Snake Game

Μαρέλας Γεώργιος
Ηλεκτρολόγων Μηχανικών &
Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας
Ελλάδα, Βόλος
gmarelas@gmail.com

Μπούσδρας Γεώργιος
Ηλεκτρολόγων Μηχανικών &
Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας
Ελλάδα, Βόλος
george.bousdras@gmail.com

Περιγραφή.

Αντικείμενο της εργαστηριακής άσκησης είναι η υλοποίηση του παιχνιδιού «Φιδάκι» σε FPGA πλακέτα(Nexys 3 Spartan6) με γλώσσα περιγραφής υλικού Verilog. Το φιδάκι θα εμφανίζεται στην οθόνη και θα κινείται πάνω-κάτω-δεξιά-αριστερά ανάλογα με ποιο κουμπί πατήθηκε από την πλακέτα.

Σκοπός του παιχνιδιού είναι το φιδάκι να καταναλώσει όσο το δυνατόν περισσότερη τροφή χωρίς να χτυπήσει το κεφάλι του στο σώμα του ή στον τοίχο.

Παρακάτω θα αναλύσουμε τον τρόπο υλοποίησης.

I. ΕΙΣΑΓΩΓΗ

Η επιλογή της συγκεκριμένης εργασίας έγινε πρωτίστως με γνώμονα την διεύρυνση των γνώσεων μας σχετικά με την Γλώσσα Περιγραφής Υλικού (hdl) Verilog. Εξίσου σημαντική ήταν η επαφή με την εξερεύνηση της τεχνογνωσίας των FPGA πλακετών καθώς και των εργαλείων της Xillinx (ISE,iSIM,iMPACT) τα οποία χρησιμοποιούνται ευρέως για την υλοποίηση και την επαλήθευση ψηφιακών κυκλωμάτων.

Η συγγραφή του προγράμματος εξολοκλήρου σε γλώσσα Verilog σε συνδυασμό με την χρήση των περιφερειακών της πλακέτας καθ'όλη την διάρκεια της υλοποίησης του, οδήγησε στην επίτευξη του στόχου. Κατά την διάρκεια υλοποίησης της εργασίας τα κυριότερα προβλήματα που αντιμετωπίσαμε ήταν η κατανόηση της λειτουργίας της θύρας VGA, ο τρόπος εμφάνισης ενός αντικειμένου στην οθόνη, η πραγματοποίηση κίνησης του, καθώς και ο τρόπος καθοδήγησης του.

Αρχικά τα αποτελέσματα ήταν αποθαρρυντικά καθώς η απειρία μας στον τομέα των γραφικών οδήγησε σε επαναλαμβανόμενες αλλοιώσεις των αντικειμένων προς απεικόνιση. Για παράδειγμα, το σώμα από το φιδάκι απεικονιζόταν σε μόλις ένα pixel και το φαγητό αρχικά εμφανιζόταν σε πολλαπλές θέσεις μέχρι να σταθεροποιηθεί η τελική του θέση. Επίσης η τυχαία εμφάνιση του φαγητού μέσα στα επιτρεπόμενα πλαίσια δεν ήταν εφικτή με αποτέλεσμα την συνεχόμενη εμφάνιση στο ίδιο σημείο. Σημαντικό επίσης πρόβλημα αποτέλεσε η επικάλυψη των αντικειμένων με λανθασμένη προτεραιότητα. Αρκετές φορές κατά τον προγραμματισμό της πλακέτας δεν υπήρχε καμία ανταπόκριση των αντικειμένων στα εισερχόμενα σήματα των κουμπιών.

Αρκετές απαντήσεις δόθηκαν στα παραπάνω προβλήματα, και ιδιαίτερα στα προβλήματα χρονισμού μελετώντας τα εγχειρίδια χρήσης των περιφερειακών. Η εκτεταμένη προσομοίωση του κυκλώματος οδήγησε στην κατανόηση των τιμών των διαφόρων σημάτων που δημιουργούσαν λανθάνουσες καταστάσεις.

Τα προβλήματα και τα σφάλματα εξάλήφθηκαν με επιτυχία μετά από αρκετές δοκιμές. Το επιθυμητό αποτέλεσμα επήλθε τόσο για την σωστή λειτουργία του παιχνιδιού όσο και για την αποκόμιση εμπειριών και τεχνογνωσίας.

II. ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ

Κάνοντας μια πρώτη ανασκόπηση των αναγκών της σχεδίασης καταλήξαμε στο γεγονός ότι χρείαζομασταν 7 διαφορετικά modules για να ικανοποιηθούν οι απαιτήσεις του κυκλώματος. Η κατάτμηση των αναγκών μας έγινε βάση των επιμέρους χαρακτηριστικών του παιχνιδιου τα οποία είναι η εμφάνιση στην οθόνη, η κίνηση, η δημιουργία φαγητού , ο χρωματισμός των αντικειμένων και το φιλτράρισμα του θορύβου από τα εξωτερικά κουμπιά.

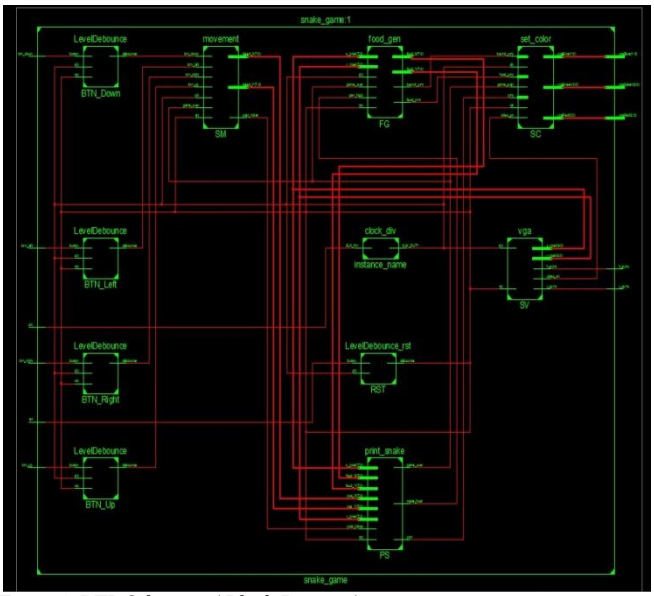

Figure 1. RTL Schematic(Block Diagram)

Το πρωταρχικό module είναι το snake_game.v το οποίο περιλαμβάνει την σύνδεση των επιμέρους module.Αποτελείται από τα σήματα εισόδου clk , rst, btn_up/down/left/right και ως έξοδο δίνονται τα 5 σήματα που χρειάζονται για να λειτουργήσει η θύρα VGA. Μέσα στο domain module πραγματοποιείται η διαίρεση του ρολογιού της FPGA από τα 100 MHz της κανονικής της λειτουργίας στα 25.175 MHz.

Στα επιμέρους modules ξεκινάμε με τον VGA controller. Ο σκοπός της διαίρεσης του ρολογιού είναι η διευκόλυνση μας στον υπολογισμό των κατάλληλων τιμών που απαιτούνται για τον χρονισμό της οθόνης σε ανάλυση 640x480. Επομένως ως είσοδο παίρνει το διαιρεμένο ρολόι και το σήμα reset. Στην έξοδο έχουμε τα 2 σήματα χρονισμού horizontical_synchronization και vertical_synchronization των οποίων η συχνότητα τους καθορίζει την λειτουργία της οθόνης στην επιθυμητή ανάλυση. Επίσης έχουμε το video_on το οποίο ενεργοποιείται όταν βρισκόμαστε στο ορατό πεδίο της οθόνης καθώς και τα 3 σήματα χρωματισμού, Red, Green, Blue.Τέλος δίνει και τα σήματα row_pixel και colum_pixel τα οποία κάθε χρονική στιγμή καθορίζουν σε ποιο pixel βρισκόμαστε.

Απαραίτητο module για την σωστή λειτουργία της κίνησης είναι το debouncer. Υλοποιείται για την εξάλειψη του θορύβου που προκαλούν τα εξωτερικά κουμπιά. Δέχεται σαν είσοδο το διαιρεμένο ρολόι, το reset και το σήμα του κουμπιού με το αντίστοιχο jitter. Κατά την είσοδο του φιλτράρεται το σήμα, δειγματοληπτείται και δίνει ως έξοδο την σωστή τιμή του σήματος (button).

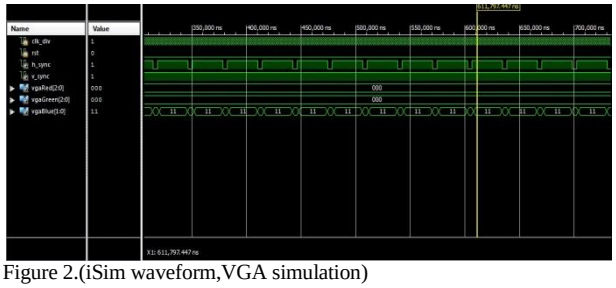

Figure 2.(iSim waveform,VGA simulation)

Σε λογική συνέχεια των 2 παραπάνω modules δημιουργήθηκε το module movement. Στην είσοδό του δέχεται το ρολόι, το reset, το game_over(το όποιο είναι output του module Print_Snake) και τα 4 buttons που έρχονται από τον debouncer. Στην έξοδο του ενεργοποιεί το Head_X και Head_Y τα οποία δείχνουν την επόμενη θέση του Snake Head . Ενεργοποιείται και το post_ticket το οποίο είναι σήμα ελέγχου για το κάθε πόσο ανανεώνονται οι θέσεις των αντικειμένων .Στο πρώτο always block καθορίζεται η ενεργοποίηση του post_ticket συναρτήσει ενός μετρητή που έχουμε ορίσει. Στο δεύτερο always_block καθορίζεται η επιθυμητή κίνηση σύμφωνα με τα κουμπιά επιλογής .Στο τρίτο ελέγχουμε αν η επιλογή κίνησης είναι εφικτό να πραγματοποιηθεί εξετάζοντας την παρούσα κίνηση του κεφαλιού. Στο τελευταίο always_block έχοντας ορίσει την επόμενη φορά κίνησης του φιδιού ελέγχουμε και ορίζουμε τις νέες συντεταγμένες του Snake Head. Επομένως κάθε φορά που το post_ticket ενεργοποιείται και γίνεται ο παραπάνω έλεγχος το Snake Head θα αλλάξει ξανά συντεταγμένες . Έτσι επιτυγχάνεται η κίνηση.

Το module Print_Snake δέχεται ως είσοδο το ρολόι και το reset, τα σήματα από τον VGA_Controller row_pixel και colum_pixel, τα σήματα post_ticket, pos_X και pos_Y (αντιστοιχούν στις συντεταγμένες του κεφαλιού) από το movement και τα food_X και food_Y(συντεταγμένες τροφής από το module food_gen). Στην έξοδο του έχουμε το prnt,το game_over και το make_food .Για να δημιουργήσουμε το snake_body έχουμε δεσμεύσει μία μνήμη 101 θέσεων των 16 bits ανά θέση. Στο πρώτο always_block, σύμφωνα με τις τιμές που έχει η κάθε θέση του snake_body και τις υπάρχουσες τιμές των r_pixel και c_pixel πραγματοποιείται η ενεργοποίηση του prnt. Με αυτό τον τρόπο επιτυγχάνεται η εμφάνιση του snake κάθε χρονική στιγμή. Στο δεύτερο always_block δημιουργούμε την κίνηση στο φιδάκι με την ανάθεση της τιμής της προηγούμενης θέσης στην επόμενη. Έτσι το snake_body ακολουθεί τις μεταβολές του snake_head. Στο ίδιο module γίνεται έλεγχος εάν το snake_head βρεθεί στις ίδιες συντεταγμένες με το food.Εάν ισχύει τότε αυξάνεται το μήκος του και ενεργοποιείται το σήμα make_food ώστε να ξαναδημιουργηθεί τροφή σε νέα θέση. Επίσης γίνεται και έλεγχος εάν το κεφάλι έχει συναντήσει κάποιο μέρος του σώματος του ή κάποιο από τα όρια του πλαισίου. Σε αυτήν την περίπτωση ενεργοποιείται το game_over και το παιχνίδι σταματάει.

Για την δημιουργία του φαγητού υλοποιείται το module food_gen. Στην είσοδο του εκτός από το ρολόι και το reset, συνδέονται τα r_pixel και c_pixel, το gen_food (αντιστοιχεί στο output make_food του Print_Snake) και το game_over από το ίδιο module. Η έξοδος του μας δίνει τις συντεταγμένες του food(food_X,food_Y) οι οποίες συνδέονται στο Print_Snake, καθώς και τα σήματα bound_prnt και food_prnt(inputs στο Module set_color). Αποτελείται από 3 always_block. Στο πρώτο εμφανίζει το food συναρτήσει των r_pixel και c_pixel. Το δεύτερο εμφανίζει τα όρια του πλαισίου στα οποία κινείται το φιδάκι. Στο τρίτο έχουμε 2 μετρητές οι οποίοι τρέχουν μεταξύ των ορίων του πλαισίου και δημιουργούν τυχαίες πιθανές θέσεις για την τροφή. Τέλος στο τέταρτο always_block αφού ενεργοποιηθεί το gen_food, το food παίρνει τις συντεταγμένες που δημιουργήθηκαν τυχαία από το προηγούμενο always_block.

Το τελευταίο module που δημιουργείται , set_color, είναι υπεύθυνο για τον χρωματισμό των αντικειμένων. Εκτός από ρολόι και reset δέχεται ως εισόδους το video_on από τον VGA Controller, το prnt από το Print_Snake, bount_prnt από το food_gen και το game_over. Στην έξοδό του έχουμε τα σήματα για τον χρωματισμό(Red,Green,Blue). Για τον έλεγχο του game_over χρησιμοποιείται ένας σημαφόρος. Χρησιμοποιείται ιεραρχία για την εμφάνιση των αντικειμένων. Αρχικά γίνεται έλεγχος για το game_over. Ακολουθεί ο χρωματισμός του πλαισίου. Έπειτα χρωματίζεται το φιδάκι και τέλος η τροφή.

III. ΑΞΙΟΛΟΓΗΣΗ

Αρχικά δημιουργήσαμε τον VGA Controller. Εφόσον διαπιστώσαμε την σωστή λειτουργία του αρχίσαμε να δίνουμε γνωστές τιμές ώστε να επαληθεύσουμε τις τιμές που είχαμε υπολογίσει για την τοποθέτηση των αντικειμένων στον «καμβά». Δεύτερος στόχος μας ηταν να φιλτράρουμε τους θορύβους μετά το πάτημα των κουμπιών. Εφόσον επετεύχθησαν οι 2 παραπάνω στόχοι επόμενο βήμα αποτέλεσε η εύρεση αλγορίθμου για την κίνηση του φιδιού και ο τρόπος εμφάνισης του. Το βασικότερο πρόβλημα για την κίνηση του φιδιού ήταν το ότι δεν μπορούσαμε να δεσμεύσουμε δυναμικά μνήμη ώστε να αυξάνεται το φιδάκι. Επομένως έπρεπε να δεσμεύσουμε στατικά ένα μεγάλο μέρος μνήμης όπου κάθε θέση υποδηλώνει ένα σημείο του σώματος του φιδιού. Μεγάλη δυσκολία επίσης συναντήσαμε στον τρόπο τυχαίας θέσης δημιουργίας του νέου φαγητού. Στην αρχή η προσπάθεια έγινε με την χρήση διαφόρων μαθηματικών εξισώσεων αφού είναι γνωστό ότι στη Verilog δεν υπάρχει κάποια συνάρτηση όπως σε άλλες γλώσσες προγραμματισμού. Αυτός ο τρόπος απέτυχε γενικά και οδηγηθήκαμε στην δεύτερη μέθοδο όπως αναλύθηκε παραπάνω.

Για την ορθή λειτουργία του κυκλώματος χρησιμοποιήσαμε τον έλεγχο με την μέθοδο των κυματομορφών καθώς και την άμεση προβολή των αποτελεσμάτων στην οθόνη. Ο πρώτος τρόπος χρησιμοποιήθηκε εφόσον δεν ήταν δυνατόν να ελεγχθούν οι τιμές των εσωτερικών σημάτων που δεν είχαν άμεση προβολή στην οθόνη. Μία τέτοια περίπτωση, όπως προαναφέραμε, ήταν η δημιουργία τυχαίας τροφής. Ακόμη μια περίπτωση που επιλύθηκε με τον εξής τρόπο ήταν το πάγωμα της οθόνης κατά το φόρτωμα του κυκλώματος στην πλακέτα. Με την δεύτερη μέθοδο καταφέραμε να επιλύσουμε προβλήματα αλλοίωσης της εικόνας ή λανθασμένων ορίων και προτεραιοτήτων. Για παράδειγμα ένα τέτοιο πρόβλημα ήταν η προβολή του φιδιού πάνω από τα όρια του πλαισίου.

IV. ΣΥΝΟΨΗ

Το αποτέλεσμα από την όλη διαδικασία που πραγματοποιήθηκε ήταν η εξοικείωση μας με τα επιμέρους περιφερειακά της πλακέτας , όπως το VGA, τα κουμπιά και άλλα. Επίσης σημαντική παράμετρος είναι η γνώση που αποκτήθηκε κατά την προσπάθεια επίλυσης των προβλημάτων που παρουσιάστηκαν καθώς και η απόκτηση εμπειρίας πάνω στον συγκεκριμένο τομέα. Η ενασχόληση με εργαλεία τέτοιου είδους μας βοήθησε στην ευκολότερη επίλυση των προβλημάτων μας. Συναρτήσει όλων των προηγούμενων καταφέραμε, παρά τα προβλήματα που αντιμετωπίσαμε, να φέρουμε εις πέρας την όλη διαδικασία και να επιτύχουμε ένα επιθυμητό και ικανοποιητικό αποτέλεσμα.

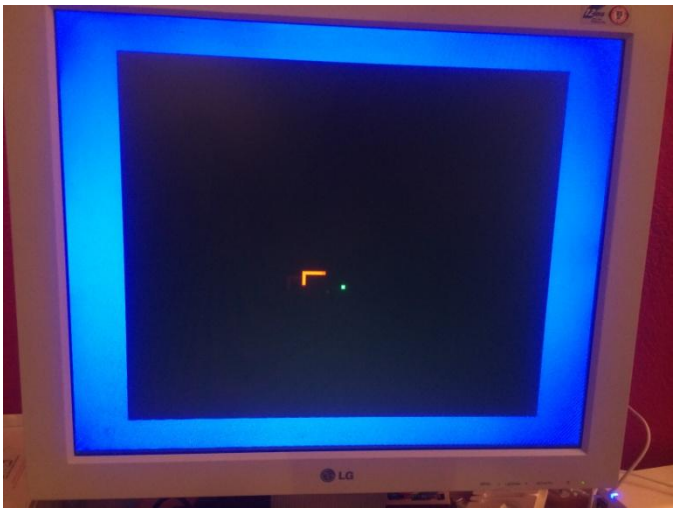

Figure 3.Snake Game in action