

TP4 – Pilotage de LED et mémoire

Partie 1

Rendu

Votre rapport devra contenir :

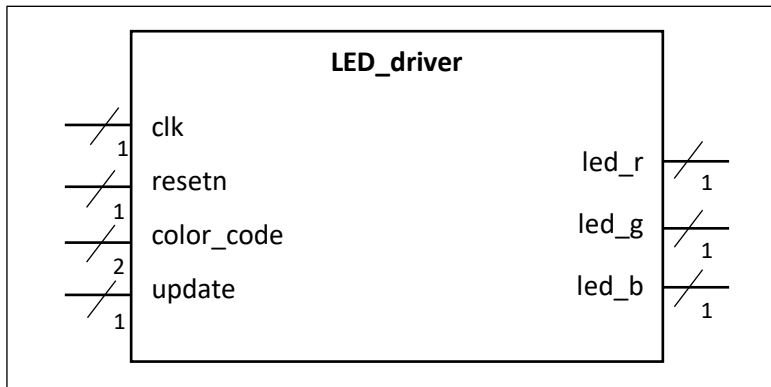
- Vos schéma RTL
- Vos résultats de simulation avec vos chronogrammes commentés
- Vos résultats de synthèse (analyse de vos ressources)
- Vos résultats de STA (analyse du rapport de timing)
- Une démonstration de votre design

Vous fournirez également vos codes source commentés.

Questions

1. Créez une architecture RTL permettant de faire clignoter une LED (par exemple la led0_r) en utilisant le module *Counter_unit* du TP2 et une machine à état.
2. Modifiez votre architecture pour piloter une LED rouge et une LED verte. Lorsque le bouton_0 est appuyé, la LED verte est allumée, sinon la LED rouge est allumée.
3. Rédigez le code VHDL de votre architecture.
4. Réalisez une simulation en rédigeant un testbench. Que se passe-t-il si le bouton est pressé pendant plus d'un cycle d'horloge ?
5. Que faudrait-il faire pour que la LED ne clignote en vert qu'une seule fois même si le bouton est maintenu ?
6. Ajoutez cette solution à votre architecture RTL.
7. Mettez à jour votre code VHDL et vérifiez votre résultat à la simulation.

8. Créez un module de pilotage d'une LED RGB en RTL. Ce dernier doit permettre de faire clignoter une LED RGB connectée en sortie d'une couleur définie par un code couleur donné en entrée. Le changement de couleur de la LED RGB n'a lieu que si un signal *update* est reçu.



A titre d'exemple voici une table mettant en relation un code couleur avec une couleur de la LED RGB :

Code couleur	Couleur de la LED RGB
01	Rouge
10	Verte
11	Bleue
00	Eteinte

9. Ajouter la logique nécessaire pour piloter les entrées/sorties de votre module.

- Le signal *update* doit recevoir 1 uniquement lorsque le bouton _0 vient d'être appuyé, maintenir le bouton enfoncé ne doit pas maintenir le signal *update* à 1
- Le signal *color_code* doit recevoir soit le code couleur « vert » si le bouton_1 est pressé, il doit recevoir le code couleur « bleu » sinon
- Les LED R, G et B sont connectées avec les couleurs respectives de la LED_0

10. Ecrivez le code VHDL correspondant à votre architecture.

11. Ecrivez un testbench pour tester votre design.

12. Vérifier votre résultat à la simulation.

13. Réalisez une synthèse et étudiez le rapport de synthèse, les ressources utilisées doivent correspondre à votre schéma RTL.

14. Effectuez le placement routage et étudiez les rapports.

15. Générez le bitstream et vérifiez que vous avez le comportement attendu sur carte.