

Ψηφιακά Συστήματα HW σε Χαμηλά Επίπεδα Λογικής – II

Θεσσαλονίκη – Εαρινό Εξάμηνο 2021

Η εργασία αποτελείται από τρεις ανεξάρτητες και υποχρεωτικές ασκήσεις, οι οποίες περιγράφονται παρακάτω. Κάθε μία από τις ασκήσεις βαθμολογείται με άριστα το 10 και η εργασία ισοδυναμεί με το 30% του συνολικού βαθμού του μαθήματος. Το υπόλοιπο 70% του βαθμού του μαθήματος θα προέρχεται από τη γραπτή (ή προφορική λόγω πανδημίας) εξέταση. Η προθεσμία υποβολής της εργασίας είναι την **Κυριακή 6 Ιουνίου** στις **23:59**. Εκπρόθεσμη υποβολή εργασίας ισοδυναμεί με 0 βαθμούς για το εργαστηριακό κομμάτι του μαθήματος.

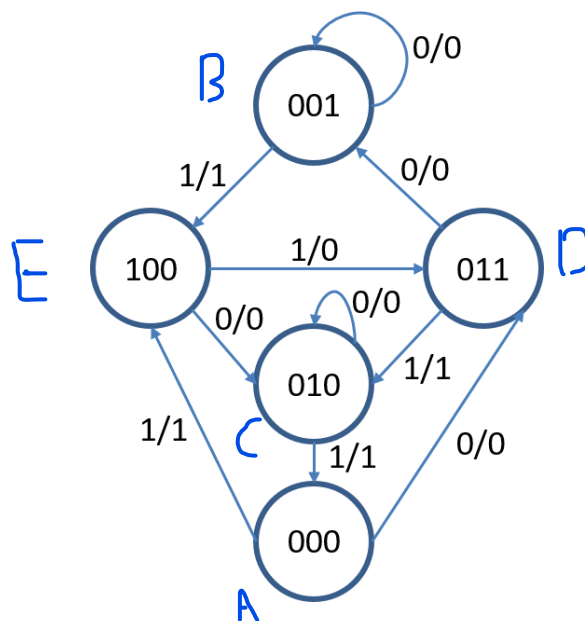
Άσκηση 1

Περιγράψτε με τη χρήση της γλώσσας Verilog το FSM το οποίο απεικονίζεται στο Σχήμα 1 το οποίο έχει μία είσοδο x_{in} και μία έξοδο y_{out} και **επαληθεύστε το**. Οι καταστάσεις που δε δείχνονται στο FSM μπορούν να θεωρηθούν ως συνθήκες αδιάφορης τιμής (don't cares). Χρησιμοποιήστε τη μεθοδολογία που διδαχθήκατε στο μάθημα για την παραγωγή του κυκλώματος. Το FSM εκκινεί από την κατάσταση 001 κάθε φορά που τίθεται ένα ασύγχρονο σήμα επαναφοράς reset (active-high). Η υλοποίηση θα πρέπει να γίνει με τρεις διαφορετικούς τρόπους (οδηγώντας σε αντίστοιχα μοντέλα/αρχεία Verilog):

α) χρησιμοποιήστε την κλασική περιγραφή FSM με συμπεριφορική (behavioral Verilog)

β) υλοποιήστε τα κυκλώματα του FSM (δηλαδή κύκλωμα προσδιορισμού επόμενης κατάστασης, εξόδων και διατήρησης τρέχουσας κατάστασης), όπου η κατάσταση του FSM αποθηκεύεται σε D flip-flop.

γ) παρόμοια με το υποερώτημα (β), όπου η κατάσταση του FSM αποθηκεύεται σε JK flip-flop.

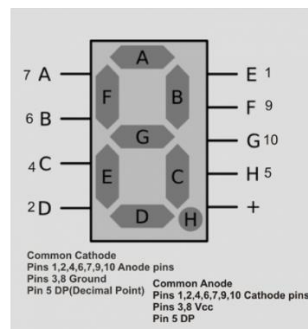


Σχήμα 1. Διάγραμμα μεταβάσεων για την άσκηση 1.

Άσκηση 2

Σχεδιάστε ένα απαριθμητή BCD τεσσάρων ψηφίων (δηλαδή μετράει από το 0000 μέχρι το 9999) με χρήση τη γλώσσας Verilog και αρχές ιεραρχική σχεδίασης (δηλαδή λύσεις με ένα μόνο module δεν είναι επιθυμητές). Επίσης, είναι επιθυμητό να εμφανίσουμε την τιμή του απαριθμητή σε ένα επτά τμημάτων ψηφίο LED όπως φαίνεται στο Σχήμα 2. Επομένως, ο κώδικάς σας θα πρέπει εκτός από την τιμή του απαριθμητή να παράγει και την τιμή των εισόδων του ψηφίου LED: A, B, C..... H, μέσω κατάλληλου αποκωδικοποιητή.

Ο μετρητής λειτουργεί οποτεδήποτε ένα σήμα επίτρεψης (En) είναι αληθές και μηδενίζεται οποτεδήποτε τίθεται ένα ασύγχρονο σήμα reset. Επίσης, μπορείτε να υποθέσετε ένα ρολόι συχνότητας 1 MHz για τη λειτουργία του απαριθμητή. Θεωρείστε ότι η έξοδος του αποκωδικοποιητή θα πρέπει να υποστηρίζει ψηφίο LED τύπου κοινής καθόδου (common cathode) και ανόδου (common anode) το οποίο θα επιλέγει ο χρήστης με ένα ακόμα σήμα ελέγχου για τον αποκωδικοποιητή (LED_type_ctl). Τα module που υλοποιούν το μετρητή θα πρέπει να υλοποιηθούν με δομική (structural) Verilog χρησιμοποιώντας βασικές πύλες (primitives). Τέλος, θα πρέπει να χρησιμοποιήσετε αποκλειστικά flip-flop τύπου T για την τιμή του απαριθμητή τα οποία επίσης θα δομηθούν με βασικές πύλες. Ο κώδικάς σας θα πρέπει να επαληθευθεί δείχνοντας τη λειτουργία του απαριθμητή και αποκωδικοποιητή με τη δημιουργία κατάλληλων δοκιμών (testbenches).

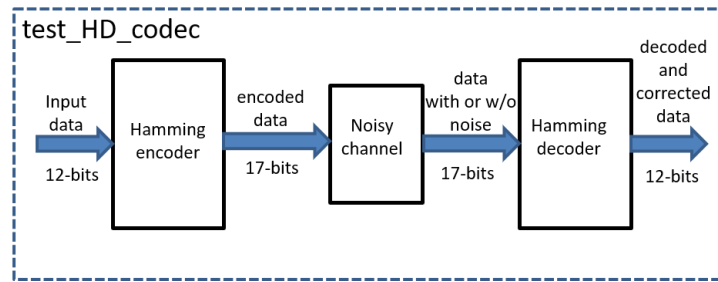


Σχήμα 2. Ψηφίο LED επτά τμημάτων (7 segment LED display).

Άσκηση 3

Περιγράψτε με τη χρήση της Verilog ένα κύκλωμα κωδικοποίησης Hamming (12,5) καθώς επίσης ανίχνευσης και διόρθωσης σφάλματος ενός bit σε λέξεις δεδομένων 12-bit. Το σύστημα που θα σχεδιάσετε θα πρέπει να επαληθευθεί ακολουθώντας τη διαμόρφωση που εικονίζεται στο Σχήμα 3. Το κύκλωμα κωδικοποίησης δέχεται (τυχαία) δεδομένα και τα κωδικοποιεί σύμφωνα με τον κώδικα Hamming (12,5) και μεταδίδει την κωδικοποιημένη λέξη στο δίαυλο επικοινωνίας όπου μπορούμε να επιλέξουμε την εισαγωγή (ή όχι) ενός λάθους σε ένα bit που επιλέγει ο χρήστης. Η λέξη αυτή λαμβάνεται, ελέγχεται για λάθη και διορθώνεται αν απαιτείται στον αποκωδικοποιητή. Δείξτε μέσω προσομοιώσεων ότι διάφορα λάθη μπορούν να διορθωθούν.

Σημείωση: Θα πρέπει να εντοπίσετε μόνοι σας τις λογικές εκφράσεις της (από)κωδικοποίησης Hamming (12,5) για λέξεις μήκους 12-bit.



Σχήμα 3. Σύστημα επαλήθευσης του κώδικα ανίχνευσης και διόρθωσης λαθών Hamming (12,5).

Οδηγίες Υποβολής και Βαθμολόγηση

Η υποβολή θα γίνει αποκλειστικά μέσω του [elearning](#) και πρέπει να περιλαμβάνει μία αναφορά (μόνο σε μορφή pdf) που να περιέχει τα ακόλουθα:

- Τον κώδικα Verilog για την κάθε άσκηση
- Τα testbench για την κάθε άσκηση
- Απαραίτητα screenshots (σας παρακαλώ να χρησιμοποιήσετε λευκό και όχι μαύρο φόντο για τις εικόνες αυτές) από τις προσομιώσεις για τα κυκλώματα της κάθε άσκησης και μία σύντομη περιγραφή των ελέγχων που έγιναν
- Συγκεκριμένα για την Άσκηση 1
 - Τον πίνακα αληθείας του FSM και για τις δύο υλοποιήσεις
 - Τον πίνακα κωδικοποίησης καταστάσεων
 - Τις λογικές εξισώσεις για τις εξόδους των FSM και για τις δύο υλοποιήσεις
- Ό,τι άλλο κρίνεται απαραίτητο για να δείξετε την επίλυση των ασκήσεων ή ό,τι ανακαλύψατε ή μάθατε.

Επίσης, μαζί με την αναφορά, πρέπει να υποβληθούν τα σχετικά αρχεία Verilog για όλες τις ασκήσεις και τα αντίστοιχα testbench. Αν χρησιμοποιήσετε κάποια αρχεία εισόδου/εξόδου για την επαλήθευση των ασκήσεων, παρακαλείσθε να τα υποβάλλετε και αυτά. Η υποβολή όλων των αρχείων να γίνει ως ένα αρχείο zip το οποίο θα ονομαστεί με το επίθετο και τον αριθμό μητρώου σας, δηλαδή ως «lastname-AEM», για τη διευκόλυνση της βαθμολόγησης.

Καλή επιτυχία!