

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών



ΑΝΑΠΤΥΞΗ ΥΛΙΚΟΥ – ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ
ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

1η Εργαστηριακή Άσκηση

Σχεδίαση απλών μονάδων υλικού με χρήση συνθέσιμης
VHDL και υλοποίηση σε FPGA

Διδάσκων: Ηλίας Μανωλάκος, Καθηγητής
Βοηθός Εργαστηρίου: Ηλίας Κουσκουμβεκάκης

Φεβρουάριος 2023

Εισαγωγικά

Στόχοι της άσκησης είναι να σας βοηθήσει :

1. Να κάνετε επανάληψη στη λογική σχεδίαση ψηφιακών κυκλωμάτων και στη χρήση της γλώσσας περιγραφής υλικού VHDL.
2. Να εξοικειωθείτε με τα εργαλεία σύνθεσης και προσομοίωσης και με την χρήση της αναπτυξιακής κάρτας του εργαστηρίου

Περιγραφή άσκησης

Πρόβλημα 1ο. Να σχεδιαστεί ψηφιακό συνδυαστικό κύκλωμα με είσοδο 'X' 4-bit, και έξοδο 'F' 3-bit. Στην έξοδο 'F' πρέπει να εμφανίζεται σε δυαδική τιμή το πλήθος των '1' που υπάρχουν στην είσοδο 'X'. Για την υλοποίηση στη κάρτα χρησιμοποιείτε ως εισόδους X_3, X_2, X_1, X_0 τέσσερις διακόπτες (switches) και συνδέστε τις εξόδους F_2, F_1, F_0 σε τρία από τα διαθέσιμα LEDs.

Το κύκλωμα να σχεδιαστεί με τους εξής δύο εναλλακτικούς τρόπους:

(a) Με περιγραφή VHDL δομής (structural) της πλέον απλοποιημένης μορφής των συναρτήσεων της 'F', ώστε να χρησιμοποιηθούν οι ελάχιστες δυνατές πύλες. Κάθε χρησιμοποιούμενη πύλη να αποτελεί ένα ξεχωριστό VHDL component που πρέπει να γίνει instantiate με port map. Οι περιγραφές των πυλών AND, OR, NOT, XOR που πιθανά θα χρειαστείτε βρίσκονται στη βιβλιοθήκη Unisim της Xilinx. Πριν ξεκινήσετε την περιγραφή του κυκλώματος σε VHDL, θα πρέπει να υπολογίσετε τον πίνακα αλήθειας των λογικών συναρτήσεων F και στην συνέχεια να απλοποιήσετε το άθροισμα γινομένων τους με χρήση πίνακα Karnaugh και άλγεβρας Boole.

(b) Με περιγραφή VHDL συμπεριφοράς (behavioral) με ένα process που υλοποιεί το ζητούμενο κύκλωμα χρησιμοποιώντας πολυπλέκτες N προς 1.

Να δημιουργηθούν διαφορετικά Vivado projects για τους δύο τρόπους περιγραφής VHDL. Αφού ελεγχθεί η σωστή λειτουργία με προσομοίωση, στην συνέχεια να φορτωθούν τα αντίστοιχα "προγράμματα" στην αναπτυξιακή κάρτα FPGA, ώστε να επιβεβαιωθεί η ορθή λειτουργία τους και στο υλικό.

Πρόβλημα 2ο. Να υλοποιηθεί κύκλωμα αντίστροφης μέτρησης (count down) συχνότητας 1 Hz. Το χρονόμετρο θα ξεκινάει τη μέτρηση από την τιμή που θα ορίζει ο χρήστης μέσω επτά διακοπών (SW6 to SW0, άρα μέγιστη τιμή το $2^7 - 1 = 127$) και θα φτάνει μέχρι και το μηδέν. Η τρέχουσα τιμή θα πρέπει να απεικονίζεται στα επτά LED της αναπτυξιακής κάρτας (LED6 to LED0) σε δυαδική μορφή. Μόλις το κύκλωμα φτάσει στο μηδέν θα πρέπει να αρχίσουν να αναβοσβήνουν και τα 8 LEDs με ρυθμό 1 Hz. Να χρησιμοποιηθεί ο διακόπτης (switch) SW7 για την έναρξη/παύση του συστήματος και για reset να χρησιμοποιηθεί ο διακόπτης πίεσης κομβίου (button) BTNC, με το οποίο θα παίρνει το κύκλωμα άμεσα την αρχικά ορισμένη τιμή και θα ξεκινάει εκ νέου η λειτουργία του κυκλώματος.

Πρόβλημα 3ο. Να υλοποιηθεί κύκλωμα υπολογισμού της ακολουθίας Fibonacci. Η τιμή που υπολογίζεται θα πρέπει να απεικονίζεται στα οκτώ LED της αναπτυξιακής κάρτας (LED7 to LED0) σε δυαδική μορφή (μη προσημασμένοι αριθμοί), με συχνότητας 1 Hz. Το κύκλωμα σταματά στην τιμή 233, η οποία είναι η μέγιστη που μπορεί να απεικονιστεί στα 8 LED. Να χρησιμοποιηθεί ο διακόπτης (switch) SW0 για την έναρξη/παύση του συστήματος και ο διακόπτης SW1 για την μετατροπή του κυκλώματος, ώστε να υπολογίζει την αντίστροφη ακολουθία Fibonacci, από την τιμή που βρισκόταν. Κατά την αντίστροφη λειτουργία το κύκλωμα σταματά στην τιμή 0. Επίσης, για reset να χρησιμοποιηθεί ο διακόπτης πίεσης κομβίου (button) BTNC.

Στα προβλήματα 2 και 3 μπορείτε να χρησιμοποιήσετε ένα συνδυασμό από structural, dataflow και behavioral VHDL περιγραφές. Θεωρήστε ότι έχετε διαθέσιμα όλα τα συστατικά στοιχεία που παρέχονται από το FPGA και αξιοποιείτε όπου είναι δυνατόν τις δυνατότητες του εργαλείου σύνθεσης.

Bonus Πρόβλημα (Προαιρετικό, για έξτρα πόντους μέχρι και 25%): Καλείστε να προσθέσετε στο σχέδιο του 2^{ου} και του 3^{ου} προβλήματος επιπλέον λειτουργίες. Να γίνεται απεικόνιση των τιμών που παράγουν τα κυκλώματα χρονομετρητή και Fibonacci στο 7 segment display της αναπτυξιακής κάρτας. Για το κύκλωμα του χρονομετρητή θεωρούμε μέγιστη την τιμή 99 και για το κύκλωμα του Fibonacci την τιμή 89 (λόγω ύπαρξης μόνο 2 ψηφίων στο 7 segment display).

“Hints and tips”

- Για την υλοποίηση του κυκλώματος του 1^{ου} Προβλήματος με τον 1^ο τρόπο, θα πρέπει να κάνετε χρήση της βιβλιοθήκης Unisim, η οποία παρέχεται από την Xilinx και περιέχει έτοιμα components λογικών κυκλωμάτων, μεταξύ αυτών και για τις πύλες που θα χρειαστείτε για την υλοποίηση του κυκλώματος σας. Για να χρησιμοποιήσετε τη βιβλιοθήκη αυτή χρειάζεται να προσθέσετε στο πρόγραμμά σας τις δηλώσεις:

library UNISIM;

use UNISIM.VComponents.all;

Οι ορισμοί των components βρίσκονται στον εξής φάκελο της εγκατάστασης του Vivado:

‘Vivado Installation/Vivado/202x.x/data/vhdl/src/unisims/[primitive, retarget]/

- Σχετικά με την υλοποίηση του κυκλώματος στο 2^ο πρόβλημα θα χρειαστεί να φτιάξετε μία μονάδα διαίρεσης συχνότητας για το ρολόι των 100 MHz που διαθέτει η κάρτα. Ανατρέξτε στον κώδικα που δίνεται για στις διαφάνειες ‘VHDL.Examples.pdf’ (slides 7 & 8) στο eclass του μαθήματος (Εργαστήριο/VHDL) για να κατανοήσετε τον τρόπο σχεδίασης μιας τέτοιας μονάδας.
- Θα πρέπει να αναπαραστήσετε όποια εσωτερικά σήματα χρειάζεστε για τις εισόδους και εξόδους των πολλαπλασιαστών και του αθροιστή που θα χρησιμοποιήσετε, με τον τύπο ‘signed’ της VHDL από τη IEEE βιβλιοθήκη ‘numeric_std’. Ο συγκεκριμένος τύπος δεδομένων αναπαριστά αριθμούς 2’s complement ώστε να γίνονται σωστά οι πράξεις του πολλαπλασιασμού και της πρόσθεσης. Μπορείτε επίσης να χρησιμοποιείτε με προσοχή τον τύπο ‘integer’, θέτοντας πάντα κατάλληλο ‘range’. Τα σήματα των οντοτήτων (entities) μπορούν να είναι τύπου ‘signed’ ή ‘std_logic_vector’ και θα τα μετατρέπετε πριν ή / και αφού τα χρησιμοποιήσετε στην κάθε μονάδα για πολλαπλασιασμό ή πρόσθεση με κατάλληλες συναρτήσεις μετατροπής ‘std_logic_vector()’, ‘signed()’, ‘to_signed()’ και ‘to_integer()’. Τα top-level entities θα πρέπει αναγκαστικά να έχουν σήματα τύπου ‘std_logic_vector’.
- Θα πρέπει να δώσετε ιδιαίτερη προσοχή να μη δημιουργούνται latches κατά τη σύνθεση κάνοντας reset όλα τα στοιχεία μνήμης (flip-flops / registers) στο σχέδιο

σας ή φροντίζοντας όλα τα μονοπάτια στον κώδικα συμπεριφοράς σας (behavioral process) να θέτουν τιμές σε αυτά.

- Όλες οι εισοδοι και οι έξοδοι του κυκλώματος που θα υλοποιήσετε θα πρέπει να αντιστοιχηθούν σε κατάλληλους ακροδέκτες του FPGA, ώστε το κύκλωμα να είναι δυνατό να λειτουργήσει σωστά πάνω στην αναπτυξιακή κάρτα. Η αντιστοίχιση αυτή γίνεται με τη χρήση ενός αρχείου *xdc*. Στο αρχείο αυτό αναφέρονται όλα τα σήματα που υπάρχουν στο entity του σχεδίου που υλοποιούμε και κάθε ένα από αυτά αντιστοιχείται σε έναν ακροδέκτη του FPGA. Στο eclass στο αρχείο *example.xdc* (φάκελος Εργαστήριο) μπορείτε να βρείτε ένα παράδειγμα συγγραφής αρχείου τύπου *xdc* το οποίο μπορείτε να διαμορφώσετε για την υλοποίηση της άσκησης. Τα ονόματα των σημάτων θα πρέπει να ταυτίζονται με αυτά των σημάτων που θα χρησιμοποιήσετε στο entity του σχεδίου σας. Το αρχείο *xdc* θα πρέπει να το εισάγετε στο project που δημιουργήσατε στο Vivado.

Παραδοτέα

Η άσκηση θα διαρκέσει 3 εργαστήρια (**Τετάρτη 22/2/2023, 1/3/2023 και 8/3/2023**). Μία εβδομάδα μετά την ολοκλήρωση της (**Τετάρτη 15/03/2023**) θα πρέπει να παραδώσετε στο e-class την υλοποίηση των προβλημάτων μαζί με συνοπτικό κείμενο σε μορφή τεχνικής έκθεσης (έγγραφο .doc μαζί με .pdf ή μόνο το .pdf) που να περιλαμβάνει τουλάχιστον τις παρακάτω ενότητες (sections) με ακρίβεια και σαφήνεια:

☐ Για το 1^ο πρόβλημα:

- Όλα τα βήματα της λογικής σχεδίασης που ακολουθήσατε (λύσεις ερωτημάτων).
- Τα σχηματικά ανώτερου επιπέδου που δείχνουν τα I/O ports των κυκλωμάτων που προέκυψαν, για κάθε ερώτημα.
- Τους πόρους που δεσμεύονται από α) τη σύνθεση και β) την υλοποίηση, για κάθε ερώτημα. Σύγκριση και σχολιασμός μεταξύ των αποτελεσμάτων της σύνθεσης και υλοποίησης για κάθε ερώτημα και μεταξύ της υλοποίησης των 2 ερωτημάτων (ποιοό είναι το πιο αποδοτικό;).

- Τεκμηρίωση μέσω προσομοίωσης της ορθής λειτουργίας των σχεδίων σας, για κάθε ερώτημα. Να παρουσιάσετε όσο πιο αναλυτικά τις προσομοιώσεις με screenshots στην έκθεση που θα παραδώσετε.

□ Για το 2^ο και το 3^ο πρόβλημα:

- Την κάθε μονάδα που σχεδιάσατε και το σύστημα που προέκυψε.
- Τα σχηματικά ανώτερου επιπέδου που δείχνουν τα I/O ports των κυκλωμάτων που προέκυψαν, για κάθε ερώτημα.
- Τους πόρους που δεσμεύονται από α) τη σύνθεση και β) την υλοποίηση. Σύγκριση και σχολιασμός μεταξύ των αποτελεσμάτων της σύνθεσης και υλοποίησης.
- Τη μέγιστη συχνότητα που μπορεί να επιτύχει το εργαλείο λαμβάνοντας υπόψη την τιμή WNS (Worst Negative Slack). Θυμίζουμε πως **Max Freq = 1 / (Clock Period - WNS)**. Θα πρέπει επίσης σε αυτή την ενότητα να παρουσιάσετε το κρίσιμο μονοπάτι (critical path) που αναφέρει το timing report που παράγει το εργαλείο και να εξηγήσετε (όσο μπορείτε) τι μας δείχνει αυτό και ποιές μονάδες και σήματα το δημιουργούν στο σχέδιο σας.
- Τεκμηρίωση μέσω προσομοίωσης της ορθής λειτουργίας του συστήματος (με διαφορετικά σενάρια συμπεριλαμβανομένων και ακραίων περιπτώσεων) και αν υπάρχουν υπομονάδες που συνθέτουν το τελικό σύστημα να γίνει προσομοίωση και για αυτές ανεξάρτητα. Να παρουσιάσετε όσο πιο αναλυτικά τις προσομοιώσεις με screenshots στην έκθεση που θα παραδώσετε.

Να συμπεριλάβετε όποιες παρατηρήσεις θέλετε σχετικά με τα προβλήματα ή / και για τις δυσκολίες που παρουσιάστηκαν και πως τις αντιμετωπίσατε. Σε περίπτωση που δεν υλοποιούνται πλήρως οι προδιαγραφές να σχολιάζονται οι τυχόν αποκλίσεις. Επίσης να αναφέρετε τυχόν αναφορές σε βιβλία, εγχειρίδια κτλ που χρησιμοποιήθηκαν. Για λόγους οικονομίας χώρου να μην συμπεριληφθεί ο κώδικας VHDL στην έκθεση.

Όλα τα παραδοτέα και η έκθεση θα πρέπει να συμπιεστούν σε φάκελο **με το όνομα και τον αριθμό μητρώου σας** και να κατατεθούν στο eclass στο πεδίο 'Εργασίες'. Για κάθε πρόβλημα και ερώτημα να υπάρχει ξεχωριστός υπο-φάκελος που να περιλαμβάνει τον κώδικα VHDL επαρκώς σχολιασμένο, καθώς και τα αρχεία με τα

οποία προσομοιώσατε το σχέδιό σας. Επίσης να υπάρχει η υλοποίηση που κατεβαίνει στο FPGA board και λειτουργεί με τους διακόπτες / leds (bitstream στο path "vivado project/project_name.runs/impl_1/project_name.bit").

Αξιολόγηση

Οι εργαστηριακές ασκήσεις είναι υποχρεωτικές. Κατά την αξιολόγηση, ιδιαίτερη έμφαση θα δοθεί στη ορθότητα της υλοποίησης, στη πληρότητα του ελέγχου της υλοποίησης και της παρουσίασης των αποτελεσμάτων. Είναι πιθανόν να κληθείτε σε επιπλέον προφορική εξέταση επί του σχεδίου σας μετά την παράδοσή του.

Προσοχή: Καθυστερημένες εργασίες δεν θα γίνουν δεκτές ούτε θα βαθμολογηθούν. Κάθε φοιτητής οφείλει να εργάζεται μόνος του. Η ανταλλαγή απόψεων επί της σχεδίασης επιτρέπεται, όμως απαγορεύεται ρητά η ανταλλαγή σχεδίων, κώδικα και εκθέσεων. Αν υπάρχει υπόνοια αντιγραφής όλοι οι εμπλεκόμενοι θα μηδενιστούν στο εργαστήριο.

Χρήσιμα Αρχεία

Στη σελίδα eclass του μαθήματος θα βρείτε χρήσιμα αρχεία και συνδέσμους για την εργασία σας στον κατάλογο **Υποστηρικτικό υλικό** και στον κατάλογο **Εργαστήριο**.