

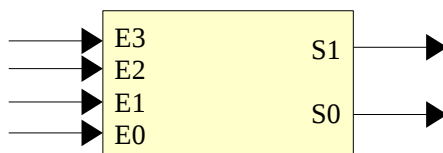
Médian

Durée 1h30, sans documents

1. Système combinatoire (7)

On considère le système combinatoire à 4 entrées et 2 sorties dont le cahier des charges est le suivant :

- S0 est à 1 si le nombre non signé sur 4 bit est supérieur strict à 9 ($S0 = 1$ si $E > 9$)
- S1 est à 1 si le nombre signé sur 4 bit est compris entre -4 et 4 ($S1 = 1$ si $-4 \leq E \leq 4$)



1. Donnez la description vhdl (entity/architecture) du système précédent

2. Donnez les équations logiques de S1 et S0

2. Systèmes séquentiels (7)

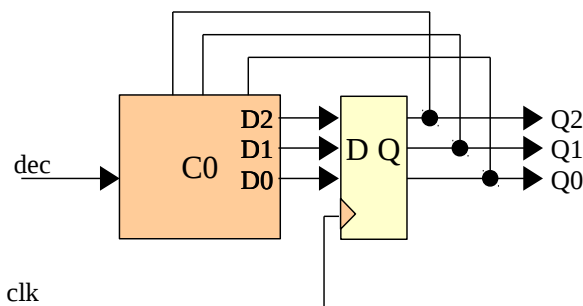
2.1. Compteur/décompteur par 5

1. Donnez la description VHDL d'un compteur/décompteur par 6 dont le cahier des charges est le suivant :

- lorsque $dec = 0$, la valeur en mémoire est incrémentée d'une unité sur le front montant du signal clk , le dispositif compte en boucle de 0 à 5
- lorsque $dec = 1$, la valeur en mémoire est décrétementée d'une unité sur le front montant du signal clk , le dispositif décompte en boucle de 5 à 0

2. Le compteur est réalisé à l'aide de 3 bascules D. Complétez la table de vérité ci-dessous et donnez les équations logiques de D2 (poids fort), D1, D0 (poids faible).

dec	Q2	Q1	Q0	D2	D1	D0

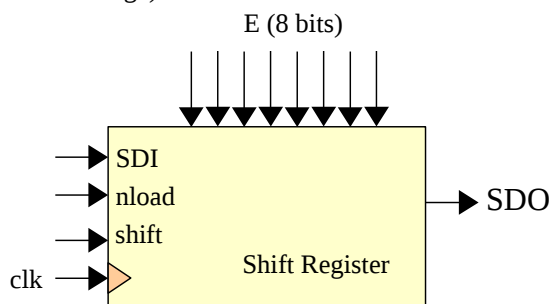


2.2. Convertisseur parallèle série (6)

1. Donnez la description VHDL d'un convertisseur parallèle série dont le cahier des charges est le suivant :

- lorsque $nload = 0$, le registre charge en mémoire la valeur présentes sur les entrées E.
- lorsque $shift = 1$, la valeur en mémoire est décalée d'un bit à droite (poids fort vers poids faible), la nouvelle valeur mémorisée sur le poids fort est la valeur présente sur l'entrée SDI.
- $nload$ est prioritaire sur $shift$.
- La sortie du composant, SDO, correspond au poids faible de la valeur en mémoire.

2. La mémoire est réalisée à l'aide de 8 bascules D. Donnez l'équation logique de l'entrée D0 de la bascule de poids faible (valeur pour le bit 0 à mémoriser au front d'horloge).



Correction

1. Exercice1

E	S0	S1
0000	0	1
0001	0	1
0010	0	1
0011	0	1
0100	0	1
0101	0	0
0110	0	0
0111	0	0
1000	0	0
1001	0	0
1010	1	0
1011	1	0
1100	1	1
1101	1	1
1110	1	1
1111	1	1

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity exo1 is
port(
    E : in std_logic_vector(3 downto 0);
    S : out std_logic_vector (1 downto 0)
);
end exo1;

architecture a of exo1 is
begin
    S(0) <= '1' when E > "1001" else '0';
    S(1) <= '1' when E >= "1100" and E <= "0100" else '0';
end a;

```

$$S0 = E_3 E_2 + E_3 E_1 \quad S1 = E_3 E_2 + \bar{E}_3 \bar{E}_2 + E_2 \bar{E}_1 \bar{E}_0$$

2. Exercice 2

dec	Q	D
0	000	001
0	001	010
0	010	011
0	011	100
0	100	101
0	101	000
0	110	xxx
0	111	xxx
1	000	101
1	001	000
1	010	001
1	011	010
1	100	011
1	101	100
1	110	xxx
1	111	xxx

```

entity exo2 is
port(
    clk, dec : in std_logic;
    Q : out std_logic_vector (2 downto 0)
);
end exo2;

architecture a of exo2 is
signal mem : std_logic_vector(2 downto 0);
begin
process(clk)
begin
    if rising_edge(clk) then
        if dec = '0' then
            if mem >= 5 then
                mem <= "000";
            else
                mem <= mem + 1;
            end if;
        else
            if mem = '0' then
                mem <= "101";
            else
                mem <= mem - 1;
            end if;
        end if;
    end if;
end process ;
Q <= mem;
end a;

```

$$D_0 = \overline{Q_0}, D_1 = \overline{dec} \overline{Q_2} \overline{Q_1} Q_0 + \overline{dec} Q_1 \overline{Q_0} + dec Q_2 \overline{Q_0} + dec Q_1 Q_0$$

$$D_2 = \overline{dec} Q_1 Q_0 + \overline{dec} Q_2 \overline{Q_0} + dec Q_2 Q_0 + dec \overline{Q_2} \overline{Q_1} \overline{Q_0}$$

3. Exercice 3

```

entity exo3 is
port(
    clk, SDI, nload, shift : in std_logic;
    E : in std_logic_vector (7 downto 0);
    SDO : out std_logic );
end exo3;

architecture a of exo3 is
signal mem : std_logic_vector (7 downto 0);
begin
process(clk)
begin
    if rising_edge(clk) then
        if nload = '0' then
            mem <= E;
        elsif shift = '1' then
            mem <= SDI & mem(7 downto 1);
        end if;
    end if;
end process;
SDO <= mem(0);
end a;

```

Avec Q_i : bits de la mémoire et D_i entrées de la mémoire :

$$D_0 = E_0 \overline{nload} + nload \overline{shift} Q_1 + nload \overline{shift} Q_0$$

$$D_1 = E_1 \overline{nload} + nload \overline{shift} Q_2 + nload \overline{shift} Q_1$$

...

$$D_7 = E_7 \overline{nload} + nload \overline{shift} SDI + nload \overline{shift} Q_7$$