1η Εργαστηριακή αναφορά

Εργαστήριο Ψηφιακών Κυκλωμάτων

Καραμέλλιος Σωτήριος 2237

ΠΕΡΙΛΗΨΗ

Η πρώτη εργαστηριακή εργασία ολοκληρώθηκε χωρίς κάποιο πρόβλημα. Όλοι οι προγραμματισμοί της πλακέτας έγιναν μία φορά για το κάθε μέρος και δεν χρειάστηκαν αλλαγές στον αρχικό κώδικα. Παρακάτω αναφέρομαι αναλυτικά στα modules που χρησιμοποίησα καθώς και σε κάποια ακόμα αξιοσημείωτα κομμάτια του κώδικά μου.

ΕΙΣΑΓΩΓΗ

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 μήνυμα που παρουσίζεται είναι οι 16 χαρακτήρες του δεκαεξαδικού συστήματος δηλαδή 0,1,2,3,4,5,6,7,8,9,A,b,C,d,E,F. Το μήνυμα παρουσιάζεται μετατοπίζοντας τους χαρακτήρες έναν προς έναν προς τα αριστερά είτε με το πάτημα ενός κουμπιού(μέρος γ) είτε μετά από ένα δεδομένο χρονικό διάστημα. Τα κοινά modules που χρησιμοποιήθηκαν και στα δύο κομμάτια της εργασίας ήταν μια μονάδα DCM για την αύξηση της περιόδου του ρολογιού από τα 20ns στα 20ns στα 320ns, μια μονάδα decoder για την αποκωδικοποιηση του μηνύματός μας από νούμερα στο δυαδικό σε ποια bits της οθόνης πρέπει να ανάψουν(να γίνουν 0) και ποια όχι(να μείνουν 1), μια μονάδα displayselector η οποία επιλέγη κατάλληλα ποιος χαρακτήρας της οθόνης πρέπει να ανάψει(από τους 4) , ποια στιγμή και φροντίζει κατάλληλα για την προφόρτοση των δεδομένων ώστε να είναι έτοιμα όταν θα ανάψει ο χαρακτήρας και ένας resetmanager, ο οποίος είναι υπεύθυνος να εξασφαλίζει ότι το σήμα ρισέτ θα μεταδοθεί στο σύστημά μας για τουλάχιστον 3 κύκλους(που χρειάζεται ο DCM), σε συνδιασμό με έναν συγχρονιστή για το κουμπί του ρισετ, για να μην υπάρξουν προβλήματα μεταστάθειας. Ακόμα υπάρχει ένα module το messageselector το οποίο είναι υπεύθυνο για την επιλογή των τεσσάρων χαρακτήρων που θα προβάλονται κάθε φορά, η υλοποίησή του έχει όμως κάποιες μικρές διαφορές. Τέλος το σύστημα με το κουμπί περιλαμβάνει επιπλέον έναν antibouncer το οποίο είναι ένα κύκλωμα αποφυγής αναπηδήσεων για το σήμα που μας έρχεται από το κουμπι.

ΜΕΡΟΣ Α

Για το μέρος Α αυτό που υλοποιήθηκε ήταν ένας αποκωδικοποιητής 7 τμημάτων (το 8ο τμημα,το dp, δεν μας χρειάζεται σε αυτήν την εργασία , οπότε είναι καρφωμένο στο 1), ο οποίος λαμβάνει σαν είσοδό του έναν χαρακτήρα του 16δικού συστήματος αρίθμησης και δίνει σαν έξοδο το 7bit σήμα που χρειάζεται για να αναπαρασταθεί ο χαρακτήρας στην οθόνη της πλακέτας μας. Ουσιαστικά είναι μια απλή εντολή case μέσα σε ένα always block. Η κωδικοποίηση που χρησιμοποίησα είναι η εξής:

case (char)

4'b0000 : LED = 7'b0000001;

4'b0001 : LED = 7'b1001111;

4'b0010 : LED = 7'b0010010;

4'b0011 : LED = 7'b0000110;

4'b0100 : LED = 7'b1001100;

4'b0101 : LED = 7'b0100100;

4'b0110 : LED = 7'b0100000;

4'b0111 : LED = 7'b0001111;

4'b1000 : LED = 7'b0000000;

4'b1001 : LED = 7'b0001100;

4'b1010 : LED = 7'b0001000;

4'b1011 : LED = 7'b1100000;

4'b1100 : LED = 7'b0110001;

4'b1101 : LED = 7'b1000010;

4'b1110 : LED = 7'b0110000;

4'b1111 : LED = 7'b0111000;

Endcase

Η επαλήθευση έγινε με την χρήση ενός testbench το οποίο έδινε σαν είσοδο στην μονάδα μας όλους τους πιθανούς χαρακτήρες και έλεγχε την έξοδο για διαφορές.

ΜΕΡΟΣ Β

Σε αυτό το μέρος χρησιμοποίησα εκτός απο τον αποκωδικοποιητή του μέρους Α, ένα module DCM για τον πολλαπλασιασμο του ρολογιού επί 16 φορές(από 20 σε 320 ns), χρόνο που χρειάζεται να μείνει ανοιχτός ο χαρακτήρας της μιας οθόνης για να φορτιστεί αρκετά ώστε να μην σβήσει μέχρι να έρθει ξανά η σειρά του να ξανανοίξει. Ακόμα χρησιμοποιούμε ένα module resetmanager το οποίο όταν μας έρχετε το σήμα reset αρχικοποιεί έναν μετρητή στο 0 και κρατάει το σήμα reset ανεβασμένο για 3 κύκλους ρολογιού(των 20 ns) ακόμα ώστε να εξασφαλίσει σίγουρα στο DCM τους 3 κύκλους ρισετ που χρειάζεται, σε συνδιασμό με έναν συγχρονιστή για το κουμπί του ρισετ, για να μην υπάρξουν προβλήματα μεταστάθειας.Επίσης, υπάρχει ένας συγχρονιστής για το σήμα reset για να αποφεύγεται η όποια πιθανότητα μεταστάθειας υπάρχει. Τελος, υπάρχει και ένα module screenselector το οποίο αποφασίζει σύμφωνα με το παρακάτω διάγραμμα ποιος χαρακτήρας πρέπει να ανοίξει, ποια στιγμή πρέπει να ανοίξει και πότε πρέπει να φωρτοθούν τα δεδομένα για να προλάβουν να αποκωδικοποιηθούν και να είναι έτοιμα όταν ανοίξει η χαρακτήρας.

Το module screenselector δουλεύει οπως δείχνει η ακόλουθη εικόνα.

Η επαλήθευση αυτού του κυκλώματος έγινε απλά με την χρήση ενός ρολογιού και ενός σήματος reset και με τον έλεγψο κηματομορφών. Το κύκλωμα δούλεψε και στην πλακέτα.

If counter == 0000

Preload an3char

Default;

If counter == 1110

An3=0;

If counter == 1101

An3=1;

If counter == 1100

Preload an2char

If counter == 1010

An2=0;

If counter == 1001

An2=1;

If counter == 1000

Preload an1char

If counter == 0110

An1=0;

If counter == 0101

An1=1;

If counter == 0100

Preload an0char

If counter == 0010

An0=0;

If counter == 0001

An0=1;

ΜΕΡΟΣ Γ

Στο μέρος αυτό χρησιμοποιήθηκε εκτός από το κύκλωμα του μέρους Β ένα module antibouncer το οποίο για να αποφύγει τις αναπιδήσεις μετράει πόσους κύκλους είναι πατημένο το κουμπί και αν το κουμπί παραμείνει πατημένο για αρκετό χρονικό διάστημα τότε μόνο περνάει την νέα τιμή του κουμπιού στο υπόλοιπο σύστημα. Αυτό γίνεται για να αποφεύγεται η πολλαπλή αλλαγή των χαρακτήρων της οθόνης με ένα πάτημα του κουμπιού. Το μέγεθος του counter σύμφωνα με τα χαρακτηριστικα της πλακετας πρεπει να ειναι 250.000 κυκλοι το οποίο είναι λιγο λιγότερο από 2^18, άρα 18bit counter.

Επίσης υλοποιήται και ένα module messageselector το οποίο αρχικοποιείται με το reset στο να δείχνει τους χαρακτήρες 0123 στην οθόνη. Έπειτα με το πάτημα του κουμπιού οι χαρακτήρες ολυσθαίνουν προς τα αριστερά, κατα μία θέση την φορά που σημαόνει στο πρώτο πάτημα η οθόνη θα δείχνει 1234 και ούτο καθεξής. Αφού το μήνυμα γίνει CDEF τα ψηφία θα ακολουθήσουν από την αρχή με το πάτημα του κουμπιού, δηλαδή στο επόμενο πάτημα το μήνυμα θα γίνει DEF0.

EF01

DEF0

CDEF

BCDE

ABCD

9ABC

89ΑΒ

789Α

6789

5678

4567

3456

1234

2345

0123

F012

Η επαλήθευση του μέρους Γ έγινε με ενα απλό testbench το οποίο αρχικοποιεί το σύστημά μας με reset και μετά πατάει το κουμπί αρκετές φορές ώστε να γίνει μια πλήρης παρουσίαση του μηνύματός μας και με αρκετά μικρότερη τιμή στον μετρητή του antibouncer.

ΜΕΡΟΣ Δ

Στο μέρος αυτό χρησιμοποιήθηκε εκτός από το κύκλωμα του μέρους Β ένα module screenselector με ελάχιστα διαφορετική υλοποίηση. Ενώ ο τρόπος εναλλγής του μηνύματος είναι ο ίδιος, αυτή την φορά η εναλλαγή αντί να γίνεται με το πάτημα ενός κουμπιού γίνεται με το πέρας της ώρας. Πιο συγκεκριμένα αυτό γίνεται με έναν 22bit μετριτή ο οποίος μετράει περίπου 1,3421 δευτερόλεπτα και μετά αλλάζει το μήνυμα κατά ένα αριστερά.

Για την επαλήθευση αυτόυ του κυκλώματος χρησιμοποιήσαμε ένα σήμα reset και μια μικρότερη τιμή στον μετρητή, 5bit, καθώς 1,3421 δευτερόλεπτα σε χρόνο προσομοίωσης είναι πάρα πολύ μεγάλο χρονικό διάστημα.

0123