ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Ανάπτυξη Εργαλείων CAD για Σχεδιάση Ολοκληρωμένων Κυκλωμάτων

(ΗΡΥ 419)

5η ΑΣΚΗΣΗ

Κωνσταντίνος Νικολός 2019030096

Σκοπός της άσκησης:

Σκοπός της συγκεκριμένης άσκησης είναι το πρόγραμμα που ήδη υλοποιήθηκε να αποκτήσει μία ΄΄γενικότητα΄΄. Δηλαδή να είμαστε σε θέση να διαβάσουμε και να παράξουμε την δομή ενός δεύτερου εργαλείου του Full_adder_Subtractor αλλάζοντας τις μεθόδους ως ΄΄μηχανκοί΄΄.

Περιγραφή της άσκησης:

Ζητείται να μπορεί να διαβαστεί ένα αρχείο που περιέχει είτε την δομή 1bit full_adder είτε την δομή 1bit subtractor και να παράξει το σωστό netlist σε μορφή components για ένα nbit adder/subtractor. Ακόμη διαβάζοντας τις δοθέντες βιβλιοθήκες το πρόγραμμα πρέπει αφού διαβάσει το netlist που παράχθηκε να διαβάσει την κατάλληλη βιβλιοθήκη και να παράξει το netlist σε μορφή πυλών

Δομές και Λειτουργία τους:

Για την παραγωγή netlist σε μορφή components (Άσκηση 4) χρησιμοποιούνται δύο βασικές δομές εκείνη του Full_adder δηλαδή του component και το entity το οποίο περιέχει τις πληροφορίες που δόθηκαν απο το αρχείο. Το entity διαβάζει ολά τα inputs που δόθηκαν για κάθε component τα επεξεργάζεται και καλεί την create_full_adder() όπου αποθυκευεί τα σωστά δεδομένα για κάθε component.

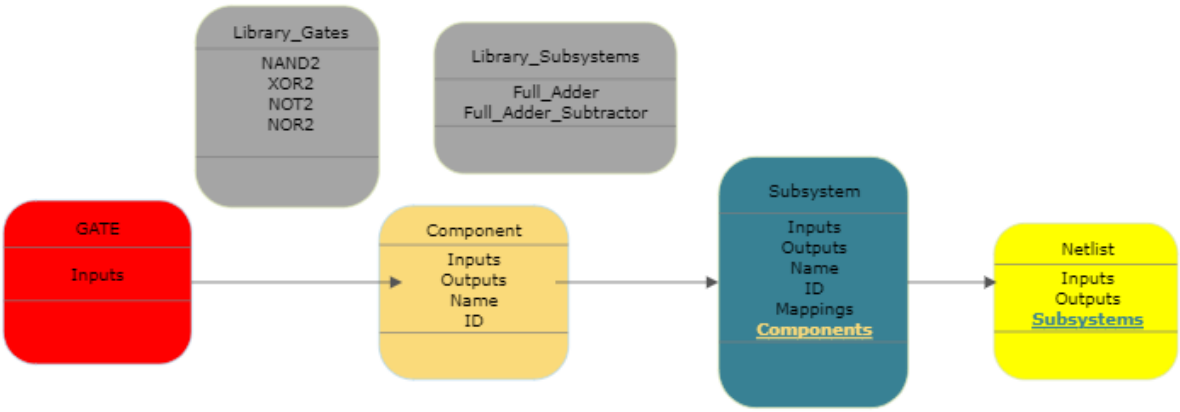
Για την παραγωγή netlist σε μορφή πυλών (Άσκηση3) χρησιμοποιούνται οι εξής:

- **Gate** αποτελεί μια απλή πύλη με εισόδους μόνο. Το σύνολο των πύλων απαρτίζουν τη βιλβιοθήκη πυλών από την οποία τι ΄΄τραβάμε΄΄

- **Component** Χρησιμποιείται για να δώσουμε μια παραπάνω λειτουργικότητα στι πύλες οι οποίες έχουν πλέον και τα outputs τους για να περαστούν στα subsystems(1bitadder).

- **Subsystem** Εδώ αποθυκεύεται η δομή του 1bit adder. Αποτελείται απο components με τα inputs και outputs κάθε component και φυσικά τα outputs του ίδιου του adder. Η δομή του είναι δυναμική διαβάζονται όλα τα inputs και για το κάθε ένα input που προκύπτει δεσμεύουμε νέο χώρο με την χρήση της remalloc. Επί της ουσίας μέσα στο subsystem αποθυκεύται η σύνδεση μεταξύ των inputs του subsystem και των inputs/outputs κάθε component ώστε να χρειάζεται αργότερα να δώσουμε μόνο τα inputs του subsystem. Επειδή πρόκεται για δυναμική δομή μπορεί να αποθυκεύσει και adder και subtractor.

- **Netlist** Αποτελεί εξίσου μια δυναμική δομή επομένως μπορεί να φιλοξενήσει είσου ένα adder και ένα subtractor δεσμεύοντα χώρο μόνο για ότι χρειάζεται. Αποτελείται από subsystems τα οποία συνδέονται κατάλληλα μεταξύ τους.



<u>Συναρτήσεις:</u>

- int read_line_from_file()
  Χρησιμοποιείται για να διαβαστεί μια γραμμή από ένα αρχείο
- char* split()
  Χρησιμοποιείται για να χωριστεί ένα string στο αναζητούμενο διαχωριστικό
- Full_adder* create_full_adder()
  Δημιουργεί ένα component full_adder με τις εισόδους που του δίνονται κατά την κλίση και επιστρέφει ενά Full_adder*.Στις θέσεις των outputs και cout υπάρχουν απλά τα strings ui_S ui_Cout που χρειάζονται κατά την εκτύπωση ενώ

τα δοσμένα outputs βρίσκονται στη δομή entity και γινέται μετά η αντιστοιχηση.

- int getEntity()

  Δημουργεί τη δομή entity με τα inputs ouputs που δίνονται και επίση δημιουργεί μια λίστα από adders καλώντας την συνάρτηση create_full_adder()

- int createTable()

  Δημιουργεί ένα πίνακα char** όπου αποθυκεύονται προσωρινά τα δεδομένα του νέου αρχείου. Εκεί γίνεται η αντιστοίχηση των εισόδων και εξόδων σύμφωνα με την μορφή του netlist ενός full_adder n bits.

- int str_to_list()

  Χρησιμοποιείται για να παραχθούν τα inputs/outputs από ένα string

- getLibraryFromFile()

  Χρησιμοποιείται για να παραχθεί η βιβλιοθήκη από gates διαβάζοντας από το δοσμένο αρχείο.

- getSubSystemFromFile()

  Χρησιμοποιείται για να παραχθεί ένα subsystem full_adder από το δοσμένο αρχείο ελέγχοντας πως κάθε component του αποτελεί μια έκτων πυλών από τη βιβλιοθήκη. Η μορφή του full_adder είναι δυναμική ,δήλαδη τα inputs ενός component που μπορεί να είναι η έξοδο κάποιο προηγούμενου component θέτουμε το input του νέου component να δείχνει στο προηγούμενο ώστε αν αλλάξει το προηγούμενο να αλλάξει και το input του νέου. Το ίδιο ισχύει και για τα inputs του full_adder αλλά και για τα outputs

- SetNetlist()

  Αρχικά διαβάζει από το αρχείο τη μορφή του netlist που δίνεται και δημιουργεί subsystem full_adders με τα αντίστοιχα inputs. Τα πρώτα δύο inputs αποτελούν και inputs του netlist (Ai,Bi) ενώ το τρίτο δίνεται από το carry_out του προηγούμενου full_adder σε μορφή πύλης (gate ID).Τα component ids κάθε full_adder αλλάζουν καθώς πρέπει να ξεκινάνε από ένα συγκεκριμένο index. Τελικά τυπώνεται κάθε component με μορφή (ID Name Inp[0],Inp[1]).

<u>Περιγραφή λειτουργίας του προγράμματος:</u>

- Το πρόγραμμα έχει υλοποιηθεί σε δύο διαφορετικά μέρη, στην παραγωγή του netlist σε μορφή components στο **Assignment5_netlist.c** και στην παραγωγή του netlist σε μορφή gates. Για την υλοποίηση του πρώτου διαβάζεται το αρχείο entity

χρησιμοποιώντας την getEntity() η οποία διαβάζει εάν πρόκειται για subtractor η απλώ adder και πράττει ανάλογα σε κάθε περίπτωση καλεί την create_full_adder και δημιουργείται κάθε component adder/subtractor με την σωστή μορφή για να τυπωθεί. Ο αριθμός των bits του τελικού netlist διαβάζεται απο το αρχείο.Τελικά παράγεται ένα αρχείο με το netlist από components

- Στο δεύτερο μέρος στην παραγωγή netlist με πύλες στο **Assignment5_gates.c** αρχικά διαβάζεται η βιλβιοθήκη πυλών.Στη συνέχεια καλείται η συνάρτηση setNelist η οποία διαβάζει το αρχείο δημιουργεί ένα netlist , αποθυκεύει τα inputs και ανάλογα εάν πρόκειται για subtractor η adder καλεί την getSubsystem για κάθε bit που ζητείται με όρισμα την βιβλιοθήκη adder ή subtractor. διάβαζεται η βιβλιοθήκη του υποσυστήματος subtractor/adder και δημιουργείται ένα subsystem ,όπως αναφέρθηκε είναι δυναμική οπότ μπορεί να αποθυκεύσει και τα δυο υποσυστήματα. Τέλος η setNelist δίνει στα υποσυστήματα τις κατάλληλες εισόδους είτε από τα αρχικά inputs του netlist είτε συνδέοντας με τα outputs προηγούμενων υποσυστημάτων και παράγει σαν έξοδο το netlist.

Running The Code

- Για την δημιουργία του netlist σε μορφή πυλών τρέχουμε το **Assignment5_gates.c** με εισόδους είτε το αρχείο **netlist_SUB.txt** για subtractor είτε το αρχείο **netlist_ADD.txt** για adder και τα αρχεία **Comp_full_adder.txt** , **Comp_full_subadder.txt** , **Gates.txt**
- Για την δημιουργία του netlist σε μορφή components τρέχουμε το **Assignment5_netlist** με εισόδους είτε το αρχείο **entity_subadder.txt** για subtractor είτε το αρχείο **entity_adder.txt** για adder

Προβλήματα:

Το πρόβλημα που αντιμετώπισα είναι το εξής, ενώ το **Assignment5_gates.c** παράγει ακριβώς το output που θέλουμε και στις δύο περιπτώσεις εάν του δώσουμε σαν όρισμα τα κατάλληλα αρχεία με netlist από components **netlist_ADD.txt** και **netlist_SUB.txt** το αρχείο που δημιουργείται από το **Assignment5_netlist.c** (netlistDon5.txt) χρησιμοποιώντας ως όρισμα το **entity_adder.txt** είτε το **entity_subadder.txt** αποτελεί ακριβώς ίδιο με αυτό που δώθηκε στο **Assignment5_gates.c (netlist_ADD.txt** είτε **netlist_SUB.txt)** αν δωθεί στο πρόγραμμα **Assignment5_gates.c** δεν θα τρέχει η δεν θα παραχθεί σωστό αποτέλεσμα. Δεν κατάφερα να βρω γιατί και να το διορθώσω.