

Examen partiel

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

le 8 novembre 2006

Vous avez droit à tous les documents. SVP, *pas* d'ordinateurs portables.
Durée de l'examen: 2 heures (13h30-15h30).

1. Types de PLDs (10 points)

- (a) (3 points) L'architecture des FPGAs de Xilinx est-elle à granularité fine ou grossière? Justifiez en quelques mots.

Elle est à granularité grossière car les CLBs sont relativement complexes et possèdent un grand nombre d'entrées - sorties.

- (b) (4 points) Quel est selon vous le lien entre la technologie de reconfiguration employée et la granularité d'une architecture FPGA?

On a tendance à adopter une granularité grossière lorsque la technologie de reconfiguration et les ressources de routage associées occupent beaucoup de place sur le dé. De cette manière, on réduit la quantité de ressources de routage nécessaires ce qui permet d'inclure plus de ressources logiques.

- (c) (3 points) Quels avantages offrent les CPLDs par rapport aux FPGAs?

À complexité équivalente, l'architecture plus rigide des CPLDs permet d'obtenir des délais entrée-sortie plus petits, qui sont de surcroît prévisibles. Ils sont également souvent moins gourmands en puissance.

2. *Modélisation et styles en VHDL* (25 points) Soit la description VHDL suivante:

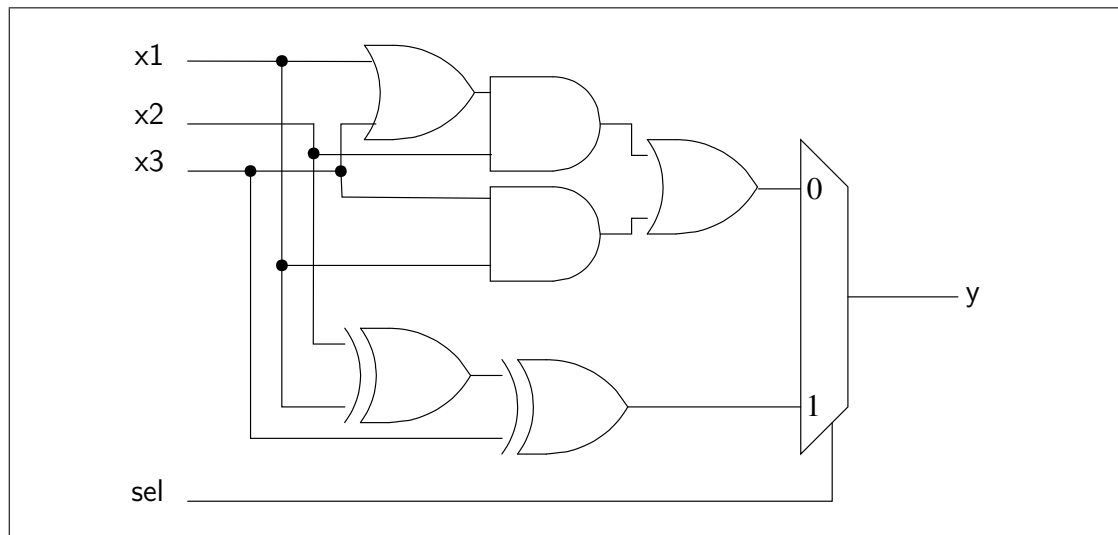
```

entity exercice2
  port( x1, x2, x3, sel: in std_ulogic;
        y: out std_ulogic);
end entity exercice2;

architecture archi of exercice2 is
  signal a, b, c, d, e, f: std_ulogic;
begin
  a <= x1 or x3;
  b <= x1 and x3;
  c <= x2 and a;
  d <= b or c;
  e <= x1 xor x2;
  f <= x3 xor e;
  P1: process (d, f, sel)
    begin
      if sel='0' then
        y <= d;
      else
        y <= f;
      end if;
    end process P1;
end architecture archi;

```

(a) (6 points) Tracez à partir d'éléments de base le schéma correspondant.



(b) (2 points) Cette description est-elle structurale ou comportementale?

C'est une description comportementale, puisqu'il n'y a aucune composante.

(c) (3 points) Le processus P1 est-il combinatoire ou séquentiel? Justifiez.

*C'est un processus combinatoire car il inclut toutes les entrées (signaux lus) dans la liste de sensibilité. De plus, toutes les valeurs possibles de **sel** sont traitées.*

(d) (6 points) Quel est selon vous la fonction de ce circuit?

C'est un plein-additionneur avec multiplexage des sorties somme et retenue.

(e) (8 points) À partir de la même entité, écrivez une architecture exploitant la logique 3-états pour la sélection de la sortie via le signal sel.

```
architecture archi2 of exercice2 is
    signal a, b, c, d, e, f: std_ulogic;
    signal ytemp: std_logic;
begin
    a <= x1 or x3;
    b <= x1 and x3;
    c <= x2 and a;
    d <= b or c;
    e <= x1 xor x2;
    f <= x3 xor e;
    P1: process (d, f, sel)
    begin
        if sel='0' then
            ytemp <= d;
        else
            ytemp <= 'z';
        end if;
    end process P1;
    P2: process (d, f, sel)
    begin
        if sel='1' then
            ytemp <= f;
        else
            ytemp <= 'z';
        end if;
    end process P2;
    y <= ytemp;
end architecture archi;
```

3. Conception en VHDL (25 points) Soit l'entité suivante:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity compteur
  port ( x, clk, reset: in std_logic;
        y: out unsigned (7 downto 0));
end entity compteur;
```

- (a) (20 points) Écrivez une description en VHDL comportemental à partir de cette entité qui compte les transitions montantes sur x et présente en tout temps le compte courant sur 8 bits à la sortie y.

Solution 1 - compteur asynchrone — comptera toutes les transitions qui sont détectables étant donné les limites du matériel, i.e. délais de propagation.

```
architecture archi of exercice3 is
begin
  P1: process (reset, x)
  begin
    if reset='1' then
      y <= "00000000";
    elsif x='1' and x'event then
      y <= y + '1';
    end if;
  end process P1;
end architecture archi;
```

Solution 2 - compteur synchrone — compte les transitions jusqu'à une fréquence au plus égale à celle de l'horloge

```
architecture archi2 of exercice3 is
  signal prec: std_logic;
begin
  P1: process (clk, reset)
  begin
    if reset='1' then
      y <= "00000000";
      prec <= '0';
    elsif clk='1' and clk'event then
      if prec='0' and x='1' then
        y <= y + '1';
      end if;
      prec <= x;
    end if;
  end process P1;
end architecture archi;
```

- (b) (5 points) Si on rajoute le port suivant:

d: **out** std_logic;

Quelle modification apporteriez-vous pour que **d** monte à '1' lorsqu'il y a débordement du compteur?

*Le port **d** doit être raccordé à la retenue sortante de l'additionneur. On définit d'abord dans l'entête de l'architecture un signal intermédiaire d'un bit plus long que **y**:*

signal temp: unsigned (8 **downto** 0);

*On remplace ensuite l'assignation incrémentant **y** par la ligne suivante:*

temp <= y+'1'

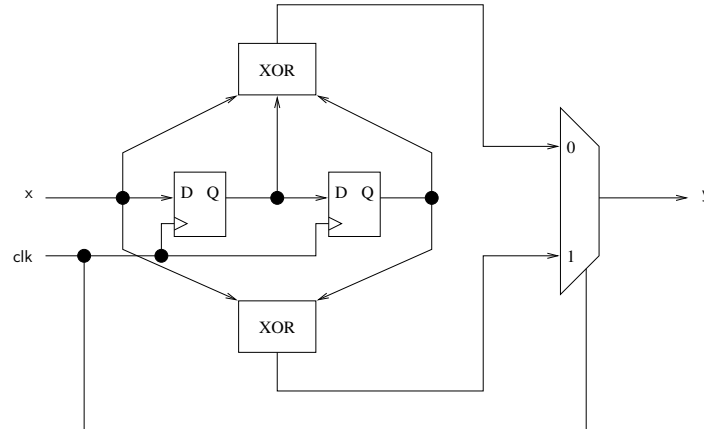
Dans le corps de l'architecture (zone parallèle), on rajoute:

d <= temp(8);

y <= temp(7 downto 0);

4. *Machines à état, conception* (40 points)

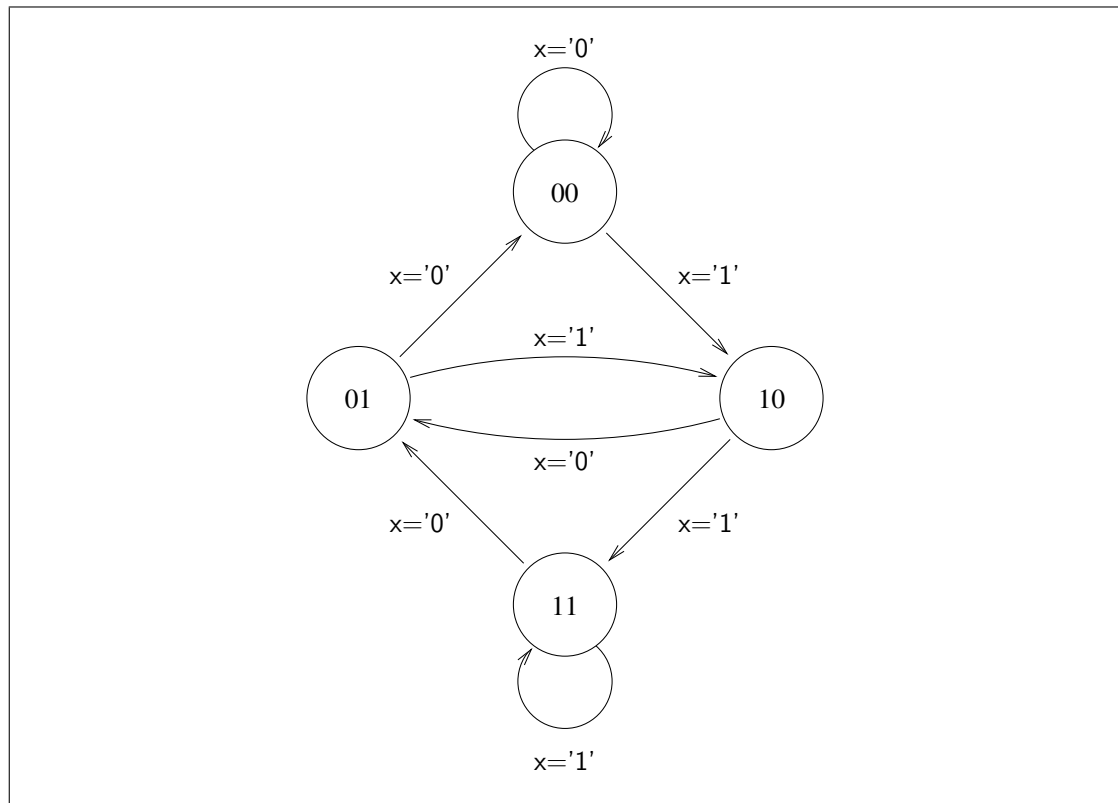
Voici le schéma d'un encodeur convolusionnel de taux $\frac{1}{2}$. Un tel circuit accepte des bits à l'entrée x à raison d'un bit par coup d'horloge, et restitue en sortie deux bits encodés par coup d'horloge.



- (a) (5 points) Ce circuit sans le multiplexeur de sortie peut être vu comme une machine à états. Est-ce une machine de Moore ou de Mealy? Justifiez.

C'est une machine de Mealy puisqu'il existe un chemin combinatoire direct, via les portes ou-exclusif, entre l'entrée et les sorties.

- (b) (15 points) Tracez le diagramme d'états de cette machine.



- (c) (20 points) À partir de l'entité suivante, écrivez une description VHDL pour l'encodeur

convolutionnel. Vous pouvez vous baser sur le diagramme d'états en (b) ou directement sur le schéma donné.

```
library ieee;  
use ieee.std_logic_1164.all;  
entity encodeur  
    port(x, clk, reset: in std_logic;  
        y: out std_logic );  
end entity encodeur;
```

```
architecture archi of encodeur is  
    signal chaine: std_logic(1 downto 0);  
    signal y1, y2: std_logic;  
begin  
    P1: process(reset, clk)  
    begin  
        if reset='1' then  
            chaine <= "00";  
        elsif clk='1' and clk'event then  
            chaine <= x & chaine(1);  
        end if;  
    end process P1;  
    y1 <= x xor chaine(1) xor chaine(0);  
    y2 <= x xor chaine(0);  
    P2: process(clk)  
    begin  
        if clk='1' then  
            y <= y2;  
        else  
            y <= y1;  
        end if;  
    end process P2;  
end architecture archi;
```

Bonne chance!