

## INTRODUCTION AUX SYSTÈMES SÉQUENTIELS

### Table des matières

<b>1</b>	<b>Les machines à état</b>	<b>2</b>
1.1	Quelques rappels . . . . .	2
1.2	L'auto maintien . . . . .	2
1.3	Un tableau de classe . . . . .	2
<b>2</b>	<b>Machine à état avec des bascules RS</b>	<b>3</b>
2.1	Méthode . . . . .	3
2.2	Télérupteur . . . . .	4

# 1 Les machines à état

## 1.1 Quelques rappels



### Définition

Une machine d'état est une abstraction mathématique utilisée pour concevoir des algorithmes. Une machine d'état lit un ensemble d'entrées et passe à un état différent en fonction de ces entrées.

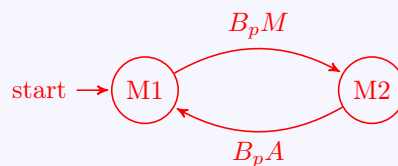
Nous nous intéresserons aux machines déterministes à états finis. Cela signifie que lorsque la machine se trouve dans un état, il est possible de savoir vers quel état la machine va évoluer lorsque l'on connaît les entrées.

## 1.2 L'auto maintien

Nous proposons ici d'implémenter l'auto-maintien sous la forme d'une machine à états.

### Activité 1

**Question 1** Dessiner la machine à Etat d'un auto-maintien commandé par les boutons  $B_pM$  (Marche) et  $B_pA$  (Arrêt).



## 1.3 Un tableau de classe

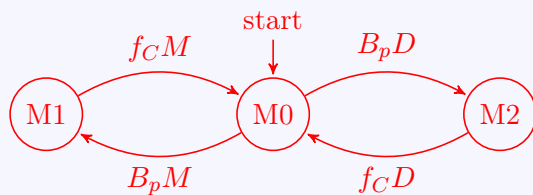
Soit un tableau de classe équipé de deux boutons poussoirs et de deux capteurs fin de courses ( $f_C H$  et  $f_C B$  et qui suit le comportement suivant :

### Cahier des charges - Tableau mobile

- Le tableau est à l'arrêt
- Si le tableau n'est pas déjà en position haute, un appui sur  $B_pM$  met en route le moteur en sens montée jusqu'à ce qu'il arrive en haut
- Si le tableau n'est pas déjà en position basse, un appui sur  $B_pD$  met en route le moteur en sens de descente jusqu'à ce que le volet arrive au capteur

**Activité 2**

**Question 2** Dessiner la machine à état associée à ce comportement



## 2 Machine à état avec des bascules RS

### 2.1 Méthode

Comme pour le LADDER, il est possible d'implémenter une machine à état sous la forme d'un circuit logique. Pour cela, nous utiliserons des bascules RS afin de gérer les états.



#### À retenir

Une démarche pour traduire une machine à état en circuits logiques est la suivante :

1. **Dessiner** la machine à état
2. Insérer une bascule et un memento pour chaque état
3. Implémenter les transitions
  - Prendre en compte l'état précédent
  - Implémenter l'équation
  - Désactiver l'état précédent tout en activant le suivant
4. **Implémenter les sorties** associées aux états

Par convention, les noms des variables par un  $M$  pour correspondre aux moments de LogoSoft.

### Représenter un état



#### À retenir

Un état est représenté par une bascule RS en série avec un memento.

Le passage de la transition de l'état suivant doit désactiver l'état actuel

**Implémenter les transitions** On passera d'un état au suivant si et seulement si la réceptivité de la transition est vérifiée et que l'état l'état précédent est actif. Cela signifie qu'une porte ET sera obligatoirement insérée après chaque état pour prendre en compte ces deux paramètres.

Aussi, le passage d'une transition **doit** désactiver l'état précédent

**Association des sorties** On ajoute enfin, à la suite de chaque état, les sorties qui sont activés lors de l'activation de chaque état.

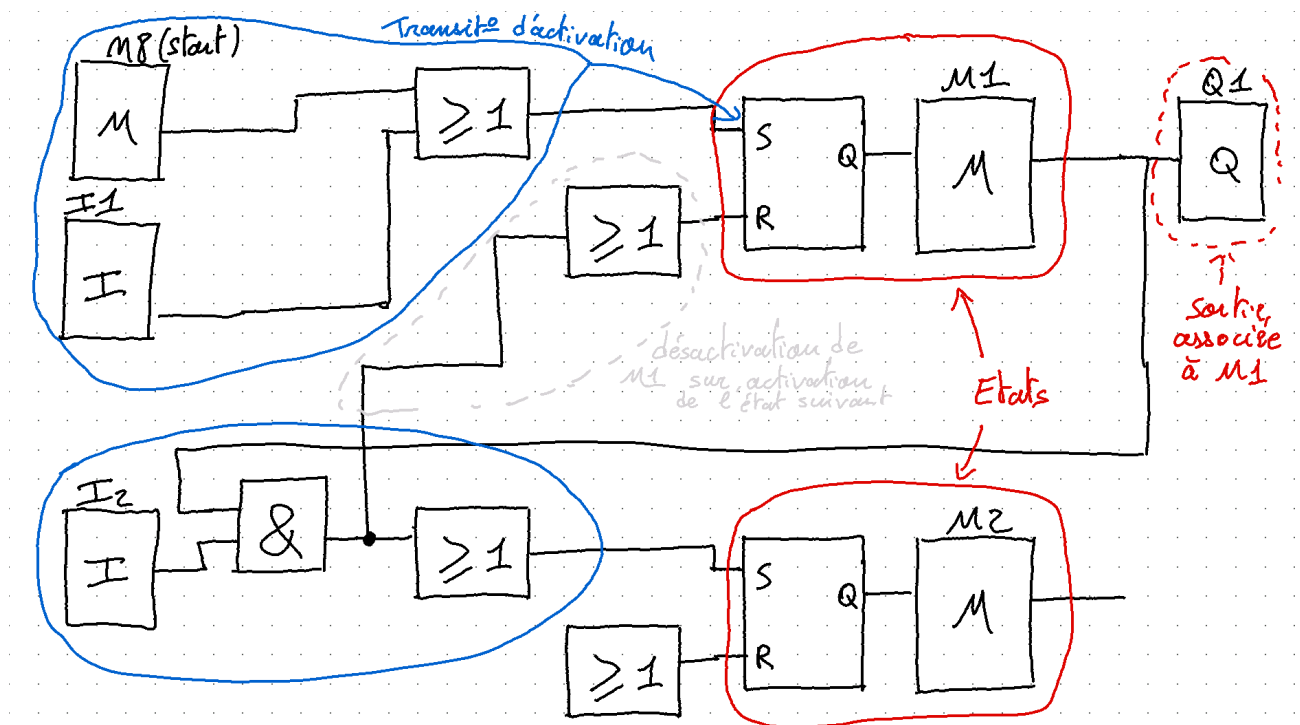


FIGURE 1 – Implémentation d'une machine à état en circuit logique

## 2.2 Télérupteur

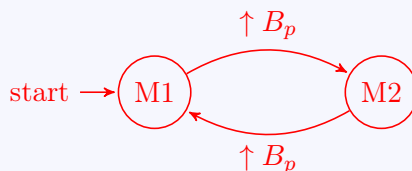
Reprenons l'exemple du télérupteur. Un télérupteur a le comportement suivant :

### — Cahier des charges - Télérupteur —

- Au départ la lumière est éteinte
- Un appui sur le bouton *BP* allume la lumière
- Un nouvel appui sur le bouton *BP* éteint la lumière et on reprend à l'état initial

### Activité 3

**Question 3** Dessiner la machine à état d'un télérupteur avec puis sans utilisation de front montant



**Question 4** En suivant le protocole, implémenter cette machine à état à l'aide d'un circuit logique.