ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΚΑΤΑΣΚΕΥΑΣΤΙΚΗ ΕΡΓΑΣΙΑ:
Κωδικοποίηση συμβολοσειράς

Ακαδημαϊκό έτος 2014-2015

ΟΜΑΔΑ:

ΚΑΝΕΛΛΟΠΟΥΛΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ ΑΜ 03112193
ΜΠΟΥΓΟΣ – ΖΕΪΜΠΕΚΗΣ ΧΡΗΣΤΟΣ ΑΜ 03112405
ΘΕΟΦΙΛΑΤΟΣ ΔΗΜΟΣΘΕΝΗΣ ΑΜ 03112434

Περιγραφή και λειτουργία κυκλώματος

Για το συγκεκριμένο κύκλωμα εμπνευστήκαμε από τη συσκευή κωδικοποίησης που χρησιμοποιούσαν οι Γερμανοί για να επικοινωνούν στον $2^{ο}$ Παγκόσμιο πόλεμο, το Enigma. Ξεκινώντας, ο χρήστης έχει τη δυνατότητα να εισάγει ένα από τα γράμματα A,B,C,D στα 4 7-segment του κυκλώματος, με τη χρήση 4 πιεστικών διακοπτών. Εν συνεχεία, υπάρχουν 4 κλειδιά στο κύκλωμα μας με βάση τα οποία κωδικοποιείται το μήνυμα που έδωσε ο χρήστης. Τα 4 αυτά κλειδιά εναλλάσονται σε κάθε παλμό του ρολογιού. Πχ. αν ο χρήστης εισάγει την ακολουθία D B C A και το κλειδί μας είναι το A D C B τότε το μήνυμα που θα προκύψει είναι B D C A. Ουσιαστικά, αν ο χρήστης έχει βάλει το ν-οστό γράμμα του ABCD αλφάβητου τότε αυτό θα αλλάξει με το ν-οστό γράμμα του καινούργιου αλφάβητου-κλειδιού. Ο χρήστης έχει τη δυνατότητα με ένα διακόπτη να επιλέγει αν στα 7-segments του κυκλώματος θα εμφανίζεται η συμβολοσειρά που έχει εισαχθεί ή η κωδικοποιημένη μορφή της. Παρακάτω θα περιγράψουμε αναλυτικά τη σχεδίαση κάθε σταδίου.

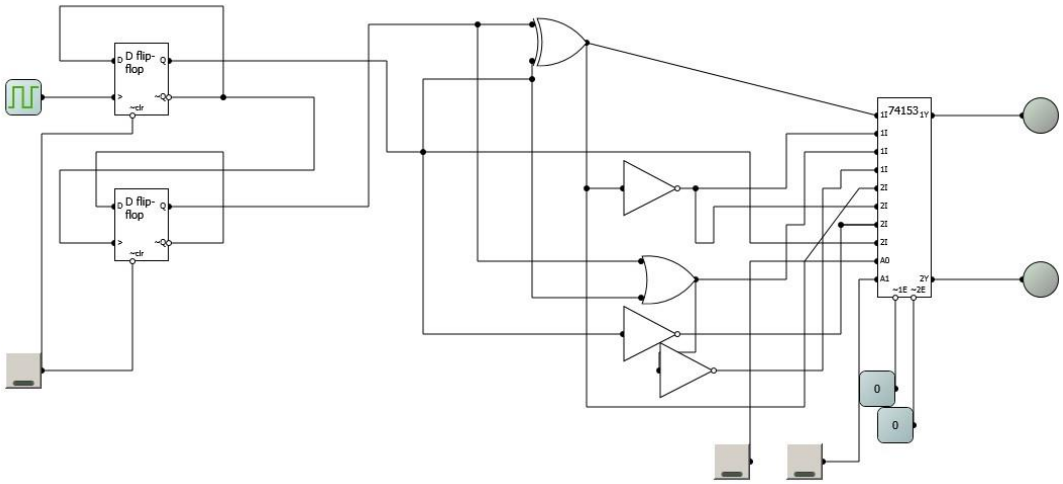
Δημιουργία Κλειδιού & Κρυπτογράφηση

Τα 4 κλειδιά που επιλέξαμε τυχαία, με μόνο κριτήριο να μην αντιστοιχούν περισσότερα του ενός γράμματος εισόδου στο ίδιο κωδικοποιημένο αποτέλεσμα.
(A=00, B=01, C=10, D=11)

Κλειδί	**k_MSB**	**k_LSB**	**A'**		**B'**		**C'**		**D'**	
0	0	0	0	0	0	1	1	1	1	0
1	0	1	1	1	0	0	1	0	0	1
2	1	0	1	1	0	1	1	0	0	0
3	1	1	0	0	1	0	0	1	1	1
Πύλη			**XOR**	**XOR**	**AND**	**k_LSB'**	**NAND**	**XNOR**	**XNOR**	**k_LSB**

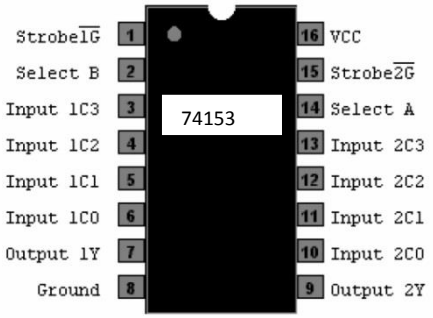
1ο κλειδί	A	B	D	C
2ο κλειδί	D	A	C	B
3ο κλειδί	D	B	C	A
4ο κλειδί	A	C	B	D

Το ακολουθιακό κύκλωμα, το οποίο σε κάθε παλμό του ρολογιού θα αλλάζει το κλειδί, θα το υλοποιήσουμε με 2 flip-flops σε διάταξη δυαδικού μετρητή ριπής (προς τα πάνω). Θα χρησιμοποιήσουμε το ολοκληρωμένο 74LS74 με τα 2 flip-flops.



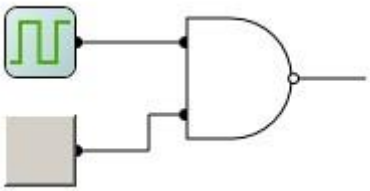
Ακολουθούν διευκρινίσεις για το κύκλωμα:

- Στο μετρητή ριπής το κουμπί που χρησιμοποιούμε είναι για το clear.
- Χρησιμοποιούμε το ολοκληρωμένο 74LS153 το οποίο είναι ένας διπλός πολυπλέκτης 4 σε 1 (με κοινό selector). Στο 1C0 ($6^{ο}$ pin) και στο 2C0 ($10^{ο}$ pin) θα μπουν τα 2 bits του γράμματος Α, τα οποία πρέπει να αντικαταστήσουν το 00 = Α. Αντίστοιχα χρησιμοποιούμε και τα υπόλοιπα.
- Στα pins 14 και 2 θα μπαίνουν τα bits του γράμματος που έχει δώσει ο χρήστης .
- Τα pins 1 και 15 είναι στο λογικό 0 ώστε να δουλεύουν και οι 2 έξοδοι του πολυπλέκτη.
- Χάρη σε αυτό το ολοκληρωμένο μπορούμε εύκολα να υλοποιήσουμε την κρυπτογράφηση (χρειαζόμαστε 4 από αυτά, 1 για κάθε 7-segment), ενώ σε άλλη περίπτωση θα έπρεπε να χρησιμοποιήσουμε πιο πολλά από 4 ολοκληρωμένα (θα μπορούσαμε με αποκωδικοποιητή 2-4 και τρισταθή).



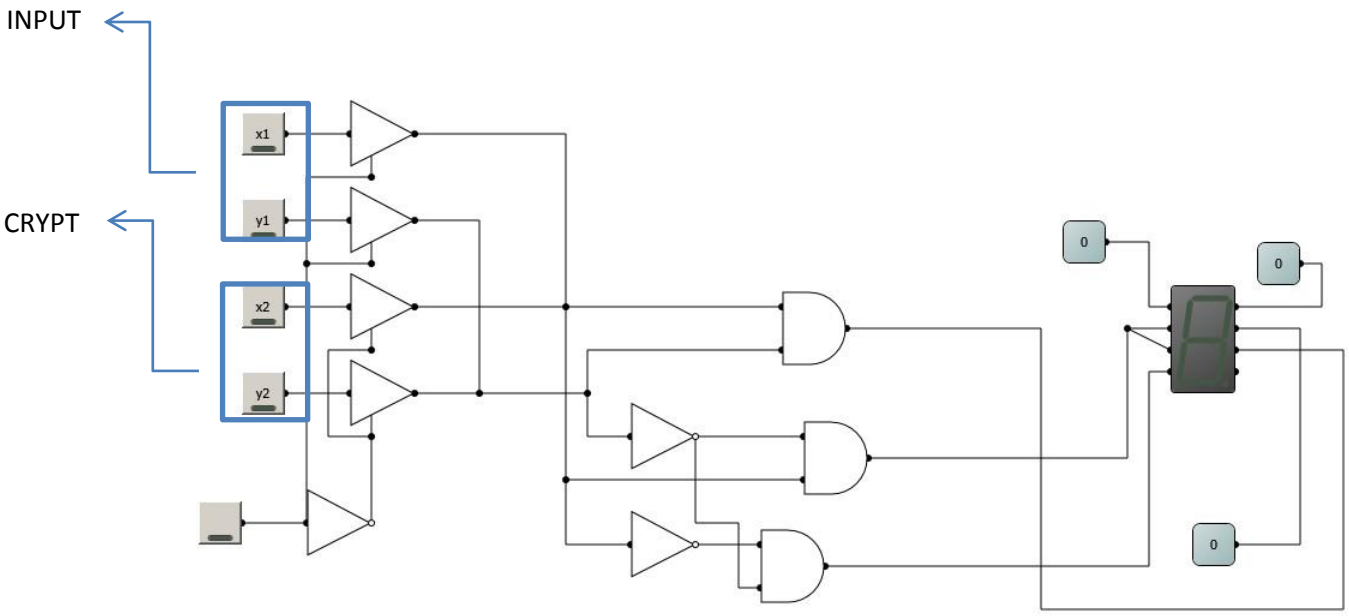
Εμφάνιση κρυπτογραφημένου στοιχείου & εισόδου σε μορφή γράμματος

Ο χρήστης με τη βοήθεια ενός πιεστικού διακόπτη θα μπορεί να αλλάζει την είσοδο. Κάθε φορά που πατάει τον διακόπτη,το κύκλωμα θα εμφανίζει το επόμενο γράμμα. Θα χρησιμοποιήσουμε και πάλι έναν μετρητή ριπής με 2 flip-flop, το ρολόι του οποίου θα ενεργοποιείται με το πάτημα του πιεστικού. Θα μπορούσαμε με απλούς διακόπτες να υλοποιήσουμε το κύκλωμα. Στην περίπτωση των 2 bits έχει λογική. Αν όμως είχαμε 8 bits θα είχαμε στο κύκλωμα 32 διακόπτες, το οποίο δε διευκολύνει τον κατασκευαστή και σε καμία περίπτωση τον χρήστη. Επίσης, θα μπορούσαμε να χρησιμοποιήσουμε το ρολόι των κλειδιών και έναν πιεστικό με την εξής συνδεσμολογία:



ώστε όταν ο διακόπτης δεν είναι πατημένος να έχω συνέχεια 1 στην έξοδο, ενώ όταν είναι πατημένος θα έχω το (CLK)’ σαν αποτέλεσμα και δεν θα χρειαστεί να τον πατήσω ν φορές για να αλλάξω ν φορές το γράμμα, αλλά 1 δίνοντας βέβαια προσοχή στο πότε θα τον αφήσω). Το πρόβλημα εδώ είναι ότι αν έχω συχνότητα λίγο μεγαλύτερη του 1 Hz δυσκολεύω το χρήστη να “πετύχει” το γράμμα που θέλει. Έτσι, χρειάζομαι ουσιαστικά και ένα ακόμα ρολόι με σταθερά μικρή συχνότητα, το οποίο να αφορά μόνο τα 4 γράμματα που δίνει ο χρήστης. (Μπορεί τελικά να το υλοποιήσουμε έτσι). Για να δώσω τη δυνατότητα στον χρήστη να επιλέγει αν θα βλέπει την είσοδο ή το κρυπτογραφημένο μήνυμα, θα βάλουμε έναν απλό διακόπτη ο οποίος θα αποτελεί είσοδο για τις επιτρέψεις τρισταθών. Όταν ο διακόπτης είναι στο λογικό 0, τότε θα ενεργοποιούνται τα τρισταθή που έχουν στην είσοδο τα γράμματα που έδωσε ο χρήστης. Στην αντίθετη περίπτωση θα ενεργοποιούνται τα τρισταθή με το κρυπτογραφημένο μήνυμα. Σε κάθε περίπτωση, πρέπει να μετατρέπουμε τα 2 bits σε ένα γράμμα. Δε θα χρησιμοποιήσουμε αποκωδικοποιητή, αλλά θα υπολογίσουμε τις 7 συναρτήσεις που χρειάζονται για το 7-segment. (Τα γράμματα θεωρώ ότι είναι όλα κεφαλαία)

X	**Y**	**f**	**G**	**b**	**a**	**c**	**d**	**e**
0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0
1	1	0	1	0	0	0	0	0
		0	AND	XY’	0	XY’	X’Y’	0



Τα x1,x2 είναι τα MSB.Οι 7 συναρτήσεις θα υλοποιηθούν με 2 ολοκληρωμένα (1 AND και 1 NOT). Για τα τρισταθή θα χρησιμοποιήσουμε το 74LS125 (στο simulation τα τρισταθή όπως και το 7-segment **δεν** είναι αντίστροφης λογικής).

<u>Μερικές παρατηρήσεις:</u>

- Θα χρησιμοποιήσουμε ένα επιπλέον 7-segment ώστε να ξέρουμε κάθε στιγμή σε ποιό κλειδί βρισκόμαστε.
- Σε κάθε 7-segment θα χρησιμοποιήσουμε 7 αντιστάσεις και όχι 1 όπως στην εργαστηριακή άσκηση, ώστε να έχουμε έντονη και σταθερή φωτοβολία σε κάθε περίπτωση.
- Για τους πιεστικούς διακόπτες θα χρησιμοποιήσουμε πύλες NAND.
- Στο ρολόι των κλειδιών θα έχουμε την επιλογή και για χρήση χειροκίνητου ρολογιού, δηλαδή πιεστικού διακόπτη.