

## Ανάπτυξη Εργαλείων CAD για Σχεδίαση Ολοκληρωμένων Κυκλωμάτων (HPY 419)

Κωνσταντίνος Νικολός 2019030096

### Σκοπός της άσκησης:

Περαιτέρω εξοικείωση με τη χρήση δομών δεδομένων και της διαδικασίας σύνθεσης υλοποιώντας πλέον τη δημιουργία βιβλιοθηκών και της χρήσης τους.

### Περιγραφή της άσκησης:

Ζητείται να διαβαστεί ένα αρχείο που περιέχει μια βιβλιοθήκη από gates, ένα αρχείο που περιέχει τη δομή ενός subsystem full\_adder με την προϋπόθεση ότι κάθε πύλη που περιέχεται στον full\_adder ανήκει και στην βιβλιοθήκη που διαβάστηκε. Τελικά ζητείται διαβάζοντας ένα αρχείο netlist και χρησιμοποιώντας το full\_adder subsystem να παραχθεί το netlist ενός full\_adder n-bits.

### Δομές:

Χρησιμοποιούνται τρεις βασικές δομές εκείνη του library, εκείνη του subsystem και το netlist.

Το κάθε library περιέχει ένα array από gates με ορίσματα το όνομα και τις εισόδους του κάθε gate. Δεν χρειάζονται outputs για τα gates.

Η δομή subsystem περιέχει πλέον και outputs, output\_mappings και ID καθώς μπορεί να αποτελεί έναν full\_adder με ξεχωριστό αριθμό και εξόδους όπως δίνεται από το netlist. Ακόμη περιέχει ένα array από components τα οποία επίσης περιέχουν outputs και ID καθώς θα αποτελέσουν τις πύλες από τις οποίες αποτελείται ένας full\_adder.

Το netlist περιέχει ένα array από full\_adders και τις εισόδους που δίνονται από το αρχείο netlist.

### Συναρτήσεις:

- `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
- `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 ,δηλαδή 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]).

- `freeLibrary()`
- `freeSubsystem()`
- `freeNetlist()`

### Περιγραφή λειτουργίας του προγράμματος:

- Καλείται αρχικά η συνάρτηση `getLibraryFromFile()` για να δημιουργηθεί η δομή των πυλών τις οποίες μπορούμε να χρησιμοποιήσουμε. Καλείται η συνάρτηση `setNetlist()` η οποία αφού διαβάσει τα inputs κάθε adder καλεί την `getSubSystemFromFile()` δημιουργώντας την δομή του `full_adder` και εισάγει μετά τα σωστά inputs. Τέλος εκτυπώνει το ζητούμενο netlist

### Προβλήματα:

Αρχικά έγραψα όλο τον κώδικα σε linux χρησιμοποιώντας και το valgrind για εύρεση λαθών. Στο δικό μου υπολογιστή σε linux ο κώδικας τρέχει και εμφανίζει τα σωστά αποτελέσματα θα ανεβάσω ένα screenshot στο τέλος του pdf. Δοκίμασα να το τρέξω στον υπολογιστή ενός φίλου μου σε windows και δεν τυπώθηκε τύπου οπότε πιθανό να μην τυπωθεί και σε εσάς σε windows. Δυστυχώς δεν μπορώ και να βρω τι πάει λάθος στα windows. Η valgrind δεν μου εμφάνιζε κάποιο λάθος πέραν από κάποια λάθη στις συναρτήσεις `freeSubsystem()` και `freeNetlist` για αυτό και τελικά δεν τις χρησιμοποίησα.

### Screenshot από linux:

```
U0 NAND2 A0,B0
U1 XOR2 A0,B0
U2 NAND2 U1,Cin
U3 XOR2 U1,Cin
U4 NAND2 U0,U2
S0=U3
U5 NAND2 A1,B1
U6 XOR2 A1,B1
U7 NAND2 U6,U4
U8 XOR2 U6,U4
U9 NAND2 U5,U7
S1=U8
U10 NAND2 A2,B2
U11 XOR2 A2,B2
U12 NAND2 U11,U9
U13 XOR2 U11,U9
U14 NAND2 U10,U12
S2=U13
U15 NAND2 A3,B3
U16 XOR2 A3,B3
U17 NAND2 U16,U14
U18 XOR2 U16,U14
U19 NAND2 U15,U17
S3=U18
U20 NAND2 A4,B4
U21 XOR2 A4,B4
U22 NAND2 U21,U19
U23 XOR2 U21,U19
U24 NAND2 U20,U22
S4=U23
Cout=U24

KOSTAS NIKOLOS
```

