

| | |
|---|---|
| <p>3App - Durée : 90 mn</p> <p>Examen CE317</p> <p>Partie VHDL</p> <p>NOM :</p> | <p>1^{er} Session année 2019 – 2020</p> <p>Un seul document autorisé :</p> <p>« VHDL résumé de syntaxe »</p> <p>Calculatrice interdite</p> <p>PRENOM :</p> |
| <p><i>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</i></p> <p><i>2 Rendre une copie séparée pour chaque partie de l'examen</i></p> | |

1 VHDL pour la simulation [12]

- 1.1 Complétez dans le chronogramme ci-dessous l'évolution de b et s obtenu par simulation du code [5].
- 1.2 Si on fait la synthèse de ce code un warning apparait et une netlist est quand même produite. Pourquoi ce warning apparait-il? [2] Complétez ci-dessous le résultat de la simulation de la netlist produite par la synthèse [5].

| | |
|---|-----------------------|
| <pre>entity exemple_proc is port (a : in bit ; b : out bit); end exemple_proc ; architecture simple of exemple_proc is signal s : bit ; begin process (a) begin s <= a ; b <= s ; end process ; end simple ;</pre> | <p>1.1</p> <p>1.2</p> |
|---|-----------------------|

2 VHDL pour la synthèse : registre à décalage n bits [84]

Le registre à décalage *shifter* que l'on veut concevoir peut être soit chargé avec un vecteur *din* (entrée) soit sa valeur *dout* (sortie) être décalée à gauche ou à droite (d'un seul bit à la fois par front montant de l'horloge). La nouvelle valeur entrante (à gauche ou à droite) est toujours '0'.

Les valeurs binaires manipulées sont tous de type *std_logic* ou *std_logic_vector*.

Les signaux d'entrée/sortie du registre à décalage sont les suivants :

- *din* : vecteur d'entrée de n bits de large utilisé pour le chargement
- *clk* : le registre est sensible au front montant de l'horloge *clk*
- *left_right* : entrée indiquant un décalage à gauche (si '1') ou à droite (si '0')
- *load* : entrée indiquant le chargement de l'entrée *din*
- *dout* : vecteur de sortie n bits de large

2.1 Décrire en VHDL l'entité générique de ce composant appelé *shifter* [5]

2.2 Décrire en VHDL l'architecture de ce composant **au niveau RTL** [20]

2.3 Combien de bascules produira la synthèse de votre code précédent ? Justifiez [2].

2.4 Faire un schéma composé de bascules D Flip-Flop et de Mux « 4 vers 1 » de ce registre [10]. Vérifiez que le nombre de bascules correspond bien au nombre produit par la synthèse précédente [2].

2.5 Décrire en VHDL l'architecture de ce composant **au niveau structurel** en instanciant les composants ci-dessous [20]

| | |
|--|---|
| component bascule is Port(d : in std_logic ; clk : in std_logic; q : out std_logic); end component; | component mux is Port(e0,e1,e2,e3 : in std_logic ; sel1 : in std_logic; sel2 : in std_logic; q : out std_logic); end component; |
| Bascule D Flip-Flop active sur front | Mux « 4 vers 1 » |

Conseil 1 : utilisez l'instruction *for...generate* vue en cours

Conseil 2 : pour laisser un port non utilisé non connecté dans le port map utilisez le mot clef **open**.

2.6 On suppose que l'horloge *clk* a une période de 10 ns et présente un premier front montant à 5 ns (puis à 15 ns, 25 ns...), complétez le tableau suivant pour un composant shifter avec n=4 : [10]

| Temps | load | left_right | Din | dout |
|-------|------|------------|----------|------|
| 0 ns | 'U' | 'U' | « UUUU » | |
| 10 ns | '1' | '0' | « 1010 » | |
| 15 ns | '1' | '0' | « 1010 » | |
| 20 ns | '0' | '0' | « 1010 » | |
| 25 ns | '0' | '0' | « 1010 » | |
| 35 ns | '0' | '0' | « 1010 » | |
| 40 ns | '0' | '1' | « 1111 » | |
| 45 ns | '0' | '1' | « 1111 » | |
| 55 ns | '0' | '1' | « 1111 » | |

2.7 Ecrire en VHDL un testbench permettant de réaliser la simulation de l'architecture RTL et **permettant de retrouver exactement les résultats contenus dans le tableau précédent** [15].