

Η τρίτη εργαστηριακή άσκηση υλοποιεί έναν Ελεγκτή/Οδηγού θύρας οθόνης VGA (Video Graphics Array), με σκοπό να επιτευχθεί η οδήγηση μιας συμβατικής οθόνης και η εμφάνιση της εικόνας σε αυτήν. Τμήμα της εσωτερικής μνήμης RAM της FPGA ,θα πρέπει να οριστεί σαν μνήμη εικόνας - Video RAM (VRAM) του οδηγού, έτσι ώστε ο οδηγός να απεικονίζει την εικόνα , με κατάλληλη οδήγηση της VGA.

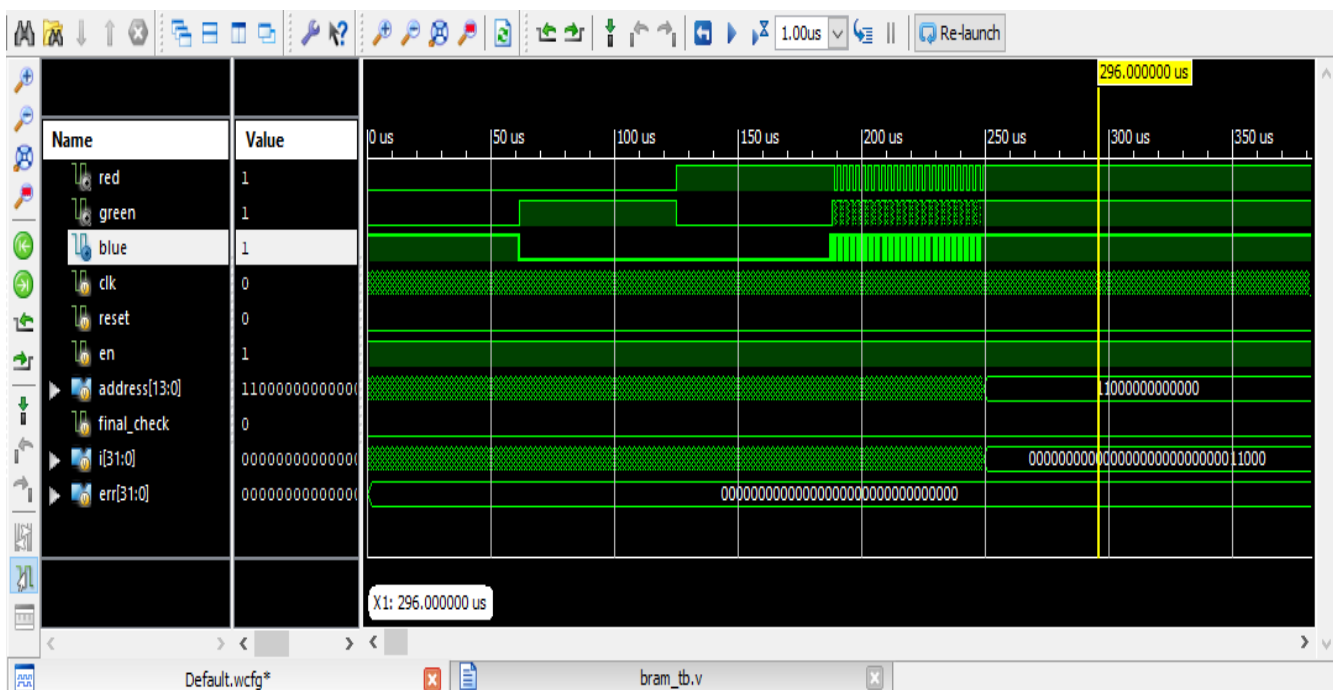
### partA

Στο πρώτο μέρος της τρίτης εργαστηριακής άσκησης υλοποιείται μια block RAM σύμφωνα με τα πρότυπα ISE για την Spartan3E , η οποία θα αξιοποιηθεί ως Video RAM. Η ανάλυση της εικόνας δίδεται 128x96 δηλαδή 12.288bits. Επιπλέον η εικόνα χωρίστηκε σε τέσσερα τμήματα , με κάθε τμήμα να αποτελείται από 3072bits (δηλαδή  $96/4=24$  σειρές από 128bits η κάθε σειρά) για να ελεγχθεί κατάλληλα η λειτουργία του Ελεγκτή θύρας οθόνης VGA .Τα τέσσερα (οριζόντια) τμήματα είναι:

1. Το πρώτο τμήμα , δηλαδή οι διευθύνσεις 0-3071 απεικονίζουν ένα κόκκινο πλαίσιο.
2. Το δεύτερο τμήμα , δηλαδή οι διευθύνσεις 3072-6143 απεικονίζουν ένα μπλε πλαίσιο.
3. Το τρίτο τμήμα , δηλαδή οι διευθύνσεις 6144-9215 απεικονίζουν ένα πράσινο πλαίσιο.
4. Το τέταρτο τμήμα , δηλαδή οι διευθύνσεις 9216-12287 απεικονίζουν ένα κάρε πολλαπλών χρωμάτων (μωβ , άσπρο , κυανό , μαύρο και κίτρινο ).

### ΕΠΑΛΗΘΕΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ

Η ορθότητα υλοποίησης στο βήμα αυτό υλοποιείται μέσω αυτόματου testbench το οποίο ελέγχει τις τιμές των bits για κάθε διεύθυνση μνήμης που έχει αρχικοποιηθεί.



## Part B

Στο δεύτερο μέρος της εργασίας υλοποιήθηκε μια συνδεσμολογία vga η οποία περιλαμβάνει μόνο τον οριζόντιο συγχρονισμό της εικόνας. Αποτελείται από ένα top module vga το οποίο περιλαμβάνει εσωτερικά δύο άλλα module τα hsync\_generator και pixel\_controller.

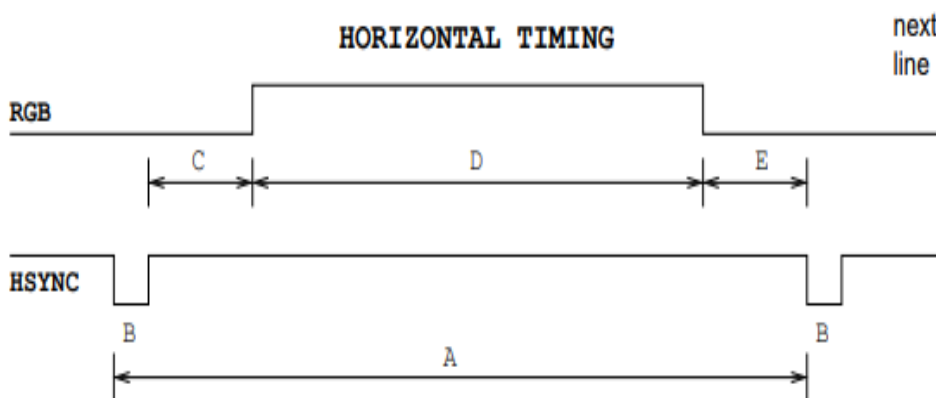
### Hsync\_generator

Λειτουργία: Αυτό το module χρησιμοποιεί έναν counter(interval\_counter) ο οποίος χρησιμοποιείται για να παράξει τα κατάλληλα σήματα hsync και display τα οποία θα χρησιμοποιηθούν σε επόμενο στάδιο για τον συγχρονισμό της εικόνας εξωτερικά (στην οδήγηση των pixel της νέας εικόνας) και εσωτερικά (στην προσπέλαση της μνήμης) , αντίστοιχα.

Υλοποίηση:Κυκλωματικά η υλοποίηση αυτού του module χρησιμοποιεί 4 always block

- Το πρώτο always block ενεργοποιείται σε κάθε θετική ακμή του clk και διαχειρίζεται την τιμή του interval\_counter.
- Το δεύτερο always block αναθέτει την τιμή του new\_display στο display εκτός αν το σύστημα μας λαμβάνει high reset.
- Το τρίτο always block αναθέτει την τιμή του new\_hsync στο hsync εκτός και αν το σύστημα μας λαμβάνει high reset.Το δεύτερο και το τρίτο always block αποσκοπούν στην διατήρηση της τρέχουσας τιμής των σημάτων και στην ανάθεση της νέας τιμής τους στον επόμενο κύκλο ρολογιού.
- Το τέταρτο always block ενεργοποιείται όπως και τα προηγούμενα σε κάθε θετική ακμή του clk και διαχειρίζεται τις τιμές των σημάτων new\_display και new\_hsync ανάλογα με την τιμή του interval\_counter.

interval	max counter value	new_hsync	new_display
initial	0	0	0
B(hsync_pulse)	192	1	0
B+C	288	1	1
B+C+D	1568	1	0
B+C+D+E	1599	0	0



### Pixel\_controller

Λειτουργία: Αυτό το module διαχειρίζεται την προσπέλαση της οριζόντιας και θέσης των pixel της bram σύμφωνα με το σήμα display.Στην περίπτωση που βρισκόμαστε στο διάστημα D (δηλαδή display=1) η εικόνα είναι ενεργή ,

εσωτερικά στο module προσπελαύνονται οριζόντια τα στοιχεία της μνήμης. Η ορθή λειτουργία του module εξάγει τιμές για την πληροφορία RGB των pixel.

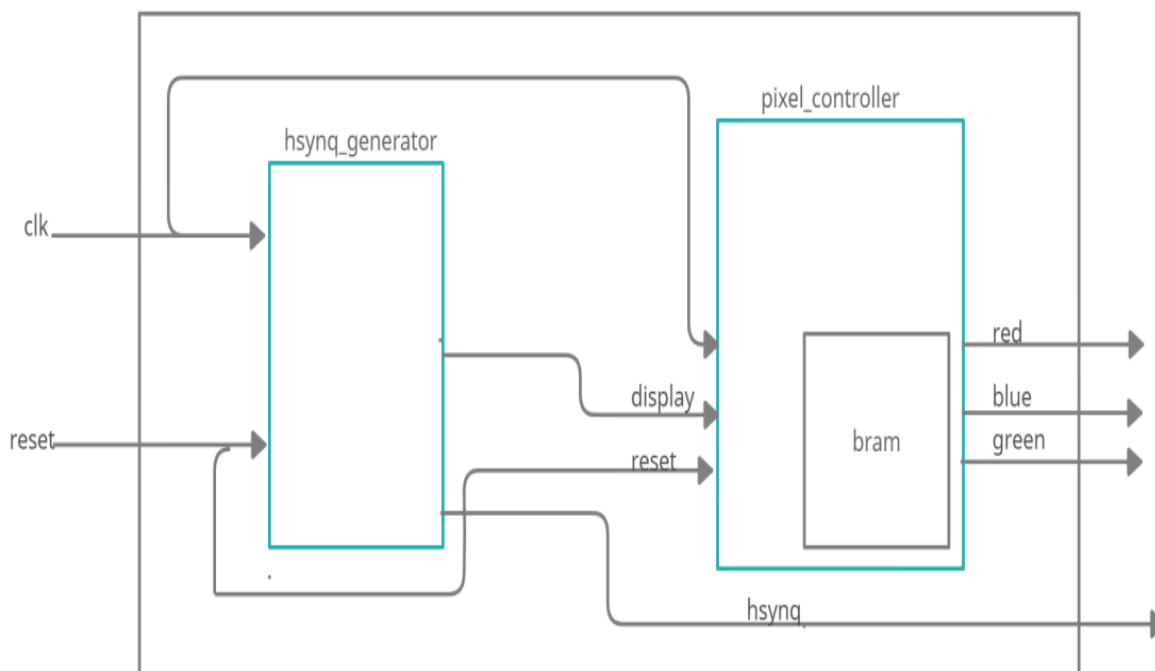
Υλοποίηση:Κυκλωματικά η υλοποίηση αυτού του module χρησιμοποιεί 2 always block

- Το πρώτο always block ενεργοποιείται σε κάθε θετική ακμή του clk και διαχειρίζεται την τιμή του HPIXEL. Στο block αυτό χρησιμοποιείται και ένας επιπλέον counter h\_anal\_counter του οποίου βασικός στόχος είναι η διατήρηση της διεύθυνσης μνήμης σταθερή για 10 διαφορετικά pixel , έτσι ώστε να επιτευχθεί ανάλυση 640x480(αντί της ανάλυσης 128x96 που έχει αποθηκευτεί στην bram).  
\*\*Η τιμή του HPIXEL παραμένει σταθερή για 10 επαναλήψεις του counter (αντί των 5 επαναλήψεων που απαιτείται για να επιτευχθεί οριζόντια ανάλυση 640 bit per line δεδομένης μνήμης 128 bit per line).Αυτή η πρακτική χρησιμοποιείται για να καλυφθεί όλος ο διαθέσιμος χρόνος του hsync και να προβληθεί τελικά μία εικόνα ανάλογη αυτής που αρχικοποιήθηκε στην μνήμη.

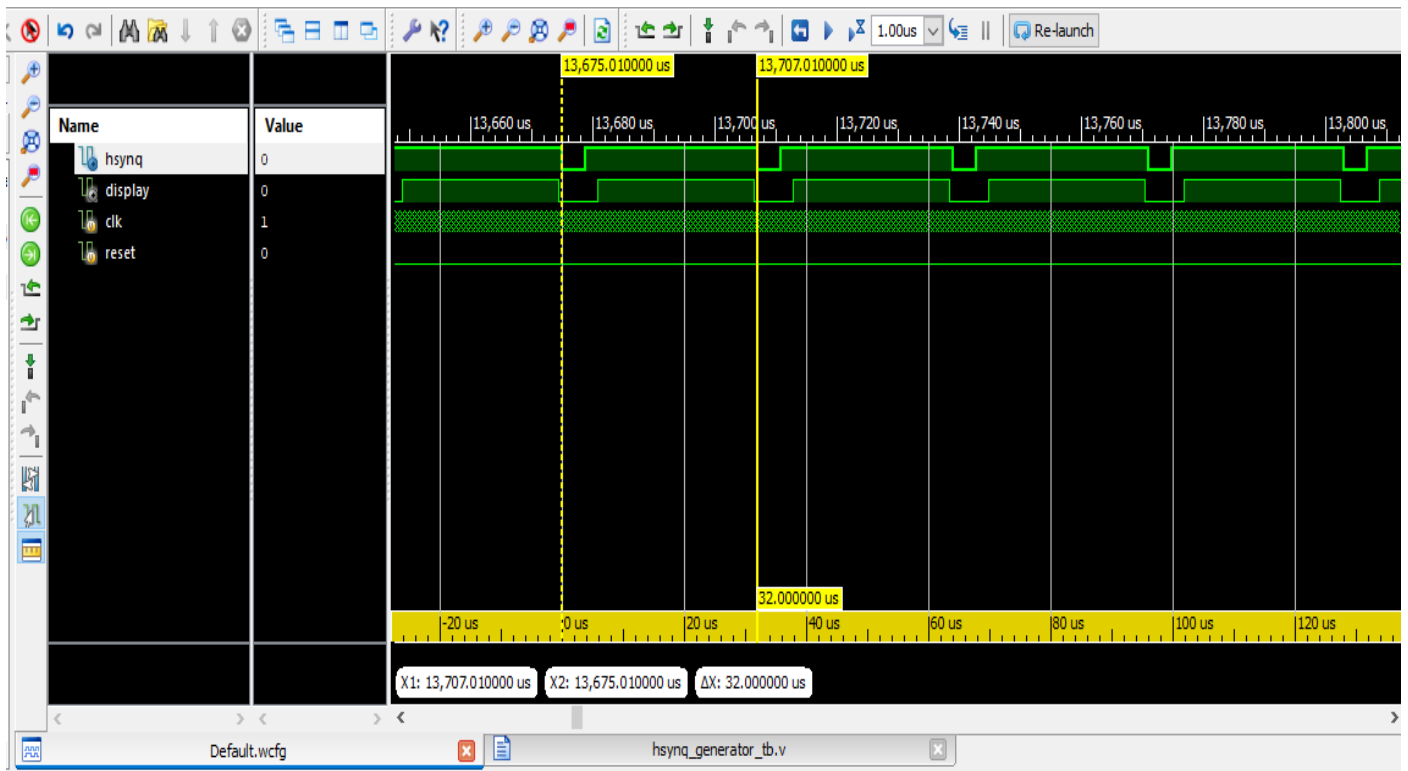
Επιπλέον χρησιμοποιούνται τέσσερα assign verilog instructions

- Το πρώτο εξασφαλίζει την ανάθεση ορθής διεύθυνσης μνήμης που θα χρησιμοποιηθεί σαν είσοδος στο bram instance .
- Το δεύτερο , το τρίτο και το τέταρτο assign αναθέτουν τις τιμές το RED , GREEN , BLUE σύμφωνα με την έξοδο του bram instance , εφόσον η εικόνα είναι ενεργή(display=1).

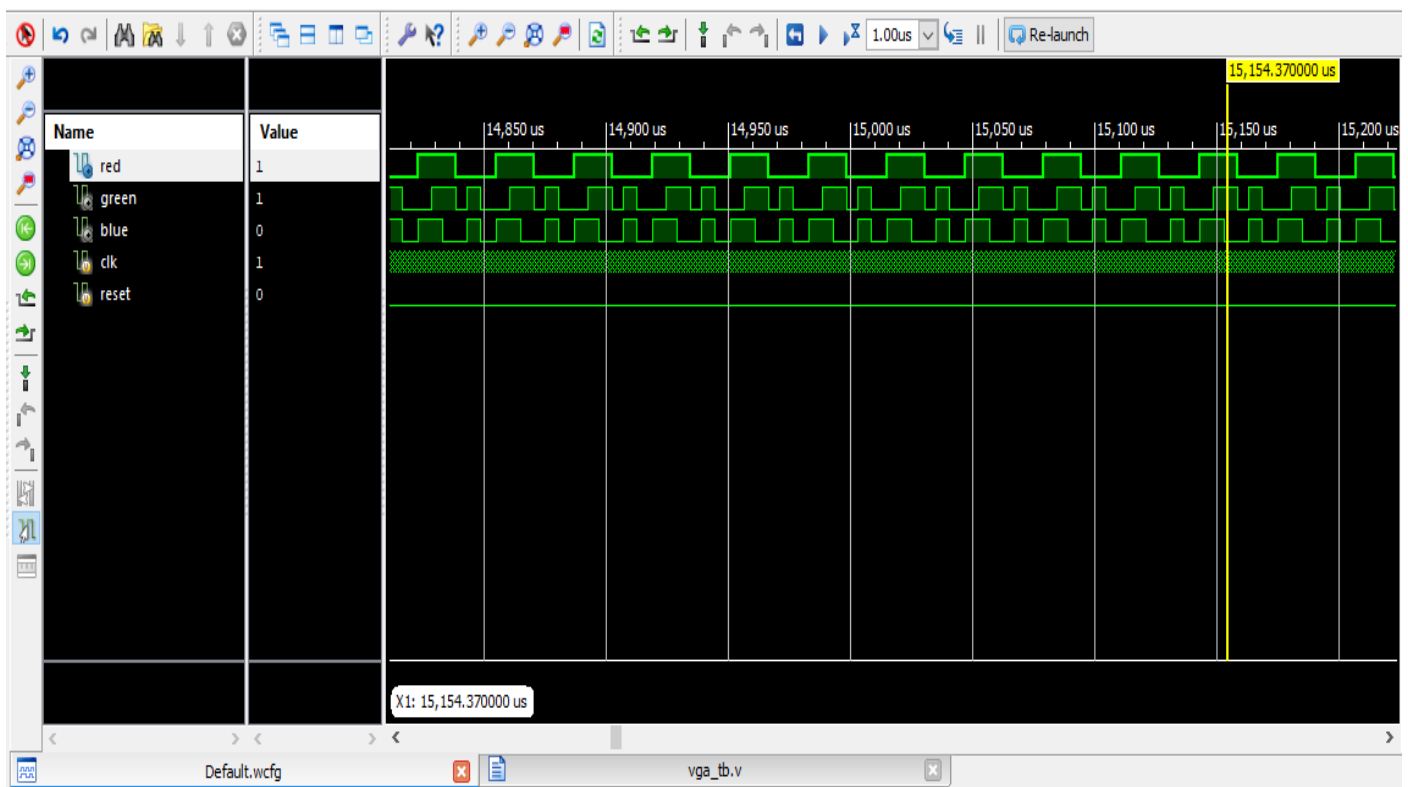
Το vga top module περιέχει instances των δύο παραπάνω module. Σχηματικά παρουσιάζεται παρακάτω:



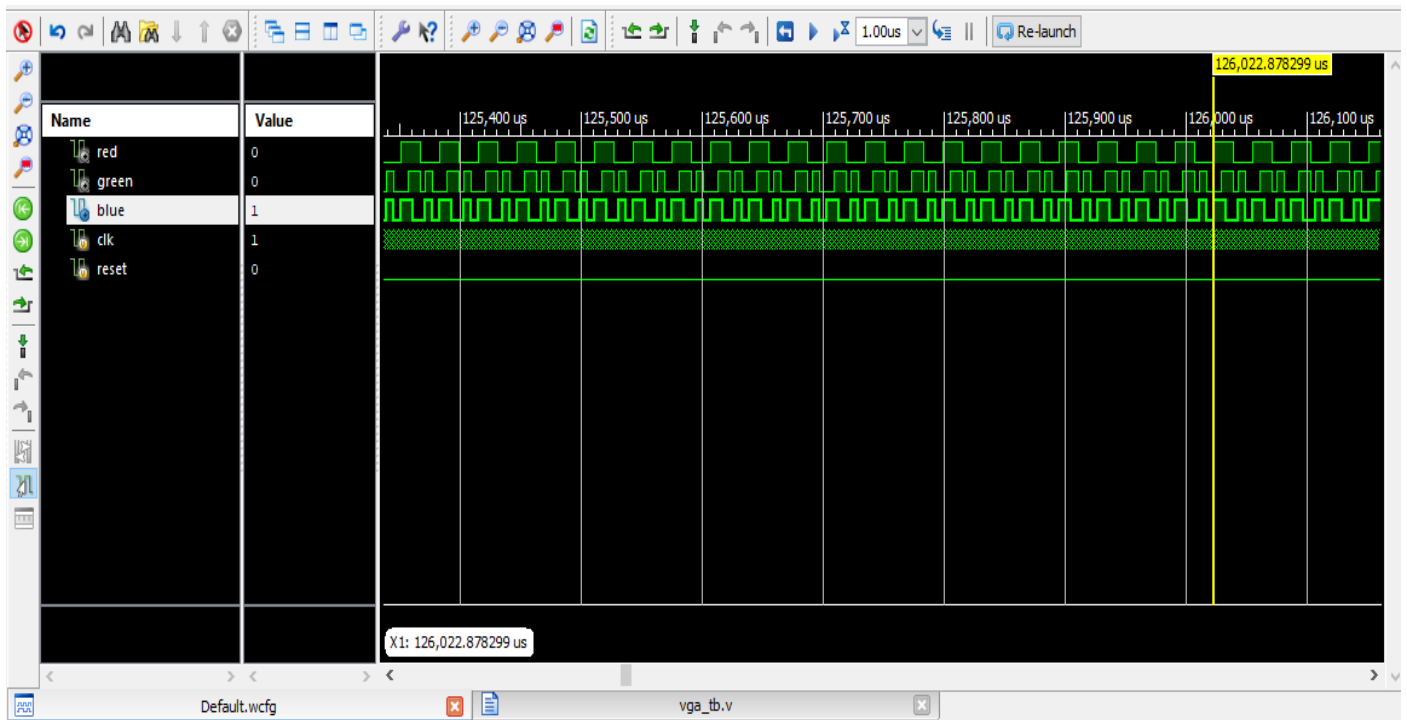
ΕΠΑΛΗΘΕΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ



Behavioral Simulation hsync\_generator(partB)



Behavioral Simulation vga(hsync\_generator and pixel\_controller)



Post and Route Simulation vga(hsynq\_generator and pixel\_controller)

### Part C

Στο τρίτο μέρος της εργασίας υλοποιήθηκε η τελική μορφή του vga που περιλαμβάνει τον οριζόντιο(μέρος Β) και τον κάθετο συγχρονισμό της εικόνα. Αποτελείται από ένα top module vga το οποίο περιλαμβάνει εσωτερικά τρία άλλα module τα hsynq\_generator, vsynq\_generator και pixel\_controller.

**Hsynq\_generator** (Όπως υλοποιήθηκε στο μέρος Β)

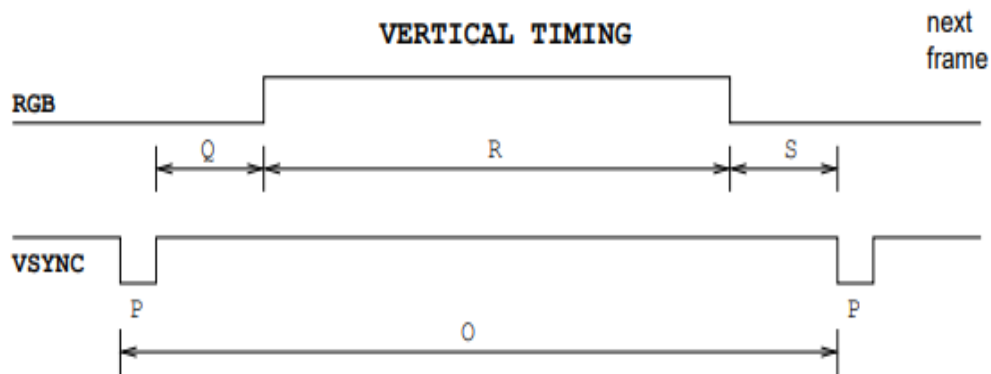
#### Vsynq\_generator

Λειτουργία: Αυτό το module χρησιμοποιεί έναν counter(interval\_counter) ο οποίος χρησιμοποιείται για να δημιουργήσει τα κατάλληλα σήματα vsynq και vdisplay τα οποία θα χρησιμοποιηθούν σε επόμενο στάδιο για τον συγχρονισμό της εικόνας εξωτερικά στην οδήγηση των pixel της νέας εικόνας και εσωτερικά στην προσπέλαση της μνήμης , αντίστοιχα.

Υλοποίηση:Κυκλωματικά η υλοποίηση αυτού του module χρησιμοποιεί 2 always block

- Το πρώτο always block ενεργοποιείται σε κάθε θετική ακμή του clk και διαχειρίζεται την τιμή του interval\_counter.
- Το δεύτερο always block ενεργοποιείται όπως και το προηγούμενο σε κάθε θετική ακμή του clk και διαχειρίζεται τις τιμές των σημάτων vdisplay και vsynq ανάλογα με την τιμή του interval\_counter.

interval	max counter value	vsynq	vdisplay
initial	0	0	0
P(vsynq_pulse)	3200	1	0
P+Q	49600	1	1
P+Q+R	817600	1	0
P+Q+R+S	833599	0	0



### Pixel\_controller

Λειτουργία: Αυτό το module διαχειρίζεται τον συγχρονισμό της οριζόντιας και της κάθετης θέσης των pixel της bram σύμφωνα με τα σήματα display και vdisplay. Στην περίπτωση που βρισκόμαστε σε διάστημα όπου display=1 και vdisplay=1 η εικόνα είναι ενεργή και εσωτερικά στο module προσπελούνται διαδοχικά τα στοιχεία της μνήμης. Η ορθή λειτουργία του module εξάγει τιμές για την πληροφορία RGB των pixel.

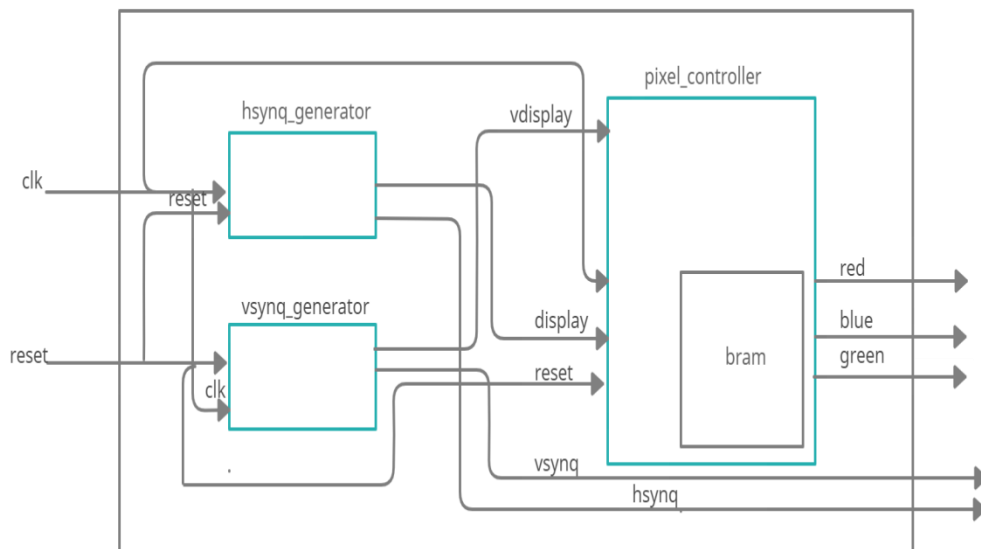
Υλοποίηση: Κυκλωματικά η υλοποίηση αυτού του module χρησιμοποιεί 1 always block

- Το πρώτο always block ενεργοποιείται σε κάθε θετική ακμή του clk και διαχειρίζεται την τιμή του HPIXEL. Στο block αυτό χρησιμοποιείται και ένας επιπλέον counter h\_anal\_counter του οποίου βασικός στόχος είναι η διατήρηση της διεύθυνσης μνήμης σταθερή για 10 διαφορετικά pixel, έτσι ώστε να επιτευχθεί ανάλυση 640x480 (αντί της ανάλυσης 128x96 που έχει αποθηκευτεί στην bram). Συγκριτικά με το μέρος B έχει προστεθεί ένα επιπλέον σήμα finished\_line το οποίο ειδοποιεί το σύστημα ότι έχει προσπελαθεί μία ολόκληρη οριζόντια γραμμή.
- Το δεύτερο always block ενεργοποιείται επίσης σε κάθε θετική ακμή του clk και διαχειρίζεται την τιμή του VPIXEL. Στο block αυτό χρησιμοποιείται και ένας επιπλέον counter v\_anal\_counter του οποίου βασικός στόχος είναι η διατήρηση της διεύθυνσης μνήμης σταθερή για 5 διαφορετικές σειρές pixel, έτσι ώστε να επιτευχθεί ανάλυση 640x480 (αντί της ανάλυσης 128x96 που έχει αποθηκευτεί στην bram). Στην περίπτωση που η εικόνα είναι ενεργή κάθετα (vdisplay=1) και έχει συμπληρωθεί η προσπέλαση μίας ολόκληρης γραμμής (finished\_line = 1) πραγματοποιείται έλεγχος για τον αν πρέπει ή όχι να μεταβληθεί η τιμή του VPIXEL.

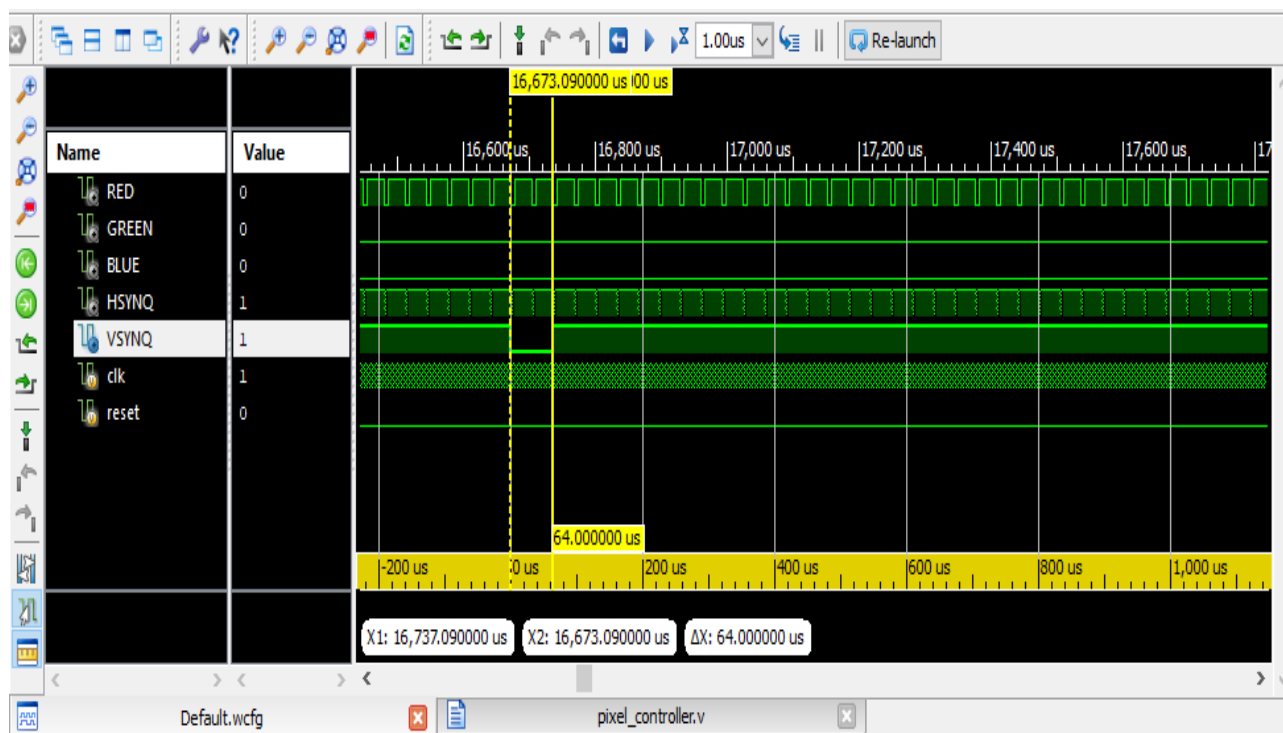
Επιπλέον χρησιμοποιούνται τέσσερα assign verilog instructions

- Το πρώτο εξασφαλίζει την ανάθεση ορθής διεύθυνσης μνήμης που θα χρησιμοποιηθεί σαν είσοδος στο bram instance.
- Το δεύτερο, το τρίτο και το τέταρτο assign αναθέτουν τις τιμές το RED, GREEN, BLUE σύμφωνα με την έξοδο του bram instance, εφόσον η εικόνα είναι ενεργή.

Το vga\_top module περιέχει instances των δύο παραπάνω module. Σχηματικά παρουσιάζεται παρακάτω:

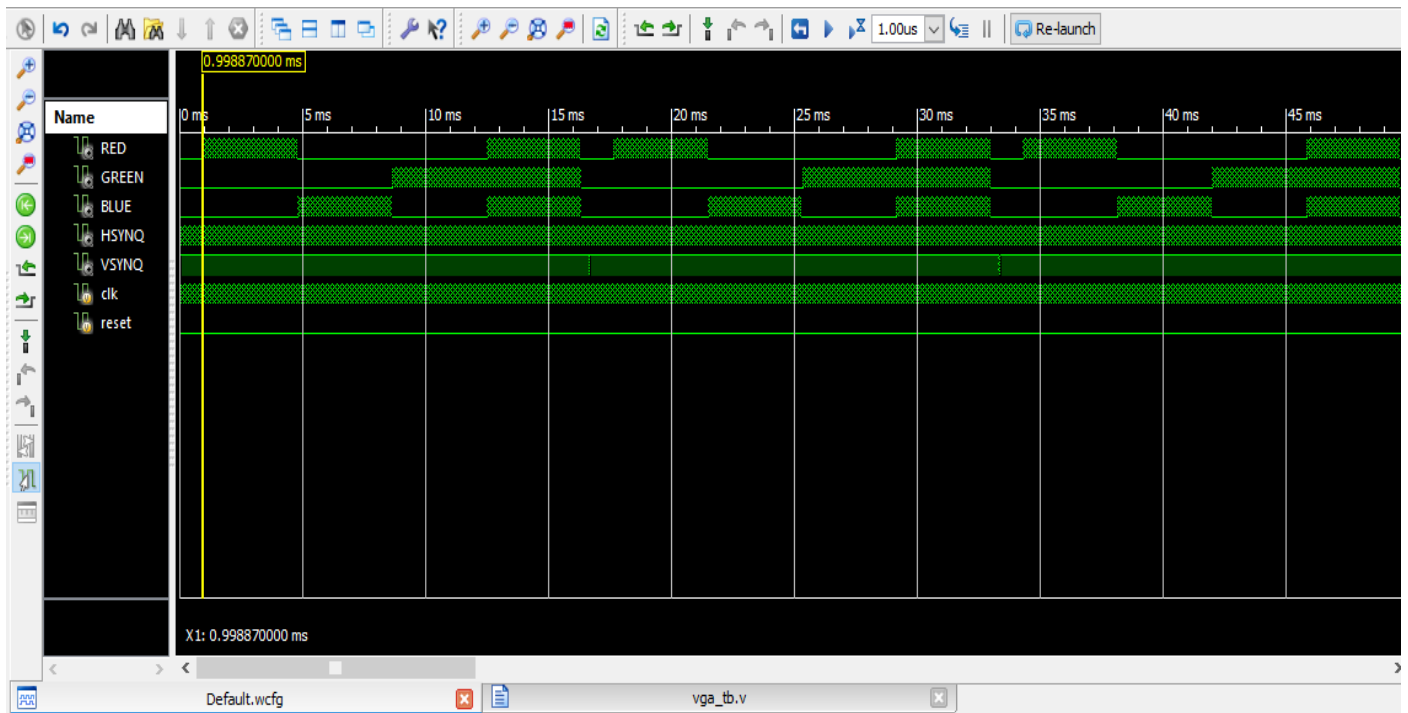


## ΕΠΑΛΗΘΕΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ

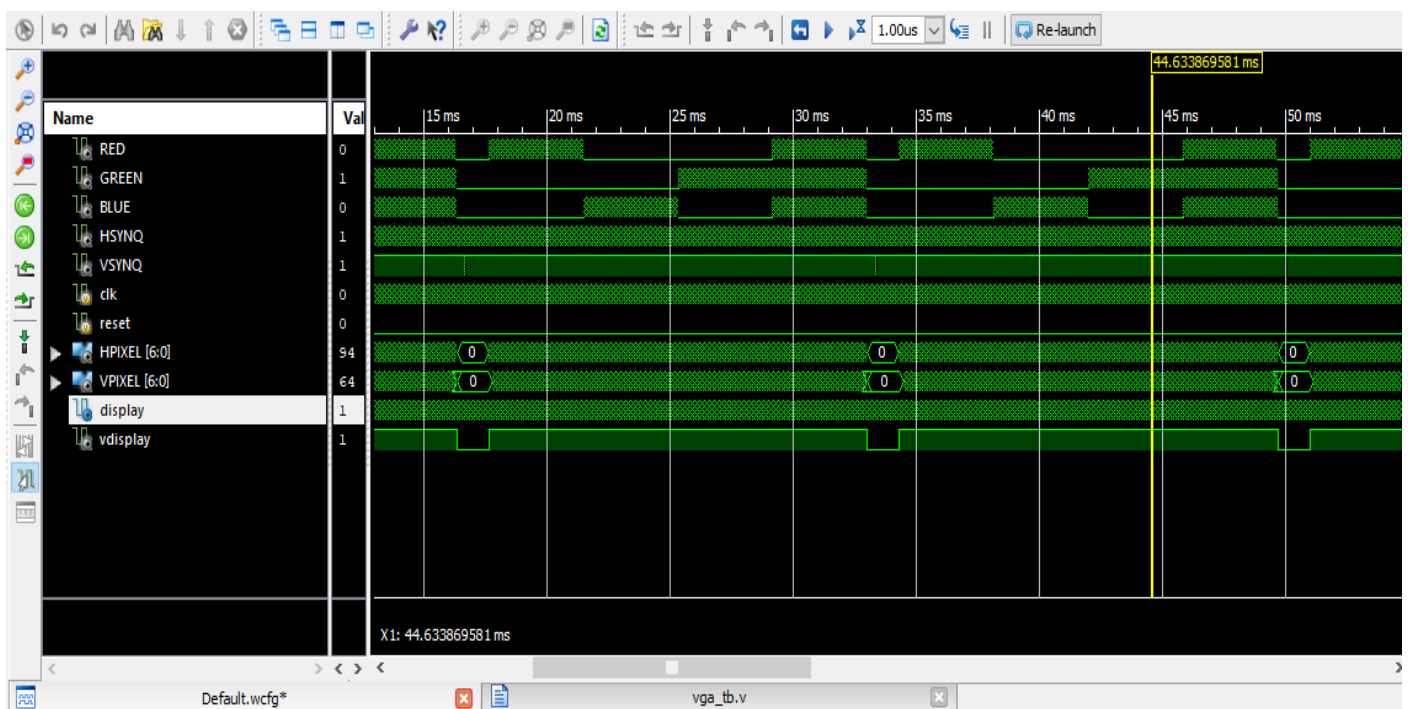


Verification of vsynq\_pulse time equal to 64μs

Post and Route Simulation vsynq\_generator(partC)



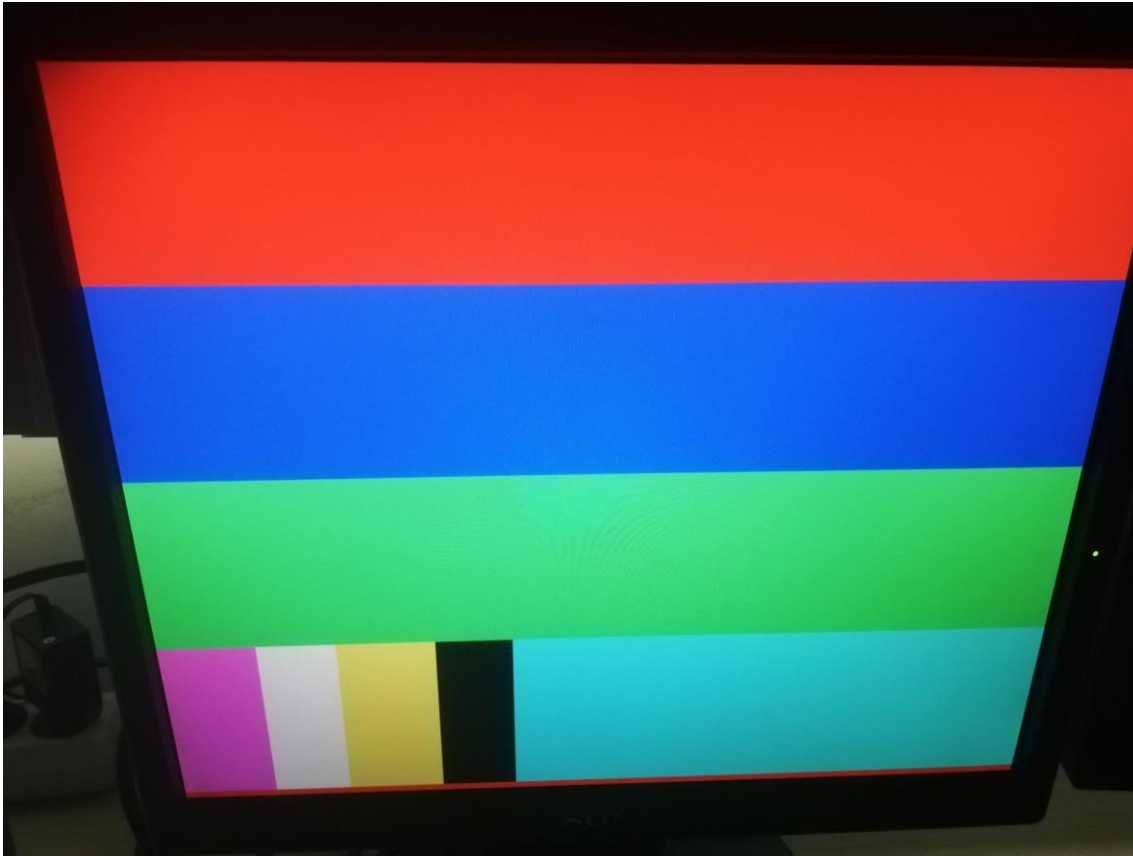
Behavioral Simulation vga (partC)



## Παρατηρήσεις

Την πρώτη φορά που ο κώδικας ελέγχθηκε στην fpga το αποτέλεσμα ήταν το παρακάτω





Παρατηρήθηκε λοιπόν

1. Αστοχία να προσπελαθούν σωστά τα pixel οριζόντια γιατί στην αρχική μορφή του ο κώδικας διατηρούσε σταθερή την τιμή του HPIXEL για 5 επαναλήψεις και έπειτα διατηρούσε σταθερή την τελική τιμή του HPIXEL. Έτσι δικαιολογείται και η γαλάζια γραμμή στο τέλος της εικόνας που υπερβαίνει σε μέγεθος τα υπόλοιπα 4 χρώματα που εμφανίζονται στην multicolor area.  
**Επίλυση:** Στην φάση αυτή το πρόβλημα ξεπεράστηκε σχετικά εύκολα με την αλλαγή της μέγιστης τιμής του `h_anal_counter` που θα απαιτούταν για να μεταβληθεί η τιμή του HPIXEL. Οι 5 επαναλήψεις του καθένα HPIXEL είχαν διάρκεια περίπου  $hsynq/2$ , έτσι διπλασιάζοντας την μέγιστη τιμή του `h_anal_counter`, από 5 σε 10 επαναλήψεις [0-9] του κάθε pixel, καλύφθηκε ο συνολικός χρόνος του `hsynq`.
2. Επιπλέον παρατηρείται μία κόκκινη γραμμή στο τέλος της οθόνης. Μετά από ελέγχους διαπιστώθηκε ότι η κόκκινη γραμμή προέρχεται από την πρώτη γραμμή στην `bram`. Το πρόβλημα οφειλόταν στο γεγονός ότι κατά την διάρκεια του `vsynq pulse` μεταβαλλόταν η τιμή του HPIXEL ενώ η εικόνα, πρακτικά σε εκείνο το διάστημα η εικόνα δεν είναι ενεργή.  
**Επίλυση:** Για να επιλυθεί αυτό το πρόβλημα προστέθηκε ένας ακόμα όρος στον έλεγχο για την τιμή του HPIXEL. Αρχικά για να αυξηθεί η τιμή του HPIXEL απαιτούταν έλεγχος μόνο του `display` ενώ στην τελική μορφή του `rtl` κώδικα στον ίδιο έλεγχο προστέθηκε και ο όρος `vdisplay` προκειμένου να αλλάξει η οριζόντια θέση του pixel.

Η τελική μορφή είναι η παρακάτω:

