

Υλοποίηση Ελεγκτή VGA

Εργαστήριο Ψηφιακών Συστημάτων (2023-24)

Ιωάννης Αθανασιάδης 03491

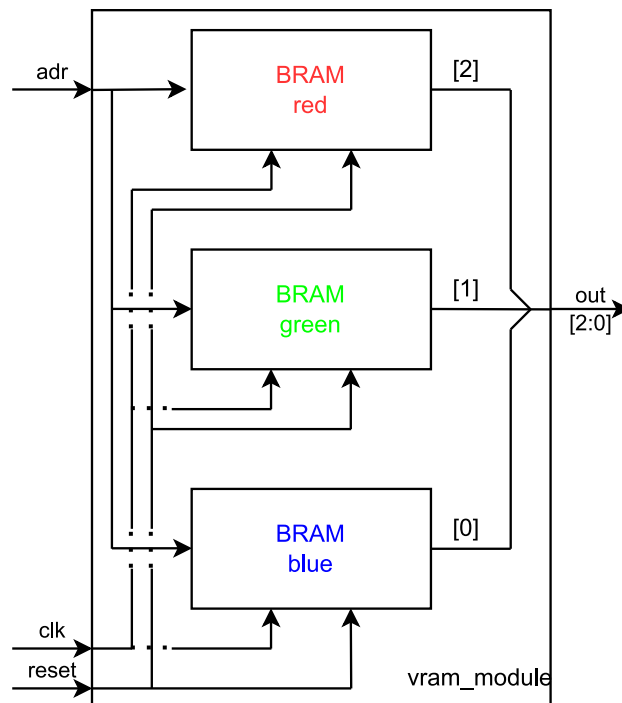
13/01/2024

Μέρος Α – Υλοποίηση VRAM

Για την υλοποίηση της **Video-RAM** χρησιμοποιούμε τρεις *block-RAM* των **16K**, μία για κάθε χρώμα του **RGB**, έτσι διαμορφώνουμε μία εικόνα με διαστάσεις **128x96**.

Για *top module* χρησιμοποιούμε το `vram_module`

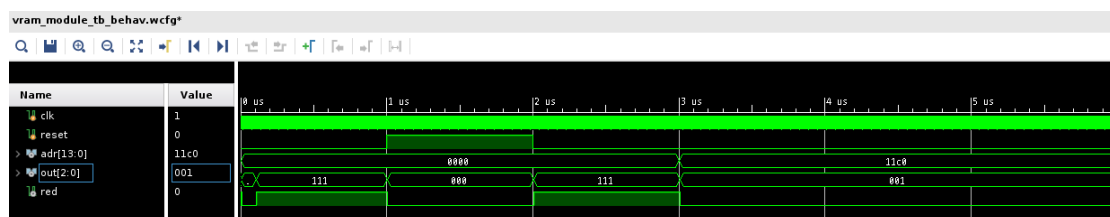
Μονάδα `vram_module`



Πλαίσιο δοκιμής

Στο πλαίσιο δοκιμής απλά δοκιμάζουμε το `reset` και έπειτα αλλάζουμε την **διεύθυνση** της μνήμης γνωρίζοντας το τι περιέχει η **VRAM**.

Παρακάτω υπάρχει ένα **screenshot** που επαληθεύει την σωστή λειτουργία της μνήμης.

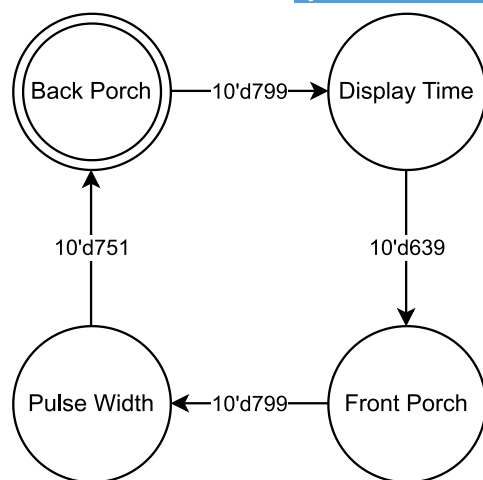


Μέρος Β – Υλοποίηση HSYNC και οριζοντίου μετρητή pixel

Για την δημιουργία του οριζόντιου συγχρονισμού *HSYNC* διαμορφώνουμε μία **μηχανή καταστάσεων** για τις διάφορες καταστάσεις, όπως τα *porch* που υπάρχουν πριν και μετά τον χρόνο προβολής των *pixel*.

Πιο συγκεκριμένα, δημιουργούμε ένα παλμό **pixel_clk**, με έχει συχνότητα 25 MHz, ο οποίος “προχωράει” τον μετρητή καταστάσεων που έχουμε.

State	Horiz. Sync	
	Time	pixel_clk
display time hsync	25.6 μ s	640
back porch hsync	1.92 μ s	48
front porch hsync	640 ns	16
pulse width	3,84 μ s	96



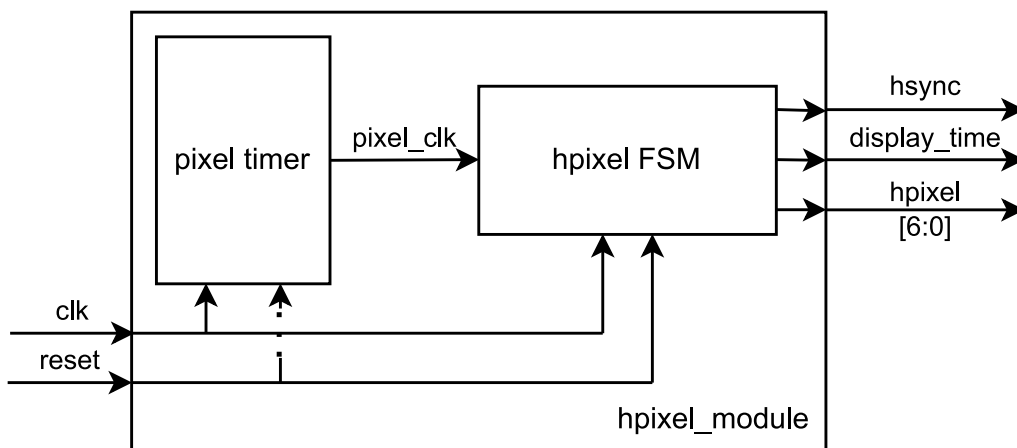
Επιπλέον, στην κατάσταση του **display time** χρησιμοποιούμε έναν νέο μετρητή για τα *pixel* μέσω του οποίου γίνεται το **upscaling** από την ανάλυση της εικόνας μας στην ανάλυση του VGA που είναι πέντε φορές μεγαλύτερη. Αυτό γίνεται μετρώντας **πέντε παλμούς** του *pixel_clk* για να προχωρήσουμε στην επόμενη διεύθυνση της VRAM.

Ο λόγος που επιλέγεται μια τέτοια υλοποίηση είναι καθαρά για την **ευκολία του σχεδιαστή** και όχι για λόγους βελτιστοποίησης που κυκλώματος.

Σημείωση: το σήμα **pixel_clk** σε καμία περίπτωση δεν μπορεί να χρησιμοποιηθεί ως **αντικατάστατο** του πραγματικού ρολογιού, *clk*, που δημιουργείται από τον ταλαντωτή της FPGA.

Μονάδα *hpixel_module*

Το *top-module* για αυτό το μέρος είναι το **hpixel_module**. Αυτό εσωτερικά περιέχει ένα χρονόμετρο (*pixel timer*) που αντιστοιχεί στον χρόνο προβολής του κάθε *pixel* για την ανάλυση 640x480 και έχει ως έξοδο το *pixel_clk*. Ο παλμός *pixel_clk* εισάγεται στην FSM του HSYNC (*hsync FSM*) και αυτή μας παράγει τις απαραίτητες εξόδους.

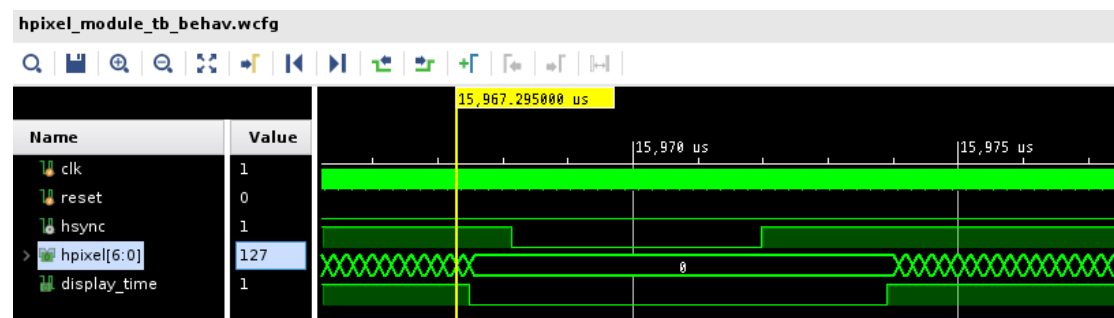


Σημείωση: στο σχεδιάγραμμα της μηχανής καταστάσεων η συνθήκη αλλαγής κατάστασης είναι η τιμή του μετρητή καταστάσεων,

π.χ. στο βέλος θα έπρεπε να γράφει `counter == 10'd799`

Πλαίσιο δοκιμής

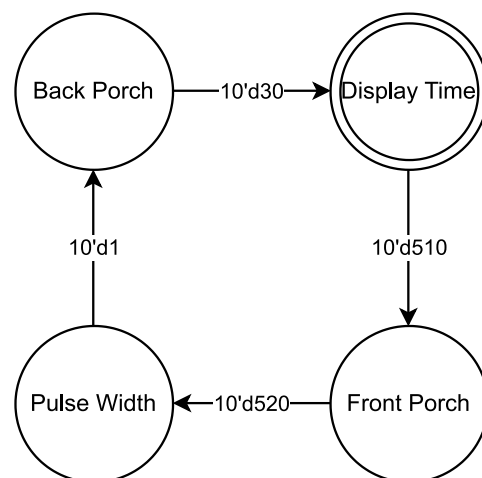
Όπως παρατηρούμε από το στιγμιότυπο στο τέλος της κάθε σειράς το *hpixel* έχει ορθώς μετρήσει μέχρι το 127 έχοντας έτσι **διατρέξει** και τα **128 pixel** της εικόνας μας. Επίσης βλέπουμε ότι τότε πέφτει το σήμα *display_time*, ρίχνοντας τα χρώματα του VGA στην **γείωση**, και λίγο αργότερα ότι έχουμε τον παλμό στην *hsync*. Τέλος τα χρονικά διαστήματα από την πτώση του **display_time** μέχρι την πτώση του *hsync* αποτελούν τα **front** και **back porches**.

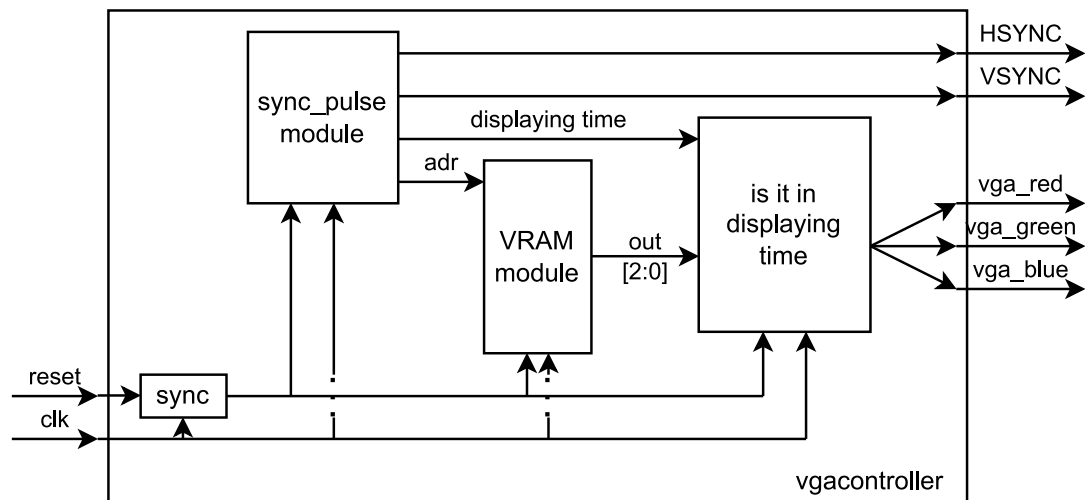
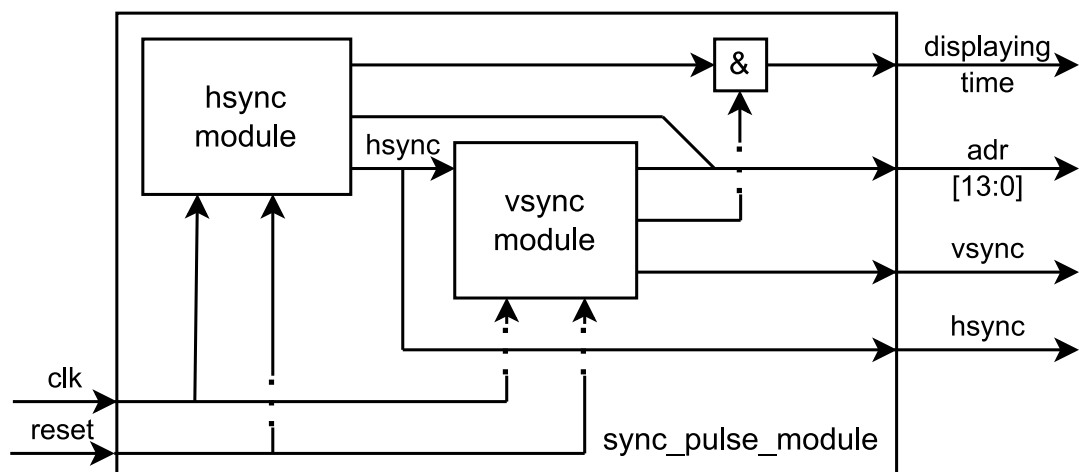


Μέρος Γ – Υλοποίηση VSYNC και κατακόρυφου μετρητή pixel

Ο κατακόρυφος συγχρονισμός *VSYNC* λειτουργεί με παρόμοια λογική με εκείνη του *HSYNC*. Οι βασικές διαφορές είναι ότι αντί για *pixel_clk* **μετράμε** σε **παλμούς HSYNC**, έτσι έχουμε διαφορετικές τιμές στον **μετρητή καταστάσεων** εφόσον έχουμε διαφορετικούς χρόνους στην **μηχανική καταστάσεων**.

State	Vertical Sync	
	Time	HSYNC pulses
display time vsync	15.36 ms	480
back porch vsync	928 μs	29
front porch vsync	320 μs	10
pulse width	64 μs	2



Μονάδα *vgacontroller*Μονάδα *sync_pulse_module*

Πλαίσιο δοκιμής

Εδώ πέρα βλέπουμε ένα *screenshot* κατά την διάρκεια της προβολής της οθόνης. Πιο συγκεκριμένα βλέπουμε 3 λωρίδες διαφορετικών χρωμάτων σε λευκό *background*. Επίσης βλέπουμε ανάμεσα από γραμμές να **γυιώνονται** τα σήματα του VGA, όπως θα περιμέναμε.

