

# TP3 compteurs

## Jean Baptiste NARI

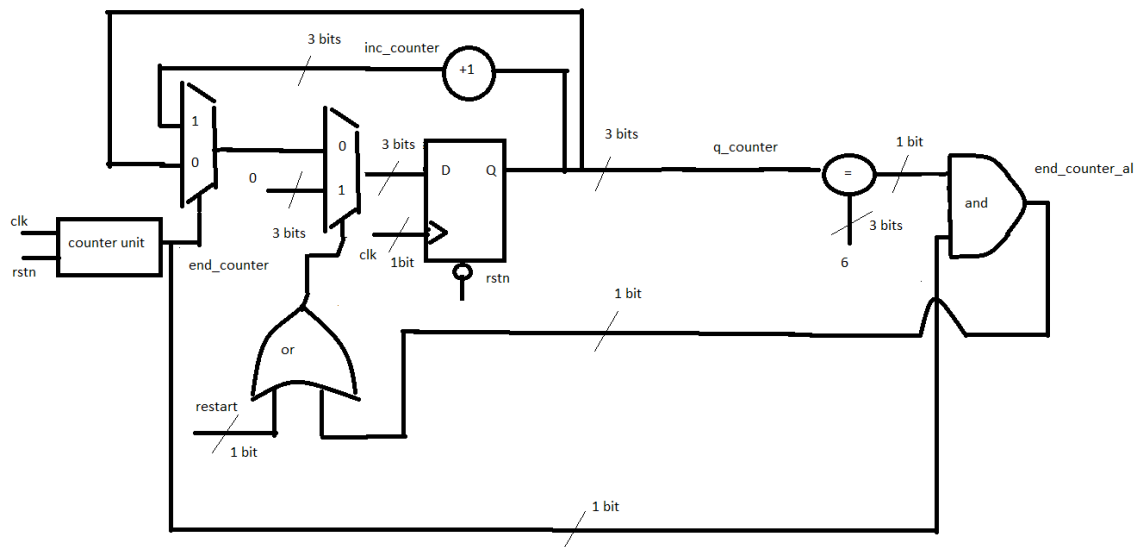
L'objectif de ce TP est de réaliser une architecture permettant de faire clignoter deux LEDs RGB en rouge, vert et bleu. Le pilotage des LEDs se fera à l'aide de machines à états.

### Question 1

Rédaction du code correspondant au module Counter\_unit à partir du compteur du TP1.

### Question 2

Schéma RTL permettant de compter le nombre de cycles allumés/eteints.



Choix de comptage du nombre de changements d'état, il faudra comparer la valeur à 6 pour detecter 3 cycles allumés/eteints.

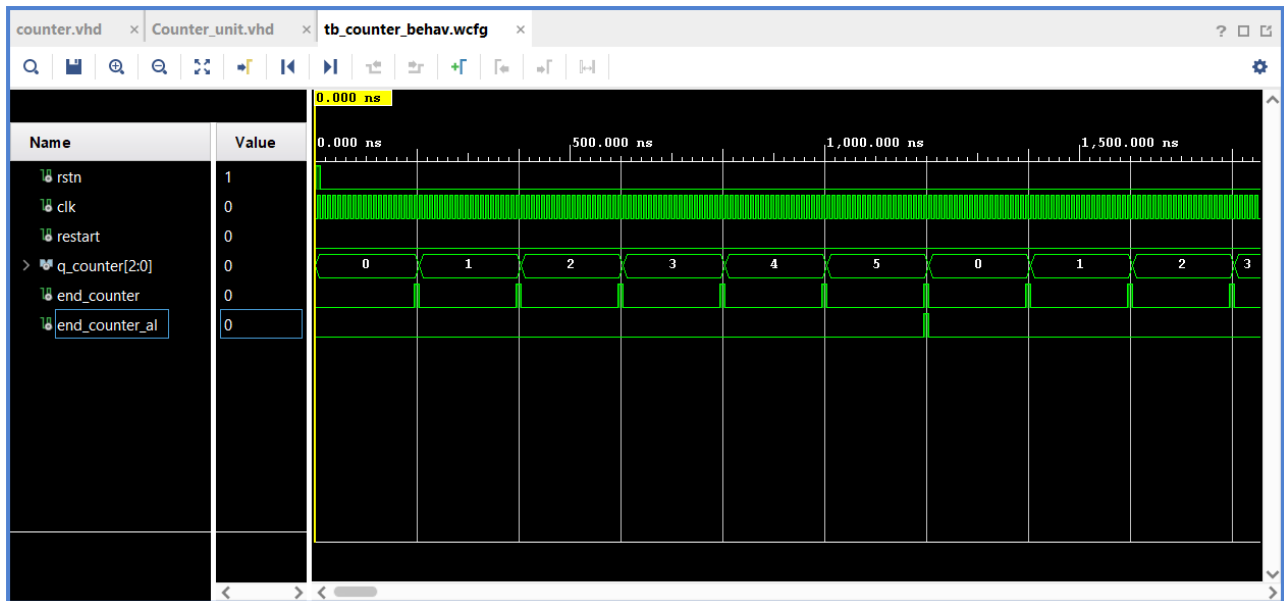
### Question 3

Rédaction du code correspondant au compteur de cycles utilisant le module Counter\_unit.

### Question 4

Rédaction du testbench.

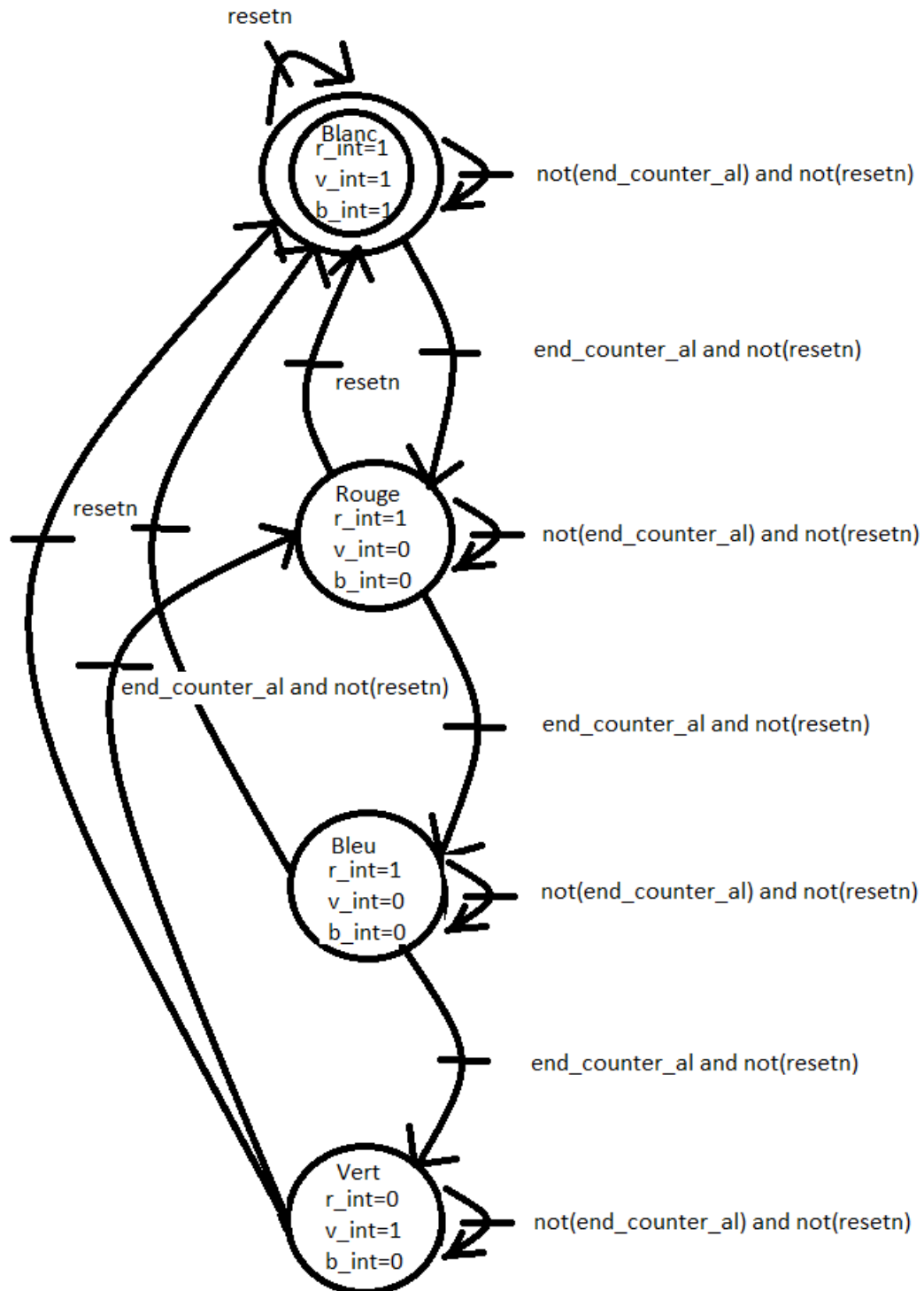
Résultat de simulation du compteur réalisé avec un compteur unit.



Le compteur compte bien jusqu'a 6 (de 0 à 5) puis le signal end\_counter\_al passe à l'état 1 et le comptage repard de 0 pour un nouveau cycle.

## Question 5

Représentation de la machine à états



## Question 6

Signaux d'entrées : rstn , restart, clk

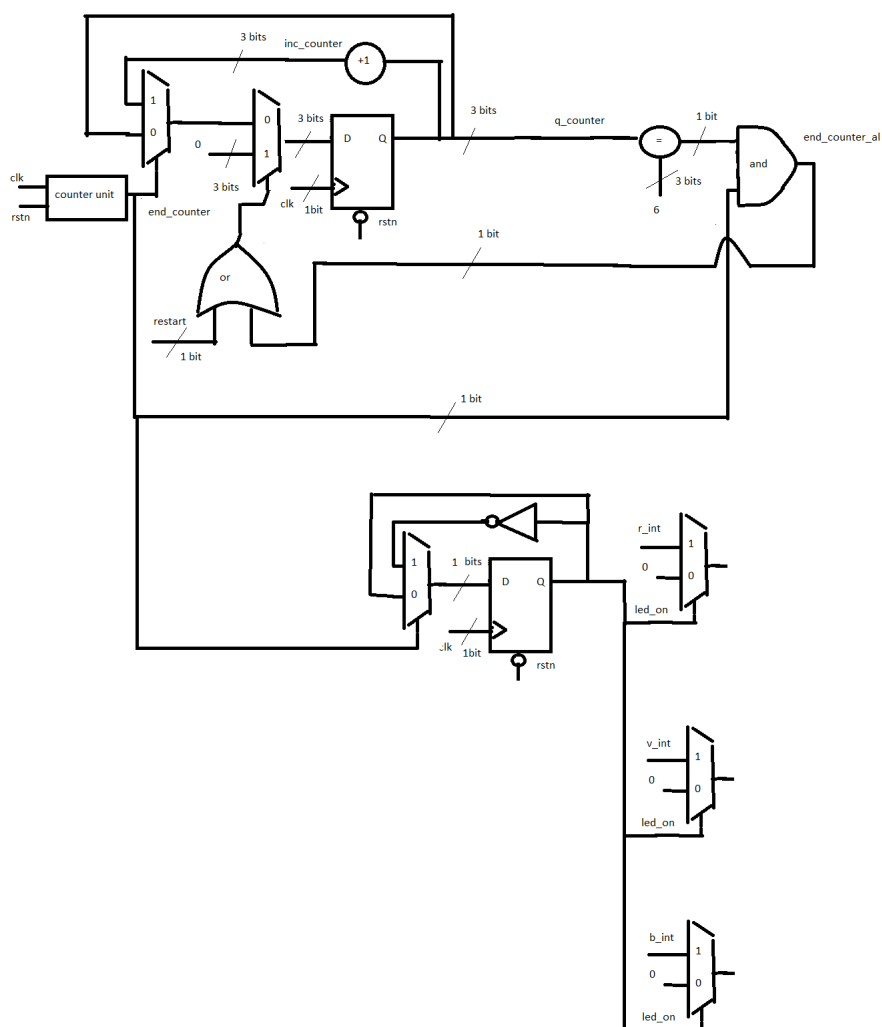
Signaux internes : end\_counter\_al, end\_counter\_2 , r\_int, v\_int, b\_int, Q, led\_on

Signaux sorties : R , V , B

## Question 7

Modification de notre code VHDL.

Schéma RTL



Détail sur la structure des composants.

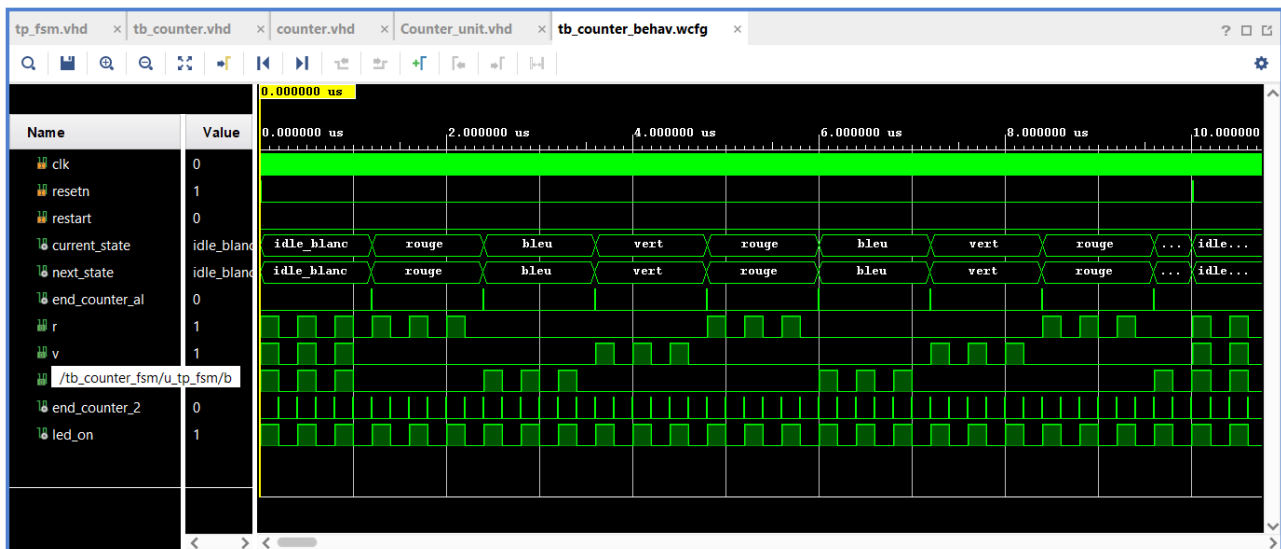
La sortie end\_counter du module counter\_unit passe à travers mon module counter\_al (qui utilise le module counter\_unit) et est présente sur sa sortie (nomé end\_counter\_2).

Le module counter\_al est utilisé par le composant tp\_fsm. C'est dans ce dernier qu'il est nécessaire d'avoir le signal end\_counter\_2, afin de créer le signal led\_on. Le signal led\_on indique si la led est allumée (état haut) ou éteinte (état bas) cette information permet de faire clignoter la led.

## Question 8

Rédaction du testbench.

Resultat de simulation:



La machine à états passe de l'état idle\_blanc à rouge puis bleu puis vert.

Les sorties rvb correspondent bien aux états:

idle\_blanc r=1 v=1 b=1

rouge r=1 v,b=0

vert v=1 r,b=0

bleu b=1 r,v=0

## Question 9

Rapport de synthèse :

Detailed RTL Component Info :

+---Adders :

2 Input 3 Bit Adders := 1

+---Registers :

3 Bit Registers := 1

2 Bit Registers := 1

1 Bit Registers := 1

+---Muxes :

2 Input 3 Bit Muxes := 1

2 Input 2 Bit Muxes := 1

6 Input 2 Bit Muxes := 1

2 Input 1 Bit Muxes := 1

4 Input 1 Bit Muxes := 3

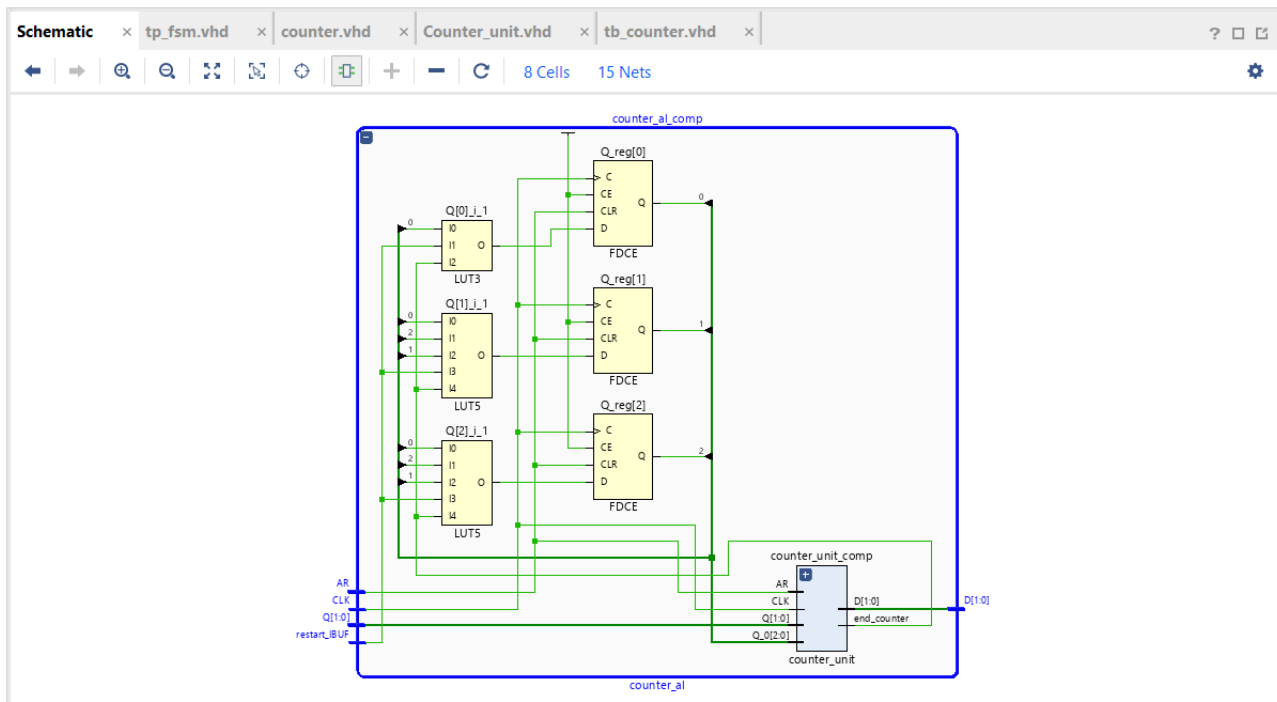
Dans cette sous partie du rapport un registre de 28 bits n'apparait pas.

Mais l'architecture est bien respectée car le registre de 28 bits apparait dans le tableau récapitulatif suivant :

	Cell	Count
1	BUFG	1
2	CARRY4	7
3	LUT2	32
4	LUT3	2
5	LUT4	4
6	LUT5	2
7	LUT6	5
8	FDCE	33
9	FDPE	1
10	IBUF	3
11	OBUF	3

Il y a 1 FDPE (registre à set asynchrone) pour la creation du signal led\_on. Le signal led\_on indique quand la led est allumée ou éteinte.

Le compteur de cycle est le bloc en bleu noté counter\_al\_comp.  
Il apparait en un seul block car j'ai fait le choix de créer un module pour le compteur de cycles.



A l'intérieur de notre compteur de cycles on retrouve notre module counter\_unit\_comp.

## Question 10

Modification du fichier de contraintes:

```
set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { clk }];
#IO_L13P_T2_MRCC_35 Sch=sysclk
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports { clk }];#set
```

# RGB LEDs

```
set_property -dict { PACKAGE_PIN L15 IOSTANDARD LVCMOS33 } [get_ports { b }];
#IO_L22N_T3_AD7N_35 Sch=led0_b
set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports { v }];
#IO_L16P_T2_35 Sch=led0_g
set_property -dict { PACKAGE_PIN N15 IOSTANDARD LVCMOS33 } [get_ports { r }];
#IO_L21P_T3_DQS_AD14P_35 Sch=led0_r
```

# Buttons

```
set_property -dict { PACKAGE_PIN D20 IOSTANDARD LVCMOS33 } [get_ports { resetn }];
#IO_L4N_T0_35 Sch=btn[0]
set_property -dict { PACKAGE_PIN D19 IOSTANDARD LVCMOS33 } [get_ports { restart }];
#IO_L4P_T0_35 Sch=btn[1]
```

## Question 11

Rapport de timing :

WNS (ns)	TNS (ns)	TNS Failing Endpoints	TNS Total Endpoints
4.953	0.000	0	34

WHS (ns)	THS (ns)	THS Failing Endpoints	THS Total Endpoints
0.248	0.000	0	34

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

Chemin critique :

Max Delay Paths

```
-----  
Slack (MET) :      4.953ns (required time - arrival time)  
Source:      counter_al_comp/counter_unit_comp/Q_reg[3]/C  
              (rising edge-triggered cell FDCE clocked by sys_clk_pin {rise@0.000ns  
fall@5.000ns period=10.000ns})  
Destination: counter_al_comp/counter_unit_comp/Q_reg[25]/D  
              (rising edge-triggered cell FDCE clocked by sys_clk_pin {rise@0.000ns  
fall@5.000ns period=10.000ns})
```

## Question 12

Génération du bitstream et programmation de la carte.

La led clignote blanche au premier démarrage de la carte ou si l'on appui sur le bouton reset.

La led prend les couleurs rouge clignotant, puis bleu puis vert et recommence un cycle avec la couleur rouge.

Le fonctionnement sur carte est conforme à la description.