



## I. DESCRIPTION DE LA FONCTION FP4

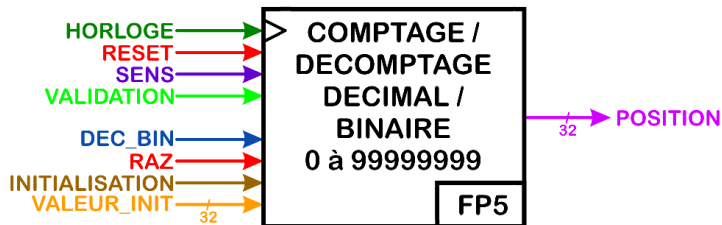
Le but est ici de concevoir un compteur/décompteur décimal/binaire sur 32 bits (ou 8 chiffres BCD).

Les bibliothèques FPGA Generic d'Altium contiennent un grand nombre de compteurs de différents types mais ne possédant pas simultanément toutes les possibilités que l'on désire.

Il existe dans la bibliothèque FPGA Configurable Generic d'Altium un compteur configurable mais il ne peut fonctionner qu'en binaire ou Johnson, pas en BCD. De plus le RESET place ce compteur soit à la valeur 0 soit à la valeur 9 suivant qu'il soit dans le sens comptage ou décomptage.

Le langage VHDL est justement conçu pour décrire facilement de nouveaux composants en fonction des besoins. Ces nouveaux composants pourront être réutilisés à loisir.

### I.1. DESCRIPTION DU COMPTEUR/DÉCOMPTEUR SUR 32 BITS



Une entrée HORLOGE active sur front montant permettra le fonctionnement de l'ensemble du compteur/décompteur en full synchrone.

Une sortie POSITION sur 32 bit représentera la valeur du compteur soit en BCD soit en BINAIRE.

Une entrée RESET active à 1 permettra d'initialiser le composant (à la mise sous tension par exemple et d'imposer la valeur 0 à POSITION).

Une entrée DEC\_BIN permettra de choisir la base de comptage: BCD (valeur 1) ou BINAIRE (valeur 0).

Une entrée RAZ active à 1 permettra de remettre POSITION à la valeur 0.

Une entrée SENS permettra soit d'effectuer un comptage (valeur 1) soit un décomptage (valeur 0).

Une entrée VALIDATION active à 1 permettra au compteur d'évoluer (comptage ou décomptage).

Une entrée INITIALISATION active à 1 permettra de placer le mot de 32 bits VALEUR\_INIT dans POSITION (chargement parallèle).

L'écriture en VHDL d'un compteur/décompteur binaire sur 32 bits est très simple.

Les entrées RESET et RAZ sont aussi faciles à intégrer, de même que le chargement parallèle.

La seule difficulté est le comptage en mode BCD, en effet, à chaque évolution du compteur dans ce mode il faut effectuer un ajustement décimal (transformer le binaire pur en BCD ou Décimal Codé Binaire).

Remarque : l'entrée DEC\_BIN sera positionnée soit à 0 soit à 1 de manière définitive dans un schéma.

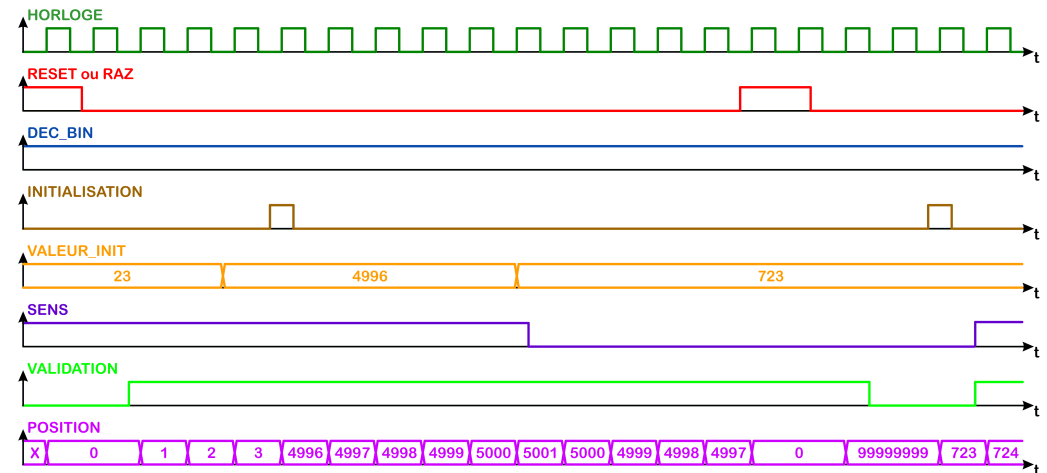
Remarques sur le fichier VHDL :

Dans la description d'un process les conditions sur les entrées les plus prioritaires doivent être placées en haut de la description. Il est conseillé de traiter en premier lieu les cas particuliers pour finir par les cas généraux.

Dans un process une variable n'a pas d'existence physique, il est donc impossible de l'observer lors d'une simulation.

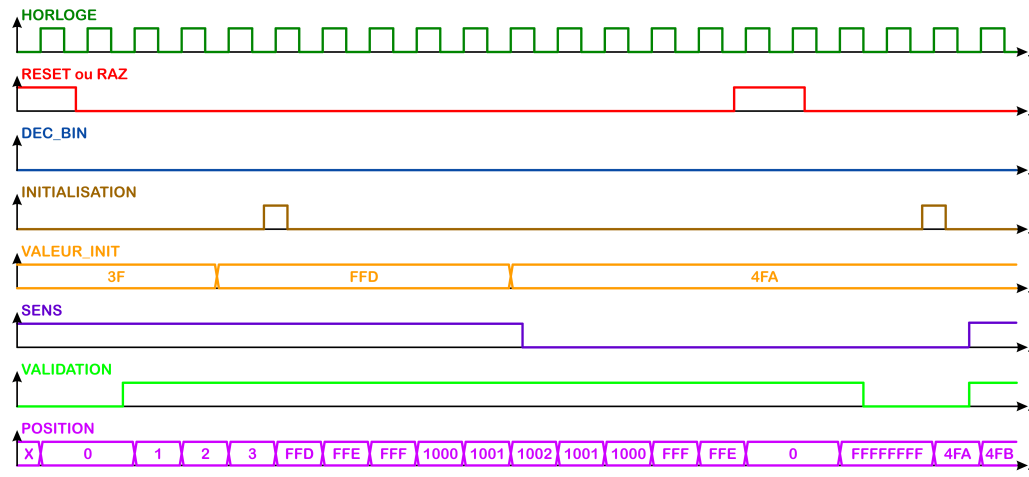
### I.2. CHRONOGRAMMES ATTENDUS

#### Compteur/décompteur BCD



Les valeurs de VALEUR\_INIT et de POSITION sont affichées en décimal.

## Compteur/décompteur binaire



Les valeurs de VALEUR\_INIT et de POSITION sont affichées en hexadécimal.

## II. CONCEPTION DE LA FONCTION FP5

### II.1. AJOUT AU PROJET D'UN NOUVEAU FICHIER VHDL

Ajouter un nouveau fichier VHDL au projet. Renommer-le **Comp-  
teur\_Decompteur\_DEC\_BIN\_32\_Bits.Vhd**.

## I.2. ÉDITION ET COMPILATION DU FICHIER VHDL

Éditer le fichier VHDL, l'enregistrer puis le compiler.  
(Voir listing en annexe pages 3 à 5).

## III. SIMULATION FONCTIONNELLE

### III.1. PRODUCTION DU FICHIER DE TEST

Créer le fichier VHDL de test.  
Proposer et ajouter à celui-ci les lignes des process de tests afin de faire évoluer les ports d'entrées suivant un jeu d'essai permettant de valider le bon fonctionnement.

Enregistrer celui-ci en gardant le nom proposé.  
Compiler ce fichier puis compiler le projet complet.

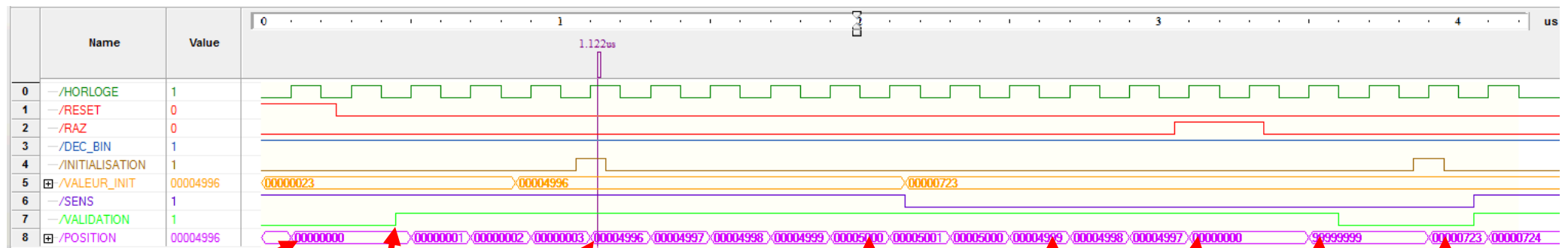
### III.2. ASSOCIATION DES FICHIERS

Ajouter le fichier de test dans le Manager Testbenches du simulateur.

### III.3. SIMULATION

Lancer la simulation.

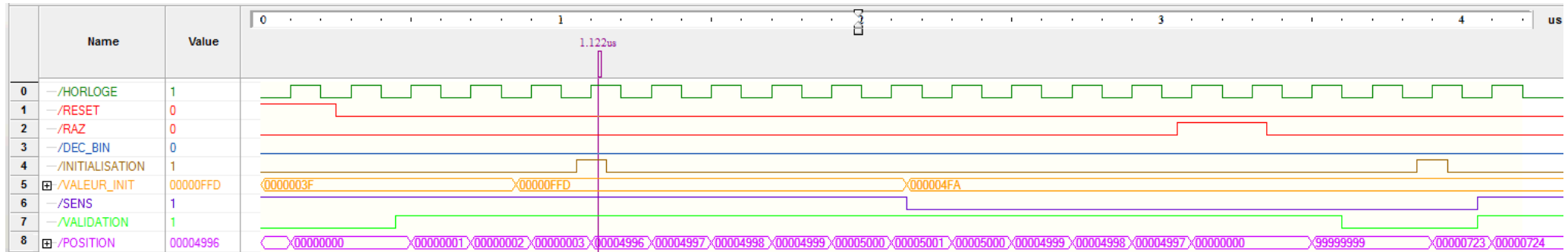
Simulation en mode BCD.



Initialisation par RESET  
Autorisation de comptage par VALIDATION  
Chargement parallèle de VALEUR\_INT par INITIALISATION

Comptage puis décomptage par SENS  
Remise à zéro par RAZ  
Décomptage avec débordement  
Chargement parallèle de la nouvelle VALEUR\_INT par INITIALISATION

Simulation en mode binaire.



#### IV. CRÉATION D'UN SYMBOLE POUR LA FONCTION FP3

Vérifier que la fenêtre de travail est bien le fichier VHDL **Comp-  
teur\_Decompteur\_Dec\_Bin\_32\_bits.Vhd** puis créer un symbole avec la commande :

⇒ **Design** ⇒ **Create Schematic Part From File**

Faire **Ok**. Le symbole apparaît alors à l'écran.

Sauvegarder le sous le nom **Compteur\_Decompteur\_Dec\_Bin\_32\_bits.SchLib**  
Fermez-le : il nous servira plus tard.

#### V. LISTING

```
1  -- Projet Gestion_Codeur_Incremental
2  -- Compteur/Décompteur BCD/BINAIRE
3
4  -- Description d'un Compteur/Decompteur sur 32 bits Full Synchrone
5  -- Entrée HORLOGE d'horloge active sur front montant
6  -- Entrée RESET d'initialisation active à 1
7  -- Entrée RAZ de remise à zéro active à 1
8  -- Entrée IINITIALISATION de commande de chargement parallèle sur 32 bits active à 1
9  -- Entrée DEC_BIN de sélection de la base décimale (valeur 1) ou hexadécimale (valeur 0)
10 -- Entrée SENS de sélection du comptage (valeur 1) ou décomptage(valeur 0)
11 -- Entrée VALIDATION de validation du comptage active à 1
12 -- Sortie POSITION représentative du comptage sur 32 bits
13 -- 19/11/2012 C. D. Lycée Chevrolier Angers
14
15 library IEEE;
16 use IEEE.STD_LOGIC_1164.ALL;
17 use IEEE.STD_LOGIC_ARITH.ALL;
18 use IEEE.STD_LOGIC_UNSIGNED.ALL;
19
```

```

20 entity COMPTEUR_DECOMPTEUR_DEC_BIN_32_BITS is
21     port(
22         HORLOGE          : in  std_logic;
23         RESET            : in  std_logic;
24         RAZ              : in  std_logic;
25         DEC_BIN          : in  std_logic;
26         SENS             : in  std_logic;
27         VALIDATION       : in  std_logic;
28         INITIALISATION   : in  std_logic;
29         VALEUR_INIT      : in  std_logic_vector (31 downto 0);
30         POSITION          : out std_logic_vector (31 downto 0);
31     );
32 end COMPTEUR_DECOMPTEUR_DEC_BIN_32_BITS;
33
34 architecture ARCH_COMPTEUR_DECOMPTEUR_DEC_BIN_32_BITS of COMPTEUR_DECOMPTEUR_DEC_BIN_32_BITS is
35     begin
36         process(HORLOGE)
37             variable NOMBRE : std_logic_vector (31 downto 0);
38             begin
39                 if (HORLOGE'event and HORLOGE = '1')
40                 then
41                     if (RESET = '1' or RAZ = '1')
42                     then
43                         NOMBRE := (others => '0');
44                     elsif (INITIALISATION = '1')
45                     then
46                         NOMBRE := VALEUR_INIT;
47                     elsif (VALIDATION = '1')
48                     then
49                         if (SENS = '1')
50                         then
51                             NOMBRE := NOMBRE + 1;
52                             if (DEC_BIN = '1')
53                             then
54                                 if (NOMBRE(3 downto 0) = "1010")
55                                 then
56                                     NOMBRE(31 downto 0) := NOMBRE(31 downto 0) + 6;
57                                 end if;
58                                 if (NOMBRE(7 downto 4) = "1010")
59                                 then
60                                     NOMBRE(31 downto 4) := NOMBRE(31 downto 4) + 6;
61                                 end if;
62                                 if (NOMBRE(11 downto 8) = "1010")
63                                 then
64                                     NOMBRE(31 downto 8) := NOMBRE(31 downto 8) + 6;
65                                 end if;
66                                 if (NOMBRE(15 downto 12) = "1010")
67                                 then
68                                     NOMBRE(31 downto 12) := NOMBRE(31 downto 12) + 6;
69                                 end if;
70                                 if (NOMBRE(19 downto 16) = "1010")
71                                 then
72                                     NOMBRE(31 downto 16) := NOMBRE(31 downto 16) + 6;
73                                 end if;

```

```

--Description de l'entité (broches du composant)
--Entrée de l'horloge
--Entrée d'initialisation (système synchrone)
--Entrée de remise à zéro
--Entrée de sélection de la base de comptage/décomptage
--Entrée de sélection du comptage/décomtrage
--Entrée de validation du comptage/décomptage
--Entrée de commande du chargement parallèle
--Entrée représentative de la valeur de chargement parallèle
--Sortie représentative de la valeur du compteur

--Fin de la description de l'entité

--Description du comportement interne (silicium)
--Partie synchrone sur le port HORLOGE
--Déclaration d'une variable sur 32 bits
--Fonctionnement complètement synchrone
--Attente d'un front montant du port HORLOGE

--Si RESET ou RAZ actif

--Remise du compteur à zéro
--Sinon si INIALISATION est actif

--Copie VALEUR_INIT dans le compteur (chargement parallèle)
--Sinon si VALIDATION actif

--Et dans le sens comptage

--Incrémentaer le compteur
--Si mode décimal
--Effectuer un ajustement décimal
--Si premier chiffre plus grand que 9

--alors ajouter 6

--Si second chiffre plus grand que 9

--alors ajouter 60

--Si troisième chiffre plus grand que 9

--alors ajouter 600

--Si quatrième chiffre plus grand que 9

--alors ajouter 6 000

--Si cinquième chiffre plus grand que 9

--alors ajouter 60 000

```

```

74         if (NOMBRE(23 downto 20) = "1010")           --Si sixième chiffre plus grand que 9
75             then
76                 NOMBRE(31 downto 20) := NOMBRE(31 downto 20) + 6; --alors ajouter 600 000
77             end if;
78         if (NOMBRE(27 downto 24) = "1010")           --Si septième chiffre plus grand que 9
79             then
80                 NOMBRE(31 downto 24) := NOMBRE(31 downto 24) + 6; --alors ajouter 6 000 000
81             end if;
82         if (NOMBRE(31 downto 28) = "1010")           --Si huitième chiffre plus grand que 9
83             then
84                 NOMBRE(31 downto 0) := (others => '0');      --Remettre à zéro
85             end if;
86     end if;                                           --Fin de la condition sur l'entrée DEC_BIN
87 else                                                 --Sinon dans le sens Décomptage
88     NOMBRE := NOMBRE - 1;                             --Décrémenter le compteur
89     if (DEC_BIN = '1')                               --Si mode décimal
90     then                                             --Effectuer un ajustement décimal
91         if (NOMBRE(3 downto 0) = "1111")           --Si premier chiffre plus petit que 0
92             then
93                 NOMBRE(3 downto 0) := "1001";      --alors le mettre à 9
94             end if;
95         if (NOMBRE(7 downto 4) = "1111")           --Si second chiffre plus petit que 0
96             then
97                 NOMBRE(7 downto 4) := "1001";      --alors le mettre à 9
98             end if;
99         if (NOMBRE(11 downto 8) = "1111")           --Si troisième chiffre plus petit que 0
100            then
101                NOMBRE(11 downto 8) := "1001";     --alors le mettre à 9
102            end if;
103         if (NOMBRE(15 downto 12) = "1111")           --Si quatrième chiffre plus petit que 0
104            then
105                NOMBRE(15 downto 12) := "1001";    --alors le mettre à 9
106            end if;
107         if (NOMBRE(19 downto 16) = "1111")           --Si cinquième chiffre plus petit que 0
108            then
109                NOMBRE(19 downto 16) := "1001";    --alors le mettre à 9
110            end if;
111         if (NOMBRE(23 downto 20) = "1111")           --Si sixième chiffre plus petit que 0
112            then
113                NOMBRE(23 downto 20) := "1001";    --alors le mettre à 9
114            end if;
115         if (NOMBRE(27 downto 24) = "1111")           --Si septième chiffre plus petit que 0
116            then
117                NOMBRE(27 downto 24) := "1001";    --alors le mettre à 9
118            end if;
119         if (NOMBRE(31 downto 28) = "1111")           --Si huitième chiffre plus petit que 0
120            then
121                NOMBRE(31 downto 28) := "1001";    --alors le mettre à 9
122            end if;
123     end if;                                           --Fin de la condition sur l'entrée DEC_BIN
124 end if;                                               --Fin de la condition sur l'entrée SENS
125 POSITION <= NOMBRE;                                  --Fin de la condition sur l'entrée RESET
126 end if;
127 end process;                                         --Fin de la condition sur l'entrée HORLOGE
128 end ARCH_COMPTEUR_DECOMPTEUR_DEC_BIN_32_BITS;      --Fin de description synchrone
129 end ARCH_COMPTEUR_DECOMPTEUR_DEC_BIN_32_BITS;      --Fin de la description interne

```