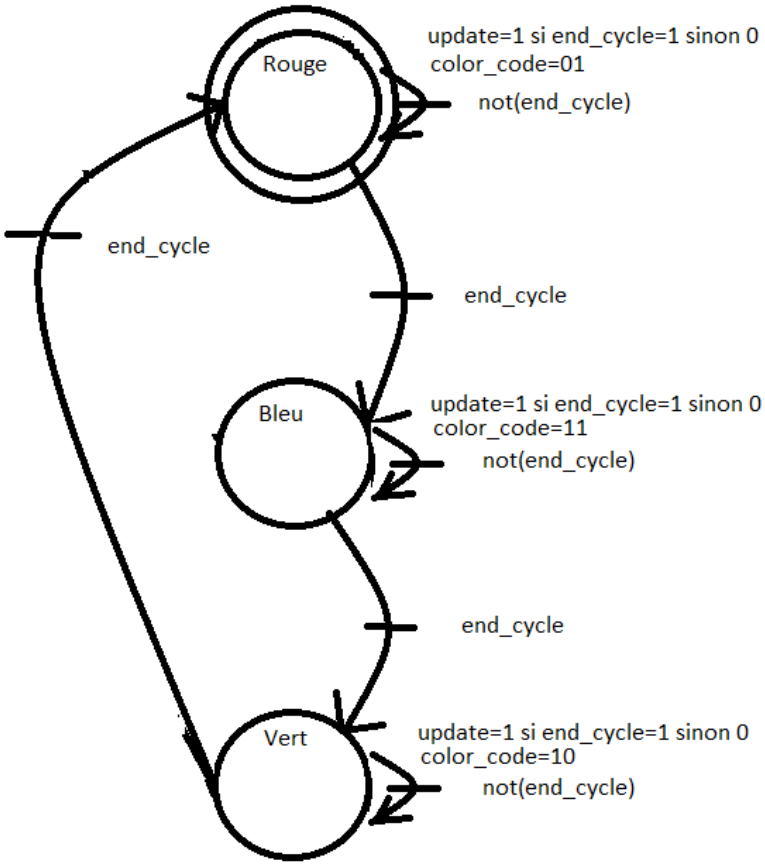
TP5

Dang Thi HUONG
et
Jean Baptiste NARI

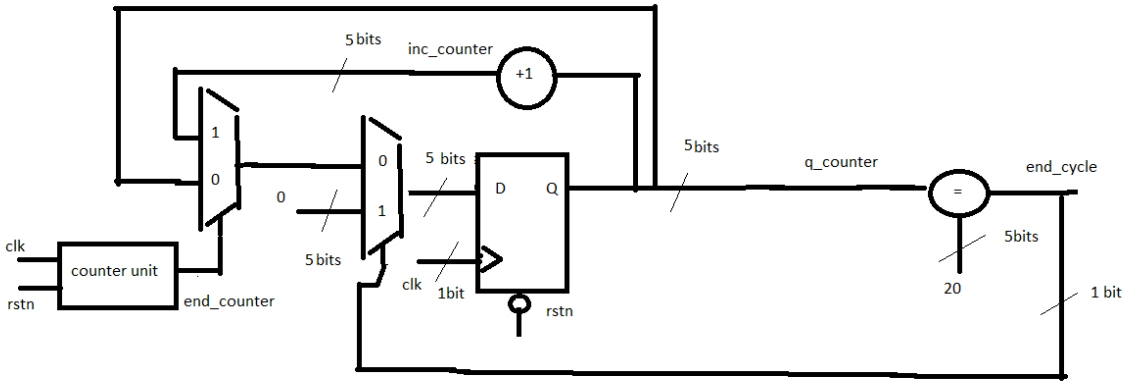
L'objectif de ce TP est de mettre en place une architecture utilisant plusieurs domaines d'horloge. Pour cela, nous utiliserons deux LEDs RGB qui clignoteront avec des fréquences différentes grâce aux horloges. Nous apprendrons également à utiliser une PLL pour générer des horloges.

Question 1

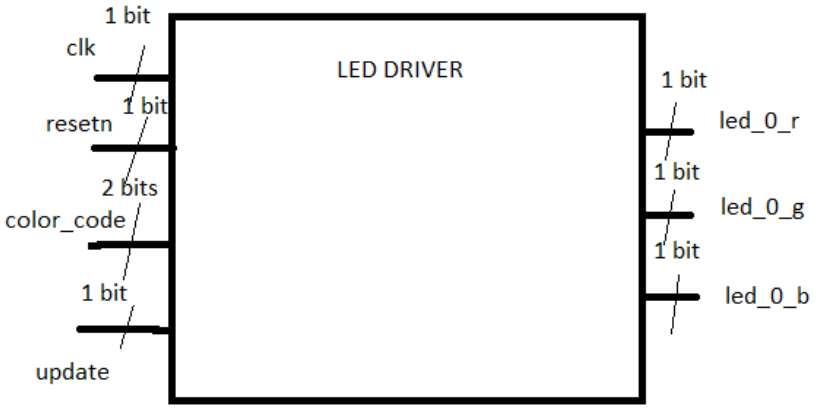
Machine à états :
La machine à états permet la gestion de la couleur à afficher.

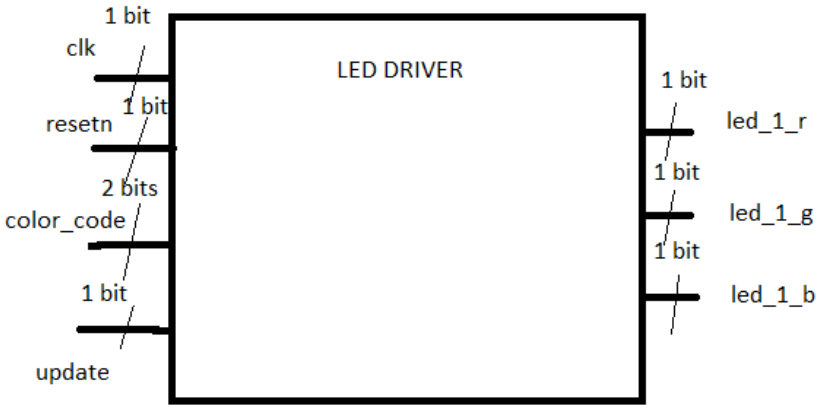


Schema RTL compteur :



Ce compteur permet de compter 10 cycles allumés/éteints.





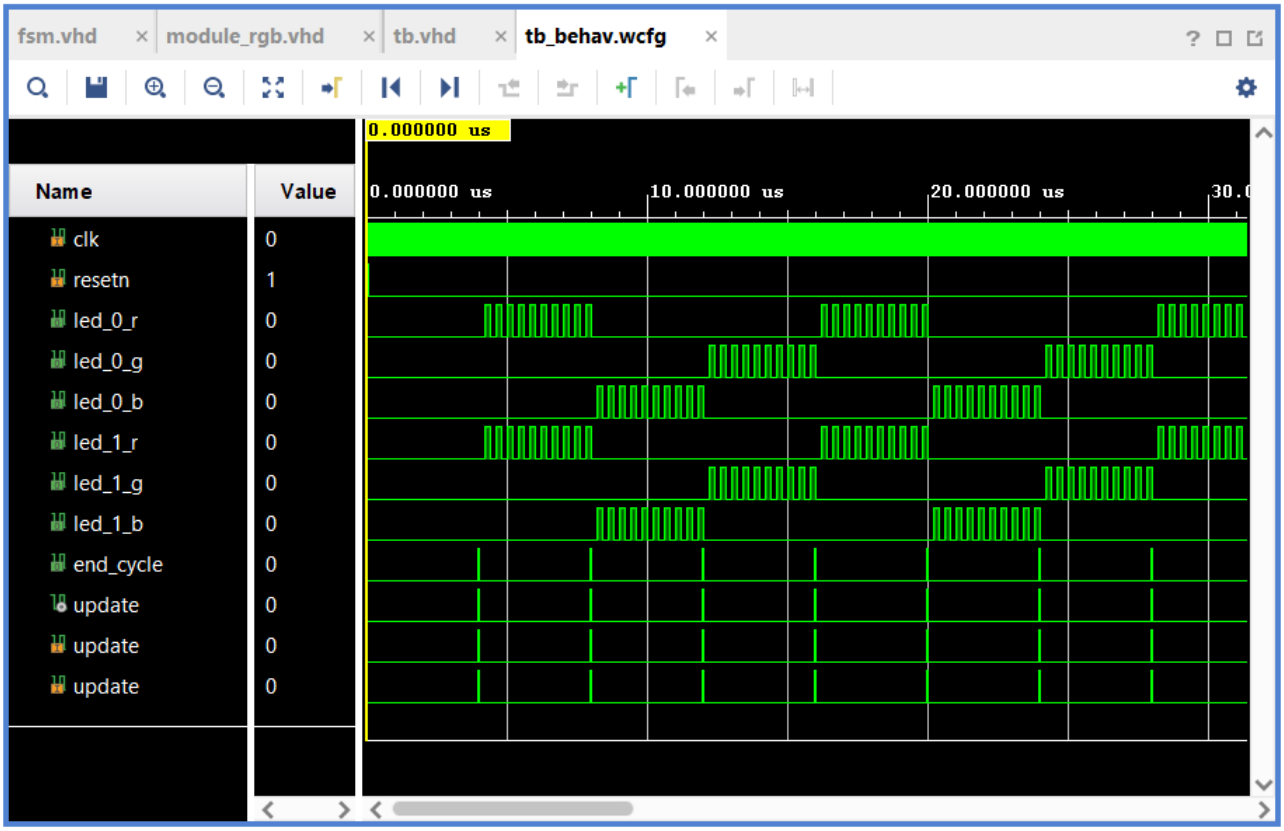
Deux modules led driver pour piloter les deux leds rgb.

Question 2

Rédaction du code correspondant à l'architecture.

Question 3

Résultat de simulation:



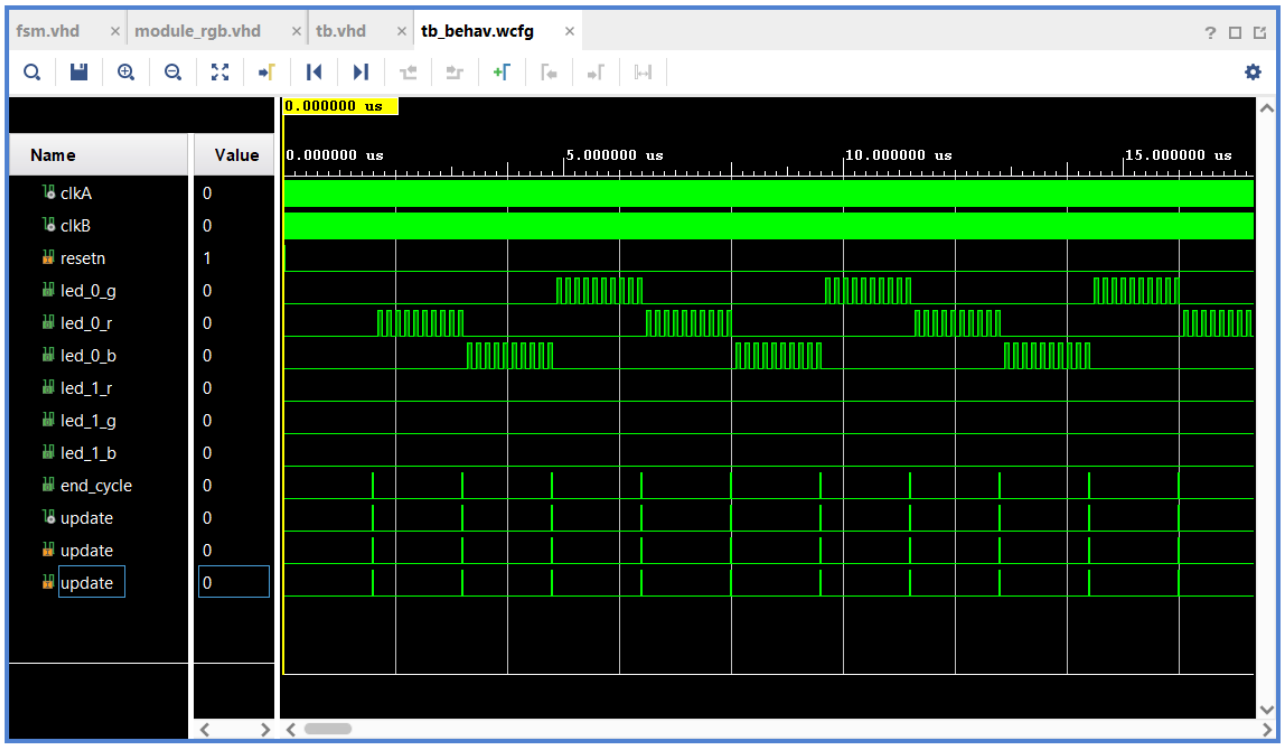
Les deux leds rouges clignotent 10 fois puis les bleus puis les vertes.
Les deux derniers signaux du chronograme représentent le signal update du module rgb0, et du module rgb1, ces modules reçoivent correctement le signal update.

Question 4,5

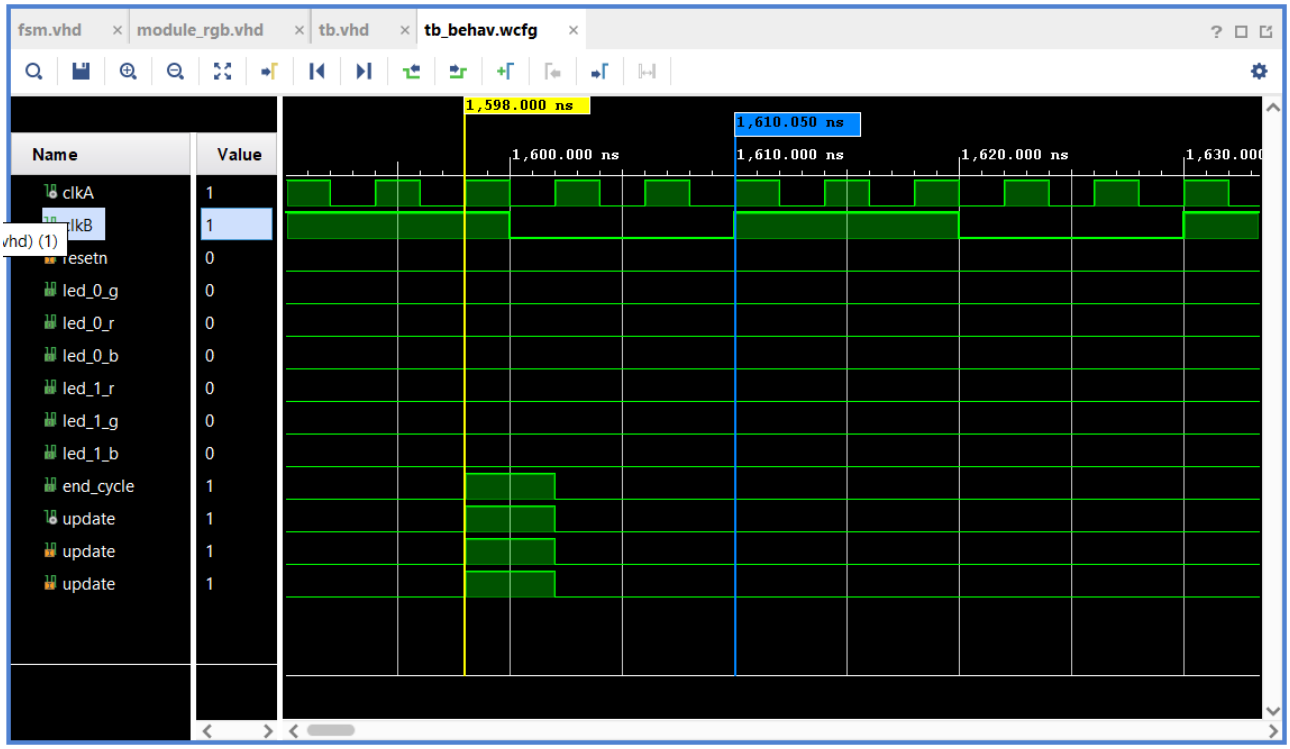
clkA à une frequence de 250MHz.
clkB à une frequence de 50MHz.

Question 6,7,8

Resultat de simulation :



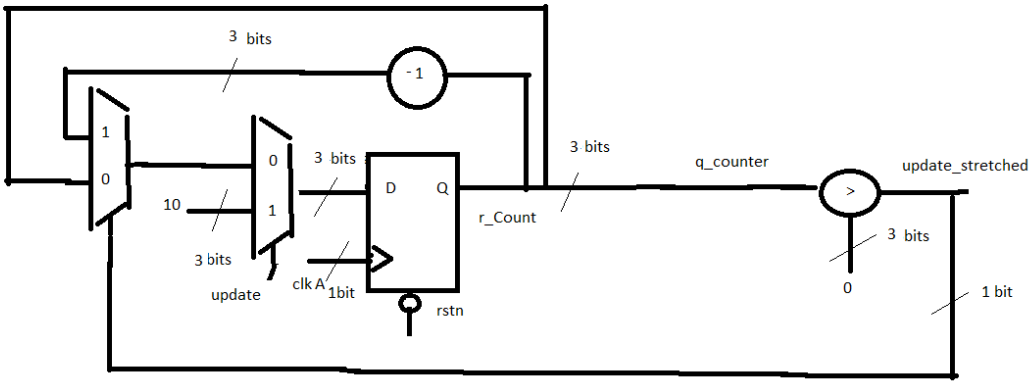
Il n'y a que la led_0 qui clignote. La led_1 ne clignote pas.



C'est parce que le module_rgb_1 ne voit jamais le signal update. Sur le chronograme précédent, curseur jaune, update est à l'état 1 au moment du front montant de clkA. Mais le signal update repasse à zero avant le prochain front montant de clkB (curseur bleu).

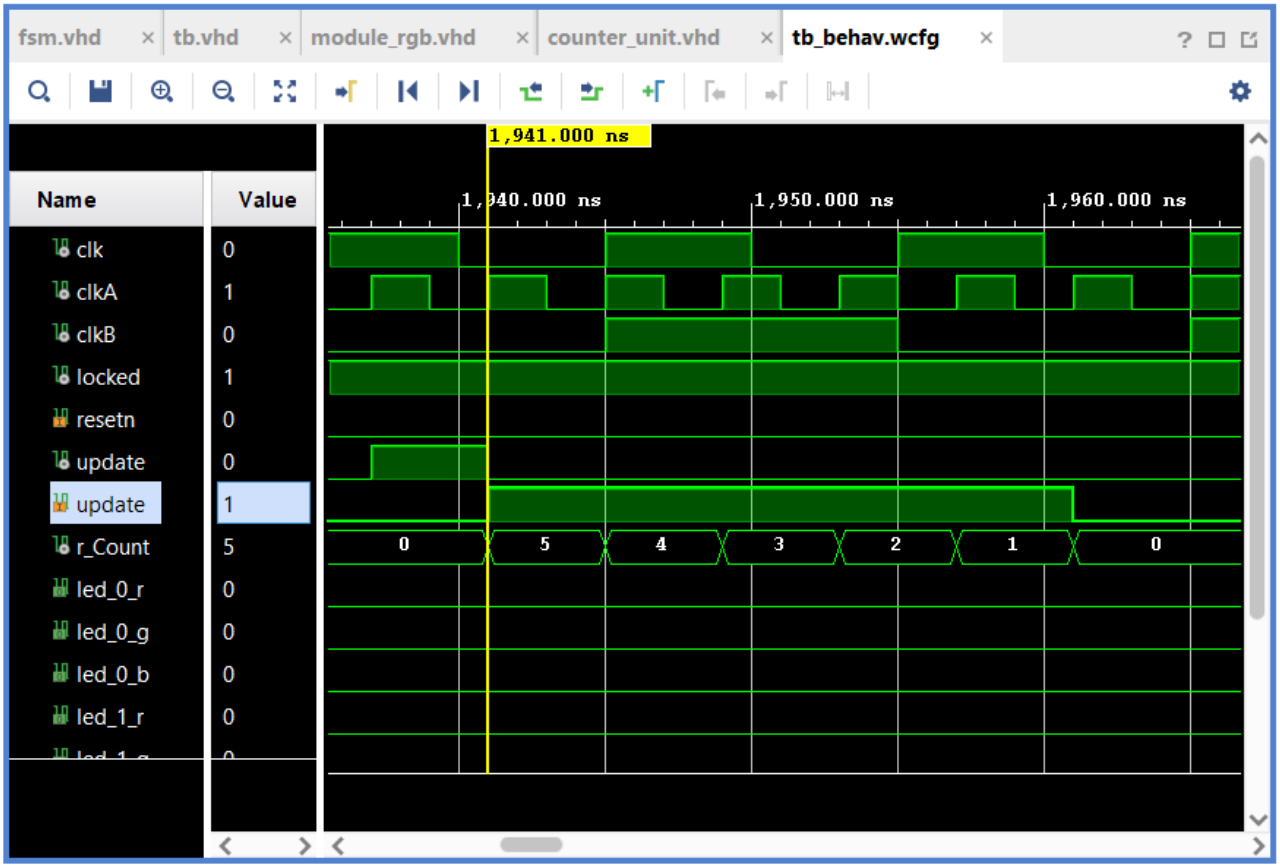
Il y a un cross clock domain, la clkA est de frequence plus élevée que la clkB ce qui fait que le signal update est trop rapide pour la clkB.Une solution serait l'étirement du signal update afin qu'il soit visible au moment du front montant de la clkB.

Schema RTL du circuit supplémentaire :



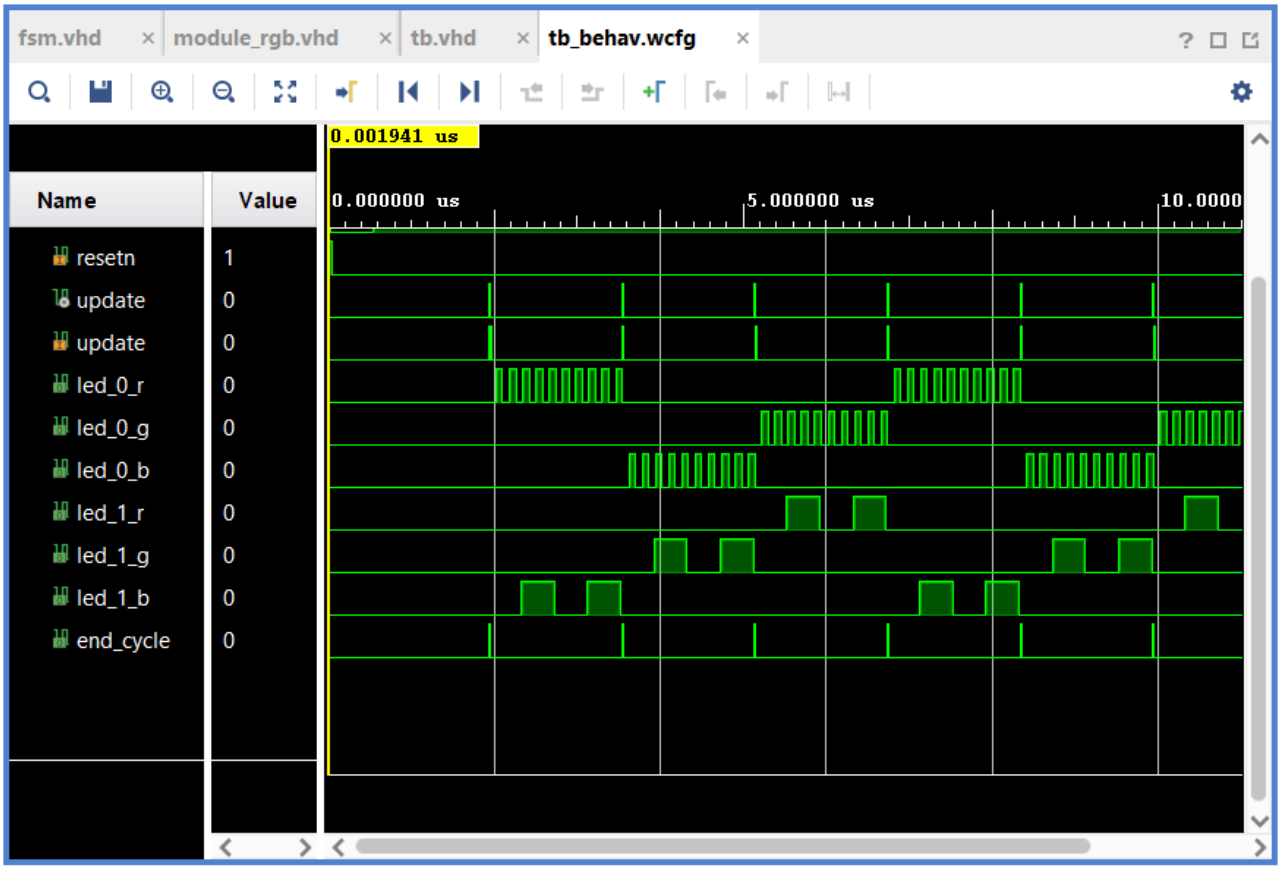
Le circuit réalise un étirement du signal.

Résultat de simulation :

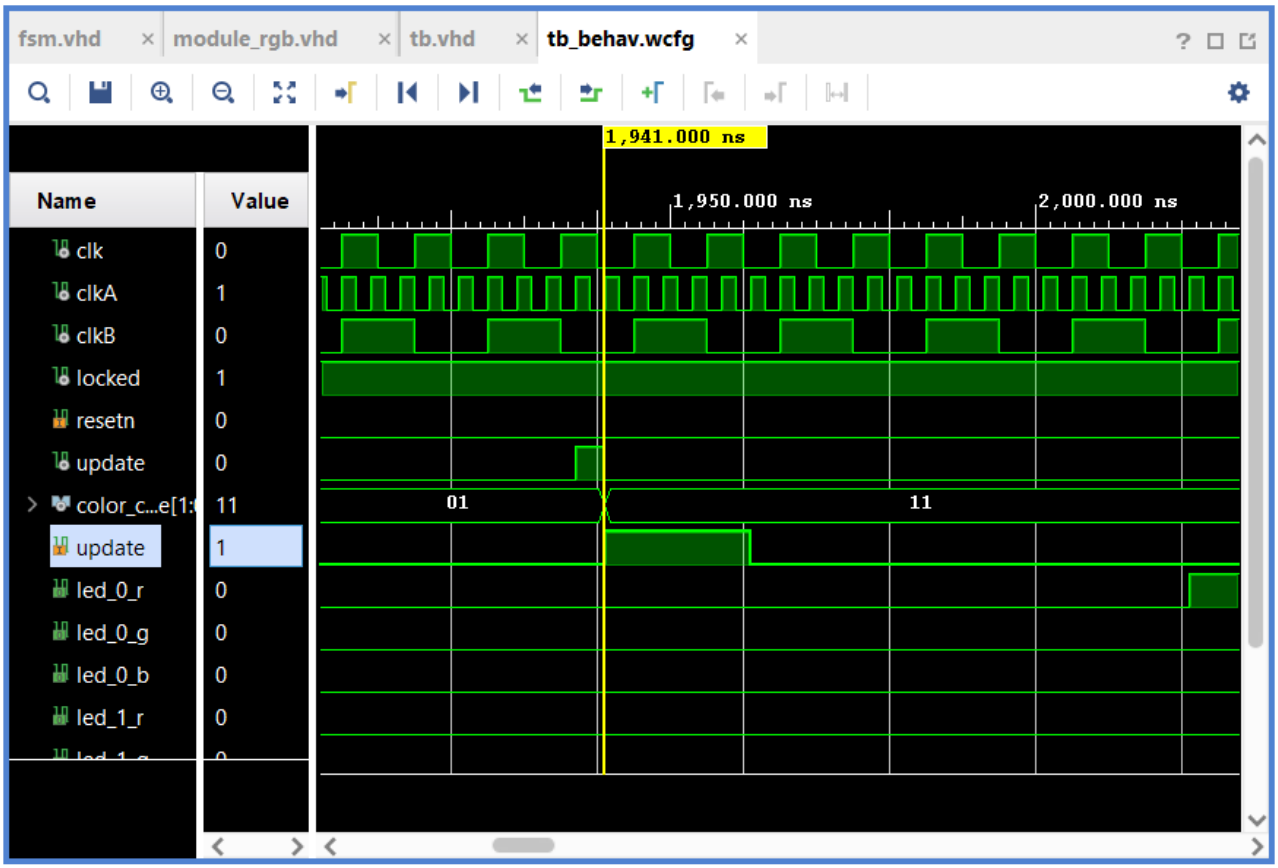


Le signal update_streched correspond bien au signal update étiré sur 5 (250M/50M) périodes d'horloges.

Une fois le signal update étiré les deux leds clignotent.



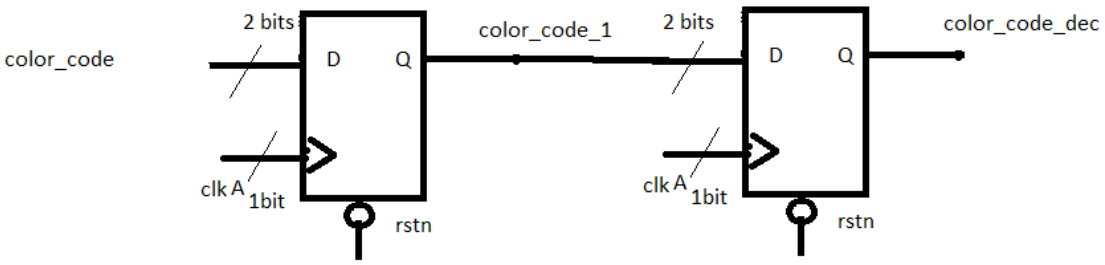
Le cycle sur led_0 est rouge, bleu, vert. Le cycle sur led_1 est bleu, vert, rouge.

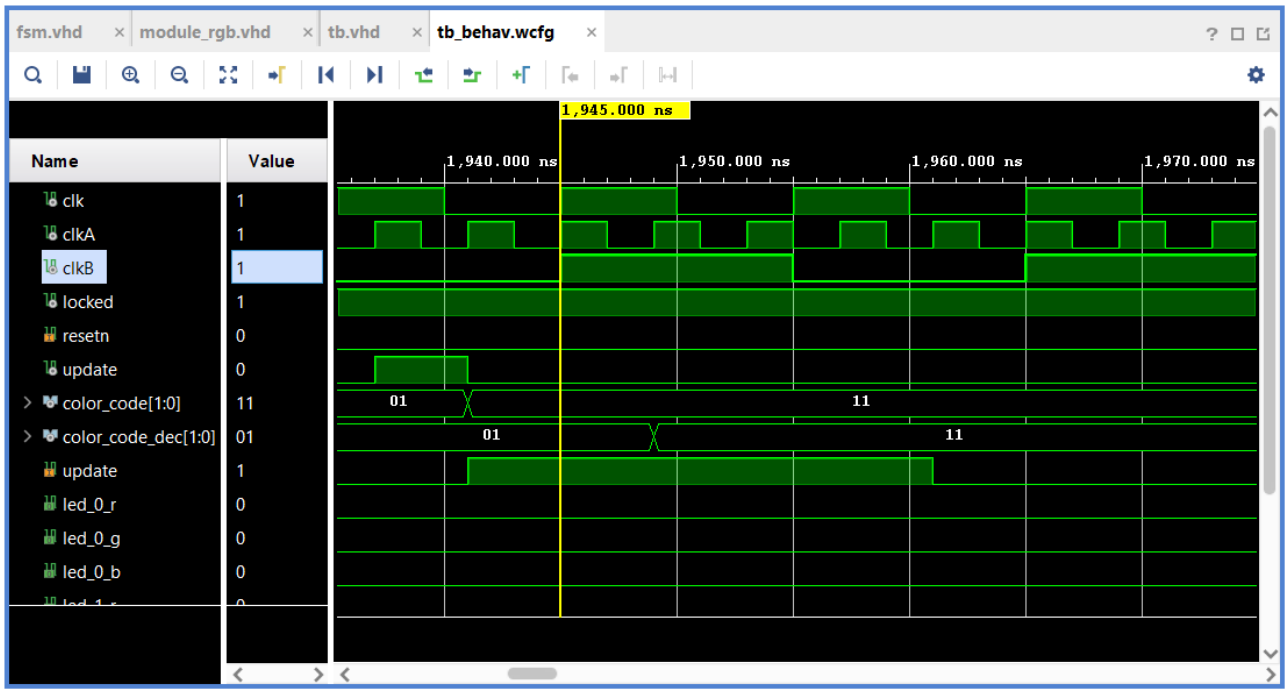


Lorsque le signal update étiré (present sur l'entrée update du module rgb led_1), est à l'état 1, le

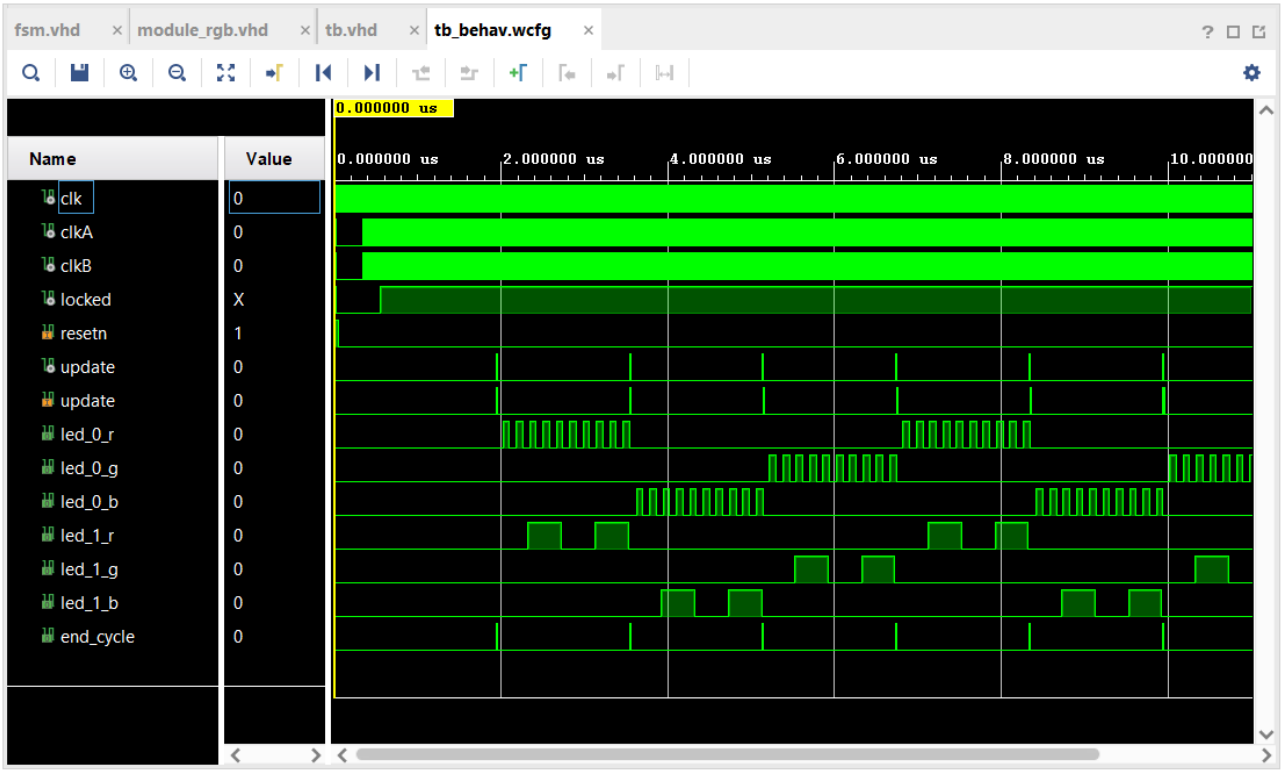
color code est celui de la couleur bleu. Le signal update étiré est décalé dans le temps par rapport au signal update originel. Pour compenser ce decalage il faut réaliser un decalage du signal color code.

Schéma RTL





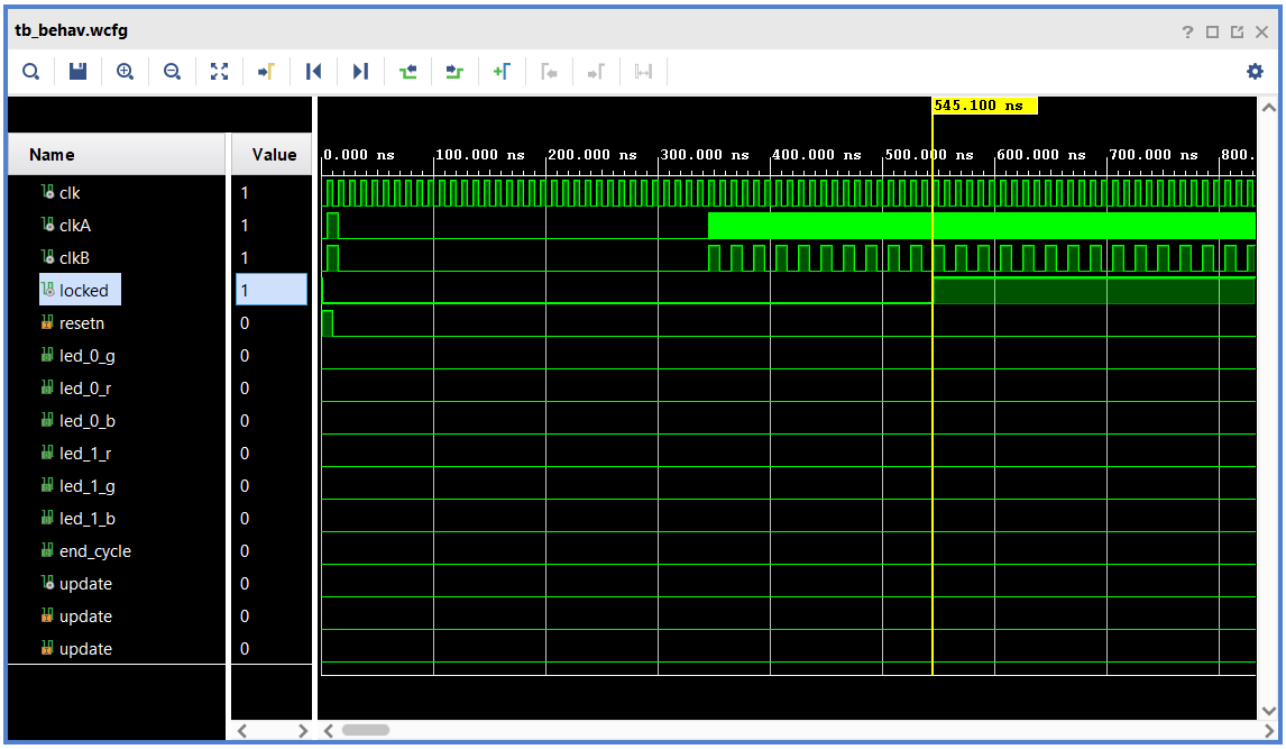
Le signal color_code_dec correspond au signal color code decalé (2 fois avec clkA). Lorsqu'il y a un front montant sur clkB et que le signal update étiré est à l'état 1, le color_code_dec est celui de la couleur rouge ("01").



Les deux leds clignotent suivant le même cycle rouge,bleu,vert.

Question 9

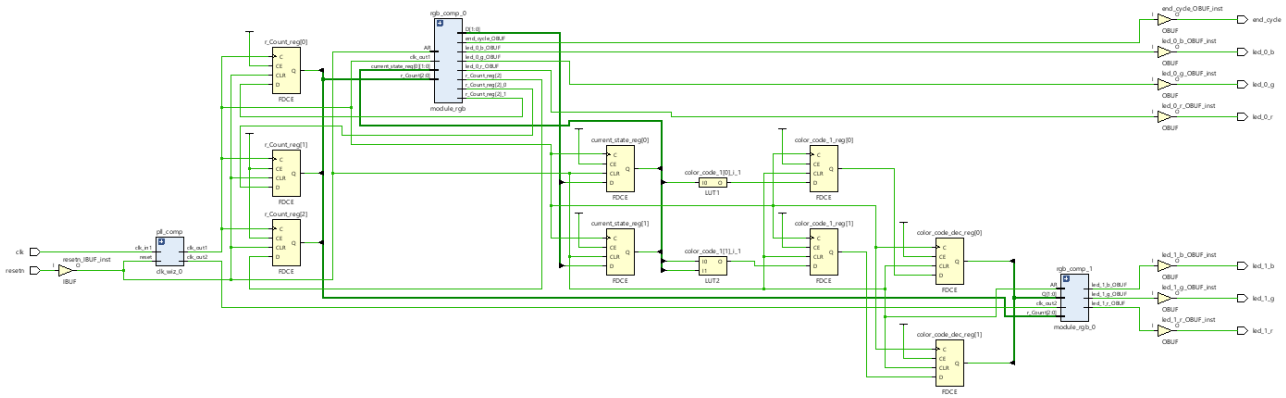
Résultat de simulation PLL:



Le signal clk de l'horloge pricipale est présent dès le début de la simulation.

Les horloges secondaires clkA, clkB s'établissent au bout de 350 ns. Le signal locked émis par la PLL passe à l'etat 1 au bout de 545ns. C'est ce signal qui indique que la PLL est pleinement operationnel. Dans notre circuit ce n'est pas gênant, mais il serait préférable de remettre à zero le circuit tant que locked est à l'etat zero.

Schématic global incluant la PLL

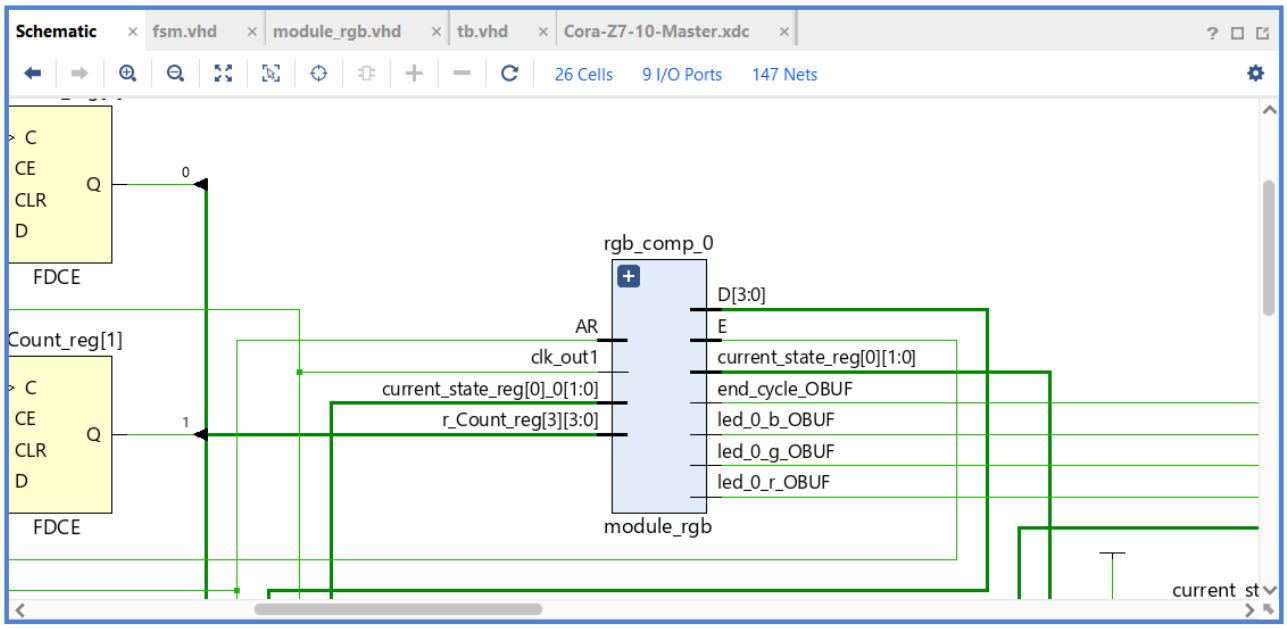


Les deux horloges crées par la PLL sont utilisées dans tout le reste du circuit.

Question 10

Il serait intéressant de regarder les horloges ainsi que le signal update et update_streched.

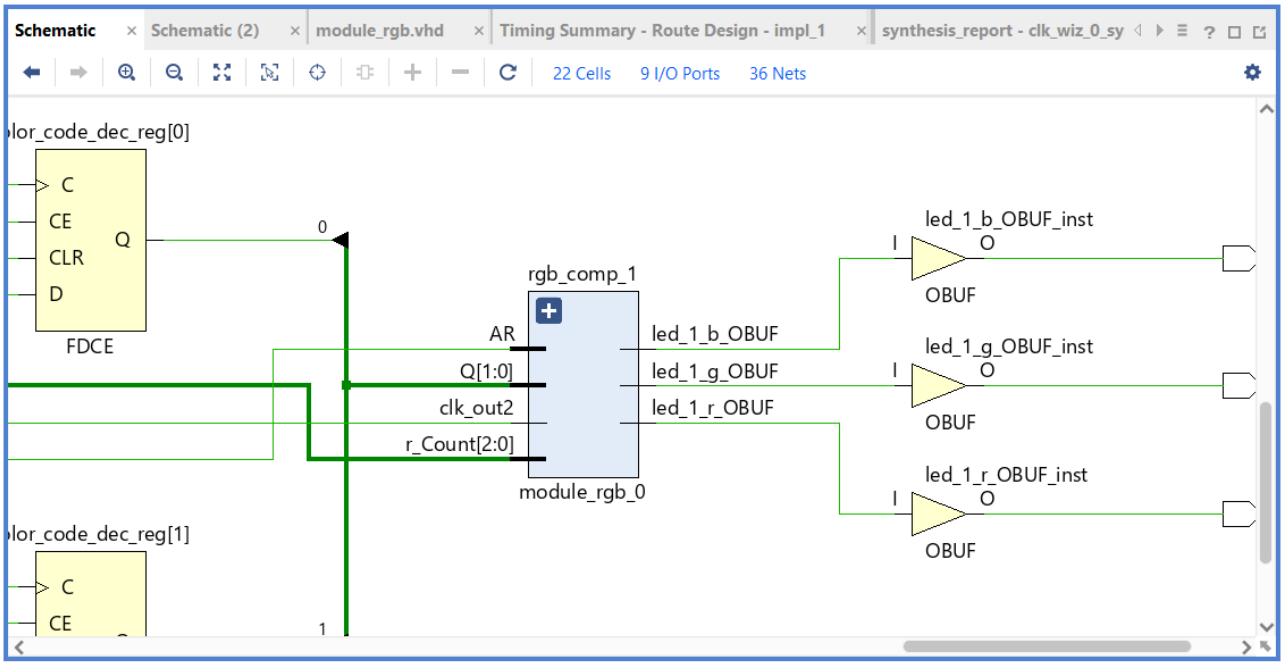
En raison d'optimisation de vivado le signal update et update_streched ne sont pas visibles dans la net list, ni même dans mon schematic.



Dans le schématic le module rgb_comp_0 a des entrées et des sorties qui ne correspondent pas à son entitée.

```
port (
        clk : in std_logic;
        resetn : in std_logic;
        color_code : in std_logic_vector(1 downto 0);
        update : in std_logic;
        led_r : out std_logic;
        led_g : out std_logic;
        led_b : out std_logic;
        end_cycle : out std_logic
    );
end module_rgb;
```

Et plus surprenant.



Le module rgb_comp_1 qui est réalisé avec le même code que le module rgb_comp_0 n'a pas les même entrées et sorties que le module rgb_comp_0. Le logiciel doit réaliser des optimisations en fonction de l'utilisation des composants.

Question 11

Rapport de synthèse :

Report Cell Usage:

	Cell	Count
1	clk_wiz	1
2	CARRY4	14
3	LUT1	2
4	LUT2	62
5	LUT3	6
6	LUT4	12
7	LUT5	9
8	LUT6	7
9	FDCE	76
10	IBUF	1
11	OBUF	7

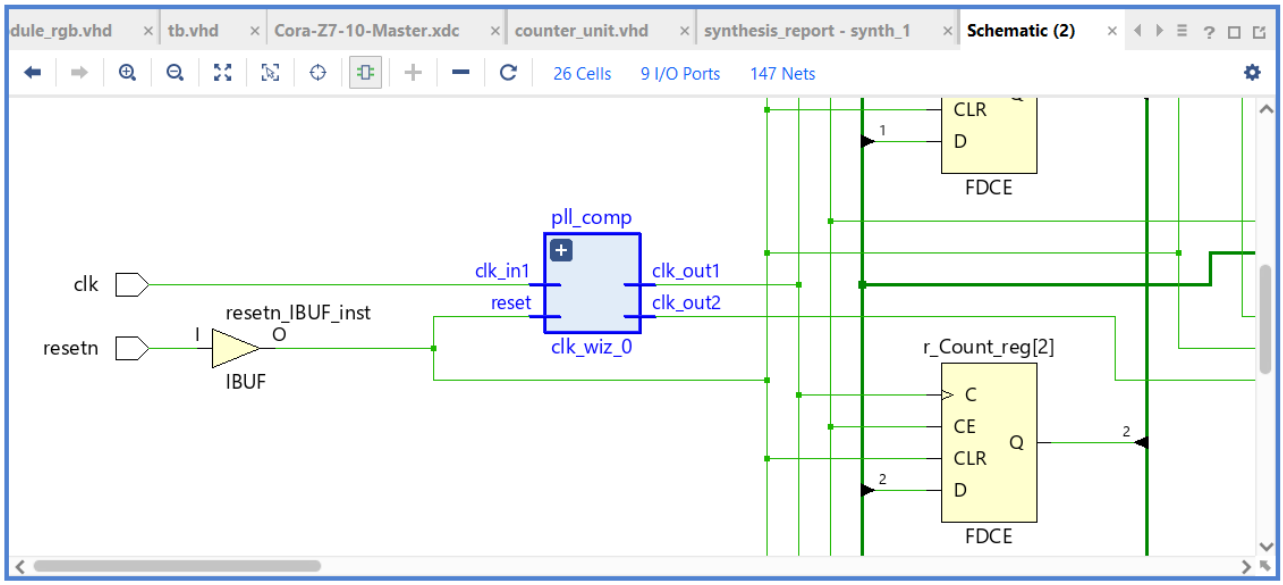
Pour les registres : 28 pour le compteur, 2 pour la gestion du signal update de color_code, 1 pour la machine à états (on, off), 5 registres pour les trois bits de comptage de cycles allumés/éteints (tout ces registres compte double car il y a deux modules)

3 pour le decomptage rCount, 2 decalage de color_code, 2 pour la machine à états (rouge, vert, bleu). Ce qui fait un total de 79 registres.

Il y a 76 FDCE (Registre à reset asynchrone) le logiciel a réalisé des optimisations.

Il y a un écart de 3 registres entre les registres calculés et réellement présents nous contatons que l'ordre de grandeur est respecté.

Il y a 1 IBUF pour l'entrée : resetn relié sur bp_0.



Clk est relié directement à la PLL elle ne passe pas par un imput buffer.

Il y a 7 OBUF pour les sorties : led_0_r , led_0_g ,led_0_b,led_1_r, led_1_g , led_1_b , end_cycle.

Question 12

Lors du placement routage une violation de timing est detectée WNS = -0.418 TNS = -3.493.
Pour corriger ce problème il est nécéssaire de reduire la frequence clkA générée par la PLL de 250MHz à 200MHz.

Rapport de timing :

WNS(ns)	TNS(ns)	TNS Failing Endpoints	TNS Total Endpoints
0.565	0.000	0	81

WHS(ns)	THS(ns)	THS Failing Endpoints	THS Total Endpoints
0.122	0.000	0	81

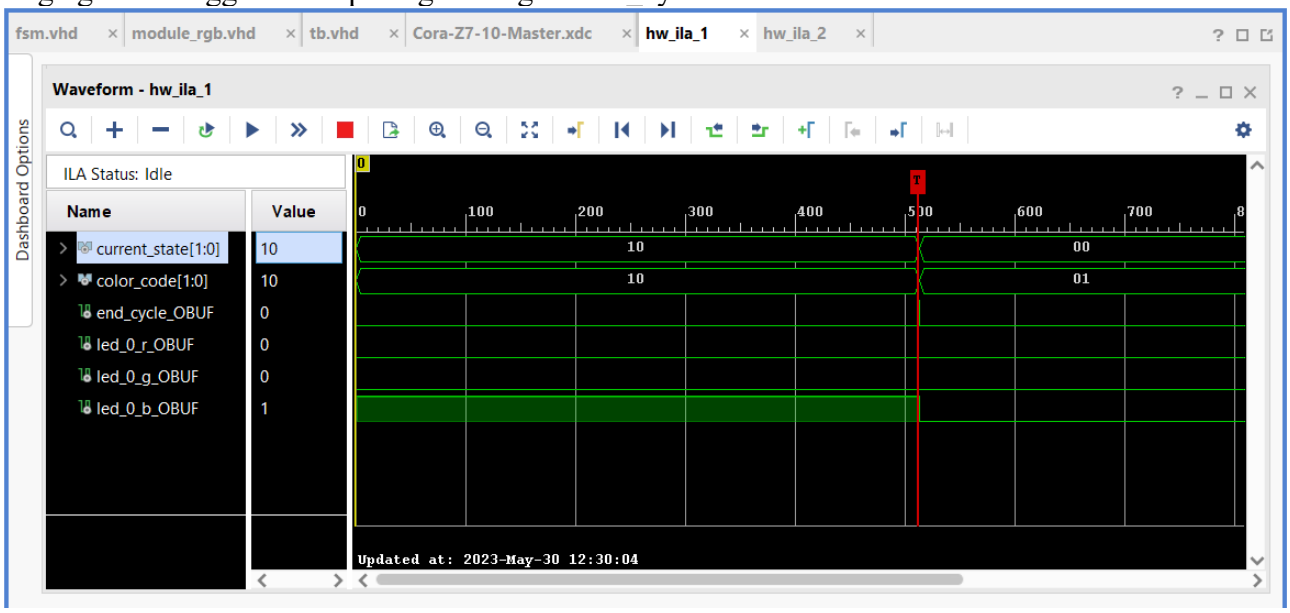
TNS et THS sont à 0 il n'y a pas de violation de setup ou de hold.

Question 13

Génération du bitstream, programmation de la carte.

Les optimisations décrites à la question 10 empêchent de réaliser une étude détaillée du changement de domaine d' horloge.
Observation du signal end_cycle qui est similaire au signal update et verification du changement d'état sur led_0.

Réglage d'un trigger sur le passage du signal end_cycle à l'état 1.



Après passage à l'état 1 du signal end_cycle la led_0_b change d'état.

Decription du fonctionnement sur carte.

Les leds suivent le cycle rouge, bleu, vert, la led 0 clignote 10 fois de chaque couleur, la led 1 clignote 2 fois de chaque couleur.

C'est conforme au fonctionnement attendu.

Dans ce tp nous avons pu mettre en évidence un probleme de cross clock domain et nous avons apporté une solution.