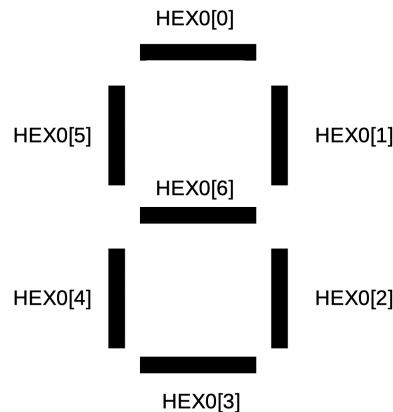


Examen de TP : électronique numérique

Durant cet examen, on utilisera **uniquement** les libraires

- *IEEE.STD_LOGIC_1164.ALL*
- *IEEE.NUMERIC_STD.ALL*

On rappelle que les sorties du premier afficheur 7 segments sur la carte DE0 sont noté/codé avec les affectations *HEX0[a]* avec $a = 0, 1, 2, 3, \dots, 6$ de la manière suivante :



Exercice 1 (45 minutes)

Soient A un nombre en binaire signé en complément à 2 sur 6 bits et B un nombre en binaire signé en complément à 2 sur 4 bits. Les deux nombres représentent des entiers relatifs. On cherche à réaliser un additionneur entre ces deux nombres. On souhaite qu'il n'y ait pas de dépassement, la sortie, notée S , sera donc codée sur 7 bits.

Avant de réaliser l'addition, on souhaite « étendre »¹ le nombre B sur 6 bits.

1. Montrer que $(0111)_{C2} = (000111)_{C2}$ et $(1101)_{C2} = (111101)_{C2}$. À la lumière de ces exemples, montrer comment passer d'un nombre binaire signé en complément à 2 sur 4 bits au même nombre sur 6 bits en utilisant le bit de poids fort du nombre sur 4 bits.

On rappelle que si X est un *std_logic_vector*, alors $X(i)$ représente le i ème bit (il s'agit donc d'un *std_logic*).

2. Effectuez la description de cet additionneur par une approche mixte Schématique/VHDL. L'entity comportera obligatoirement 2 *std_logic_vector* de tailles 6 et 4 représentant respectivement les nombres A et B en entrée et un *std_logic_vector* de taille 7 représentant la sortie S . On utilisera les 10 switch SW comme entrées ($SW[5]$ à $SW[0]$ pour A et $SW[9]$ à $SW[6]$ pour B). Par manque de temps, une conception itérative de l'additionneur est **interdite** ici, on utilisera **obligatoirement** la fonction « + » avec les opérateurs de conversions de types. Il n'est pas demandé d'afficher le résultat, donc il n'est pas nécessaire à ce stade de téléverser la description sur la carte. Il n'est pas non plus nécessaire d'utiliser un *process*.

3. À partir du résultat du sommateur, compléter votre description pour afficher sur un afficheur 7 segments si la valeur absolue de la somme S , dans sa représentation décimale signée, est un nombre pair ou impair. On allumera les sorties *HEX0[1]* et *HEX0[2]* de l'afficheur si le nombre est impair et les sorties *HEX0[0]*, *HEX0[1]*, *HEX0[4]*, *HEX0[5]* et *HEX0[6]* si le nombre est pair.

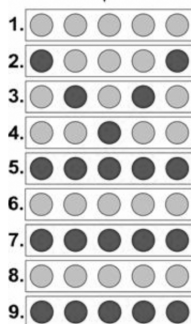
1. Pour additionner deux nombres en binaire signé en complément à 2, il est crucial de s'assurer que les deux nombres ont la même longueur en bits pour que l'addition soit correcte. On peut s'en convaincre facilement sur un exemple.

Exercice 2 (30 minutes)

On veut réaliser un petit jeu à l'aide d'un compteur en VHDL. On sélectionne avec les switch $SW[0]$ à $SW[2]$ un nombre codé en binaire naturel sur 3 bits (le bit de poids fort est $SW[2]$). Un compteur compte de 0 à 7 de manière cyclique avec une fréquence de 4 Hertz et affiche le nombre sur un afficheur 7 segments. Lorsqu'on met le switch $SW[3]$ à 1, on met le compteur en pause. Si le nombre affiché par le compteur est égal au nombre sélectionné avec les switch $SW[0]$ à $SW[2]$ alors on gagne et on fait clignoter toutes les LEDs à la fréquence du compteur (4 Hertz avec un rapport cyclique de 50%), sinon, on perd et on allume toutes les LEDs (sans les faire clignoter). Lorsqu'on joue, le switch $SW[3]$ est à 0 et toutes les LED sont éteintes. A l'aide de vos connaissances, réaliser cette description par une approche mixte Schématique/VHDL. On utilisera **obligatoirement** une seule description VHDL pour le compteur/comparateur, une description VHDL séparée pour la gestion de l'afficheur 7 segments et une troisième description VHDL séparée pour la gestion des LEDs.

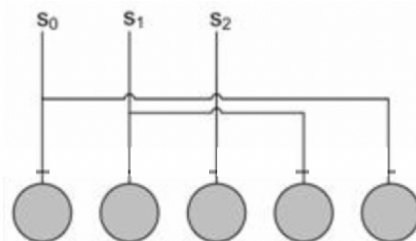
Exercice 3 (45 minutes)

On dispose de 5 lampes avec lesquelles on veut réaliser un effet « Disco ». On pose les cinq lampes les unes à côté des autres, et veut obtenir le résultat suivant représenté par 9 états (les lampes allumées sont en noir et les lampes éteintes sont en grises) :



Lorsqu'on arrive à l'état 9, on recommence à l'état 1. En plus de cette séquence, on ajoute un bouton B à ce système qui permet de le mettre en pause au dernier état rencontré lorsque $B = 1$. Le système entier est synchrone et on se propose de le synthétiser à l'aide d'une machine de Moore. On utilisera un switch de la carte DE0 pour B et une fréquence d'horloge judicieusement choisie pour cadencer le système.

Ce système utilisera 3 sorties s_0 , s_1 et s_2 seulement en utilisant le circuit ci-dessous.



1. Donner le diagramme d'état d'un tel circuit sous forme de machine de Moore.
2. Réaliser cette description par une approche mixte Schématique/VHDL en utilisant obligatoirement la fonction *type* pour représenter les états et trois *process* pour représenter les trois blocs de la machine de Moore (calcul des états suivants, mémorisation par registre d'état et calcul des sorties).