

TP03 – Compteurs

Cédrine Socquet

Rendu

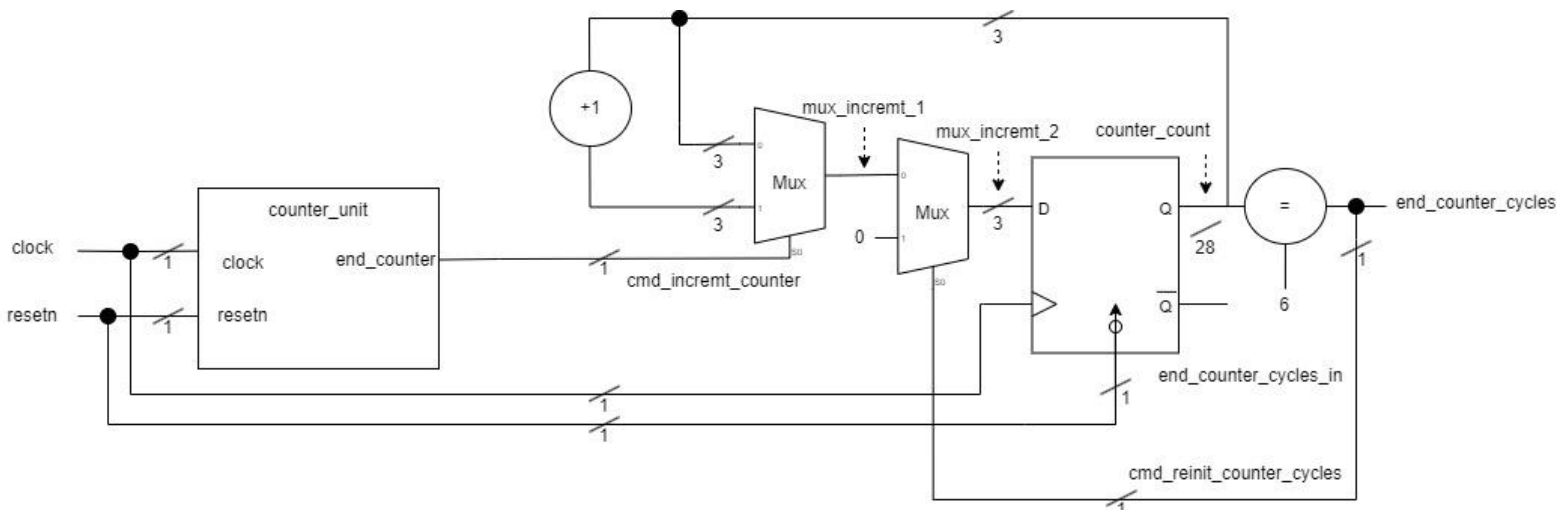
Votre rapport devra contenir :

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

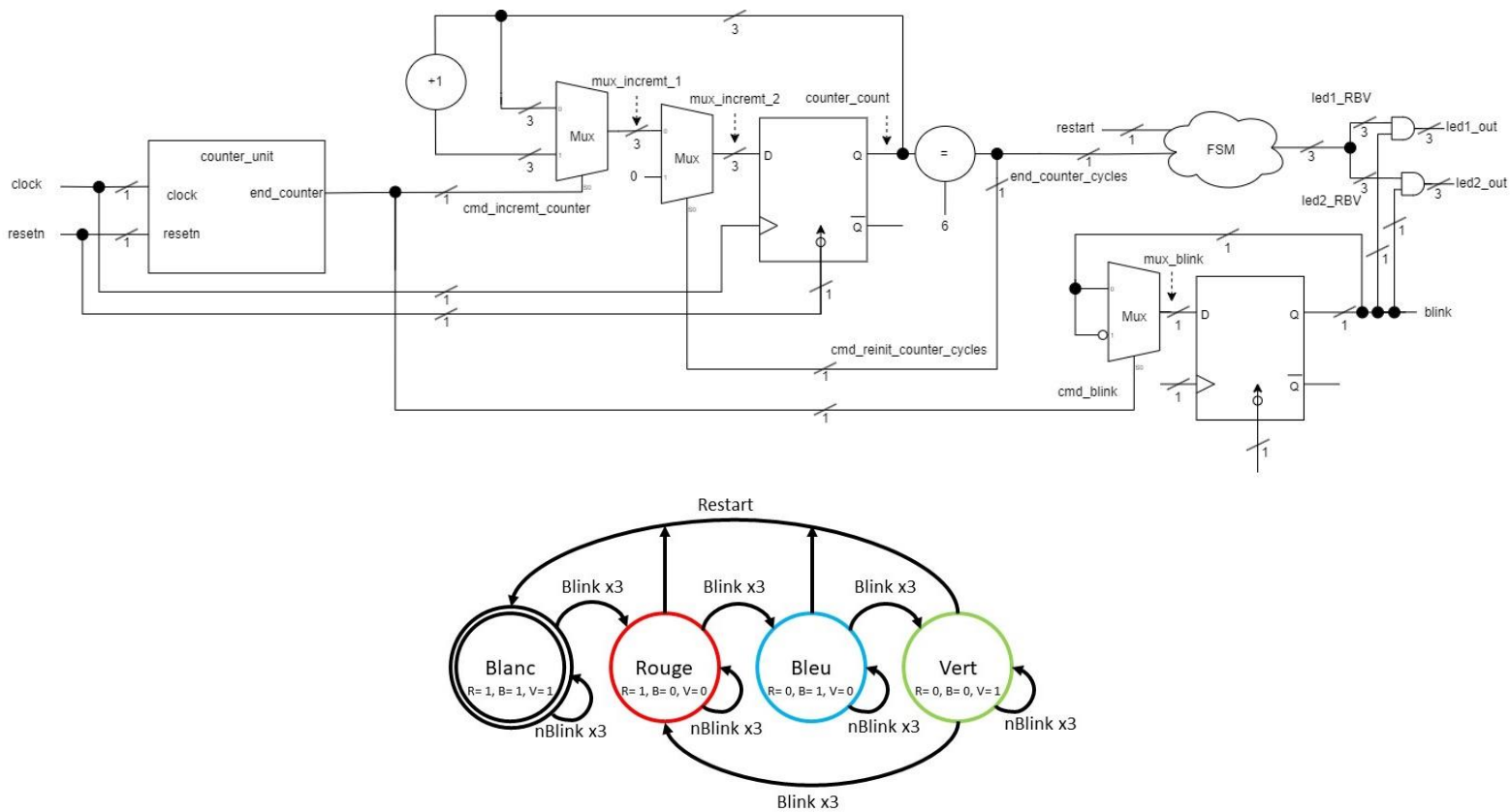
Vous fournirez également vos codes source commentés.

Réponses

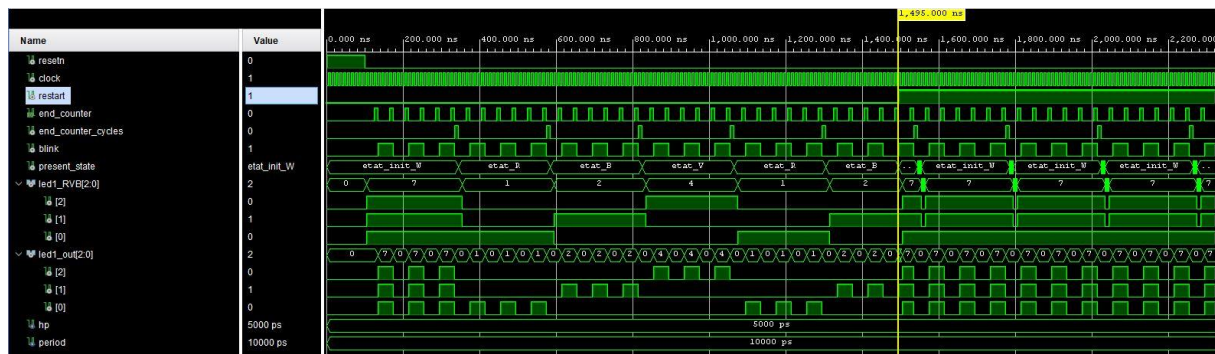
1. Voir le code counter_unit dans counter.vhd et tb_counter_fsm.vhd.
2. Voici le schéma RTL qui permet de déterminer le nombre de cycles allumé/éteint qui ont été effectués par la LED. Le compteur peut être remis à 0, maintenir sa valeur actuelle ou s'incrémenter.



3. Voir le code VHDL dans counter_fsm.vhd et tp_fsm.vhd.
4. Voir testbench de tb_counter_fsm.vhd. Voir chronogramme question 8.
5. Voici mon schéma mon schéma RTL suivi de ma FSM.



6. Les signaux d'entrées de mon architecture sont clock, resetn et restart.
Les signaux internes sont end_counter, cmd_incremt_counter, mux_incremt_1, mux_incremt_2, counter_count, cmd_reinit_counter_cycles, end_counter_cycles, cmd blink, mux_blink, blink, blink x3, led1_RBV (led1_R, led1_B, led1_V) et led2_RBV (led2_R, led2_B, led2_V).
Les signaux de sorties sont led1_out et led2_out.
7. Voir le code tp_fsm.vhd et Cora-Z7-10-Master.xdc.
8. Voir le code tb_tp_fsm.vhd.



Sur le chronogramme il y a le resetn qui reset l'ensemble de l'architecture dès le départ. Il y a la clock qui régit tout le système. Le signal end_counter_cycles vaut 1 toute les 6 periodes de end_counter ce qui permet de rythmer le changement d'état de la FSM present_state. Le signal blink permet de donner la cadence pour faire clignoter la Led rvb. Led1_RVB donne la couleur de la led à chaque état. Led1_out donne le signal de la led clignotante et son changement de couleur à chaque état. (Le schéma RTL indique 2 leds, j'ai fait le choix d'en afficher qu'une sur le chronogramme pour alléger la lecture, car le fonctionnement de la deuxième est identique à la première).

9. Voici les ressources utilisées dans la synthèse.

Report Cell Usage:

	Cell	Count
1	BUFG	1
2	CARRY4	7
3	LUT1	2
4	LUT2	34
5	LUT4	7
6	LUT6	5
7	FDCE	37
8	IBUF	3
9	OBUF	3
10	OBUFT	3

----Registers :

3 Bit Registers := 2
1 Bit Registers := 1

On voit dans registers le compteur de cycle sur 3bits, ainsi que le registre de la FSM sur 3bits également et le registre du signal blink sur 1 bit permettant de faire clignoter la LED.

On retrouve aussi dans report cell usage les entrées (IBUF) et sorties (OBUF) ainsi que le nombre total de registres (FDCE).

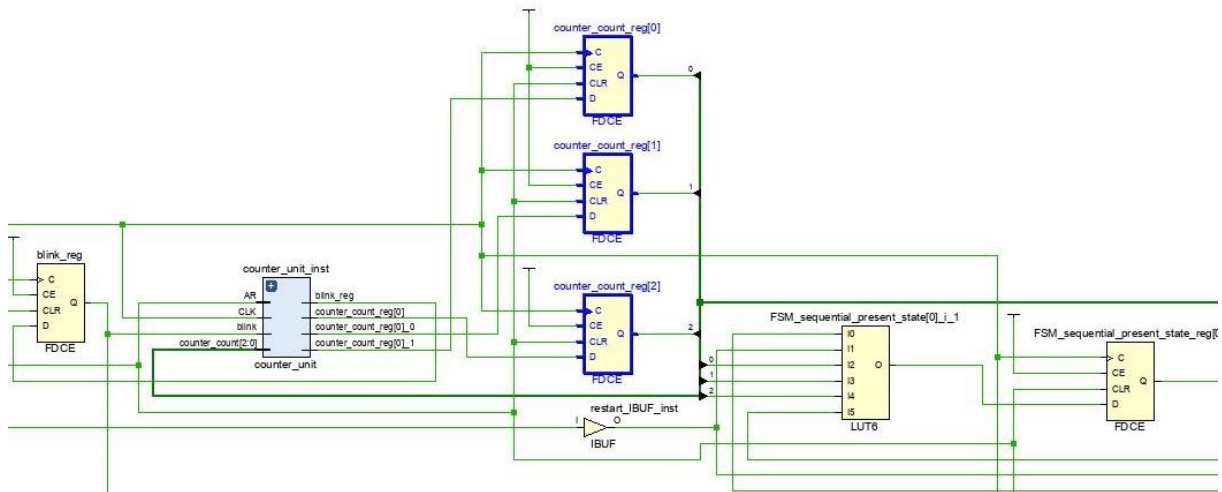
On voit ci-dessous la FSM ainsi que l'ensemble de ces états.

INFO: [Synth 8-802] inferred FSM for state register 'present_state_reg' in module 'tp_fsm'

State	New Encoding	Previous Encoding
etat_init_w	00	00
etat_r	01	01
etat_b	10	10
etat_v	11	11

INFO: [Synth 8-3354] encoded FSM with state register 'present_state_reg' using encoding 'sequential' in module 'tp_fsm'

Voici sur la schématique, où se situe le compteur de cycle (sur 3 bits).



10. Voir fichier de contrainte Cora-Z7-10-Master.xdc.

11. Voici le rapport de timing :

From Clock: sys_clk_pin
To Clock: sys_clk_pin

Setup :	0	Failing Endpoints,	Worst Slack	5.198ns,	Total Violation	0.000ns
Hold :	0	Failing Endpoints,	Worst Slack	0.214ns,	Total Violation	0.000ns
PW :	0	Failing Endpoints,	Worst Slack	4.500ns,	Total Violation	0.000ns

Les violations de setup (TNS) et de hold (THS) sont à zéro, donc il n'y a pas de violation. Le chemin critique (WNS) est de 5.198 ns.

WNS(ns)	TNS(ns)	TNS Failing Endpoints	TNS Total Endpoints	WHS(ns)	THS(ns)
5.198	0.000	0	37	0.214	0.000

12. Le système a été vérifié sur carte et fonctionne comme attendu : La led clignote 3 fois en blanc, puis idem en rouge, puis en bleu, puis en vert, puis à nouveau en rouge et ainsi de suite. Lorsque le bouton restart est actionné, la led recommence son cycle à partir de son état blanc.