ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΡΓΑΣΤΗΡΙΟ

SPACE DATA SYSTEMS

13^{Ο} Εργαστηριακό Μάθημα
6^{ο} Εργαστήριο

Βασιλόπουλος Διονύσης
ΕΔΙΠ Τμήματος Πληροφορικής & Τηλεπικοινωνιών - ΕΚΠΑ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ
2024-2025

Ανιχνευτής Ακολουθίας Ψηφίων (Sequence Detector)

Στην άσκηση αυτή, θα υλοποιήσετε έναν ανιχνευτή ακολουθίας ψηφίων (sequence detector) που εντοπίζει την ακολουθία διαδοχικών ψηφίων "**101"** που μεταδίδονται στην είσοδό του (*data_in*) και παράγει στην έξοδό του (*detect_out*) την τιμή 1 κάθε φορά που λαμβάνονται διαδοχικά τα τρία αυτά ψηφία.

Το κύκλωμα του sequence detector χρησιμοποιεί το ρολόι των 100MHz που παράγεται από τον κρύσταλλο της κάρτας (ακροδέκτης Y9) και μια είσοδο *enable* που δίνεται εξωτερικά από τον χρήση με το user push button (BTNC). Ουσιαστικά φτιάχνεται ένα FSM που ανιχνεύει την ακολουθία "**101".**

Οι είσοδοι *reset*, *enable* και *data*_in του sequence detector συνδέονται στα DIP switches και στα push-buttons της κάρτας ZedBoard σύμφωνα με τον παρακάτω πίνακα.

Είσοδοι	**Περιγραφή**	**DIP Switch**
reset	Ασύγχρονο reset	BTNU
enable	Clock enable	BTNC
data_in	Είσοδος ψηφίων	SW0

Πίνακας 2. Είσοδοι του Sequence detector

Η έξοδος *detect_out* του sequence detector συνδέεται στα LEDs της κάρτας ZedBoard σύμφωνα με τον παρακάτω πίνακα.

Έξοδοι	**Περιγραφή**	**LED**
detect_out	Έξοδος ανίχνευσης ακολουθίας "**101**"	LED0

Πίνακας 3. Έξοδος του Sequence detector

Για την αποφυγή των προβλημάτων αποκλειδωνισμού στο push-button BTNC που αφορά την είσοδο *enable*, σας παρέχεται έτοιμο ένα πρόγραμμα VHDL (FSM_detector) το οποίο ενσωματώνει το κύκλωμα του sequence detector μαζί με έτοιμα, κατάλληλα κυκλώματα αποκλειδωνισμού (debouncer) και ανίχνευσης ανερχόμενης ακμής (edge_detect). Ο debouncer υλοποιείται με χρήση ενός μετρητή που περιμένει την σταθεροποίηση της εισόδου δειγματοληπτώντας σε τακτά χρονικά διαστήματα (πχ. ανά **10 ms=10000000ns=>1MHz**) την είσοδο και όταν ανιχνεύσει δύο ίδιες διαδοχικές τιμές, περνάει την είσοδο στην έξοδό του, οδηγώντας τον edge detector. Ο edge detector είναι απαραίτητος ώστε να αποφευχθεί η παραγωγή περισσότερων του ενός παλμού, σε περίπτωση που ο χρήστης κρατήσει πατημένο το κουμπί. Ο edge detector δέχεται την έξοδο του debouncer, ανιχνεύει μια ανερχόμενη ακμή (rising edge) και παράγει έναν παλμό διάρκειας ενός κύκλου στη συχνότητα του συστήματος (100MHz). Το κύκλωμα του edge detector υλοποιείται με την χρήση λογικών πυλών (εναλλακτικά μπορεί να σχεδιαστεί και ως FSM).

Στο e-class θα σας δοθεί τόσο η κύρια οντότητα (FSM_detector) μαζί με τον debouncer και τον edge detector όσο και ένα template πηγαίου κώδικα του sequence detector που καλείστε να υλοποιήσετε με την περιγραφή του entity του sequence detector (ουσιαστικά θέλετε να φτιάξετε ένα Moore FSM που ανιχνεύει την ακολουθία "101" και θα τροποποιήσετε τον κώδικα των διαφανειών ώστε το διάβασμα της εισόδου να γίνετε σύμφωνα με ένα σήμα enable).

ΜΕΡΟΣ Α

- Συμπληρώστε το architecture του entity sequence_detector
- Μελετήστε το testbench sequence_detector_tb σε VHDL που σας δίνεται
- Ελέγξτε τη λειτουργία του κυκλώματος του sequence_detector με προσομοίωση
- Ελέγξτε τη λειτουργία του κυκλώματος του edge_detection και για τις 2 περιπτώσεις (1st, 2nd, κοιτάξτε τα σχόλια του κώδικα) με προσομοίωση (φτιάχνετε το πρόγραμμα testbench).

ΜΕΡΟΣ Β

- Μελετήστε προσεκτικά το σύστημα (FSM_detector.vhd) και τα κυκλώματα του debouncer (debouncer.vhd) και του edge detection (edge_detection.vhd) που σας παρέχονται έτοιμα.
- Γράψτε το κατάλληλο αρχείο .xdc με τα pin constraints για το σύστημα (system) και ενσωματώνει το sequence detector που σχεδιάσατε μαζί με τα κατάλληλα κυκλώματα debouncing και edge detection

  Για τον ορισμό της συχνότητας ρολογιού δώστε το παρακάτω constraint
  set_property -dict {PACKAGE_PIN Y9 IOSTANDARD LVCMOS33 } [get_ports { clk }];

  Θα συμπληρώσετε μόνοι σας τις υπόλοιπες γραμμές που χρειάζονται. Στο eclass υπάρχει το manual της κάρτας ώστε να βρείτε ότι χρειάζεστε.
- Ελέγξτε τη λειτουργία του sequence detector σας μέσω του συστήματος που ενσωματώνει τον μετρητή σας στο υλικό προγραμματίζοντας το FPGA
- Συμπληρώστε τα παρακάτω στοιχεία κοιτάζοντας το report μετά το implementation:

  FF: _____
  LUT: _____
  I/O: _____
  BUFG: _____
- Προσδιορίστε την ελάχιστη περίοδο (minimum period) σε ns και το πιο κρίσιμο μονοπάτι (max delay path) εκτελώντας την εντολή *report_timing_summary –datasheet ή από τα μενού του implemented design.*

  Min Period (ns): _____
  Max Delay Path
  Slack (ns): _____
  Source: _____
  Destination: _____
  Logic Levels: _____