

## HY220: Εργαστήριο Ψηφιακών Κυκλωμάτων

Τμήμα Επιστήμης Υπολογιστών  
Πανεπιστήμιο Κρήτης  
Εαρινό Εξάμηνο 2022

### Εργαστήριο 1 (2 εβδομάδες)

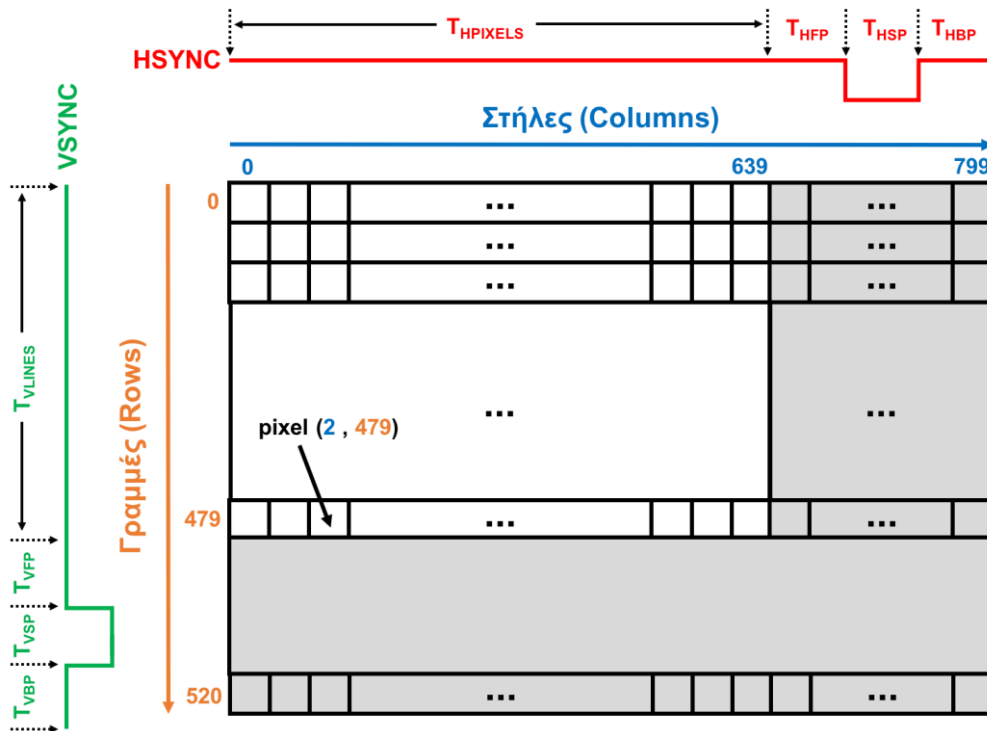
- **Εβδομάδα 04/04 έως 08/04** (αναλόγως το εργαστηριακό τμήμα που έχετε δηλώσει)
  - (A) Τρίτη 05/04 12:00 – 14:00 στην αίθουσα B.110
  - (B) Τρίτη 05/04 18:00 – 20:00 στην αίθουσα B.110
  - (Γ) Πέμπτη 07/04 12:00 – 14:00 στην αίθουσα B.110
  - (Δ) Πέμπτη 07/04 18:00 – 20:00 στην αίθουσα B.110
- **Εβδομάδα 11/04 έως 15/04** (αναλόγως το εργαστηριακό τμήμα που έχετε δηλώσει)
  - (A) Τρίτη 12/04 12:00 – 14:00 στην αίθουσα B.110
  - (B) Τρίτη 12/04 18:00 – 20:00 στην αίθουσα B.110
  - (Γ) Πέμπτη 14/04 12:00 – 14:00 στην αίθουσα B.110
  - (Δ) Πέμπτη 14/04 18:00 – 20:00 στην αίθουσα B.110

Κατά τη διάρκεια των εργαστηρίων θα υλοποιήσετε σε τρεις φάσεις το παιχνίδι «Λαβύρινθος» (Maze). Περιληπτικά, το τελικό παιχνίδι θα εμφανίζει σε VGA οθόνη ένα λαβύρινθο και τη φιγούρα ενός παίκτη. Ο χρήστης θα μπορεί να μετακινήσει τον παίκτη με τα κουμπιά που υπάρχουν πάνω στην πλακέτα και ο σκοπός είναι να τον οδηγήσει στην έξοδο το ταχύτερο δυνατόν.

Στο *Εργαστήριο 1* θα υλοποιήσετε την πρώτη φάση (σε 2 εβδομάδες εργαστηρίων) που περιλαμβάνει την οδήγηση μιας οθόνης VGA για την εμφάνιση οριζόντιων γραμμών με διαφορετικά χρώματα. Για το εργαστήριο αυτό θα σας δοθεί ο σκελετός αρκετών κομματιών σε SystemVerilog, ένα κατάλληλο testbench, reference outputs και ένας προσομοιωτής VGA (VGA Simulator) που σας δίνει τη δυνατότητα να βλέπετε τι θα εμφανίζονταν σε μια πραγματική οθόνη από τον κώδικά σας.

### **Χρονισμός VGA 640 x 480 @ 60Hz**

Ένα πλήρες frame (εικόνα) αποτελείται από pixels που το καθένα περιέχει τιμές για τα χρώματά του σε RGB (red/green/blue). Το frame αποτελείται από στήλες (columns) και γραμμές (rows). Για κάθε γραμμή πρέπει να δίνονται τα pixels της κάθε στήλης σε διαδοχικούς κύκλους ρολογιού και μόλις ολοκληρωθεί ο απαιτούμενος αριθμός από pixels (columns) τότε πρέπει να σηματοδοτηθεί το τέλος της γραμμής έτσι ώστε να ξεκινήσει η επόμενη γραμμή (N+1) από την αρχή στην στήλη 0. Η σηματοδότηση για το τέλος της γραμμής γίνεται με τον παλμό HSYNC (horizontal synchronization – οριζόντιος συγχρονισμός). Όταν τελειώσουν όλες οι γραμμές τότε πρέπει να σηματοδοτηθεί το τέλος της εικόνας (frame) έτσι ώστε να ξεκινήσει το επόμενο frame από την αρχή στη γραμμή 0 και τη στήλη 0. Η σηματοδότηση για το τέλος του frame γίνεται με τον παλμό VSYNC (vertical synchronization – κατακόρυφος συγχρονισμός). Στην παρακάτω εικόνα φαίνεται το παράδειγμα ενός frame 640 x 480 (δηλαδή 640 στήλες και 480 γραμμές).



Για κάθε γραμμή απαιτείται χρόνος  $T_{\text{HPixels}}$  για τα ενεργά pixels που θέλουμε να εμφανίσουμε και μετά πρέπει να ακολουθήσει ο παλμός HSYNC (με κόκκινο στην εικόνα) ο οποίος είναι active-low για χρονισμό VGA 640 x 480. Για τη δημιουργία του παλμού HSYNC (δηλαδή μετά τα ενεργά pixels  $T_{\text{HPixels}}$ ) απαιτείται κάποιος κενός χρόνος πριν τον παλμό  $T_{\text{HFP}}$  (horizontal front porch), κατά τη διάρκεια του παλμού  $T_{\text{HSP}}$  (horizontal sync pulse) και μετά τον παλμό  $T_{\text{HBP}}$  (horizontal back porch). Σε όλη αυτή τη διάρκεια τα pixels είναι «κενά», δεν λαμβάνονται υπόψιν και δεν εμφανίζονται (με γκρι στο δεξί μέρος της εικόνας).

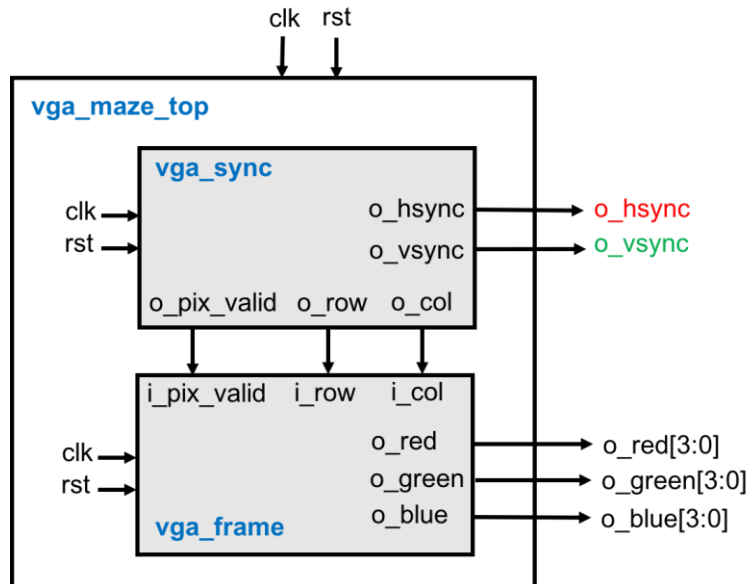
Για κάθε εικόνα (frame) απαιτείται χρόνος  $T_{\text{VLines}}$  για τις ενεργές γραμμές που θέλουμε να εμφανίσουμε και μετά πρέπει να ακολουθήσει ο παλμός VSYNC (με πράσινο στην εικόνα) ο οποίος είναι active-low για χρονισμό VGA 640 x 480. Για τη δημιουργία του παλμού VSYNC (δηλαδή μετά τις ενεργές γραμμές  $T_{\text{VLines}}$ ) απαιτείται κάποιος κενός χρόνος πριν τον παλμό  $T_{\text{VFP}}$  (vertical front porch), κατά τη διάρκεια του παλμού  $T_{\text{VSP}}$  (vertical sync pulse) και μετά τον παλμό  $T_{\text{VBP}}$  (vertical back porch). Σε όλη αυτή τη διάρκεια τα pixels είναι «κενά», δεν λαμβάνονται υπόψιν και δεν εμφανίζονται (με γκρι στο κάτω μέρος της εικόνας).

Για ανάλυση (resolution) 640 x 480 με ρυθμό ανανέωσης 60 Hz (60 frames το δευτερόλεπτο) απαιτείται ρολόι 25MHz (περίοδος ρολογιού 40 ns) και οι απαιτούμενοι χρόνοι φαίνονται στους πίνακες παρακάτω:

<b>Horizontal</b>	Clock Cycles	Time (ns)
$T_{\text{HPixels}}$	640	25600
$T_{\text{HFP}}$	16	640
$T_{\text{HSP}}$	96	3840
$T_{\text{HBP}}$	48	1920
<b>Total</b>	800	32000

Vertical	Rows	Clock Cycles	Time (us)
T <sub>V</sub> LINES	480	384000	15360
T <sub>V</sub> F <sub>FP</sub>	10	8000	320
T <sub>V</sub> S <sub>P</sub>	2	1600	64
T <sub>V</sub> B <sub>P</sub>	29	23200	928
<b>Total</b>	521	416800	16672

Για το Εργαστήριο 1 σας δίνεται το σχέδιο που φαίνεται στην παρακάτω εικόνα:



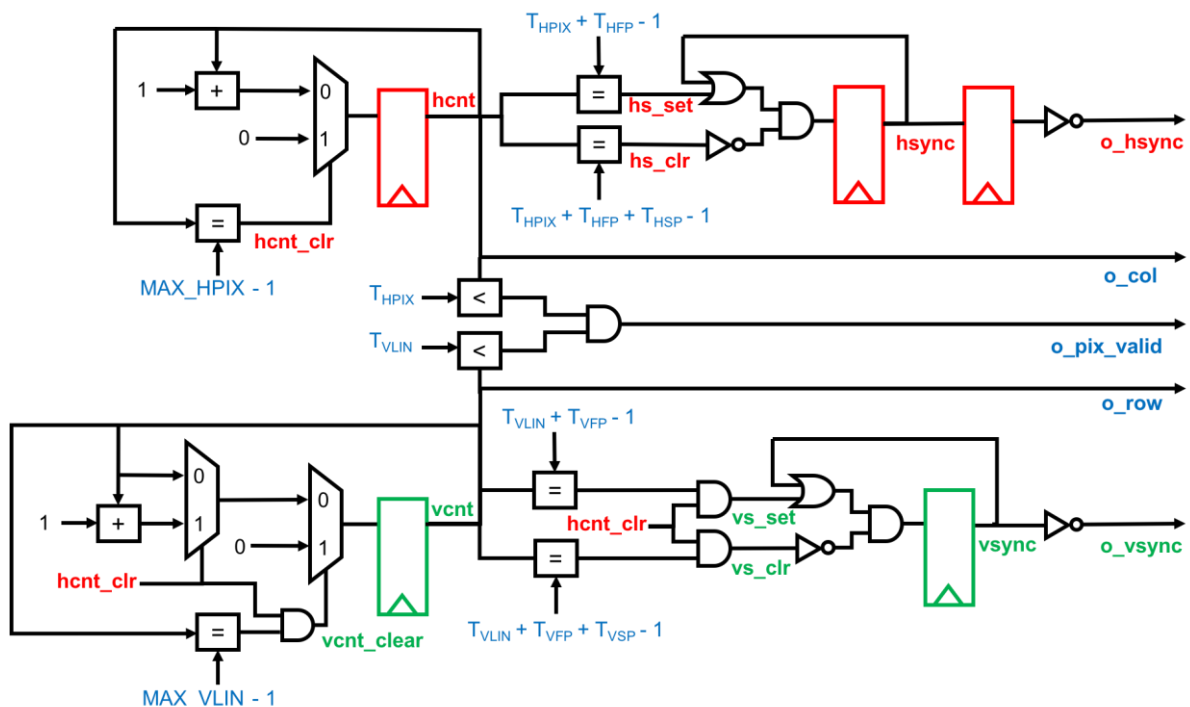
Το σχέδιο έχει την εξής ιεραρχία:

- **vga\_maze\_top** (*vga\_maze\_top.sv*): περιέχει instances από 2 μπλοκ, το **vga\_sync** και το **vga\_frame** (εξηγούνται παρακάτω), τα οποία συνδέονται όπως φαίνεται στο σχήμα.
- **vga\_sync** (*vga\_sync.sv*): Το μπλοκ αυτό είναι υπεύθυνο για την υλοποίηση του χρονισμού του πρωτοκόλλου VGA για ανάλυση 640 x 480 και πρέπει να δημιουργεί 2 βασικά σήματα: (1) το HSYNC (horizontal synchronization – οριζόντιος συγχρονισμός) και (2) το VSYNC (vertical synchronization – κατακόρυφος συγχρονισμός). Επίσης δημιουργεί και 3 σήματα προς το μπλοκ **vga\_frame** που δίνουν τις τρέχουσες συντεταγμένες των pixel που πρέπει να εμφανιστούν στην οθόνη. Το σήμα *o\_col* δίνει τη τρέχουσα τιμή της στήλης (column), το σήμα *o\_row* δίνει την τρέχουσα τιμή της γραμμής (row) και το σήμα *o\_pix\_valid* σηματοδοτεί αν οι τιμές αυτές (*o\_col* και *o\_row*) είναι έγκυρες (valid). Λεπτομέρειες για το τι θα πρέπει να υλοποιήσετε για αυτό το μπλοκ περιγράφονται παρακάτω.
- **vga\_frame** (*vga\_frame.sv*): Το μπλοκ αυτό είναι υπεύθυνο για τη δημιουργία των χρωμάτων RGB (red/green/blue) για κάθε pixel που εμφανίζεται στην οθόνη και σας δίνεται έτοιμο. Το μπλοκ δέχεται σαν είσοδο τις συντεταγμένες του pixel μέσω των εισόδων *i\_col* για τη γραμμή και *i\_row* για τη στήλη και το σήμα *i\_pix\_valid* που δείχνει αν το pixel είναι έγκυρο. Με βάση τις συντεταγμένες παράγει στον «επόμενο κύκλο» τιμές για τα τρία χρώματα έτσι ώστε να εμφανίζονται στην οθόνη εναλλάξ κόκκινες, πράσινες, μπλε και μαύρες μπάρες που έχουν ύψος 16 γραμμές. Στην πλακέτα του εργαστηρίου το κάθε χρώμα είναι 4-bit οπότε μπορούμε να δημιουργήσουμε  $2^4 \times 2^4 \times 2^4 = 4096$  διαφορετικά χρώματα. Η τιμή RGB (15, 0, 0) δημιουργεί κόκκινο χρώμα, η τιμή RGB (0, 15, 0) πράσινο χρώμα, η τιμή RGB (0, 0, 15) μπλε χρώμα και η τιμή RGB (0, 0, 0) μαύρο χρώμα. Η τιμή RGB (15, 15, 15) δημιουργεί άσπρο χρώμα και υπάρχουν όλοι οι υπόλοιποι συνδυασμοί που δημιουργούν αποχρώσεις π.χ. το RGB (2, 4, 1) δημιουργεί μια απόχρωση του πράσινου.

- **vga\_tb** (*vga\_tb.sv*): ένα testbench για προσομοίωση που δημιουργεί το ρολόι (25 MHz – 40 ns) και το reset. Το testbench αποθηκεύει την έξοδο του κυκλώματός σας σε ένα αρχείο (*vga\_log.txt*) με το κατάλληλο format έτσι ώστε να μπορείτε να χρησιμοποιήσετε τον VGA Simulator για να βλέπετε τι θα εμφανίζεται στην οθόνη από τον κώδικά σας. Περισσότερες λεπτομέρειες για τον VGA Simulator παρακάτω.
- **Reference output:** Στο φάκελο reference υπάρχει ένα πρότυπο *vga\_log.txt* output που είναι αυτό που θα πρέπει να δημιουργεί ένας σωστός κώδικας. Μπορείτε να κάνετε diff αυτό που παράγει ο δικός σας κώδικας με το reference output για να εντοπίσετε λάθη κατά την προσομοίωση. Το format είναι συμβατό με τον VGA Simulator και είναι πολύ απλό. Σε κάθε γραμμή περιέχει το χρόνο σε ns ακολουθούμενο από τις τιμές των σημάτων hsync (1-bit), vsync (1-bit), red (4-bits), green (4-bits), blue (4-bits).
- **VGA Simulator:** Μέσα στο φάκελο vga-simulator υπάρχει μια ιστοσελίδα που μπορείτε να ανοίξετε τοπικά στον web browser σας. Εκεί μπορείτε να επιλέξετε το log file που έχει δημιουργηθεί από την προσομοίωσή σας και όταν πατήσετε το κουμπί submit τότε θα εμφανιστεί σε μια εικονική VGA οθόνη η έξοδός σας. Μπορείτε να το δοκιμάσετε επίσης με το reference output. Μην αλλάξετε τις παραμέτρους που υπάρχουν ήδη στη σελίδα! **Credits:** Ο VGA Simulator έχει δημιουργηθεί από τον Eric Eastwood στο παρακάτω website <http://ericeastwood.com/lab/vga-simulator/>

**Τι πρέπει υλοποιήσετε και να προσομοιώσετε πριν πάτε στα εργαστήρια:**

Για το εργαστήριο αυτό θα πρέπει να υλοποιήσετε το μπλοκ **vga\_sync** του οποίου ένα άδειο module σας δίνεται. Με βάση τις προδιαγραφές του χρονισμού VGA 640 x 480 που παρουσιάστηκαν παραπάνω σας δίνεται το παρακάτω σχηματικό με πύλες που υλοποιεί τις προδιαγραφές. Περιλαμβάνει 2 μετρητές τον **hcnt** και τον **vcnt** που μετρούν τις στήλες και τις γραμμές αντίστοιχα. Μετά τους μετρητές υπάρχουν set-clear flip-flops για να δημιουργηθούν κατάλληλα οι παλμοί HSYNC και VSYNC. Επίσης δημιουργούνται τα σήματα *o\_col*, *o\_row* και *o\_pix\_valid* που δέχεται το μπλοκ **vga\_frame** (σας δίνεται έτοιμο). Για τον παλμό HSYNC έχει προστεθεί ένας κύκλος καθυστέρησης έτσι ώστε το μπλοκ **vga\_frame** να βγάλει την έξοδό του (pixel) συγχρονισμένα με το μπλοκ **vga\_sync** στον «επόμενο κύκλο».



Θα πρέπει να υλοποιήσετε σε SystemVerilog RTL το κύκλωμα για το μπλοκ **vga\_sync**. Πριν πάτε στο εργαστήριο θα πρέπει να προσομοιώσετε και να επαληθεύσετε με το έτοιμο testbench και τα reference outputs τον κώδικά σας. Θα πρέπει να μπορείτε να δείτε την έξοδο που θα έβγαζε το κύκλωμα με τον προσομοιωτή VGA Simulator. Μη ξεχάσετε να βάλετε reset στους καταχωρητές!

#### **Τι πρέπει να κάνετε στο εργαστήριο:**

Θα πρέπει να πάτε στο εργαστήριο με τον **κώδικά σας έτοιμο και προσομοιωμένο από πριν** και να ακολουθήσετε την ροή του εργαλείου Xilinx Vivado και τα βήματα που χρειάζεται για να «κατεβάσετε» το σχέδιο στην FPGA και να το δείτε να δουλεύει. Δείξτε το κύκλωμα που δουλεύει στο βοηθό.

**Σημείωση:** Το ρολόι του κυκλώματος πρέπει να είναι 25MHz (περίοδος ρολογιού 40ns) για τη σωστή υλοποίηση του χρονισμού VGA με ανάλυση 640x480 (resolution) και ρυθμό ανανέωσης 60Hz (refresh rate), δηλαδή 60 VGA frames per second. Η πλακέτα του εργαστηρίου όμως έχει εξωτερικό ρολόι 100 MHz και στο εργαστήριο ο βοηθός θα σας βοηθήσει να δημιουργήσετε ρολόι 25 MHz μέσω του μπλοκ «Clocking Wizard» που υπάρχει έτοιμο στο Vivado. Στο testbench το ρολόι είναι δηλωμένο σωστά στα 25 MHz οπότε δεν χρειάζεται να κάνετε κάτι επιπλέον. Για την ανάθεση pins (pin assignment) χρησιμοποιείτε το constraint file που σας δίνεται (lab1.xdc).

#### **Τι πρέπει να παραδώσετε:**

Πρέπει να παραδώσετε τον SystemVerilog RTL κώδικα του μπλοκ **vga\_sync**. Η παράδοση θα πρέπει να γίνει **τη 2<sup>η</sup> εβδομάδα των εργαστηρίων (Εβδομάδα 11/04 έως 15/04)** στο τέλος της ώρας που έχετε εργαστήριο. Στείλτε τον κώδικά σας με e-mail στο [hy220@csd.uoc.gr](mailto:hy220@csd.uoc.gr) με τίτλο: *Lab1 – Ονοματεπώνυμο – ΑΜ*.

**Οι κώδικες θα ελέγχονται για αντιγραφές με ειδικό λογισμικό!**