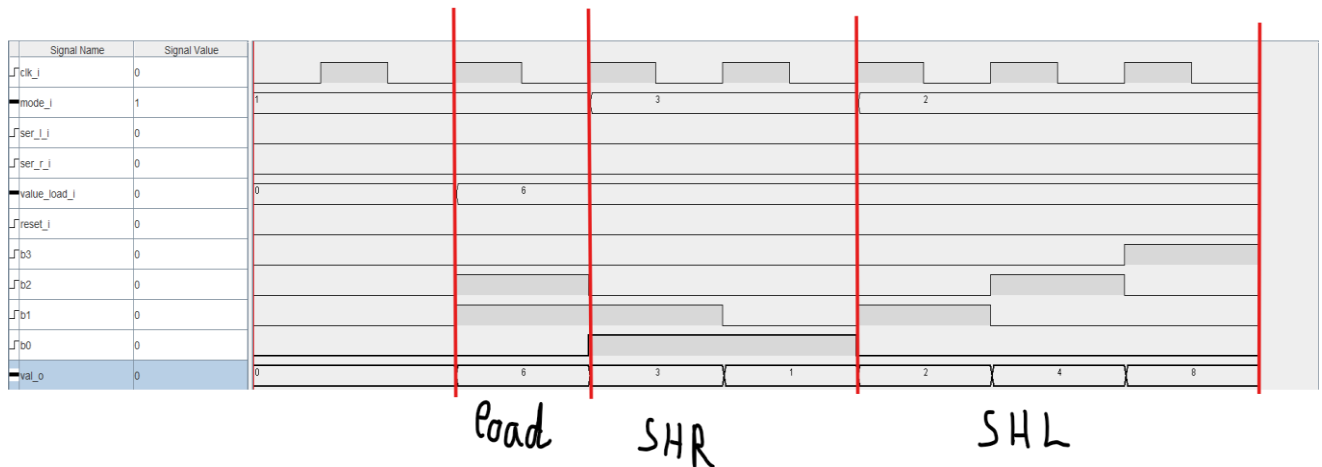


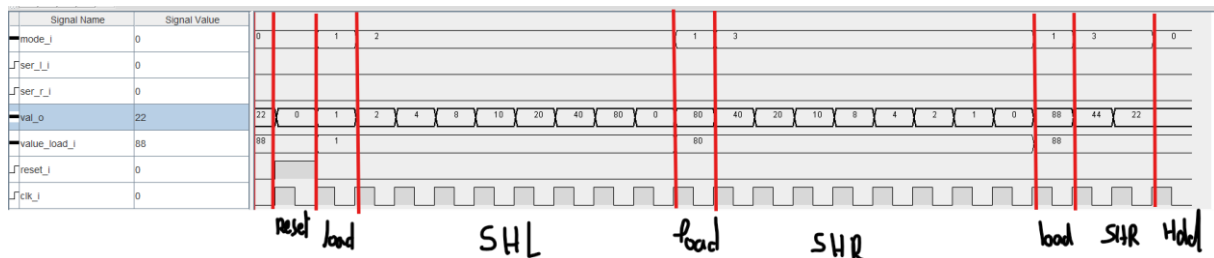
TP5 – Chenillard

Question 1



Valeur de sortie : 1000

Question 2



A notre étapes 3 qui fait un shift left d'un tour complet, nous avons une erreur à cause d'un bit qui disparaît tout à gauche mais ne revient pas tout à droite. Nous avons la même erreur pour l'étape 5 mais c'est un shift right, donc le bit disparaît à droite et en revient pas à gauche. Le reste du chronogramme est correct.

Question 3

Le décompteur va pouvoir nous laisser choisir une fréquence dans laquelle les actions du circuit seront contrôlé. Quand le décompteur arrivera à zéro cela signifiera qu'une action sera effectuée, nous pourrons donc mettre n'importe quelle fréquence que l'on souhaite et d'une façon très précis

Question 4

Fréquence visée	Nombre de cycle	Equivalence 28bits en hexadécimal	Value_i en hexadécimal
10MHz	3	0x0000003	0X0000002
2.0 Hz	15'000'000	0x0E4E1C0	0X0E4E1BF
1.0 Hz	30'000'000	0X1C9C380	0X1C9C380
0.5 Hz	60'000'000	0X3938700	0X39386FF

Question 5

Le signal PULSE_LOAD est connecté à un multiplexeur qui sélectionne entre différentes sources de données en fonction de l'état de ses entrées de contrôle. Lorsque le signal PULSE_LOAD est activé, il permet au MUX de transmettre une valeur spécifique (par exemple, un '1' ou une autre donnée) à la bascule RS. Cette bascule RS, en réponse au signal PULSE_LOAD, modifie son état pour indiquer qu'une nouvelle valeur doit être chargée. Ensuite, cette sortie de la bascule RS est connectée à l'entrée ser_L_i du registre à décalage 8 bits. Ainsi, lorsque le PULSE_LOAD est activé, il déclenche le chargement d'une valeur dans le registre à décalage, permettant de contrôler le défilement des données dans le chenillard. Ce mécanisme assure que le registre reçoit la valeur appropriée au moment voulu, synchronisé avec le reste du circuit.

Question 6

L'utilisation d'une bascule RS permet de gérer efficacement le chargement de données dans le registre à décalage. Elle offre un moyen simple et fiable de contrôler l'état de chargement, tout en garantissant que le registre ne change d'état qu'en réponse à des signaux de contrôle spécifiques. Cela contribue à la robustesse et à la prévisibilité du fonctionnement du chenillard.

Question 7

Les priorités de fonctionnement du chenillard sont

- load_i
- enable_i
- right_nleft_i

1. **load_i** : La première priorité est l'opération de chargement (load_i). Lorsque l'entrée load_i est activée, le chenillard doit immédiatement charger la valeur présente sur val_i dans le registre à décalage. Cela signifie que, peu importe l'état actuel du chenillard ou les autres signaux, le chargement d'une nouvelle valeur doit être effectué en premier. Cette priorité est essentielle car elle permet à l'utilisateur de définir une nouvelle valeur de départ pour le chenillard à tout moment, garantissant ainsi la flexibilité et le contrôle sur le fonctionnement du système.
2. **enable_i**: La deuxième priorité est l'activation du fonctionnement (enable_i). Une fois que la valeur a été chargée, le chenillard peut être activé ou désactivé en fonction de l'état de enable_i. Lorsque enable_i est à 1, le chenillard est en mode opérationnel et peut commencer à effectuer des décalages. Si enable_i est à 0, le chenillard ne doit pas effectuer de décalage, même si d'autres signaux sont actifs. Cela permet de contrôler le moment où le chenillard commence à fonctionner,
3. **right_nleft_i** : La troisième priorité concerne les opérations de décalage (SHIFT RIGHT et SHIFT LEFT). Une fois que les opérations de chargement et de maintien ont été traitées, le chenillard peut alors effectuer des décalages en fonction de l'entrée right_nleft_i . Cela signifie que, lorsque le chenillard est activé (enable_i à 1) il peut commencer à décaler les LEDs vers la gauche ou vers la droite selon la direction choisie. La fréquence de décalage est déterminée par speed_i, ce qui permet de contrôler la vitesse à laquelle les LEDs se déplacent.

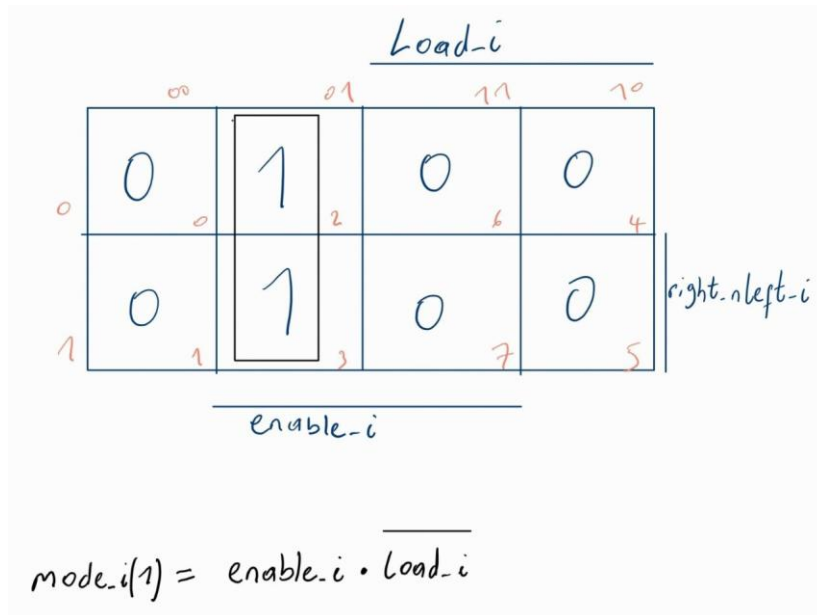
Question 8

load_i	enable_i	right_nleft_i	mode_i(1)	mode_i_(0)
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

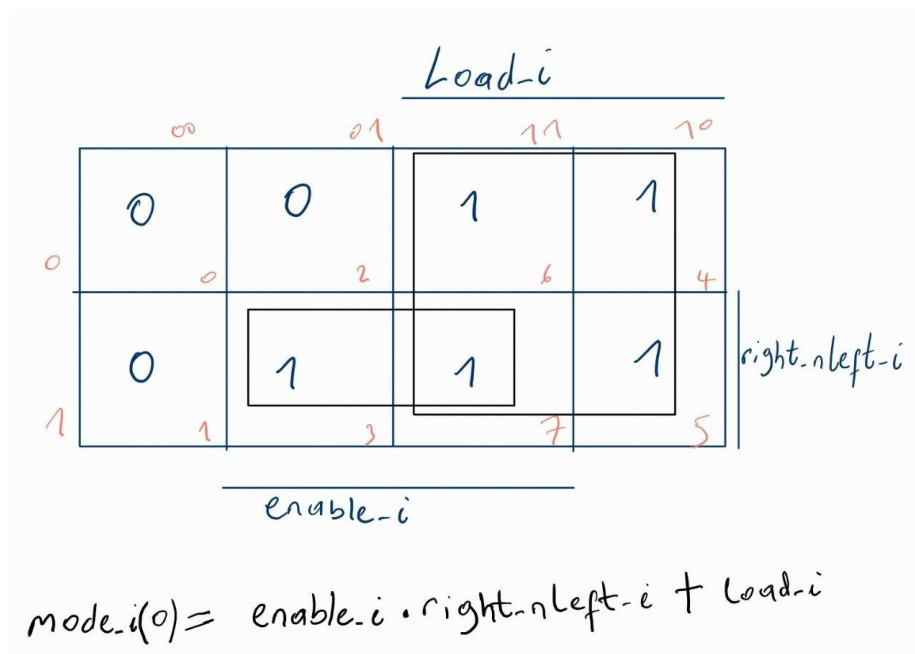
Cette table de vérité permet de gérer les différentes opérations du registre à décalage 8 bits en fonction des entrées load_i, enable_i, et right_nleft_i. Elle assure que le registre fonctionne correctement.

Question 9

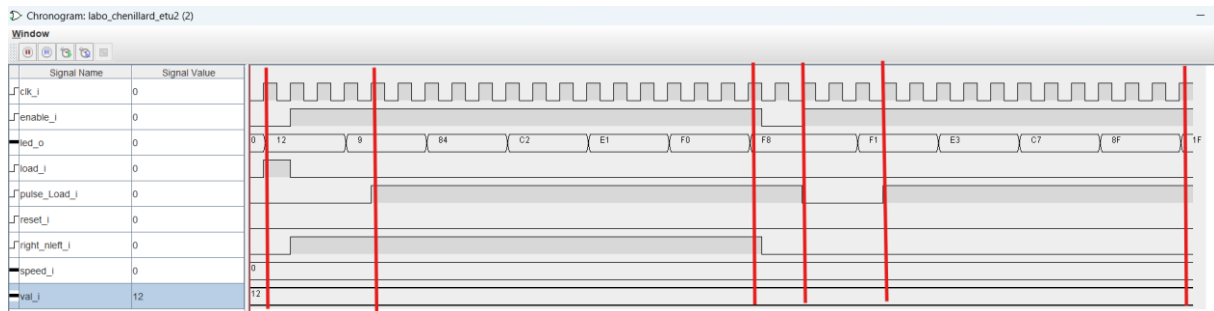
- mode_i(1)



- mode_i(0)



Question 10



Au début, on charge une valeur entre les deux premiers traits rouges. Ensuite, durant trois périodes où `right_nleft_i` et `enable` sont activés (égaux à 1), un décalage à droite est effectué. Cela devrait produire la valeur 9, comme indiqué par le signal `led_o`.

Par la suite, on effectue cinq autres décalages à droite, mais cette fois en ajoutant un 1 à gauche. On peut observer que le signal `led_o` s'incrémente correctement à chaque étape.

Enfin, en désactivant les signaux `enable` et `right_nleft_i`, des 1 sont insérés à droite. Sur plusieurs cycles d'horloge, on remarque que la valeur diminue progressivement.