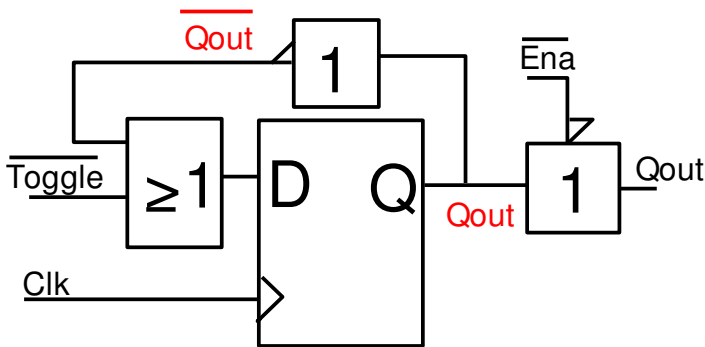


NOM : LeCorrecteur**Feuille réponse n°1****Prénom :**

Tout ce qui est en rouge, y compris dans les dessins est la correction et ne faisait pas partie de l'énoncé

DS MCENSL1 (Nov. 2011)

Problème 1 (variation sur l'exercice 1 du TD4)

On veut réaliser le circuit schématisé en figure 1 ci-contre dans une PAL 16R8 dont le schéma partiel est présenté ci-dessous en figure 2. On a simplement remplacé le ET de l'exercice original par un OU et "Toggle" par "/Toggle".

Figure 1: Schéma à implanter

1°) Trouver l'équation de récurrence à partir du schéma.

Réponse : $Q_{out}^+ = D = \text{/Toggle} + \text{/}Q_{out}$

2°) Pourquoi l'entrée d'autorisation de basculement s'appelle maintenant "/Toggle" ? Vous pouvez répondre en disant ce qui se passe pour /Toggle=0 et /Toggle=1.

Réponse : /Toggle=0 => Basculement, /Toggle=1 => $Q_{out} = \text{cste} = 1$

3°) Placez les croix correspondantes au schéma d'origine sur la figure ci-après. On rappelle que l'entrée de la bascule D ci-dessous est $\text{/}Q_{out}^+$

Réponse : $\text{/}Q_{out}^+ = D = \text{//Toggle} \cdot \text{//}Q_{out} = \text{Toggle} \cdot Q_{out}$ par DeMorgan

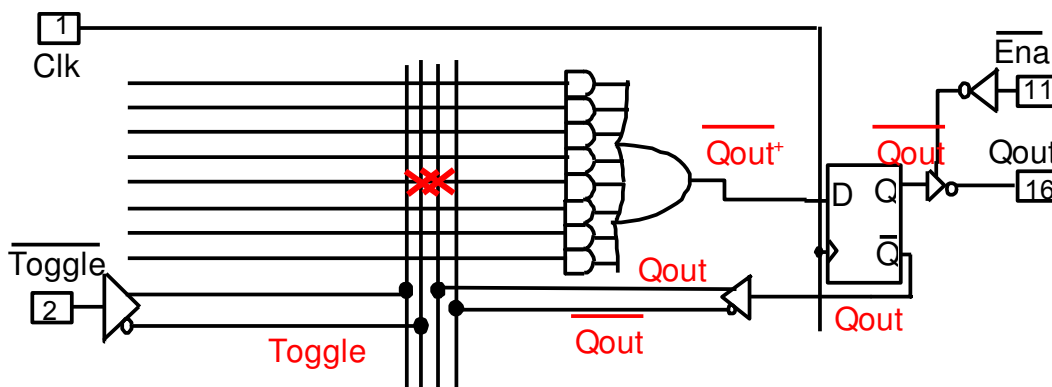


Figure 2: Ajouter les petites croix ici (schéma partiel d'une 16R8)

Problème 2 (conversion Binaire vers BCD)

Le but de ce problème est de réaliser une conversion binaire vers BCD. Ce type de conversion a été réalisé par les étudiants de deuxième année pendant l'année scolaire 2009/2010 en TP de MCENL1, mais on cherche dans ce problème à utiliser une autre méthode. La méthode de l'année dernière utilisait 255 fronts d'horloge pour convertir le nombre 255 tandis qu'on cherchera à le

Feuille réponse n°2

réaliser en 8 fronts d'horloge dans ce problème.

1°) Si l'horloge a une période de 50 MHz, combien de temps prenait l'ancienne conversion et combien de temps prend la nouvelle.

Réponse de la question 1°) :

$$T_H = 1/50M = 20ns, T_1=255 \cdot T_H = 5,1\mu s, T_2 = 8 \cdot T_H = 160 ns$$

2°) Le nombre binaire à convertir est dans un registre à décalage gauche (vers les poids forts) sur 8 bits. Comment modifier le listing du registre à décalage droit du TD, un peu modifié pour la circonstance, pour que le décalage soit à gauche et qu'il ait un chargement parallèle asynchrone.

```

1      -- registre à décalage
2      library IEEE;
3      use IEEE.STD_LOGIC_1164.ALL;
4      entity ShiftReg is
5          port(clk,load : in std_logic;
6              entrees : in std_logic_vector(7 downto 0);
7              q : out std_logic_vector(7 downto 0));
8      end ShiftReg;
9      architecture aShiftReg of ShiftReg is
10         signal dataq : std_logic_vector(7 downto 0);
11     begin
12         process(clk,load) begin

```

Réponse ici : `if load='1' then dataq <= entrees;`

```

13             elsif clk'event and clk='0' then
14                 -- dataq <= entree& dataq(7 downto 1);

```

Réponse ici : `dataq <= dataq(6 downto 0) & entree;`

```

15         end if;
16     end process;
17     process(dataq)begin
18         q<=dataq;
19     end process;
20 end aShiftReg;

```

3°) On rappelle que le code BCD d'un nombre consiste à coder chacun de ses chiffres sur 4 bits. On considèrera dans la suite tous nos nombres sur trois chiffres, c'est à dire codés en BCD sur 12 bits.

Par exemple, $(015)_{10}$ est codé $(0000\ 0001\ 0101)_{BCD}$, ce qui nécessite 12 bits (4 bits par chiffre).

Comment est codé le nombre $(249)_{10}$ en BCD ?

Réponse : `0010 0100 1001`

ALGO

Le point délicat de la conversion BCD proposée est un registre à décalage gauche BCD appelé SLR_BCD dans la suite. C'est évidemment un registre sur 12 bits. Il marche comme un registre à décalage normal sauf que lorsqu'un des DIGITs (sur 4 bit) est plus grand que neuf, on ajoute 6 au digit en question et un au digit de poids plus fort.

Dans un premier temps nous allons essayer d'expliquer pourquoi.

4°) Soit une valeur de $(0000\ 0001\ 0111)_{BCD}$, (soit $(017)_{10}$). Quelle valeur BCD puis décimale (ou hexadécimale) est obtenue par un décalage gauche si un zéro entre à droite ?

NOM :

Feuille réponse n°3

Prénom :

Réponse : il n'y a aucun piège dans la question, faite un décalage et remplacez les "?" par les valeurs trouvées

(0000 00?? ?????)BCD, (soit (0??)₁₀) ou (soit (0??)₁₆)

(0000 0010 1110)BCD soit (2E)₁₆

5°) En binaire un décalage vers la gauche d'une case équivaut à une multiplication par deux. Monter en appliquant ALGO (décrit ci-dessus) au nombre de la question 4°) qu'on trouve là aussi une multiplication par deux.

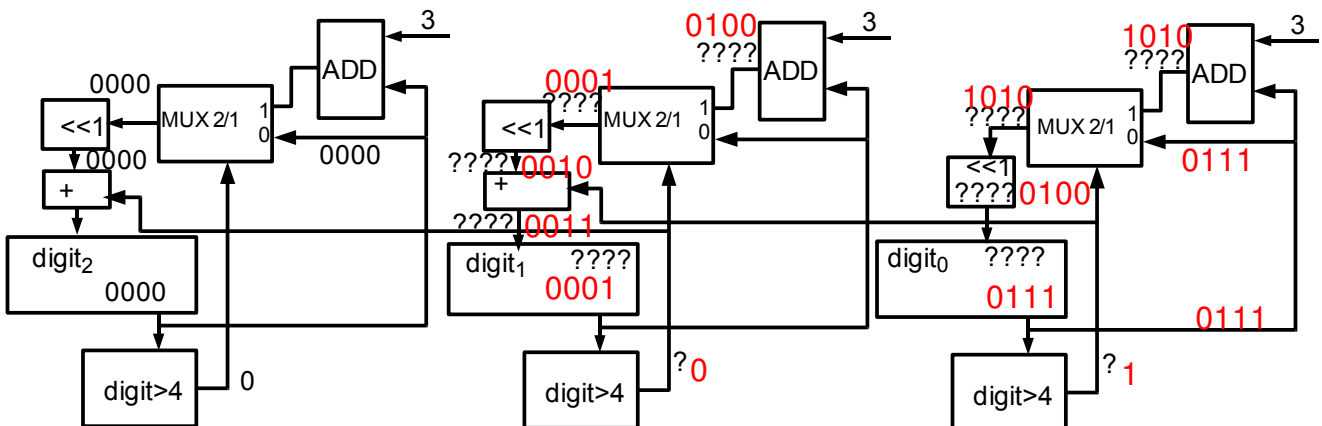
Réponse :

0 2 E (E>9 on lui ajoute 6 et 1 au digit suivant)

+0 1 6

=0 3 4 (E+6=14 mais le 1 du poids fort est perdu)

6°) On désire implanter ALGO de la manière suivante :



Dans ce schéma tous les fils sont en fait des bus de 4 fils sauf la sortie des boîtes **combinatoires** "digit>4" qui sont sur 1 bit. Les boîtes digit2, digit1 et digit0 sont des registres sur 4 bits.

Reprendre les valeurs de la question 4, soit (0000 0001 0111)BCD que l'on met dans digit2, digit1 et digit0, on vous demande de calculer les valeurs où sont marquées des points d'interrogations (1 par bit)

Réponse : sur le schéma ci-dessus

Indication : Ce qu'il y a dans digit2, digit1 et digit0 constitue l'état présent. Suivez la sortie de ces registres pour trouver en final ce qui va entrer dedans. La notation "<<1" signifie décalage vers la gauche d'un cran avec un 0 qui entre à droite. La notation + est réservée pour un addition 4 bits d'un côté et un bit de l'autre. La notation ADD signifie addition de deux nombres de 4 bits, donc 3 est à convertir en binaire sur 4 bits.

7°) Implanter en VHDL le composant combinatoire "digit>4" à l'aide d'une équation si possible simplifiée en commençant par remplir le tableau de Karnaugh

Feuille réponse n°4

Réponse :

	DC	00	01	11	10
00		0	0	0	0
01		0	1	1	1
11		1	1	1	1
10		1	1	1	1
		Dsup4			

Dsup4 est la sortie du comparateur dont l'entrée est $(D,C,B,A)_2$.

Réponse : entité et architecture en VHDL

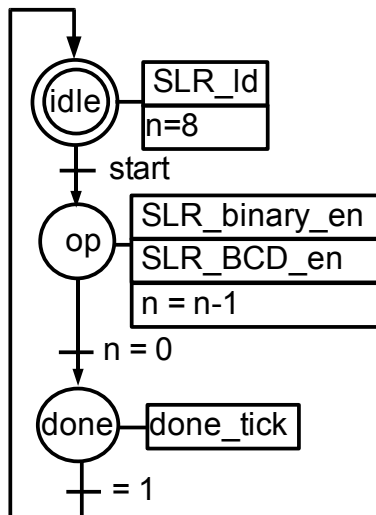
-- entité

```
entity digitsup4 is port(
  A,B,C,D : in std_logic;
  s : out std_logic);
end digitsup4;
```

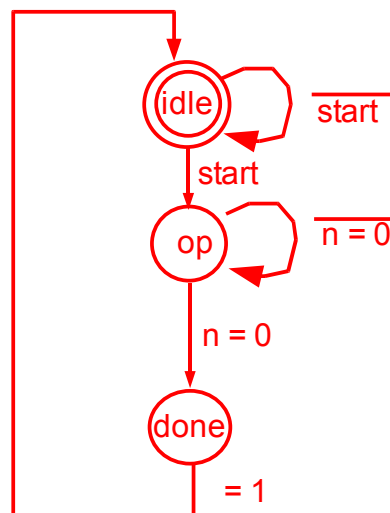
--architecture

```
architecture ads4 of digitsup4 is
begin
  s <= D OR (C AND A) OR (C AND B);
end ads4;
```

8°) Le séquencement de l'ensemble est spécifié par le graphe d'états ci-dessous. On vous demande de transformer ce graphe d'état en graphe d'évolution sans les actions.



Réponse : ci-dessous



9°) Combien de bits faut-il pour coder les trois états ? Combien de bits faut-il pour coder le décompteur n qui est initialisé à 8 ?

Réponses :

2 bits pour 3 états et 4 bits pour compter jusqu'à 8

NOM :**Feuille réponse n°5****Prénom :**

10°) En déduire le tableau état présent état futur et les équations de récurrences si l'on prend comme état futur de "11" l'état "00". On ne vous demande pas de tenir compte d'une entrée d'initialisation.

Réponse :

Etat présent		Conditions		Etat futur	
Q1	Q0	start	n==0	Q1+	Q0+
0	0	0	X	0	0
0	0	1	X	0	1
0	1	X	0	0	1
0	1	X	1	1	0
1	0	X	X	0	0
1	1	X	X	0	0

Réponses :

Les équations de récurrences sont :

$$Q1^+ = /Q1 . Q0 . (n==0)$$

$$Q0^+ = /Q1 . /Q0 . start + /Q1 . Q0 . /(n==0)$$

11°) Dans ALGO on vous explique que l'on ajoute 6 dès que le nombre est plus grand que 9.

Dans le schéma de la question 6°, c'est dès que votre nombre est plus grand que 4 que vous lui ajoutez 3. Le 9 s'est transformé en 4 et le 6 en 3. Pouvez-vous expliquer pourquoi ?

Réponse :

Parce qu'on fait les tests avant les décalages :

$$9 \gg 1 = 4$$

$$6 \gg 1 = 3$$