

Jeu de conversion d'une valeure décimale à une valeure binaire

Zakaria TBER
Alexis TREILLES

Mai 2023



Encadrant :
JEGO Christophe

Table des matières

1	Introduction	3
2	Cahier des charges	3
3	Déscription des différents modules VHDL	4
3.1	Compteur Modulo 10	4
3.2	Compteur du score	4
3.3	Registres	4
3.4	Module de génération du nombre entre 0 et 99	5
3.5	Comparateur	5
3.6	Transcodeur	5
3.7	Contrôleur sept segments	6
3.7.1	GestionCE	7
3.7.2	Compteur modulo 3	7
3.7.3	Multiplexeur	7
3.8	Machine d'état	8
4	Conclusion	8
5	Résultat final	9

1 Introduction

Ce rapport présente la conception d'un système électronique mettant en œuvre un affichage de score et des afficheurs 7 segments pour représenter des nombres décimaux. Lors de l'initialisation, le système affiche un score initial de 0 sur l'afficheur 3 et la valeur "00" sur les afficheurs 0 et 1. Le système est ensuite capable d'afficher des nombres décimaux aléatoires sur les afficheurs 0 et 1 à l'aide du bouton BTNC.

L'utilisateur peut ensuite utiliser les interrupteurs (switchs) pour positionner une valeur binaire qu'il estime correspondre à la valeur décimale affichée. En appuyant sur le bouton BTND, le système compare la valeur binaire proposée par l'utilisateur à la valeur binaire calculée à partir du nombre décimal affiché. Si les deux valeurs binaires sont identiques, le score est incrémenté de 1. Dans le cas contraire, le score reste inchangé.

Ce système nécessite une architecture électronique comprenant des afficheurs 7 segments, des boutons et des interrupteurs. Les afficheurs sont utilisés pour l'affichage des nombres décimaux et du score, tandis que les boutons permettent de contrôler les différentes étapes du jeu. Les interrupteurs sont utilisés par l'utilisateur pour proposer une valeur binaire.

Le rapport détaillera les composants électroniques utilisés, les schémas de câblage, les spécifications des afficheurs et des boutons, ainsi que les résultats des tests effectués. L'objectif principal est de présenter une conception fonctionnelle et robuste du système, en mettant l'accent sur les aspects électroniques et les choix de conception pertinents.

2 Cahier des charges

Permettre à l'utilisateur de tester ses connaissances en conversion d'un nombre binaire à un nombre décimal. Pour cela la carte doit afficher un nombre décimal sur deux chiffres (de 0 à 99) puis permettre à l'utilisateur de saisir le nombre binaire correspondant, et d'augmenter le score si la réponse de l'utilisateur est correcte.

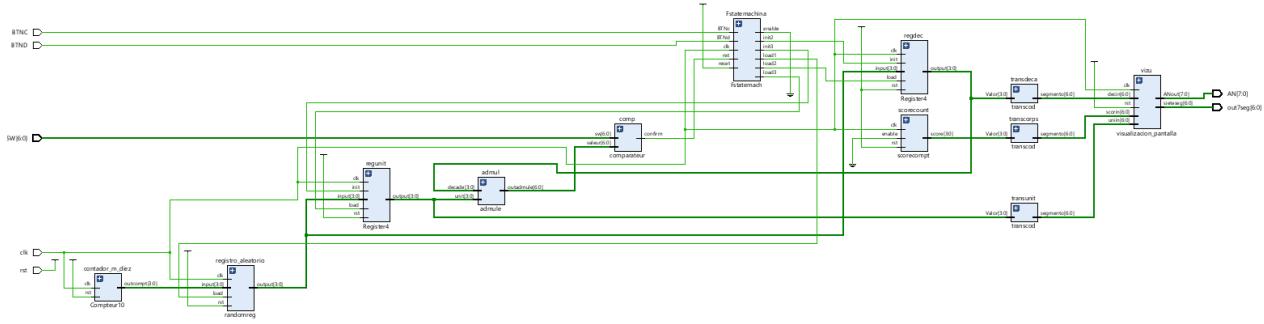


Figure 1: Découpe architecturale

3 Description des différents modules VHDL

3.1 Compteur Modulo 10

Le compteur modulo 10 reçoit en entrée le signal de l'horloge, le reset, et en sortie ce module compte de 0 à 9 à chaque front montant de l'horloge. Pour cela on utilise le process ci-dessous qui permet d'incrémenter à chaque front montant de l'horloge le signal count-val, qui va être mis en sortie du compteur.

```

1 counter_value_register : process(rst, clk)
2   begin
3     if(rst = '1') then
4       count_val <= (others => '0');
5     elsif(clk'event and clk = '1') then
6       count_val <= count_val + 1;
7     end if;
8     if(count_val = "1001") then
9       count_val <= (others => '0');
10    end if;
11 end process counter_value_register;

```

Listing 1: Process du compteur modulo 10

3.2 Compteur du score

L'architecture de module est similaire au compteur modulo 10, cependant on ajoute l'entrée enable qui permet de déclencher l'incrémantation du compteur.

3.3 Registres

Les registres permettent de conserver la valeur d'entrée en sortie pendant une période de l'horloge. Cette opération s'effectue uniquement si la valeur de load est mis à '1'. Pour exprimer cette fonctionnalité en langage VHDL, il faut utiliser un process assez simple qui permet de vérifier que load est bien mis à '1', puis d'affecter à la sortie du registre la valeur à son entrée.

```

1 process (clk, rst)
2   begin
3     if rst = '1' then
4       output <= (others => '0');
5     elsif (clk'event and clk = '1') then
6       if load = '1' then

```

```

7           output <= input;
8       end if;
9   end if;
10  end process;

```

Listing 2: Process du Registre

On a utilisé un process similaire pour le registre qui donne en sortie une valeur aléatoire entre 0 et 9, et pour les registres qui génère les chiffre d'unités et de dizaines, la seule différence étant l'ajout de l'entrée init, qui permet de remettre à 0 les valeurs en sorties de ces registres.

3.4 Module de génération du nombre entre 0 et 99

Pour générer le nombre il faut multiplier le chiffre du registre qui génère un chiffre pour les dizaines par 10, et de l'ajouter à la valeur en sortie du registre qui génère un chiffre des unités.

```

1 s_outamule <= resize(s_decade * to_unsigned(10, 4), s_outamule'length) + ns_unit;

```

Listing 3: Affectation du signal de sortie su module aux valeurs liés aux unités et dizaines issue des registres

3.5 Comparateur

La description VHDL de ce module consiste comparer entre la valeur en binaire du nombre affiché à l'utilisateur et la valeur saisie par ce dernier. Si ils sont égaux, la sortie du comparateur vaut '1' et sert à incrémenter le score. Sinon la sortie est mise à '0', et on permet à l'utilisateur de réessayer.

```

1 process(sw, valeur)
2 begin
3     if( valeur = sw) then
4         confirm <= '1';
5     else
6         confirm <= '0';
7     end if ;
8 end process;

```

Listing 4: Process du comparateur

3.6 Transcodeur

Le transcodeur sert à traduire le chiffre en binaire (de 0 à 9) à écrire sur l'écran sept segments, en un code binaire de 7 bits qui commande les leds du sept segments, allumé ('0') ou éteinte ('1'), afin de pouvoir afficher le chiffre souhaité.

```

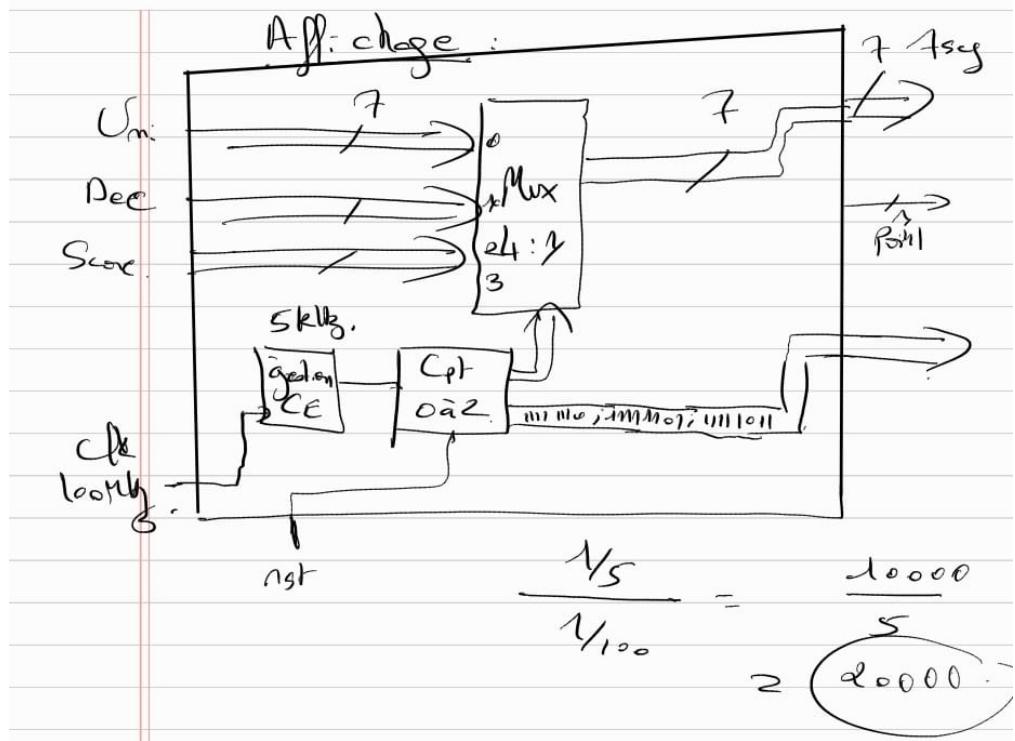
1 0000 => "1000000"
2 0001 => "1111001"
3 0010 => "0100100"
4 0011 => "0110000"
5 0100 => "0011001"
6 0101 => "0010010"
7 0111 => "1111000"
8 1000 => "0000000"
9 1001 => "0010000"

```

Listing 5: Le code binaire sept bits correspondant à chaque chiffre

3.7 Contrôleur sept segments

Ce module permet d'orienter le code binaire sept bits vers l'écran sept segment correspondant. Il est constitué de trois module : Gestion CE, qui adapte le signal de l'horloge à 5Khz, un multiplexeur qui a en sortie la valeur du score, des unité ou dizaine. Un compteur modulo 3 qui permet de donner les 3 valeurs d'entrée au multiplexeur.



3.7.1 GestionCE

Ce module est un compteur qui compte jusqu'à la valeur 20000 pour chaque front montant de l'horloge, puis met sa sortie à '1', cela permet d'obtenir un signal d'horloge de fréquence 5Khz, cette manipulation est nécessaire pour que l'affichage soit visible par l'oeil à cause de la persistance rétinienne.

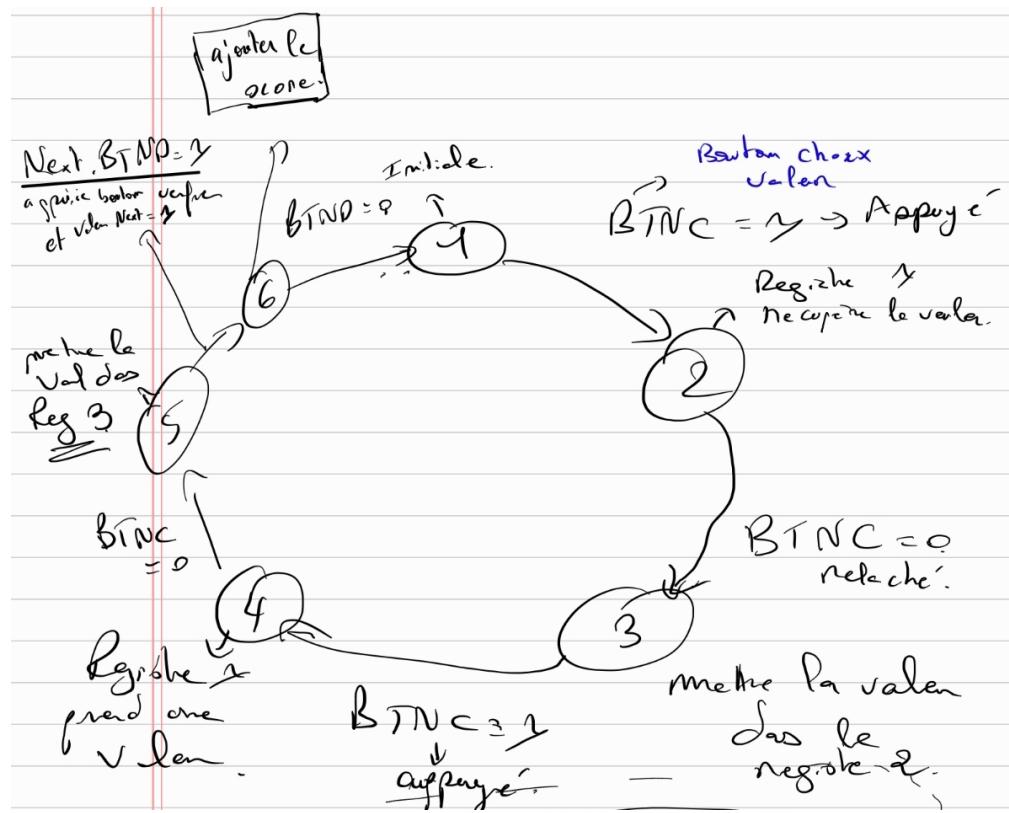
3.7.2 Compteur modulo 3

Le compteur modulo 3 est similaire au compteur modulo 10 décrit précédemment, seule la condition d'arrêt est mis à la valeur 3 cette fois-ci.

3.7.3 Multiplexeur

Le multiplexeur permet de mettre une des valeur à son entré en sortie, et cela selon le signal de commande, qui permet de choisir quelle valeur en entrée mettre en sortie.

3.8 Machine d'état



La machine d'état comporte 6 états différents :

- L'état initial ou aucune valeur n'est tiré
- Les états de 2 à 5 concernent les appuis pour charger les valeurs aléatoires dans les registres
- Le dernier état attend que la valeur rentrée par les switchs correspondent à la valeur binaire.

4 Conclusion

Dans ce Projet, on a eu quelques difficultés avant de réussir. Au début, on a décidé d'écrire l'architecture des différents modules sans faire de testbench, et procéder à l'utilisation d'un test bench global pour le top level. Une fois les différents module décrit en VHDL, on a procédé à la résolution d'erreurs, et on a généré notre bitstream, on a testé dabord notre affichage sur la carte qui a parfaitement fonctionné, mais après l'élaboration du toplevel, on a pas eu le résultat souhaité sur la carte, la correction d'erreur et la validation par la simulation n'a pas aussi réglé notre problème, puisqu'on a remarqué juste après notre dernière séance, le fait que au niveau du multiplexeur on avait affiché des valeurs constantes qu'on avait mise pour tester l'affichage. Cette erreur qu'on a pas pu remarqué durant la dernière séance ne nous a malheureusement pas permis de tester ce qu'on a validé en simulation sur la carte, en travaillant hors horaire de cours on a pu constater le problème et une fois résolu on a pu faire fonctionner, le jeu de conversion sur la carte.

5 Résultat final

