

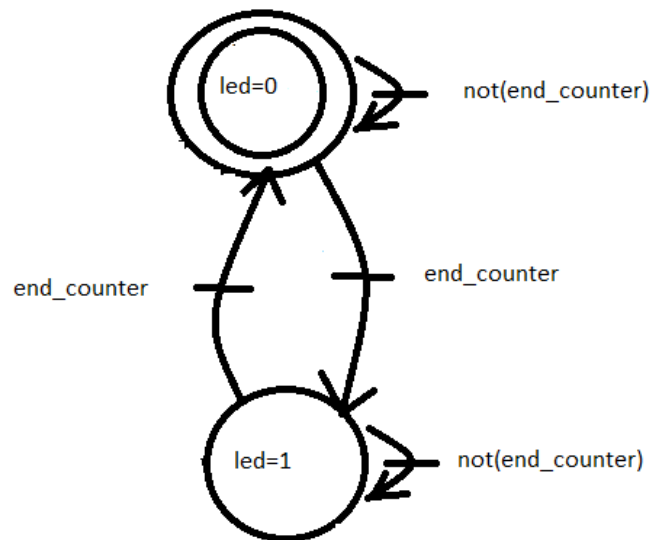
# TP4

## Jean Baptiste NARI

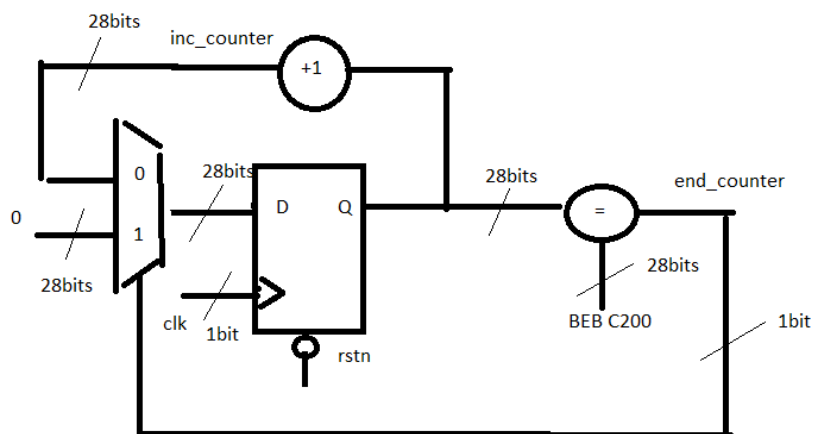
Pilotage de LED et mémoire

### Question 1

Machine à états:

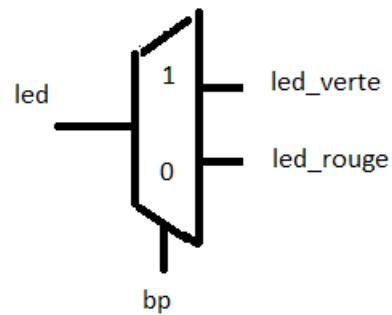


Compteur:



## Question 2

Demultiplexeur :



Un demultiplexeur permet le choix de la led verte ou rouge.

## Question 3

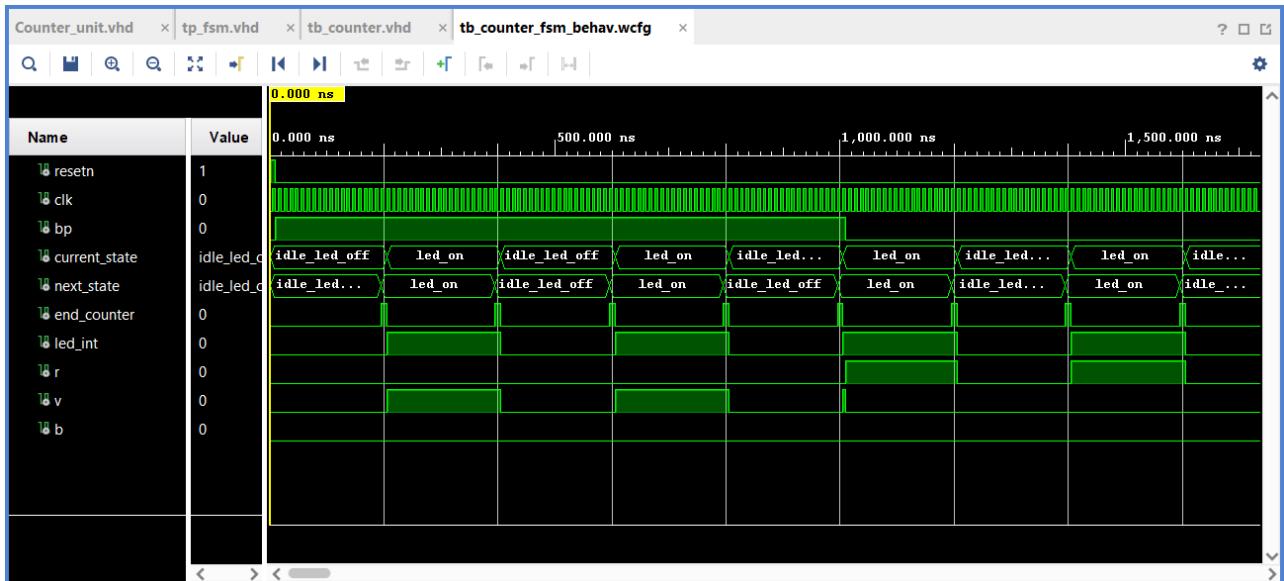
```
process(bp,led_int)
begin
  if(bp='1') then
    v <=led_int;
    r <='0';
  else
    v<='0';
    r <=led_int;
  end if;
end process;
```

Ceci est le code du demultiplexeur.

Les autres parties de code ont été étudiées dans les tp précédents.

## Question 4

Résultat de simulation :



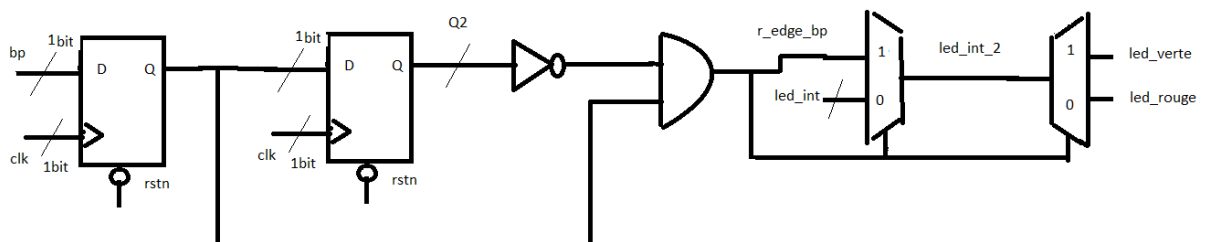
La machine à états oscille entre les états idle\_led\_off et led\_on. Ce qui crée un clignotement sur le signal led\_int. Ce signal led\_int est reporté sur la broche r ou v en fonction de l'appui sur le bouton poussoir bp.

Lorsque le signal bp est pressé pendant plus d'un cycle d'horloge la led verte clignote pendant toute la durée d'appui.

## Question 5

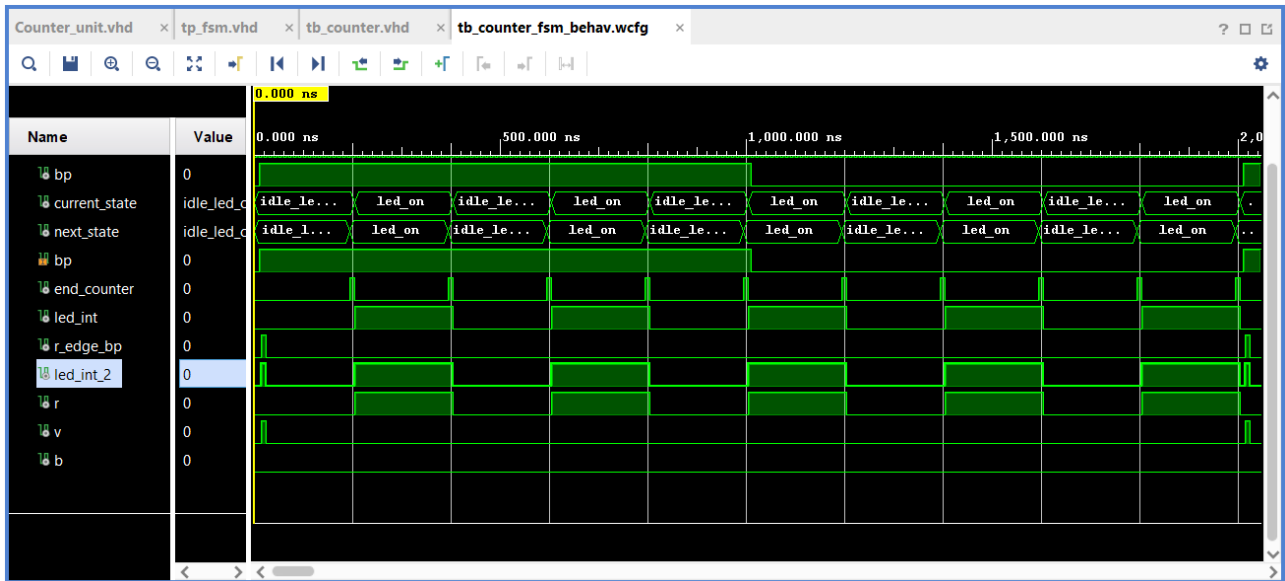
Il est nécessaire de modifier notre schéma afin de detecter les fronts montants du signal bp.

## Question 6

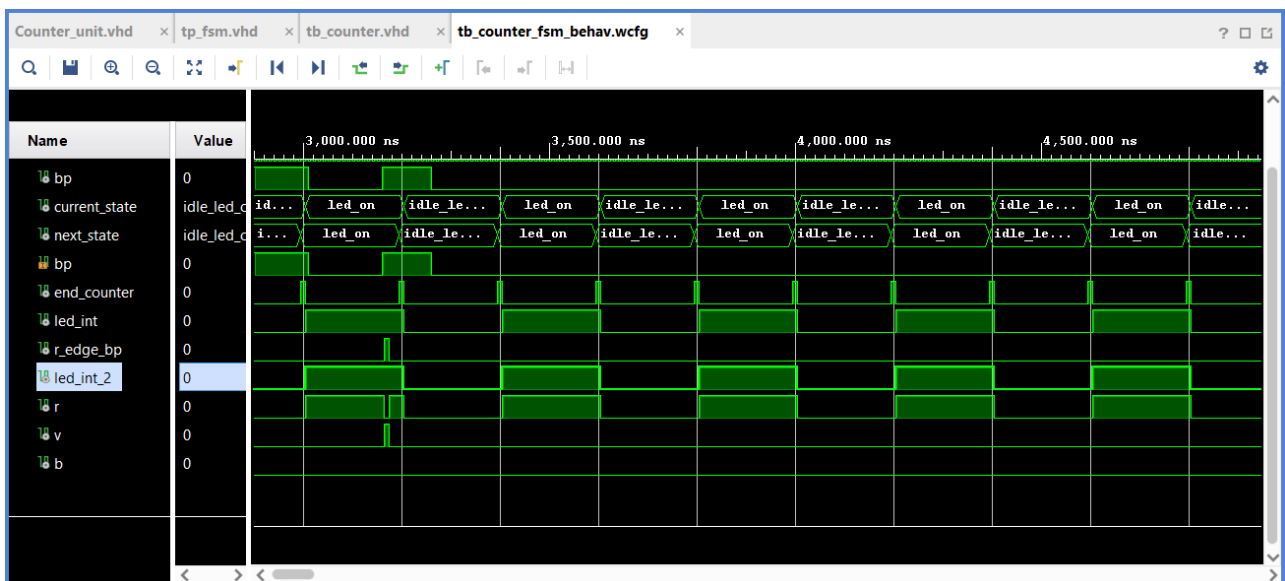


## Question 7

Resultat de simulation :



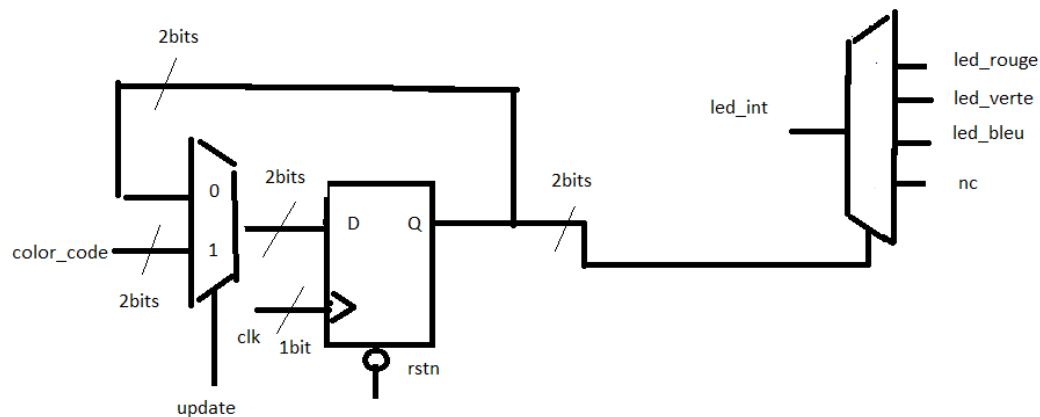
Lors de l'appui sur le bouton poussoir la led verte passe à l'état 1 pendant 1 cycle d'horloge.



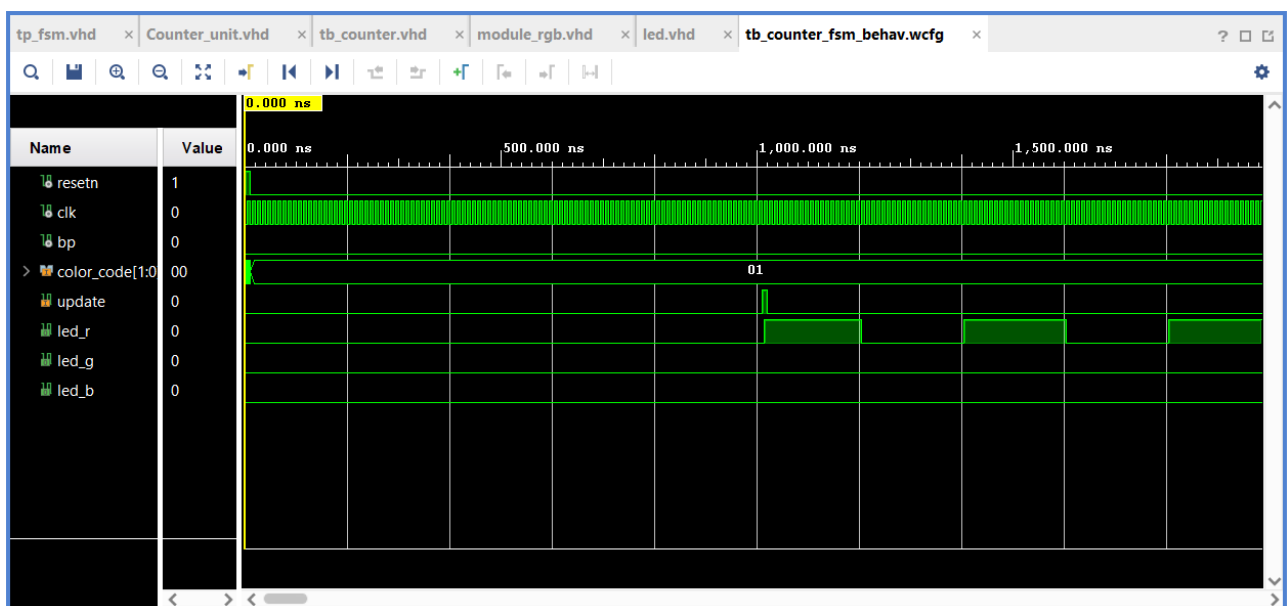
Si l'appui sur le bouton poussoir arrive lorsque la led rouge est allumée, celle-ci est éteinte et la led verte est allumée pendant un cycle d'horloge, ensuite la led rouge reprend son cycle.

## Question 8

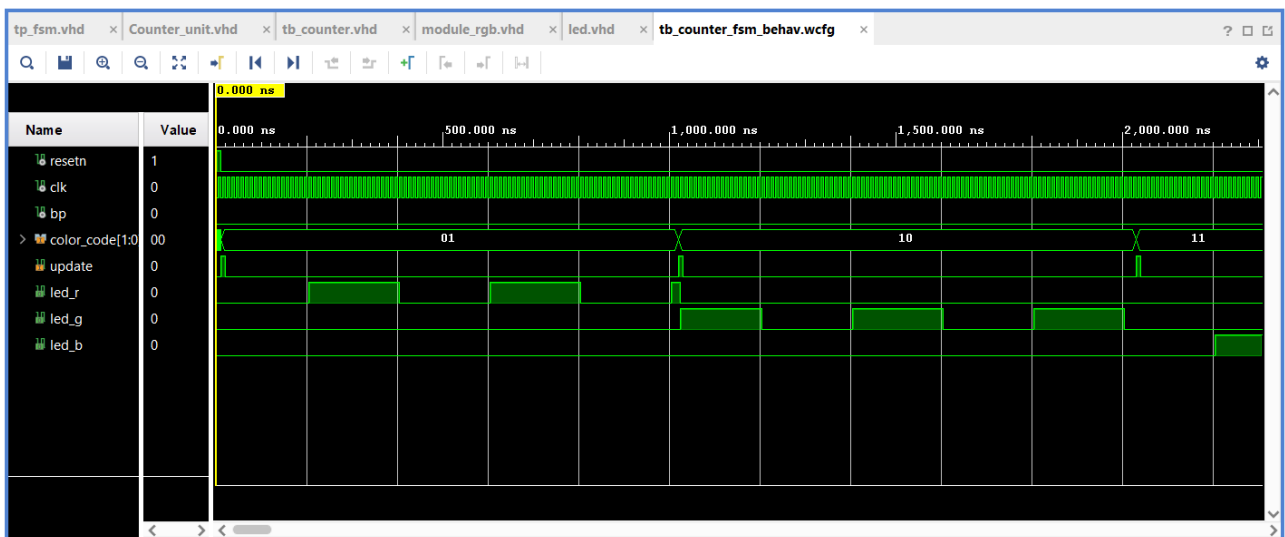
Creation d'un module de pilotage d'une led rgb en schéma RTL.  
Pour cela la partie machine à états et le compteur ne change pas.  
Ajout du circuit suivant :



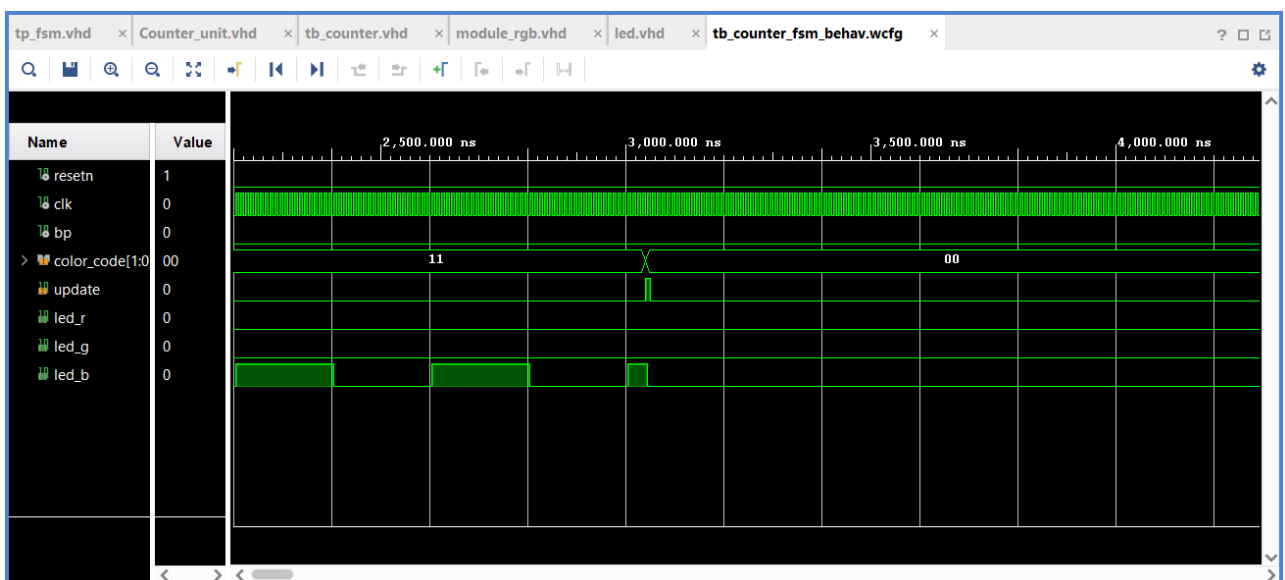
Resultat de simulation :



Une fois que le signal `update` passe à l'état 1 la led de couleur correspondant à `color_code` se met à clignoter. Avant le passage à l'état 1 du signal `update` le circuit maintient son comportement précédent.



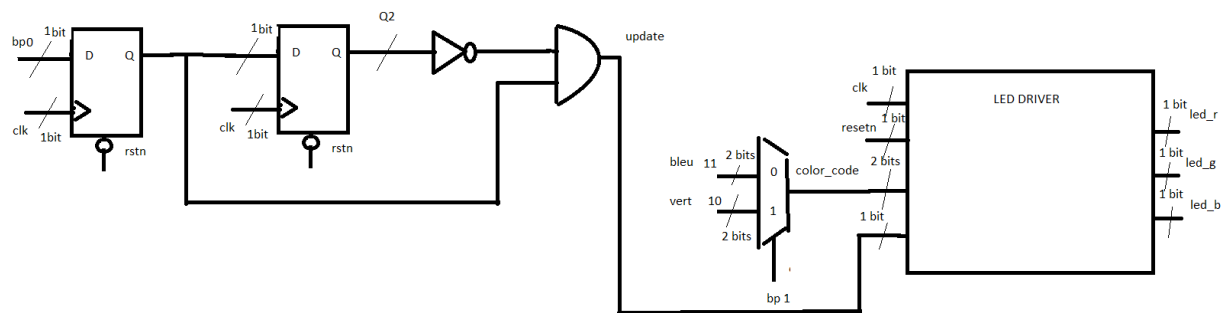
Après passage à l'état 1 du signal update la led correspondante à color\_code clignote bien.  
 Color code 01 led rouge clignotante.  
 Color code 10 led verte clignotante.  
 Color code 11 led bleu clignotante.



Color code 00 aucune led clignote.  
 Le module se comporte comme attendu.

## Question 9

Schéma RTL du circuit de pilotage du module:

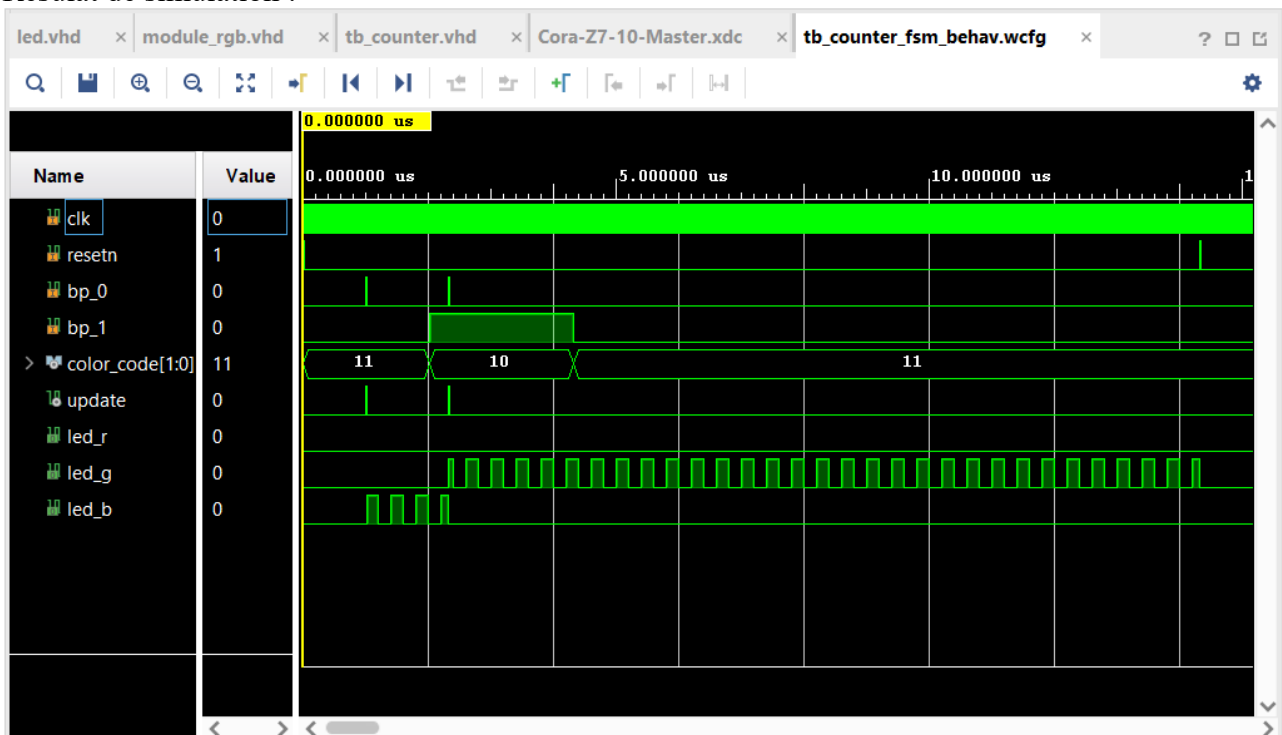


## Question 10,11

Ecriture code vhdl et testbench.

## Question 12

Resultat de simulation :



Lorsque le bouton bp1 est relâché le color code est bleu. Lorsque le bouton bp1 est enfoncé le color code est vert. C'est une fois que bp0 est enfoncé que la led clignote à la couleur du color code correspondante.

## Question 13

Rapport de synthèse :

Report Cell Usage:

	Cell	Count
1	BUFG	1
2	CARRY4	7
3	LUT1	1
4	LUT3	4
5	LUT4	9
6	LUT5	2
7	LUT6	1
8	FDCE	33
9	IBUF	4
10	OBUF	3

Il y a 33 FDCE (Registre à reset asynchrone) 28 pour le compteur, 2 pour le detecteur de fronts montants, 2 pour la gestion du signal update, 1 pour la machine à état.

Il y a 4 IBUF pour les entrées : clk, resetn, bp\_0, bp\_1.

Resetn est relié sur une digital IO car il n'y a que deux boutons poussoir présents sur la carte.

Il y a 3 OBUF pour les sorties : led\_r, led\_g, led\_b.

## Question 14

Rapport de timing :

WNS(ns)	TNS(ns)	TNS Failing Endpoints	TNS Total Endpoints
5.531	0.000	0	32
WHS(ns)	THS(ns)	THS Failing Endpoints	THS Total Endpoints
0.191	0.000	0	32

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



## Question 15

Génération du bitstream, programmation de la carte.

Après un premier appui sur bp0 la led clignote en bleu. Si l'on appui sur bp1 puis bp0 la led clignote en vert. Si l'on réappui de nouveau sur sur bp0 la led clignote en bleu. C'est conforme au fonctionnement attendu.