Nom : _____ Matricule : _____

INF3500 : Conception et réalisation de systèmes numériques

Examen intra - vendredi 11 octobre 2013

Durée: 2h; Pondération: 20%

Documentation: Une feuille 8.5”×11” ou A4 (recto-verso) permise.

Ordinateurs interdits. Calculatrice interdite. Appareils mobiles interdits.

Répondre à toutes les questions, la valeur de chaque question est indiquée.

Répondre sur le questionnaire et le remettre. Vous pouvez utiliser le verso du questionnaire si nécessaire. Ne détachez pas de pages du questionnaire.

Ne posez pas de questions. En cas de doute sur le sens d’une question, énoncez clairement toute supposition que vous faites.

Question 1. (4 points)

a. (0.5 point) Dans les laboratoires de ce cours, quelle(s) information(s) est/sont contenue(s) dans le fichier de type .ucf ? À quelle(s) étape(s) du flot le fichier .ucf est-il utilisé?

Solution :

Le fichier de contraintes (.ucf – *User Constraints File*) peut être utilisé par les processus de synthèse et d’implémentation. Dans le processus de synthèse, le fichier peut renfermer des contraintes de temps et de latence. Dans le processus d’implémentation, on peut en plus ajouter des contraintes physiques comme par exemple l’assignation de ports d’entrée-sortie à des pattes du FPGA.

b. (0.5 point) Selon votre expérience dans les laboratoires d’INF3500, énumérez trois métriques d’implémentation en termes de ressources matérielles utilisées.

Réponses acceptables :
tables de correspondances (LUT), bascules (FF), tranches (*slices*), multiplieurs, blocs d’entrée/sortie (IOB), mémoire BRAM

c. (0.5 point) Donnez 5 valeurs possibles d’un signal de type `std_logic` en VHDL, et expliquez la signification de ces cinq valeurs dans un circuit.

Réponses acceptables : il y a 9 valeurs possibles, voir les notes de cours

d. (0.5 point) Expliquez la différence entre les types VHDL `std_logic_vector`, `unsigned` et `signed`.

Réponse : les trois types correspondent à un vecteur de valeurs de type `std_logic`. La différence relève de l’interprétation de ces valeurs, respectivement : états électriques, valeurs numériques non-signées, valeurs numériques signées en complément à deux.

Question 2. (2 points)

Décrire un module combinatoire en VHDL synthétisable qui accepte en entrée un nombre non signé de six bits, de type `unsigned`, et qui indique si le nombre est divisible par sept à l'aide d'une sortie de type `std_logic`.

Solution

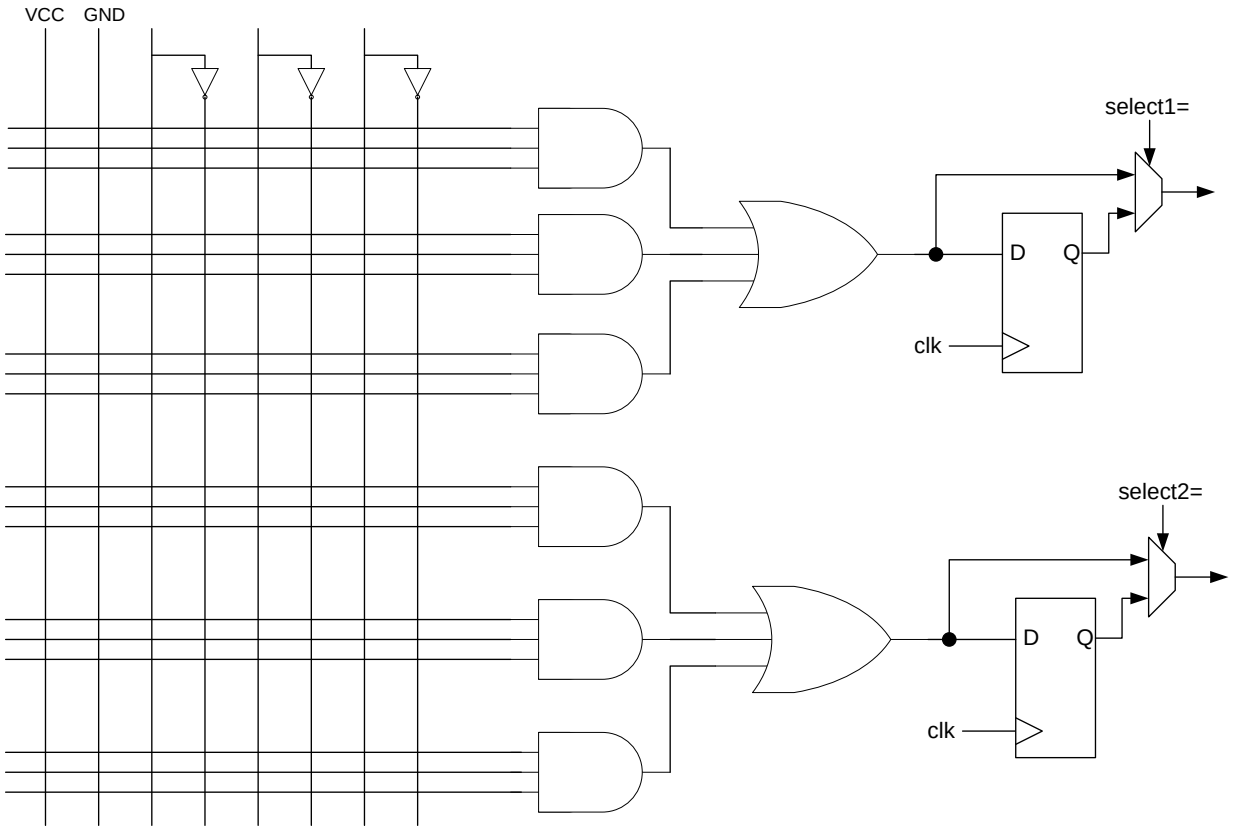
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity divpar7 is
port (
    nombre : in unsigned(5 downto 0);
    divpar7: out std_logic
);
end divpar7;

architecture arch1 of divpar7 is
begin
    with to_integer(nombre) select
        divpar7 <=
        '1' when 0 | 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63,
        '0' when others;
end arch1;
```

Question 3. (3 points)

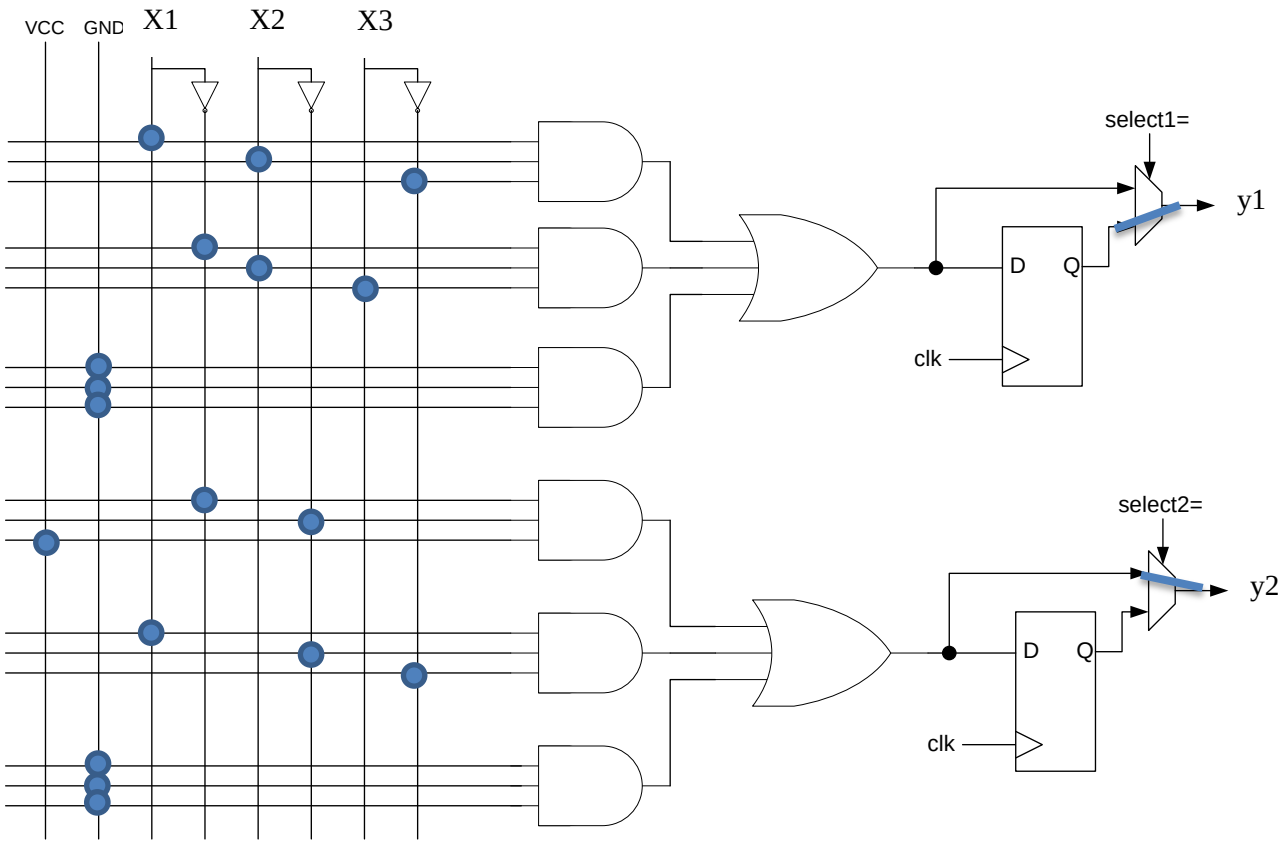
Soit le code VHDL suivant. Complétez le circuit PAL donné pour implémenter la fonctionnalité du module `combinatoire2`. Identifiez les lignes du PAL par des noms de ports et de signaux dans le code. Placez des boulets (•) aux intersections de lignes que vous voulez relier ensemble. Indiquez clairement comment programmer les deux multiplexeurs. Les lignes 'VCC' et 'GND' ont des valeurs logiques '1' et '0', respectivement.

```
library ieee;
use ieee.std_logic_1164.all;

entity combinatoire2 is
  port (
    clk : in std_logic;
    x1, x2, x3 : in std_logic;
    y1, y2 : out std_logic
    );
end combinatoire2;
```

```
architecture arch of combinatoire2 is

signal p1, p2, p3, p4 : std_logic;

begin

  y2 <= p3 or p4;
  p1 <= x1 and x2 and not x3;
  p2 <= not x1 and x2 and x3;
  p3 <= not x1 and not x2;
  p4 <= x1 and not x2 and not x3;

  process (clk)
  begin
    if rising_edge(clk) then
      y1 <= p1 or p2;
    end if;
  end process;

end arch;
```



Solution



Question 4. (2 points)

Tracer un diagramme d'un bloc de logique programmable d'un FPGA de la famille Virtex-5, comme celui que vous utilisez dans les laboratoires, montrant ses composantes principales. Indiquez comment les composantes sont reliées entre elles. Indiquez sur le diagramme ce qui est programmable et donnez quelques options de programmation.

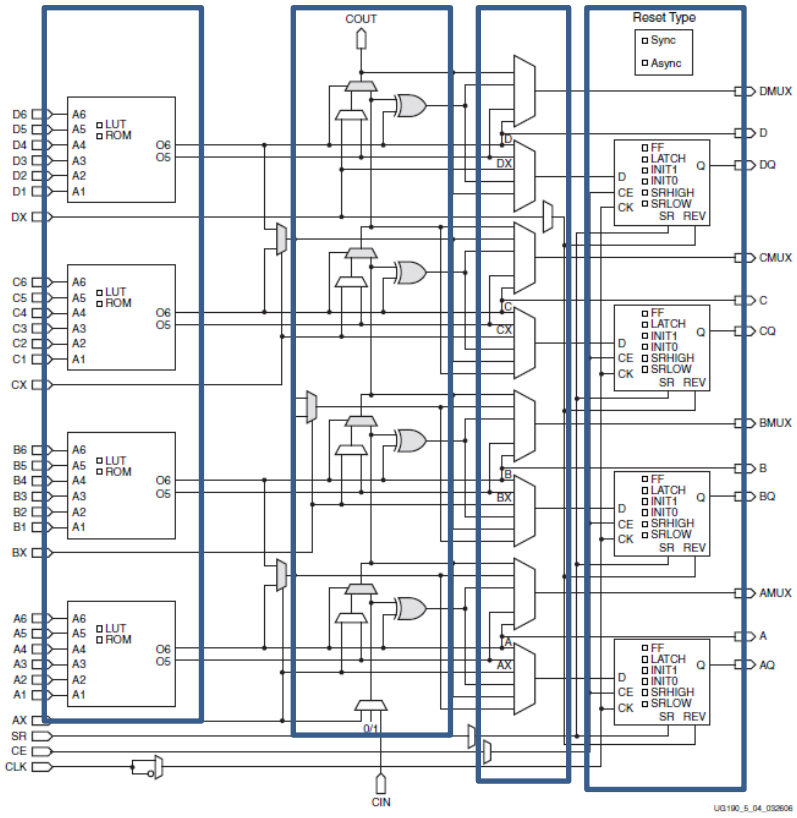
Solution :

Les blocs de logique programmables comprennent chacun deux tranches, de type L ou M.

Chacune des tranches comprend :

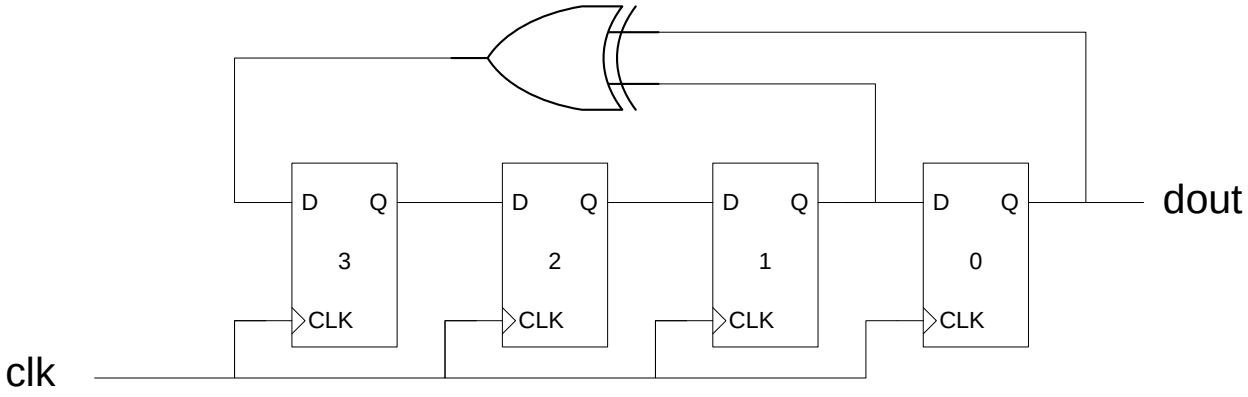
- 4 tables de correspondances (LUT) à 6 entrées chacune (programmables : fonction logique, mémoire ROM, mémoire RAM, registre à décalage)
- Un circuit de propagation de retenue pour l'addition rapide
- Des multiplexeurs de routage des signaux internes (programmables : pour router les signaux)
- 4 éléments à mémoire (programmables : type d'éléments, polarité de la réinitialisation, valeur de la réinitialisation)

Le diagramme suivant montre une tranche de type 'L'.



Question 5. (3 points)

Donnez une description VHDL du circuit suivant. L'entité prend en entrée le signal d'horloge `clk`, un signal de réinitialisation actif bas ('0') `rst` (qui alimente chaque bascule, mais qui n'est pas représenté sur le schéma) et produit en sortie le bit `dout`. L'initialisation doit être asynchrone et mettre le contenu des bascules $Q_3Q_2Q_1Q_0$ à l'état `0100`.



Solution possible:

```
library ieee;
use ieee.std_logic_1164.all;

entity lfsr4bits is
    port(
        clk, rst : in  std_logic;
        dout  : out std_logic
    );
end lfsr4bits;

architecture arch of lfsr4bits is
begin
    process(clk, rst)
    variable reg : std_logic_vector(3 downto 0);
    begin
        if rst = '0' then
            reg := "0100";
        elsif rising_edge(clk) then
            reg := (reg(1) xor reg(0)) & reg(3 downto 1);
        end if;
        dout <= reg(0);
    end process;
end arch;
```

Question 6. (4 points)

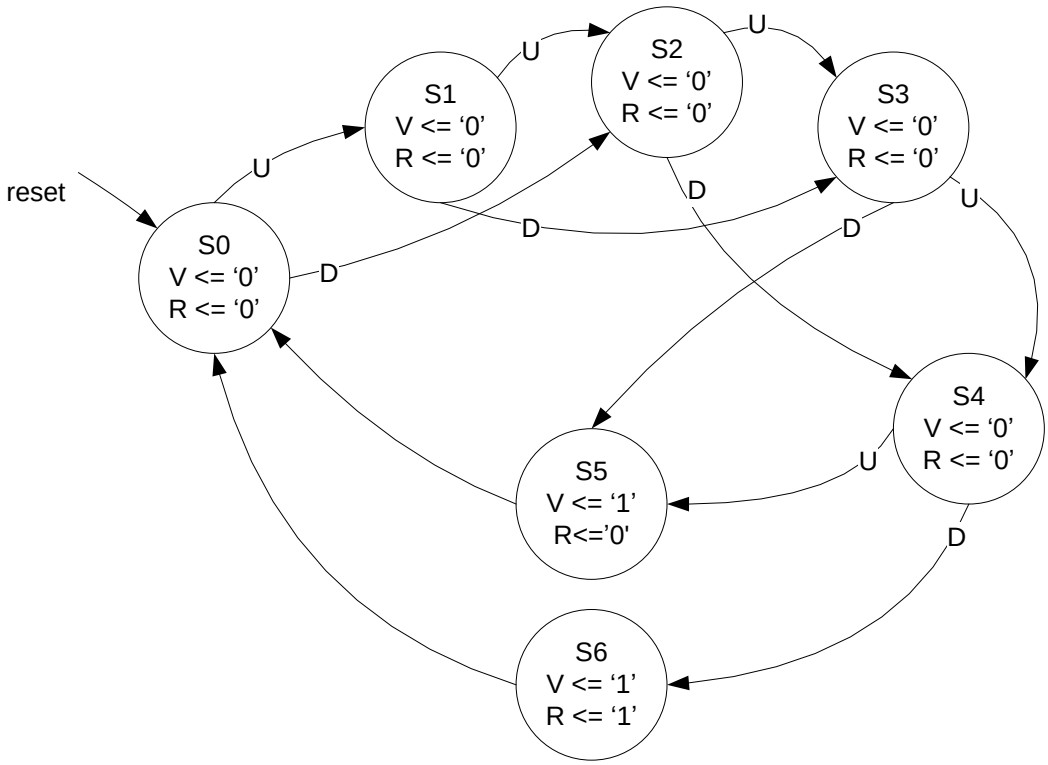
Une machine distributrice de sandwiches accepte uniquement les pièces de un dollar et de deux dollars. Le prix de tous les sandwiches est de cinq dollars.

Le contrôleur de la machine reçoit les entrées U (pour un dollar) et D (pour deux dollars) d'un détecteur de pièces. Les entrées U et D ont un niveau de '1' pendant un cycle quand une pièce correspondante est détectée. Il n'est pas possible d'avoir U et D à '1' simultanément dans un même cycle d'horloge.

Le contrôleur a les sorties V (pour vente), qui active l'actuateur de distribution d'un sandwiche, et R (pour rendre la monnaie), qui active le retour de la monnaie, une pièce de un dollar à la fois, si un montant total supérieur à cinq dollars a été déposé. Lorsque nécessaire, R et V doivent avoir un niveau de '1' pendant un cycle d'horloge.

Donnez le diagramme d'état d'une machine de Moore pour le contrôleur de la machine distributrice.

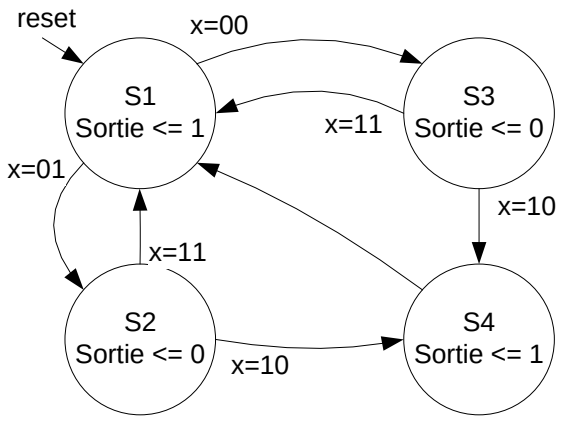
Solution



Question 7. (4 points)

Donnez une architecture pour l'entité suivante en VHDL synthétisable afin qu'elle corresponde au diagramme d'états donné. Utilisez un signal de réinitialisation asynchrone actif sur le niveau '0'.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity machineAEtats is
    port (
    reset, CLK : in STD_LOGIC;
    x : in STD_LOGIC_VECTOR(1 downto 0);
    sortie : out STD_LOGIC
    );
end machineAEtats;
```



Solution

```
architecture arch of machineAEtats is

type type_etat is (S1, S2, S3, S4);
signal etat : type_etat := S1;

begin

    process(CLK, reset) is
    begin
        if (reset = '0') then
           etat <= S1;
       elsif (rising_edge(CLK)) then
           case etat is
               when S1 =>
                   if x = "00" then
                       etat <= S3;
                   elsif x = "01" then
                       etat <= S2;
                   end if;
               when S2 | S3 =>
                   if x = "10" then
                       etat <= S4;
                   elsif x = "11" then
                       etat <= S1;
                   end if;
               when S4 =>
                   etat <= S1;
           end case;
       end if;
    end process;

    process(etat)
       begin
           case etat is
               when S1 | S4 =>
                   sortie <= '1';
               when S2 | S3 =>
                   sortie <= '0';
           end case;
       end process;

end arch;
```