

Καταχωρητές και Μετρητές

Σχεδιασμός ενός Οικουμενικού Καταχωρητή Ολισθησης, Μετρητή Δακτυλίου και Μετρητή BCD.

ΕΡΓΑΣΤΗΡΙΟ ΥΛΙΚΟΥ/ΛΟΓΙΣΜΙΚΟΥ 5

Εισαγωγή

Στην προηγούμενη εργαστηριακή άσκηση, δείξαμε πως χρησιμοποιούμε ένα flip-flop ως στοιχείο μνήμης ικανό για την αποθήκευση ενός bit πληροφορίας, και πως τα JK flip-flops μπορούν να χρησιμοποιηθούν για τον σχεδιασμό ενός modulo 16/10 μετρητή. Ολοκληρωμένα σύνολα από flip-flops, τα οποία εκτελούν πράξεις σύμφωνα με κοινά σήματα ελέγχου ονομάζονται ΚΑΤΑΧΩΡΗΤΕΣ (REGISTERS) ή ΜΕΤΡΗΤΕΣ (COUNTERS). Ένας καταχωρητής είναι μια διάταξη μνήμης που χρησιμοποιείται για την αποθήκευση και χειρισμό πολλαπλών bit πληροφορίας. Οι καταχωρητές χωρίζονται σε κατηγορίες αναλόγως του τρόπου με τον οποίο οι πληροφορίες εισάγονται ή αφαιρούνται από αυτούς. Ένας σειριακός (serial) καταχωρητής ή καταχωρητής ολισθησης (shift register) είναι μια διάταξη όπου τα δεδομένα εισάγονται ανά ένα bit, ενώ ο παράλληλος (parallel) καταχωρητής δέχεται ή μεταφέρει όλα τα bit δεδομένων ταυτόχρονα. Ένας σύγχρονος μετρητής είναι ένα συγκεκριμένο είδος καταχωρητή το οποίο περιοδικά αλλάζει την τιμή των δεδομένων που περιέχει, ακολουθώντας μια συγκεκριμένη σειρά αριθμών (καταστάσεων) σε κάθε ακμή του ρολογιού (θυμηθείτε, για παράδειγμα, τον μετρητή ριπής που σχεδιάσατε στον προηγούμενο εργαστήριο).

Στην άσκηση αυτή θα μελετήσουμε και θα σχεδιάσουμε ένα «οικουμενικό» καταχωρητή ολισθησης (universal shift register) ο οποίος δέχεται και διαχειρίζεται δεδομένα τόσο σειριακά όσο και παράλληλα. Επίσης, θα σχεδιάσουμε ένα Αμφίδρομο Μετρητή Δακτυλίου (bi-directional ring counter) 4^{ov} bit και ένα modulo 100 BCD μετρητή ριπής.

Για την άσκηση αυτή χρειάζονται 2 *Προσπαιτούμενες Προετοιμασίες*, όπως φαίνεται στις σελίδες 5 και 7. Αυτές πρέπει να πραγματοποιηθούν κατά τη διάρκεια του εργαστηρίου.

Μέρος Α (Πρώτη Εβδομάδα)

Σημείωση: Τα πιο κάτω αρχεία θα χρησιμοποιηθούν στην άσκηση αυτή. Πριν ξεκινήσετε κατεβάστε τα από την ιστοσελίδα του μαθήματος στον χώρο σας σε ένα κατάλογο με το όνομα **lab5**.

ushift4.vhd ushift4.scf dualmux4x1.vhd quaddff.vhd ushift4_struct.vhd ushift4_struct.scf lab5b.scf

I. Μέρος Α.Ι: Σχεδιασμός ενός Οικουμενικού Καταχωρητή Ολισθησης.

Ο Πίνακας Αληθείας και το Σχηματικό Σύμβολο του οικουμενικού καταχωρητή ολισθησης 4 bits 74194¹ (4-bit universal shift register) φαίνονται στην επόμενη σελίδα. Η ονομασία «οικουμενικός» προέρχεται από το γεγονός ότι περιλαμβάνει όλα τα χαρακτηριστικά που θα μπορούσε να απαιτεί ένας σχεδιαστής από έναν καταχωρητή και έτσι τον καθιστούν σημαντικό δομικό στοιχείο σχεδίασης. Το κύκλωμα διαθέτει παράλληλες εισόδους και εξόδους, σειριακές εισόδους αριστερής και δεξιάς ολισθησης, εισόδους ελέγχου της κατάστασης της λειτουργίας και μια γραμμή για άμεση αρχικοποίηση (asynchronous initialization). Τα δύο σήματα ελέγχου της κατάστασης λειτουργίας (mode control) S_1 και S_0 καθορίζουν τις ακόλουθες τέσσερις καταστάσεις λειτουργίας:

- $(S_1, S_0) = 0\ 0$: Διατηρεί την τρέχουσα κατάσταση (καμιά αλλαγή).
- $0\ 1$: Δεξιά Ολισθηση (με κατεύθυνση από το QA προς το QD)
- $1\ 0$: Αριστερή Ολισθηση (με κατεύθυνση από το QD προς το QA)
- $1\ 1$: Παράλληλη Φόρτωση των A, B, C, D στα QA, QB, QC, QD

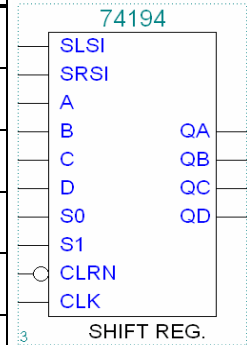
¹ Το 74194 είναι λειτουργικά ισοδύναμο με το 74LS194A. Στο εργαστήριο υλικού θα χρησιμοποιηθεί το 74LS194A.

Οι λειτουργίες ολίσθησης και φόρτωσης γίνονται στην θετική ακμή της εισόδου του ρολογιού το οποίο ενώνεται εσωτερικά με όλα τα flip-flops. Μελετήστε τους πίνακες για να καταλάβετε τις λειτουργίες.

74194 Οικουμενικός Καταχωρητής Ολίσθησης 4-Bit

Θετικά Ακμοπυροδοτούμενος με Σειριακές και Παράλληλες Εισόδους και Ασύγχρονο Clear

CLRΝ	MODE		CLK	SERIAL IN		PARALLEL IN				OUTPUTS				ACTION
	S1	S0		SLSI	SRSI	A	B	C	D	QA	QB	QC	QD	
0	X	X	X	X	X	X	X	X	X	0	0	0	0	Reset
1	X	X	≠↑	X	X	X	X	X	X	QA	QB	QC	QD	Retain
1	1	1	↑	X	X	a	b	c	d	a	b	c	d	Parallel Load
1	0	1	↑	X	0	X	X	X	X	0	QA	QB	QC	Shift Right
1	0	1	↑	X	1	X	X	X	X	1	QA	QB	QC	Shift Right
1	1	0	↑	0	X	X	X	X	X	QB	QC	QD	0	Shift Left
1	1	0	↑	1	X	X	X	X	X	QB	QC	QD	1	Shift Left
1	0	0	X	X	X	X	X	X	X	QA	QB	QC	QD	Retain

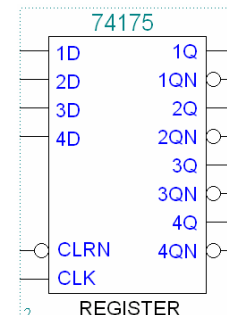
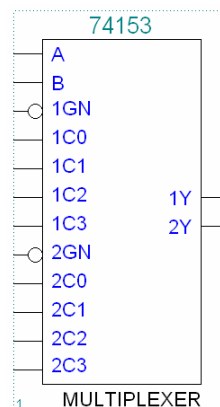


74153 Δύο 4-σε-1 Πολυπλέκτες

SELECT		DATA INPUTS				EN	OUT
B	A	C0	C1	C2	C3	GN	Y
X	X	X	X	X	X	1	0
0	0	0	X	X	X	0	0
0	0	1	X	X	X	0	1
0	1	X	0	X	X	0	0
0	1	X	1	X	X	0	1
1	0	X	X	0	X	0	0
1	0	X	X	1	X	0	1
1	1	X	X	X	0	0	0
1	1	X	X	X	1	0	1

74175 Καταχωρητής Τεσσάρων D Flip-Flop με Clear

CLRΝ	D	CLK	Q ⁺	QN ⁺
0	X	X	0	1
1	X	≠↑	Q	QN
1	0	↑	0	1
1	1	↑	1	0



Ένα VHDL entity και η behavioral αρχιτεκτονική για τον οικουμενικό καταχωρητή ολίσθησης φαίνονται πιο κάτω. Επιβεβαιώστε ότι η πιο κάτω περιγραφή ισοδυναμεί με την συμπεριφορά που περιγράφεται στην προηγούμενη σελίδα για το στοιχείο αυτό.

```
-- ushift4.vhd
-- Model for 4-bit universal shift register 74194
-- ECE-211 Experiment LAB5A

library ieee;
use ieee.std_logic_1164.ALL;

ENTITY ushift4 IS
    PORT (clk, clrn, slsi, srsi : in bit;
          mode : in bit_vector ( 1 downto 0 );
          d : in bit_vector ( 3 downto 0 );
          q : inout bit_vector (3 downto 0 ));
END ushift4;

ARCHITECTURE behavior OF ushift4 IS
BEGIN
    PROCESS (clk, clrn)
    begin
        -- Asynchronous, active-low Clear input:
        if clrn = '0' then
            q <= "1111";
        -- Rising edge-triggered D flip-flops:
        elsif clk'event and clk = '1' then
            case mode is
                -- "Do Nothing" mode: retain current flip-flop outputs
                when "00" => null;
                -- Shift Right Serial Input mode:
                when "01" =>
                    q <= srsi & q(3 downto 1);
                -- Shift Left Serial Input mode:
                when "10" =>
                    q <= q(2 downto 0) & slsi;
                -- Parallel (Broadside) Load mode:
                when "11" =>
                    q <= d;
            end case;
        end if;
    end process;
END behavior;
```

a) Προσομοίωση της Behavioral Αρχιτεκτονικής του Καταχωρητή Ολίσθησης

Στην άσκηση αυτή θα φορτώσουμε, θα μεταφράσουμε και θα προσομοιώσουμε τον πιο πάνω κώδικα.

1. Αντιγράψετε το αρχείο `ushift4.vhd` και το αρχείο με την κυματομορφή `ushift4.scf` στον κατάλογο **lab5** στον χώρο εργασίας σας.
2. Μεταφράστε τον σχεδιασμό `ushift4` με επιλεγμένη μόνο την λογική προσομοίωση.
3. Εκτελέστε την προσομοίωση με την χρήση του αρχείου `ushift4.scf`.
4. Κρατήστε μιαν εκτύπωση των αποτελεσμάτων της προσομοίωσης.

Επιβεβαιώστε ότι τα αποτελέσματα σας αντιστοιχούν στην συμπεριφορά του 74194 που περιγράφηκε στην σελίδα 1.

Προσομοίωση της Structural Αρχιτεκτονικής του Καταχωρητή Ολίσθησης

Για καλύτερη αντίληψη του καταχωρητή ολίσθησης 74194 και για να εξοικειωθείτε περισσότερο με τους πολυπλέκτες και τα flip-flops τα οποία είδαμε σε προηγούμενες ασκήσεις, θα κληθείτε να αλλάξετε ένα VHDL αρχείο structural αρχιτεκτονικής έτσι που, με την χρήση δύο 74153 (δύο 4-σε-1 πολυπλέκτες) και ενός 74175 (τέσσερα D flip-flops) να κατασκευάσετε ένα κύκλωμα που να εκτελεί ακριβώς την ίδια λειτουργία και να έχει τις ίδιες εισόδους με ένα 74179A οικουμενικό καταχωρητή ολίσθησης.

Οι πίνακες αληθείας και τα σύμβολα που χρησιμοποιεί το MAX+PLUS II για τα ολοκληρωμένα 74153 και 74175 δίνονται στην σελίδα 2 της άσκησης αυτής. Μελετήστε τα αρχεία που ακολουθούν για να κατανοήσετε πως περιγράφουν τα σύμβολα και την συμπεριφορά που παρουσιάστηκε στην σελίδα 2.

```
-- dualmux4x1.vhd
-- Model for Dual four to one multiplexer 74153
-- ECE-211 Experiment LAB5A

library ieee;
use ieee.std_logic_1164.ALL;

ENTITY dualmux4x1 IS
    PORT (gb1, gb2 : in bit;
          sel : in bit_vector ( 1 downto 0 );
          d1, d2 : in bit_vector ( 3 downto 0 );
          y1, y2 : out bit);
END dualmux4x1;

ARCHITECTURE behav OF dualmux4x1 IS
    BEGIN
        PROCESS (sel, gb1, gb2, d1, d2)
            begin
                if gb1 = '1' then y1 <= '0';
                elsif sel = "00" then
                    y1 <= d1(0);
                elsif sel = "01" then
                    y1 <= d1(1);
                elsif sel = "10" then
                    y1 <= d1(2);
                elsif sel = "11" then
                    y1 <= d1(3);
                end if;
                if gb2 = '1' then y2 <= '0';
                elsif sel = "00" then
                    y2 <= d2(0);
                elsif sel = "01" then
                    y2 <= d2(1);
                elsif sel = "10" then
                    y2 <= d2(2);
                elsif sel = "11" then
                    y2 <= d2(3);
                end if;
            end process;
        end process;
```

```

-- quaddff.vhd
-- Model for Quad D Flip-Flop 74175

library ieee;
use ieee.std_logic_1164.ALL;

ENTITY quaddff IS
    PORT (clkf, rstb : in bit;
          df : in bit_vector (3 downto 0);
          qf : inout bit_vector (3 downto 0));
END quaddff;

ARCHITECTURE behav OF quaddff IS
    signal Q: bit_vector(3 downto 0);
BEGIN
    PROCESS (clkf, rstb)
    begin
        if rstb = '0' then
            qf <= "0000";
        elsif clkf'event and clkf = '1' then
            qf <= df;
        end if;
    end process;
END behav;

```

Προαπαιτούμενη Προετοιμασία 1:

Σχεδιάστε το σχηματικό διάγραμμα ενός σχεδιασμού, όπως καθορίζεται στην αρχή της σελίδας 4, χρησιμοποιώντας τα στοιχεία και τις θύρες σημάτων που δίνονται από τα αρχεία VHDL *dualmux4x1* και *quaddff*.

Ένας ημιτελής κώδικας VHDL structural αρχιτεκτονικής για τον Οικουμενικό Καταχωρητή Ολίσθησης φαίνεται στην επόμενη σελίδα. Χρησιμοποιήστε το σχηματικό σας διάγραμμα για να συμπληρώσετε τις γραμμές που λείπουν από τον κώδικα VHDL.

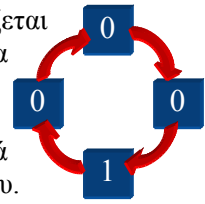
5. Αντιγράψτε τα αρχεία VHDL, *dualmux4x1.vhd*, *quaddff.vhd*, *ushift4_struct.vhd* και *ushift4_struct.scf* από την σελίδα του μαθήματος στο κατάλογο σας *lab5*.
 6. Χρησιμοποιήστε τα αποτελέσματα της *Προαπαιτούμενης Προετοιμασίας 1* για να αλλάξετε το αρχείο *ushift4_struct.vhd*. Κρατήστε μια εκτύπωση του αλλαγμένου αρχείου.

Ανατρέξτε στην επόμενη σελίδα για το αρχείο. Ο ευκολότερος τρόπος για να αλλάξετε το αρχείο είναι να το ανοίξετε από τον editor του MAX+PLUS II.
 7. Αφού έχετε αλλάξει κατάλληλα το structural κώδικα του καταχωρητή, μεταφράστε τον (για λογική προσομοίωση) και εκτελέστε την προσομοίωση.
 8. Προσομοιώστε με την χρήση του αρχείου *ushift4_struct.scf*. Εκτυπώστε το αποτέλεσμα.

Παρατηρήστε ότι στην προσομοίωση περιλαμβάνεται ένας «αφανής» κόμβος, ο οποίος αντιστοιχεί στις εισόδους των D flip-flops από τις οποίες μπορείτε να δείτε την επίδραση των χρονισμένων flip-flops στην έξοδο Q.
-

II. Μέρος A.II : Σύγχρονος Αμφίδρομος Μετρητής Δακτυλίου 4-Bit.

Ένας μετρητής δακτυλίου αποτελείται από ένα καταχωρητή ολίσθησης ο οποίος σχεδιάζεται με τέτοιο τρόπο ώστε, το bit που ολισθαίνει προς τα έξω από την μία πλευρά του να εισάγεται στην άλλη. Ο μετρητής αρχικοποιείται έτσι που **ακριβώς ένα bit** στο καταχωρητή να είναι 1 και όλα τα άλλα να είναι μηδέν. Σε διαδοχικές θετικές ακμές του ρολογιού, το 1 μεταφέρεται κυκλικά στο «δακτύλιο» του καταχωρητή. Στα δεξιά μπορείτε να δείτε μια γραφική αναπαράσταση της λειτουργίας του καταχωρητή δακτυλίου.



Ένας Αμφίδρομος Μετρητής Δακτυλίου (bi-directional ring counter) διαθέτει ένα σήμα ελέγχου κατάστασης το οποίο ελέγχει την κατεύθυνση της ολίσθησης. Στο πιο πάνω παράδειγμα, η έξοδος του δεξιότερου flip-flop ενώνεται με την είσοδο του αριστερότερου flip-flop. Μπορεί, όμως, να συμβεί και το αντίθετο, δηλ., η έξοδος του αριστερότερου flip-flop να ενωθεί με την είσοδο του δεξιότερου flip-flop. Η λειτουργία της παράλληλης φόρτωσης για το στοιχείο αυτό χρησιμοποιείται για τον καθορισμό της αρχικής κατάστασης του μετρητή. Ένας αμφίδρομος μετρητής δακτυλίου μπορεί να υλοποιηθεί χρησιμοποιώντας έναν οικουμενικό καταχωρητή ολίσθησης 74194.

Προαπαιτούμενη Διαδικασία 2:

Σχεδιάστε στο χαρτί έναν αμφίδρομο μετρητή δακτυλίου, όπως καθορίστηκε στην προηγούμενη παράγραφο με την χρήση του οικουμενικού καταχωρητή ολίσθησης 74194 και τα σήματα που δίνονται στο μέρος a.3 πιο κάτω. Ο μετρητής πρέπει να έχει τις εξής λειτουργίες:

- (i) MODE = 00 – διατηρεί την τρέχουσα κατάσταση,
- (ii) MODE = 01 – μετρητής δακτυλίου προς δεξιά,
- (iii) MODE = 10 – μετρητής δακτυλίου προς αριστερά,
- (iv) MODE = 11 – αρχικοποιεί τον μετρητή με την τιμή 1000.

a) Κατασκευή και Προσομοίωση Σχηματικού

1. Δημιουργήστε ένα νέο αρχείο στον Graphic Editor με όνομα **lab5b.gdf** και αποθηκεύστε το στον κατάλογο σας **lab5**.
2. Εισάγεται τα ακόλουθα στοιχεία από το παράθυρο διαλόγου Symbol Dialog: **input**, **output**, **gnd**, **vcc** και **74194**.
3. Αντιγράψτε αναλόγως τα στοιχεία, ώστε να έχετε 3 θύρες εισόδου στα αριστερά και μία θύρα εξόδου στα δεξιά του σχηματικού σας. Δώστε τα ακόλουθα ονόματα στις θύρες: **RESETN**, **CLK**, **MODE[1..0]** και **COUNT[3..0]**

Σημειώστε: ΠΡΕΠΕΙ να χρησιμοποιήσετε τα ονόματα αυτά ώστε να συμβαδίζει το σχηματικό σας με το αρχείο κυματομορφών που σας έχει δοθεί.

4. Συνδέστε τα στοιχεία μεταξύ τους σύμφωνα με το σχεδιασμό σας. Χρησιμοποιήστε δί-αυλους για την έξοδο **COUNT[3..0]** και την είσοδο **MODE[1..0]**.
5. Ελέγξτε, αποθηκεύστε και μεταφράστε τον σχεδιασμό με επιλεγμένη μόνο την λογική προσομοίωση. Κρατήστε μια εκτύπωση του σχηματικού σας.
6. Αντιγράψτε το αρχείο κυματομορφής, **lab5b.scf** στο κατάλογο σας **lab5**.
7. Προσομοιώστε το project **lab5b** και κρατήστε μια εκτύπωση των αποτελεσμάτων της. Μελετήστε τις κυματομορφές εξόδου και σχολιάστε τα αποτελέσματα.

III. Μέρος A.III: BCD Μετρητής Ριπής Modulo 100.

Το σχηματικό στην τελευταία σελίδα αυτής της άσκησης περιγράφει ένα Binary-Coded-Decimal Μετρητή δύο ψηφίων χρησιμοποιώντας μόνο ένα ολοκληρωμένο 74390. Κάθε ψηφίο BCD υλοποιείται με την διαδοχική σύνδεση ενός μετρητή ενός bit που διαιρεί δια 2 και ενός μετρητή 3^{ov} bit που διαιρεί δια 5, κατασκευάζοντας έτσι ένα μετρητή ριπής που μετράει από το δεκαδικό 00 έως το 99.

1. Με την χρήση του Graphic Editor υλοποιήστε τον μετρητή αυτό σε ένα αρχείο με το όνομα `lab5c.gdf` στον κατάλογο `lab5`. Ονομάστε τις δύο εισόδους `CLR` και `CNTB` και την έξοδο `COUNT[7..0]`. Ελέγξτε και αποθηκεύστε το σχηματικό και κρατήστε μια εκτύπωση του.
2. Δημιουργήστε μια κυματομορφή διάρκειας 11200ns στο αρχείο `lab5c.scf` για την προσομοίωση του μετρητή σας. Δημιουργήστε ένα ρολόι με περίοδο 100ns για το κόμβο `CNTB`. Καθορίστε τον κόμβο `CLR` έτσι που ο μετρητής να μηδενίζεται την χρονική στιγμή 0ns.
3. Προσομοιώστε τον σχεδιασμό σας και κρατήστε μια εκτύπωση για τα χρονικά διαστήματα από 0 έως 1300ns και από 8800 έως 10400 ns.

Αναφορά Μέρους A

Η γραπτή σας αναφορά για το μέρος A πρέπει να περιέχει (i) σχολιασμό για την λειτουργία του κάθε καταχωρητή και μετρητή που σχεδιάστηκε και (ii) σχολιασμό για τα σχετικά πλεονεκτήματα και μειονεκτήματα των καταχωρητών και των μετρητών που μελετήσατε στην άσκηση αυτή. Για το σκοπό αυτό, παραπέμπεστε στα ακόλουθα σχηματικά και αποτελέσματα προσομοιώσεων τα οποία είναι απαραίτητο να συμπεριλάβετε στην αναφορά:

1. Κυματομορφές που προέκυψαν από την προσομοίωση του behavioral μοντέλου του καταχωρητή ολίσθησης στην παράγραφο a.4 του A.I, στην σελίδα 3.
2. Ο Κώδικας VHDL για το αλλαγμένο `ushift4_struct.vhd` στο b.2 του A.I στη σελίδα 5.
3. Κυματομορφές που προέκυψαν από την προσομοίωση του structural μοντέλου του καταχωρητή ολίσθησης στο b.4 του A.I., στην σελίδα 5.
4. Το σχηματικό του αμφίδρομου μετρητή δακτυλίου στο a.5 (A.II), στην σελίδα 7.
5. Κυματομορφές που προέκυψαν από την προσομοίωση του μετρητή αμφίδρομου δακτυλίου στο a.7 (A.II) στην σελίδα 7.
6. Σχηματικό του μετρητή BCD Modulo 100 από το 1 του A.III στην σελίδα 8.
7. Δύο κυματομορφές που προέκυψαν από την προσομοίωση του μετρητή BCD Modulo 100 από το 3 του A.III, στην σελίδα 8.

Επίσης αναφέρατε οποιεσδήποτε δυσκολίες έχετε συναντήσει στην χρήση των εργαλείων του MAX+PLUS II και εξηγήστε τι κάνατε για να τις αντιμετωπίσετε.

Μέρος Β(α) (Δεύτερη Εβδομάδα)

Σκοπός αυτής της εργαστηριακής άσκησης είναι να διαπιστωθεί πειραματικά η λειτουργία του Σύγχρονου Αμφίδρομου Μετρητή Δακτυλίου, ο οποίος προσομοιώθηκε στο μέρος Α.ΙΙ και του μετρητή BCD Modulo 100, από το μέρος Α.ΙΙΙ. Πρώτα θα κατεβάσετε και θα ελέγξετε τον Σύγχρονο Αμφίδρομο Μετρητή Δακτυλίου. Η έξοδος του Καταχωρητή Δακτυλίου θα παρουσιάζεται σε ένα από τα SSDs. Ακολούθως, θα κατεβάσετε τον BCD μετρητή στο FLEX 10K και οι έξοδοι θα παρουσιάζονται στα δύο SSDs.

α) Σύγχρονος Αμφίδρομος Μετρητής Δακτυλίου

1. Ανοίξετε το σχηματικό **lab5b.gdf** schematic.
2. Αντιγράψετε τα περιεχόμενα της βιβλιοθήκης **UP1core library functions** στον κατάλογο σας **lab5**.
3. Εισάγεται το σύμβολο **clk_div** στον σχεδιασμό σας και συνδέστε κατάλληλα το κύκλωμα ώστε το σήμα του ρολογιού να μειωθεί από τα 25.2MHz στο 1 Hz.
4. Εισάγεται δύο πύλες **not** στον σχεδιασμό σας και τοποθετήστε τις κατάλληλα ώστε να αντιστρέφονται οι εισοδοί **MODE**.

Ένα μέρος του SSD θα παρουσιάζει τον δακτύλιο με ένα από τα κομμάτια του δείχνει την θέση του bit που είναι 1. Ο δακτύλιος θα αποτελείται από τα μέρη **a, b, f** και **g** του SSD.

5. Αναθέστε στο project το ολοκληρωμένο **FLEX 10K EPF10K70RC240-4**.
6. Αναθέστε στον πρώτο διακόπτη (pin 41) την είσοδο **RESETN**, στους δύο διακόπτες πίεσης (pin **28** και **29**) την είσοδο **MODE[1..0]** και στον ταλαντωτή (pin **91**) την είσοδο **CLK**.
7. Αναθέστε τα κατάλληλα pins στις εξόδους **COUNT** έτσι που τα **a, b, f** και **g** στο πρώτο SSD να καθορίζουν τον «δακτύλιο». Ανατρέξτε στο βήμα b.7 στην σελίδα 10 για την αρίθμηση των pins.

Επειδή τα SSDs είναι αρνητικής λογικής, ο άσπος του δακτυλίου θα παρουσιάζεται ως το μη αναμμένο κομμάτι τους.

8. Αποθηκεύστε, ελέγξετε και μεταφράστε τον σχεδιασμό.
9. Κατεβάστε τον σχεδιασμό σας στο FLEX 10K με την εφαρμογή **Programmer**
10. Ελέγξετε την ορθότητα λειτουργίας τους σχεδιασμού.

Κατεβάστε τον 1^ο διακόπτη για τον καθαρισμό του μετρητή και μετά ανεβάστε τον. Χρησιμοποιήστε τους διακόπτες πίεσης 1 και 2 για να ελέγξετε τις 4 στάσεις λειτουργίας. Αρχίστε με την παράλληλη φόρτωση (11) για την αρχικοποίηση του μετρητή. Μετά εφαρμόστε τα 00, 01 και 10 για τις υπόλοιπες λειτουργίες του.

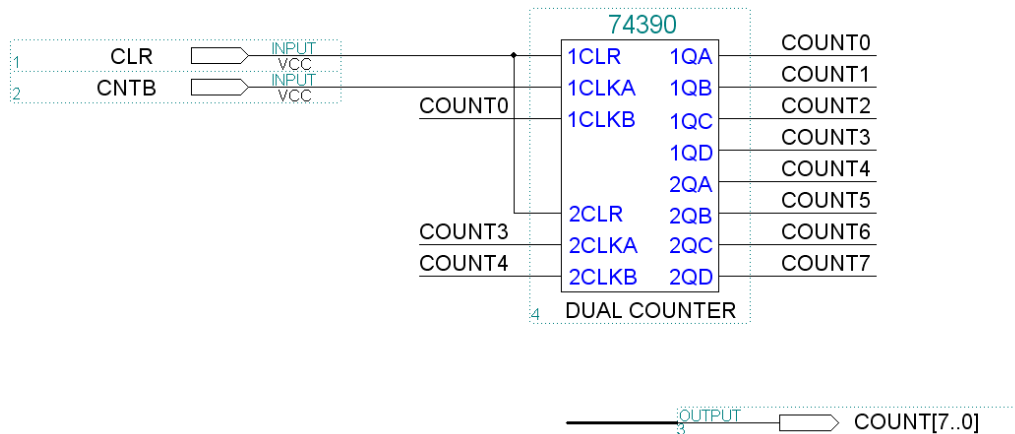
b) Μετρητής BCD Modulo 100

1. Ανοίξετε το σχηματικό lab5c.gdf.
2. Εισάγετε μια πύλη not για την αντιστροφή της εισόδου CLR.
 Η είσοδος CLR θα ελέγχεται από διακόπτη πίεσης έτσι που ο μετρητής να μηδενίζεται όταν είναι πατημένος.
3. Εισάγετε ένα σύμβολο clk_div και δύο σύμβολα dec_7seg στον σχεδιασμό.
 Βεβαιωθείτε ότι τα σύμβολα αυτά βρίσκονται στο κατάλογο που δουλεύετε. Αν όχι προσθέστε τα όπως στο βήμα a2.
4. Συνδέστε το σύμβολο clk_div έτσι που να μετατρέψετε την είσοδο CNTB σε 10 Hz. Συνδέστε τα σύμβολα dec_7seg στις εξόδους του μετρητή.
5. Εισάγετε τον κατάλληλο αριθμό θυρών εξόδου.
6. Αναθέστε στο project το ολοκληρωμένο FLEX 10K EPF10K70RC240-4.
7. Αναθέστε στα κατάλληλα pins τις εισόδους και τις εξόδους του σχεδιασμού σας. Η ασύγχρονη είσοδος CLR θα ελέγχεται από το διακόπτη πίεσης 1 (pin 28). Το περισσότερο σημαντικό ψηφίο του μετρητή θα παρουσιάζεται στο 1^ο SSD και το λιγότερο σημαντικό στο 2^ο. Η είσοδος CNTB θα ελέγχεται από τον ταλαντωτή (pin 91).
8. Κατεβάστε τον σχεδιασμό στο ολοκληρωμένο FLEX 10K .

Συνδέσεις του FLEX με τα SSDs

Display Segment	Pin for Digit 1	Pin for Digit 2
a	6	17
b	7	18
c	8	19
d	9	20
e	11	21
F	12	23
g	13	24

9. Πατήστε τον διακόπτη πίεσης 1 για να μηδενίσετε τον μετρητή BCD και μετά απελευθερώστε τον για να επιβεβαιώσετε την ορθή λειτουργία απαρίθμησης από το 00 έως το 99.
10. Καταγράψετε τα σχόλια σας στο σημειωματάριο του εργαστηρίου σας για την επαλήθευση των πειραματικών αποτελεσμάτων σας. Επίσης, σημειώστε οποιεσδήποτε δυσκολίες συναντήσατε και τον τρόπο που τις αντιμετωπίσατε.



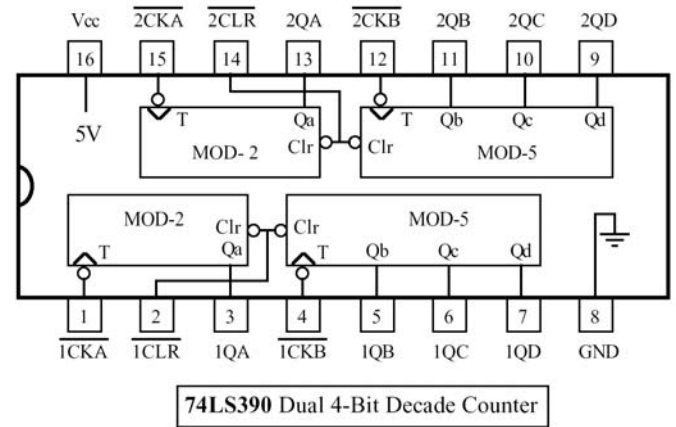
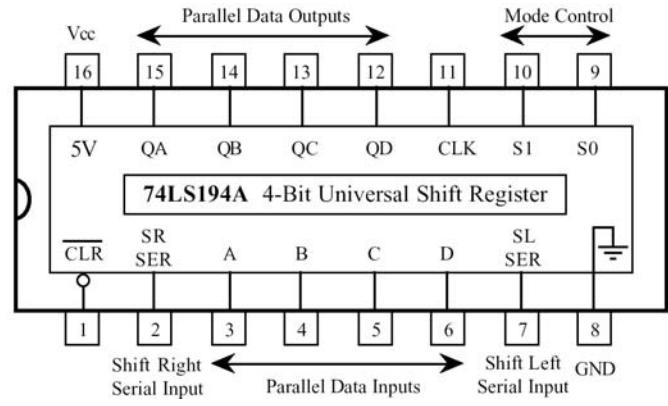
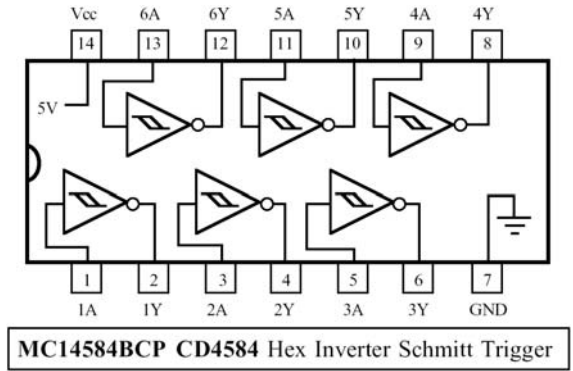
Μέρος B(b) (Δεύτερη Εβδομάδα)

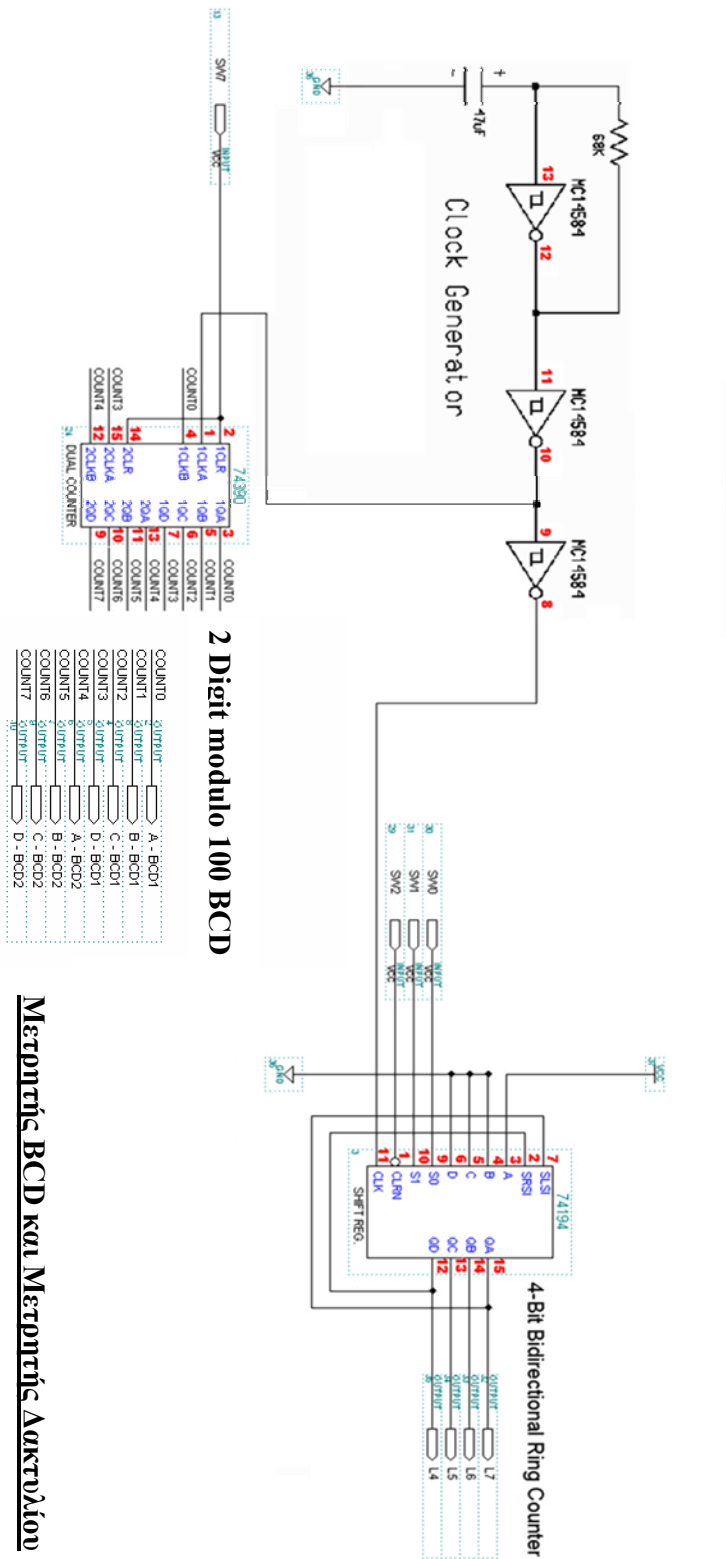
Στόχος αυτής της άσκησης υλικού είναι να εξακριβώσετε πειραματικά την λειτουργία του Σύγχρονου Αμφίδρομου Μετρητή Δακτυλίου, που προσομοιάσατε στο μέρος A.II και του μετρητή 2 ψηφίων BCD modulo 100 του μέρους A.III. Η έξοδος του μετρητή BCD θα παρουσιάζεται στα SSDs της πλακέτας ETS-7000.

Η πλακέτα ETS-7000 διαθέτει δύο SSDs και δύο μετατροπείς από BCD σε SSD. Οι είσοδοι του κάθε μετατροπέα είναι τα A, B, C και D, με το A να δηλώνει το λιγότερο σημαντικό ψηφίο. Κάθε μία από τους μετατροπείς BCD1 και BCD2 αντιστοιχεί στα SSDs d1 και d2 αντίστοιχα. Για να παρουσιάσετε τα αποτελέσματα στα SSDs συνδέστε τις εξόδους BCD που δίνει το κύκλωμα σας με τις αντίστοιχες A,B,C και D για τα BCD1 και BCD2.

Τα σχηματικά διαγράμματα για τα ολοκληρωμένα βρίσκονται στην σελίδα 12 ενώ ένα ολοκληρωμένο σχηματικό διάγραμμα που δείχνει και τα αριθμημένα pins για όλες τις συνδέσεις (εκτός τροφοδοσία και γείωση) δίνεται στην σελίδα 13. Επίσης στην σελίδα 13 δίνεται ένα σχηματικό το οποίο υλοποιεί μια γεννήτρια ρολογιού (Clock Generator) με την χρήση CMOS Schmitt Trigger αντιστροφών και παθητικών στοιχείων (πυκνωτής και αντίσταση). Η γεννήτρια μας δίνει ρολόι χαμηλής συχνότητας (περίπου 0.6 Hz) το οποίο θα μας επιτρέψει να παρατηρούμε τις εξόδους του κυκλώματος.

1. Εισάγετε το *74ls390* (δύο μετρητές BCD), στην αριστερή στήλη της πλακέτας ETS-7000. Αυτό αποτελεί τον μετρητή 2 ψηφίων BCD modulo 100. Ο διακόπτης SW7 θα χρησιμοποιηθεί ως είσοδος για το ασύγχρονο CLEAR του μετρητή. Συνδέστε τις εξόδους του όπως φαίνονται στο σχήμα της σελίδα 13.
2. Εισάγετε τον Οικουμενικό Καταχωρητή 4-bit *74LS194A* και το αντιστορφέα Schmitt Trigger *MC15484*, με αυτή την σειρά από πάνω προς τα κάτω στην τρίτη στήλη της πλακέτας ETS-7000. Αυτά τα κομμάτια θα αποτελέσουν τον Αμφίδρομο Μετρητή Δακτυλίου 4-bit και το ρολόι συχνότητας 3 Hz. Ο διακόπτης SW2 θα χρησιμοποιηθεί για το είσοδο του ασύγχρονου, αρνητικής λογικής, σήματος CLEAR του μετρητή, ενώ οι SW1 και SW0 θα ελέγχουν την κατάσταση λειτουργίας του. Οι έξοδοι του μετρητή θα παρουσιάζονται στους λαμπτήρες L7, L6, L5 και L4.
3. Συνδέστε τα κυκλώματα όπως φαίνεται στο σχηματικό της σελίδα 13. Συμπεριλάβετε την αντίσταση 68K και τον πυκνωτή 47μF με την πολικότητα όπως αναφέρεται στο σχήμα. Σημειώστε ότι όλα τα pins των ολοκληρωμένων φαίνονται στο διάγραμμα.
4. Βάλτε τον διακόπτη SW7 στην θέση 1 για να μηδενίσετε τον μετρητή 2 ψηφίων BCD και ακολούθως βάλτε τον στην θέση 0 για να μπορέσετε να επιβεβαιώσετε την σωστή αρίθμηση από το 00 έως και το 99.
5. Βάλτε τον διακόπτη SW2 στην θέση 0 για να μηδενίσετε τον Μετρητή Δακτυλίου και μετά βάλτε τον στην θέση 1 και χρησιμοποιώντας τους SW1 και SW0 ελέγξτε τις τέσσερις λειτουργίες του μετρητή. Αρχίστε με την παράλληλη φόρτωση (11) για να αρχικοποιήσετε τον μετρητή και μετά ελέγξτε τις καταστάσεις 00, 01 και 10 για τις λειτουργίες Hold (καμία αλλαγή), Δεξιά Ολίσθηση και Αριστερή Ολίσθηση, αντίστοιχα.
6. Γράψετε σχόλια στο σημειωματάριο του εργαστηρίου για τα πειραματικά αποτελέσματα. Ακόμα σημειώστε οποιεσδήποτε δυσκολίες συναντήσατε και τον τρόπο που τις αντιμετωπίσατε.





Όλοι οι αριθμοί των pins εκτός από αυτά της τροφοδοσίας και της γείωσης φαίνονται με κόκκινο. Ανατρέξετε στην σελίδα 10 για τους αριθμούς των pins για τα Gnd και το Vcc.

Μετρητής BCD και Μετρητής Δεκαδίου

Λίστα Στοιχείων και Οργάνων

A/A	Qty	Αρ. Στοιχ.	Περιγραφή
1	1	WK-1	Wire Kit
2	1	74LS194A	4-bit Universal Shift Register
3	1	74LS390	Dual 4-bit Decade Counter
4	1	ETS-7000	Digital – Analog Training System unit
5	1	CD4584	Hex Inverter Schmitt Trigger
6	1		68K Resistor
7	1		47μF Capacitor