Nom:

Évaluation de connaissances

Durée max : 1h30

Aucun document autorisé.

- Barème et temps sont donnés à titre indicatif.
- N'oublier pas de rendre le sujet avec votre copie.
- Lecture du sujet et impondérables temporels : 5' à 10' max
- Aucune rature ni correcteur "blanco" n'est admis pour les QCM.
- Le poids faible (LSB) d'un nombre binaire sera toujours ici "indicé" 0

Q1) [2pts] On considère un démultiplexeur 2 vers 4 et une entrée de validation. On notera :

- *les entrées I_n (n=0 à 1)*
- *les sorties actives au niveau logique bas, S_n (n=0 à 3)*
- *EN l'entrée de validation active au niveau logique bas.*

a) Donner les équations logiques de ce multiplexeur.

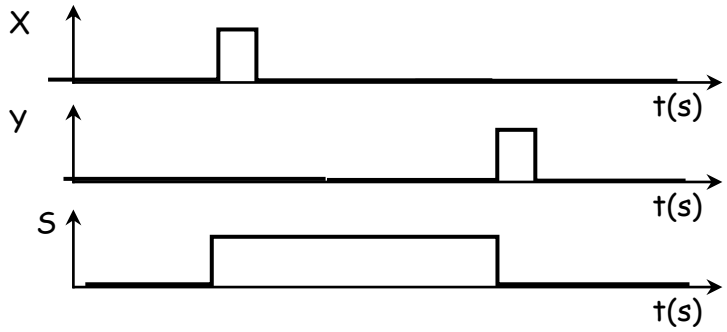
$\overline{S_0} =$

$\overline{S_1} =$

$\overline{S_2} =$

$\overline{S_3} =$

Q2) [1pt] On considère un petit système numérique décrit pas le chronogramme suivant :



a) Est-il combinatoire ou est-il séquentiel ? Justifier votre réponse.

*Q3) [1pt] Une architecture en VHDL utilisant **un ou plusieurs process** (concurrents) peut décrire*

- ☐ uniquement des fonctions logiques séquentielles.
- ☐ uniquement fonctions logiques combinatoires.
- ☐ A la fois des fonctions logiques séquentielles et combinatoires.

*Q4) [2pts] Quelles sont les conditions nécessaires pour qu'un **process** soit purement combinatoire ?*

Q5) [0.5pt] Un FPGA est une cible

- ☐ logicielle.
- ☐ matérielle.

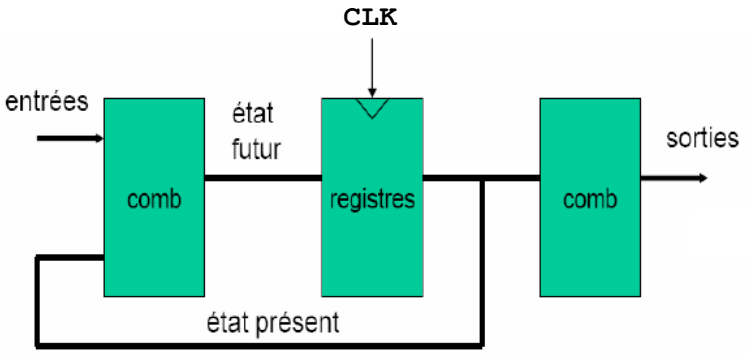
Q6) [0.5pt] Un micro-controleur est une cible

- ☐ logicielle.
- ☐ matérielle.

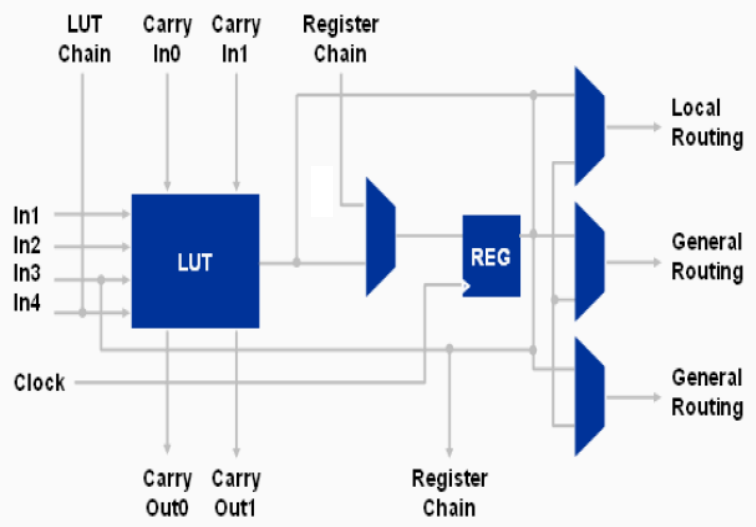
Q7) [1pt] Donner un exemple simple montrant rapidement pourquoi le VHDL n'est pas un langage de programmation.

Q8) [0.5pt] La représentation ci-dessous est une machine de

- ☐ Moore.
- ☐ Mealy.



Q9) [1.5pts] Un FPGA intègre de nombreuses ressources dont des macro-cellules dont une est représentée ci-dessous de façon très "simplifiée" :



On y retrouve en particulier une LUT à 4 entrées.

- Que signifie LUT ?
- A partir de quel composant est construite notre LUT ?
- Combien de fonctions booléennes permet de créer notre LUT ?

Q10) [7 pts] Reconnaissance de séquence.

On souhaite réaliser une petite machine de Moore qui reconnaît la séquence **110** arrivant sur <u>son unique</u> entrée **E**. Lorsque qu'une séquence est reconnue la sortie de la machine **passe à 0** et revient à l'état initial si E=0. De plus après le **premier 1** détecté de la séquence si E=0 on **retourne** à l'état initial.

On notera A, B, etc, les états que l'on codera en "one hot".

a) Proposer un graphe ou diagramme d'états en vérifiant qu'il est complet et non contradictoire.

b) Compléter alors les tables de transition et d'états ci-dessous :

Etat actuel (présent)	Etat suivant (futur)		S
	E=0	E=1	
A			

Etat actuel (présent)	Etat suivant (futur)		S
	E=0	E=1	
0001			

On implémentera la MAE avec des bascules D.

c) Combien de bascules **D** seront nécessaires pour implémenter cette MAE ?

d) Rechercher sur votre copie les équations simplifiées de l'état futur et de S en fonction des $Qi_{(i=0 \text{ à } x)}$ et de E. Les états actuels seront tels que Q_0 sera le LSB et Q_x le MSB et les Q_i* les états futurs.

e) Dessiner sur votre copie le schéma structurel correspondant à cette MAE.
On utilisera des opérateurs logiques à 2 entrées et on délimitera bien les IFL, les registres et les OFL

Q11) [3 pts] Soit l'architecture suivante utilisant un process.

a) Analyser cette description et compléter les chronogrammes ci-dessous.

```
signal compt : integer range 0 to 7 ;
signal etat : std_logic;
begin
      process(CLK)
      begin
            if (CLK'event and CLK='1') then
                  if EN ='1' then
                        compt <= compt + 1;
                        case etat is
                        when '0' =>
                              if compt = 3 then
                              compt <= 0
                              SORTIE <= '1;
                              etat <= '1';
                              end if;
                        when '1' =>
                              if compt = 2 then
                              compt <= 0;
                              SORTIE <= '0;
                              etat <= '0';
                              end if;
                        end case;
                  end if;
            end if;
      end process;
```