



**HAL**  
open science

## “ Pong ” en VHDL

Sophie Dupuis, Vincent Creuze, Vincent Thomas, Didier Crestani, Bertrand Gelis, Eric Pommier, Isabelle Pinon

► **To cite this version:**

Sophie Dupuis, Vincent Creuze, Vincent Thomas, Didier Crestani, Bertrand Gelis, et al.. “ Pong ” en VHDL. Journées pédagogiques du CNFM, Nov 2016, Montpellier, France. 2016. lirmm-01430000

**HAL Id: lirmm-01430000**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01430000>**

Submitted on 9 Jan 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# « Pong » en VHDL

Sophie Dupuis, Vincent Creuze, Vincent Thomas,  
Didier Crestani, Bertrand Gelis, Eric Pommier et Isabelle Pinon

IUT de Montpellier, Département GEII, 99 avenue D'Occitanie, 34296 Montpellier, France  
<mailto:{prénom.nom}@umontpellier.fr>

**RESUME :** Cet article présente le module complémentaire « Composants complexes FPGA » de deuxième année de DUT Génie Electrique et Informatique Industrielle (GEII). Ce module est composé de deux séances de cours magistraux, huit séances de travaux dirigés (TDs) et cinq séances de travaux pratiques (TPs) permettant aux étudiants d'approfondir leurs connaissances en VHDL. Durant les TD/TPs, le logiciel Quartus est utilisé, ainsi que des cartes FPGAs DE1 d'Altera. Les TDs servent à se remémorer la syntaxe VHDL acquise en première année pour les circuits combinatoires et séquentiels, ainsi qu'à appréhender la description hiérarchique et des fonctionnalités non abordées en première année (boucles, fonctions, paramètres génériques, ...). Après deux TPs de « remise à niveau », les trois derniers TPs se focalisent sur l'utilisation de la sortie VGA des cartes avec pour but la programmation du jeu « Pong ».

## I. INTRODUCTION

Durant leur cursus de DUT GEII, les étudiants appréhendent la numération et la logique booléenne durant le module « Systèmes d'Information Numériques » (SIN) du premier semestre. Ils s'initient également au langage VHDL, langage de description de matériel destiné à représenter le comportement et l'architecture d'un circuit numérique [1]. Ces notions n'étant pas forcément bien acquises en sortie de premier semestre (ou oubliées depuis ...), il nous a paru intéressant de proposer de nouveau un module sur ce même thème, durant un des modules complémentaire du quatrième semestre : « Composants complexes FPGA ».

Une première partie du module sert à consolider les bases acquises, puis une seconde sert à approfondir les connaissances en VHDL. Durant les séances de TPs, une carte Altera DE1 est utilisée (cf. Fig. 1). Outre les boutons, LEDs et afficheurs 7-segments, l'utilisation du port VGA nous a semblé intéressante dans le but de rendre ces séances ludiques. Après avoir pris en main le format VGA et réussi à afficher des couleurs puis des formes, le but final est de programmer un jeu de « Pong », un des premiers jeux vidéo dans années 1970 dans lequel deux joueurs s'affrontent dans une partie inspirée du tennis de table, en vue de dessus, comme montré dans la figure 2 [2].



Figure 1. Carte Altera DE1

Dans cet article, nous présentons les différents aspects de ce module, en nous focalisant sur les manipulations faites en TDs/TPs.



Figure 2. Pong

## II. COURS MAGISTRAL

Les séances de cours magistral ont pour but d'expliquer le contexte économique de la microélectronique, ce que sont les ASICs et les FPGAs, comment ils sont construits, les avantages/désavantages de chacun. Il nous a semblé important d'introduire la notion d'ASIC pour pouvoir expliquer comment un FPGA était créé. Les similitudes et les différences dans le flot de conceptions ainsi que le coût nous ont paru également importantes à aborder.

## III. TDs et TPs

Tout d'abord, les séances de TDs et TPs s'alternent de façon à ce que les TPs permettent de mettre en pratique les notions présentées dans les TDs précédents. Puis le module se termine par une série de trois TPs ayant pour sujet le format VGA. Par ailleurs, des TDs de rappel s'alternent avec des TDs proposant un approfondissement des connaissances en VHDL.

### A. Rappels sur les circuits combinatoires

Les deux premières séances de TDs sont focalisées sur les circuits combinatoires. Le premier TD a pour objet la description en VHDL et la simulation d'un circuit affichant sur deux afficheurs sept-segments la valeur décimale (sur 2 digits : de « 00 » à « 15 ») correspondant à un nombre binaire non signé sur 4 bits (de « 0000 » à « 1111 ») en entrée du circuit. Le second TD s'intéresse à la mise en œuvre en schéma bloc et la simulation d'un additionneur à propagation de retenue [3].

Les étudiants vont se « remémorer » la création des schéma-blocs, la syntaxe VHDL (entity, architecture...), ainsi que

la conception des circuits combinatoires : utilisation d'instructions concurrentes (with select ou when else et expression booléennes, les instruction séquentielles if et case interdites), et la façon de simuler un circuit. Par ailleurs, un second but est d'approfondir leurs connaissances du logiciel Quartus et de leur faire comprendre comment leur description VHDL est synthétisée. Pour cela, il leur est demandé d'analyser le rapport de compilation et d'utiliser les outils de Netlist Viewer pour visualiser le schéma généré (correspondant donc à leur description VHDL). Une étude théorique de la table de vérité du circuit est réalisée et des tableaux de Karnaugh sont utilisés pour en déduire les expressions booléennes de chaque sortie, de façon à faire un lien avec le schéma créé par la synthèse. Une analyse de délai est également faite, et une seconde synthèse est effectuée après avoir fixé un temps de propagation inférieur à celui trouvé sans contrainte.

Ensuite, la première séance de TP permet de prendre en main la carte DE1. Le but est d'implanter le convertisseur fait dans le TD1 (entrées sur des interrupteurs et sorties sur deux afficheurs 7-segments) et l'additionneur fait dans le TD2 (entrées sur des interrupteurs et sorties sur des LEDs). Il est ensuite demandé de créer un circuit en schéma bloc utilisant les deux composants de façon à voir le résultat d'une addition (entrées sur interrupteurs) sur les afficheurs 7-segments. Enfin, il est demandé d'apporter les modifications nécessaires à ce circuit pour ajouter une entrée permettant de choisir l'opération à effectuer : addition ou soustraction.

### B. Approfondissement des circuits combinatoires

Deux TDs ont ensuite pour but d'approfondir les connaissances en VHDL. L'additionneur à propagation de retenue est effectué en VHDL structurel et devient générique. Ensuite un générateur et un vérificateur de parité génériques sont créés.

Ces TDs permettent d'introduire les notions non abordées en première année: généricité (mot clé generic), fonctions, et hiérarchie. Les étudiants sont amenés à comprendre qu'une architecture hiérarchique permet de séparer un circuit en un ensemble de blocs plus petits, ayant des fonctions bien identifiées, ce qui amène à une architecture plus compréhensible.

### C. Rappel sur les circuits séquentiels

Les deux TDs suivants ont pour objet les circuits séquentiels. Tout d'abord, il est demandé de coder en VHDL un compteur/décompteur (interface montrée dans la figure 3). Le but est de se remémorer la notion de process en mettant un accent particulier sur les notions de mode synchrone, mode asynchrone et priorité. Ensuite, la notion de machine à états est abordée : fonctionnement d'un feu tricolore avec bouton piéton pour pouvoir traverser. La netlist générée lors de la synthèse est de nouveau analysée et comparée avec la version trouvée lors de l'étude théorique.

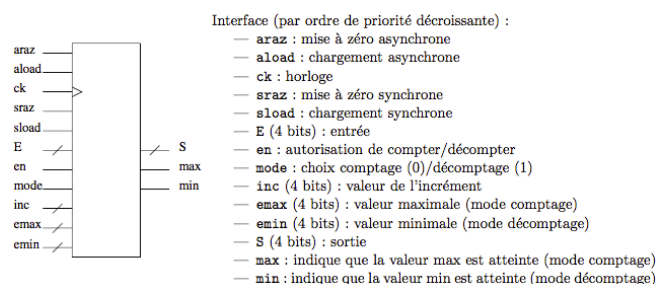


Figure 3. Compteur-Décompteur

Le second TP a ensuite pour but d'implanter dans la carte le compteur/décompteur (entrées sur interrupteurs, horloge sur bouton poussoir, sorties sur LEDs), puis de le coupler au convertisseur pour que l'affichage du comptage se fasse sur les afficheurs 7-segments. Enfin, il est demandé que le cadencement ne se fasse plus avec un bouton-poussoir mais « automatiquement ». Pour ce faire, il faut utiliser une des horloges internes de la carte. Les horloges étant à plusieurs méga Hertz, un diviseur de fréquence est créé de façon à ce que l'affichage se mette à jour à chaque seconde.

### D. Approfondissement des circuits séquentiels

Dans un premier TD, la machine à états codée précédemment est réalisée avec l'éditeur de machine à états du logiciel Quartus.

Dans un second TD, un générateur d'impulsions (PWM) est codé en prenant en compte trois paramètres : la fréquence voulue, le rapport cyclique et un nombre d'impulsions à générer. Ensuite, une liaison synchrone SPI est codée. Le circuit a pour but de transformer une entrée série (entrée sur 1 bit et 24 coups d'horloge nécessaires) en trois sorties parallèles de 8 bits (cf. Fig. 4).

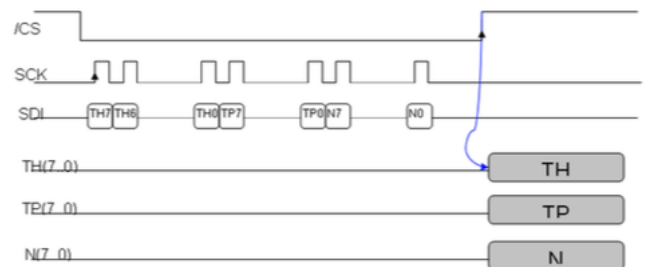


Figure 4. Liaison SPI

## IV. Projet VGA

Le projet VGA est effectué lors des derniers TPs. L'objectif final est de créer un jeu de « Pong ». Pour atteindre ce but, trois séances sont décomposées comme suit :

- 1) Prise en main du format VGA et affichage de couleurs,
- 2) Affichage d'un rectangle et déplacement du rectangle,
- 3) Programmation du jeu « Pong ».

### A. Affichage de couleurs

La première séance commence par une explication du format VGA [4]. Les notions de fréquence de rafraichissement ainsi que de synchronisation verticale et synchronisation horizontale sont présentées (cf. Figure 8) ainsi que les cinq signaux de la norme VGA (cf. Figure 9): deux signaux de synchronisation et trois signaux de couleur au format RGB. Les signaux RGB peuvent être sur plusieurs bits d'après la norme VGA, nous choisissons de n'utiliser qu'un bit par couleur.

Le premier exercice a pour but d'afficher différentes couleurs en fonction des valeurs de 3 interrupteurs correspondant aux trois couleurs rouge, vert et bleu (par exemple si les entrées rouge et bleu sont mise à '1' et l'entrée vert est mise à '0', c'est du magenta qui doit être affiché). La première tâche consiste donc à créer le composant GeneSync générant les signaux de synchronisation du format VGA (cf. Fig. 5). Cette tâche étant considérée comme la plus difficile, les étudiants sont guidés de façon à comprendre comment créer une machine à états générant les signaux voulus.

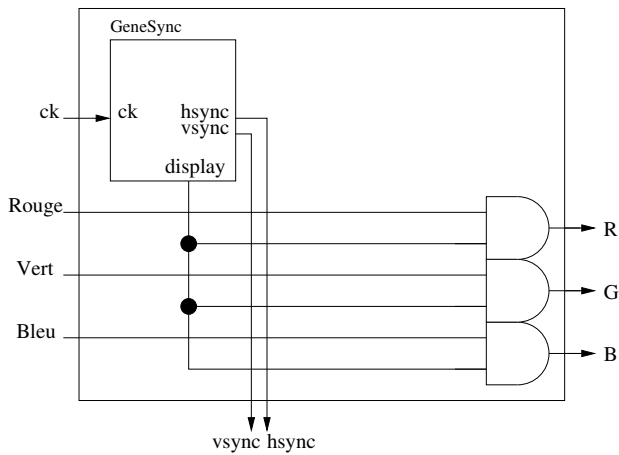


Figure 5. Affichage de couleurs (1)

La machine à états créée est cadencée par une des horloges internes de la carte, de façon à ce que chaque période d'horloge corresponde à l'affichage d'un pixel. L'horloge à 27MHz est utilisée pour que la fréquence de rafraichissement soit quasiment de 60Hz. Comme montré dans la figure 9, deux compteurs imbriqués sont utilisés pour générer le balayage XY de l'écran, permettant de déterminer si chaque signal de synchronisation doit être à '0' ou à '1'. Nous faisons le choix d'également générer un signal correspondant au temps de « display ». Ce signal a pour rôle d'autoriser ou inhiber les couleurs en sortie (grâce à des portes AND).

Il est demandé dans un second temps de n'avoir qu'une entrée permettant de choisir soit la couleur magenta, soit la couleur cyan. Pour cela, le composant GeneRGB est ajouté, permettant de générer la couleur à afficher (cf. Fig. 6).

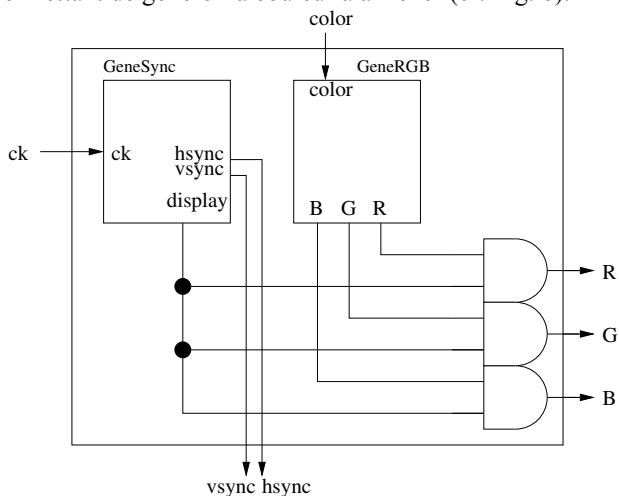


Figure 6. Affichage de couleurs (2)

#### A. Affichage et déplacement d'un rectangle

Le but du TP2 n'est plus d'afficher une seule couleur sur tout l'écran mais un rectangle d'une couleur donnée, à des coordonnées données, sur un fond noir. Pour cela, on utilise les compteurs du composant générant les signaux de synchronisation pour connaître la coordonnée (X, Y) courante d'affichage. Le composant GeneRGB permettant de choisir la couleur à afficher utilise cette information pour n'afficher une couleur donnée qu'aux coordonnées choisies (cf. Fig. 7). On affiche ainsi un rectangle immobile.

Le rectangle doit ensuite parcourir l'écran horizontalement. Pour cela, il faut mettre à jour l'image à chaque rafraichissement. Le composant affichant les couleurs va être cadencé par une horloge. Les étudiants doivent choisir parmi les signaux à leur disposition celui qui peut jouer le rôle de cette horloge (vsync).

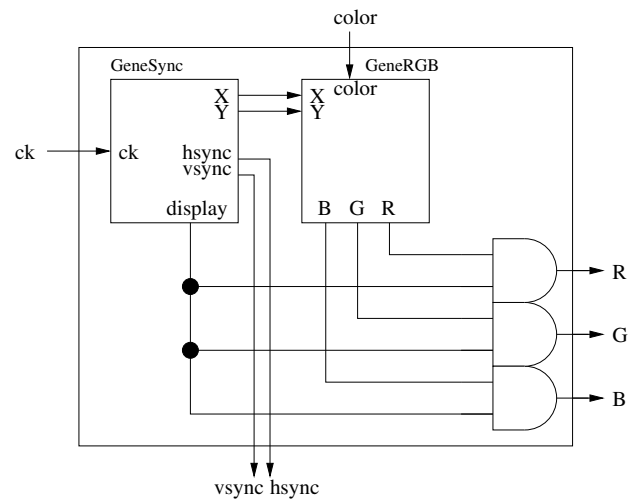


Figure 7. Affichage d'un rectangle

#### A. Pong

Après ces deux TP de prise en main du format VGA, la réalisation d'un jeu « Pong » n'apportant aucune nouveauté, les étudiants sont laissés le plus possible en autonomie avec le cahier des charge suivant (variante de « Pong » avec un seul joueur) :

*Au début du jeu, une balle est positionnée au milieu de l'écran et une raquette est positionnée à droite de l'écran. Un interrupteur start permet (à '1') de faire démarrer le jeu et (à '0') de réinitialiser le jeu. Un interrupteur pause permet d'arrêter les mouvements de la balle et de la raquette. Une fois le jeu démarré, la balle rebondit sur les "murs" du haut, du bas et de gauche. Elle rebondit également sur la raquette si cette dernière est à l'endroit adéquat. La raquette se déplace de haut en bas grâce à deux boutons-poussoir. Si la balle atteint le mur de droite, la partie est perdue. A chaque fois que la balle rebondit sur la raquette, le joueur gagne 1 point. Une fois 15 points atteints, la partie est gagnée.*

Les étudiants sont guidés par les items suivants :

- 1) Positionner une balle jaune (carré de largeur 10 pixels) au milieu de l'écran et une raquette bleue (rectangle vertical de longueur 60 pixels et de largeur 10 pixels) à droite de l'écran (cf. Fig. 10),
- 2) A condition que l'interrupteur start soit à '1', et l'interrupteur pause à '0', faire se déplacer la raquette en haut et en bas grâce à deux boutons poussoirs.
- 3) A condition que l'interrupteur start soit à '1', et l'interrupteur pause à '0', faire se déplacer la balle en diagonale, et rebondir sur les 4 murs (tout comme pour la raquette, si l'interrupteur pause est à '1', la balle ne peut pas bouger. Si l'interrupteur start est à '0', la position de la balle est réinitialisée à sa position d'origine),
- 4) Gérer le rebondissement de la balle sur la raquette,
- 5) Si la balle atteint le mur de droite, elle est stoppée. Un nouveau jeu peut démarrer en passant l'interrupteur start à '0' puis à '1',
- 6) Rajouter l'affichage du score sur les afficheurs 7-segments de la carte, ainsi que l'arrêt du jeu sur le score 15 est atteint (cf. Fig. 11).

## V. CONCLUSION

Le cours magistral de ce nouveau module du semestre 4 de DUT GEII a pour but de familiariser les étudiants avec le flot de conception d'un circuit intégré, que ce soit un ASIC ou un FPGA. Les TDs/TPs permettent d'approfondir l'enseignement en VHDL dispensé au semestre 1, ainsi que les connaissances dans le logiciel Quartus. Cet enseignement est dispensé pour la troisième année consécutive, aux étudiants de semestre 4 de DUT ainsi qu'en licence mécatronique (environ 80 étudiants par an). Pour la première fois en 2017, il est envisagé que les TDs/TPs soient dispensés en anglais à un groupe de semestre 4.

Il est à noter que le thème ludique « programmation d'un jeu vidéo » est également abordé dans l'enseignement de langage C du premier semestre, dans lequel un jeu de Pong

est créé à l'aide de la librairie Simple DirectMedia Layer [5]. Suite à cela, des projets tutorés sont proposés chaque année au semestre 2. Plusieurs jeux vidéos ont vu le jour : Casse briques, Space Invaders, Mario, Demineur, Candy Crush.

## REFERENCES

- [1] <https://fr.wikipedia.org/wiki/VHDL>
- [2] <https://fr.wikipedia.org/wiki/Pong>
- [3] <https://fr.wikipedia.org/wiki/Additionneur>
- [4] <http://javiervalcarce.eu/html/vga-signal-format-timing-specs-en.html>
- [5] [https://fr.wikipedia.org/wiki/Simple\\_DirectMedia\\_Layer](https://fr.wikipedia.org/wiki/Simple_DirectMedia_Layer)

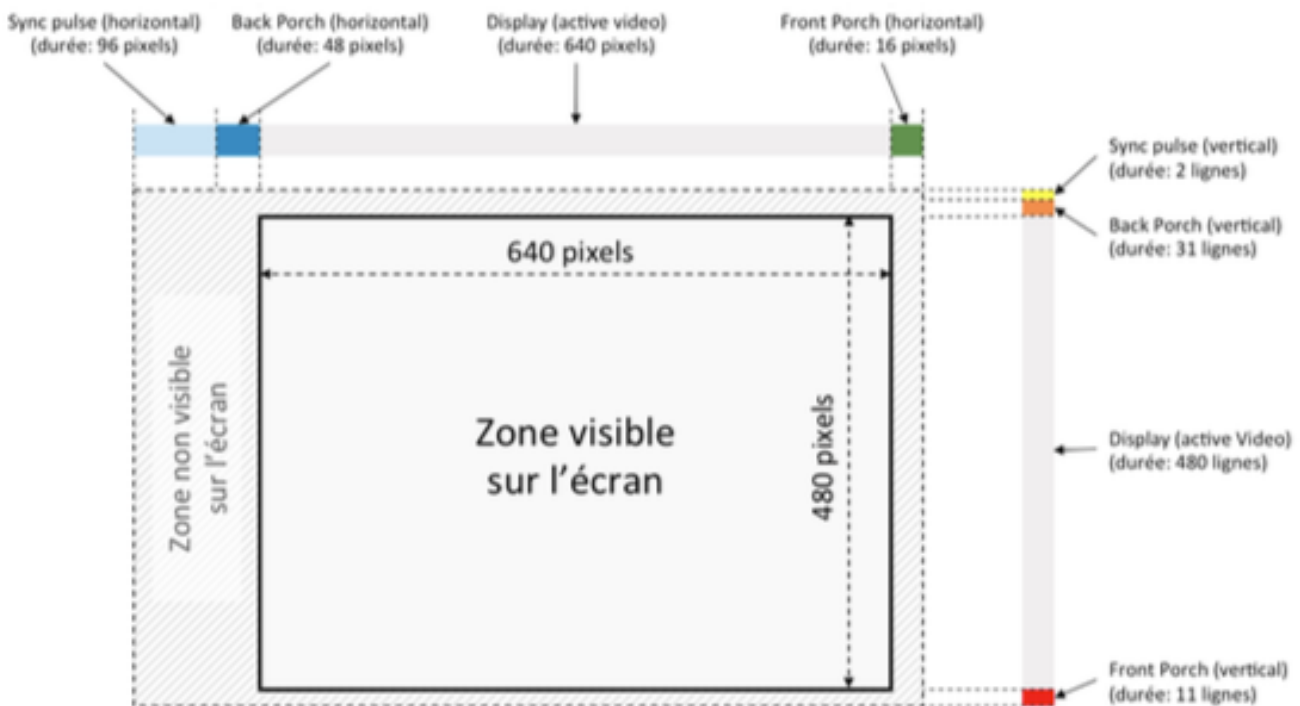


Figure 8. VGA (1)

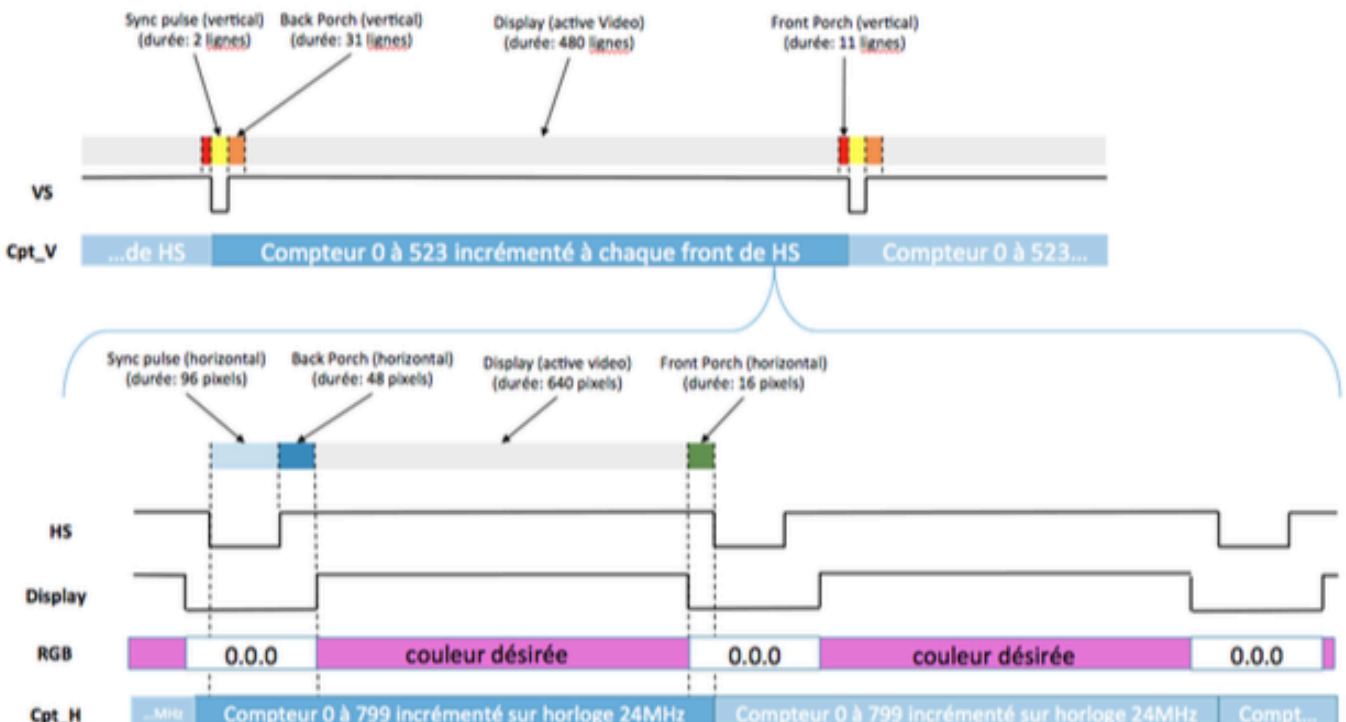


Figure 9. VGA (2)

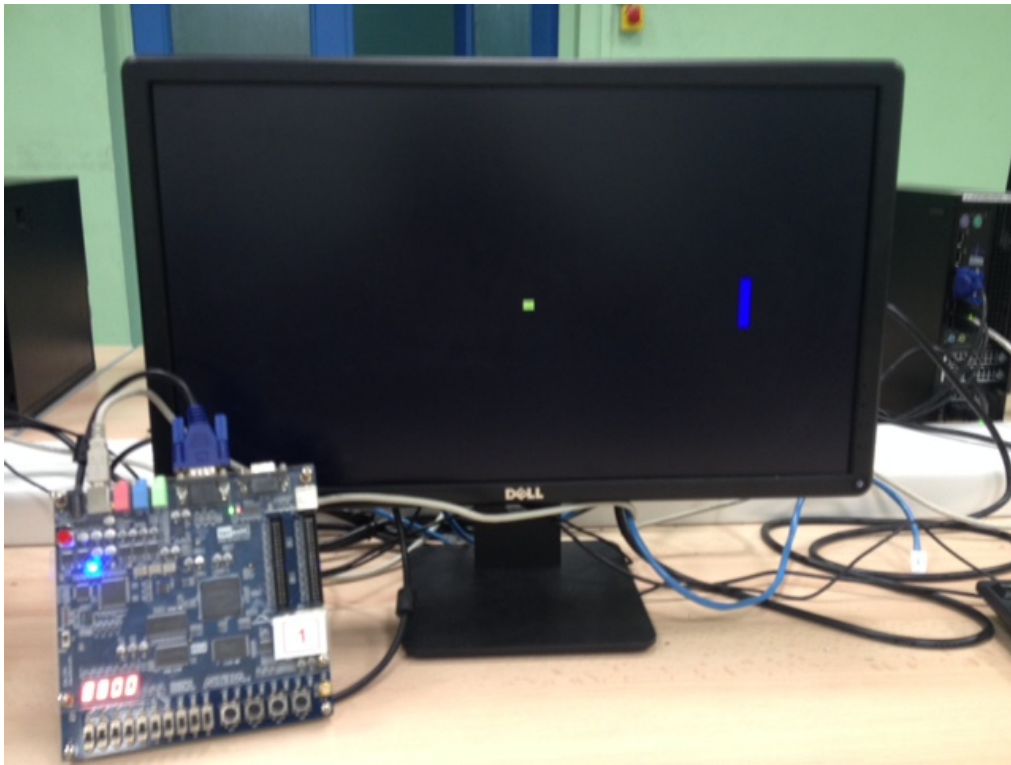


Figure 10. Pong : Démarrage

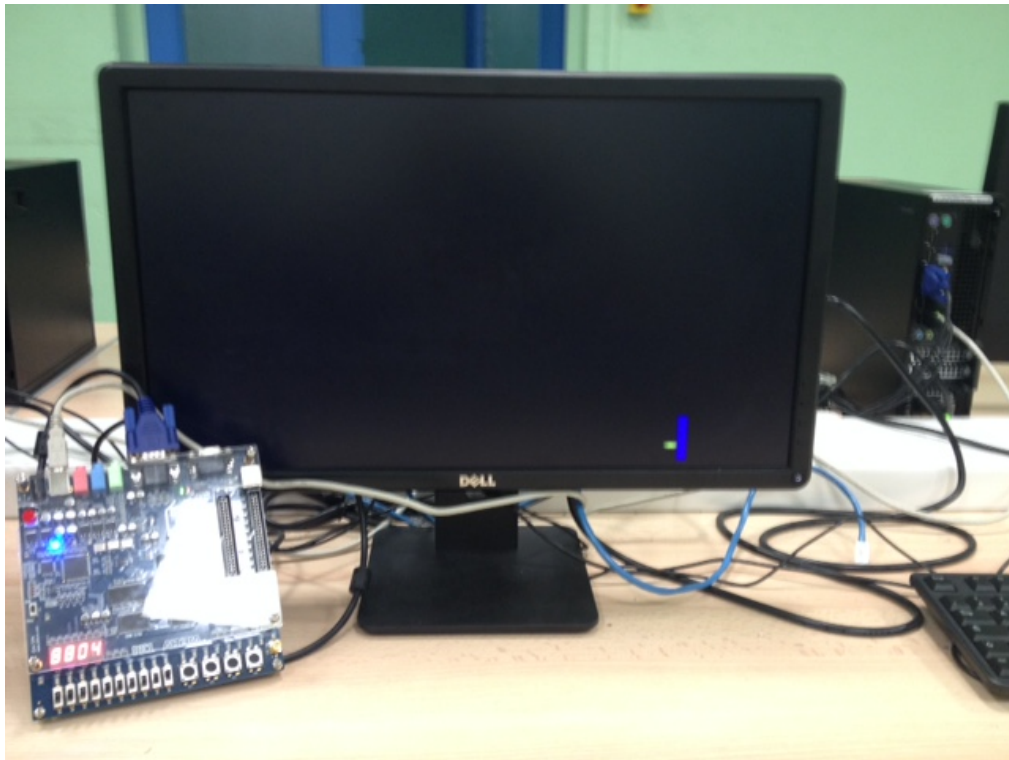


Figure 11. Pong : Jeu en cours