



Εργαστήριο Μικροϋπολογιστών

Όγδοη εργαστηριακή άσκηση

Ομάδα Δ07

Παπαλεξανδράκης Εμμανουήλ (Α.Μ.: 03114203)

Παπασκαρλάτος Αλέξανδρος (Α.Μ.: 03111097)

Ημερομηνία Υποβολής Αναφοράς: 13 Ιανουαρίου 2019

Θα παρουσιάσουμε μια σύντομη και ουσιαστική ανάλυση των προγραμμάτων που κατασκευάσαμε.

Τα προγράμματα αυτά καθ' αυτά είναι σε ξεχωριστά αρχεία.

Σημειώνουμε δε πως οι ρουτίνες που δίνονται στο pdf της εκφώνησης θεωρούνται γνωστές και δε θα τις αναλύσουμε περαιτέρω.

Σε αυτή τη σειρά ασκήσεων ασχολούμαστε με το αναπτυξιακό σύστημα ZYNQ, και συγκεκριμένα στο SoC FPGA "Zybo Development Board".

Ο κώδικας είναι σε C.

Επειδή πρόκειται για μια πρώτη επαφή με αυτό το σύστημα, οι ασκήσεις είναι σχετικά απλές (θεωρώντας δεδομένες τις ρουτίνες/βιβλιοθήκες που δίνονται στα βοηθητικά αρχεία για τις διάφορες αρχικοποιήσεις).

Μάλιστα, έχουμε υλοποιήσει τις ίδιες ουσιαστικά ασκήσεις και στον AVR σε προηγούμενες σειρές.

Άσκηση 8.1

Κατασκευάζουμε το πρόγραμμα απεικόνισης led που ζητείται.

Θέλουμε να απεικονίσουμε ένα led το οποίο εκκινώντας από το LSB να κινείται παλινδρομικά (και αενάως) ανάμεσα στο LSB και το MSB.

Όταν το button 0 είναι πατημένο, θέλουμε το πρόγραμμα να παύει.

Θυμίζουμε πως ο zybo έχει 4 led.

Καταρχάς, κάνουμε τις κατάλληλες αρχικοποιήσεις για τον έλεγχο των λειτουργιών του zybo, όπως περιγράφονται στον κώδικα της εκφώνησης.

Επιπλέον, αρχικοποιούμε τη μεταβλητή “leds_value” στην τιμή 0x01 και μια μεταβλητή “flag” στο 0x00.

Στη συνέχεια, έχουμε ένα αέναο loop.

Στην αρχή του loop, διαβάζουμε την τιμή των κουμπιών και ανν το lsb είναι 1 (δηλαδή το BTN0 είναι πατημένο), περιμένουμε, διαβάζοντας συνεχώς την τιμή των κουμπιών.

Εν συνεχεία (εφόσον το BTN0 δεν είναι πατημένο), προβάλλουμε την τιμή της leds_value στα led.

Έπειτα, ελέγχουμε το flag.

Αν το flag ισούται με 0x00, κάνουμε ένα shift προς τα αριστερά, αν είναι 0xFF, κάνουμε shift δεξιά.

Για να κάνουμε το shift, αρκεί να ολισθήσουμε τη δυαδική τιμή της leds_value κατά μία θέση προς την κατάλληλη κατεύθυνση.

Προτού τελειώσει το iteration, ελέγχουμε αν το led έχει φτάσει σε ένα από τα άκρα, στην οποία περίπτωση, θέλουμε να αντιστρέψουμε τη φορά της κίνησης.

Λοιπόν, ανν η leds_value έχει τιμή 0x01 ή 0x08, τότε συμπληρώνουμε το flag (οπότε αν είναι 0x00, το κάνουμε 0xFF και αντιστρόφως).

Τελευταίο βήμα του iteration είναι να καλέσουμε μια χρονοκαθυστέρηση προκειμένου να είναι εμφανής η εναλλαγή των led.

Άσκηση 8.2

Κατασκευάζουμε το πρόγραμμα προσομοίωσης λογικών πυλών που ζητείται.

Οι συναρτήσεις boole που δίνονται είναι $F0 = (AB + BC + CD + DA)'$,

$F1 = ABCD + D'A'$,

$F2 = F0 + F1$

Τα bit εισόδου A,B,C,D δίνονται από τους διακόπτες του zybo και τα bit εξόδου προβάλλονται στα led.

Δημιουργούμε ένα αέναο loop.

Στην αρχή του loop, διαβάζουμε την τιμή των διακοπών.

Χρησιμοποιώντας μάσκα του ενός bit και κατάλληλα shift στην τιμή των διακοπών, “τοποθετούμε” τα αντίστοιχα bit εισόδου στα lsb τεσσάρων μεταβλητών a, b, c, d.

Στη συνέχεια, χρησιμοποιούμε τις λογικές πράξεις που επιτρέπει εύκολα η C (not, and, or) και τοποθετούμε τα αντίστοιχα bit εξόδου στα lsb τριών μεταβλητών f0, f1, f2 σύμφωνα με τις τιμές των a, b, c, d.

Τέλος, αξιοποιώντας κατάλληλα shift (και τη λογική πράξη or) τοποθετούμε τα bit των f0, f1, f2 στα τρία lsb της μεταβλητής leds_value και προβάλλουμε στα led.

Άσκηση 8.3

Κατασκευάζουμε το πρόγραμμα απεικόνισης led που ζητείται.

Θέλουμε, έχοντας αρχικά αναμμένο μόνο το led 0, να επιτελείται μια λειτουργία όταν πατήσουμε ένα κουμπί. Θυμίζουμε πως ο zybo έχει 4 led.

α) BTN0 ολίσθηση-περιστροφή μία θέση αριστερά (κυκλικά)

β) BTN1 ολίσθηση-περιστροφή μία θέση δεξιά (κυκλικά)

γ) BTN2 μετακίνηση του αναμμένου led στην θέση (MSB-LD3)

δ) BTN3 μετακίνηση του αναμμένου led στην αρχική του θέση (LSB-LD0)

Όλες οι αλλαγές να γίνονται αφήνοντας (επανερχόμενα) τα push buttons BTNx.

Καταρχάς, κάνουμε τις κατάλληλες αρχικοποιήσεις για τον έλεγχο των λειτουργιών του zybo, όπως περιγράφονται στον κώδικα της εκφώνησης.

Επιπλέον, αρχικοποιούμε τη μεταβλητή "leds_value" στην τιμή 0x01 και μια μεταβλητή "state" στην τιμή 0x00.

Έχουμε ένα αέναο loop.

Αρχικά, προβάλλουμε την τιμή της leds_value στα led.

Διαβάζουμε τα κουμπιά.

Έπειτα, έχουμε τη μεταβλητή state η οποία θα έχει κάποιο bit 1, ανν το αντίστοιχο κουμπί έχει πατηθεί (αλλά δε μηδενίζεται αυτόματα όταν το αντίστοιχο κουμπί αφεθεί).

Για να το πετύχουμε αυτό, σε αυτό το σημείο, κάνουμε bitwise or τη state με την buttons_value (και αποθηκεύουμε το αποτέλεσμα στη state).

Λοιπόν, έχουμε ένα for loop, κατά το οποίο εξετάζουμε ένα-ένα αν κάποιο κουμπί έχει πατηθεί και, εν συνεχεία, αφεθεί (και σταματάμε στο πρώτο που ικανοποιεί αυτή τη συνθήκη).

Για να το πετύχουμε αυτό κάνουμε bitwise and ανάμεσα στη state και το συμπλήρωμα της buttons_value (και μια ολισθαίνουσα μάσκα του ενός bit).

Έτσι, η συνθήκη θα ικανοποιείται ανν το αντίστοιχο bit της state είναι 1 (άρα το κουμπί έχει πατηθεί) και το αντίστοιχο bit της buttons_value είναι 0 (άρα το κουμπί δεν πατιέται αυτή τη στιγμή).

Τέλος, αν δε βρήκαμε κανένα, το iteration τερματίζει.

Διαφορετικά, αξιοποιώντας ένα conditional τύπου switch-case, για το κουμπί που βρήκαμε στο προηγούμενο βήμα, επιτελούμε την αντίστοιχη λειτουργία, αλλάζοντας την τιμή της leds_value κατάλληλα.

Επίσης, εφόσον "ικανοποιήσαμε" αυτή τη λειτουργία, ξανααρχικοποιούμε το αντίστοιχο bit (και μόνο αυτό) της state στο 0.

Επαναλαμβάνουμε από την αρχή του loop.