
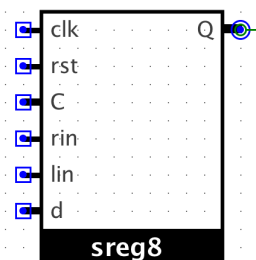














1) Mise en œuvre d'un registre à décalage

Dans la première partie de ce TP, vous mettrez en œuvre un registre à décalage 8 bits qui suit le fonctionnement décrit ci-dessous.

Pour ce faire vous complétez le circuit **sreg8** avec l'implémentation qui se basera sur l'utilisation d'un registre, de multiplexeurs, etc. Pour mettre en œuvre des « râteaux » de fils, vous utiliserez le composant *splitter* (icône ).

Symbole	Table de transition																	
	<table><tr><th>Clk</th><th>C^t</th><th>Q^t</th><th>Q^{t+1}</th></tr><tr><td></td><td>00</td><td rowspan="4">q₇q₆...q₁q₀</td><td>q₇q₆...q₁q₀</td></tr><tr><td></td><td>01</td><td>d₇d₆...d₁d₀</td></tr><tr><td></td><td>10</td><td>r_{in}q₇q₆...q₁</td></tr><tr><td></td><td>11</td><td>q₆...q₁q₀l_{in}</td></tr></table>	Clk	C ^t	Q ^t	Q ^{t+1}		00	q ₇ q ₆ ...q ₁ q ₀	q ₇ q ₆ ...q ₁ q ₀		01	d ₇ d ₆ ...d ₁ d ₀		10	r _{in} q ₇ q ₆ ...q ₁		11	q ₆ ...q ₁ q ₀ l _{in}
Clk	C ^t	Q ^t	Q ^{t+1}															
	00	q ₇ q ₆ ...q ₁ q ₀	q ₇ q ₆ ...q ₁ q ₀															
	01		d ₇ d ₆ ...d ₁ d ₀															
	10		r _{in} q ₇ q ₆ ...q ₁															
	11		q ₆ ...q ₁ q ₀ l _{in}															

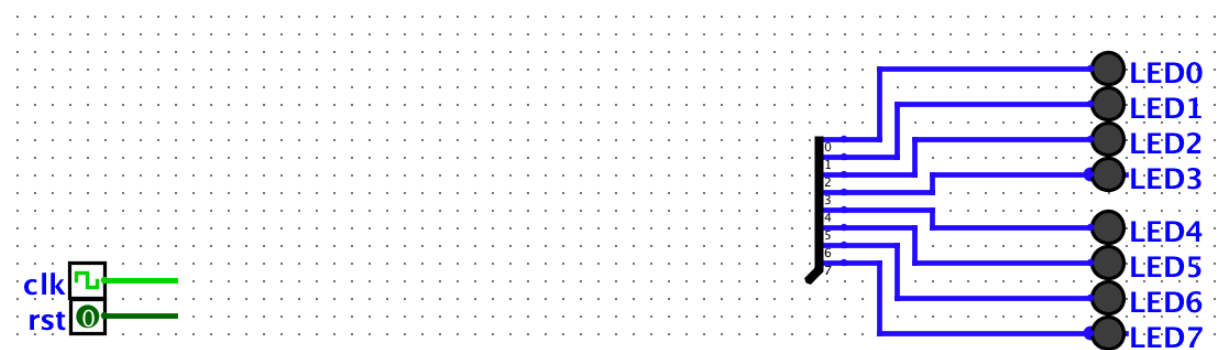
Vous validerez ensuite le fonctionnement du circuit à l'aide d'un chronogramme, en vous basant sur le banc de test **test_sreg8**.

2) Mise en œuvre d'un circuit chenillard

La seconde étape de ce TP est de réaliser un chenillard sur 8 bits en s'aidant du composant de la question 1. Le principe de fonctionnement d'un chenillard sur n bits est résumé ci-dessous :

- A tout moment, il n'y a qu'un seul bit de sortie actif (c.a.d à 1) sur les n disponibles.
- A chaque top d'horloge, le bit actif se décale d'une position vers la droite (resp. gauche).
- Lorsqu'il atteint le bit de poids faible (resp. poids fort), il change de direction de décalage.

Complétez la mise en œuvre disponible dans le circuit **chenille_sreg**, qui utilise le registre à décalage de la question 1. Vous validerez ensuite le fonctionnement de votre composant (par des chronogrammes, puis sur la carte).



Indice : en plus de la valeur courante du chenillard, qui est stockée dans le registre à décalage, il faut également penser à mémoriser la direction courante (droite, gauche) du décalage.

3) Réalisation d'un chenillard à partir d'un diagramme d'état

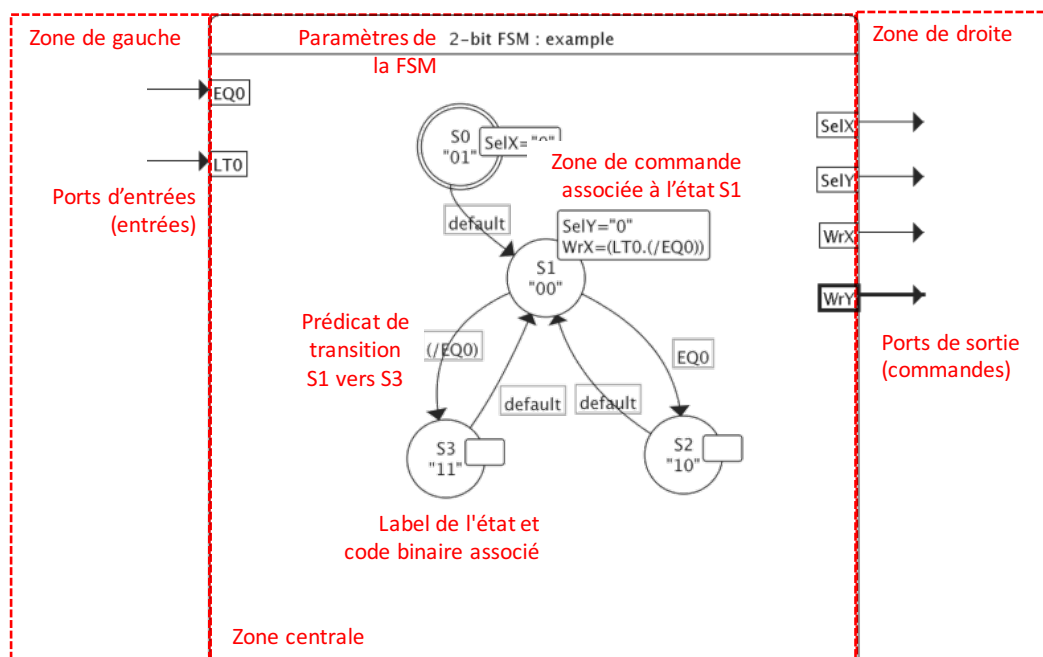
Comme vous avez pu vous en rendre compte, la mise en œuvre de la logique de contrôle du registre à décalage n'est pas si simple. Afin de se simplifier le travail, les concepteurs évitent ce type d'approche ad-hoc, et préfèrent se baser sur l'utilisation de machines à état spécifiées à partir de diagrammes d'états, celles-ci étant plus rapides à spécifier et plus simples à comprendre par d'autres concepteurs.

Dans cette partie, vous allez implémenter le même chenillard qu'à la question précédente, mais cette fois-ci directement sous forme de diagramme d'états.

- Combien y-a-t-il d'états différents dans un chenillard sur 8 bits ?
- En vous basant sur la réponse à la question précédente, de combien de bascule D a-t-on besoin pour réaliser la machine à état du chenillard ?
- Comment expliquez-vous la différence par rapport à l'approche de la question 2 ?

Ouvrez le circuit **chenillard_fsm** dans l'éditeur. Celui-ci contient un composant machine à état (FSM chenillard) avec trois entrées (**clk**, **en**, **clr**) et une sortie (**L[7:0]**). Pour visualiser/modifier le diagramme de transition associé à ce circuit, sélectionnez la machine à l'aide de la souris, placez-vous dans l'onglet *properties* en bas à gauche de l'éditeur, puis cliquez sur (*click to edit*).

Une nouvelle fenêtre devrait s'ouvrir, celle-ci se décompose en plusieurs zones, représentées ci-dessous.

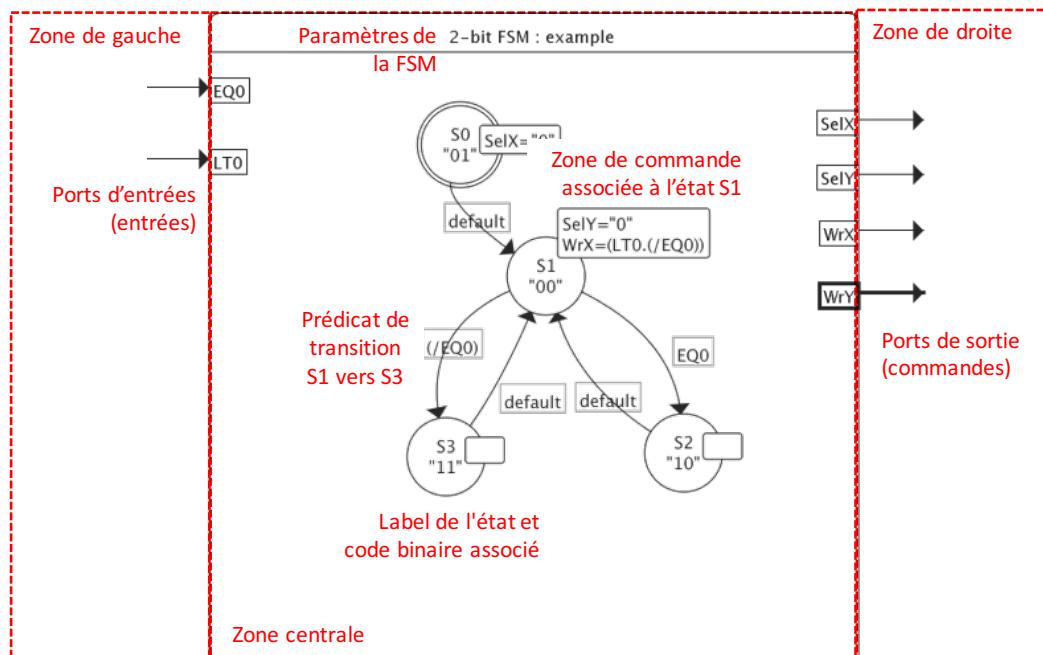


Les commandes de l'éditeur sont décrites dans l'annexe A, celles-ci permettent d'ajouter, de modifier, ou de supprimer des états, transition, commandes ou entrées/sorties. Les prédicats de transition et les commande s'expriment à l'aide d'un langage simple dont les principes sont également décrits dans l'annexe A.

Complétez le diagramme de transition du chenillard, puis vérifiez ensuite son bon fonctionnement en simulation, puis sur la carte à l'aide du fichier **test_chenille_fsm.circ**.

Annexe A : éditeur de machine à état

La fenêtre de l'éditeur de machine à état se décompose en plusieurs zones, représentées sur la copie d'écran annotée fournie ci-dessous.



Les opérations courantes sont résumées dans la table ci-dessous :

Quoi faire	Comment le faire
Ajouter un état	Placez-vous sur la zone centrale et faites apparaître le menu contextuel (clic droit). Sélectionnez l'entrée <i>add state</i> , puis déplacez-vous pour placer le nouvel état à l'endroit qui vous convient. Validez avec un clic gauche.
Supprimer un état	Sélectionnez un état (clic gauche), puis faites apparaître le menu contextuel (clic droit). Sélectionnez l'entrée <i>delete</i> . L'état et toutes les transitions entrantes sortantes seront supprimés du diagramme.
Ajouter une transition	Sélectionnez un état (clic gauche), puis faites apparaître le menu contextuel (clic droit). Sélectionnez l'entrée <i>add transition</i> . Une transition partant de l'état sélectionné doit apparaître, sélectionner l'état destination en déplaçant le pointeur de souris vers l'état concerné, puis en validant par un clic gauche.
Ajouter un port d'entrée	Placez-vous sur la zone de gauche et faites apparaître le menu contextuel (clic droit). Sélectionnez l'entrée <i>add input port</i> , puis déplacez-vous sur la bordure gauche du diagramme pour placer le nouveau port, validez avec un clic gauche.
Ajouter un port de sortie (commande)	Placez-vous sur la zone de droite et faites apparaître le menu contextuel (clic droit). Sélectionnez l'entrée <i>add output port</i> , puis déplacez-vous sur la bordure droite du diagramme pour placer le nouveau port, validez avec un clic gauche.
Modifier le prédicat associé à une transition/commande	Double-cliquez sur le prédicat associé à la transition/commande qui vous intéresse, une boîte de dialogue s'ouvrira alors qui vous permettra de spécifier sous forme textuelle un nouveau prédicat/nouvelle commande. Si le texte saisi n'est pas syntaxiquement correct (voir syntaxe plus loin), un message d'erreur apparaît.

Les valeurs assignées aux ports de sortie sont exprimées à l'aide d'expressions dont le résultat est un mot binaire. La syntaxe de ces expressions est décrite ci-dessous. Les assignations sont séparées entre elles par un point-virgule.

$ident := expr$	Assignation d'une valeur $expr$ à la sortie $ident$
$exp := expr . expr$	Et logique sur les opérandes (exemple $A+B$)
$exp := expr + expr$	Ou logique (exemple $A+B$)
$exp := /expr$	Négation logique (exemple $/X$)
$exp := expr == expr$	Egalité (exemple $A[2:1]==B[3:2]$)
$exp := expr != expr$	Différence (exemple $A != "010"$)
$expr := expr [val:val]$	extraction (pour mot de n bits) (exemple $X[3:2]$)
$expr := \{ ' expr, expr ' \}$	Concaténation (exemple $\{A[1], "0"\}$)
$expr := " \{0,1\}^* "$	Constantes binaire (ex $"010010"$)
$expr := ident$	Identificateur (nom d'un port d'entrée)

Les prédicats de transition suivent ce même principe, mais ceux-ci doivent forcément s'évaluer comme un résultat booléen (0 ou 1). Il est par ailleurs également possible d'utiliser le mot clé "default" qui est évalué à vrai lorsque toutes les autres transitions issues du même état sont fausses.