

# Laboratoire CSF

## Jeu du morpion

semestre printemps 2016 - 2017

## Objectifs pédagogiques

Ce laboratoire a pour but de réaliser un jeu du morpion où chaque joueur est sur une plateforme différente. L'implémentation est faite dans la FPGA, avec une communication SPI entre les deux systèmes.

## Cahier des charges

Le jeu de morpion consiste en une grille de  $3 \times 3$  cases, où chaque joueur à son tour place un jeton de sa couleur (ou forme). Celui qui arrive à aligner trois jetons gagne.

Le système à implémenter devra établir une communication entre deux cartes DE1SoC, chaque joueur disposant d'une carte. La communication pourra reprendre le composant SPI développé lors du labo précédent.

L'affichage se fera sur un écran connecté sur le port VGA de la carte. Toute la gestion de l'affichage vous est fournie, votre composant offrant en sortie l'état du jeu.

Afin de rendre la mise au point la plus aisée possible, il est suggéré d'utiliser le bus SPI pour transmettre l'information qu'une touche a été appuyée, et qu'ensuite chaque système gère l'état du jeu.

## Entrées/sorties

```
entity tictactoe_top is
port (
  clk_i          : in  std_logic; -- 27Mhz
  rst_i          : in  std_logic; -- KEY3
  -- Game controls
  keys_i         : in  std_logic_vector( 2 downto 0);
  switches_i     : in  std_logic_vector( 7 downto 0);
  -- Game state
  game_state_o   : out game_state_t;
  -- Master interface
  master_en_i    : in  std_logic;
  sclk_o         : out std_logic;
  mosi_o         : out std_logic;
  miso_i         : in  std_logic;
  n_ss_o         : out std_logic;
  -- Slave interface
  sclk_i         : in  std_logic;
  mosi_i         : in  std_logic;
  miso_o         : out std_logic;
  n_ss_i         : in  std_logic
);
end entity;
```

Les deux interfaces SPI permettent de mettre en place la communication entre les deux cartes. Le port d'entrée `master_en_i` indique que le système est master (1) ou slave (0).

L'état du jeu utilise une structure déclarée dans le paquetage `tictactoe_config_pkg.vhd`.

```

type game_state_t is record
  player1_reg : std_logic_vector(8 downto 0); -- player 1
  player2_reg : std_logic_vector(8 downto 0); -- player 1
  sel_reg      : std_logic_vector(8 downto 0); -- pointer/selector
  -- bits encoding :
  --   0 | 1 | 2
  -- -----
  --   3 | 4 | 5
  -- -----
  --   6 | 7 | 8

  curr_player : std_logic;
  -- '0': player 1
  -- '1': player 2

  curr_player_win : std_logic;
  draw_match      : std_logic;

  win_line : std_logic_vector(7 downto 0);
  -- bit 0 : mark(0,1,2) -> row 1
  -- bit 1 : mark(3,4,5) -> row 2
  -- bit 2 : mark(6,7,8) -> row 3
  -- bit 3 : mark(0,3,6) -> col 1
  -- bit 4 : mark(1,4,7) -> col 2
  -- bit 5 : mark(2,5,8) -> col 3
  -- bit 6 : mark(0,4,8) -> diag \
  -- bit 7 : mark(2,4,6) -> diag /
end record game_state_t;

```

L'état du jeu (visuel) comprend les signaux suivants :

player1_reg	Pour chaque position, le symbole du joueur est affiché si le bit est à '1'.
player2_reg	Idem.
sel_reg	Pour chaque position, le symbole du curseur est affiché si le bit est à '1'.
curr_player	Indique le joueur à qui c'est le tour de jouer.
curr_player_win	'1' si le joueur courant a gagné.
draw_match	'1' si match nul.
win_line	Indique où placer la ligne qui affiche la combinaison gagnante.

Le bouton poussoir KEY\_3 est utilisé comme reset, vous avez donc les trois boutons restant pour contrôler le jeu. Vous pouvez aussi utiliser les switches 7 à 0 si vous le désirez. Une idée est d'utiliser deux boutons pour déplacer un curseur et un pour valider le choix, mais ceci est libre.

## Travail à réaliser

Vous devez :

1. Réaliser le composant `tictactoe_top`
2. Réaliser un banc de test pour valider son bon fonctionnement
  - Vous disposez d'un squelette de banc de test pour un `tictactoe_top`
  - Vous disposez également d'un squelette de banc de test pour deux `tictactoe_top` connectés
3. Tester le système sur carte

Pour le test sur carte, un projet vous est fourni. Il permet de faire communiquer deux cartes en les connectant grâce au connecteur fourni. Le switch S9 permet de sélectionner si la carte est maître ou esclave.

## Travail à rendre

Une archive contenant le projet complet. Un mini-rapport expliquant votre analyse, décomposition et conception du système.