



PROJET EXPÉRIMENTAL DE CONCEPTION DE CIRCUIT NUMÉRIQUE

# JEU MORPION

CAHIER DES CHARGES

Sylvain **MARIEL**  
Thomas **MOREAU**

Promotion SEE3  
Mars 2013

Dans le cadre du module PR 209, nous devons mener à bien, un projet expérimental de conception de circuit numérique. Nous devons nous inspirer du processeur minimal 8 bits à quatre instructions que nous avons réalisé à partir du module EN 217.

Au final, le circuit implémenté devra réaliser des fonctions mettant en œuvre différentes entrées et sorties de contrôle de la carte et un ou plusieurs calculs sur les données.

## Définition du projet

L'application que nous avons choisi de réaliser est un jeu de type "Morpion". Le Morpion est un jeu de réflexion se pratiquant à deux joueurs au tour par tour et dont le but est de créer le premier un alignement sur une grille. Celle-ci se compose de 9 cases (3lignes et 3 colonnes) où chaque joueur vient y inscrire un symbole.



Figure 1. Grille de Morpion

Celui-ci sera intégralement réalisé sur la carte Digilent Nexys 3 et utilisera deux périphériques :

- Le port VGA, sur lequel sera relié un écran d'ordinateur, et servira de contrôle visuel.
- Les boutons poussoirs permettront aux joueurs d'interagir avec le système.

Les boutons poussoirs sont utilisés pour :

- Se déplacer dans la grille en sélectionnant une case.
- Valider la sélection de la case afin d'y déposer un symbole.

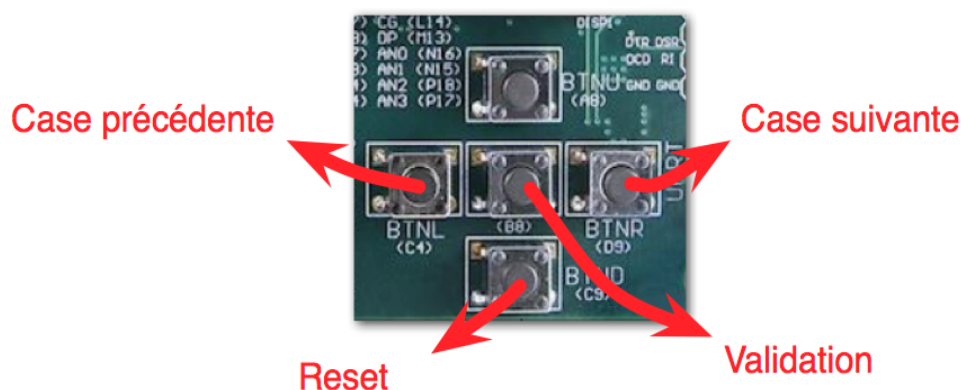


Figure 2. Action des différents boutons poussoirs

L'écran a pour but :

- Afficher la grille de jeu et visualiser le positionnement de curseur. L'utilisateur doit pouvoir observer la case qu'il sélectionne.
- Visualiser les cases validées en affichant les symboles placés par les joueurs.

D'après la documentation de la carte Digilent Nexys 3, le port VGA utilisé nous permet d'afficher sur une zone de 640 par 480 pixels. Ainsi, nous proposons la gestion de l'affichage minimal suivant :

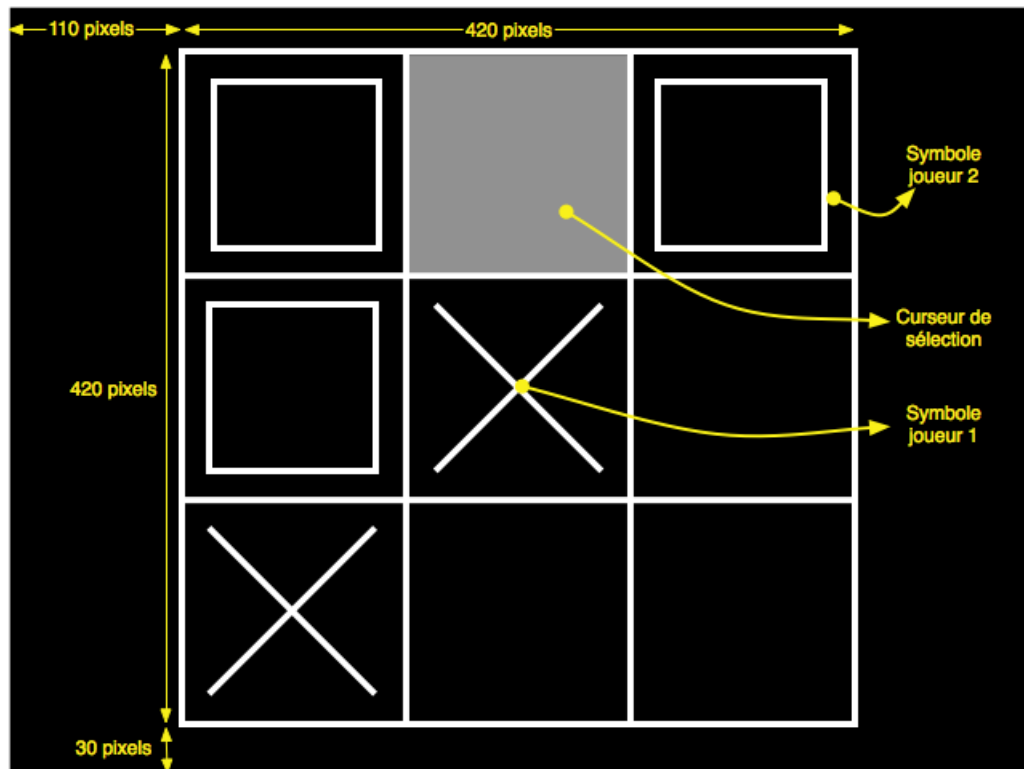


Figure 3. Affichage minimal attendu

## Rôle du processeur

Dans ce projet, le processeur aura un rôle fondamental. Il permettra tout d'abord d'interpréter les différentes actions de l'utilisateur afin de séquencer le jeu. En fonction de celles-ci, il pourra par la suite commander l'affichage.

Pour cela, il doit constamment communiquer avec les périphériques partageant les bus de données et d'adresses, et donc pleinement jouer son rôle de "maître du bus".

Afin d'y parvenir, un programme utilisant les 4 types d'instructions devra être rédigé. Par exemple, l'opération d'addition nous sera particulièrement utile pour incrémenter ou décrémenter la position du curseur. L'opération NOR quant à elle nous sera par exemple très utile pour réinitialiser la valeur du registre accumulateur.

Sa description VHDL sera similaire à celle réalisée lors du cours EN 217. Nous nous contenterons de réutiliser l'UC et l'UT au sein d'un module VHDL "Processeur".

## Le système et son architecture

L'architecture de notre système se décompose de la façon suivante :

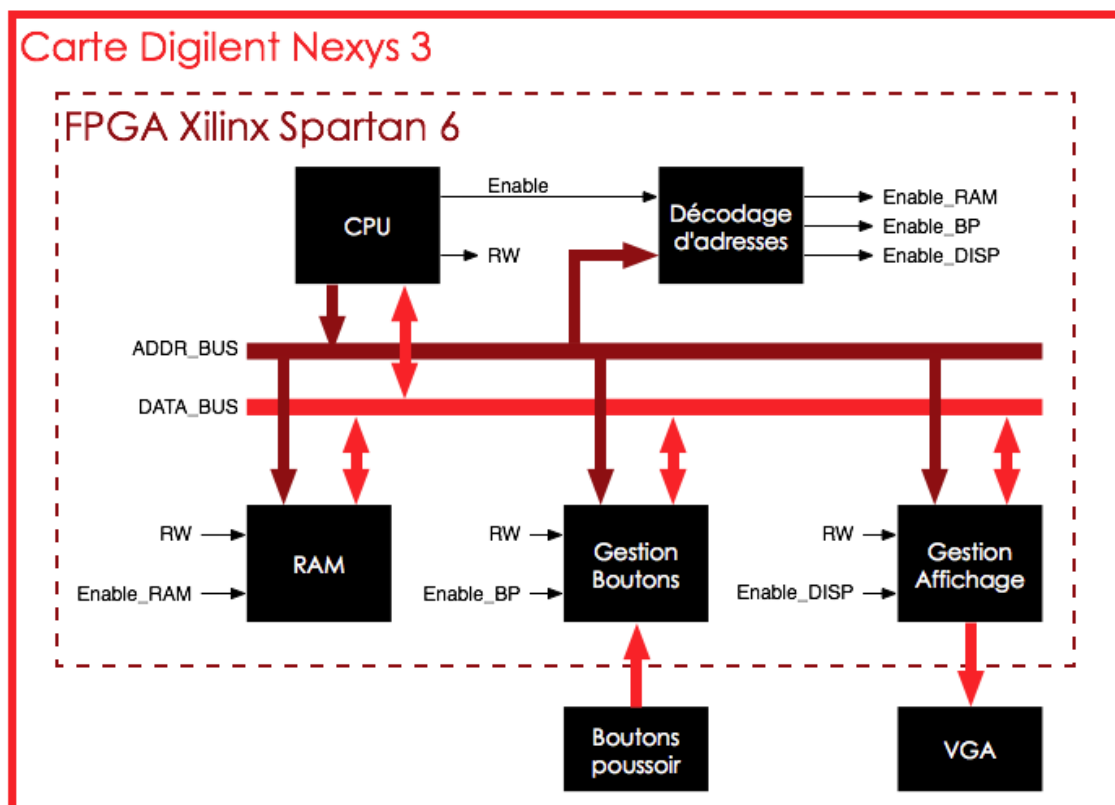


Figure 4. Schéma système

Elle contient trois types de bloc :

- Le CPU 8 bits
- Les périphériques
- Le décodage d'adresse
- Les contrôles

Lorsque le CPU veut communiquer avec un périphérique, il l'adresse par le biais du **bus d'adresse**. Les données transitent ensuite par le **bus de données**, que ce soit une lecture ou une écriture. Le CPU est donc le "maître" du bus et échange avec les trois périphériques suivants :

- La gestion des boutons poussoirs. Son rôle est d'observer les différentes actions des joueurs et les communiquer au CPU.
- La gestion de l'affichage qui permet d'illustrer la zone de jeu ainsi que les actions de l'utilisateur par le biais d'une liaison VGA.
- La RAM qui contiendra les différentes instructions ainsi que les données nécessaires.

Chacun d'entre eux dispose d'une ou plusieurs adresses parmi les  $2^{16}$  disponibles.

Le bloc décodage d'adresse détient le rôle d'aiguilleur du système. Son principe est de lire l'adresse émise par le CPU et de générer un signal de type "enable" au périphérique concerné, et uniquement celui-ci. Le tout en concordance avec le signal "enable" émis par le CPU.

## Évolutions possibles

Dans l'éventualité on nous arriverions à la fin du projet plus tôt que prévu, nous avons listé plusieurs évolutions possibles du système que nous pourrions réaliser :

- Gestion de fin de partie, détermination du résultat (match nul ou victoire d'un joueur)
- Utilisation d'autres contrôles visuels (7 segments, LEDs)
  - Les 7 segments afficheraient le joueur dont c'est le tour de jouer
  - Les LEDs définiraient l'état de la partie (en cours ou terminée)
- Ajout d'informations à l'écran tel que l'état de la partie (symbole du joueur, tour du joueur, fin de partie)
- Amélioration esthétique de l'affichage : utilisation de couleurs et de formes prédéfinies