

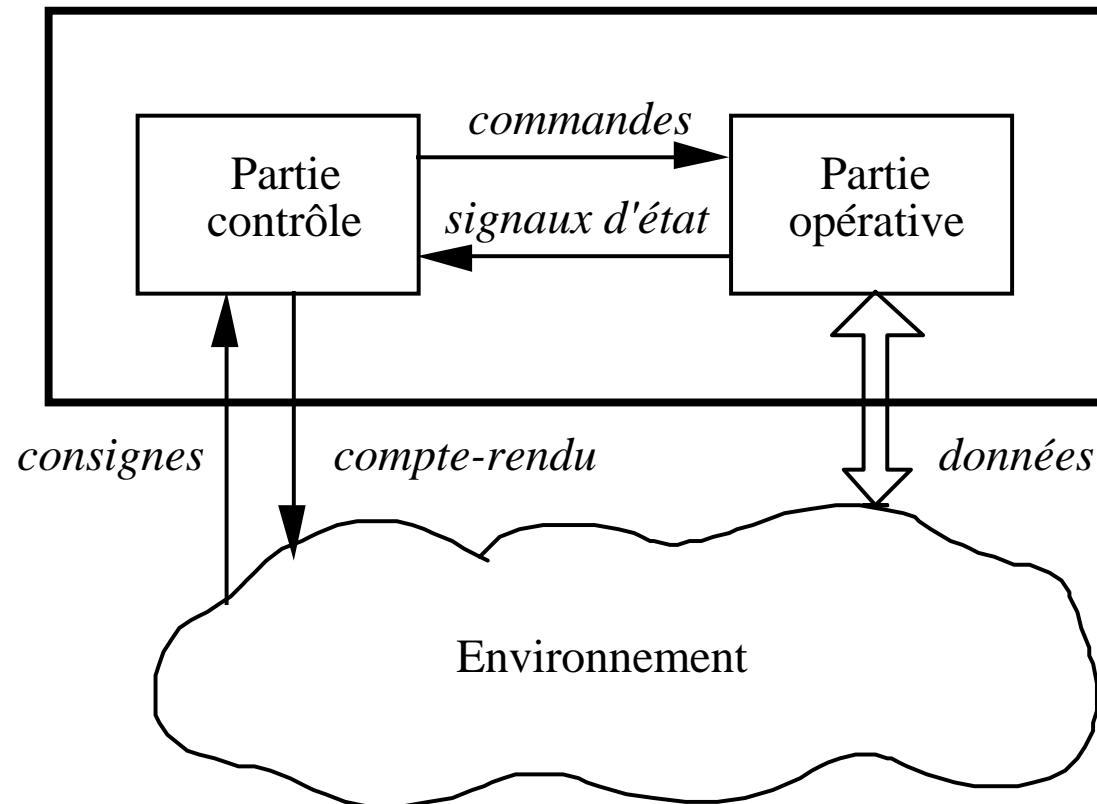


## Cours 9

**Introduction aux systèmes séquentiels complexes.  
Conception de machines à états.**



# Structure générale d'un système numérique



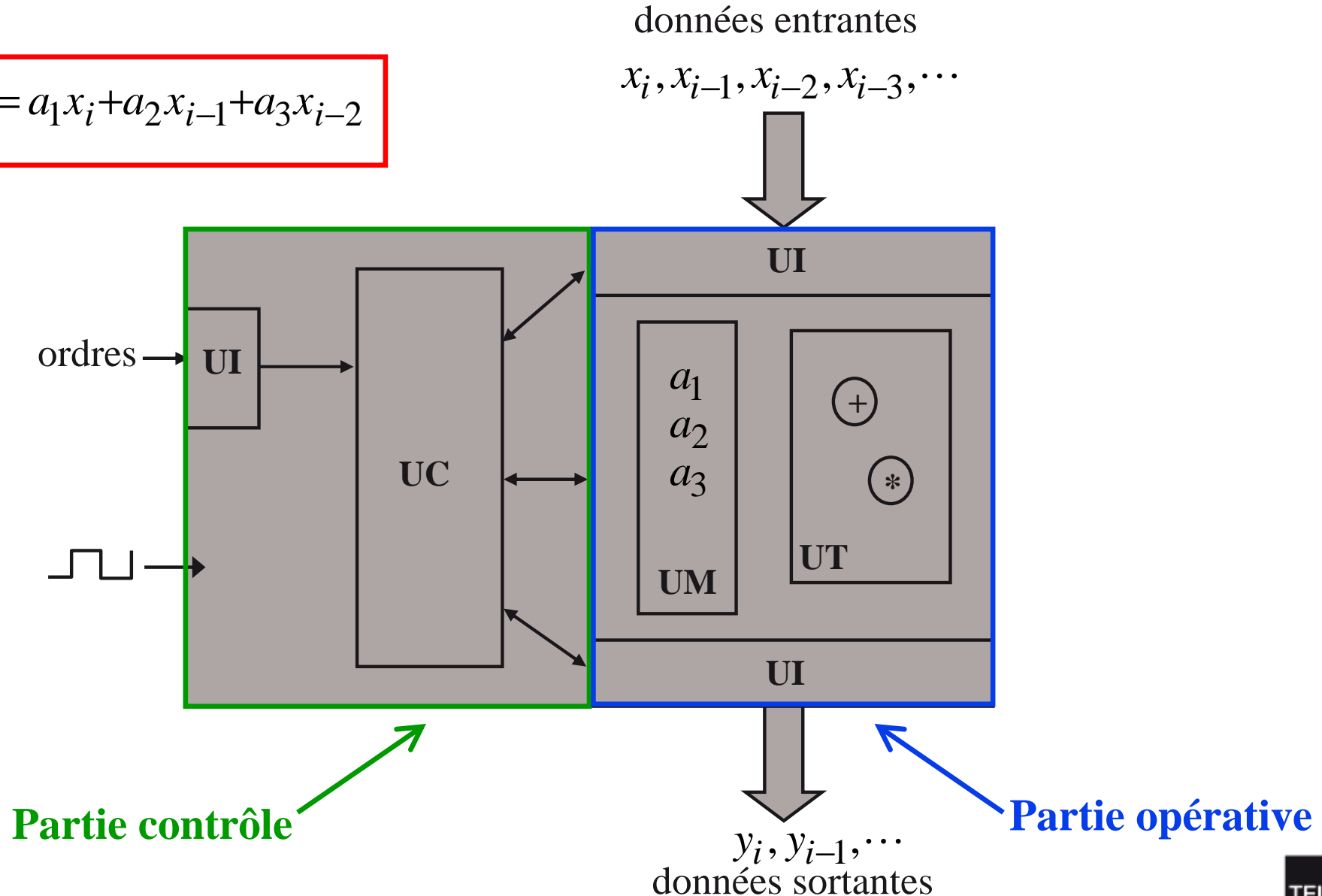
- **Partie opérative** : traite les données
- **Partie contrôle** : pilote le traitement

# Les quatre types d'unités fonctionnelles dans un système numérique

- Unité de **traitement**      => Traiter
- Unité de **mémorisation**      => Stocker
- Unité de **contrôle**      => Piloter
- Unité d'**interface**      => Communiquer

# Assemblage des unités fonctionnelles : exemple d'un filtre numérique programmable

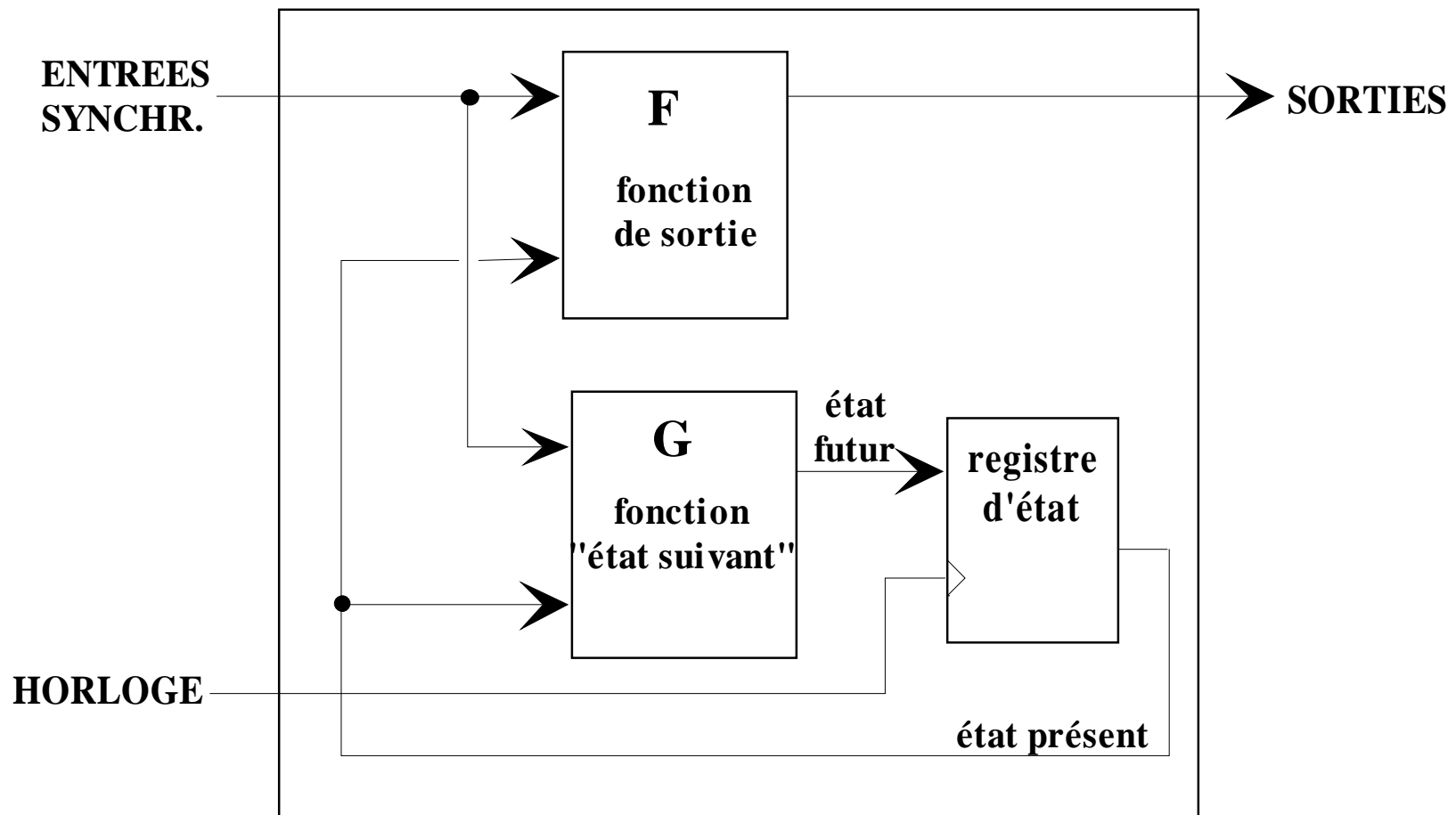
$$y_i = a_1x_i + a_2x_{i-1} + a_3x_{i-2}$$



# Conception d'une unité de contrôle

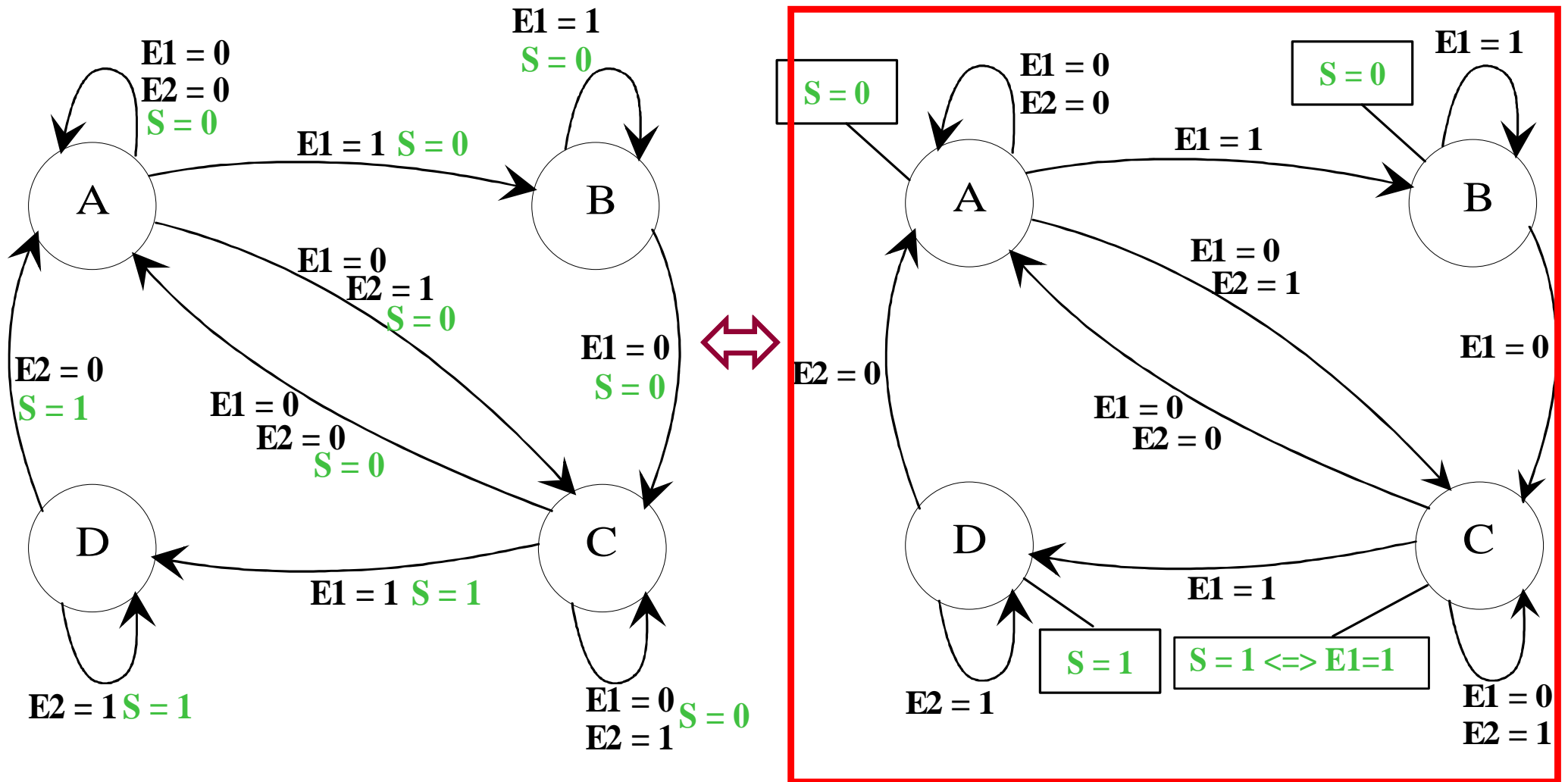
- **Opérateurs séquentiels spécialisés dans le contrôle**
  - => **machine à états finis (automate)**
  - => **séquenceur**
  - => **microcontrôleur**

# Les machines à états : machine de Mealy

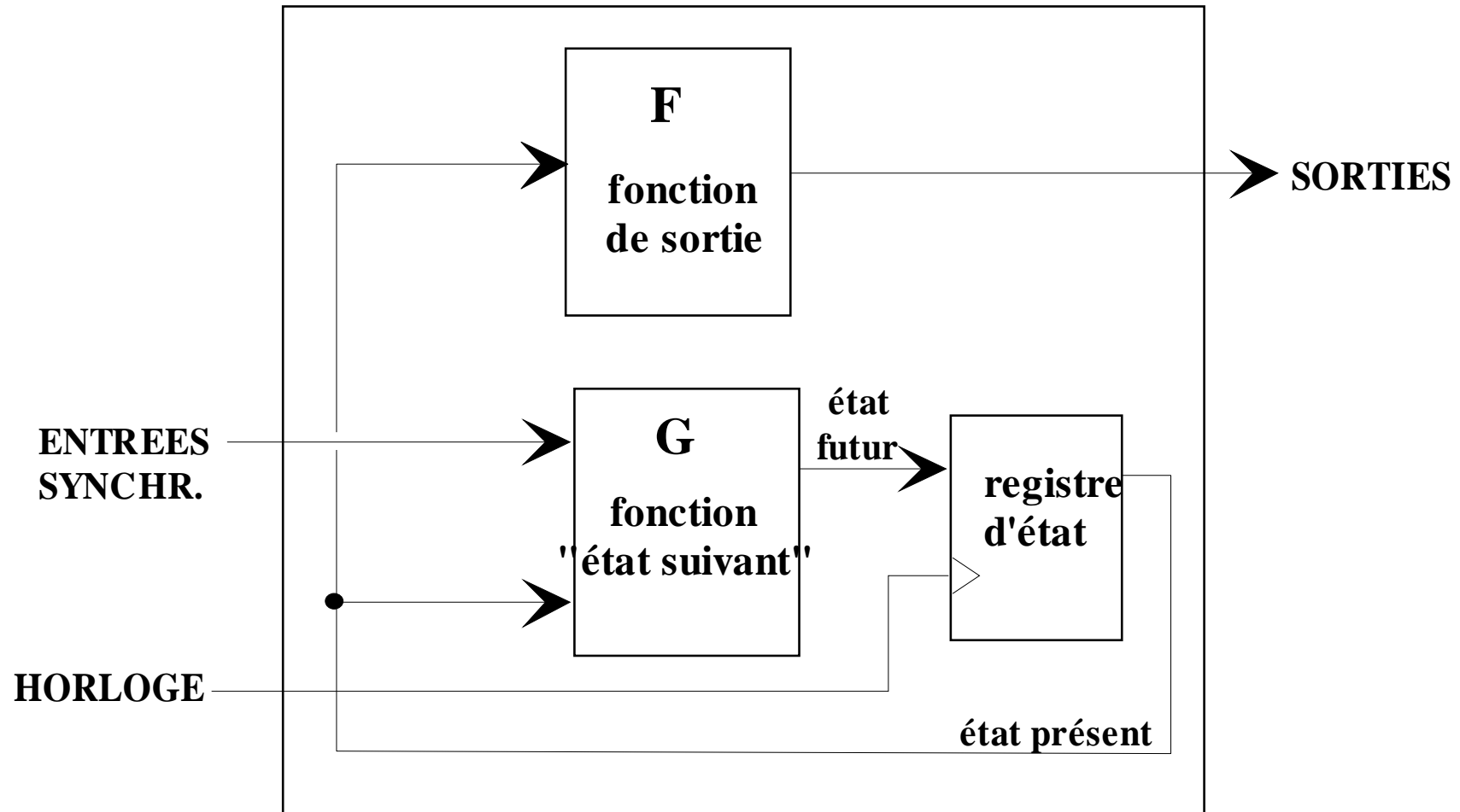


**Machine de Mealy synchrone**

# Comportement d'une machine de Mealy : graphe d'états



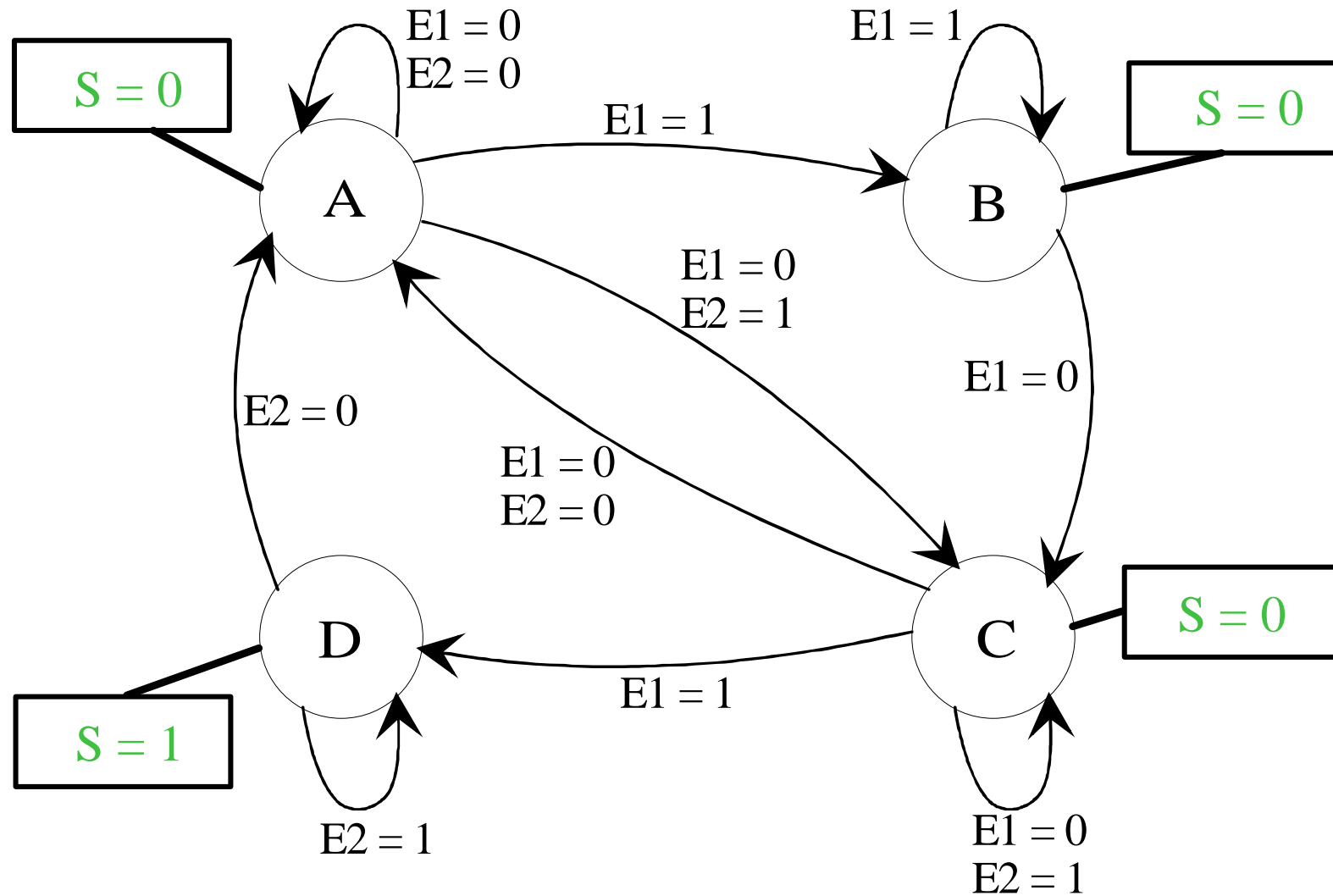
# Les machines à états : machine de Moore



## Machine de Moore synchrone



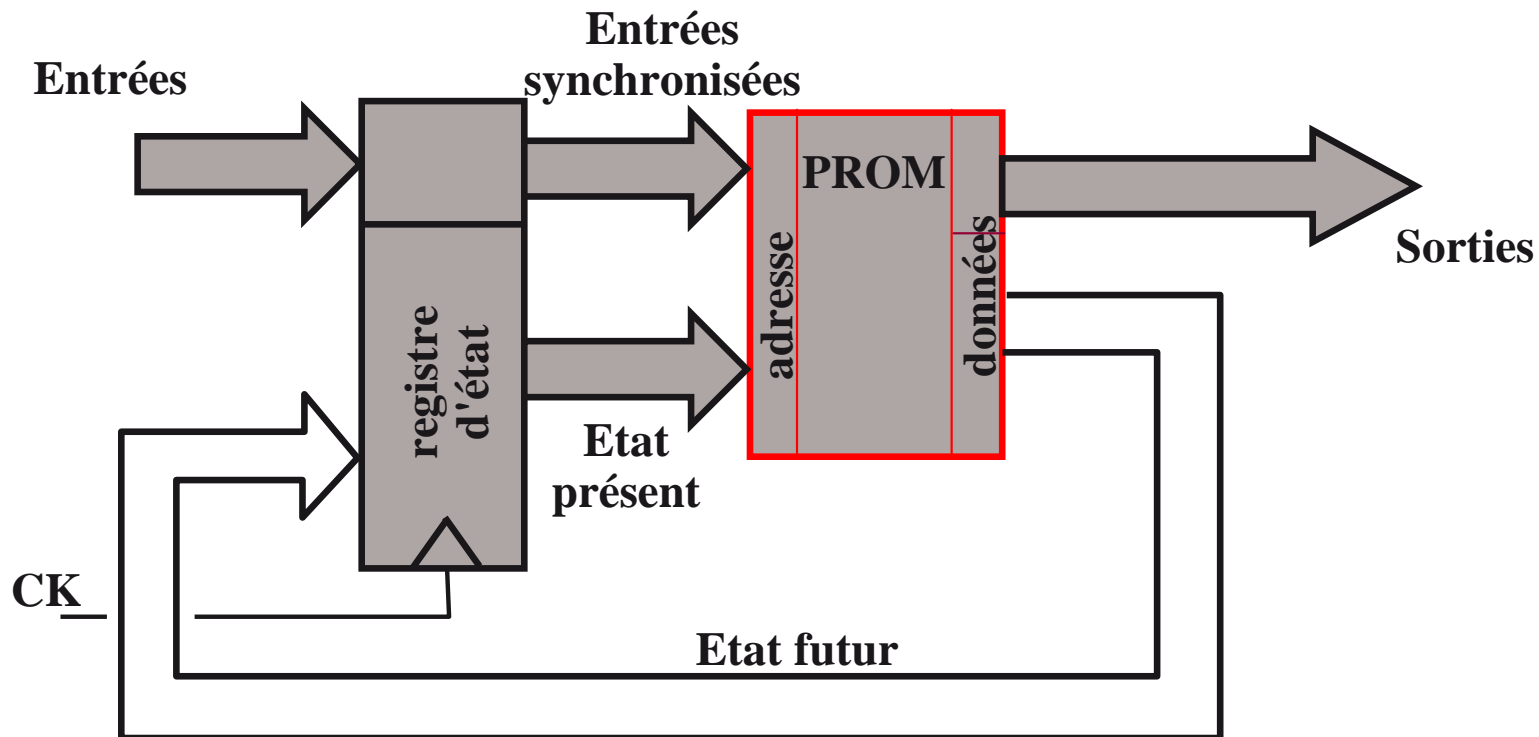
# Graphe d'états d'une machine de Moore



# Mise en œuvre des automates

## ■ Les fonctions F et G peuvent être réalisées :

- A l'aide d'opérateurs logiques élémentaires
- A l'aide d'une PROM :



# Limitations des automates d'états finis

- **Un automate d'états finis ne peut être mis en œuvre que si la spécification de l'unité de contrôle est relativement simple**
  - ↪ Les graphes d'états ne sont pas adaptés à la description de comportements complexes (> qqes dizaines d'états)
  - ↪ La complexité de réalisation des fonctions F et G croît comme  $2^{n+p}$ , si  $n$  est le nombre d'entrées et  $p$  le nombre de variables nécessaires au codage des états.
- **Différentes stratégies possibles pour gérer la complexité**
  - ↪ Décomposer l'unité de contrôle en plusieurs automates communicants
  - ↪ Utiliser un séquenceur
  - ↪ Utiliser un  $\mu$ -contrôleur du marché (si  $\exists$  assez rapide)

# Principe de la conception "à la main" d'un automate à états finis

- La méthode dite de *Huffman*

1. Etablir la spécification comportementale de l'automate sous la forme d'un **graphe d'états**. Optimiser le graphe si nécessaire (réduction du nombre d'états).

2. **Coder les états** du graphe.

3. Etablir la **table de transition** de l'automate.

entrée s	état présent	état futur	sortie s

4. Choisir le **support matériel** de réalisation.

5. Synthétiser les **fonctions F et G** en les optimisant par rapport au support choisi.

6. Etablir le **schéma logique**.

7. **Réaliser et tester** l'automate.

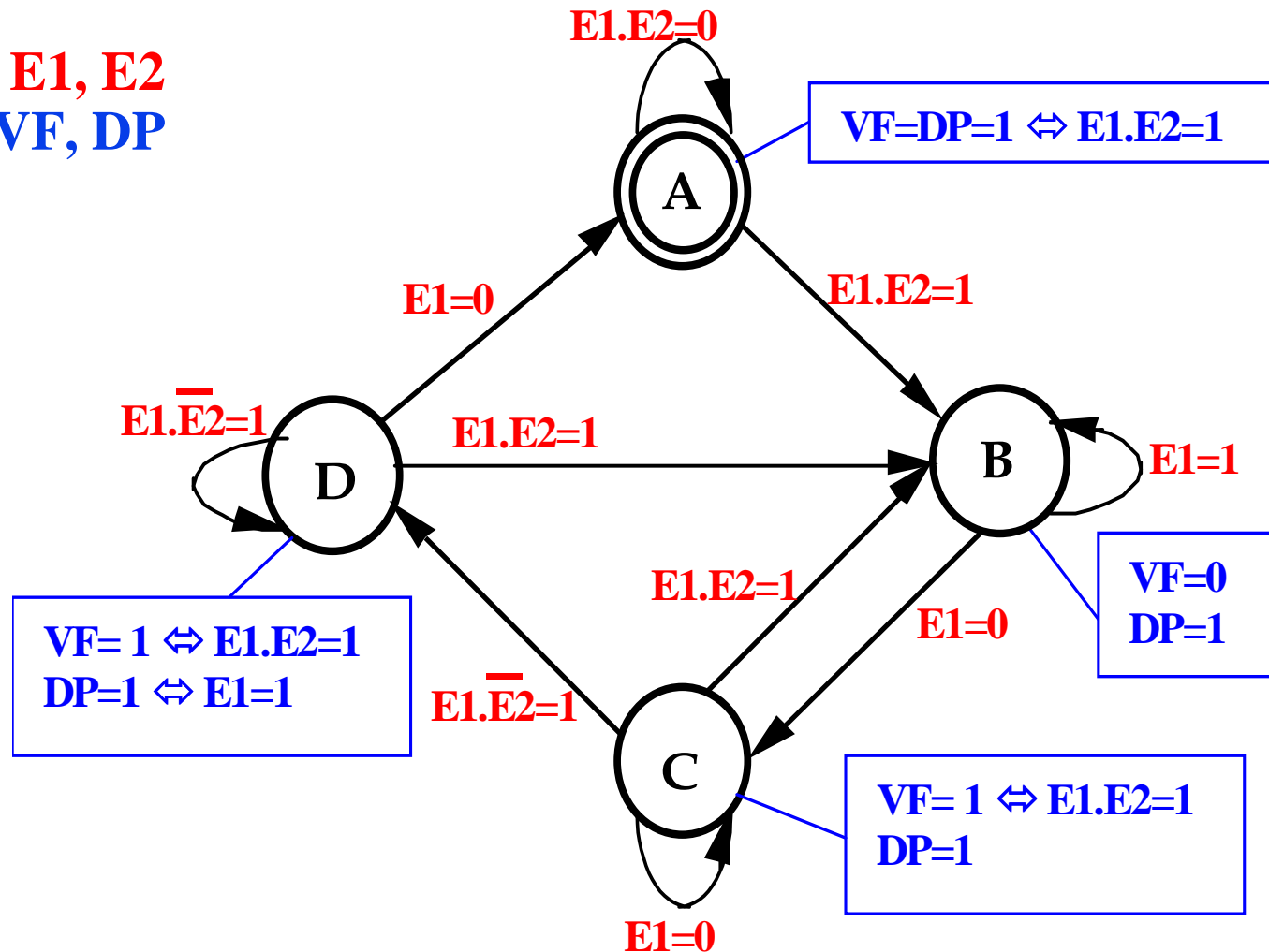
# Principe de la conception d'un automate à états finis à l'aide d'un environnement de CAO

1. Etablir la spécification comportementale de l'automate sous la forme d'un **graphe d'états**. Optimiser le graphe.
2. Transcrire le graphe d'états optimisé dans un langage accepté par l'environnement de CAO (**VHDL**).
3. Vérifier la spécification par une **simulation fonctionnelle** de l'automate.
4. Choisir le **support matériel** de réalisation. Fixer les **paramètres d'optimisation** (surface, vitesse,...). Faire la **synthèse logique** de l'automate. Le synthétiseur fournit automatiquement la "**netlist**" (description au niveau portes élémentaires) du circuit ou le **contenu de la PROM**.
5. **Simuler le fonctionnement** de l'automate après synthèse (simulation fonctionnelle + temporelle).
6. **Réaliser et tester l'automate**.

# Exemple de conception "à la main" d'un automate (I)

## 1. Graphe d'états de l'automate

Entrées : **E1, E2**  
Sorties : **VF, DP**



# Exemple de conception "à la main" d'un automate (II)

## 2. Codage des états

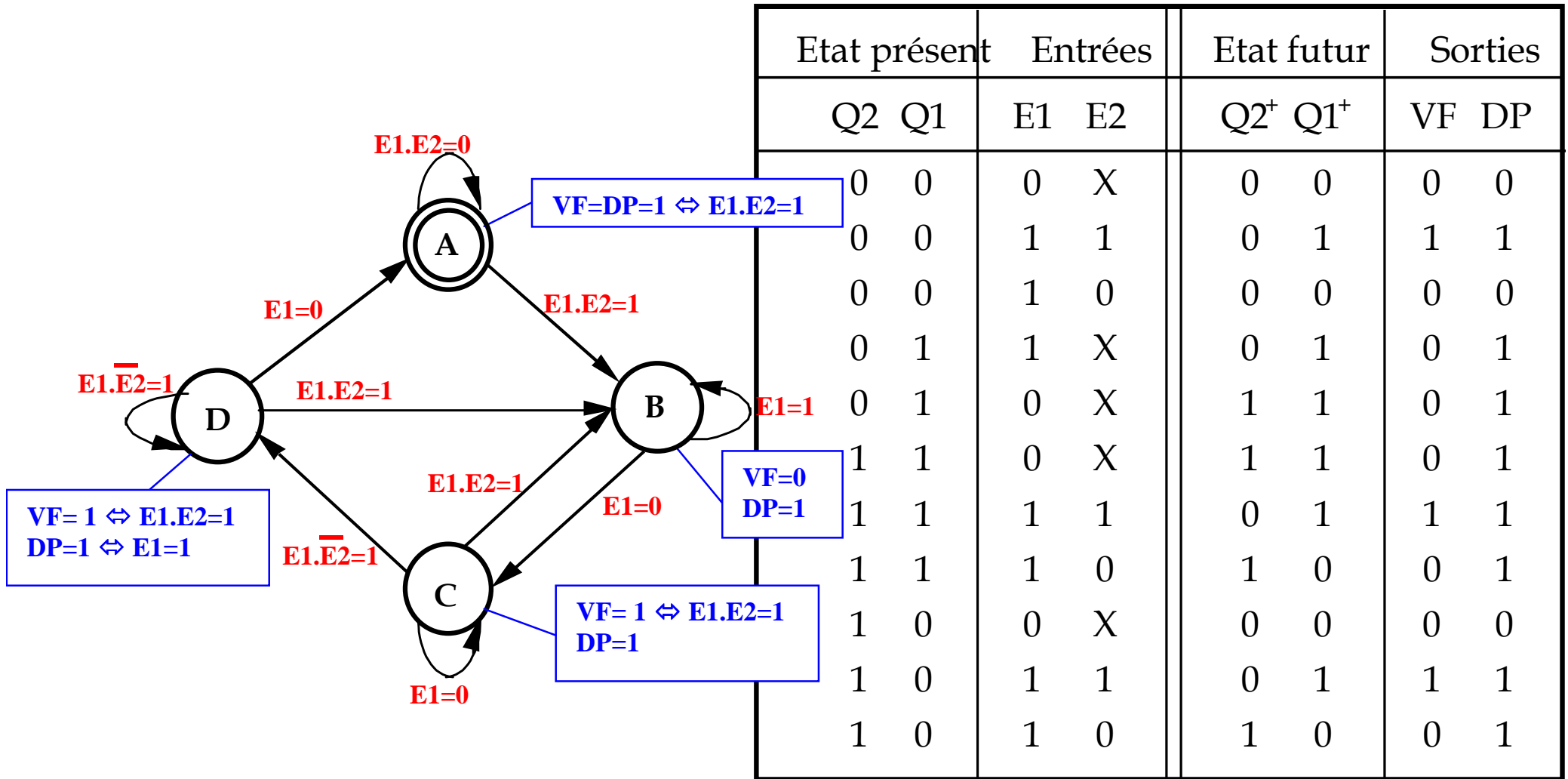
Etat	Q2	Q1
A	0	0
B	0	1
C	1	1
D	1	0

Q1, Q2 : variables **internes**  
ou variables **d'état**.

**NB:** le codage de l'état d'initialisation doit être cohérent avec les commandes d'initialisation des bascules

# Exemple de conception "à la main" d'un automate (III)

## 3. Table de transition





# Exemple de conception "à la main" d'un automate (IV)

## 4. Synthèse des fonctions F et G

$Q_1^+$

	E1		
	0	0	1
	0	1	1
Q2	1	1	1
	0	0	1
	E2		

Q1

$Q_2^+$

	E1		
	0	0	0
	1	1	0
Q2	1	1	0
	0	0	1
	E2		

Q1

$$Q_1^+ = E_1 E_2 + \overline{Q_2} Q_1 + \overline{E_1} Q_1$$

$$Q_2^+ = \overline{E_1} Q_1 + E_1 \overline{E_2} Q_2$$

$VF$

	E1		
	0	0	1
	0	0	0
Q2	0	0	1
	0	0	1
	E2		

Q1

$DP$

	E1		
	0	0	1
	1	1	1
Q2	1	1	1
	0	0	1
	E2		

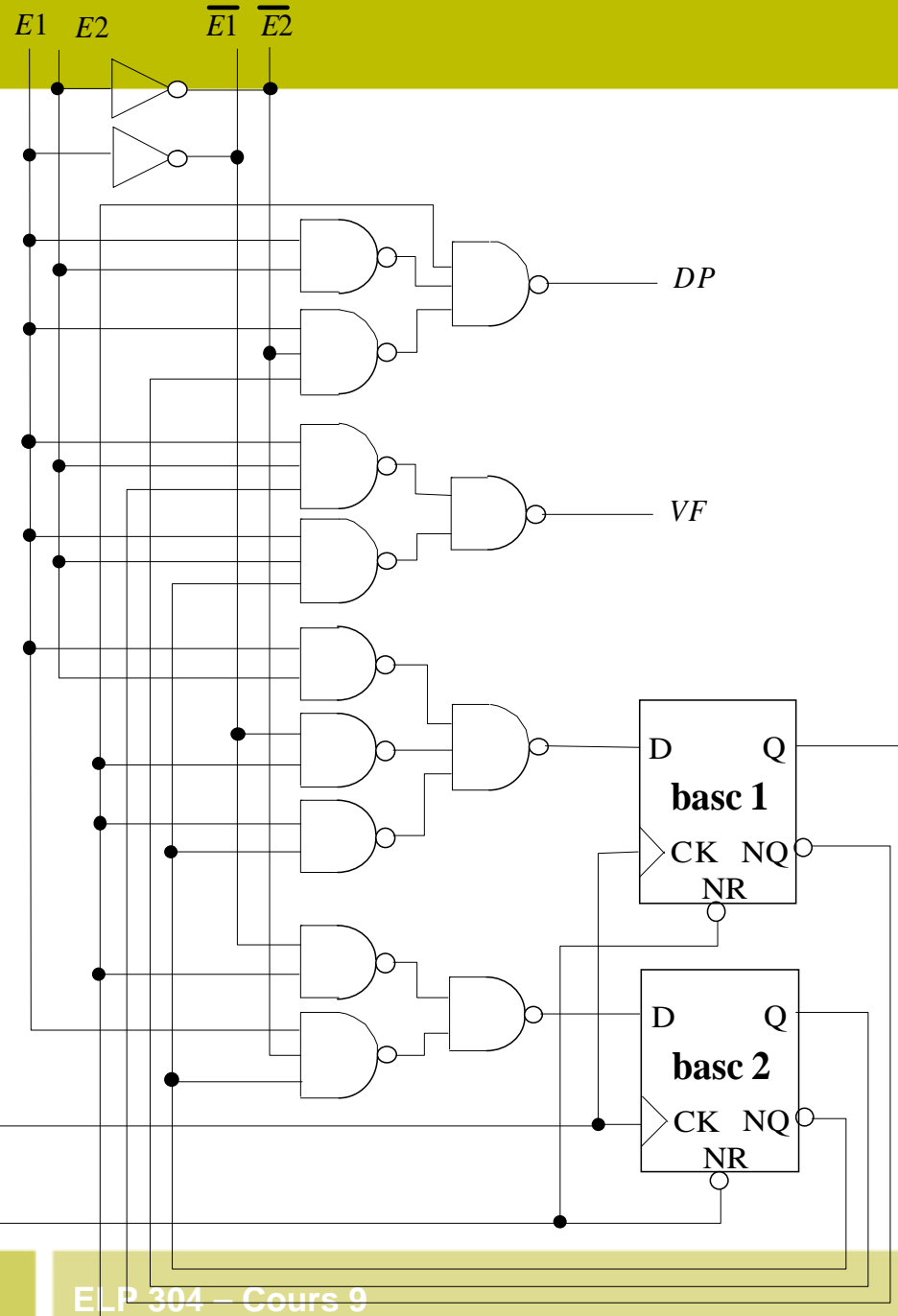
Q1

$$VF = E_1 E_2 \overline{Q_1} + E_1 E_2 Q_2$$

$$DP = Q_1 + E_1 E_2 + E_1 Q_2$$

# Exemple de conception "à la main" d'un automate (V)

## 5. Schéma logique



## 6. Réalisation et test de l'automate