

M1 SESI 2017-2018  
Architecture Multi-Processeurs

Kevin Mambu

February 2, 2018

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Automates d'Etats Finis Communiquants Synchrones</b>	<b>3</b>
2.1	Automates de Mealy vs Automates de Moore . . . . .	4
<b>3</b>	<b>Méthode Générale de Synthèse des FSM</b>	<b>5</b>
3.1	ETAPE A . . . . .	5
3.2	ETAPE B . . . . .	6
3.3	ETAPE C . . . . .	6
3.4	ETAPE D . . . . .	6
3.5	ETAPE E . . . . .	6
3.6	ETAPE F . . . . .	6
3.7	Exemple : capteur de trois '1' consécutifs, MOORE . . . . .	7

# 1 Introduction

Responsable : Alain Grener, alain.grener@lip6.fr

Problématique de l'UE : synchronisation de plusieurs composants dans une architecture multi-processeur. Exemple : accès du même périphérique par plusieurs processeurs.

Observer le moyen de communication entre processeurs et l'environnement des processeurs.

Un processeur pipeline reste synchronisé : plusieurs instructions sur le même flux de transition. Dans le cas d'une architecture multi-processeurs  $\Rightarrow$  les unités sont nativement asynchrones, mais doivent se synchroniser.

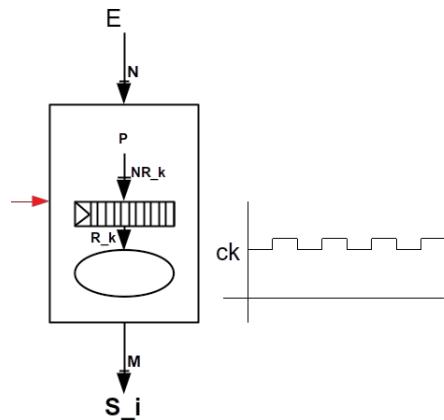
## 2 Automates d'Etats Finis Communiquants Synchrones

Il s'agit d'un modèle mathématique, qui permet de décrire tous les systèmes matériels numériques.

"Les états" ici est l'ensemble des valeurs intégrées à l'automate des ensembles mémorisants. Un état est une valeur stockée dans les registres.

Exemple avec le processeur MIPS32 : comptons le banc de registres, PC, HI, LO, etc. On peut comptabiliser environ 50 registres, dans ce cas le processeur MIPS32 possède  $2^{50}$  états.

À chaque front montant, on met à jour l'ensemble des registres internes. Les sorties d'un automate sont les entrées d'un autre. Un automate produit est l'ensemble des sous-systèmes (automates).



$$S_i = g_i(E_0, E_1, \dots, E_{(N-1)}, R_0, R_1, \dots, R_{(P-1)}),$$

M fonctions booléennes dépendant de  $N + P$  bits.

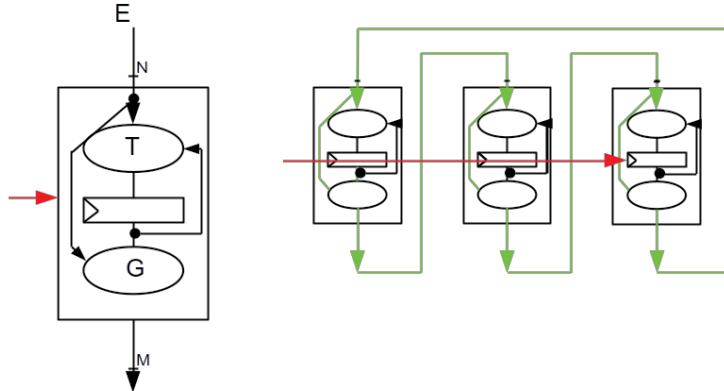
$$NR_k = T_k(E_0, E_1, \dots, E_{(N-1)}, R_0, R_1, \dots, R_{(P-1)}),$$

P fonctions booléennes dépendant de  $N + P$  bits.

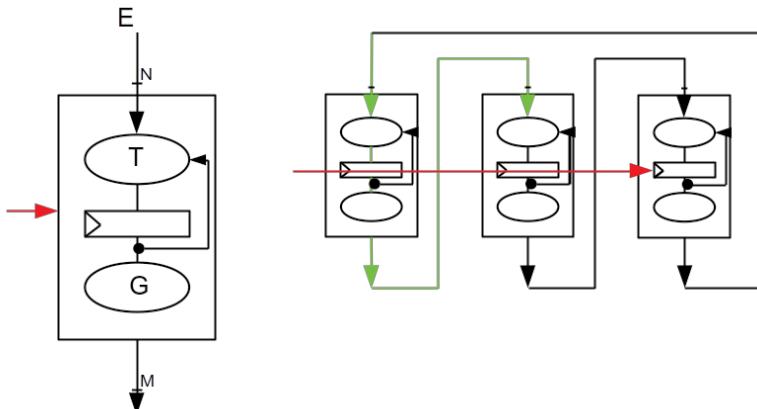
Pour décrire complètement le comportement d'un automate particulier, il faut :

1. quelles valeurs vont prendre les sorties en fonction des entrées  $E_i$  et de l'état.  $\Rightarrow$  décrire un système séquentiel (à mémoire).
2. les valeurs de l'état suivant  $NR_k$  en fonction des entrées  $E_i$  et de l'état courant  $R_k$ .
3. un mécanisme permettant d'initialiser l'état interne  $\Rightarrow$  état initial  $R_k^0$ .

## 2.1 Automates de Mealy vs Automates de Moore



La génération de la sortie dépend de l'état de transition et des entrées



La génération de la sortie dépend uniquement de l'état de transition

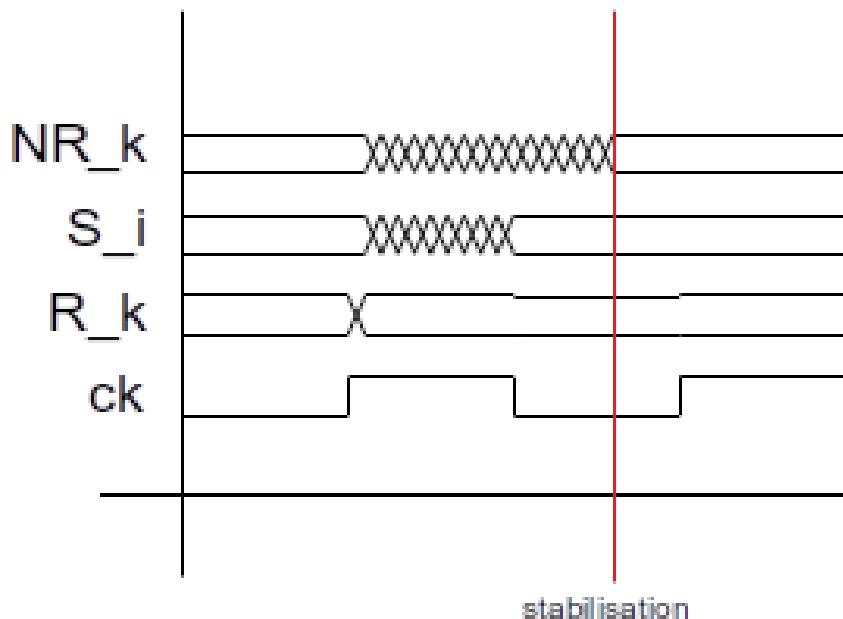
Avantages de l'Automate de Moore :

- Cet automate est facilement prédictible (synchronisation assurée)
- Beaucoup plus stable sur la propagation de l'information

Inconvénients :

- Moins réactif

Le plus long chemin possible est le temps de cycle minimal d'un registre à un autre.



La génération de la sortie dépend uniquement de l'état de transition

### 3 Méthode Générale de Synthèse des FSM

Toute l'industrie de la CAO repose sur la technologie des FSM.  
⇒ Par extension, toute l'industrie du High-Tech.

#### 3.1 ETAPE A

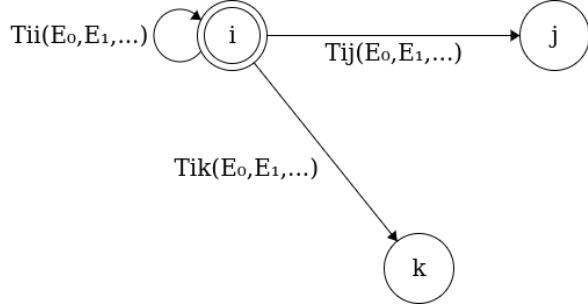
Définition des "états abstraits" Il s'agit d'identifier les états avec des symboles et faire abstraction de leur description pour leur nomination.

### 3.2 ETAPE B

Représentation abstraite du graphe de transition.

Il va s'agir d'un graphe orienté dont on doit précisément spécifier :

- Les états (un noeud par état accessible)
- Les transitions (les arcs orientés) :



On doit s'assurer des conditions d'orthogonalité et de complétude :

- orthogonalité  $\Rightarrow$  2 transitions de sortie ne peuvent pas être simultanément accessibles :

$$\forall E_i \forall i \forall j \forall k (T_{ij} \bullet T_{ik}) \equiv 0 \text{ si } i \neq j$$

- complétude  $\Rightarrow$  Il y a toujours une transition de sortie qui est vraie :

$$\forall E_i \forall i \sum j = 0(T_{ij}) = 1$$

Nb : Les états sont étiquetés en fonction de fonction booléenne de l'expression de la valeur de sortie

### 3.3 ETAPE C

Choix d'un codage des états

### 3.4 ETAPE D

Traduction du graphe en table de vérité

### 3.5 ETAPE E

Déterminer les équations booléennes simplifiées.

### 3.6 ETAPE F

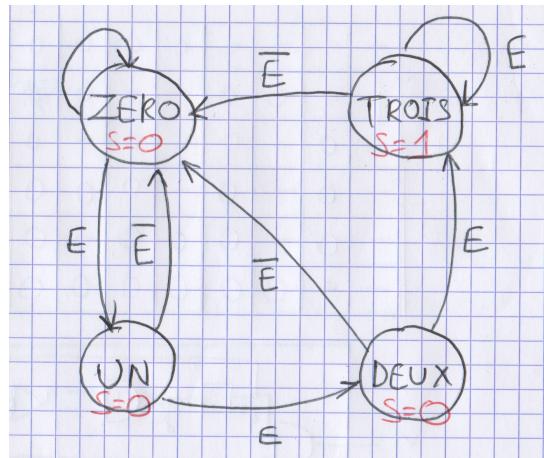
Traduction en portes logiques

### 3.7 Exemple : capteur de trois '1' consécutifs, MOORE

Etape A :

- "ZERO", aucun chiffre mémorisé
- "UN", le dernier caractère détecté est 1
- "DEUX", les deux derniers caractères détectés sont '1' et '1'
- "TROIS", les trois derniers caractères détectés sont '1','1' et '1'

Etape B :



Etape C :

	$R_1$	$R_0$	$R_3$	$R_2$	$R_1$	$R_0$
ZERO	0	0	0	0	0	1
UN	0	1	0	0	1	0
DEUX	1	0	0	1	0	0
TROIS	1	1	1	0	0	0

NBC                      one-hot

Etape D

$R_1$	$R_0$	$E$	$NR_1$	$NR_0$	$S$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Etape E

- $S = R_1 \bullet R_0$
- $NR_1 = E \bullet (R_0 + R_1)$
- $NR_1 = E \bullet (\overline{R_0} + R_1)$

Etape F

