

3App - Durée : 180mn Examen du Cours CE317 « Partie Matérielle »	Session 1 année 2015 – 2016 Un seul document autorisé : « VHDL résumé de syntaxe » Calculatrice interdite
<p>1 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</p> <p>2 La qualité de la rédaction (lisibilité, orthographe) sera prise en compte</p>	

Rédiger vos réponses sur des copies séparées : une pour la partie matérielle et une autre pour la partie logicielle

I Partie cours [34/90]

1.1 Codez en virgule fixe avec le minimum de bits 511,75 [6]

1.2 Simplifiez en utilisant un tableau de Karnaugh la fonction suivante [12] :

$$F = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + ABC$$

1.3 En VHDL, quel type **normalisé** doit être utilisé pour réaliser des opérations arithmétiques sur des **vecteurs binaires non signés** [2]?

Quels bibliothèques et paquetages doivent-êtré préalablement déclarés pour utiliser ce type [2]?

Montrez en détaillant l'opération sur des nombres binaires signés en « complément à 2 » (de tailles suffisantes) comment le résultat de l'opération « -5 x -7 » est obtenu [6].

1.4 En VHDL est-ce que l'ordre des processus dans le code a une importance ? Si non, dans quel ordre sont exécutés les processus ? [6]

Commentaire [BV1]:

512 sur 9 bits donc de 0 -> 511

511 = 1_1111_1111.11

$0,5_{10} = 2^{-1}$
 $0,25_{10} = 2^{-2}$

Commentaire [BV2]: F=C+AB

Commentaire [BV3]: signed

Commentaire [BV4]:

Biblio : IEEE
 Paquetage :
 std_logic_1164
 numeric_std

Commentaire [BV5]:

En signé
 -5 : 0101 => 1010+1=> 1011
 8 : 01000
 En fait le nombre minimal de bit est de 8 (en fait 4+5-1 bit de signe)
 Mais le plus simple est de travailler en valeur absolue positive puis de prendre l'opposé du résultat :
 01000
 00101

 0001000
 0100000

 0101000 (40)
 => 1010111+1=101_1000 (-40)

Commentaire [BV6]: Non, les processus sont exécutés quand ils sont réveillés (par exemple, quand un des signaux de leur liste de sensibilité change)

II Partie exercices - Conception d'un registre à décalage SIPO [56/90]

2.1 Dessinez le schéma d'un **registre à décalage de n bits SIPO (Serial In – Parallel Out)** [10] en utilisant uniquement n bascules D Flip-Flop (cf. l'entité ci-dessous, entité qui contient une entrée reset synchrone et une entrée enable).

Ce registre à décalage comporte :

- un simple bit d'entrée (din)
- une capture de (din) et un décalage des bits à chaque front montant de l'horloge (clk)
- une entrée d'autorisation de décalage (enable) active à l'état haut
- une sortie parallèle (dout) sur n bits (les n sorties des n bascules)
- une remise à zéro (reset) synchrone active à l'état haut.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity bascule is
    port(    d, clk, reset, enable : in std_logic ;
           q : out std_logic);
end entity bascule;
```

2.2 Ecrire en **VHDL RTL** l'entité **générique** (*shiftN*) et l'**architecture** (*rtl*) du composant précédent [14]

Commentaire [BV7]: [4]

2.3 Ecrire cette fois en **VHDL structurel** l'architecture (*struct*) de la même entité générique (shiftN) précédente [8].

Commentaire [BV8]: [8]

2.4 Ecrire en VHDL un testbench permettant de simuler l'**architecture structurelle struct** précédente pour n=8 [14].

Remarques : Votre testbench montrera sur **plusieurs bits d'entrée** que chacune des fonctions « décalage, autorisation de décalage et remise à zéro » fonctionne correctement.

2.5 Dessinez les chronogrammes des signaux **din, enable, reset, clk, dout** correspondant **très exactement** à la simulation du testbench précédent [10].

Remarques : Pensez à **indiquer clairement l'échelle des temps et les temps correspondant à chaque évènement** sur l'axe temporel