Χριστοφορίδης Ιωάννης Ραφαήλ 368

Ενσωματωμένα συστήματα

Εργασία εξαμήνου

OLKI

Περιγραφή

Ο OLKI είναι ένας απλός επεξεργαστής που έχει τέσσερις 8-bitους καταχωρητές. Εκτελεί απλές λειτουργίες και πράξεις μεταξύ των αυτών των καταχωρητών. Επίσης έχει διάφορους ελέγχους και εμφανίζει σφάλμα αν κάτι πάει λάθος. Είναι υλοποιημένος σε πλακέτα FPGA Spartan-3A starter kit.

Είσοδοι / έξοδοι

Ο OLKI έχει είσοδο μόνο το πληκτρολόγιο(ps2) καθώς και το ρολόι της πλακέτας. Ως έξοδο χρησιμοποιεί τα 8 leds της πλακέτας.

Εντολές / Σύμβολα

Οι εντολές έχουν όλες 3 μέλη με εξαίρεση την show που έχει μόνο 2. Κάθε εντολή πρέπει να επιβεβαιώνεται με το enter προτού εκτελεστεί. Αποτελούνται μόνο από ένα γράμμα. Σε περίπτωση σφάλματος ανάβουν όλα τα λαμπάκια.

Τα σύμβολα που χρησιμοποιούνται είναι:

O,L,K,I – Οι 4 καταχωρητές του επεξεργαστή.

0123456789 Τα νούμερα από το 0 μέχρι το 9

-,=,backspace,[,], Τα νούμερα από το A μέχρι το F αντίστοιχα.

Esc – Ακυρώνει κάθε εντολή και βγάζει το σφάλμα.

Οι εντολές είναι οι εξής:

T(take)number1number2 – αναθέτει τιμή στον καταχωρητή «Ο». Παράδειγμα: Τ4] > Ο καταχωρητής «Ο» θα έχει την δυαδική τιμή 01001110

M(move)storage1storage2/number – μετακινεί την τιμή από το δεύτερο στο πρώτο. Το δεύτερο σύμβολο μπορεί και να μην είναι καταχωρητής. Παραδείγματα: ΜΚL > ο καταχωρητής «Κ» παίρνει την τιμή του καταχωρητή «L». ΜΙ4 > Ο καταχωρητής «Ι» παίρνει την τιμή 4.

A(add)storage1storage2/number – προσθέτει 2 στοιχεία και αποθηκεύει το αποτέλεσμα στο πρώτο σύμβολο. Το δεύτερο σύμβολο μπορεί και να μην είναι καταχωρητής. Παραδείγματα: ΑΙΟ > Ι=Ι+Ο. ΑΚ4 > Κ=Κ+4.

S(sub)storage1storage2/number – Αφαιρεί 2 στοιχεία και αποθηκεύει το αποτέλεσμα στο πρώτο σύμβολο. Το δεύτερο σύμβολο μπορεί και να μην είναι καταχωρητής. Παραδείγματα: SIK > I=I-K. SL- > K=K-10.

X(multiply)storage1storage2/number – πολλαπλασιάζει 2 στοιχεία και αποθηκεύει το αποτέλεσμα στο πρώτο σύμβολο. Το δεύτερο σύμβολο μπορεί και να μην είναι καταχωρητής. Παραδείγματα: ΧΟL > Ο=Ο*L. ΧΚ2 > Κ=Κ*2.

H(show)storage – Δέχεται μόνο ένα όρισμα, εμφανίζει τον καταχωρητή που επιλέγουμε. Παράδειγμα: HL > εμφανίζει τον καταχωρητή L.

Παράδειγμα λειτουργίας

- Τ1= : Ο καταχωρητής Ο θα πάρει την τιμή 00011011
- ML1 : Ο καταχωρητής L θα πάρει την τιμή 00000001
- ALO : Ο καταχωρητής L θα πάρει την τιμή L+O, δηλαδή 00011100
- XL4 : Ο καταχωρητής L θα πάρει την τιμή L*4, δηλαδή 01110000
- HL : Θα εμφανιστεί η τιμή του καταχωρητή L

Επεξήγηση κώδικα

Τα στοιχεία που χρειάζονται είναι τα εξής:

Main.vhd

ps2_keyboard.vhd

ps2_keyboard_to_ascii.vhd

debounce.vhd

mpla.ucf

Το main.vhd είναι ουσιαστικά ο κύριος κώδικας. Χρησιμοποιεί τα παραπάνω σήματα για τις λειτουργίες του:

- output - Για διάφορους λόγους όπως διαχείριση εξόδου, το output έχει μπει ως σήμα, και βεβαίως είναι η έξοδος και συνδέεται με τα leds της πλακέτας.
- nextdude - είναι το σήμα που στέλνεται όταν έχει ολοκληρωθεί μία εντολή ώστε να προχωρήσουμε στην επόμενη
- yesdude, yesmydude, misticflag, moveit, put_dispenser_here, dispenspy - Χρησιμεύει στην μεταχείριση των flip flop του επεξεργαστή. Το ένα κλείνει και ανοίγει το άλλο.
- ps2_code_new : Είναι 1 όταν πατιέται ένα πλήκτρο.
- ps2_code: Ο 7-bit κώδικας του πλήκτρου.
- Phase : Χρησιμεύει στην καταχώρηση της εντολής στο σύστημα.
- uno,dos,tres,returnme : Η πρώτη, δεύτερη, τρίτη παράμετρος και το enter αντίστοιχα. Το returnme χρησιμεύει και ως πρόχειρο σε μερικά σημεία.

- katax1, katax2, katax3, katax4, : Οι καταχωρητές OLKI αντίστοιχα.
- mistic - Χρησιμοποιείται ως πρόχειρο στην μεταφορά δεδομένων σε καταχωρητή.

Ο κώδικας αποτελείται σε μεγάλο μέρος με flip flops. Κατά την διάρκεια εισαγωγής έχουμε αλλαγή του signal "phase" και όταν πατηθεί το enter ενεργοποιείται το flip flop "yesdude". Σε αυτήν την φάση, το signal returnme παίρνει την τιμή της παραμέτρου 3. Στο τέλος, δίνει την σειρά του στο flip flop "yesmydude". Το συγκεκριμένο flip flop ελέγχει την εντολή που δώσαμε στην παράμετρο 1 και προχωρεί προς την υλοποίησή της, ειδικά για την μετακίνηση μεταβλητής ενεργοποιεί το flip flop "moveit" όπου και μεταφέρεται η τιμή του ενός καταχωρητή σε έναν άλλο. Στην εντολή take, η παράμετρος 2 επεξεργάζεται κατάλληλα και ενεργοποιείται το flip flop "put_dispenser_here". Εκεί σχηματίζεται το καλούπι της τελικής τιμής που θα πάρει ο καταχωρητής Ο μεταφέροντας τα 4 πρώτα bit στην θέση τους. Στην συνέχεια έχουμε το flip flop "dispenspy" όπου γίνεται η τελική ανάθεση του καταχωρητή Ο.

Ανάλυση απαιτήσεων

Γενικώς, οι περισσότερες εντολές τελειώνουν μέσα σε περίπου 2 κύκλους του ρολογιού. Η μόνη λειτουργία που θέλει το διπλάσιο αριθμό κύκλων είναι η εντολή take που χρειάζεται να ενεργοποιήσει αρκετά flip flop για την υλοποίησή της. Παρ' όλα αυτά η καθυστέρηση δεν είναι αισθητή στον χρήστη καθώς το ρολόι είναι 50MHz.

Σε θέμα μνήμης ο παρών επεξεργαστής είναι αρκετά "σπάταλος". Χρησιμοποιούνται πολλές θέσεις μνήμης για διάφορες δουλειές και όχι μόνο οι 4 καταχωρητές. Βέβαια, ο κώδικας έχει σχεδιαστεί έτσι ώστε να είναι δυνατή η αναβάθμιση του επεξεργαστή ώστε να εκτελεί περισσότερες λειτουργίες χωρίς επιπλέον κόστος.