

TD 3

Module : Synthèse en VHDL des systèmes embarqués - Section ICE3

Enseignant : Chiheb Ameur ABID

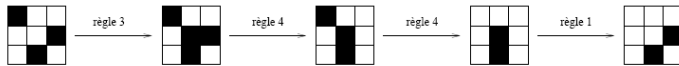
Année universitaire : 2023-2024

Exercice 1.

Le célèbre « jeu de la vie » de Conway est une simulation impliquant un ensemble de « cellules » sur une grille cartésienne. À partir d'une configuration initiale, l'état de chaque cellule évolue en fonction de l'état des cellules voisines, ce qui en fait un automate cellulaire. L'état de toutes les cellules change en même temps, ce qui correspond à une génération. Une cellule est soit « vivante » (état 1) ou « morte » (état 0). L'automate est entièrement défini par les règles simples suivantes qui sont évaluées pour passer d'une génération à l'autre:

1. Si une cellule vivante a moins de 2 voisins, elle meurt d'isolement.
2. Si une cellule vivante a plus de 3 voisins, elle meurt d'étouffement.
3. Si une cellule morte a exactement 3 voisins, elle « naît », i.e. elle devient vivante.
4. Si une cellule vivante a 2 ou 3 voisins, elle conserve son état.

On considère qu'une cellule possède 8 voisins sur grille cartésienne. L'illustration suivante montre l'évolution d'une cellule (cellule au centre) en fonction de ses 8 voisins au fil de 4 générations. On voit que l'état des voisins change également, en fonction de leurs 8 voisins respectifs.



On s'intéresse dans ce qui suit à la conception d'une machine à états qui réalise une cellule dans le jeu de la vie. Assumez qu'il y a une génération par coup d'horloge et utilisez l'entité suivante:

```
entity cellule is port(
    nb_voisins: in std_logic_vector(3 downto 0);
    etatcellule: out std_logic;
    clk, reset: in std_logic);
end entity;
```

On assume également que l'entrée `nb_voisins` donne le nombre de voisins vivants.

- 1) Donner la machine à états décrivant ce circuit, puis traduire la machine à état en une description VHDL.
- 2) Donner le chronogramme des variations de tous les signaux y compris le signal indiquant l'état courant pour l'exemple donné.

Exercice 2.

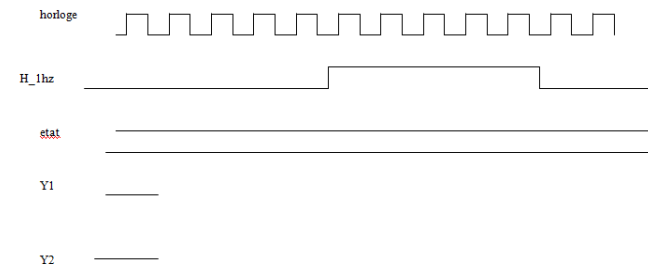
Soit la description VHDL ci-dessous :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity monocoup is Port ( horloge : in std_logic;
    h_1hz : in std_logic;
    y1 : out std_logic;
    y2 : out std_logic);
end monocoup;
architecture Behavioral of monocoup is
    signal etat : std_logic_vector(1 downto 0);
begin
    process (horloge)
```

```
begin
    if (horloge'event and horloge='1') then
        case etat is
            when "00" =>
                if (h_1hz='0') then etat <= "01";
                end if;
            when "01" =>
                if (h_1hz='1') then etat <= "10";
                end if;
            when "10" => etat <= "11";
            when others => etat <= "00";
        end case;
    end if;
end process;
y1 <= '1' when etat = "10" else '0';
y2 <= '1' when etat = "11" else '0';
end Behavioral;
```

- 1) Reconstituer la machine d'états correspondante à cette description VHDL. Quel est le type de la machine à états ? Justifier la réponse.

- 2) Compléter l'évolution des différents signaux pour le chronogramme ci-après.



- 3) Modifier la description VHDL pour prévoir une initialisation asynchrone à l'état 0.

Exercice 3.

Dans un laboratoire de recherche, une expérience nécessite le maintien d'une salle à une température fixe. Le refroidissement ne doit être en aucun cas interrompu.

Pour ce faire, on utilise deux bouteilles de gaz réfrigérant B1 et B2 utilisées en alternance. Dès qu'une bouteille est vide, le système doit commuter immédiatement sur l'autre et vice-versa. Cela laisse ainsi le temps de changer la bouteille vide pendant que l'autre est utilisée. Si par mégarde, les deux bouteilles venaient à être vides, une alarme se déclenche, afin d'alerter tout le monde sur l'urgence de la situation.

Ce système de surveillance est réalisé à l'aide d'un module admettant les entrées/sorties suivantes :

- H : Horloge du système
- Reset : Réinitialisation asynchrone du système
- PL1, PL2 : Informations issues de capteurs de pression positionnées sur les bouteilles. Ces signaux passent à '1' si la bouteille est vide
- EV1, EV2 : Actionneurs reliés aux électrovannes des bouteilles. La bouteille i est ouverte et délivre son gaz si EVi = '1'
- ALARM : Commande de l'alarme du système, active au niveau haut.

Les contraintes du cahier des charges sont les suivantes :

- À l'état initial, les deux bouteilles sont fermées. On cherchera à utiliser prioritairement la bouteille 1.
- Une bouteille de gaz est utilisée jusqu'à ce qu'elle soit vide. À ce moment uniquement, on commute sur l'autre bouteille.
- L'alarme ne peut être désactivée que par un reset asynchrone du système.
- Après le déclenchement de l'alarme, le système doit reprendre le refroidissement, sans désactiver l'alarme, dès qu'une bouteille pleine soit disponible.

1) Donner la représentation en bloc de fonction du module.

2) Donner la machine à états de Moore décrivant le comportement du module.

3) Indiquer le nombre de bistables nécessaires à cette réalisation en argumentant la réponse.

4) Traduire la machine à états en code VHDL.