

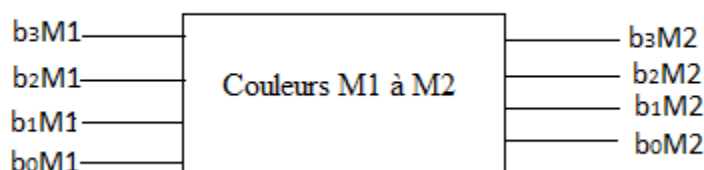
IFT 1227 – Architecture des ordinateurs

Devoir 2

- Remise : Le 9 mars 2023 avant minuit au plus tard.

1. Conception des circuits combinatoires (VHDL)

Réaliser un circuit *colorsM1toM2*, « Couleurs M1 à M2 », qui permet de convertir les codes des couleurs sur une machine vers les couleurs sur une autre. Le tableau sur la figure 1 montre le code sur 4 bits $b_3b_2b_1b_0$ des 16 couleurs affichables par une machine M1 et le tableau de la figure 2 – codes des couleurs de la machine M2. On convertit seulement les mêmes couleurs + les codes de la machine M1 Gris foncé et Gris Moyen on associe aux couleurs Noir pâle et Gris de la machine M2. Pour les autres couleurs de M1, on génère comme sorties « Don't care » ou la couleur Noir de la machine M2.



b_3M1	b_2M1	b_1M1	b_0M1	Couleurs
0	0	0	0	Noir
0	0	0	1	Blanc
0	0	1	0	Rouge
0	0	1	1	Cyan
0	1	0	0	Violet
0	1	0	1	Vert
0	1	1	0	Bleu
0	1	1	1	Jaune
1	0	0	0	Orange
1	0	0	1	Brun
1	0	1	0	Rose
1	0	1	1	Gris foncé
1	1	0	0	Gris Moyen
1	1	0	1	Vert pâle
1	1	1	0	Bleu pâle
1	1	1	1	Gris pâle

Figure 1. Codage des couleurs de la machine M1.

b_3M2	b_2M2	b_1M2	b_0M2	Couleurs
0	0	0	0	Noir
0	0	0	1	Bleu
0	0	1	0	Vert
0	0	1	1	Cyan
0	1	0	0	Rouge
0	1	0	1	Violet
0	1	1	0	Brun
0	1	1	1	Gris
1	0	0	0	Noir pâle
1	0	0	1	Bleu pâle
1	0	1	0	Vert pâle
1	0	1	1	Cobalt
1	1	0	0	Rose
1	1	0	1	Mauve
1	1	1	0	Jaune
1	1	1	1	Blanc

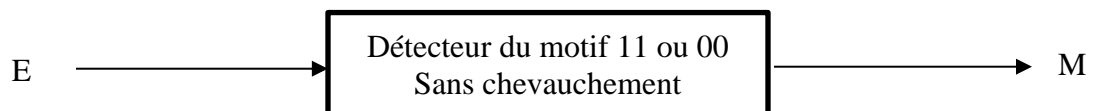
Figure 2. Codage des couleurs de la machine M2.

Implémenter le circuit en utilisant les différentes méthodes d'implémentations :

- Modélisation comportementale ;
- Modélisation structurelle : utilisez 4 multiplexeurs 8 à 1 pour réaliser les fonctions logiques b_3M2 , b_2M2 , b_1M2 , b_0M2 . Implémentez d'abord le multiplexeur 3 à 1 (modèle comportemental), par la suite, utilisez ce composant pour créer le modèle structurel de votre circuit. Le code VHDL du ***mux8*** mettez dans le répertoire ***files*** créé dans le répertoire de votre projet du modèle structurel.
- Synthétisez les 2 modèles et vérifiez le comportement en connectant les entrées du circuit aux interrupteurs SW[3] – SW[0] et les sorties aux lumières rouges : LEDR[3] – LEDR[0].

I. Circuits logiques séquentiels

Concevoir un circuit séquentiel qui prend une entrée en série : E et génère 1 en sortie M lorsque les 2 motifs se présentent comme 2 derniers bits observés (sans chevauchement).



Exemple du comportement :

E : 011110000101...

M : 001010101000...

Temps : —————→

- a) Créer le diagramme d'états;
- b) Codez la machine en VHDL ;
- c) Synthétisez le circuit. Utilisez le signal **reset_n** (reset négatif) pour initialiser votre FSM. Vérifiez le comportement du circuit en connectant les entrées **reset_n** et **CLK** aux pins des boutons poussoirs, KEY[0] et KEY[1] respectivement, E – SW[0], M - LEDR[0].

Vérifiez le comportement de votre machine. Le fonctionnement du circuit est commandé par une horloge. Chaque événement sera pris en compte en appuyant le bouton « Horloge », KEY[0] (on modélise une horloge de manière manuelle). Pour pouvoir utiliser le bouton poussoir sur la carte pour le signal **reset**, il faudra plutôt utiliser un reset négatif :

```
process (clk, reset_n) begin
    if reset_n = '0' then state <= État initial;
```

Puisque on modélise l'horloge avec le bouton, il est normal que la machine pourrait sursauter certains états.

Remise

La remise électronique : Créer un fichier compressé (*.zip) contenant les 3 répertoires de vos 3 projets + le fichier pdf du rapport.

Le rapport doit contenir :

- q.1(a), q.1(b) - images des circuits synthétisés (netlist);
- q.2.a – le diagramme de transition d'états (FSM) de votre machine;
- q.2.b – schéma du circuit synthétisé ;

Le rapport pourrait être écrit à la main (version scannée doit être convertie en format pdf) ou par traitement de texte. Le rapport devra être clair, **présentable** et concis en format pdf.

Barème de correction

Numéro 1 – 40 pts; 1a – 15; 1b – 15; 1c – 10.

Numéro 2 – 45 pts; 2a –15; 2b –15; 2c – 15;

Présentation de rapport : 15 – propreté, lisibilité, respect des directives etc.

Total : 100 points

Bon travail!