

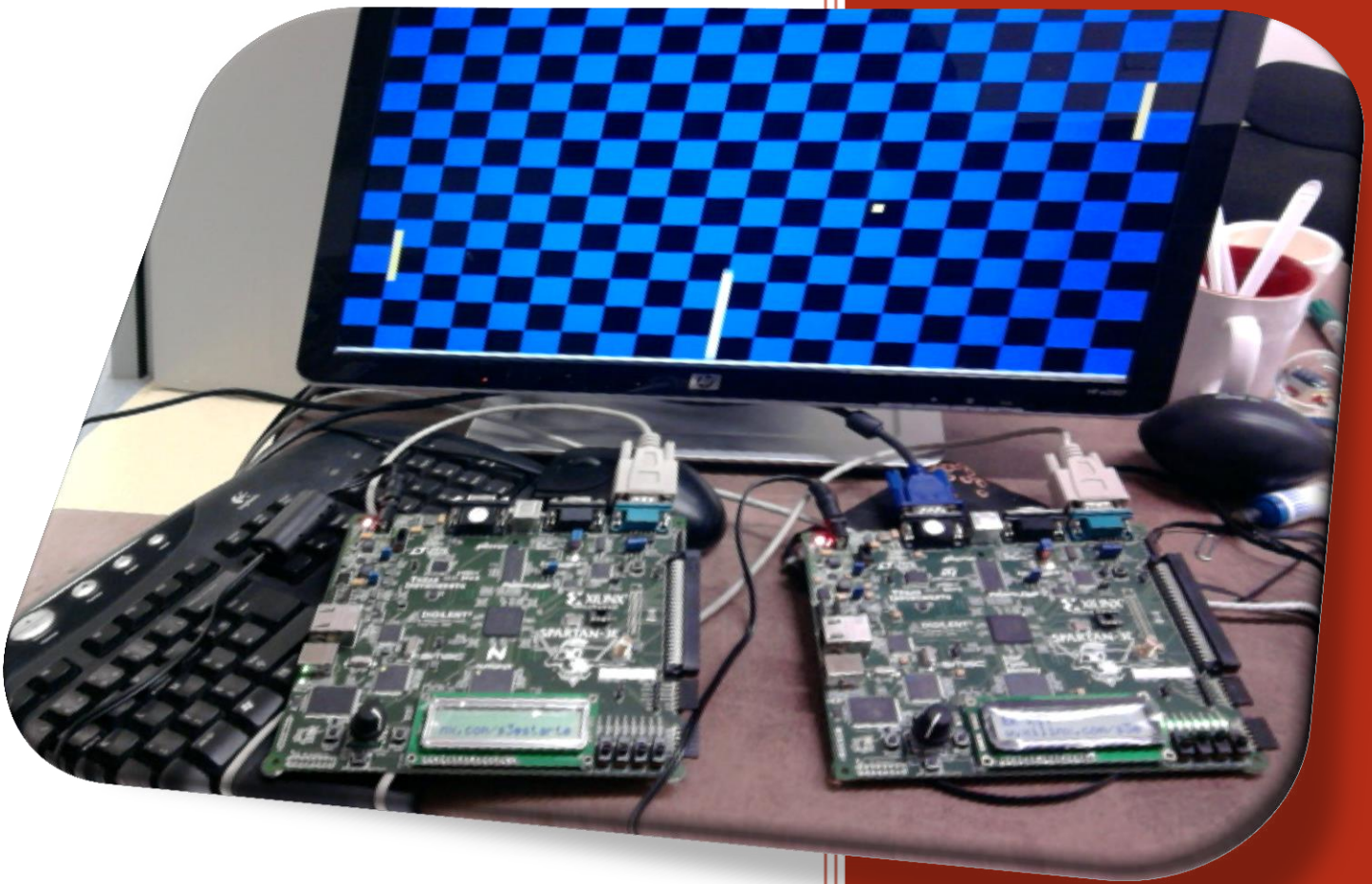


**SORBONNE
UNIVERSITÉ**

CRÉATEURS DE FUTURS
DEPUIS 1257

LU3EE100

Console SU-EE100 sur FPGA



Manuel Utilisateur + Carte FPGA

Julien Denoulet

LU3EE100 - Mini Projet

L3 EEA

PRESENTATION GENERALE DE LA CONSOLE SU-EE100

1) Introduction

La console SU-EE100 contient deux jeux comptant parmi les plus célèbres de l'histoire des jeux vidéo

- Casse-Briques (pour 1 joueur)
- Pong (pour 2 joueurs)

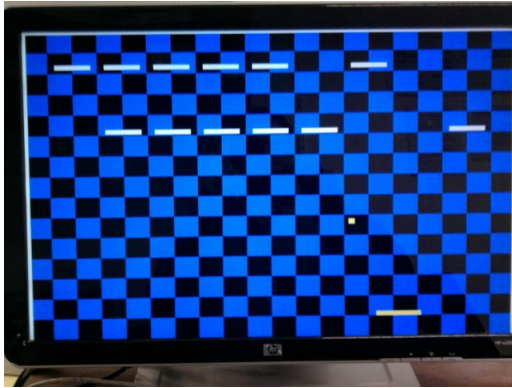


Figure 1 – Casse Briques

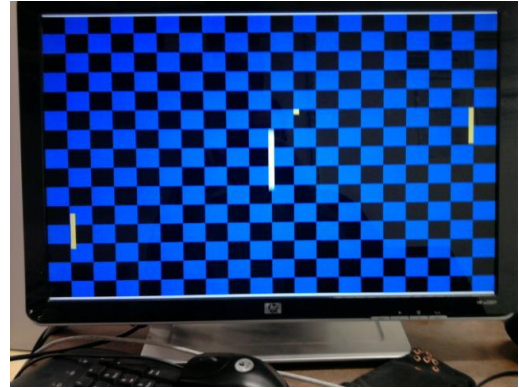


Figure 2 – Pong

L'utilisateur peut à tout moment choisir le jeu auquel il a envie de jouer, fixer un certain nombre d'options (vitesse de la balle, taille des raquettes...) et également configurer la SU-EE100 en mode console ou en mode manette.

Pour déplacer la raquette du Casse-Briques, le joueur peut utiliser l'accéléromètre présent sur la carte ou bien un encodeur rotatif connecté à la console. Pour le jeu Pong, il faut utiliser deux SU-EE100 : l'une en mode console, et l'autre en mode manette. La manette se connecte à la console via un câble.

L'affichage se fait sur un écran grâce à la sortie VGA de la console.



Figure 3 : Borne d'arcade originale du jeu Pong (1972)



2) Présentation des jeux



Figure 4 : Borne d'arcade jeu Casse Briques (1975)

- **Casse Briques**

- Le principe général de ce jeu est de détruire, à l'aide d'une balle, un ensemble de briques se trouvant dans la partie supérieure de l'écran.
- Pour cela, le joueur contrôle une raquette qu'il peut seulement déplacer sur un axe horizontal au bas de l'écran.
- Le but est d'empêcher la balle de franchir cette ligne en la frappant avec la raquette.
- Si le joueur ne parvient pas à rattraper la balle :
 - il perd la partie. Cela se traduit par un écran rouge pendant quelques secondes.
 - La balle est alors remise en jeu pour donner une nouvelle chance au joueur.
- Si le joueur parvient à casser toutes les briques,
 - La partie est gagnée. L'écran devient entièrement vert.
 - Pour recommencer une partie, il faut réinitialiser le jeu en faisant un Reset de la console
- Il y a par ailleurs plusieurs options de jeu :
 - Réglage de la taille de la raquette
 - Réglage de la vitesse de la balle
 - Mise en pause du jeu

- **Pong**

- Pong est une simulation simpliste de tennis de table (ping-pong) qui se joue à deux.
 - Le 1^{er} joueur utilise l'encodeur rotatif de la SU-EE100 configurée en mode Console.
 - Il contrôle ainsi la raquette située à gauche de l'écran
 - Le 2^{ème} joueur utilise l'encodeur rotatif d'une autre SU-EE100, configurée en mode Manette, et reliée à l'autre SU-EE100 via le port série.
 - Le 2^{ème} joueur contrôle la raquette située à droite de l'écran.
- Une balle se déplace à travers l'écran, rebondissant sur les rebords du haut et du bas
- Les deux joueurs commandent chacun une raquette, la faisant glisser verticalement entre les extrémités de l'écran.
- Si la balle frappe la raquette, elle rebondit vers l'autre joueur.
- Si le joueur manque la balle avec sa raquette, il perd la partie. Cela se traduit par un écran rouge pendant quelques secondes.
- Il y a par ailleurs plusieurs options de jeu :
 - Réglage de la taille de la raquette
 - Réglage de la vitesse de la balle
 - Insertion d'un obstacle mobile au milieu de l'écran. Si la balle rebondit contre cet obstacle, elle repart dans le sens inverse.
 - Mise en pause du jeu



3) Interface joueur

Pour déplacer les raquettes des jeux, le joueur peut utiliser l'accéléromètre présent sur la carte ou bien un encodeur rotatif (Figure 5.1 ci-contre).

- L'encodeur tourne vers la gauche ou la droite.
- Il est relié à la carte via le connecteur **PMOD B** ou **PMOD C** selon que la carte est configurée en mode Console (PMOD B) ou Manette (PMOD C).

L'interface joueur de la SU-EE100 se compose également de (Figure 5.2):

- 16 interrupteurs dont 5 sont utilisés :
 - o **S15, S3, S2, S1 et S0.**
 - o **S15** est l'interrupteur le plus à gauche.
- 5 boutons poussoirs
 - o **Nord, Sud, Est, Ouest et Centre**

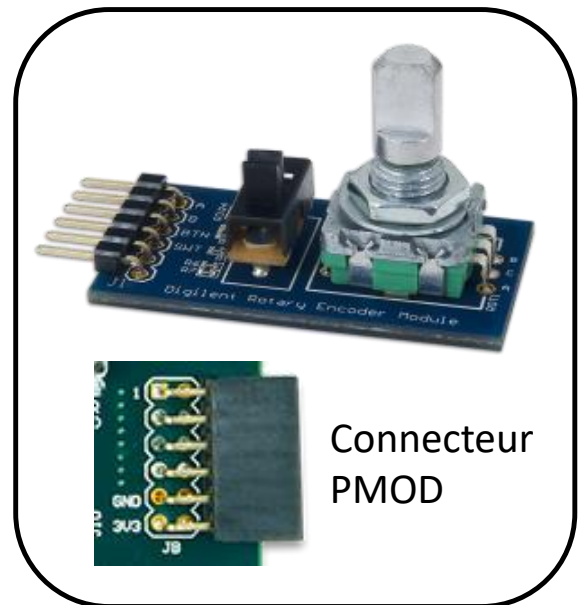


Figure 5.1 : Encodeur rotatif et connecteur PMOD

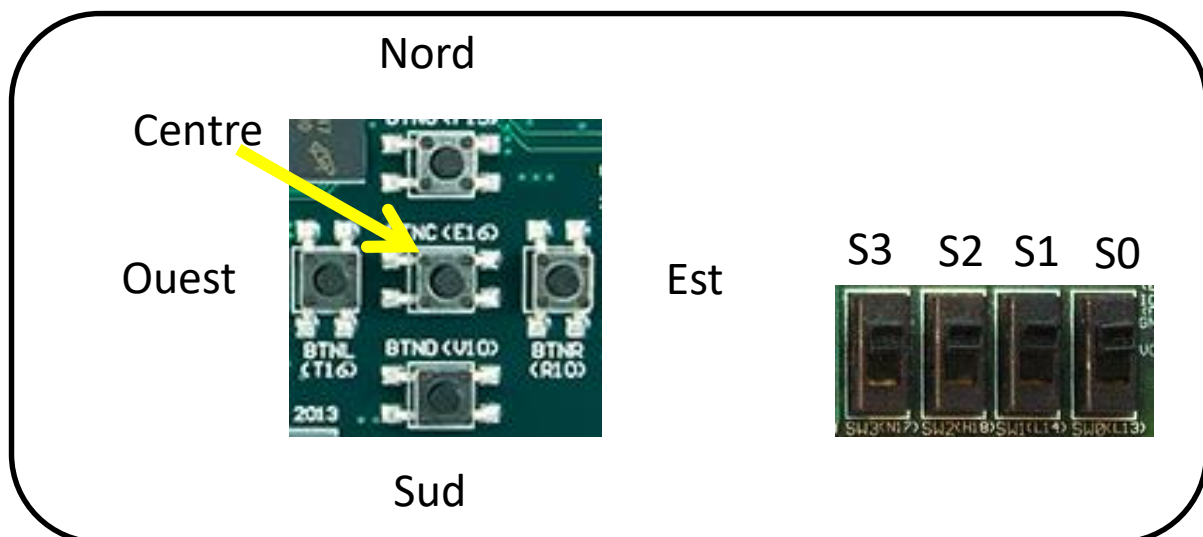


Figure 5.2 : Interface joueur de la SU-EE100

- **Commandes générales de la SU-EE100**
 - o **S0** : Commande Marche/Arrêt de la console
 - Interrupteur vers le bas → Arrêt
 - Interrupteur vers le haut → Marche
 - o **S15** : Choix du composant pour le déplacement des raquettes
 - Interrupteur vers le bas → Encodeur rotatif
 - Interrupteur vers le haut → Accéléromètre

- **Ouest** : Reset système
 - Actif si on appuie sur le bouton
 - Permet de redémarrer une nouvelle partie de Casse Briques après une partie gagnée.
 - **Sud**: Sélection du mode de la SU-EE100
 - L'appui sur le bouton permet de passer du mode Console au mode Manette et inversement.
 - **Nord** ou **Est**: Sélection du jeu actif
 - L'appui sur l'un des boutons permet de passer du jeu Casse Briques au jeu Pong et inversement.
 - **Centre**: Mise en pause du jeu.
 - L'appui sur le bouton permet de mettre le jeu en pause.
 - Un nouvel appui permet de reprendre le jeu.
- **Commandes communes aux deux jeux**
- **S3** : Commande de la taille des raquettes
 - Interrupteur vers le bas → Raquettes courtes
 - Interrupteur vers le haut → Raquettes longues
 - **S2** : Commande de la vitesse de la balle
 - Interrupteur vers le bas → Vitesse lente
 - Interrupteur vers le haut → Vitesse rapide
- **Commande spécifique au jeu Casse Briques**
- **Encodeur rotatif ou accéléromètre**:
 - Déplacement horizontal de la raquette
- **Commandes spécifiques au jeu Pong**
- **Accéléromètre (carte console seulement) ou encodeur rotatif (pour les deux joueurs)**:
 - Déplacement vertical de la raquette selon le
 - **S1** : Ajout d'un obstacle mobile
 - Interrupteur vers le bas → Pas d'obstacle
 - Interrupteur vers le haut → Présence d'un obstacle



Figure 6 – La Magnavox Odyssey, 1^{ère} console vidéo (1972)



PLATE-FORME FPGA

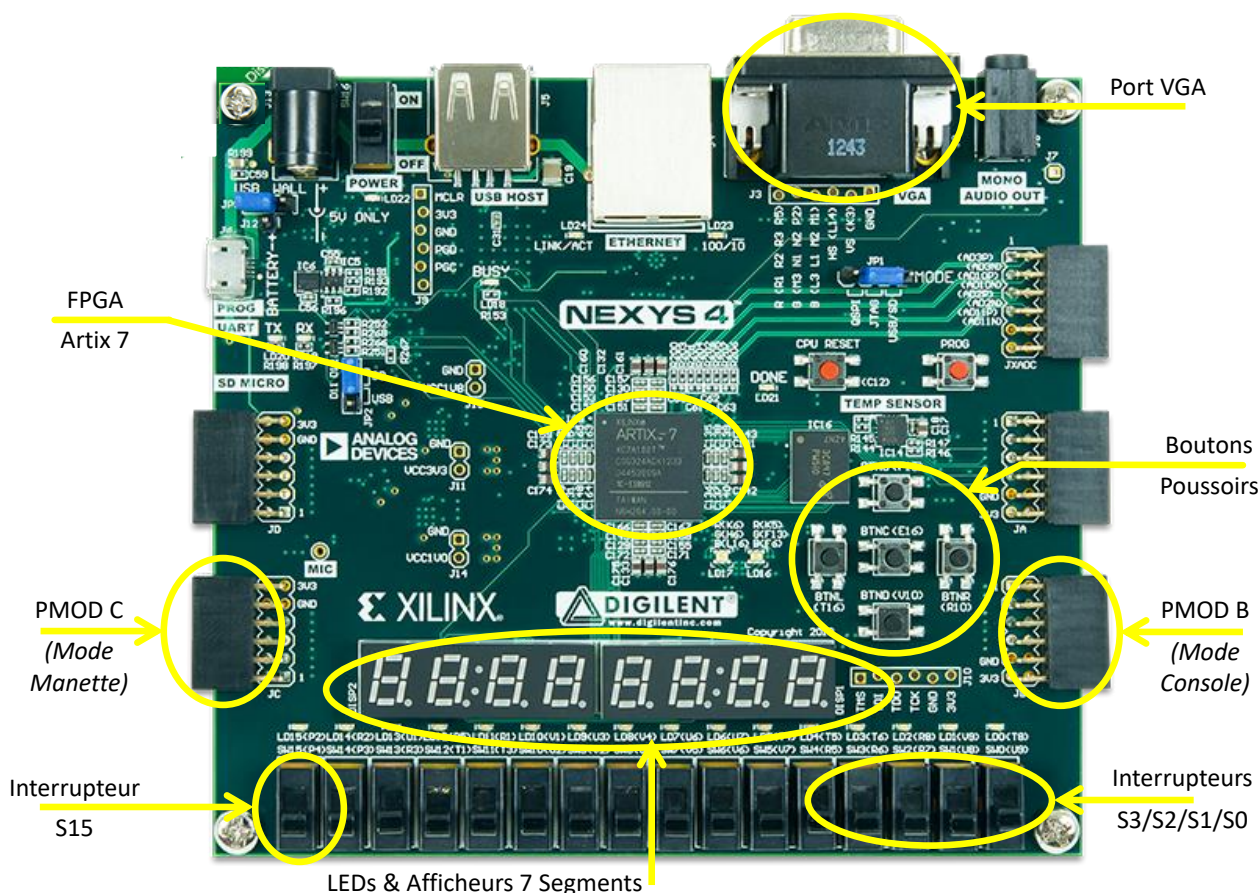


Figure 7 – Carte Nexys 4

La console de jeux SU-EE100 est implémentée sur une carte Xilinx Nexys 4 (Figure 7). Cette carte dispose des éléments suivants :

- **FPGA Artix 7**
 - Ce composant programmable contient toute l'architecture numérique de la console
 - Il est cadencé par une horloge interne de fréquence **100 MHz**
- **3 Connecteurs PMOD**
 - PMOD B : On y connecte l'encodeur rotatif en mode Console
 - PMOD D : On y connecte l'encodeur rotatif en mode Manette
 - Pour jouer à 2 joueurs, les 2 PMOD doivent également être reliées par un câble RJ45.
- **Boutons Poussoirs**
 - Les 5 boutons poussoirs (Nord, Sud, Est, Ouest et Centre), permettent de paramétrer la console
- **Interrupteurs**
 - Ces 5 interrupteurs (S15 et S3-S0) permettent également de paramétrer l'activité de la console
- **Afficheurs 7 segments**
 - Fournissent des messages (mise en pause, jeu activé...)
- **LED**
 - Les 16 LED indiquent si la SU-EE100 est en mode console (allumées) ou manette (éteintes).
- **Port VGA**
 - Ce connecteur permet à la carte d'envoyer des signaux vidéo vers un écran VGA.

- **Accéléromètre**
 - Ce composant se trouve sur la face arrière de la carte Nexys 4.

4) Architecture générale du FPGA

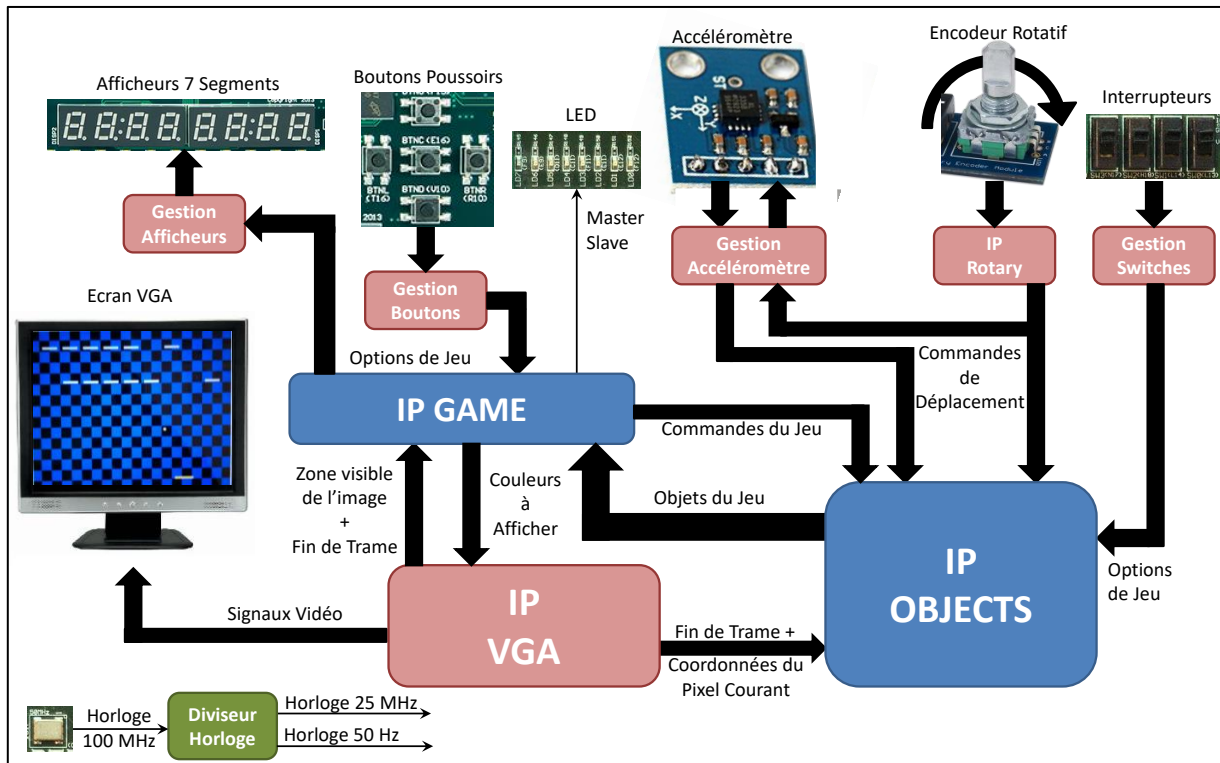


Figure 8 – Architecture interne du FPGA

L'architecture de la console implémentée dans le FPGA est donnée en Figure 8. On distingue dans ce schéma-blocs simplifié 8 modules.

1) **Diviseur Horloge**

- Ce module comprend en fait 2 blocs qui génèrent 2 horloges à partir de l'horloge référence de 100 MHz. Une horloge 25 MHz pour cadencer la quasi-totalité des modules de l'architecture, et une horloge à 25 Hz pour traiter les signaux issus de l'accéléromètre.

2) **Gestion Boutons**

- Ce module détecte les événements survenant sur les 5 boutons poussoirs et génère en conséquence plusieurs signaux pour modifier le comportement de la console (reset système, sélection mode console/manette, choix du jeu, demande de pause...)

3) **Gestion Switches**

- Ce module détecte les événements survenant sur les 5 interrupteurs et génère en conséquence plusieurs signaux pour modifier le comportement de la console (marche /arrêt du système, vitesse de la balle, taille des raquettes, choix entre l'encodeur rotatif et l'accéléromètre...)

4) **IP Rotary**

- Ce module détecte les événements survenant sur l'encodeur rotatif (rotary encoder en Anglais) connecté à la carte et génère en conséquence des commandes de déplacement pour les objets du jeu (les raquettes) qui sont pilotés par cet encodeur.



5) Gestion Accéléromètre

- Ce module détecte les déplacements de la carte mesurés par l'accéléromètre et génère en conséquence des commandes de déplacement pour les objets du jeu (les raquettes) qui sont pilotés par cet accéléromètre.

6) IP VGA

- Ce module envoie à l'écran VGA les signaux vidéo permettant d'afficher les images du jeu. La couleur des pixels est donnée en entrée par le module **IP Game**.
- Le module transmet également aux autres blocs de l'architecture des informations sur l'image qui est en train d'être affichée (coordonnées du pixel courant, fin de l'image...)

7) Gestion Afficheurs

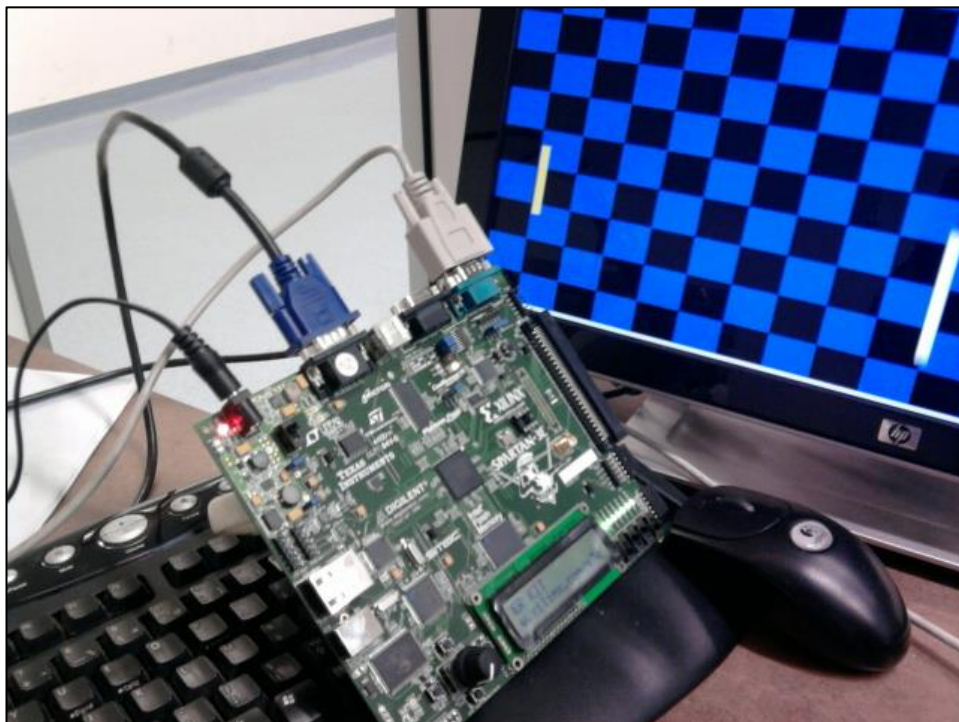
- Ce module contrôle les 8 afficheurs 7 segments de la carte pour afficher des messages d'information au joueur.

8) IP Objects

- Ce module a pour but de générer tous les objets nécessaires au jeu Pong ou Casse-Briques (Raquettes, balle, murs, briques...). Pour cela, il prend en entrée :
 - Des commandes de jeu transmises par **IP Game** et **Gestion Switches**
 - Des commandes de déplacement transmises par **IP Rotary** et **Gestion Accéléromètre**
 - Des informations sur l'image à afficher transmises par **IP VGA**.

9) IP Game

- Ce module génère :
 - Des commandes vers **IP Objects**, **IP RS232** et les **LED** donnant le comportement général de la SU-EE100 (mode console/manette, choix du jeu, mode pause...)
 - La couleur du pixel à afficher. Cette information est transmise à **IP VGA**.
- Pour cela, il reçoit en entrée des informations provenant des autres modules du système.

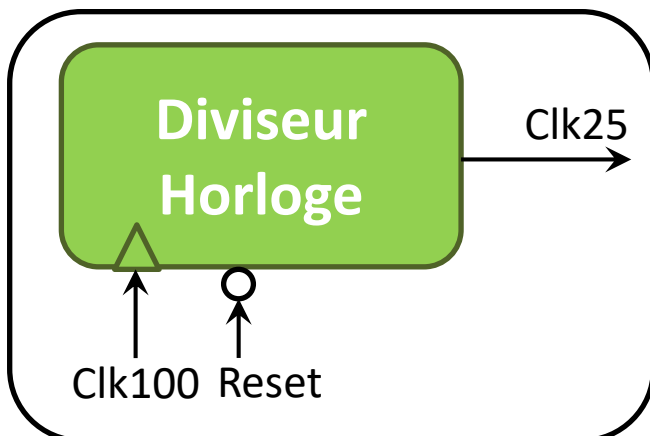


PREPARATION

1) Lire ce Manuel Utilisateur. Noter toutes vos questions si vous en avez pour les poser à l'intervenant de TP.

2) Codage du Diviseur d'horloge

- L'horloge de la carte **Nexys 4** est cadencée à 100 MHz.
 - Cependant, la fréquence de fonctionnement de la console **SU-EE100** doit être de 25 MHz.
 - Il faut donc pour cela diviser par 4 l'horloge de la carte.
- Cette fonction de génération d'horloge ne sera pas présente dans le code VHDL de la console qui vous sera fourni en début de projet. C'est donc à vous de l'ajouter à l'aide d'un module VHDL que vous allez décrire.
 - Proposer un schéma bloc du diviseur d'horloge.
 - Décrire cette architecture en VHDL. L'entité devra **OBLIGATOIREMENT** s'appeler **ClkDiv**, et **les ports d'entrée/sortie devront être les suivants (bien respecter les noms également)**



Entrées :

- **Clk100** : Horloge 100 MHz fournie par la carte
- **Reset** : Reset asynchrone (Actif à l'état bas)

Sortie :

- **Clk25** : Horloge de sortie à 25 MHz

Pour réaliser la vision d'horloge, il faut implémenter un compteur cadencé par l'horloge **Clk100**. Ce compteur va compter le nombre de périodes de **Clk100** qui correspondent à 1 cycle de l'horloge **Clk25**. Le niveau logique de la sortie **Clk25** sera alors fixé en fonction de la valeur du compteur.

Faire valider le schéma et le code par l'intervenant de TP.

Les codes source du TP ne vous seront pas donnés sans cette validation



