Synthèse des systèmes séquentiels

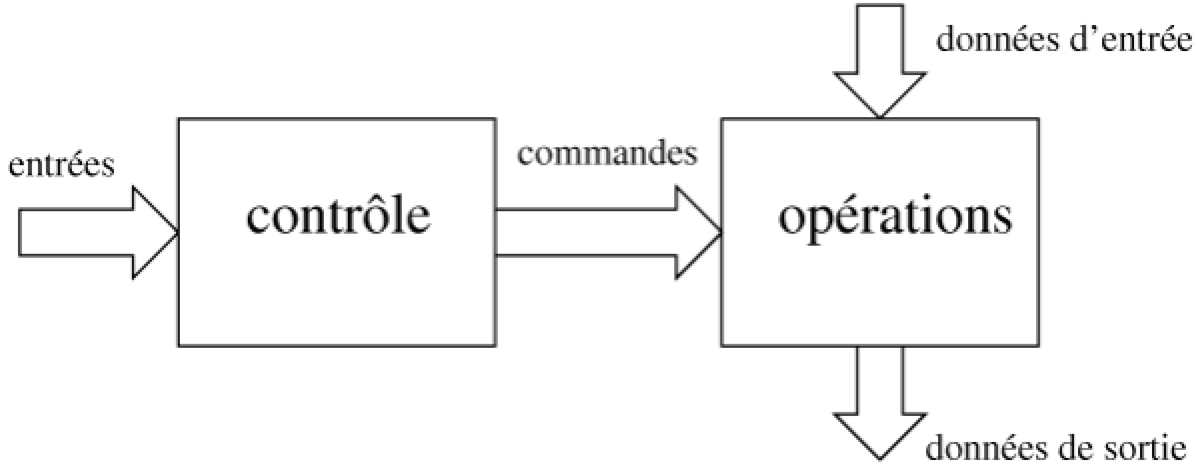
Redouane KARA
Professeur, UMMTO

Machine à états

- Les machines à états sont des circuits de logique séquentielle qui permettent de générer des signaux de commande.

- On a deux types de signaux:

- Signaux à traiter: les données
- Signaux qui pilotent le traitement: les commandes



Machines à états

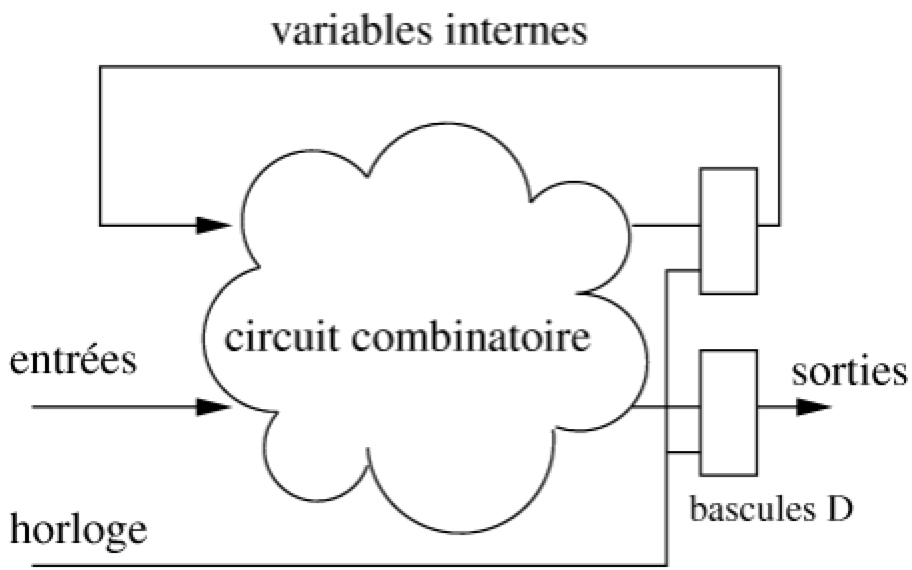
- En logique séquentielle synchrone, il existe deux signaux de commande importants:
- L'horloge: pour le déroulement des séquences
- Le Reset: pour l'initialisation du système.

- La machine à état représente la partie contrôle.
- Compliqués : les µp
- Simples: les contrôleurs d'ascenseurs, machines à café
  - dans ce type: on a pas de données car les commandes servent à piloter des actionneurs, valves, moteurs …

Machine à états

- Les états de la machine à états représente toutes les valeurs que peuvent prendre les variables internes du circuit logique.

- Le schéma générique d'une machine à état est le suivant:



Exemple

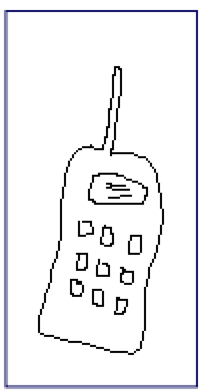
- Pour la machine à café les états peuvent être:

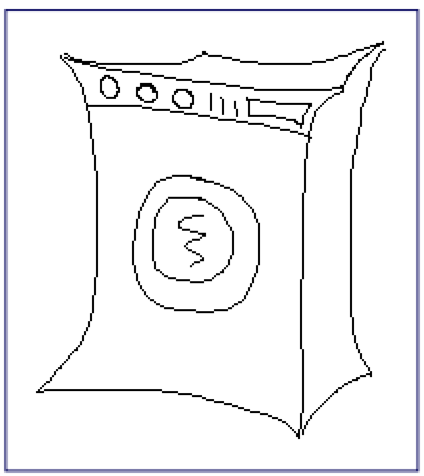
1. Attente de pièce,
2. Descendre le gobelet,
3. Verser le poudre à café,
4. Verser l'eau chaude,
5. Indiquer que c'est prêt.

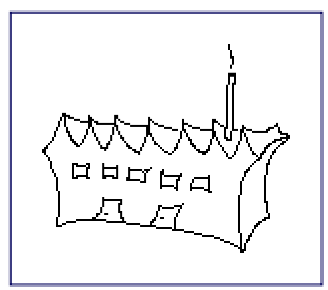
- Cette machine peut se compliquer en prenant en compte d'autre spécifications comme: le choix de la boisson, le dosage du sucre …

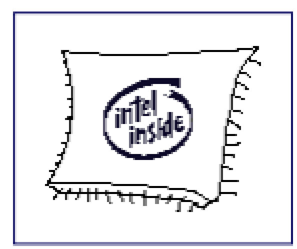
Où trouve t'on les machines à états ?

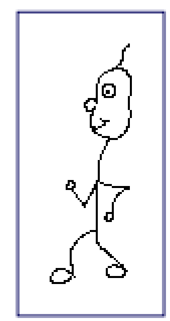
- Conduite d'une centrale nucléaire,
- Automatisation d'une usine de production.
- Microprocesseurs spécialisées pour le graphisme ou les télécommunications.









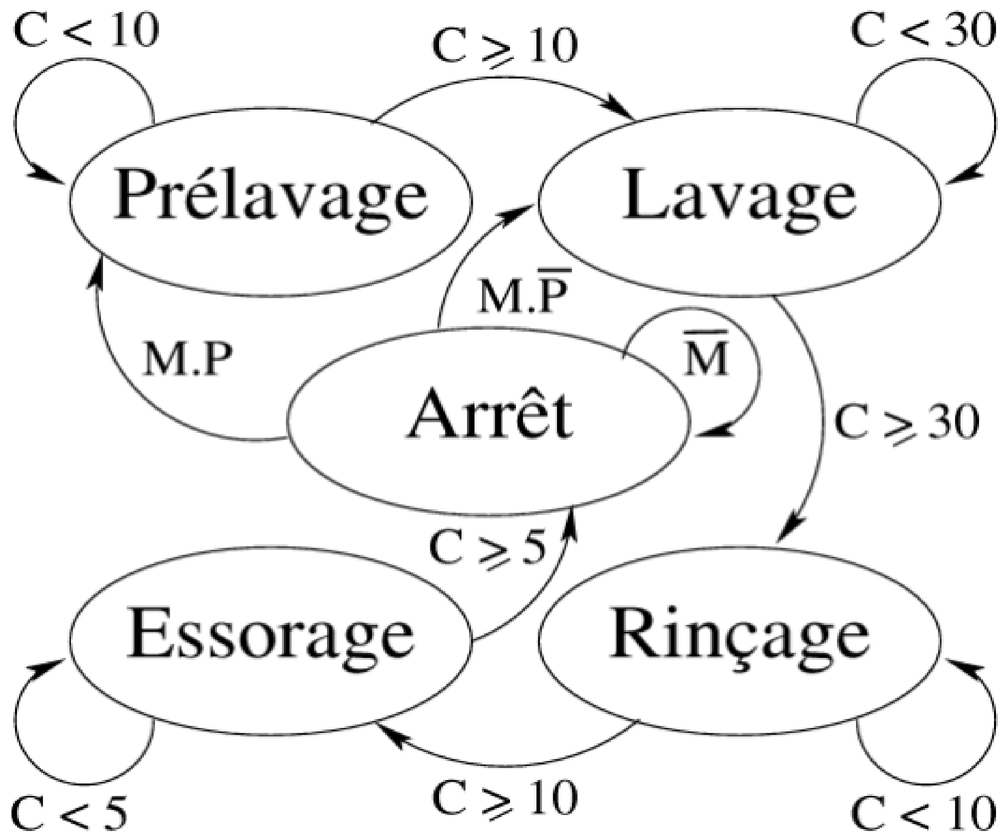


Les graphes d’états

- Le graphe d’états, représente graphiquement les états d’une machine à états.
- Chaque état est dessiné sous la forme d’une bulle contenant son nom.
- Les graphe dans son ensemble est complété par des arcs (où flèches) représentant les transitions entre états.
- A tout instant, la machine est dans l’un des états représentés, qu’on appelle état courant.
- Les transitions entre les états sont rythmées par l’horloge.
- On comprend immédiatement que cet outil ne sera pas d’un grand secours lorsque le nombre d’états de la machine dépassera quelques dizaines.

Exemple: Machine à laver

- Prenons l'exemple d'une machine à laver où on considère 5 états comme illustré dans la figure suivante :



Exemple: Machine à laver

- Variables d'entées:

- M : variable booléenne qui traduit la position du bouton Marche/Arrêt du lave-linge.
- P : variable booléenne qui indique si le programme de lavage sélectionné par l'utilisateur comporte ou non une phase de prélavage.
- C : valeur en minutes d'un chronomètre qui est remis à zéro automatiquement au début de chaque étape de lavage.

- Les durées des étapes de lavage sont fixées par le constructeur :

- Prélavage : 10 minutes
- Lavage : 30 minutes
- Rinçage : 10 minutes
- Essorage : 5 minutes

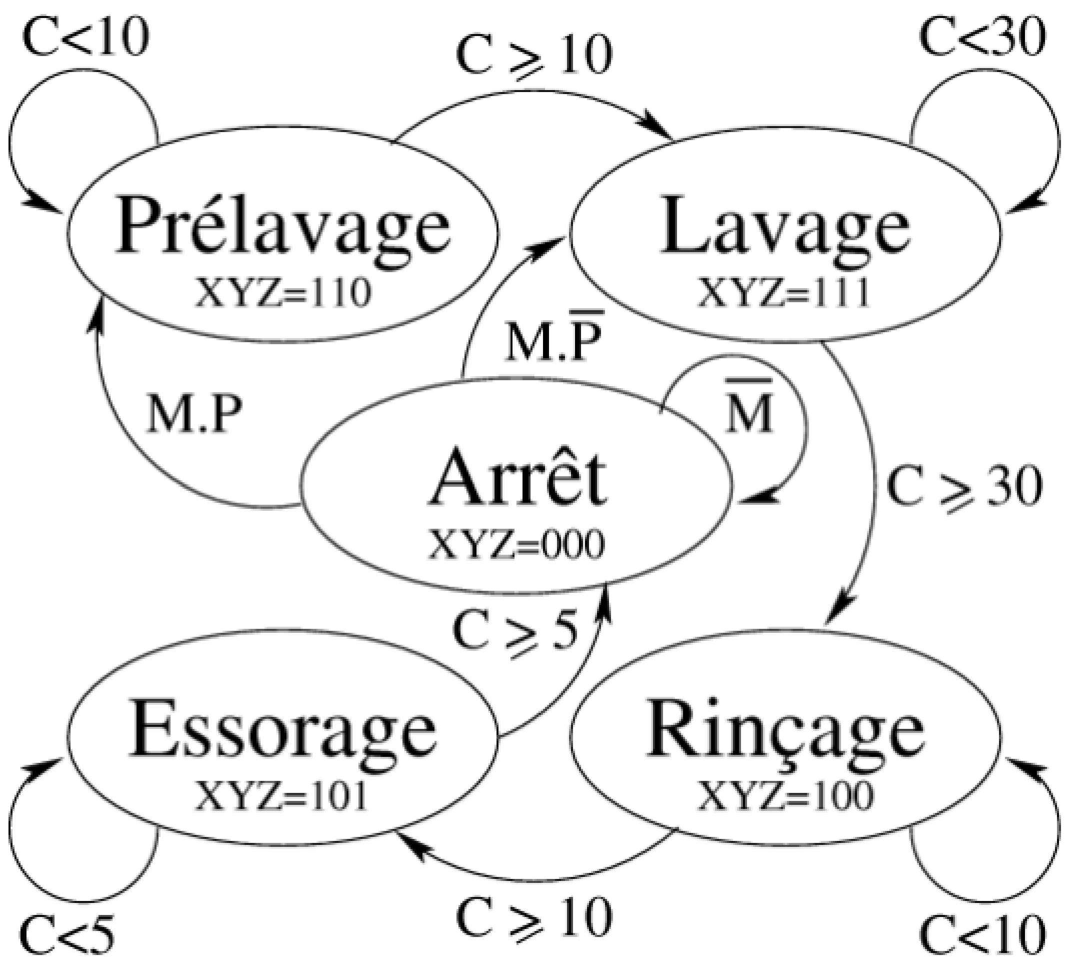
Exemple: Machine à laver

- Les variables de sortie correspondent aux signaux de commande.
- Il existe deux sortes de machines à états :

- Machine de Moore : les sorties ne dépendent que de l'état courant
- Machine de Mealy: les sorties dépendent de l'état courant et des entrées

- Dans cette exemple, nous allons traiter le cas d'une machine de Moore.
- Le programmateur de notre lave-linge est donc une machine de Moore dont les sorties ne dépendent que de l'état courant. Nous supposerons que ses sorties sont trois signaux booléens, X, Y et Z destinés à piloter les différents moteurs du lave-linge.
- Nous pouvons encore compléter le graphe d'états afin d'y faire figurer cette information.

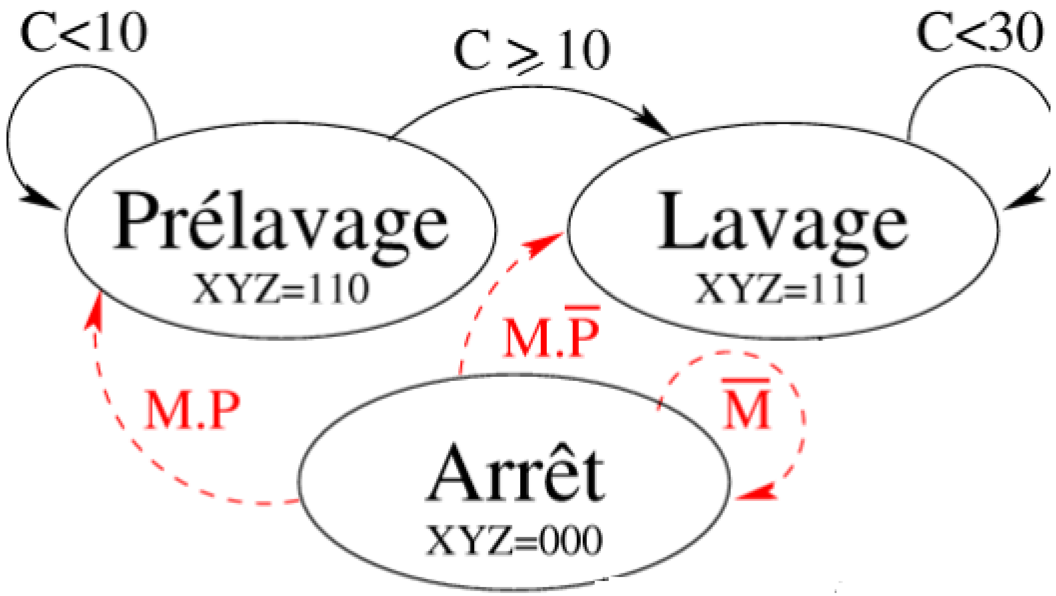
Graphe d'état du programmateur



Vérification du graphe

- Le graphe doit vérifier deux propriétés principales:

1. Complet ou non ambigu ⇒ comportement défini ⇒ état suivant connu. soient C1, C2, ..., Ci, ..., CN, N conditions en partant d'un état donné, il faut : $\sum_{i=1}^{i=N} C_i = 1$



Vérification du graphe

- Les transitions partant de l'état Arrêt sont au nombre de 3: $\overline{M}$, $M.P$, $M.\overline{P}$
- Le OU logique ($+$) de ces trois conditions vérifie donc :

$$\overline{M} + M.P + M.\overline{P} = \overline{M} + M.(P + \overline{P}) = \overline{M} + M = 1$$

- L'état Arrêt respecte donc la première règle.
- Vous pouvez vérifier que c'est également le cas pour les quatre autres états.

Deuxième condition

- Graphe non contradictoire $\Rightarrow$ à chaque front montant d'horloge une seule transition est possible. Elle garantit que deux actions incompatibles ne sont pas simultanément possibles.

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} C_i C_j = 0$$

- En reprenant l'état Arrêt du programmateur de lave-linge comme exemple, cette condition s'écrit :

$$\overline{\mathrm{M}}.\,\mathrm{M}.\,\overline{\mathrm{P}} + \overline{\mathrm{M}}.\mathrm{M}.\mathrm{P} + \mathrm{M}.\,\overline{\mathrm{P}}.\mathrm{M}.\mathrm{P} = 0.$$

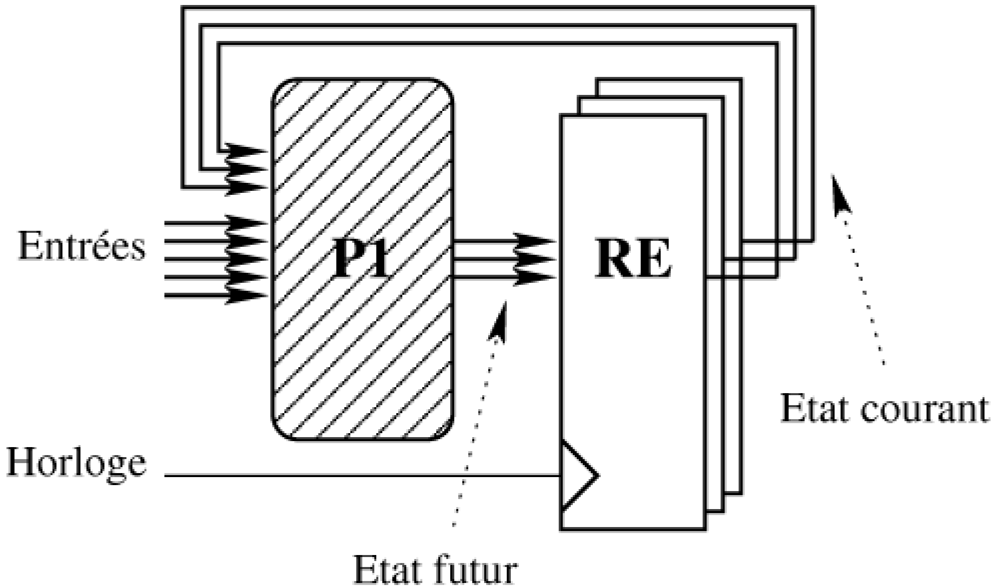
- L'état Arrêt respecte donc également la deuxième règle.

- Vous pouvez vérifier que c'est également le cas pour les quatre autres états.

Calcul de l'état futur

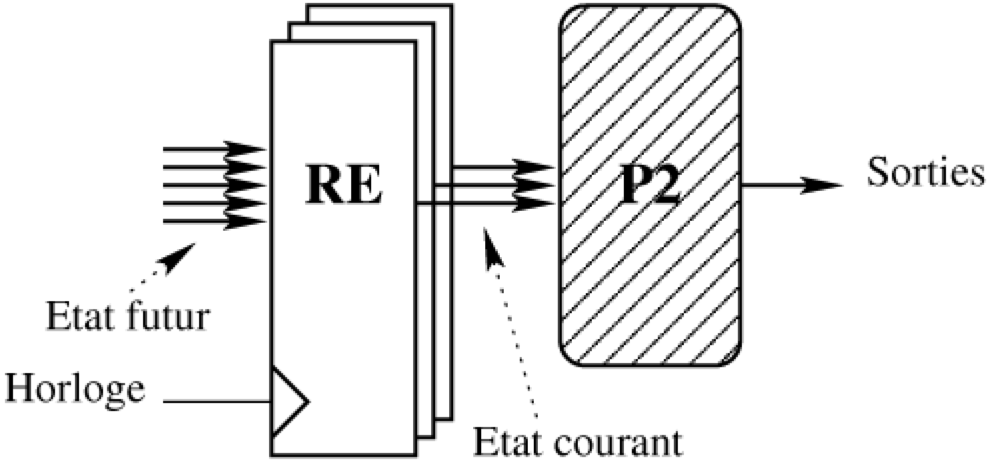
- Le calcul de l'état futur

- A chaque front montant de l'horloge, l'état courant est modifié
- Entre deux fronts montants (une période d'horloge) l'état reste stable
- Un circuit combinatoire (P1) calcule le prochain état.



Calcul de la sortie

- Un circuit combinatoire de sorties (P2) permet de calculer la sortie.
- Dans une machine de Moore, ses entrées sont l'état courant.
- Dès que l'état courant change, après un front montant, ce circuit calcule les sorties du nouvel état.



- Les circuits P1 et P2 il doivent disposer d'assez de temps pour faire le calcul durant une période d'horloge.

Vérification du graphe

- Les transitions partant de l'état Arrêt sont au nombre de 3: $\overline{M}$, $M.P$, $M.\overline{P}$
- Le OU logique ($+$) de ces trois conditions vérifie donc :

$$\overline{M} + M.P + M.\overline{P} = \overline{M} + M.(P + \overline{P}) = \overline{M} + M = 1$$

- L'état Arrêt respecte donc la première règle.
- Vous pouvez vérifier que c'est également le cas pour les quatre autres états.

Deuxième condition

- Graphe non contradictoire $\Rightarrow$ à chaque front montant d'horloge une seule transition est possible. Elle garantit que deux actions incompatibles ne sont pas simultanément possibles.

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} C_i C_j = 0$$

- En reprenant l'état Arrêt du programmateur de lave-linge comme exemple, cette condition s'écrit :

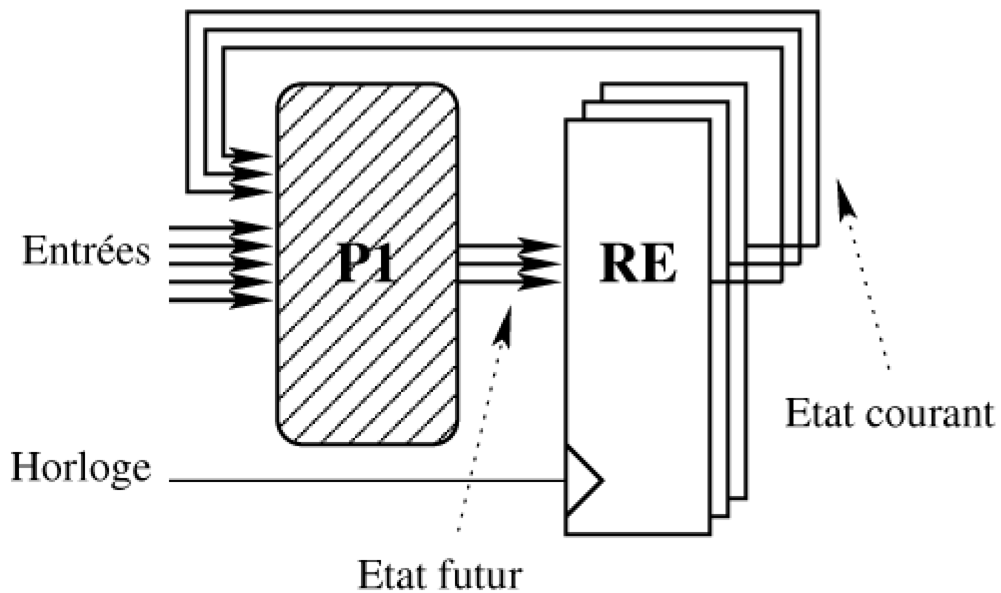
$$\overline{\mathrm{M}}.\,\mathrm{M}.\,\overline{\mathrm{P}} + \overline{\mathrm{M}}.\mathrm{M}.\mathrm{P} + \mathrm{M}.\,\overline{\mathrm{P}}.\mathrm{M}.\mathrm{P} = 0.$$

- L'état Arrêt respecte donc également la deuxième règle.
- Vous pouvez vérifier que c'est également le cas pour les quatre autres états.

Calcul de l’état futur

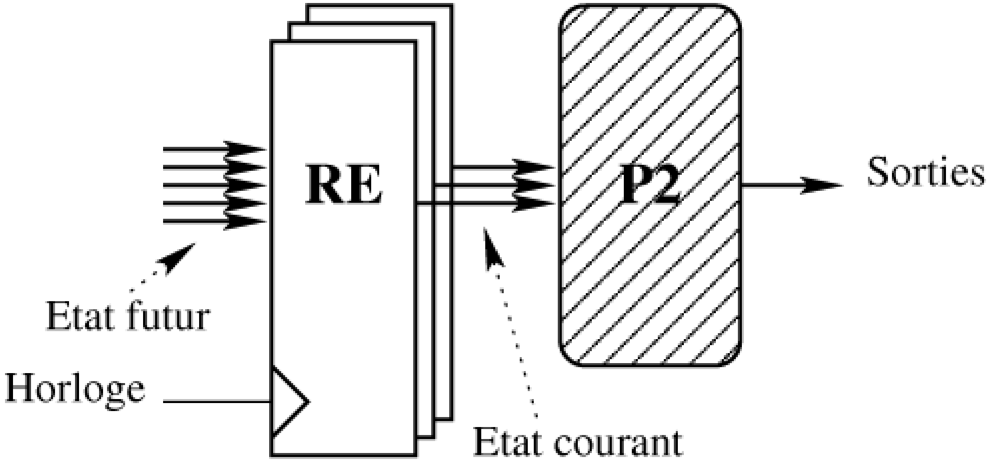
- Le calcul de l’état futur

- A chaque front montant de l’horloge, l’état courant est modifié
- Entre deux fronts montants (une période d’horloge) l’état reste stable
- Un circuit combinatoire (P1) calcule le prochain état.



Calcul de la sortie

- Un circuit combinatoire de sorties (P2) permet de calculer la sortie.
- Dans une machine de Moore, ses entrées sont l’état courant.
- Dès que l’état courant change, après un front montant, ce circuit calcule les sorties du nouvel état.



- Les circuits P1 et P2 il doivent disposer d’assez de temps pour faire le calcul durant une période d’horloge.

Vérification du graphe

- Les transitions partant de l'état Arrêt sont au nombre de 3: $\overline{\mathrm{M}}$, $\mathrm{M.P}$, $\mathrm{M.}\,\overline{\mathrm{P}}$

- Le OU logique ($+$) de ces trois conditions vérifie donc :

$$\overline{\mathrm{M}} + \mathrm{M.P} + \mathrm{M.}\,\overline{\mathrm{P}} = \overline{\mathrm{M}} + \mathrm{M.}(\mathrm{P} + \overline{\mathrm{P}}) = \overline{\mathrm{M}} + \mathrm{M} = 1$$

- L'état Arrêt respecte donc la première règle.

- Vous pouvez vérifier que c'est également le cas pour les quatre autres états.

Le codage des états

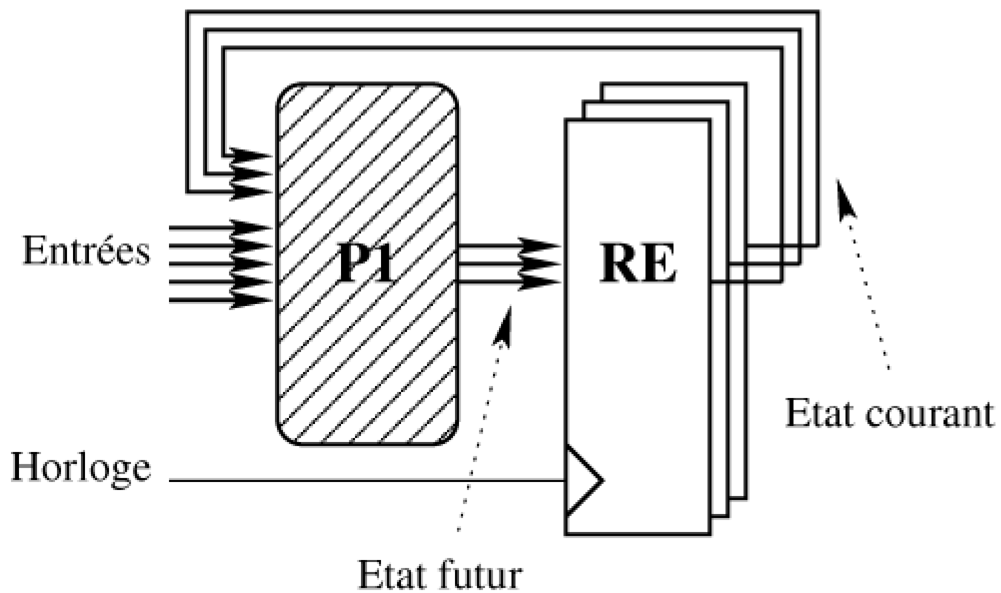
- Dans l'électronique numérique on ne manipule que des 0 et des 1.
- Pour chaque état d'une machine il va falloir trouver un nom unique exprimé avec des 0 et des 1, un mot binaire.
- On appelle codage la représentation en mots binaires des noms des états.
- un exemple de codage à trois, quatre, cinq et six bits pour notre exemple

Etat	3 bits	4 bits	5 bits	6 bits
Arrêt	100	0001	11110	110001
Prélavage	000	0110	10100	101010
lavage	001	1111	01100	110111
Rinçage	010	0000	01101	010110
Essorage	111	1011	01110	010111

Calcul de l’état futur

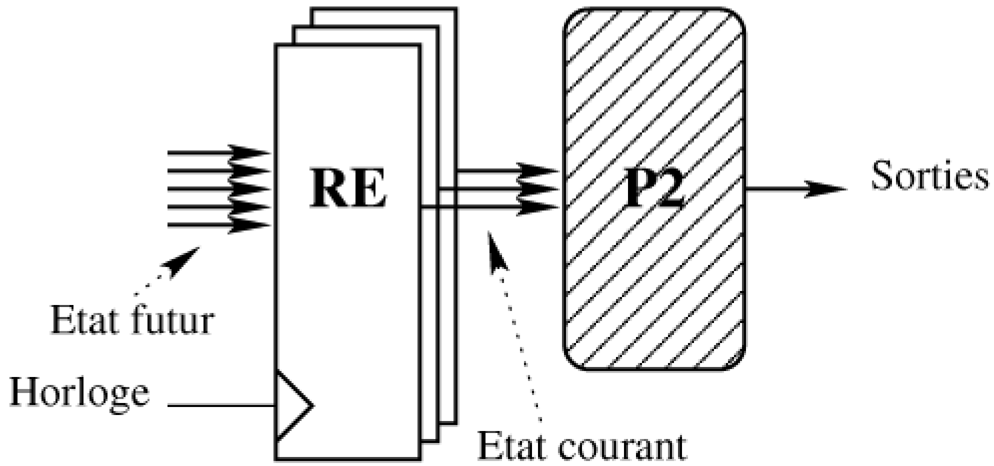
- Le calcul de l’état futur

- A chaque front montant de l’horloge, l’état courant est modifié
- Entre deux fronts montants (une période de l’horloge) l’état reste stable
- Un circuit combinatoire (P1) calcule le prochain état.



Calcul de la sortie

- Un circuit combinatoire de sorties P2 permet de calculer la sortie.
- Dans une machine de Moore, les entrées de P2 sont l'état courant.
- Dès que l'état courant change, après un front montant, ce circuit calcule les sorties du nouvel état.



- Les circuits P1 et P2 doivent disposer d'assez de temps pour faire le calcul durant une période d'horloge.

Codage des états

- Le circuit combinatoire P1 est influencée par le codage des états.
- On pourrait conclure qu'il faut coder les états avec le moins de bits possible

$\Rightarrow$ circuit combinatoire soit la plus petite possible.

- Cette conclusion n'est pas vrai en général.

- Exemple:

  $f(x_1, x_2, x_3, x_4) = x_1$ est plus simple que $g(x_1, x_2) = x_1 \oplus x_2$

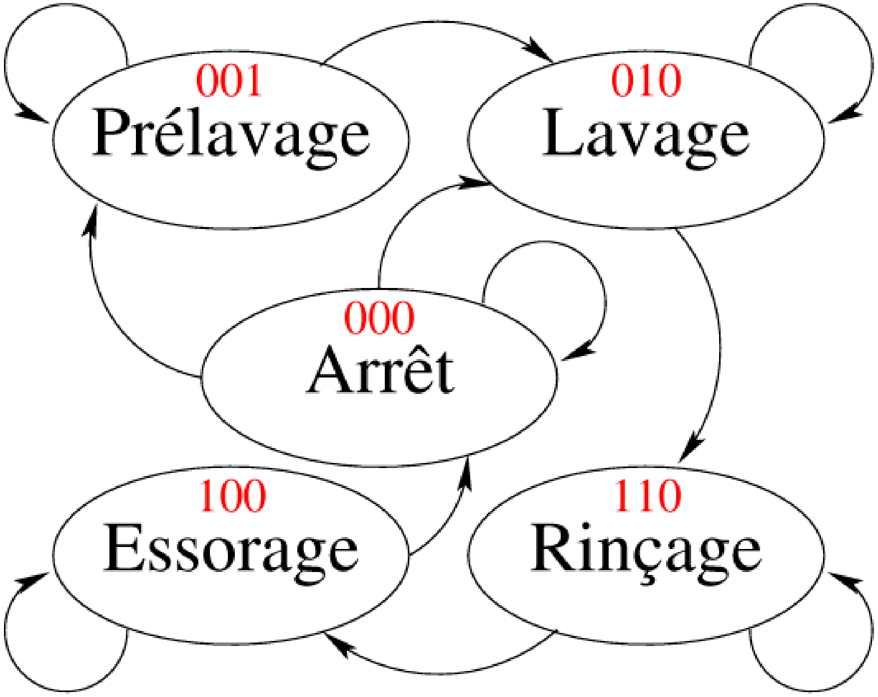
- Quelle méthode permet d'avoir le meilleur codage ?

- Meilleur codage $\Rightarrow$ définir un critère (taille, consommation, vitesse, simplicité de conception)
- Parmi les méthodes de codage, nous allons citer que 3 méthodes.

Le codage adjacent

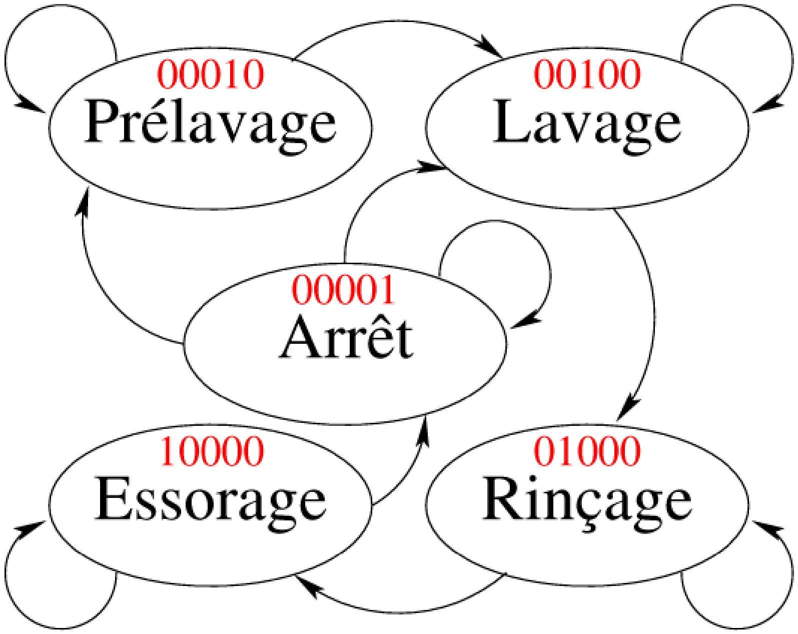
- **Le codage adjacent:**

- Il utilise un nombre de bit minimum et se caractérise par le fait que le passage d'un état à un autre ne modifie qu'un seul bit.
- C'est un codage qui donne de bon résultats en taille et en vitesse pour la partie combinatoire qui calcule l'état futur.



Codage one-hot

- **Codage « one hot »:**
- Il utilise un nombre de bit égal au nombre d'état $\Rightarrow$ Taille du registre importante.
- Chaque états est représenté par un mot binaire dont tous les bits sauf 1 valent 0.



- **Ce codage donne les machines les plus simples à concevoir**

Codage aléatoire

- **Codage aléatoire:**

- Coder sur un nombre de bit minimum sans autre préoccupation que d'éviter que deux états aient le même codage.
- Résultats imprévisibles.

- **Remarque :**

- Il existe des solutions logicielles qui peuvent aider le concepteur à trouver le « bon » codage.

Exemple Machine à laver

- L’interface de la machine avec le monde extérieur est spécifié dans le tableau:

Nom	Mode	Description
H	Entrée	Horloge
R	Entrée	Reset actif à 0 , initialise à l'état Arrêt
M	Entrée	Position du bouton Marche/Arrêt
P	Entrée	Existence d'une phase de prélavage
C5	Entrée	Chronomètre supérieur ou égal à 5 minutes
C10	Entrée	Chronomètre supérieur ou égal à 10 minutes
C30	Entrée	Chronomètre supérieur ou égal à 30 minutes
X	Sortie	Vaut 0 dans l'état Arrêt, 1 dans les autres
Y	Sortie	Vaut 1 dans les états Prélavage et Lavage, 0 dans les autres
Z	Sortie	Vaut 1 dans les états Lavage et Essorage, 0 dans les autres

Codage

- Considérons le codage suivant:

Etat	Codage
Arrêt	000
Prélavage	001
Lavage	010
Rinçage	110
Essorage	100

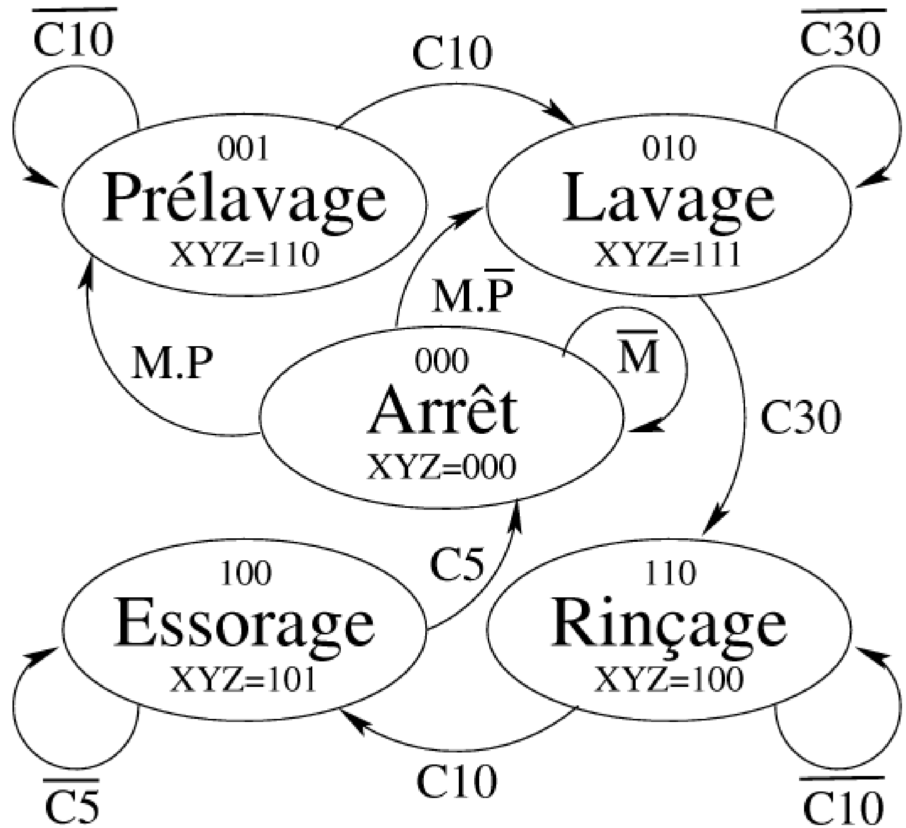


Table d'évolution

Etat courant			Entrées					Etat futur		
EC2	EC1	EC0	M	P	C5	C10	C30	EF2	EF1	EF0
0	0	0	0	X	X	X	X	0	0	0
0	0	0	1	1	X	X	X	0	0	1
0	0	0	1	0	X	X	X	0	1	0
0	0	1	X	X	X	0	X	0	0	1
0	0	1	X	X	X	1	X	0	1	0
0	1	0	X	X	X	X	0	0	1	0
0	1	0	X	X	X	X	1	1	1	0
1	1	0	X	X	X	0	X	1	1	0
1	1	0	X	X	X	1	X	1	0	0
1	0	0	X	X	0	X	X	1	0	0
1	0	0	X	X	1	X	X	0	0	0

Synthèse

- A partir de cette table, une méthode systématique permettant le calcul des circuits combinatoires P1 et P2 existe.
- C'est ce que nous allons voir dans le cours qui va suivre !!!