

Numéro Anonymat: - - - - -

L3 SPI - VHDL - ESI

Yoan MARIN - Denis PELLION

Session 2 - Année 2018-2019 - Durée 2h

Une feuille (A4) recto manuscrite autorisée - calculatrice interdite

Les réponses doivent être justifiées par une rédaction précise et détaillée.

1 FPGA - généralités (4 points)

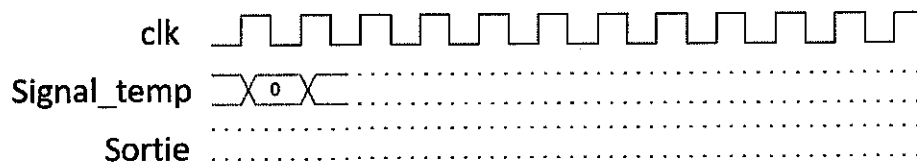
1. Quelles sont les étapes nécessaires, dans ISE, à la génération d'un fichier de configuration pour FPGA " Bitstream"?
2. Donner les définitions des mots clés "generate" et "generic"? (Pour chaque définition vous donnerez un exemple)

2 Compréhension de code (2 points)

Le but de cet exercice est de comprendre la fonction que réalise le composant décrit ci-dessous. Après avoir regardé et compris le code ci-dessous, vous devez **remplir le chronogramme sur cette feuille (à rendre)**. Vous donnerez les valeurs des signaux *Sortie* et *Signal_temp* en fonction des cycles d'horloge *clk*.

```
entity Exam is
port(
  clk      : in std_logic;
  Sortie   : out std_logic);
end entity;

Architecture Exam_arch of Exam is
signal signal_temp: std_logic_vector(3 downto 0) :=(others=>'0');
begin
  process(clk)
  begin
    if (clk'event and clk='1')then
      signal_temp <= signal_temp +1;
      if signal_temp="0011" then
        Sortie    <= '1';
        signal_temp <= (others=>'0');
      else
        Sortie    <= '0';
      end if;
    end if;
  end process;
end Exam_arch;
```



3 Machine à états (8 points)

Comme il a été vu en cours vous allez décrire une machine d'états, et plus particulièrement une machine de Moore (les sorties ne dépendent que le l'état).

Cette fois-ci vous décrierez le fonctionnement d'une porte avec Digicode et système d'alarme.

Description du fonctionnement

Pour ouvrir la porte il est nécessaire de faire la séquence suivante: # , 2 , 5 , 8 , 0 , *; où # indique le fait que vous allez entrer un code, et * le valide.

Si la séquence est bonne alors la porte s'ouvre (**PO=1**), et l'alarme reste désactivée (**AL=0**). Cependant si vous faites une erreur dans la séquence, la machine revient à l'état initial et on incrémente un signal nommé **Erreur**. Lorsque **Erreur=3** l'alarme s'active (**AL=1**), la machine reste bloquée dans un état de *maintenance* et la porte reste fermée (**PO=0**). Pour revenir à l'état initiale et mettre les signaux **Erreur** et **AL** à 0, il faut que l'entrée signalant une maintenance en cours soit active **Maint=1** (la porte reste fermée **PO=0**).

Cette fois-ci vous décrierez une machine à états utilisées pour la gestion de l'ouverture d'une porte avec un digicode.

1. Vous donnerez tout d'abord le graphe d'états-transitions.
2. Vous décrierez en VHDL le comportement de cette machine à état. (**Attention à gérer tous les signaux et états cités dans la consigne**)

4 Diviseur générique (6 points)

Vous allez décrire en VHDL comportemental un diviseur **32 bits**. La valeur de la division **Div** est générique. La division sera synchrone sur le front *descendant* de l'horloge **Clk**. Le signal **RAZ** est lui aussi synchrone (*de la même manière que la division*). Les entrées et sorties de ce composant sont explicitées dans la figure 1. Vous devez impérativement respecter le nom donné aux variables. La division permet de faire seulement des divisions en puissance de 2 (*voir equation 1*).

$$DataOut = \frac{DataIn}{2^{Div}} \quad (1)$$

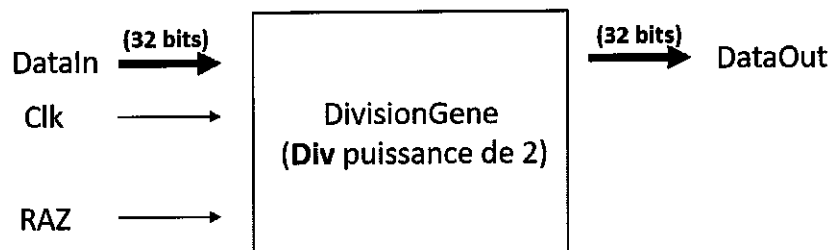


Figure 1: Description des entrées et sortie du composant **DivisionGene** à décrire, **Veillez respecter les noms des entrées/sorties**

Il n'est pas possible d'utiliser l'opérateur mathématique "/" pour la division. De plus il vous est conseillé d'utiliser la concaténation et la sélection de bits