



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη Εργαλείων CAD για Σχεδίαση Ολοκληρωμένων Κυκλωμάτων (HPY 419 - HPY 608)

PROJECT 3 – ΜΕΤΑΤΡΟΠΗ NETLIST ΑΠΟ SUBSYSTEMS ΣΕ ΠΥΛΕΣ

ΑΝΑΦΟΡΑ

ΜΠΙΜΠΙΡΗΣ ΠΕΤΡΟΣ (Α.Μ.: 2019030135)

Εισαγωγή:

Το παρόν project ήταν η υλοποίηση ενός εργαλείου που δέχεται ως είσοδο ένα netlist (για παράδειγμα το netlist εξόδου της προηγούμενης άσκησης) και δημιουργεί ένα αρχείο στο οποίο περιέχονται τα ίδια (από πλευράς λειτουργικότητας) υποσυστήματα με το αρχείο εισόδου, υλοποιημένα μόνο με πύλες.

Χρήση του εργαλείου

Το πρόγραμμα καλείται από την γραμμή εντολών (command line) ως εξής:

```
$ ./translate_to_gates.exe
```

και εκτελεί την λειτουργία του χωρίς κάποια αλληλεπίδραση με τον χρήστη. Αν η εκτέλεση ολοκληρωθεί επιτυχώς, τυπώνεται αντίστοιχο μήνυμα και το πρόγραμμα τερματίζει. Σε αντίθετη περίπτωση, τυπώνεται μήνυμα λάθους που πληροφορεί τον χρήστη για το σφάλμα που προέκυψε.

Για να λειτουργήσει σωστά το εργαλείο αυτό πρέπει να «γνωρίζει» όλα τα υποσυστήματα (subsystems) και τις πύλες που χρησιμοποιούνται στο netlist εισόδου, οπότε είναι απαραίτητο να βρίσκονται στην ίδια τοποθεσία με το εκτελέσιμο και οι βιβλιοθήκες που τα περιέχουν.

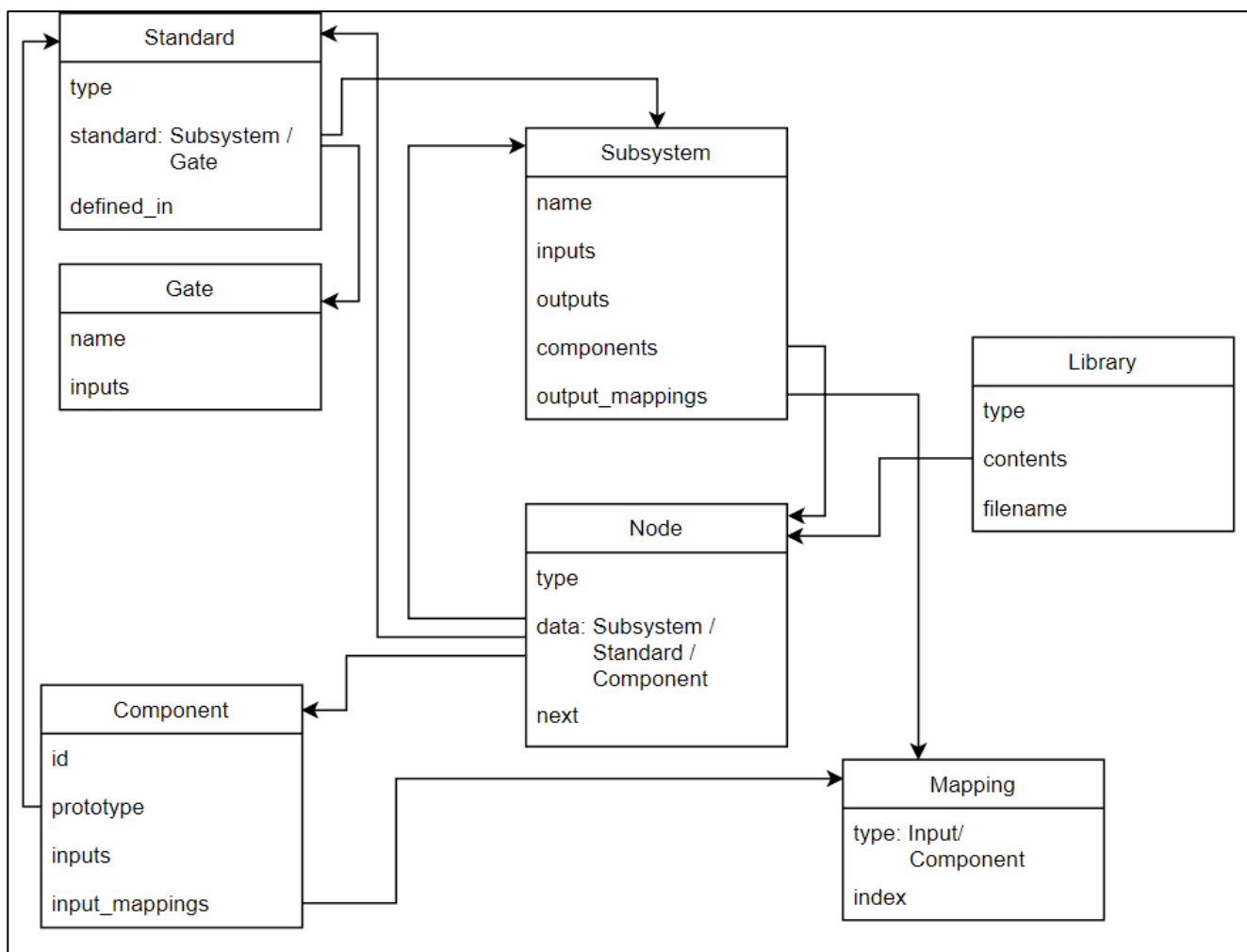
Τα ονόματα των βιβλιοθηκών αυτών, καθώς και των netlist εισόδου και εξόδου είναι παραμετροποιήσιμα μέσω των σταθερών (constants) που ορίζονται στο αρχείο `translate_to_gates.c`.

Δομή Υλοποίησης:

Ο κώδικας του εργαλείου βρίσκεται στο αρχείο `translate_to_gates.c`.

Χρησιμοποιούνται δομές και συναρτήσεις από την «βιβλιοθήκη» `netlist.h`, όπως και στην προηγούμενη άσκηση. Στην βιβλιοθήκη έχουν προστεθεί επιπλέον δομές και συναρτήσεις που υλοποιούν την επιπρόσθετη λειτουργικότητα που ήταν απαραίτητη για το παρόν project. Μια σύντομη περιγραφή των δομών που χρησιμοποιήθηκαν καθώς και των συνδέσεων μεταξύ τους φαίνεται παρακάτω στο διάγραμμα 1.

Για τον ίδιο λόγο προστέθηκαν οι αντίστοιχες συναρτήσεις στην βιβλιοθήκη χειρισμού strings, `str_util.h`.



Διάγραμμα 1

Υλοποίηση – Μετατροπή υποσυστημάτων σε πύλες:

Τα βήματα που ακολουθεί το πρόγραμμα προκειμένου να επιτευχθεί η επιθυμητή λειτουργικότητα περιγράφονται συνοπτικά παρακάτω (για μεγαλύτερη λεπτομέρεια, ο κώδικας περιέχει εκτενή σχόλια):

- 1) Αρχικά διαβάζονται από το `component library` οι πύλες που θεωρούνται «γνωστές» και μπορούν να χρησιμοποιηθούν για να κατασκευαστούν πιο περίπλοκα υποσυστήματα. Η λειτουργία αυτή εκτελείται από την συνάρτηση `gate_lib_from_file()`, η οποία:
 - ο διαβάζει το αρχείο γραμμή – γραμμή,
 - ο δημιουργεί για κάθε γραμμή ένα μια μεταβλητή τύπου `Gate`, η οποία «τοποθετείται» μέσα σε μια μεταβλητή τύπου `Standard`
 - ο προσθέτει την μεταβλητή τύπου `standard` στην βιβλιοθήκη (μεταβλητή τύπου `Library`)
- 2) Στην συνέχεια καλείται η `subsys_lib_from_file()`, η οποία ακολουθεί παρόμοια διαδικασία με την προηγούμενη, για να δημιουργήσει μια μεταβλητή τύπου `Library` που περιέχει ορισμούς (μεταβλητές τύπου `standard`) υποσυστημάτων. Οι κύριες διαφορές από την `subsys_lib_from_file()` είναι οι εξής:

- όταν βρεθεί γραμμή που ξεκινά με την λέξη COMP (δηλαδή γραμμή όπου δηλώνεται καινούριο subsystem), οι επόμενες γραμμές μέχρι και το END ... NETLIST χρησιμοποιούνται για να οριστεί το πρωτότυπο ενός υποσυστήματος.
 - συγκεκριμένα στην δομή μεταφέρονται πληροφορίες για το πως σχετίζονται οι είσοδοι κάθε υποσυστήματος με τις εισόδους του «εξωτερικού» υποσυστήματος ή με τις εξόδους των υπόλοιπων υποσυστημάτων – components
- 3) Στην συνέχεια διαβάζεται με παρόμοιο τρόπο και αποθηκεύεται σε μια δομή τύπου Library (ωστόσο αυτή τη φορά τα περιεχόμενα είναι τύπου Subsystem (και όχι Standard)) το netlist «εισόδου» που περιέχει τον n-bit full adder
- 4) Στην συνέχεια μετατρέπονται τα components του υποσυστήματος σε πύλες με την εξής διαδικασία (ή οποία επαναλαμβάνεται για κάθε single-bit full adder):
- αρχικά δημιουργείται ένα array όπου αποθηκεύονται όλοι οι single-bit full adders, αποτελούμενοι από τις πύλες τους.
 - στην συνέχεια «ερμηνεύονται» τα inputs από strings σε συνδέσεις με άλλα υποσυστήματα. Π.χ. το U3_COUT θα ερμηνευόταν σε έναν δείκτη στον full adder με ID U3 και στην έξοδο COUT.
 - στην συνέχεια με την βοήθεια του πρωτότυπου του single-bit full adder μετατρέπεται η κάθε απαραίτητη έξοδος (π.χ. COUT) στην πύλη με την οποία συνδέεται
 - με τα παραπάνω βήματα μετατρέπονται όλες οι συνδέσεις μεταξύ εξαρτημάτων (components) σε συνδέσεις μεταξύ πυλών.
- 5) Τέλος εκτυπώνονται στο αρχείο εξόδου όλες οι πύλες (ομαδοποιημένες κατά 5 – όσες υλοποιούν τον single-bit full adder – για ευκολία ανάγνωσης) καθώς και οι συνδέσεις των εξόδων

Σημείωση:

Η ανάπτυξη του κώδικα έγινε σε περιβάλλον Linux (συγκεκριμένα Ubuntu 20.04), οπότε και το Makefile είναι σχεδιασμένο αντίστοιχα. Σε οποιοδήποτε περιβάλλον Linux η εντολή make είναι κατά πάσα πιθανότητα αρκετή για να γίνουν σωστά compile τόσο οι βιβλιοθήκες όσο και το εργαλείο καθ' αυτό.