



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΩΝ ΜΕ ΧΡΗΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΦΟΡΑ PROJECT:

ΜΕΛΗ ΟΜΑΔΑΣ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ

A.M.

E-MAIL

Οδυσσέας Ζάχος

1072640

st1072640@ceid.upatras.gr

Βλάσιος Παναγιώτης Παναγιώτου

1067517

st1067517@ceid.upatras.gr

Κωνσταντίνος Παρασκευόπουλος

1072608

st1072608@ceid.upatras.gr

Ανάλυση των επιμέρους Modules

- **module div_clk_100_2_25** : Το συγκεκριμένο Module παίρνει σαν είσοδο το CLK του FPGA (100 MHz) και το διαιρεί δημιουργώντας ρολόι 25MHz το οποίο χρησιμοποιείται για την λειτουργία της οθόνης.
- **module div_clk_25Mhz_4Khz**: Το συγκεκριμένο Module παίρνει σαν είσοδο το CLK του FPGA (100 MHz) και το διαιρεί στα 6250MHz (~0.156ms) για την δημιουργία του σήματος PWM. Το Module αυτό χρησιμοποιεί 3 μετρητές modulo 25.
- **module pwm**: Το συγκεκριμένο Module παράγει το σήμα PWM παίρνοντας σαν είσοδο την τιμή compare_value η οποία παράγεται από το module scan2address που θα εξηγήσουμε παρακάτω και ορίζει την διάρκεια (ακέραια πολλ/σια του 0,156ms) για την οποία ο παλμός θα έχει την τιμή 1.

F	6	0.936ms
Q	8	1.248ms
H	10	1.56ms
X	13	2.028ms

Το παραπάνω module υλοποιήθηκε με προσεγγιστική λογική διότι στην περίπτωση που προσπαθήσαμε να παράξουμε ρολόι με περίοδο 0,25ms (1*cnt25 και 3*cnt10) υπήρχαν προβλήματα συγχρονισμού και το κύκλωμα δεν λειτουργούσε.

- **module Counters**: Το συγκεκριμένο Module μετρά τις συντεταγμένες του Pixel που «χρωματίζεται» την δεδομένη στιγμή
- **module sync_signal_generator**: Το συγκεκριμένο Module ελέγχει τις συντεταγμένες για παράξει αντίστοιχα τα σήματα hsync, vsync και video_on(όταν οι συντεταγμένες βρίσκονται εντός της ορατής περιοχής της οθόνης και πρέπει τα pixel να χρωματιστούν κατάλληλα)
- **module characterRom**: Στο συγκεκριμένο Module χρησιμοποιούμε την λογική των LUTs για να σχεδιάσουμε τους χαρακτήρες που θα εμφανίζονται στην οθόνη
- **module kbd_protocol**: Το συγκεκριμένο Module το χρησιμοποιήσαμε από την άσκηση 8 με την μόνη διαφορά ότι προσθέσαμε την έξοδο enable η οποία ενεργοποιείται όταν αφεθεί το πλήκτρο και χρησιμεύει σαν σήμα ενεργοποίησης καταχωρητών που σχετίζονται με το πληκτρολόγιο και την εμφάνιση των χαρακτήρων
- **module scan2Address**: Το συγκεκριμένο Module λαμβάνει σαν είσοδο το scan_code από το παραπάνω module και το αντιστοιχίζει στην τιμή που θα πάρει το σήμα compare_value που χρησιμοποιήσαμε στο module pwm ενώ το αντιστοιχίζει και στο πρώτο μισό της διεύθυνσης του χαρακτήρα που θέλουμε να εμφανίσουμε στο VGA Monitor και βρίσκεται στην **characterRom**.
- **module charRegister**: Το συγκεκριμένο module αφού λάβει τις τιμές από το παραπάνω module και υπολογίσει με βάση τις συντεταγμένες X,Y την τιμή που πρέπει να θέσει στα RGB pins του VGA (είτε 1FF είτε 000 ανάλογα με το εάν το pixel-bit (που διαβάζουμε από την characterROM)

που πρέπει να εκτυπωθεί είναι εντός της περιοχής που εκτυπώνεται ο χαρακτήρας {217-233 Vertical}, {396-404 Horizontal} και ανάλογα με το αν είναι 1 ή 0)

- **module vga_controller:** Το συγκεκριμένο Module χρησιμοποιεί τα υποmodules **div_clk_100_2_25**, **Counters**, **sync_signal_generator** για να ελέγξει την έξοδο στο VGA
- **module vga_test:** Το συγκεκριμένο είναι το top Module του σχεδιασμού μας στο οποίο αρχικοποιούνται τα αντίστοιχα υποmodules σχετικά με το VGA, το PS/2 πληκτρολόγιο, την διαχείριση χαρακτήρων, και το PWM και ελέγχονται με κατάλληλα σήματα για την υλοποίηση του ολοκληρωμένου κυκλώματος ΣΑΠΕΡΒΑΟ.

Στο αρχείο “parameters.v” ορίζονται GLOBAL μεταβλητές που χρησιμοποιούνται σε όλα τα αρχεία.

Για την υλοποίηση του παραπάνω κυκλώματος χρησιμοποιήσαμε το παρακάτω (ομολογουμένως αρκετά απλοϊκό) block Diagram που μας βοήθησε να μην χαθούμε στην υλοποίηση Modules και ΥποModules

