

Πανεπιστήμιο Θεσσαλίας - Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών

ΗΥ430 - Εργαστήριο Ψηφιακών Κυκλωμάτων

Χειμερινό Εξάμηνο - Ακαδημαϊκό Έτος 2018-2019

Εργαστηριακή Εργασία 1η - Οδηγός Ένδειξης 7-τμημάτων

1/10/2018 έως 24/10/2018

Χ. Σωτηρίου

1 Στόχος της 1ης Εργασίας

Ο στόχος της πρώτης εργαστηριακής εργασίας είναι η υλοποίηση ενός οδηγού των τεσσάρων ενδείξεων 7-τμημάτων LED της πλακέτας Spartan 3, για να επιτευχθεί η περιστροφική παρουσίαση ενός μηνύματος μεγέθους 16 χαρακτήρων. Το προτεινόμενο (αλλά βαρετό) μήνυμα είναι οι 16 χαρακτήρες των ψηφίων του δεκαεξαδικού συστήματος, δηλ. 0123456789ABCDEF, αλλά μπορείτε, αν θέλετε, να υλοποιήσετε ένα εναλλακτικό μήνυμα της αρεσκείας σας, κάνοντας τις κατάλληλες τροποποιήσεις στις παρακάτω οδηγίες.

Το μήνυμα θα παρουσιαστεί μετατοπίζοντας τους χαρακτήρες, έναν προς έναν, κατάλληλα προς τα αριστερά, είτε:

- (α) με το πάτημα ενός κουμπιού, είτε
- (β) μετά απο ενα δεδομένο χρονικό διάστημα.

Μετά τον τελευταίο χαρακτήρα του μηνύματος, θα ακολουθεί ο πρώτος, έτσι το μήνυμα ουσιαστικά θα περιστρέφεται διαρκώς.

2 Οι Τέσσερις Ενδείξεις 7 Τμημάτων

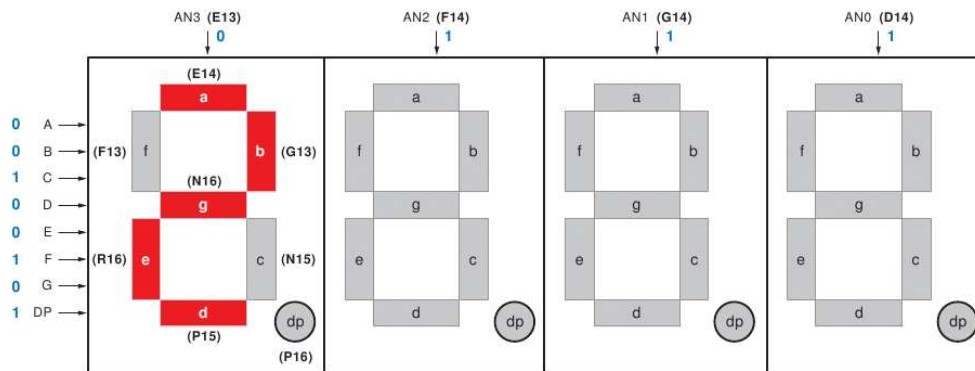
Στην σελίδα 15 του τεχνικού δελτίου της πλακέτας Spartan 3, παρουσιάζεται αναλυτικά η συνδεσμολογία των τεσσάρων ενδείξεων 7-τμημάτων στην πλακέτα. Τα τέσσερα ψηφία μοιράζονται τα 7 συν 1 (δεκαδικό σημείο) σήματα A, B, C, D, E, F, G και DP, όπως φαίνεται στο Σχήμα 1.

Για να ανάψει ενα συγκεκριμένο τμήμα LED ενός ψηφίου πρέπει:

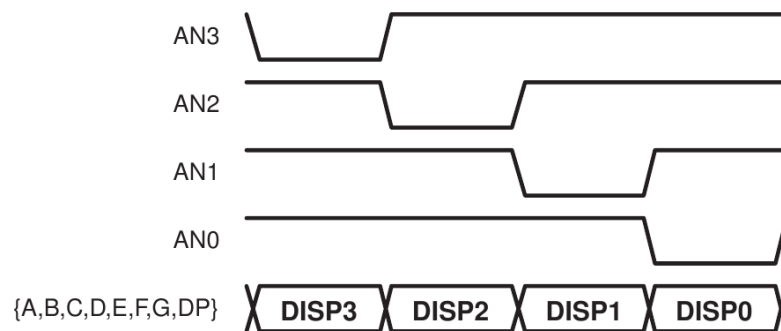
- το σχετικό σήμα ανόδου, AN0-3, που επιλέγει το ψηφίο, να οδηγείται στο μηδέν,
- το σχετικό σήμα τμήματος, A-G/DP, που επιλέγει το τμήμα, να οδηγείται στο μηδέν.

Στο Σχήμα 2 φαίνεται η κατάλληλη πολύπλεξη στον χρόνο που απαιτείται για να εμφανιστούν δεδομένα και στα τέσσερα ψηφία. Οι τέσσερις άνοδοι πρέπει να οδηγούνται εναλλάξ στο μηδέν, ενώ αλλάζουν κατάλληλα τα σήματα των 7 συν 1 ενδείξεων. Η διαδικασία πρέπει να επαναλαμβάνεται διαρκώς. Παρότι που τα LED ουσιαστικά αναβοσβήνουν, λόγω της γρήγορης εναλλαγής και της διατήρησης της φωτεινότητας τους, στο ανθρώπινο μάτι φαντάζουν σταθερά.

Όταν μια άνοδος, ANx, επιστρέφει στο ένα, οι τιμές των ενδείξεων δεν διατηρούνται, και καθώς το κύκλωμα των LED εκφορτίζεται, τα φωτισμένα τμήματα σβήνουν. Κατά την εναλλάξ διαδικασία σάρωσης, η άνοδος και τα σήματα των τμημάτων πρέπει να μένουν σταθερά για ενα σχετικά μεγάλο χρονικό διάστημα, ως προς το ρολόι της πλακέτας, λόγω της μεγάλης χωρητικότητας του κυκλώματος των LED. Έτσι, τα σήματα θα πρέπει να μένουν σταθερά για τουλάχιστον $\frac{1}{4}$ του μsec .



Σχήμα 1: Δομή Τεσσάρων Ψηφίων των 7-Ενδείξεων



Σχήμα 2: Χρονοδιάγραμμα Οδήγησης Ψηφίων

Ως χρόνος εναλλαγής των ανόδων προτείνονται τα 0.32μsec, τα οποία είναι πολλαπλάσια του ρολογιού των 50MHz της FPGA. Παρακάτω παρουσιάζονται τα μέρη υλοποίησης της εργασίας. Τα μέρη Β, Γ και Δ απαιτούν και δοκιμή προγραμματισμού της συσκευής.

3 Μέρος Α - Υλοποίηση Αποκωδικοποιητή 7-τμημάτων

Για την αποκωδικοποίηση των 7 συν 1 τμημάτων υλοποιήστε ένα συνδυαστικό αποκωδικοποιητή, ο οποίος θα έχει ως είσοδο το ψηφίο ή χαρακτήρα που θέλετε να εμφανίσετε, και θα παράγει ως έξοδο τις τιμές οδήγησης των 8 τμημάτων. Για την αναπαράσταση των 16 ψηφίων απαιτείται οι χρήση αριθμών 4-bit, ενώ τα 7 τμήματα απαιτούν 7 ξεχωριστά bit (το δεκαδικό ψηφίο, DP, μπορεί να μένει σβηστό, οδηγώντας το μόνιμα στο 1).

Έτσι, μια μορφή του αποκωδικοποιητή 7-τμημάτων είναι η εξής:

```
module LEDdecoder(char, LED);
input [3:0] char;
output [6:0] LED;

...

endmodule
```

Σχήμα 3: Μια Μορφή Υλοποίησης του Αποκωδικοποιητή

Συμπληρώστε την υλοποίηση του αποκωδικοποιητή και ελέγξτε λειτουργικά, μέσω προσομοίωσης, ότι οι τιμές που προκύπτουν στα ψηφία του LED είναι οι κατάλληλες, και έχουν και την σωστή πολικότητα (0 = αναμμένο και 1 = σβηστό). Για ευκολία στην επαλήθευση, αλλά και για την οδήγηση τους στην πλακέτα, διαχωρίστε τα ψηφία του αποτελέσματος LED στις ξεχωριστές εξόδους των τμημάτων A έως G.

Όταν θεωρήσετε ότι η υλοποίηση είναι ορθή, επιδείξτε τον κώδικα Verilog που γράψατε και τα αποτελέσματα της προσομοίωσης σε επιτηρητή του εργαστηρίου.

4 Μέρος Β - Οδήγηση Τεσσάρων Ψηφίων

Όπως εξηγήθηκε νωρίτερα, στην Παράγραφο 2, για την χρήση των τεσσάρων ψηφίων απαιτείται η εναλλάξ οδήγηση του κάθε ψηφίου με προτεινόμενη ελάχιστη καθυστέρηση φόρτισης τα 0.32μsec. Ο ευκολότερος τρόπος οδήγησης των ανόδων στην ταχύτητα των 0.32μsec ή 320ns είναι να πολλαπλασιαστεί η περίοδος του ρολογιού της FPGA (περιόδου 20ns) κατά δεκαέξι φορές, και κατόπιν η οδήγηση των σημάτων να είναι ορισμένη στην Verilog ανα κύκλο, βασισμένη στο νέο αργότερο ρολόι. Εναλλακτικά, θα έπρεπε να χρησιμοποιηθεί μετρητής ο οποίος, κάθε δεκαέξι λ.χ. κύκλους του ρολογιού των 20ns, θα σηματοδοτούσε την αλλαγή των σημάτων των ανόδων και των τμημάτων.

4.1 Δομή της μονάδας

Για τον πολλαπλασιασμό της περιόδου (ή ισοδύναμα διαίρεση της συχνότητας του ρολογιού), σας προτείνεται να χρησιμοποιήσετε μια δομική μονάδα DCM (Digital Clock Manager). Η δομική αυτή μονάδα θα πρέπει να βρίσκεται εσωτερικά στην κύρια μονάδα του οδηγού που θα υλοποιήσετε, μαζί με την εμφάνιση του LEDdecoder, που υλοποιήσατε στο πρώτο μέρος, όπως φαίνεται παρακάτω:

```
module FourDigitLEDdriver(reset, clk, an3, an2, an1, an0,
a, b, c, d, e, f, g, dp);
input clk, reset;
output an3, an2, an1, an0;
output a, b, c, d, e, f, g, dp;

...

    DCM #( ...
        ) DCM_inst (
        );

...

    LEDdecoder LEDdecoderINSTANCE (...);

...

endmodule
```

Σχήμα 4: Μια Μορφή Υλοποίησης του Οδηγού

Για να εμφανίσετε το πρότυπο της μονάδας DCM, το οποίο θα πρέπει και να παραμετροποιήσετε και να συνδέσετε κατάλληλα, ακολουθείστε τα εξής βήματα στο ISE Project Navigator.

1. Επιλέξτε το μενού Edit→ Language Templates

2. Επιλέξτε απο τα πρότυπα Verilog→ Device Primitive Instantiation → Clock Components → Digital control manager.
3. Αντιγράψτε το πρότυπο, ορίστε την παράμετρο διαίρεσης (CLKDV_DIVIDE) ως 16, οδηγήστε το CLKIN και χρησιμοποιείστε το CLKDV σαν το νέο ρολόι.

4.2 Οδήγηση των Ανόδων

Έχοντας διαιρέσει το ρολόι, θα πρέπει να οδηγήσετε τα σήματα των ανόδων όπως στο Σχήμα 2. Τα σήματα των ανόδων δεν θα πρέπει να επικαλύπτονται όσο βρίσκονται στο μηδέν, αλλιώς θα αλλοιωθούν οι ενδείξεις. Έτσι, απαιτείται και ένα περιθώριο ασφαλείας απο την επιστροφή ενός σήματος ανόδου στο λογικό ένα, μέχρι την πτώση του επόμενου.

Η μη-επικαλυπτώμενη οδήγηση των ανόδων μπορεί να επιτευχθεί με την χρήση ενός περιστροφικού μετρητή, όπου κάποιες τιμές του μετρητή θα ενεργοποιούν (στο μηδέν) εναλλάξ τις ανόδους, ενώ οι ενδιάμεσες θα τις επιστρέφουν στο λογικό ένα. Παραδείγματος χάριν, ένας 4-bit μετρητής θα μπορούσε να ενεργοποιεί τις ανόδους όπως φαίνεται στον παρακάτω πίνακα, αρχικοποιημένος στην τιμή 1111.

Τιμή μετρητή	AN3	AN2	AN1	ANO
1111	1	1	1	1
1110	0	1	1	1
1101	1	1	1	1
1100	1	1	1	1
1011	1	1	1	1
1010	1	0	1	1
1001	1	1	1	1
1000	1	1	1	1
0111	1	1	1	1
0110	1	1	0	1
0101	1	1	1	1
0100	1	1	1	1
0011	1	1	1	1
0010	1	1	1	0
0001	1	1	1	1
0000	1	1	1	1

Πρακτικά, ο μετρητής χρησιμοποιείται εμμέσως ως ΜΠΚ (Μηχανή Πεπερασμένων Καταστάσεων), όπου η μέτρηση του αντιστοιχεί πρακτικά σε μια κατάσταση.

Συμπληρώστε την υλοποίηση της οδήγησης των ανόδων, είτε βάση της προτεινόμενης υλοποίησης, είτε με κάποια εναλλακτική, οδηγώντας τις τιμές των ενδείξεων σταθερά, λ.χ. 0123. Για την οδήγηση των σημάτων των ενδείξεων χρησιμοποιείτε πάλι τιμές του μετρητή, έτσι ώστε να υπάρχει χρόνος προετοιμασίας (πρόθεσης - setup) μεταξύ των ανόδων και των δεδομένων των ενδείξεων. Δώστε περιθώριο αλλαγής τουλάχιστον ενός βήματος. Λ.χ. για την οδήγηση της AN3 στο μηδέν στο 1110, αλλάζτε τα δεδομένα στην μέτρηση 0000, για την οδήγηση της AN2 στο μηδέν στο 1010, αλλάζτε τα δεδομένα στο 1100, κ.ο.κ.. Ο λόγος που απαιτείται και αυτό το περιθώριο είναι η μεγάλη χωρητικότητα των ενδείξεων στην πλακέτα.

4.3 Αρχικοποίηση

Για την αρχικοποίηση του κυκλώματος χρησιμοποιείτε ένα ασύγχρονο σήμα αρχικοποίησης `reset` σε κάθε ακολουθιακό τμήμα `always`. Το σήμα αρχικοποίησης θα πρέπει να οδηγείται απο το κουμπί `BTN3` (User Reset) της συσκευής, το οποίο και αντιστοιχεί στην ακίδα `L14` της `FPGA`. Προερατικά, χρησιμοποιείτε κύκλωμα αποφυγής αναπηδήσεων (`anti-bounce`) για το σήμα του κουμπιού.

4.4 Περιορισμοί UCF (Universal Constraint Format)

Παρακάτω, σας παραθέτουμε το σχετικό UCF που πρέπει να χρησιμοποιήσετε, το οποίο περιέχει την αντιστοίχιση των σημάτων στις σχετικές ακίδες της συσκευής:

```
NET clk LOC = T9;
NET an3 LOC = E13;
NET an2 LOC = F14;
NET an1 LOC = G14;
NET an0 LOC = D14;

NET a LOC = E14;
NET b LOC = G13;
NET c LOC = N15;
NET d LOC = P15;
NET e LOC = R16;
NET f LOC = F13;
NET g LOC = N16;
NET dp LOC = P16;
NET reset LOC = L14;

NET "clk" TNM_NET = "clk";
TIMESPEC "TS_clk" = PERIOD "clk" 20 ns HIGH 50 %;
```

4.5 Λειτουργικός Έλεγχος

Όταν ολοκληρώσετε την υλοποίηση του Μέρους B, πραγματοποιείτε έλεγχο λειτουργίας, με προσομοίωση σε επίπεδο συμπεριφοράς, και κατόπιν σε επίπεδο πυλών, χρησιμοποιώντας ένα testbench που οδηγεί κατάλληλα το ρολόι. Όταν βεβαιωθείτε ότι η υλοποίηση του κυκλώματος είναι ορθή, παράξετε το κατάλληλο bitfile, προγραμματίστε την συσκευή και ελέγξτε ότι και στην πράξη, οι τέσσερις ενδείξεις οδηγούνται σωστά.

Όταν το κύκλωμα λειτουργεί όπως πρέπει, επιδείξτε τον κώδικα Verilog που γράψατε, τα αποτελέσματα της προσομοίωσης, και το κύκλωμα εν λειτουργία σε επιτηρητή του εργαστηρίου.

5 Μέρος Γ - Βηματική Περιστροφή του Μηνύματος με χρήση Κουμπιού

Στο τρίτο μέρος θα υλοποιήσετε την περιστροφή του μηνύματος στο πάτημα ενός κουμπιού. Για αυτήν, θα χρειαστεί καταρχήν να αποθηκευτεί σε δομή μνήμης το μήνυμα. Η μνήμη μπορεί να φτιαχτεί από καταχωρητές, και για το εν λόγω μήνυμα απαιτούνται 16 στοιχεία των 8-bit. Επιπλέον, τα τέσσερα ψηφία προς αποκωδικοποίηση, θα πρέπει τώρα να είναι μεταβλητά και όχι σταθερά, ανάλογα με την ενεργή διεύθυνση της μνήμης, η οποία θα περιστρέφεται στο μηδέν. Αυτό μπορεί να επιτευχθεί, χρησιμοποιώντας έναν μετρητή - δείκτη στην ενεργή διεύθυνση του μηνύματος, από τον οποίο θα προέρχονται (διαβάζοντας τα μνήμη) τα τέσσερα ψηφία. Κατόπιν, θα αποκωδικοποιούνται όπως και στο δεύτερο μέρος.

Η μνήμη μπορεί να οριστεί ως εξής:

```
reg [3:0] message [0:15];
...
```

Σχήμα 5: Μνήμη Μηνύματος από Καταχωρητές

Ο μετρητής - δείκτης θα αυξάνεται στο πάτημα ενός κουμπιού της πλακέτας. Χρησιμοποιήστε το κουμπί BTN2, το οποίο αντιστοιχεί στην ακίδα L13 της συσκευής, και υλοποιήστε κύκλωμα αποφυγής αναπηδήσεων (anti-bounce) για το σήμα του κουμπιού.

Όταν ολοκληρώσετε την υλοποίηση του Μέρους Γ, πραγματοποιείστε έλεγχο λειτουργίας, με προσομοίωση σε επίπεδο συμπεριφοράς, και κατόπιν σε επίπεδο πυλών, χρησιμοποιώντας ένα test-bench που οδηγεί κατάλληλα το ρολόι και προσομοιώνει και την συμπεριφορά του κουμπιού για ένα η δύο πατήματα. Όταν βεβαιωθείτε ότι η υλοποίηση του κυκλώματος είναι ορθή, παράξτε το κατάλληλο bitfile, προγραμματίστε την συσκευή και ελέγξτε ότι οι τέσσερις ενδείξεις οδηγούνται σωστά.

Όταν το κύκλωμα λειτουργεί όπως πρέπει, επιδείξτε τον κώδικα Verilog που γράψατε, τα αποτελέσματα της προσομοίωσης, και το κύκλωμα εν λειτουργία σε επιτηρητή του εργαστηρίου.

6 Μέρος Δ - Βηματική Περιστροφή του Μηνύματος με σταθερή Καθυστέρηση

Στο τέταρτο μέρος πρέπει να αντικαταστήσετε την λειτουργία του κουμπιού με μια σταθερή καθυστέρηση. Έτσι, κάθε 1 δευτερόλεπτο, ή λίγο περισσότερο, θα πρέπει να περιστρέφεται κατά ένα χαρακτήρα το μήνυμα. Η σταθερή καθυστέρηση μπορεί να υλοποιηθεί με έναν μετρητή κύκλων ρολογιού, ο μηδενισμός του οποίου θα επιδεικνύει το πέρας του χρόνου. Σας προτείνεται η υλοποίηση ενός μετρητή 22-bit, ο οποίος μετράει 1,3421 δευτερόλεπτα.

Όταν ολοκληρώσετε την υλοποίηση του Μέρους Δ, πραγματοποιείστε έλεγχο λειτουργίας, με προσομοίωση σε επίπεδο συμπεριφοράς, και κατόπιν σε επίπεδο πυλών χρησιμοποιώντας ένα test-bench που οδηγεί κατάλληλα το ρολόι. Επειδή 1,3421 δευτερόλεπτα είναι πολύ μεγάλος χρόνος προσομοίωσης, δοκιμάστε το κύκλωμα με μικρότερο μετρητή (λιγότερων ψηφίων), και κατόπιν αλλάξτε το πλάτος του. Όταν βεβαιωθείτε ότι η υλοποίηση του κυκλώματος είναι ορθή, παράξτε το κατάλληλο bitfile, προγραμματίστε την συσκευή και ελέγξτε ότι οι τέσσερις ενδείξεις οδηγούνται σωστά.

Όταν το κύκλωμα λειτουργεί όπως πρέπει, επιδείξτε τον κώδικα Verilog που γράψατε, τα αποτελέσματα της προσομοίωσης, και το κύκλωμα εν λειτουργία σε επιτηρητή του εργαστηρίου.

7 Προθεσμία Παράδοσης, Υποβολή της Εργασίας και Αναφορά

Η προθεσμία παράδοσης της 1ης εργασίας είναι η **24/10/2018**.

Μέχρι την προθεσμία της εργασίας θα πρέπει:

- να έχετε επιδείξει όλα τα επιμέρους μέρη της εργασίας στους επιτηρητές,
- να έχετε υποβάλλετε τον κώδικα σας,
- να έχετε υποβάλλει μια εργαστηριακή αναφορά, βάση των σημειώσεων από το βιβλίο του εργαστηρίου.

8 Ερωτήσεις και Απορίες

Για οποιεσδήποτε ερωτήσεις και απορίες εκμεταλλευτείτε τον χρόνο και τον χώρο του εργαστηρίου και ρωτήστε τους επιτηρητές.