INF3500 : Conception et réalisation de systèmes numériques

Examen final

Avril 2011

Durée: 2h30.

Pondération: 40%.

Documentation: Une feuille recto verso 8.5”×11” ou A4 permise.

Calculatrice: Programmable permise.

Directives particulières:

- Ordinateurs interdits.
- Répondre à toutes les questions, la valeur de chaque question est indiquée.
- Répondre dans le cahier fourni.
- Ne pas remettre le questionnaire, sauf la dernière page qui doit être annotée et remise.
- Ne posez pas de questions. En cas de doute sur le sens d’une question, énoncez clairement toute supposition que vous faites.

Question 1. (6 points)

Une porte universelle peut réaliser toutes les fonctions logiques possibles de ses entrées. La fonction désirée est spécifiée par un port de contrôle spécial. Pour une porte universelle à deux entrées *x* et *y*, il y a 16 possibilités de fonctions logiques F_{op}, où $0 \leq op \leq 15$, tel que montré dans la table de vérité combinée suivante. On observe que les fonctions F_0 et F_{15} sont des constantes, alors que les fonctions F_3, F_5, F_{10} et F_{12} se réduisent à des fonctions à une seule variable.

x	y	F_o	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Donnez une architecture pour l'entité suivante en VHDL synthétisable, afin de réaliser une porte universelle à deux entrées.

```
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.all;

entity porteUniverselle2 is
    port (
        x, y : in std_logic;
        op : in unsigned(3 downto 0); -- opération à effectuer
        F : out std_logic
    );
end porteUniverselle2;
```

Solution :

```
-- solution 'simple'
architecture arch of porteUniverselle2 is
begin
  process (x, y, op)
  begin
    case to_integer(op) is
      when 0 => F <= '0';
      when 1 => F <= x and y;
      when 2 => F <= x and not(y);
      when 3 => F <= x;
      when 4 => F <= not(x) and y;
      when 5 => F <= y;
      when 6 => F <= x xor y;
      when 7 => F <= x or y;
      when 8 => F <= not(x or y);
      when 9 => F <= not(x xor y);
      when 10 => F <= not(y);
      when 11 => F <= not(not(x) and y);
      when 12 => F <= not(x);
      when 13 => F <= not(x and not(y));
      when 14 => F <= not(x and y);
      when 15 => F <= '1';
      when others => F <= 'X';
    end case;
  end process;
end arch;
```

```
-- solution la plus simple
architecture arch2 of porteUniverselle2 is
begin
F <= op(to_integer(not(unsigned(x & y))));
end arch2;
```

Question 2. (10 points)

Faites la conception d'un processeur pour un télémètre laser. Le télémètre a un bouton pour déclencher la prise de mesure. Quand le bouton est pressé, une impulsion lumineuse est générée et un chronomètre est activé. L'impulsion lumineuse se propage dans l'air, frappe la cible et revient vers un détecteur. Quand l'écho de l'impulsion lumineuse est perçu par le détecteur, le chronomètre est arrêté et la distance peut être calculée en tenant compte de la vitesse de propagation de la lumière dans l'air. Soit T le temps mesuré par le chronomètre, en secondes, alors la distance D est donnée par $D = T \times c / 2$, où $c = 3 \times 10^8$ m/s.

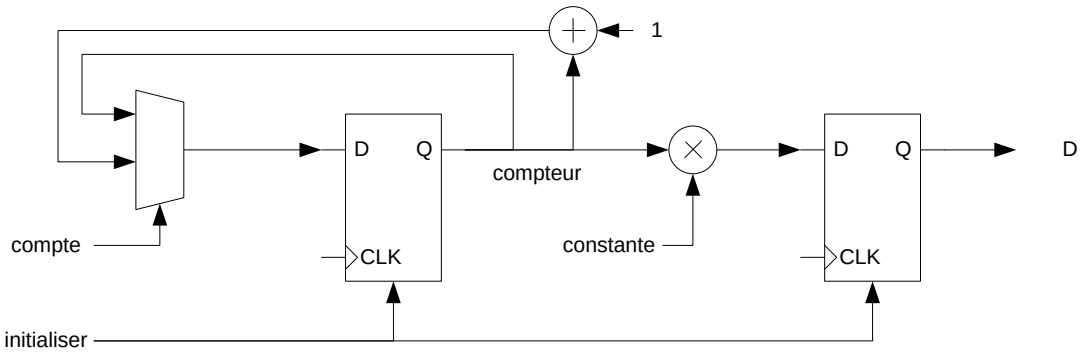
Le processeur a deux entrées : le bouton et un signal provenant du détecteur indiquant qu'une impulsion lumineuse a été reçue. Il a deux sorties : un signal vers le laser pour déclencher une impulsion lumineuse et un autre signal indiquant la distance mesurée.

a. Donnez un diagramme représentant le chemin des données du processeur.

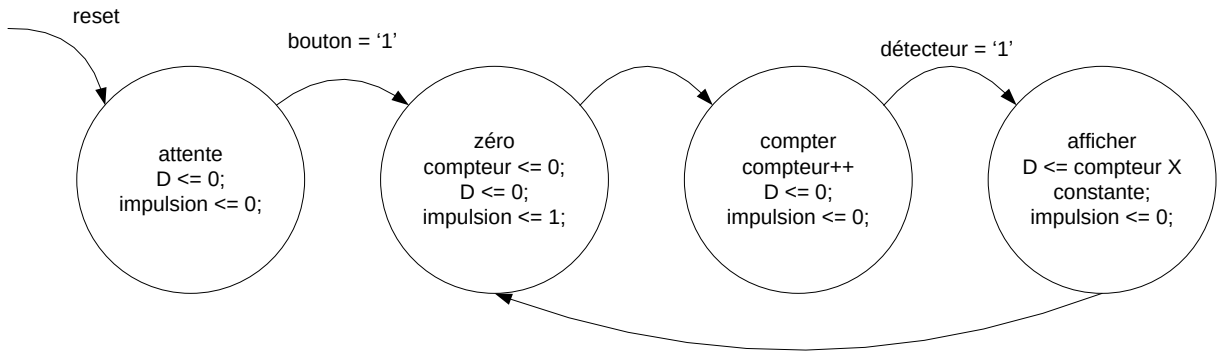
b. Donnez le diagramme d'états de l'unité de contrôle du processeur.

Solution :

a.



b.



Il faudrait ajouter une valeur maximale au compteur. Quand cette valeur serait atteinte, indiquant qu'un écho lumineux n'a pas été reçu, le système afficherait un message d'erreur.

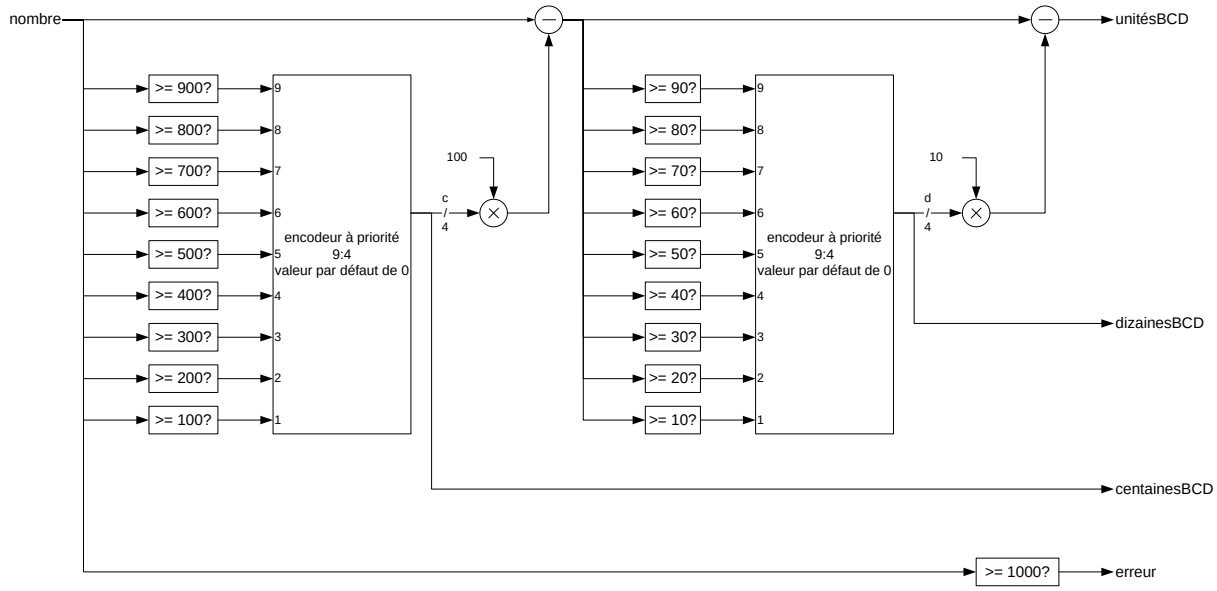
Question 3. (6 points)

Considérez le code VHDL suivant. Le circuit accepte en entrée un nombre non signé encodé sur 10 bits, et il a trois sorties. Ces sorties donnent le nombre de centaines, de dizaines et d'unités (de 0 à 9 inclusivement, encodées sur 4 bits) du nombre. La sortie erreur est activée si le nombre est égal ou supérieur à 1000.

Donnez un diagramme correspondant composé de modules combinatoires (multiplexeurs, décodeurs, etc.) et d'unités fonctionnelles (additionneurs, multiplicateurs, comparateurs, etc.) tel qu'il pourrait être généré par un processus de synthèse.

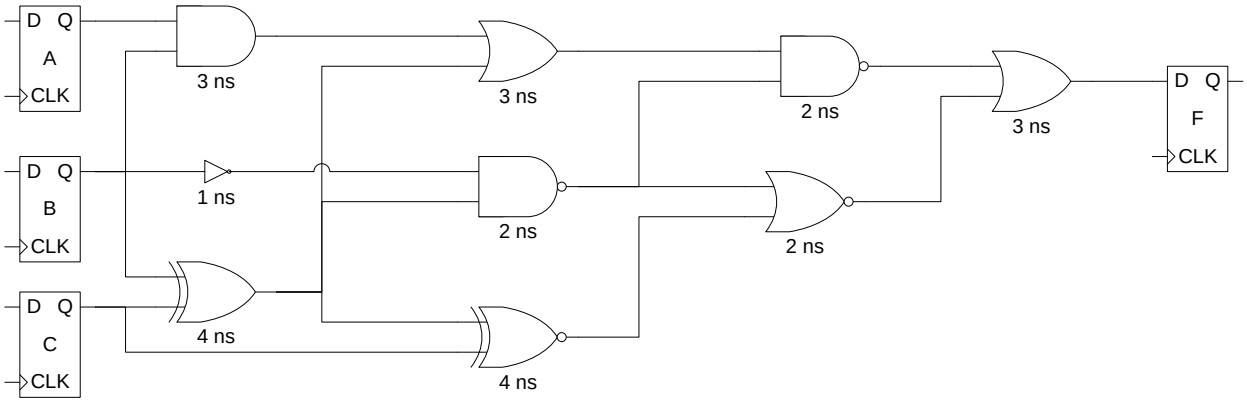
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity unsigned2dec is
    port(
        nombre : in unsigned(9 downto 0);
        centainesBCD, dizainesBCD, unitesBCD : out unsigned(3 downto 0);
        erreur : out std_logic
    );
end unsigned2dec;

architecture arch of unsigned2dec is
begin

    erreur <= '1' when nombre >= 1000 else '0';

    process(nombre)
    variable n, c, d, u : natural := 0;
    begin

        n := to_integer(nombre);

        for centaines in 9 downto 1 loop
            c := 0;
            if n >= centaines * 100 then
                c := centaines;
                exit;
            end if;
        end loop;

        n := n - c * 100;

        for dizaines in 9 downto 1 loop
            d := 0;
            if n >= dizaines * 10 then
                d := dizaines;
                exit;
            end if;
        end loop;

        u := n - d * 10;

        centainesBCD <= to_unsigned(c, 4);
        dizainesBCD <= to_unsigned(d, 4);
        unitesBCD <= to_unsigned(u, 4);

    end process;

end arch;
```

Solution :


nombre
unitésBCD
>= 900?
>= 800?
>= 700?
>= 600?
>= 500?
>= 400?
>= 300?
>= 200?
>= 100?
encodeur à priorité 9:4
valeur par défaut de 0
100
c
/
4
>= 90?
>= 80?
>= 70?
>= 60?
>= 50?
>= 40?
>= 30?
>= 20?
>= 10?
encodeur à priorité 9:4
valeur par défaut de 0
10
d
/
4
dizainesBCD
centainesBCD
>= 1000?
erreur

Question 4. (6 points)

Considérez le circuit suivant. Les délais des portes logiques combinatoires sont indiqués. Les bascules ont un temps de propagation de 2 ns, un temps de préparation de 1 ns, et un temps de maintien de 0.5 ns. Chaque fil du circuit a un délai de propagation de 0.1 ns.



a. Supposez que toutes les bascules sont alimentées par le même signal d'horloge. Donnez le chemin critique en indiquant les composantes sur ce chemin, et donnez la fréquence maximale d'horloge.

b. On veut maximiser la fréquence d'horloge du circuit en insérant des registres de pipeline. Les registres ajoutés ont les mêmes caractéristiques que les registres actuels du circuit, et chaque fil ajouté a un délai de propagation de 0.1 ns. Il n'y a pas de déphasage d'horloge. En supposant que vous pouvez ajouter autant de registres que vous le voulez, quelle est la fréquence maximale d'opération du circuit? Expliquez complètement votre réponse.

c. Le circuit initial sans pipeline a été disposé de façon inappropriée sur une puce, et la bascule A reçoit un signal d'horloge qui est 3 ns en retard sur les signaux d'horloge des autres bascules. Quelle est la fréquence maximale d'horloge? Expliquez complètement votre réponse.

Solution :

a. Bascule C, OUX, ETX, NOU, OU, Bascule F. Délai : 2 + 0.1 + 4 + 0.1 + 4 + 0.1 + 2 + 0.1 + 3 + 0.1 = 15.5 ns, + tsu(1 ns) = 16.5 ns. Fmax = 60.6 MHz.

b. On ajoute une bascule après chaque porte logique. Il en faut une aussi sur le fil entre la bascule C et l'entrée et la porte ETX. Le chemin critique débute dans une bascule, passe à travers la porte OUX ou la porte ETX, et va à une autre bascule. Le délai est 2 + 0.1 + 4 + 0.1 + tsu(1ns) = 7.2 ns, fmax = 138.8 MHz.

c. Le chemin critique devient alors bascule A, ET, OU, NON-ET, OU, bascule F. Délai : tcs(3 ns) + 2 + 0.1 + 3 + 0.1 + 3 + 0.1 + 2 + 0.1 + 3 + 0.1 + tsu(1 ns) = 17.5 ns. Fmax = 57.1 MHz.

Question 5. (6 points)

Réponses brèves.

a. Considérez le code VHDL suivant. Quelle valeur prendra le port F lors de la simulation? Pourquoi?

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity exempleSignaux is
    port (F : out std_logic);
end exempleSignaux;
architecture arch1 of exempleSignaux is
begin
    F <= '1';
    F <= '0';
end arch1;
```

b. Considérez le code VHDL suivant. Quelle valeur prendra le port F lors de la simulation si A = '1'? Pourquoi?

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity exempleSignaux is
    port (A : in std_logic; F : out std_logic);
end exempleSignaux;
architecture arch2 of exempleSignaux is
begin
    process(A)
    begin
        F <= A;
        F <= not(A);
    end process;
end arch2;
```

c. En VHDL, les types `STD_LOGIC_VECTOR`, `UNSIGNED` et `SIGNED` sont très utilisés pour la simulation et la synthèse de système numériques. Les trois types ont exactement la même définition :

```
type STD_LOGIC_VECTOR is array ( NATURAL range <>) of STD_LOGIC;
type UNSIGNED is array ( NATURAL range <>) of STD_LOGIC;
type SIGNED is array ( NATURAL range <>) of STD_LOGIC;
```

Expliquez la nécessité d'avoir ces trois types et ce qui rend l'utilisation de chacun distincte des autres.

d. Il y a quatre considérations fondamentales pour l'implémentation d'un système numérique, dont la taille du système. Énumérez les trois autres.

e. Nommez et décrivez brièvement une des trois technologies de programmation les plus populaires pour FPGA.

f. Pour le FPGA XC2VP30-FF896 utilisé dans les laboratoires, il y a deux façons d'implémenter de la mémoire RAM : la mémoire RAM distribuée (Distributed RAM) et la mémoire RAM bloc (Block RAM). Décrivez brièvement ces deux ressources et expliquez leurs différences.

Solutions :

a. Le port F prendra la valeur ‘X’ parce qu’il a deux sources de deux valeurs différentes.

b. Le port F prendra la valeur ‘0’ parce que dans le processus c’est la dernière assignation à F qui sera gardée.

c. Les trois types ont des utilisations différentes. Le type STD_LOGIC_VECTOR est utilisé pour les valeurs logiques, les deux autres pour des valeurs numériques. L’utilisation de chaque type est distincte à cause des fonctions, procédures et opérateurs qui sont définis et/ou surchargés pour chaque type.

d. Taux de traitement, puissance consommée et précision des calculs.

e. SRAM, antifusibles et Flash – voir les notes de cours, section 3.7.7.

f. La mémoire bloc est composée de modules spécifiques intercalés à travers les blocs de logique configurable du FPGA. La mémoire distribuée utilise les ressources internes de blocs de logique configurable.

Question 6. (6 points)

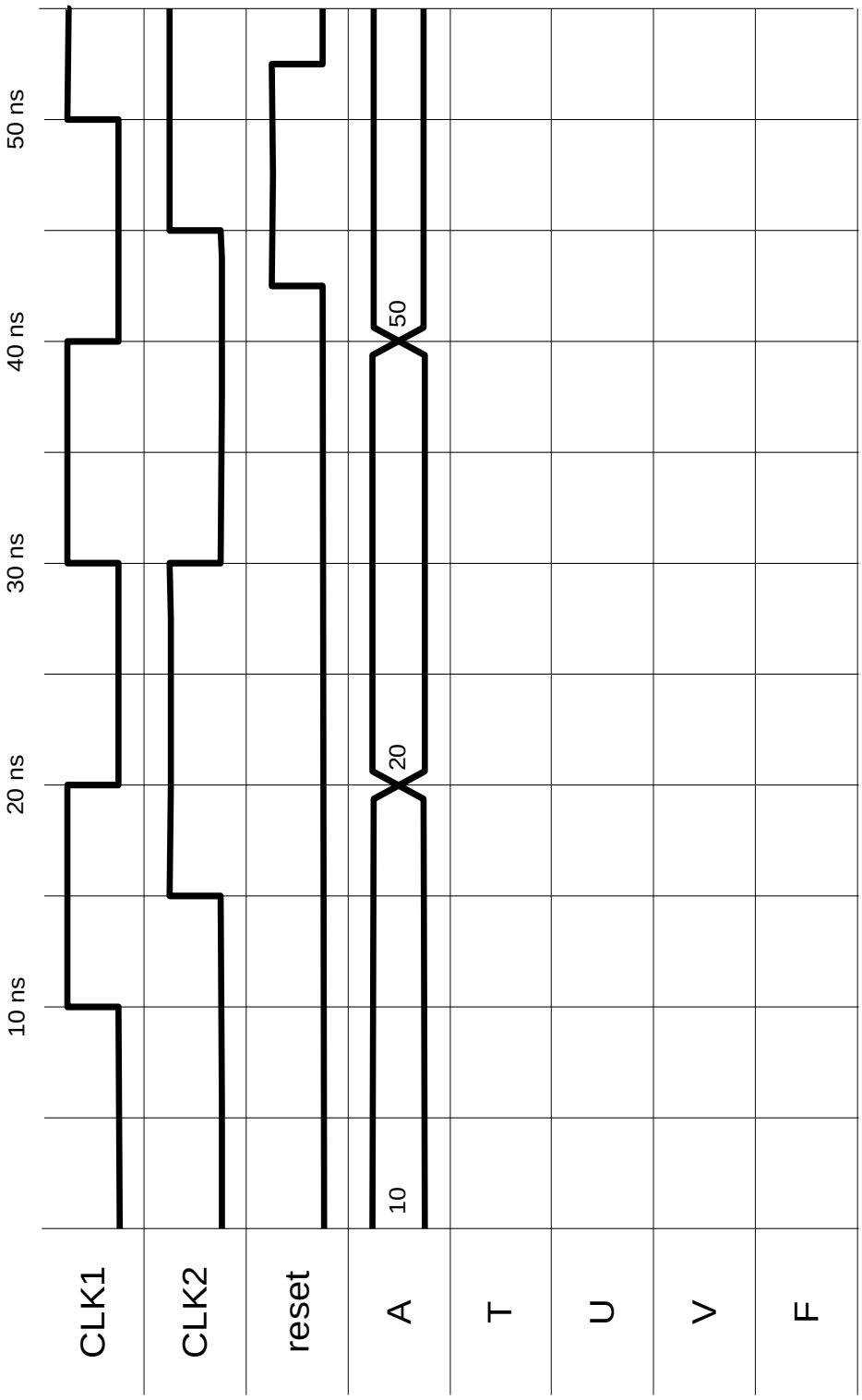
Considérez l'extrait de code VHDL suivant et les valeurs des signaux CLK1, CLK2, reset et A montrés sur le chronogramme de la page suivante.

Complétez le chronogramme pour les signaux et variables T, U, V et F. Remettez la page avec votre cahier d'examen.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity VHDLEstMonAmi is
   port (
      clk1, clk2, reset : in std_logic;
      A: in integer;
      F : out integer
   );
end VHDLEstMonAmi;

architecture jaimeVHDL of VHDLEstMonAmi is
signal T, U : integer := -5;
begin

   process (clk1)
   variable V : integer := 3;
   begin
      if reset = '1' then
         T <= 0;
      elsif (rising_edge(clk1)) then
         T <= T + V;
         V := V + 2;
      end if;
   end process;

   process (clk2)
   begin
      if clk2 = '1' then
         if (reset = '1') then
            U <= 5;
         else
            U <= A + T;
            F <= U;
         end if;
      end if;
   end process;

end jaimeVHDL;
```

Question 6 - Compléter et remettre cette page avec votre cahier d'examen

Nom : _____ Matricule : _____



Solution :

La variable V ne dépend d'aucun autre signal. Sa valeur initiale est 3, spécifiée lors de sa déclaration. Elle est mise à jour lors de transitions positives du signal clk1, sauf lors d'un reset où elle conserve sa valeur.

Le signal T dépend de V. Sa valeur initiale est -5, spécifiée lors de sa déclaration. Il est mis à jour lors de transitions positives du signal clk1. La valeur de V utilisée pour le calcul est celle de la dernière itération du processus. Lors d'un reset, il prend la valeur 0.

Le signal U dépend de T et de A. Sa valeur initiale est -5, spécifiée lors de sa déclaration. Il est mis à jour lors de transitions positives du signal clk2, parce que clk2 est dans la liste de sensitivité du processus et qu'on a ajouté la condition (if clk2 = '1'). Attention, le résultat de la synthèse serait différent de celui de la simulation. Lors d'un reset, il prend la valeur 5.

Le signal F dépend uniquement de U. Sa valeur initiale est -2^{31}, parce que son type est integer et qu'on ne spécifie pas de valeur par défaut. Il est mis à jour lors de transitions positives du signal clk2, parce que clk2 est dans la liste de sensitivité du processus et qu'on a ajouté la condition (if clk2 = '1'). Attention, le résultat de la synthèse serait différent de celui de la simulation. La valeur assignée à F est celle de U lors de la dernière exécution du processus.