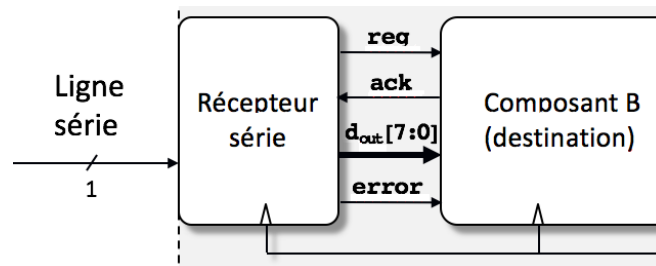


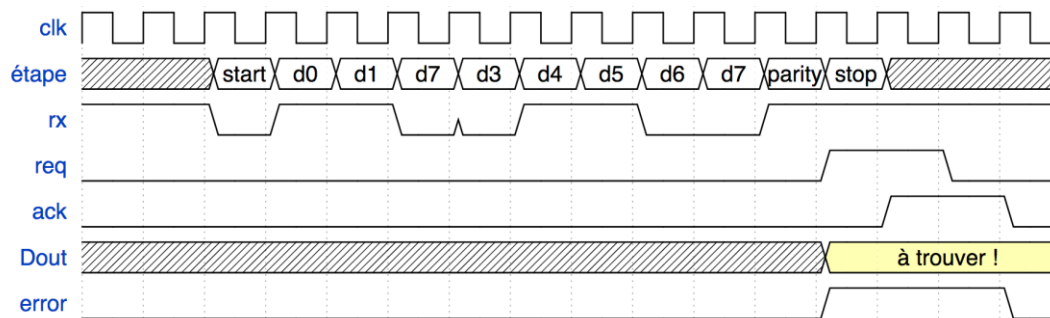
## TP3 : récepteur série

Dans le TD3, vous avez conçu un circuit d'émission de données en série, cette séance de TP va maintenant vous amener à concevoir le circuit récepteur.

Son rôle, illustré via le diagramme fonctionnel ci-dessous, est de reconstruire la valeur 8 bit émise via la liaison série, pour ensuite l'envoyer vers un autre composant via le port **dout**, et en utilisant le protocole REQ/ACK étudié en CM et en TD.



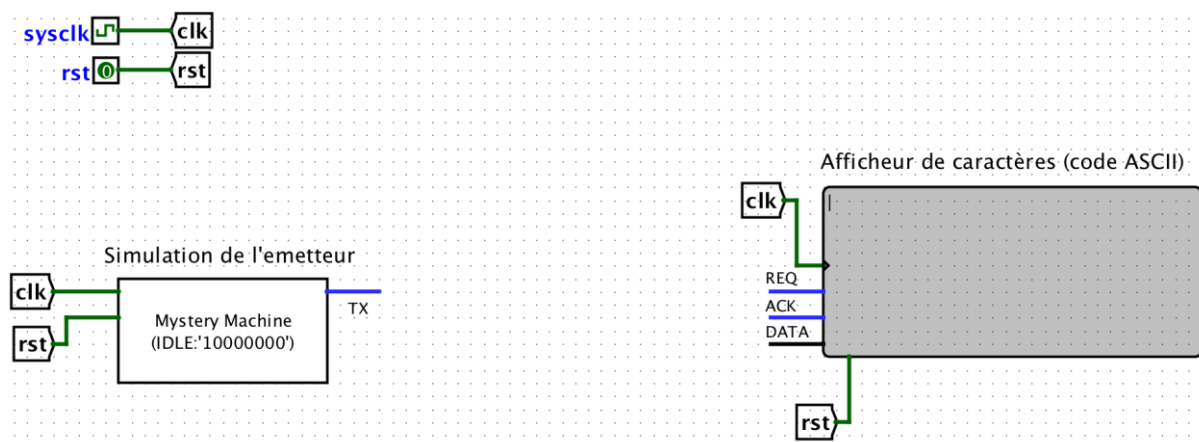
Le fonctionnement du récepteur est très similaire à celui de l'émetteur, et est illustré par le chronogramme fourni ci-dessous.



La principale différence réside dans l'étape de détection d'erreur : une fois la donnée sur 8 reconstruite à partir du flux série, le récepteur calcule la valeur du bit de parité associé à cette valeur, et le compare la valeur de parité transmise par l'émetteur. En cas de non correspondance, cela signifie qu'une erreur de transmission a eu lieu.

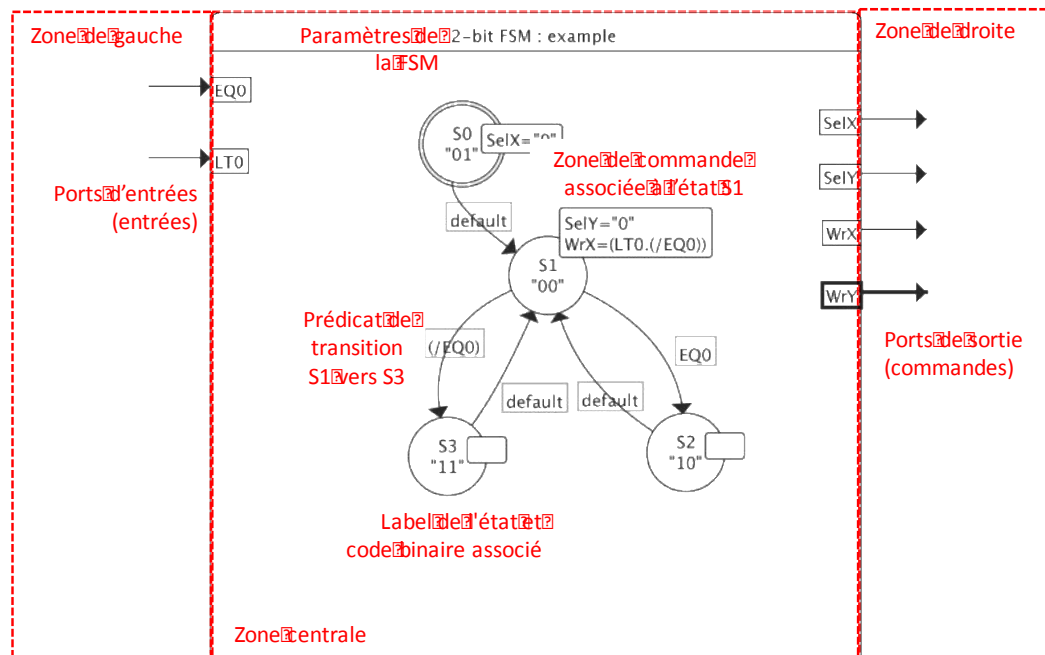
### Travail à réaliser

Complétez le circuit ci-dessous, en y ajoutant une UT et une UC.



## 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 (en notation BNF) est décrite ci-dessous. Les assignations sont séparées entre elles par un point-virgule.

$ident = expr$	Assignation d'une valeur $expr$ à la sortie $ident$ ( $Y="0"$ )
$exp := expr . expr$	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 **default** qui est évalué à vrai lorsque toutes les autres transitions issues du même état sont fausses.