

**Αναφορά Πέμπτης εργαστηριακής άσκησης**  
**HPY 419 - HPY 608 - Ανάπτυξη Εργαλείων CAD για Σχεδίαση**  
**Ολοκληρωμένων Κυκλωμάτων 2022- 2023**  
**ΑΙΚΑΤΕΡΙΝΗ ΤΣΙΜΠΙΡΔΩΝΗ:2018030013**

**Εισαγωγή:**

Σε αυτό το κομμάτι της άσκησης μας ζητήθηκε να γίνει τελική σύνθεση των δύο προηγούμενων εργαστηρίων δηλαδή ξεκινώντας από το τέταρτο εργαστήριο και φτιάχνοντας την είσοδο για το τρίτο. Επίσης μας ζητήθηκε επέκταση δηλαδή να μπορεί να δέχεται 2 subsystem libraries ένα για full adder και ένα για full adder subtractor. Οπότε όταν εμείς δίνουμε ως είσοδο στο εργαστήριο 4 το entity είτε του full adder είτε του full adder subtractor, θα πρέπει να βγαίνει το σωστό netlist στην έξοδο. Καθώς και όταν μπαίνει ως είσοδος στο εργαστήριο 3 θα πρέπει να διαλέγει το σωστό subsystem library ώστε να βγάλει τις σωστές πύλες.

**Υλοποίηση:**

Αρχικά, έγιναν κάποιες αλλαγές στο εργαστήριο 3 στο κομμάτι της σχεδίασης μέσα στο subsystem library για μία καλύτερη σχεδίαση δηλαδή άλλαξα το netlist ώστε να είναι πιο κομψό. Οπότε το τελικό netlist για τον full adder 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 NAND2 U2,U3
```

```
S=U1
```

```
COUT=U4
```

Έπειτα για τη σωστότερη υλοποίηση του εργαστηρίου 3 έβγαλα τις αναρτήσεις που αφορούν τη δημιουργία των subsystem Library και το component library, διότι αρχικά είχα αντιληφθεί ότι πρέπει να δημιουργηθούν μέσα και μετά να διαβαστούν, όμως μετά από υπόδειξη κατάλαβα ότι δεν χρειάζεται να κατασκευαστούν μέσα στο πρόγραμμα οπότε οι συναρτήσεις αυτές αφαιρέθηκαν. Αυτές είναι οι επιμέρους αλλαγές που έγιναν καθαρά στο εργαστήριο 3 για βελτιστοποίηση.

Τώρα για το σκοπό της άσκησης δηλαδή την επέκταση, αρχικά στο εργαστήριο 3 προστέθηκε μία νέα δομή, η παρακάτω:

```
typedef struct subsystemFilesArray{  
    subsystemFile sub_array[MAX_COMPONENTS];  
    int sub_array_num;  
}  
subsystemFilesArray;
```

που στην ουσία περιέχει έναν πίνακα από SubSystemfiles ώστε να μπορούν να αποθηκεύονται περισσότερα από ένα subsystem libraries σε αυτή την περίπτωση έχουμε το full adder και του full adder subtractor.

Οπότε το νέο subsystem library περιέχει δύο περιγραφές ενός full adder και ενός full adder subtractor, και 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 NAND2 U2,U3
```

```
S=U1
```

```
COUT=U4
```

```
END FULL_ADDER NETLIST
```

```
** SUBSYSTEM LIBRARY
```

```
COMP FULL_ADDER_SUBTRACTOR; IN:A,B,C,CONTROL ; OUT: S, COUT
```

```
BEGIN FULL_ADDER_SUBTRACTOR NETLIST
```

```
U0 XOR2 B,CONTROL
```

```
U1 XOR2 A,U0
```

```
U2 XOR2 U1,C
```

```
U3 NAND2 C,U1
```

```
U4 NAND2 A,B
```

```
U5 NAND2 U3,U4
```

```
S=U2
```

```
COUT=U5
```

```
END FULL_ADDER_SUBTRACTOR NETLIST
```

Αυτό διαβάζετε και αποθηκεύεται μέσα στο πίνακα από sap System files, για την αποθήκευση αυτή φτιάχτηκε μία επιπλέον add συνάρτηση για αυτή τη δομή η add\_subfile\_array ,η οποία στην ουσία προσθέτει το SubSystemfile (όπως έχει ονομαστεί) στον πίνακα. Εγιναν μικροδιορθώσεις μέσα στον κώδικα ώστε να προσαρμοστεί αυτή η καινούργια δομή. Όμως ο κύριος κορμός παρέμεινε ο ίδιος δηλαδή δεν έγινε κάποια επιμέρους αλλαγή.

Όσον αφορά το εργαστήριο 4 το μόνο που άλλαξε ήταν να μπορεί να διαβαστεί και η Τετάρτη είσοδος για το full adder subtractor, δηλαδή προστέθηκε ελάχιστος κώδικας μέσα στη συνάρτηση makeOutputFile που στην ουσία κατασκευάζει το τελικό netlist του εργαστηρίου 4 δηλαδή προστέθηκε μία παραπάνω συνθήκη που λέει ότι όταν είσοδοι του entity είναι περισσότεροι από τον N τότε η είσοδος που περισσεύει θα μπει ως είσοδος σε όλα τα στοιχεία του τελικού netlist. Εμείς για παράδειγμα έχουμε N=3 αλλά αυτοί που δηλώνονται ως εισόδοι (IN) είναι τέσσερις οπότε θα μπει στο κομμάτι εκείνο που η 4η είσοδος θα μπει είσοδος σε όλα τα στοιχεία του netlist του full\_adder\_subtractor, και οι παραπάνω εισοδοι θα ακολουθήσουν κανονικά ότι ισχύει και για τον full adder που έχει αναλυθεί και στο προηγούμενο εργαστήριο.

Στην ουσία αυτές ήταν οι αλλαγές για το εργαστήριο 3 και 4 ώστε να μπορεί να γίνει επέκταση και η τελική σύνθεση.

Για την τελική σύνθεση κατασκευάστηκε ένα απλό script το οποίο τρέχει με σειριακό τρόπο τα εργαστήρια 3 και 4.

**Σημείωση:** ο κώδικας στο εργαστήριο 3 μπήκε μέσα σε ένα .c αρχείο το οποίο ονομάστηκε `ergastirio5.c` το εργαστήριο 4 παρέμενε μέσα στο ίδιο .c αρχείο (`ergastirio4.c`) και το τελικό script ονομάστηκε (`ergastirio5_final.c`) διευκρινίζεται για τυχόν παρανοήσεις.

### Αποτελέσματα:

Όταν βάζω entity:

```
ENTITY adder IS
```

```
VAR N = 3 %% Number of Bits in the Adder
```

```
LIB FULL_ADDER
```

```
PORT( IN A:{ SO, YOU,THINK}    %% A, MSB TO LSB
```

```
IN B:{ CAN, TELL,HEAVEN}    %% B, MSB TO LSB
```

```
IN Cin:{FROM}    %% Cin,
```

```
OUT S:{ HELL, BLUE,SKY}    %% S, MSB TO LSB
```

```
OUT COUT:{PAIN}    %% Cout,
```

```
)
```

```
END adder NETLIST
```

Το αποτέλεσμα που παίρνουν στην έξοδο είναι το παρακάτω:

```
U0 XOR2 THINK,HEAVEN
```

```
U1 XOR2 U0,FROM
```

```
U2 NAND2 FROM,U0
```

```
U3 NAND2 THINK,HEAVEN
```

```
U4 NAND2 U2,U3
```

```
U5 XOR2 YOU,TELL
```

```
U6 XOR2 U5,U4
```

```
U7 NAND2 U5,U4
```

```
U8 NAND2 YOU,TELL
```

```
U9 NAND2 U7,U8
```

```
U10 XOR2 SO,CAN
```

```
U11 XOR2 U10,U9
```

```
U12 NAND2 U10,U9
```

```
U13 NAND2 SO,CAN
```

```
U14 NAND2 U12,U13
```

```
SKY = U1
```

```
BLUE = U6
```

```
HELL = U11
```

```
PAIN = U14
```

Το όποιο είναι λειτουργικά σωστό το κρατούμενο μπαίνει κανονικά στο επόμενο και στο τέλος βγάζει τις σωστές εξόδους.

Ενώ όταν βάζω είσοδο το entity:

```
ENTITY adder_subtractor IS
VAR N = 3 %% Number of Bits in the Adder/Subtractor
LIB FULL_ADDER_SUBTRACTOR
PORT ( IN A:{ A2, A1,A0} %% A, MSB TO LSB
IN B:{ B2, B1,B0} %% B, MSB TO LSB
IN Cin:{ADD_SUB_MODE} %% Cin
IN ADD'_SUB:{ADD_SUB_MODE} %% Control Signal
OUT S:{S2, S1,S0} %% Sum, MSB TO LSB
OUT COUT:{COUT} %% Cout,
)
```

Βγάζει ως έξοδο:

```
U0 XOR2 B0,ADD_SUB_MODE
U1 XOR2 A0,U0
U2 XOR2 U1,ADD_SUB_MODE
U3 NAND2 ADD_SUB_MODE,U1
U4 NAND2 A0,B0
U5 NAND2 U3,U4
U6 XOR2 B1,ADD_SUB_MODE
U7 XOR2 A1,U6
U8 XOR2 U7,U5
U9 NAND2 U7,U5
U10 NAND2 A1,B1
U11 NAND2 U9,U10
U12 XOR2 B2,ADD_SUB_MODE
U13 XOR2 A2,U12
U14 XOR2 U13,U11
U15 NAND2 U13,U11
U16 NAND2 A2,B2
U17 NAND2 U15,U16
S0 = U2
S1 = U8
S2 = U14
COUT = U17
```

Βλέπουμε ότι το κύκλωμα είναι λειτουργικά σωστό και ακολουθεί το SubSystem library και βγάζει και τις σωστές εξόδους.

Εδώ το κρατούμενα και το επιπλέον control σήμα έχουν το ίδιο όνομα όπως φαίνεται και στο entity μπαίνουν κανονικά ,το control σήμα παραμένει ενώ το κρατούμενο μεταφέρεται κανονικά.