

TD 5

Questions de cours

Répondre par vraie ou faux

1. L'instruction conditionnelle (if...then...endif) est utilisée seulement à l'intérieur d'un process.
2. Tous les processus dans une architecture sont activés tous le temps lorsque l'architecture est activée.

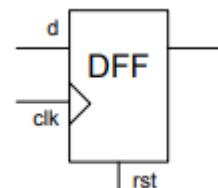
Choisir la (les) bonne(s) réponse(s) :

1. En VHDL, les instructions concurrentes ...
 - ne sont déclarés que dans l'architecture.
 - ne sont déclarés que dans les sous-programmes.
 - sont exécutés d'une manière séquentielle.
2. En VHDL, les instructions séquentielles
 - sont définis dans l'architecture
 - sont définis dans le processus
3. Un processus contient
 - toujours une instruction WAIT.
 - toujours une liste de sensibilité.
 - obligatoirement soit une ou plusieurs instructions WAIT, soit une liste de sensibilité, soit les deux.
 - toujours une liste de sensibilité et éventuellement une ou plusieurs instructions WAIT.
 - obligatoirement soit une instruction WAIT, soit une liste de sensibilité, mais jamais les deux.

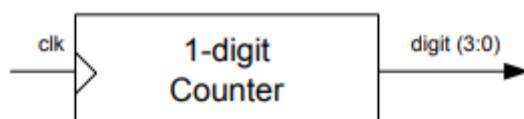
Exercice 1

Ecrire un code en VHDL pour les circuits ci-dessous.

1. Bascule D avec remise à zéro asynchrone.
 - Avec un PROCESS ayant une liste de sensibilité.
 - Avec un PROCESS ayant un WAIT.
 - Avec un PROCESS incluant un CASE
2. Compteur décimal 1 chiffre



Programmer un compteur de 0 à 9 qui est incrémenté (modulo 10) à chaque front montant d'horloge.



Exercice 2

Le code VHDL comportemental de la figure 1 décrit un compteur de 3 bits avec des entrées set et reset asynchrone.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity counterreset is
    Port ( CLOCK : in STD_LOGIC;
            reset : in STD_LOGIC;
            init: in STD_LOGIC;
            COUNT_OUT : OUT STD_LOGIC_VECTOR (3 downto 0));
end counterreset;

architecture Behavioral of counterreset is
    signal count_int : STD_LOGIC_VECTOR (3 downto 0):="0000";

begin

    process (CLOCK,reset,init)
        begin
            if reset='1' then
                count_int <= "0000" ;
            elsif(CLOCK='1' and CLOCK'event) then
                if init ='1' then
                    count_int <= "1001" ;
                else
                    count_int <= count_int+1 ;
                end if;
            end if;
        end process;

    COUNT_OUT<=count_int;

end Behavioral;
```

Figure 1:Code VHDL

1. Donner le modulo de ce compteur.
2. Quelle est le rôle de l'entrée init dans ce code VHDL ?
3. Commenter la ligne du code VHDL : $\text{CLOCK}='1'$ and $\text{CLOCK}'\text{event}$
4. Quelles sont les entrées synchrones et asynchrones?
5. Pour le *process* étudié, donner la liste de sensibilité.
6. Modifier le code VHDL afin d'obtenir un compteur modulo 10.

Exercice 3

Ecrire un processus permettant de générer un train d'impulsions toutes les microsecondes : pas d'impulsion, puis une impulsions de 100 ns puis une impulsions de 200ns, jusqu'à 1 microseconde comme le montre la figure ci-dessous.



Exercice 4

Tracer la forme du signal généré par le code VHDL suivant :

```
Process
Begin
  Clk <= '0';
  Wait for (now + 10 ns);
  Clk <= '1';
  Wait for (now + 10 ns);
End process;
```

Exercice 5

Ecrivez en VHDL un Processus qui peut générer le signal de la figure ci-dessous.

