Αναφορά Δεύτερης εργαστηριακής άσκησης

HPY 419 - HPY 608 - Ανάπτυξη Εργαλείων CAD για Σχεδίαση Ολοκληρωμένων Κυκλωμάτων 2022- 2023

ΑΙΚΑΤΕΡΙΝΗ ΤΣΙΜΠΙΡΔΩΝΗ:2018030013

Εισαγωγή: Αυτό που ζητήθηκε στην άσκηση 3 ήταν η ανάγνωση ενός αρχείου εισόδου καθώς και την ανάγνωση των αρχείων και τοποθέτηση μέσα σε δομές των βιβλιοθηκών του υποσυστήματος και τέλος στη δημιουργία ενός τελικού αρχείου εξόδου που συνδυάζει τις δύο λίστες αυτές για να φτιάξει το πλήρες κύκλωμα.

Υλοποίηση : Το πρόγραμμα υλοποιήθηκε σε γλώσσα C.
Αρχικά έγινε ανάγνωση των αρχείων εισόδου(βιβλιοθηκες,εισοδος) και αποθήκευση σε δομές. Από την component library που επίσης δημιουργεί το πρόγραμμά μας φτιάχνουμε την Subsystem library με την προϋπόθεση ότι οι πύλες υπάρχουν μέσα και μετά αφού έχουμε δημιουργήσει το Subsystem System Library το διαβάζουμε και τα τοποθηκε μέσα σε δομές. Παίρνοντας και το αρχείο εισόδου που είχαμε από την προηγούμενη άσκηση γίνεται ένας συνδυασμός ώστε να φτιαχτεί ένα τελικό αρχείο το οποίο περιέχει όλες τις συνδεσμολογίες τις εισόδους και τις εξόδους καθώς και των ονόματα των πυλών για την κατασκευή ενός **FULL ADDER.**

Πιο συγκεκριμένα για να φτιάξω το subsystem Library έπρεπε να σκεφτώ πώς θα φτιάξω FULL ADDER χωρίς πήρες AND και OR αυτό ήταν εύκολο κάνοντας de Morgan στις κλασικές εξισώσεις του FULL ADDER . Την XOR την κράτησα όπως είναι εφόσον την υπάρχει μέσα στη βιβλιοθήκη ενώ ξέροντας ότι NOT(AND)=NAND και NOT(OR)=NOR από το de Morgan βγήκε εύκολα.
Το subsystem file είναι το παρακάτω:

```
%% THIS IS THE SUBSYSTEM LIBRARY, COMPRISING OF SUBSYSTEMS
%% MADE OF GATES FROM THE COMPONENT LIBRARY
** SUBSYSTEM LIBRARY
COMP FULL_ADDER ; IN:A,B,C ; OUT: S, COUT
BEGIN FULL_ADDER NETLIST
U0 XOR2 A,B
U1 XOR2 U0,C
U2 NAND2 C,U0
U3 NAND2 A,B
U4 NOT U3
U5 NOT U2
U6 NOR2 U4,U5
U7 NOT U6
Sum=U1
Cout=U7
END FULL_ADDER NETLIST
```

Μετά για να την ανάγνωση του αρχείου subsystem File για την τοποθέτηση σε μία δομή χρησιμοποίησα structs όπως Component_Dec_Array, netlist, OutputArray που και αυτά έχουν δικές τους δομές και το συγκέντρωσα τελικά σε μία δομή που την ονόμασα subsystemFile.Του αρχείου χωρίστηκε σε τρεις δομές ένα το declaration array που περιέχει έναν πίνακα από δομες Component_Dec πού μέσα αυτό περιέχει πίνακες και δομές για να αποθηκεύσουμε το όνομα , τις εισόδους και τις εξόδους .Σε ένα netlist που περιέχει το κύριο σώμα με τις πύλες που και αυτό από πίσω του περιέχει μία δομή component με τα ονόματα και τις εισόδους και τις συνδέσεις τον πηλων και και μία δομή για την έξοδο που ακολουθεί ακριβώς την ίδια λογική. Με αυτή τη λογική είναι φτιαγμένες όλες μου οι δομές δηλαδή ξεκινάνε από το στάδιο που έχουμε ένας struct πού αποθηκεύει την πληροφορία που θέλουμε να κρατήσουμε και αυτή η δομή μετά αποθηκεύεται σε ένα μεγαλύτερο πίνακα μαζί με παρόμοιες δομές με αυτή ώστε να μπορέσουμε να διαβάσουμε ευκολότερα και περισσότερα στοιχεία .Ακριβώς το ίδιο ισχύει και για οποιαδήποτε άλλη δομή συναντήσουμε στο πρόγραμμα.

Για το αρχείο εισόδου μάλιστα χρησιμοποιήθηκε η ίδια δομή netlist που στην ουσία είναι ένας πίνακας από πολλά ακόμα Component για να αποθηκευτεί το σώμα και μία δομή outputArray για να αποθηκευτούν τα διάφορα output. Τώρα για το τελικό αρχείο ακολούθησα την ίδια λογική και κατασκεύασα μία δομή finalFile που περιέχει ένα πίνακα πόνε κλείστηκε ένα πινακάκι output ώστε εκεί να αποθηκευτούν οι τελικές πληροφορίες πιο αναλυτικά θα δείτε παρακάτω που εξηγούνται οι συναρτήσεις.

Συναρτήσεις: το πρόγραμμα περιέχει αρκετές συναρτήσειςδεν γίνεται περιγράφουν όλες αλλά θα περιγράψω τις πιο βασικές .

- ComponentLibrary *MakeComponentLibrary(void):

Αυτή η συνάρτηση φτιαχνει και επιστρέφει μια δομή ComponentLibrary που περιέχει πληροφορίες σχετικά με τους επιτρεπόμενους τύπους στοιχείων. Οι πληροφορίες διαβάζονται από ένα αρχείο με το όνομα "component_library.txt". Η δομή περιέχει έναν πίνακα τύπων Component_S, όπου κάθε τύπος περιέχει ένα όνομα, έναν αριθμό εισόδων, έναν αριθμό εξόδων και και τον αριθμό τον στοιχείων περιέχονται μέσα.

- void MakeSubsystemFile

Η συνάρτηση χρησιμοποιεί τους τύπους στοιχείων και τις συμπεριφορές τους που ορίζονται στη δομή ComponentLibrary για να δημιουργήσει ένα αρχείο για τη δομή του full_adder. Αν προσπαθήσω να εισάγω ένα στοιχείο που δεν είναι μέσα στη δομή component Library πετάει λάθος.Στην ουσία αυτό που κάνει αυτή η συνάρτηση είναι να ελέγχει ποιες πύλες χρησιμοποιούμε και απλά το γράφει σε ένα αρχείο ώστε να μπορούμε μετά να πάρουμε αυτό το αρχείο και να κατασκευάσουμε τις δομές.

- subsystemFile* StoreTheSubsystemfile

Αυτό ακριβώς κάνει η Storesubsystemfile παίρνει το αρχείο που κατασκευάσαμε Τα σπάει σε δομές με τη βοήθεια βοηθητικός συναρτήσεων οπως readTheComponent,readThebody,ReadTheOutput οπου η καθεμία διαβάζει ένα συγκεκριμένο κομμάτι και αποθηκεύει σε δομές και στο τέλος της βάζει σε μία κοινή δομή subsystemFile την οποία και επιστρέφει.

- InputFile* readAndStoreInputFile

Ακριβώς Με την ίδια λογική λειτουργεί και η readAndStoreInputFile η οποία διαβάζει το input file από το προηγούμενο εργαστήριο το σπαει σε δομές και το αποθηκεύει σε μία τελική δομή την InputFile.

- Function makeTheFinalOutputFile

Αυτή είναι η τελική συνάντηση η οποία στην ουσία παίρνει τα δύο αρχεία που έχουμε δημιουργήσει παραπάνω στην αρχή βρίσκει πόσα στοιχεία υπάρχουν μέσα στο αρχείο εισόδου και με μία καινούργια δομή finalFile η οποία περιέχει πίνακες από netlist (που εχω ορισει καταχρηστικά τις συνδέσεις των διαφορετικών component και όχι τις εξόδους) δηλαδή πίνακες με πύλες σε αυτή την περίπτωση και πίνακες από output.Για κάθε ένα στοιχείο που υπάρχει στον πίνακα εισόδου φτιάχνουμε ένα πίνακα netlist και το βάζουμε μέσα σε ένα πίνακα με άλλα netlist πού προέρχονται από το ίδιο αρχείο και μέσα ως σώμα βάζουμε το σώμα που έχουμε από το θα subsystemfile. Ενώ για τις εξόδους βάζουμε αποκλειστικά ότι υπάρχει στο input file Και μετά με διάφορες βοηθητικές συναρτήσεις ενημερώνουμε τις εισόδους τις εξόδους και τις διασυνδέσεις ώστε να βγαίνει σωστό το αποτέλεσμα .

Αποτελέσματα:

```
% THIS IS THE COMPONENT LIBRARY, COMPRISING OF GATES
** COMPONENT LIBRARY
COMP NOT ; IN:P
COMP NAND2 ; IN:P,Q
COMP NOR2 ; IN:P,Q
COMP XOR2 ; IN:P,Q
```

```
%% THIS IS THE SUBSYSTEM LIBRARY, COMPRISING OF SUBSYSTEMS
%% MADE OF GATES FROM THE COMPONENT LIBRARY
** SUBSYSTEM LIBRARY
COMP FULL_ADDER ; IN:A,B,C ; OUT: S, COUT
BEGIN FULL_ADDER NETLIST
U0 XOR2 A,B
U1 XOR2 U0,C
U2 NAND2 C,U0
U3 NAND2 A,B
U4 NOT U3
U5 NOT U2
U6 NOR2 U4,U5
U7 NOT U6
Sum=U1
COUT=U7
END FULL_ADDER NETLIST
```

```
U0 XOR2 A0,B0
U1 XOR2 U0,cin
U2 NAND2 cin,U0
U3 NAND2 A0,B0
U4 NOT U3
U5 NOT U2
U6 NOR2 U4,U5
U7 NOT U6
U8 XOR2 A1,B1
U9 XOR2 U8,U7
U10 NAND2 U8,U7
U11 NAND2 A1,B1
U12 NOT U11
U13 NOT U10
U14 NOR2 U12,U13
U15 NOT U14
U16 XOR2 A2,B2
U17 XOR2 U16,U15
U18 NAND2 U16,U15
U19 NAND2 A2,B2
U20 NOT U19
U21 NOT U18
U22 NOR2 U20,U21
U23 NOT U22
```

```
U33 XOR2 U32,U31
U34 NAND2 U32,U31
U35 NAND2 A4,B4
U36 NOT U35
U37 NOT U34
U38 NOR2 U36,U37
U39 NOT U38
U40 XOR2 A5,B5
U41 XOR2 U40,U39
U42 NAND2 U40,U39
U43 NAND2 A5,B5
U44 NOT U43
U45 NOT U42
U46 NOR2 U44,U45
U47 NOT U46
S0 = U1
S1 = U9
S2 = U17
S3 = U25
S4 = U33
S5 = U41
Cout = U47
```

Αυτή είναι η έξοδος για ένα 6 bit full adder και φαίνεται να λειτουργεί όπως πρέπει δηλαδή να βγάζει σωστά τις εξόδους τις διασυνδέσεις καθώς και ID του κάθε component. Όπως φαίνεται ας πούμε η έξοδος βγαίνει από την τελευταία not όπως φαίνεται και στο σχήμα και η πρώτη έξοδος που είναι και εύκολο να το συγκρίνουμε με το πρωτότυπο στο subsystem file βγαίνει από την U1 όπως και εδώ και προφανώς δεύτερη οπως φαίνεται Βγαίνει σωστά από την XOR2 η οποία έχει id 9.

Παρατηρήσεις:Άλλαξα λίγο το αρχείο fulladder που είχα στείλει διότι το είχα γράψει το όνομα με μικρά γράμματα και το πρόγραμμα που έφτιαξα τώρα δεν το αναγνώριζε οπότε επισυνάπτω και το καινούργιο. Επίσης δεν έχω συμπεριλάβει μέσα στο κώδικα τώρα άσκηση 2 δεν ξέρω αν ήταν απαραίτητο απλά το έκανα για να εξοικονομήσω λίγο χώρο εφόσον ο κωδικός μου είναι εκτεταμένος.