Examen partiel

Département de génie électrique et de génie informatique
Systèmes VLSI - GIF19264

le 26 octobre 2007

Vous avez droit à tous les documents. SVP, *pas* d'ordinateurs portables.
Durée de l'examen: 3 heures (11h30-14h20).

---

1. *dispostifs logiques programmables* (10 points) Donnez le nom de la technologie ou du dispositif qui correspond à l'énoncé.

(a) Premier type de puce utilisé historiquement pour implanter de la logique programmable.

(b) Dispositif logique programmable simple constitué d'un plan-ET et d'un plan-OU, tous deux étant programmables.

(c) Type de mémoire et de technologie de reconfiguration d'un FPGA qui est une forme d'EEPROM, mais qui supporte en plus l'effaçage en bloc, ce qui est beaucoup plus rapide que l'effaçage d'un EEPROM conventionnel.

(d) Technologie de configuration utilisée dans une famille de FPGAs à granularité fine préconisée pour les applications aérospatiales (e.g. satellites).

(e) Dispositif logique programmable compexe ayant une structure rigide et des délais entrée-sortie prévisibles.

2. *syntaxe, sémantique, types, attributs* (20 points)

(a) (6 points) Soit la description VHDL suivante:

```
entity parite
   port( X: in std_logic_vector(3 downto 0);
         Y: out std_logic));
end entity mux2a1;

architecture flot of parite is
   signal interm: std_logic_vector(1 downto 0);
begin
   P1: process (X(1 downto 0))
   begin
      interm(0) <= X(1) xor X(0);
   end process P1;

   P2: process (X(3 downto 2))
   begin
      interm(1) <= X(3) xor X(2);
   end process P2;

   P3: process(interm(0), Y)
   begin
      Y <= interm(1);
      Y <= interm(1) xor interm(0);
   end process P3;

   Y <= 'Z' when interm = "ZZ";
end architecture flot;
```

Cette description réalise-t-elle un circuit combinatoire? Comporte-t-elle des erreurs? Peut-elle être synthétisée avec ISE? Justifiez.

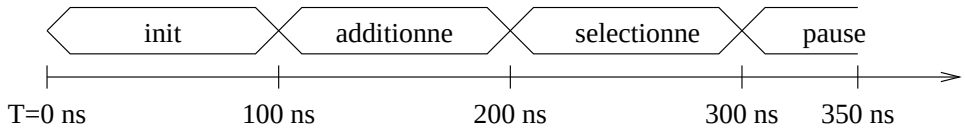
(b) (2 points) Étant donné la description en (a), quelle est la valeur de l'attribut flot'Structure?

(c) (2 points) Étant donné la description en (a), quelle est la valeur de l'attribut flot'Behavior?

(d) (10 points) Soit la déclaration suivante:

```
type type_etat is (init, additionne, compare, selectionne, pause);
signal etat: type_etat;
```

Quelle est la valeur des expressions suivantes? Supposez que le signal etat possède l'historique montré ci-dessous et que l'évaluation des expressions se fait au temps de simulation $T = 350ns$.



(i) type_etat'left

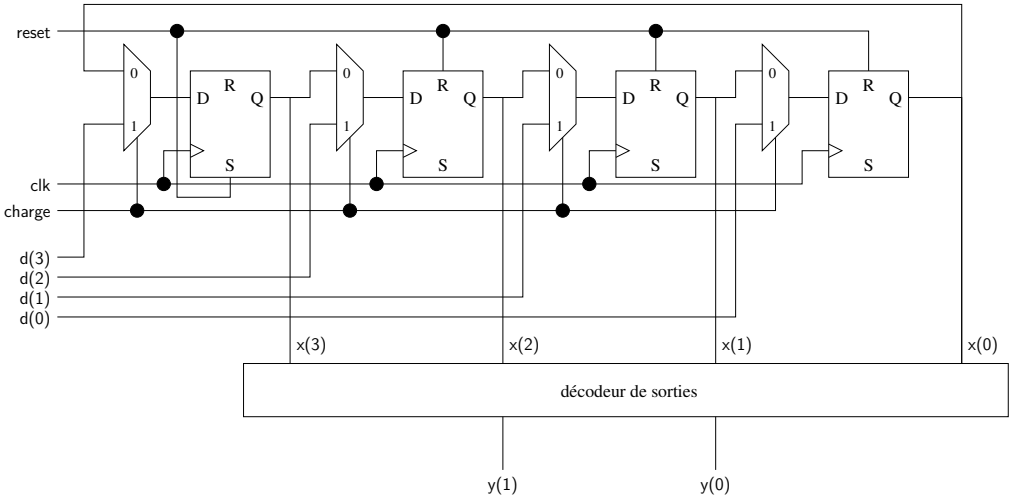
(ii) type_etat'high(1)

(iii) type_etat'rightof(pause)

(iv) etat'delayed(225 ns)

(v) etat'last_active

3. *Conception, théorie des machines à état* (30 points) Le circuit suivant est un compteur à 4 états qui restitue une séquence sur 2 bits. On remarque que le signal reset initialise la première bascule à 1 et les autres à zéro. Cet état initial correspond à la sortie 00 et la séquence de sortie du compteur doit correspondre à la séquence naturelle $\{00, 01, 10, 11\}$. On note également qu'il y a un signal de chargement (charge) qui permet d'initialiser les bascules à une valeur arbitraire.



(a) (5 points) Ce compteur correspond à une machine à états. Si les bascules constituent ensemble le registre d'états, quel(les) composante(s) correspond au décodeur d'entrée?

(b) (5 points) Est-ce une machine de Mealy ou une machine de Moore? Expliquez.

(c) (10 points) Tracez le diagramme d'états de cette machine en faisant abstraction de l'entrée charge (faites comme si l'entree charge était toujours en zéro).

(d) (10 points) Étant donné l'entité suivante, écrivez le code VHDL du décodeur de sorties.

```
entity decode_sorties is
   port( x: in std_logic_vector(3 downto 0);
         y: out std_logic_vector(1 downto 0));
end entity decode_sorties
```

4. *Méthodologie de conception séquentiel* (40 points) Le célèbre "jeu de la vie" de Conway est une simulation impliquant un ensemble de "cellules" sur une grille cartésienne. À partir d'une configuration initiale, l'état de chaque cellule évolue en fonction de l'état des cellules voisines, ce qui en fait un *automate cellulaire*. L'état de toutes les cellules change en même temps, ce qui correspond à une *génération*. Une cellule est soit "vivante" (état 1) ou "morte" (état 0). L'automate est entièrement défini par les règles simples suivantes qui sont évaluées pour passer d'une génération à l'autre:

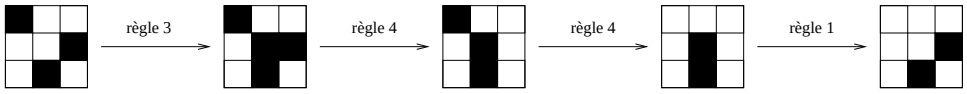
1. Si une cellule vivante a moins de 2 voisins, elle meurt d'isolement.

2. Si une cellule vivante a plus de 3 voisins, elle meurt d'étouffement.

3. Si une cellule morte a exactement 3 voisins, elle "nait", i.e. elle devient vivante.

4. Si une cellule vivante a 2 ou 3 voisins, elle conserve son état.

On considère qu'une cellule possède 8 voisins sur grille cartésienne. L'illustration suivante montre l'évolution d'une cellule en fonction de ses 8 voisins au fil de 4 générations. On voit que l'état des voisins change également, en fonction de leurs 8 voisins respectifs.



On s'intéresse dans ce qui suit à la conception d'une machine à état de type C qui réalise une cellule dans le jeu de la vie. Assumez qu'il y a une génération par coup d'horloge et utilisez l'entité suivante:

```
entity jeudelavie
   port( etatvoisins: in std_logic_vector(3 downto 0);
         etatcellule: out std_logic;
         clk, reset: in std_logic));
end entity jeudelavie;
```

On assume également que le port etatvoisins donne le nombre de voisins vivants.

(a) (15 points) Dessinez le graphe conceptuel (actions) d'une machine à états réalisant cette fonction, en utilisant etatvoisins comme variable.

(b) (15 points) Dessinez schématiquement un circuit d'éléments fonctionnels permettant d'éliminer la variable dans le graphe en (a).

(c) (10 points) Écrivez une description VHDL (architecture) correspondante en séparant la partie synchrone et la partie combinatoire en des processus distincts.

Bonne chance!