012924 sec	0.001013 sec
Design_HP_W	Vanilla Python	0.031873 sec	0.004306 sec	0.002072 sec
Design_width_height	Vanilla Python	0.000070 sec	0.000039 sec	0.000027 sec
Design_total_area	Vanilla Python	0.000002 sec	0.000002 sec	0.000002 sec
Design_total_cell_area	Vanilla Python	0.030341 sec	0.004694 sec	0.001827 sec
Design_density	Vanilla Python	0.000017 sec	0.000011 sec	0.000009 sec
design_hpw	DataFrames	0.000197 sec	0.000117 sec	0.000139 sec
design_height_width	DataFrames	0.000525 sec	0.000406 sec	0.000469 sec
design_total_area	DataFrames	0.000001 sec	0.000001 sec	0.000001 sec
design_total_cell_area	DataFrames	0.000254 sec	0.000175 sec	0.000232 sec
design_density	DataFrames	0.000003 sec	0.000002 sec	0.000002 sec

Εικόνα 54: Χρόνοι Εκτέλεσης κοινών συναρτήσεων

Εύκολα συμπεραίνει κανείς, τόσο από τα γραφήματα όσο και από την ακριβή απεικόνιση των χρόνων εκτέλεσης, πως οι συναρτήσεις, που υλοποιήθηκαν με την συμβολή των DataFrame, ήταν αρκετά πιο γρήγορες σε όλα τα κυκλώματα, με μόνη εξαίρεση τη συνάρτηση **design_width_height**. Ήταν αναμενόμενο, πως με την μεγάλη αύξηση των δεδομένων, θα αυξανόταν και ο χρόνος εκτέλεσης των συναρτήσεων. Αξιοσημείωτο όμως αποτελεί το γεγονός, πως οι συναρτήσεις των DataFrame, εμφάνιζαν πολύ μικρές ανοδικές μεταβολές ή και σε ορισμένες περιπτώσεις, μηδαμινές. Αντιθέτως, στις συναρτήσεις της Vanilla Python, η αύξηση του χρόνου ήταν αισθητή.

4.5.2 ΧΡΟΝΟΙ ΕΚΤΕΛΕΣΗΣ ΟΛΩΝ ΤΩΝ ΣΥΝΑΡΤΗΣΕΩΝ

Στους ερχόμενους τρεις πίνακες, παραθέτονται οι χρόνοι που καταμετρήθηκαν, για όλες τις συναρτήσεις του προγράμματος.

Συναρτήσεις	Υλοποίηση	ibm18	ibm05	ibm01
parsing_files_to_lists	Vanilla Python	9h 22min 53sec	14min	1min 51sec
convert_lists_to_dfs	DataFrames	9.178966 sec	1.0704656 sec	0.429705 sec
create_nodes_df	DataFrames	1.323239 sec	0.1278426 sec	0.060444 sec
create_nets_df	DataFrames	1.976269 sec	0.5283564 sec	0.181114 sec
create_rows_df	DataFrames	3.807625 sec	0.4006536 sec	0.178974 sec
create_design_df	DataFrames	0.071833 sec	0.013613 sec	0.009173 sec

Εικόνα 55: Χρόνοι εκτέλεσης συναρτήσεων - Μέρος 1

Συναρτήσεις	Υλοποίηση	ibm18	ibm05	ibm01
number_of_non_terminal_nodes	DataFrames	0.090661 sec	0.014509 sec	0.007369 sec
biggest_non_terminal_node	DataFrames	0.092996 sec	0.015971 sec	0.007876 sec
smallest_non_terminal_node	DataFrames	0.099164 sec	0.016655 sec	0.009107 sec
mean_size_non_terminal_nodes	DataFrames	0.092132 sec	0.015644 sec	0.007873 sec
number_of_terminal_nodes	DataFrames	0.062529 sec	0.011138 sec	0.005155 sec
number_of_nets	DataFrames	0.000020 sec	0.000013 sec	0.000012 sec
biggest_net_based_on_nodes	DataFrames	0.140895 sec	0.022878 sec	0.012639 sec
smallest_net_based_on_nodes	DataFrames	0.165607 sec	0.031143 sec	0.016972 sec
mean_size_of_nets_based_on_nodes	DataFrames	0.070576 sec	0.010383 sec	0.005529 sec
biggest_net_based_on_size	DataFrames	0.001140 sec	0.000837 sec	0.000783 sec
smallest_net_based_on_size	DataFrames	0.001083 sec	0.000664 sec	0.000649 sec
mean_net_based_on_size	DataFrames	0.000384 sec	0.000155 sec	0.000137 sec
number_of_rows	DataFrames	0.000021 sec	0.000013 sec	0.000013 sec
biggest_row	DataFrames	0.002118 sec	0.001288 sec	0.001217sec
smallest_row	DataFrames	0.001949 sec	0.001132 sec	0.001099 sec
mean_num_of_nodes_on_rows	DataFrames	0.000750 sec	0.000364 sec	0.000344 sec

Εικόνα 56: Χρόνοι εκτέλεσης συναρτήσεων - Μέρος 2

Συναρτήσεις	Υλοποίηση	ibm18	ibm05	ibm01
allocation_of_non_terminal_node_sizes	DataFrames	0.903277 sec	0.589718 sec	0.579805 sec
allocation_of_net_sizes	DataFrames	0.810542 sec	0.6031644 sec	0.489902 sec
allocation_of_net_sizes_based_on_nodes	DataFrames	0.995666 sec	0.6706338 sec	0.574667 sec
allocation_of_cells_on_each_row	DataFrames	1.087535 sec	0.6863266 sec	0.570032 sec
allocation_of_row_densities	DataFrames	1.577376 sec	0.6618704 sec	0.612749 sec

Εικόνα 57: Χρόνοι εκτέλεσης συναρτήσεων - Μέρος 3

4.6 ΨΕΥΔΟΚΩΔΙΚΕΣ

4.6.1 ΕΙΣΑΓΩΓΗ

Όπως έχει αναφερθεί και προηγουμένως, με τη χρήση της βιβλιοθήκης `pandas`, είναι εφικτό να υλοποιηθούν και πιο σύνθετες λειτουργίες. Στη συνέχεια παραθέτονται δύο ψευδοκώδικες, οι οποίοι περιγράφουν τους αλγορίθμους «Tetris-like Legalization» και «Detailed Placement», οι οποίοι εφαρμόζονται πάνω στα κυκλώματα.

4.6.2 ΑΛΓΟΡΙΘΜΟΣ TETRIS-LIKE LEGALIZATION

Σκοπός του Tetris-like Legalization αλγορίθμου, είναι να εξαλείψει τις αλληλεπικαλύψεις κελιών, που ενδέχεται να έχουν προέλθει από την διαδικασία της καθολικής χωροθέτησης, ώστε να νομιμοποιηθεί το κύκλωμα.

Ο αλγόριθμος ξεκινάει ταξινομώντας τα κελιά ως προς την συντεταγμένη x , για να βρεθεί το αριστερότερο κελί, ανεξάρτητα από τη σειρά στην οποία είναι τοποθετημένο. Αρχικά όλες οι θέσεις των σειρών θεωρούνται διαθέσιμες και ξεκινώντας από το πιο αριστερό κελί, υπολογίζονται όλες οι ευκλείδειες αποστάσεις μεταξύ αυτού και της αριστερότερης διαθέσιμης θέσης, της κάθε σειράς. Στην πορεία, επιλέγοντας την μικρότερη ευκλείδεια απόσταση (μετατόπιση), μετακινείται το κελί στην αντίστοιχη θέση. Αφού ολοκληρωθεί η μετατόπιση, το κελί αυτό θεωρείται αμετακίνητο και εκτελούμε την ίδια σειρά ενεργειών για το επόμενο. Η μόνη διαφορά, είναι ότι η αριστερότερη πλέον διαθέσιμη θέση για μία από όλες τις σειρές, δεν θα είναι από την αρχή αυτής, αλλά από πιο μέσα, μιας και στην αρχή της τοποθετήθηκε το πρώτο κελί. Ομοίως το ίδιο θα ισχύει και για τα ερχόμενα κελιά, που θα εξεταστούν, καθώς όσο πραγματοποιούνται μετατοπίσεις, οι αριστερότερες ελεύθερες θέσεις των σειρών αλλάζουν. Η παραπάνω διαδικασία, ολοκληρώνεται όταν διαπεραστούν όλα τα κελιά του κυκλώματος.

```

1  Ψευδοκώδικας 1: Αλγόριθμος Tetris-like Legalization
2  -----
3  Είσοδοι: Μη νομιμοποιημένο κύκλωμα που περιγράφεται στα
4           Nodes_DataFrame(NDDF), Nets_DataFrame(NTDF),
5           Rows_DataFrame(RDF), Design_DataFrame(DDF)
6
7  Εξοδοι: Νομιμοποιημένο κύκλωμα, ενημερωμένα DataFrames
8  -----
9
10 sort NDDF σύμφωνα με την στήλη της συντεταγμένης-x
11
12 foreach cell_name in (column names of NDDF):
13     NDDF["cell_name"] get x_minimum
14
15     # next_available_x είναι στήλη του Rows_DataFrame
16     # όπου αποθηκεύει την επόμενη διαθέσιμη θέση της σειράς
17
18     # ελέγχει την αριστερότερη ελεύθερη θέση όλων των σειρών
19     get(row_name): = εύρεση μικρότερης ευκλείδειας απόστασης του cell_name, RDF["row_name"].next_available_x
20     insert cell_name into RDF["row_name"]
21
22     update NDDF["cell_name"] με τις νέες συντεταγμένες, της μετατόπισης
23 end foreach
24
25 update NTDF
26 update DDF

```

Εικόνα 58: Tetris-like Legalization ψευδοκώδικας

4.6.3 ΑΛΓΟΡΙΘΜΟΣ DETAILED PLACEMENT

Ο αλγόριθμος Detailed Placement, διαδέχεται το στάδιο της νομιμοποίησης και συνήθως περιλαμβάνει ανταλλαγές κελιών ή επανατοποθέτηση αυτών, στα πλαίσια των σειρών.

Ξεκινάει, έχοντας ως είσοδο ένα νομιμοποιημένο κύκλωμα και πραγματώνει ανταλλαγές κελιών, ίδιου μεγέθους, όταν αυτές καταλήγουν σε καλύτερο συνολικό μήκος καλωδίου. Η σειρά με την οποία διαπερνώνται τα κελιά, είναι πάντα η ίδια, ενώ για το καθένα από αυτά, πραγματοποιείται η πρώτη συμφέρουσα ανταλλαγή. Ο αλγόριθμος ολοκληρώνεται, όταν δεν υπάρχουν ανταλλαγές, οι οποίες μπορούν να βελτιώσουν περαιτέρω το αποτέλεσμα.

```

1  Ψευδοκώδικας 2: Αλγόριθμος Detailed Placement
2  -----
3  Είσοδοι: Νομιμοποιημένο κύκλωμα που περιγράφεται στα
4           Nodes_DataFrame(NDDF), Nets_DataFrame(NTDF),
5           Rows_DataFrame(RDF), Design_DataFrame(DDF)
6
7  Εξοδοι: Βελτιωμένο Νομιμοποιημένο κύκλωμα, ενημερωμένα DataFrames
8  -----
9
10 # roll-back σημεία
11 old_NDDF = προϋπάρχον NDDF πριν τις ανταλλαγές θέσεων
12 old_NTDF = προϋπάρχον NTDF πριν τις ανταλλαγές θέσεων
13 old_RDF = προϋπάρχον RDF πριν τις ανταλλαγές θέσεων
14 old_DDF = προϋπάρχον DDF πριν τις ανταλλαγές θέσεων
15
16 cell_names_list = NDDF["cell_name"] # Λίστα με τα ονόματα των cells
17
18 while βρίσκεται_καλύτερο_αποτέλεσμα
19     foreach cell_name in cell_names_list:
20
21         size = NDDF["cell_name"].size # Μέγεθος τρέχων κόμβου
22
23         while True:
24
25             next_cell_name := εύρεση_επόμενου_cell_με_το_ίδιο_μέγεθος_στο_NDDF
26             swap_positions(NDDF["cell_name"], NDDF["next_cell_name"])
27
28             update_NTDF # υπολογισμός νέων καλωδίων για τα Nets
29             update_DDF # υπολογισμός νέου συνολικού καλωδίου
30
31             if DDF.total_hrw > old_DDF then: # η αλλαγή δεν κατέληξε σε καλύτερο μήκος καλωδίου.
32                 DDF = old_DDF # άρα πάμε πάλι πίσω στις παλιές θέσεις.
33                 NTDF = old_NTDF # για να αναιρέσουμε τις αλλαγές
34                 NDDF = old_NDDF
35             end if
36
37             if DDF.total_hrw < old_DDF.total_hrw then: # η αλλαγή κατέληξε σε καλύτερο μήκος καλωδίου.
38                 old_DDF = DDF # φτιάχνουμε νέα roll-back σημεία
39                 old_NTDF = NTDF
40                 old_NDDF = NDDF
41                 break και_συνεχίζουμε_με_το_επόμενο_cell_name_στο_foreach
42             end if
43         end while
44     end foreach
45 end while
46
47 update RDF # ενημέρωση, αφού ολοκληρωθούν οι μετατοπίσεις

```

Εικόνα 59: Detailed Placement ψευδοκώδικας

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα εργασία εξετάστηκε η ανάλυση μεγάλου όγκου δεδομένων και η εξαγωγή πορισμάτων, με την αρωγή της γλώσσας Python και κυριότερα με την βιβλιοθήκη αυτής, την `pandas`. Τα δεδομένα στα οποία εφαρμόστηκαν οι διαδικασίες, που υλοποιήθηκαν, περιγράφουν ολοκληρωμένα κυκλώματα, τα οποία δίνονται σε `bookshelf` μορφή.

Στη σύγχρονη εποχή, εξαιτίας της ταχύρρυθμης ανάπτυξης της τεχνολογίας αλλά και του όγκου δεδομένων και πληροφοριών, που έχουν ψηφιοποιηθεί, κρίνεται αναγκαία η εύρεση νέων και γρήγορων τρόπων ανάλυσης αυτών, καθώς έχουν ενσωματωθεί και επηρεάζουν την καθημερινότητα του ανθρώπου.

Με αφορμή το γεγονός αυτό, εξετάσαμε τη βιβλιοθήκη `pandas`, η οποία δημιουργήθηκε και ενδείκνυται για τον προαναφερόμενο σκοπό. Πιο αναλυτικά, η προσέγγιση μας ξεκίνησε με τον μέχρι πρότινος καθιερωμένο τρόπο, δηλαδή, χωρίς την χρήση κάποιας πιο εξειδικευμένης βιβλιοθήκης. Στο στάδιο αυτό, με τα βασικά εργαλεία της Python ξεκίνησε η εξαγωγή κάποιων αποτελεσμάτων και στην πορεία έγινε η μετάβαση στη βιβλιοθήκη `pandas`. Με τη συμβολή αυτής, υλοποιήθηκαν ορισμένες λειτουργίες, οι οποίες είχαν πραγματοποιηθεί και προηγουμένως με την Python, με σκοπό την σύγκριση των δύο τρόπων, ενώ στη συνέχεια έγινε ανάπτυξη και περαιτέρω διαδικασιών ανάλυσης και διεξαγωγής αποτελεσμάτων.

Ως απόρροια των ανωτέρω ενεργειών, παρατηρήθηκε ότι η βιβλιοθήκη `pandas` παρουσιάζει προβάδισμα χρόνου εκτέλεσης, σε σύγκριση με τις συναρτήσεις, που γράφτηκαν, χωρίς την συνδρομή αυτής. Επιπρόσθετα, είναι ικανή να επιτυγχάνει σύνθετες λειτουργίες σε μικρό πλήθος εντολών, συμβάλλοντας έτσι στη μείωση της διάρκειας, που απαιτείται για την ανάπτυξη μίας εφαρμογής, γεγονός που αυξάνει την παραγωγικότητα ενώ ταυτόχρονα, αξιοσημείωτο είναι το γεγονός ότι παρόλο που ο όγκος των δεδομένων αυξανόταν, οι χρόνοι εκτέλεσης παρέμεναν αποδοτικοί.

Παράλληλα, συνδυάζοντας τις δυνατότητες της με το πλήθος βιβλιοθηκών, που προσφέρει η γλώσσα, επεκτείνει τη χρησιμότητά της και συγχρόνως, λόγω της εύκολα κατανοητής μορφής τους, η ανάγνωση της καθίσταται προσιτή για το ευρύ κοινό.

Καταλήγοντας, στην εποχή των Μεγάλων Δεδομένων, όλα τα ανωτέρω αποτελούν κομβικά σημεία. Για το λόγο αυτό, η βιβλιοθήκη pandas οφείλει να είναι από τις πρώτες επιλογές, για τέτοιες ανάλογες χρήσεις.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

Mano, M. M. & Ciletti, M. D. (2014). Ψηφιακή Σχεδίαση. Αθήνα: Εκδόσεις Παπασωτηρίου.

Wakerly, J. F. (2005). Ψηφιακή Σχεδίαση: Αρχές & Πρακτικές. Αθήνα: Εκδόσεις Κλειδάριθμος.

Μπούρας Α. Σ. & Κάππος Γ. Θ. (2017). Αλγοριθμική και προγραμματισμός υπολογιστών σε PYTHON. Αθήνα: Εκδόσεις Κλειδάριθμος.

About pandas. Ανακτήθηκε από <https://pandas.pydata.org/about/index.html>