Examen CE317 **Partie matérielle « conception numérique & VHDL »**	**3ème année 2016-2017 Session 2** **Durée indicative de la partie : 90 mn** **Document autorisé : syntaxe VHDL essentielle** **Calculatrice interdite**
NOM :	**PRENOM :**

Les points donnés dans l'énoncé entre crochets [X] après chaque question représentent le barème et indiquent le temps à passer en minutes sur chaque question.

Rédigez sur des feuilles séparées les 2 parties de cet examen.

Conception d'un chronomètre [96]

1 Codage d'un compteur modulo 16 [36]

1.1 En utilisant l'entité décrite ci-dessous, codez en VHDL RTL l'architecture d'un compteur modulo 16 sur 4 bits : [12]

a. Reset synchrone actif sur niveau haut
b. Comptage des impulsions sur front montant d'horloge
c. Activation / désactivation du comptage (e.q. enable)
d. Gestion du débordement (retour à 0 après la valeur maximale)
e. Temoin de débordement

```
entity compteur_16 is
    Port ( clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           enable : in STD_LOGIC;
           deb  : out STD_LOGIC;
           out_count : inout STD_LOGIC_VECTOR (3 downto 0)
           );
end compteur_N;
```

1.2 Codez un testbench simple permettant de vérifier le fonctionnement de votre compteur modulo 16 [12].

1.3 Réalisez manuellement les chronogrammes de toutes les entrées/sorties et valeurs internes de votre compteur modulo 16 que vous obtiendrez en simulant votre testbench précédent [12].

2 Codage d'un compteur modulo N [24]

2.1 En utilisant l'entité décrite ci-dessous, codez en VHDL RTL un compteur modulo N respectant les specifications ci-dessous : [12]

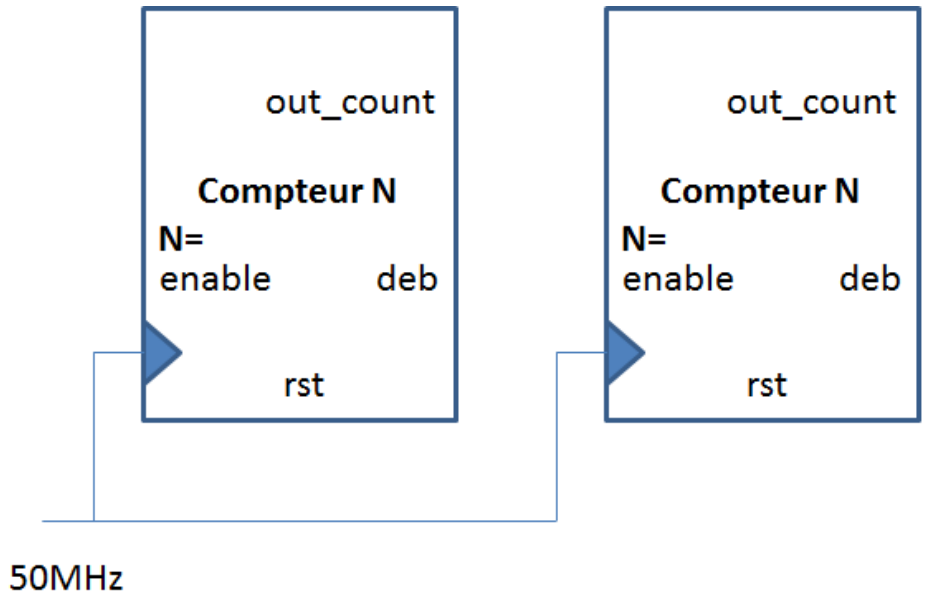
a. Reset synchrone actif sur niveau haut
b. Comptage des impulsions sur front montant d'horloge
c. Activation / desactivation du comptage (e.q. enable)
d. Gestion du débordement (retour à 0)
e. Temoin de débordement
f. N paramétrable sous forme de deux generics : un paramètre C_NB_BIT_COUNTER pour la profondeur du compteur, un paramètre pour la valeur du modulo N C_MODULO.

```
entity compteur_N is
    Generic (
        C_NB_BIT_COUNTER        : integer := 4;
        C_MODULO                : integer := 10
        );
    Port ( clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        enable : in STD_LOGIC;
        deb  : out STD_LOGIC;
        out_count : inout STD_LOGIC_VECTOR (C_NB_BIT_COUNTER - 1 downto 0)
        );
end compteur_N;
```

2.2 Codez un testbench permettant de vérifier le fonctionnement de votre compteur modulo N pour N=10 (c'est-à-dire avec C_NB_BIT_COUNTER=4 et C_MODULO=10) [12]

3 Architecture du chronomètre [36]

1.1 Complétez le schéma ci-dessous représentant l'achitecture du chronomètre « secondes / minutes » basée sur deux compteurs modulo N et d'une horloge systéme (commandant ces 2 compteurs) à 50MHz [12].



Remarque : Pensez à bien indiquer les valeurs des modulos N des différents compteurs.

1.2 Combien de bascules D seront nécessaires afin d'implémenter cette architecture ? [12]

1.3 Codez l'architecture du chronomètre en instanciant les compteurs modulo N comme représenté dans la figure précédente [12].