

ELEC-H-305

Circuits logiques et numériques
2011-2012

Cours 10

Dragomir Milojevic

dmilojev@ulb.ac.be

Aujourd'hui

- ❖ Synthèse de la **fonction de sortie** dans le cas d'une **Machine de Moore**
- ❖ Synthèse des **Machines de Meally**
(Fusionnement des états ayant les sorties différentes)
- ❖ Synthèse de la **fonction de sortie** dans la cas de la **Machine de Meally**

*A la fin de la séance d'aujourd'hui on disposera de tous les outils permettant la synthèse des systèmes
séquentiels asynchrones*

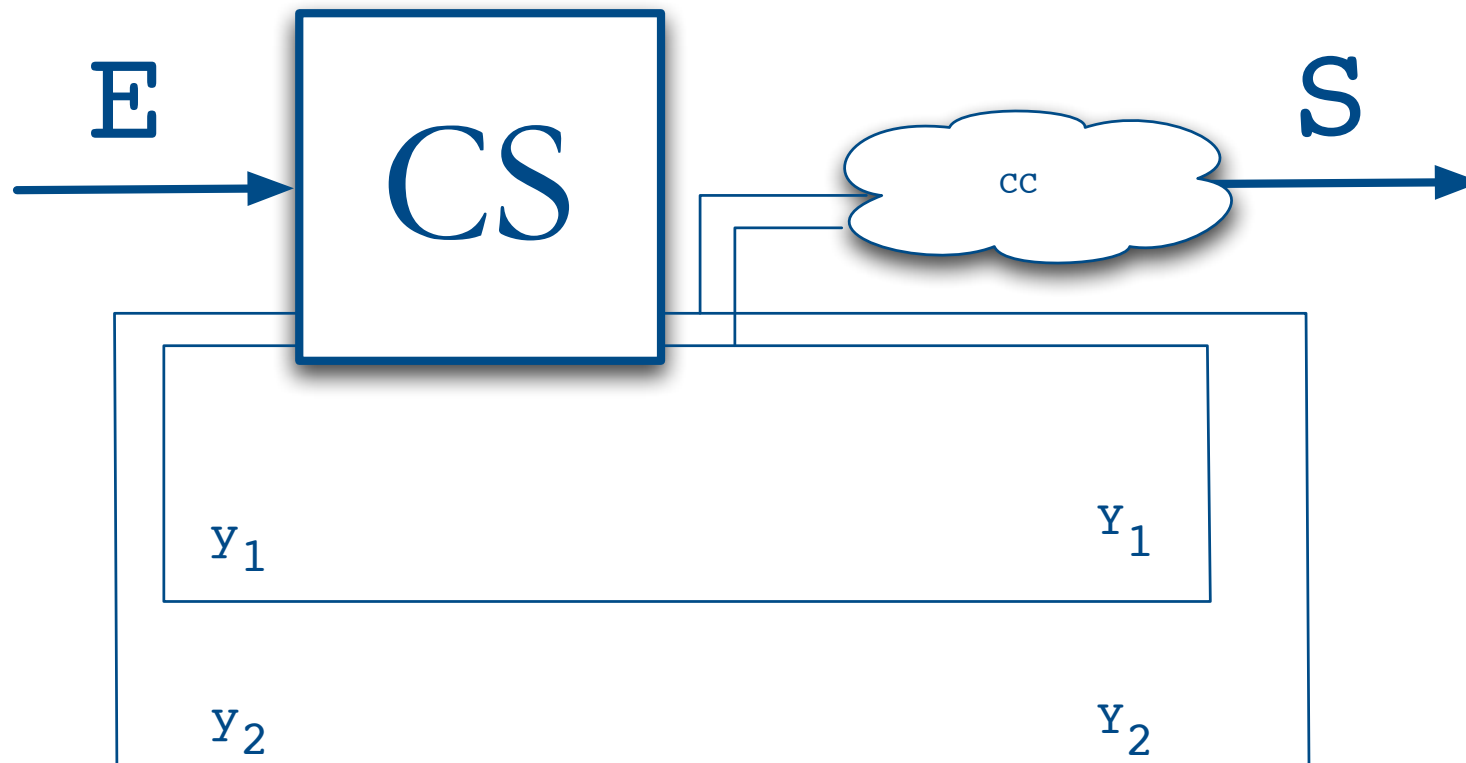
Synthèse de la machine de Moore

- ❖ Lors de fusionnement (la construction de la **Table d'équivalence des états**) on examine uniquement les états ayant les mêmes sorties
- ❖ Table d'état obtenue après fusionnement (**Table d'état réduite**) : pour chaque état stable (pour chaque ligne de la Table d'état) nous avons une valeur de sortie
- ❖ En principe la sortie est donc uniquement fonction de l'état dans lequel la machine se trouve (**Machine de Moore**)

Synthèse de la machine de Moore

Nous avons donc: $S = f(Y_i)$,

La sortie est produite par un circuit combinatoire, dont les entrées sont des variables d'état de système.

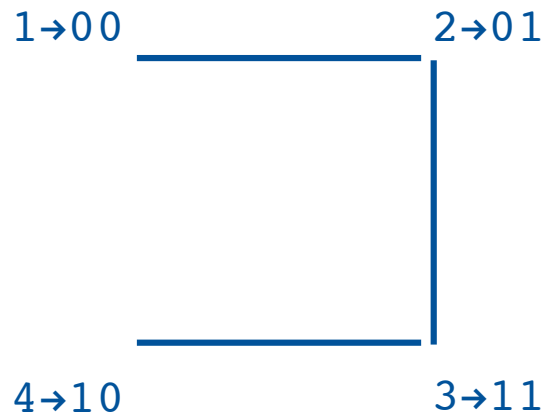


Synthèse de la machine de Moore

Pour la table d'état réduite suivante, le codage $1 \rightarrow 00$, $2 \rightarrow 01$, $3 \rightarrow 11$, $4 \rightarrow 10$ ne présente pas de problème de courses critiques (voir **Graphe de codage**).

ab

	00	01	11	10	z
1	1	2	2	–	0
2	1	2	3	–	1
3	3	2	3	4	0
4	–	4	3	4	1



ab

	00	01	11	10
00	00	01	1	–
01	00	01	11	–
11	11	01	11	10
10	–	10	11	10

Synthèse de la machine de Moore

	ab			
	00	01	11	10
00	00	01	01	–
01	00	01	11	–
11	11	01	11	10
10	–	10	11	10

	ab			
Y ₁	00	01	11	10
00	0	0	0	–
01	0	0	1	–
11	1	0	1	1
10	–	1	1	1

Fonctions de
rétroaction

	ab			
Y ₂	00	01	11	10
00	0	1	1	–
01	0	1	1	–
11	1	1	1	0
10	–	0	1	0

$$Y_1 = y_1 y_2 + y_1 b' + a y_2$$

$$Y_2 = ab + y_1' b + y_2 b + y_1 a' b'$$

Synthèse de la machine de Moore

Synthèse de la sortie uniquement en fonction des variables d'état : on regarde les états stables pour lesquels la sortie vaut 1 et on dérive la fonction de sortie directement (on fait un décodeur d'état).

Ce n'est pas une manière correcte de faire les choses (**transitions!**)

	ab				
	00	01	11	10	z
00	00	01	01	-	0
01	00	01	11	-	1
11	11	01	11	10	0
10	-	10	11	10	1

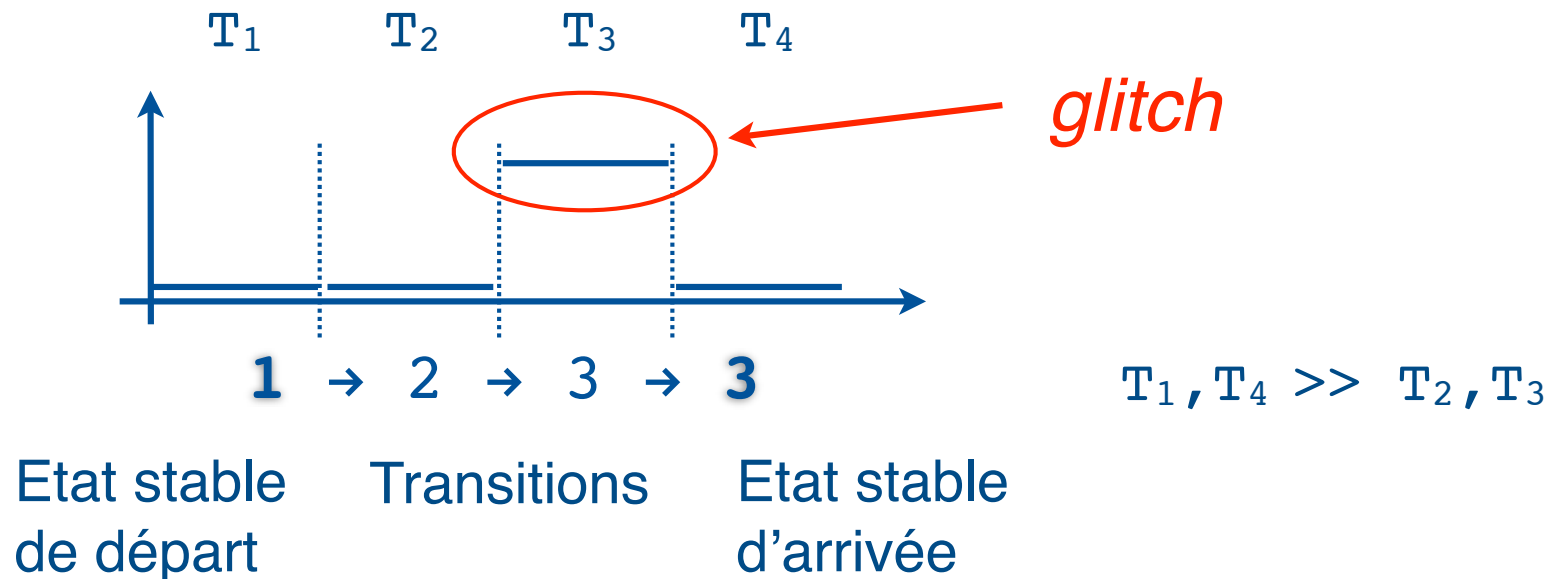
Terme 01 Terme 10

$$Z(y_1, y_2) = \cancel{y_1' y_2} + y_1 y_2'$$

Synthèse de la fonction de sortie

Que se passe-t-il avec la sortie lors des transitions ?

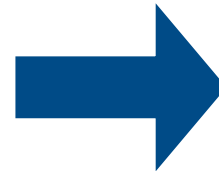
Lors des transitions il est possible que le système combinatoire reçoit en entrée des valeurs transitoires des variables d'états. Pendant ces transitoires la sortie sera calculée en fonction de ces transitoires...



Synthèse de la fonction de sortie

On part d'une **Table d'état** (réduite, codé ou pas) et on souhaite construire une table de sortie (autant de tables que de variables de sortie).

	ab				
	00	01	11	10	z
00	00	01	01	–	0
01	00	01	11	–	1
11	11	01	11	10	0
10	–	10	11	10	1



	ab				
z	00	01	11	10	
1	?	?	?	?	
2	?	?	?	?	
3	?	?	?	?	
4	?	?	?	?	

On doit fixer la sortie pour toutes les transitions de la table d'état !

Synthèse de la fonction de sortie

- ❖ On souhaite contrôler la valeur de la sortie de façon à éviter les **multiples variations** (en fait les *glitches*) pendant les transitions
- ❖ ... mais aussi on souhaite avoir l'expression pour la sortie la plus simple possible (économie des portes)
- ❖ On dresse une K-Map pour TOUTES les transitions de façon à ce qu'il n'y ait plus de *glitches* et que l'expression finale soit la plus simple possible

Synthèse de la fonction de sortie

Comme la sortie est fixée pour les états stables, on commence par remplir la table des sorties pour les états stables uniquement. Si l'état futur est un *don't care*, alors la sortie a pour valeur *don't care*.

Ainsi pour l'exemple précédant :

	ab				
	00	01	11	10	z
1	1	2	2	–	0
2	1	2	3	–	1
3	3	2	3	4	0
4	–	4	3	4	1

	ab				
z	00	01	11	10	
00	0			–	
01		1		–	
11	0		0		
10	–	1		1	

Il reste à fixer la sortie pour les transitions...

Synthèse de la fonction de sortie — transitions

Dans une **Table d'état réduite** deux cas sont possibles :

- I. Le passage d'un état stable à l'autre avec une seule transition
- II. Le passage d'un état stable à l'autre avec plusieurs transitions

Origine de multiples transitions:

- Soit à cause de cahier de charges,
- Soit à cause de la résolution de problème des courses (p.e. ajout d'une 3ème variable d'état lorsque Méthode 1 et Méthode 2 ne peuvent pas être appliquées, implique au moins 3 transitions).

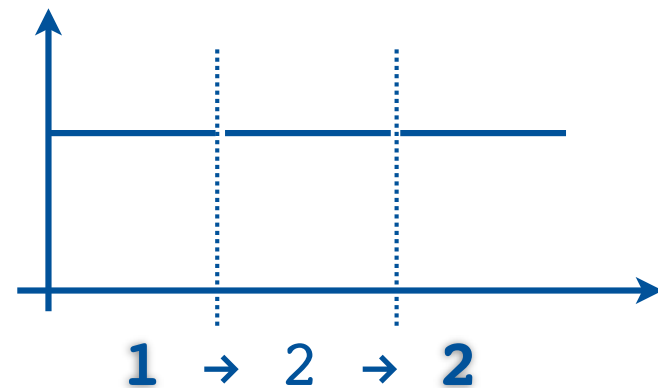
Synthèse de la fonction de sortie — une transition

Nous avons deux possibilités pour les valeurs de sortie des états de départ et d'arrivée :

- a) **La valeur de la sortie de l'état de départ est d'arrivée est la même** → La transition doit garder la même valeur (0 ou 1).



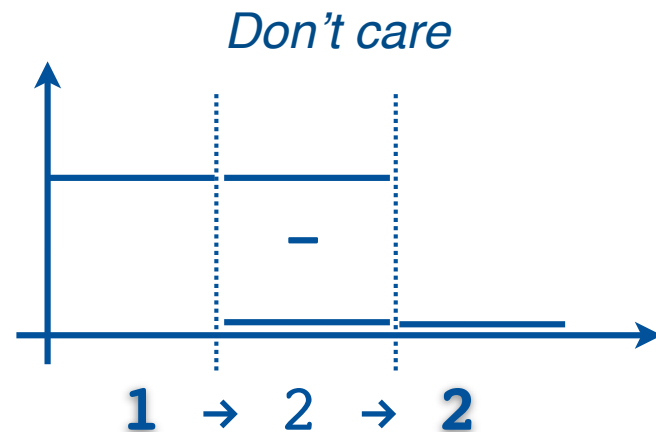
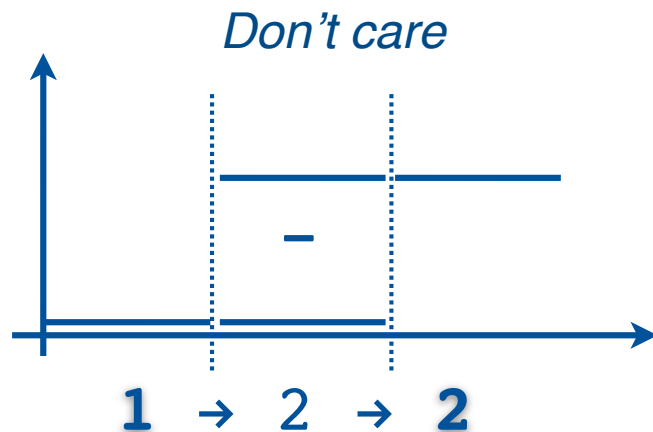
Etat stable de départ Transition Etat stable d'arrivée



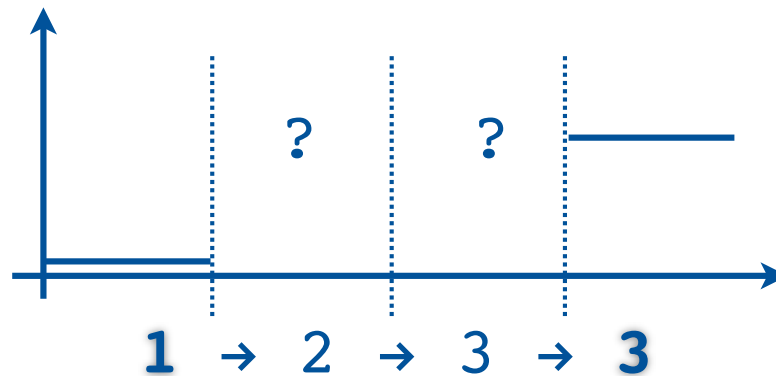
Synthèse de la fonction de sortie — une transition

b) La valeur de la sortie de l'état de départ est d'arrivé est différente

Lors de la transition la sortie peut prendre la valeur de 0 ou de 1 (la sortie peut prendre sa valeur définitive un peu plus tôt ou un peu plus tard). On note donc un *don't care*.

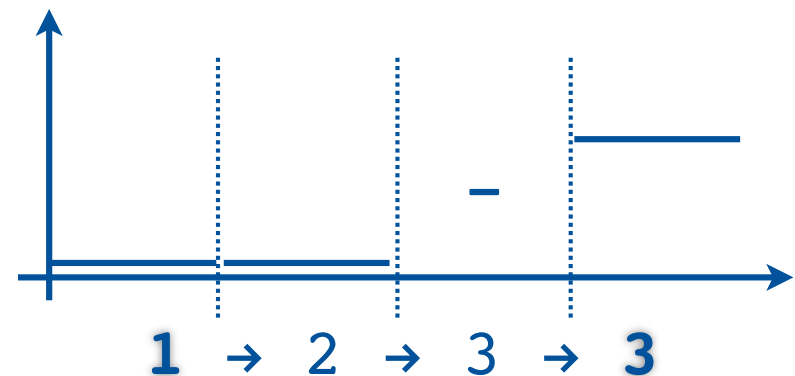
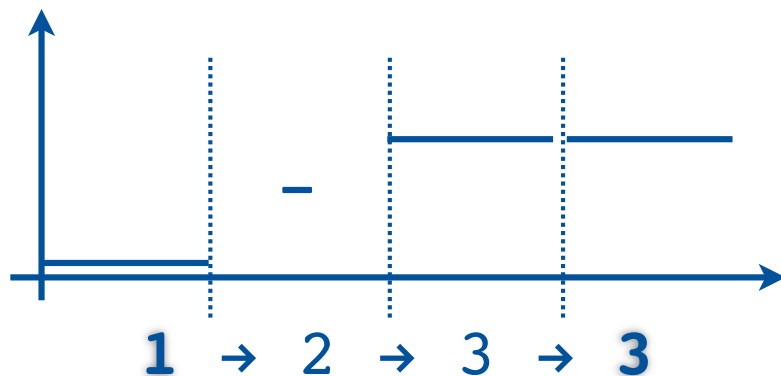


Synthèse de la fonction de sortie : plusieurs transitions



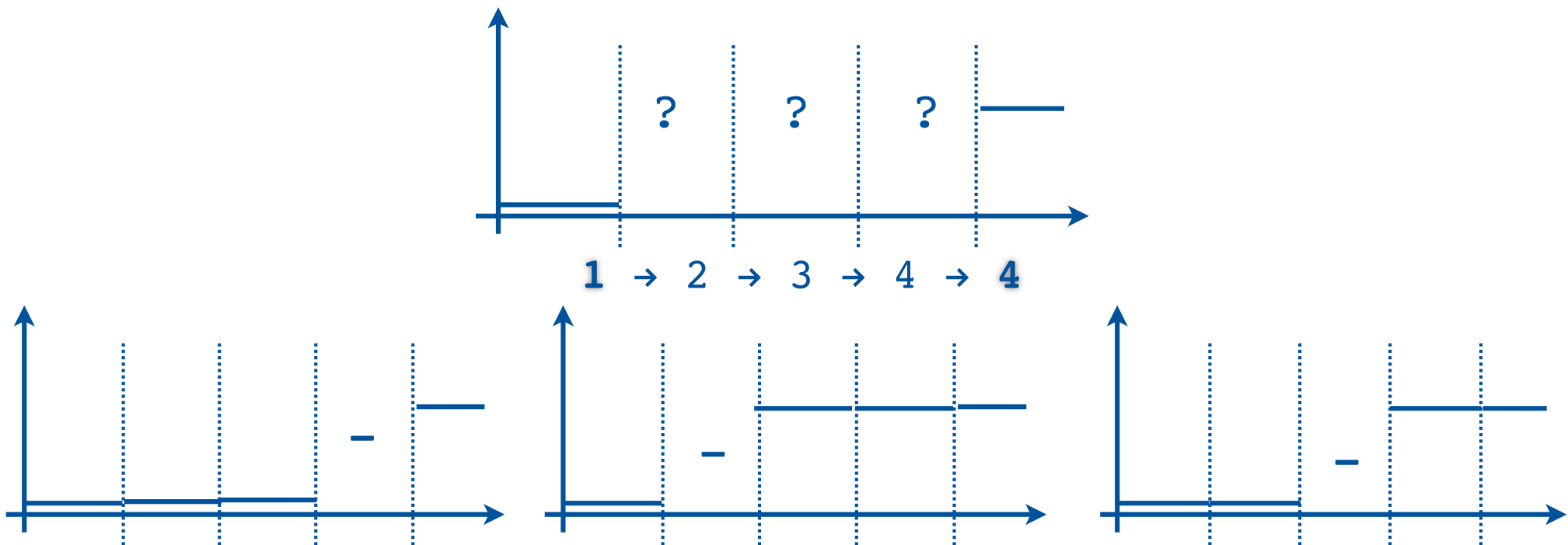
Règle est très simple : on ne peut avoir qu'un seul changement de la sortie de 0 à 1 (ou de 1 à 0).

Pour deux transitions on doit imposer la valeur de la sortie pour une transition, l'autre peut prendre n'importe quel autre valeur.



Synthèse de la fonction de sortie : plusieurs transitions

Même principe lorsque trois transitions (ou plus) : deux transitions doivent être fixées et une transition peut prendre la valeur de *don't care*. On choisit l'une des trois possibilités en fonction de l'expression finale.



Synthèse de la fonction de sortie

Attention!

Une transition peut être utilisée pour le passage entre différentes états stables :

	ab				
	00	01	11	10	Z
1	1	2	2		0
2		2	3		1
3			3		0
4					

$2 \rightarrow 3 \rightarrow 3$

mais aussi

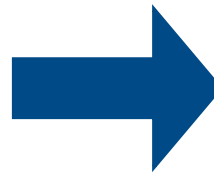
$1 \rightarrow 2 \rightarrow 3 \rightarrow 3$

C'est le passage de $1 \rightarrow 2 \rightarrow 3 \rightarrow 3$ qui va fixer la valeur de la sortie pour la transition 3 car la sortie vaut 0 pour les l'états 1 et 3

Synthèse de la fonction de sortie — Machine de Moore

Table de sortie complète et l'expression de Z pour l'exemple considéré :

	ab				
	00	01	11	10	z
1	1	2	2	—	0
2	1	2	3	—	1
3	3	2	3	4	0
4	—	4	3	4	1



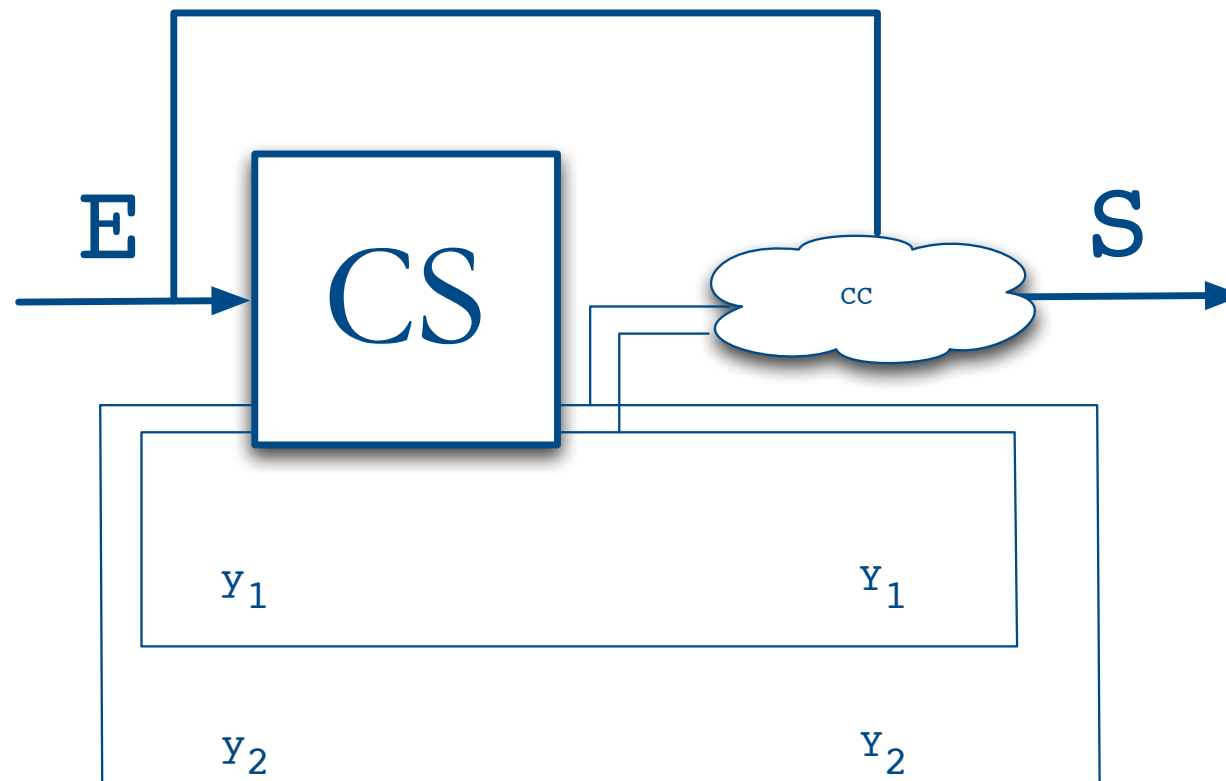
	ab				
z	00	01	11	10	
00	0	—	0	—	
01	—	1	0	—	
11	0	—	0	—	
10	—	1	—	1	

Deux solutions: $z = a'b + y_1y_2'$ ou $z = a'b + ab'$

Synthèse de la machine de Meally

Machine de Meally : la sortie est fonction des variables d'état et des entrées

$$S = f(Y_i, E)$$



Synthèse de la machine de Meally

Dans une machine de Moore la sortie est fonction des états uniquement...

Afin d'éviter les *glitches* à la sortie, on fixe la valeur de(s) sortie(s) pour les transitions.

L'expression finale de la sortie est quand même fonction des entrées (en fonction des valeurs des transitions fixées...). Pour l'exemple nous avons:

$$Z = a' b + y_1 y_2'$$

Est-ce une machine de Meally? → **Non!**

Synthèse de la machine de Meally

Dans une Machine de Meally, la sortie est fonction des entrées aussi
→ un même état stable peut produire des sorties différentes !!!

On note la valeur de la sortie **à côté de l'état stable** (ce sont les entrées qui permettent de faire la distinction entre les deux situations).

	ab			
	00	01	11	10
1	1/1		2	1/0
2			2/0	
3				
4				

On différencie **1/1** de **1/0** à l'aide des entrées $ab=00$, $ab=10$

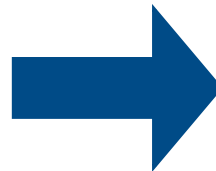
Synthèse de la machine de Meally

- ❖ La possibilité d'avoir un même état stable avec des sorties différentes veut dire qu'il est possible de fusionner les états ayant des sorties différentes!
- ❖ Lors de la synthèse d'une machine de Meally la contrainte de fusionnement des états avec la même sortie n'est donc plus valable!
- ❖ Ce que l'on gagne : à priori on peut fusionner plus d'états, ce qui nous permet d'avoir une économie d'organes de mémoire et de fonctions logiques associées (i.e. portes)

Synthèse de la machine de Meally

Exemple : les états 1 et 4 peuvent être fusionnés malgré la sortie différente (création d'un nouvel état 1).

	ab				
	00	01	11	10	z
1	1	3	2	–	1
2					0
3					0
4	1	3		4	0



	ab				
	00	01	11	10	
1	1/1	3	2	1/0	
2					
3					

Synthèse de la machine de Meally

Exception : On ne peut pas fusionner deux états stables pour la même combinaison des entrées (se trouvant dans la même colonne) ayant la sortie différente.

	ab				
	00	01	11	10	z
1	1		2		1
2			2		0
3	3				0
4					

Les états 1 et 3 ne peuvent être fusionnés en aucun cas car il serait impossible de faire la distinction entre les deux états !

Synthèse de la machine de Meally — Exemple complet

Même **Table d'état primitive** utilisée pour la synthèse

I. Machine de Moore

et puis

II. Machine de Meally

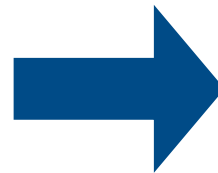
Nous allons comparer
les deux solutions obtenues en terme
de ressources nécessaires pour leur
réalisation.

	ab				
	00	01	11	10	z
1	1	2	–	3	0
2	1	2	4	–	1
3	1	–	4	3	0
4	–	5	4	–	1
5	6	5	4	–	0
6	6	5	–	3	1

I. Synthèse de la machine de Moore

La table d'équivalences pour une Machine de Moore (on respecte la fonction de sortie, elles doivent être les mêmes).

	ab				
	00	01	11	10	Z
1	1	2	–	3	0
2	1	2	4	–	1
3	1	–	4	3	0
4	–	5	4	–	1
5	6	5	4	–	0
6	6	5	–	3	1



Première passe					
2	X				
3	OK	X			
4	X	2–5	X		
5	1–6 2–5	X	1–6	X	
6	X	1–6 2–5	X	OK	X
	1	2	3	4	5

I. Synthèse de la machine de Moore

Table et graphe de fusionnement.

Deuxième passe : 1-6, 2-5 NOK...

2	X				
3	OK	X			
4	X	2 5	X		
5	1 6 2 5	X	1 6	X	
6	X	1 6 2 5	X	OK	X
	1	2	3	4	5

1-3

4-6

2

I. Synthèse de la machine de Moore

Différent fusionnements : Table d'état primitive et la Table d'état réduite

2 → 2

1-3 → 1

4-6 → 3

5 → 4

	ab				
	00	01	11	10	Z
1	1	2	–	3	0
2	1	2	4	–	1
3	1	–	4	3	0
4	–	5	4	–	1
5	6	5	4	–	0
6	6	5	–	3	1

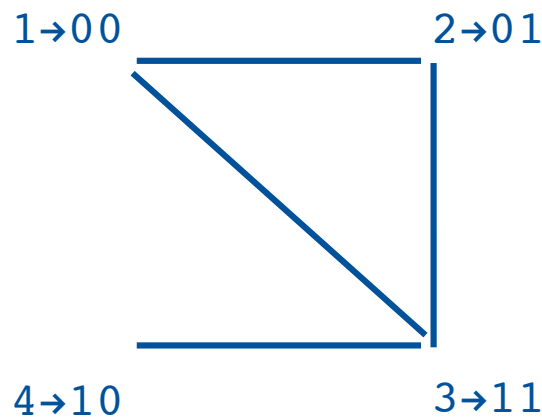
	ab				
	00	01	11	10	Z
1	1	2	–	1	0
2	1	2	3	–	1
3	3	4	3	1	1
4	3	4	3	–	0

I. Synthèse de la machine de Moore

Codage de la table d'état réduite : nous avons une course critique à résoudre

ab

	00	01	11	10	Z
1	1	2	-	1	0
2	1	2	3	-	1
3	3	4	3	1	1
4	3	4	3	-	0



ab

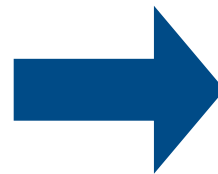
	00	01	11	10
0	00	01	-	0
1	00	01	11	-
11	11	10	11	00
10	11	10	11	-

Course

I. Synthèse de la machine de Moore

Résolution de problème de course critique :
moyennant la transition via l'état 01.

	ab				
	00	01	11	10	z
00	00	01	–	00	0
01	00	01	11	–	1
11	11	10	11	00	1
10	11	10	11	–	0



	ab				
	00	01	11	10	z
00	00	01	–	00	0
01	00	01	11	00	1
11	11	10	11	01	1
10	11	10	11	–	0

Course

I. Synthèse de la machine de Moore

Fonctions de
rétroaction

	ab				Z
	00	01	11	10	
00	00	01	–	00	0
01	00	01	11	00	1
11	11	10	11	01	1
10	11	10	11	–	0

	ab			
Y ₁	00	01	11	10
00	0	0	–	0
01	0	0	1	0
11	1	1	1	0
10	1	1	1	–

$$Y_1 = a'y_1 + ab$$

	ab			
Y ₂	00	01	11	10
00	0	1	–	0
01	0	1	1	0
11	1	0	1	1
10	1	0	1	–

$$Y_2 = ab + b'y_1 + by_1'$$

I. Synthèse de la machine de Moore

Fonction de sortie

	ab				
	00	01	11	10	z
00	00	01	–	00	0
01	00	01	11	00	1
11	11	10	11	01	1
10	11	10	11	–	0

Etats stables

	ab			
z	00	01	11	10
00	0		–	0
01		1		
11	1		1	
10		0		–

Transitions

	ab			
z	00	01	11	10
00	0	–	–	0
01	–	1	1	–
11	1	–	1	1
10	–	0	–	–

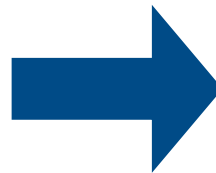
$$Z = y_2$$

II. Synthèse de la machine de Meally

La table d'équivalences pour une Machine de Meally — on peut fusionner les états avec les sorties différentes — sauf **1** et **6** et **2** et **5**.

ab

	00	01	11	10	Z
1	1	2	—	3	0
2	1	2	4	—	1
3	1	—	4	3	0
4	—	5	4	—	1
5	6	5	4	—	0
6	6	5	—	3	1



Première passe

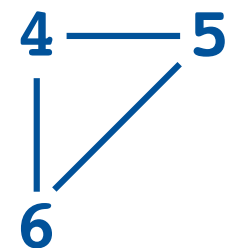
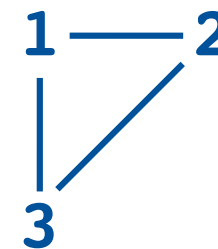
2	OK				
3	OK	OK			
4	2-5	2-5	OK		
5	1-6	X	1-6	OK	
6	X	1-6 2-5	1-6	OK	OK
	1	2	3	4	5

II. Synthèse de la machine de Mealy

Table et graphes de fusionnement

Deuxième passe : à cause de 2-5, 1-6...

2	OK				
3	OK	OK			
4	2 5	2 5	OK		
5	1 6	X	1 6	OK	
6	X	1 6 2 5	1 6	OK	OK
	1	2	3	4	5



II. Synthèse de la machine de Meally

Fusionnement et la table d'état réduite

1, 2, 3 → 1

4, 5, 6 → 2

	ab				
	00	01	11	10	Z
1	1	2	–	3	0
2	1	2	4	–	1
3	1	–	4	3	0
4	–	5	4	–	1
5	6	5	4	–	0
6	6	5	–	3	1

	ab			
	00	01	11	10
1	1/0	1/1	2	1/0
2	2/1	2/0	2/1	1

II. Synthèse de la machine de Meally

Codage de la table d'état réduite : une seule variable d'état Y_1

	ab			
	00	01	11	10
1	1/0	1/1	2	1/0
2	2/1	2/0	2/1	1

	ab			
Y_1	00	01	11	10
0	0	0	1	0
1	1	1	1	0

$$Y_1 = y_1 a' + ab$$

Il ne reste plus que la synthèse de la fonction de sortie ...

Ici, on ouvre une petite parenthèse :

Synthèse de la sortie pour une Machine de Meally

Fonction de sortie — cas d'une Machine de Meally

Le principe est le même comme pour une machine de Moore :

On ne peut avoir qu'un seul changement de la valeur de sortie, quelque soit le nombre de transitions entre deux états stables.

Mais attention!

Dans une Machine de Meally deux mêmes états stables peuvent avoir une sortie différente.

Ceci a une implication sur les évolutions possibles du système.

Fonction de sortie — cas d'une Machine de Mealy

Cas d'une seule transition

On arrive à l'état 2 à partir de l'état 1 avec une entrée $ab=11$.

Mais on peut partir :

- a) de 1 avec $ab=00$, sortie 1 (transition en trait pointillé) ou
- b) de 1 avec $ab=10$, sortie 0 (transition en trait plein).

	ab			
	00	01	11	10
1	1/1		2	1/0
2			2/0	
3				
4				

C'est le passage de 1 pour $ab=10$ avec sortie 0 (transition en trait plein) qui va déterminer la valeur de la sortie.

Fonction de sortie — cas d'une Machine de Meally

Cas d'une seule transition

Pour la transition 2 on est obligé de mettre un 0 !
(attention dans ce cas un *don't care* serait une faute grave)



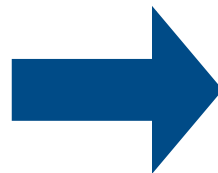
Fonction de sortie — cas d'une Machine de Meally

Cas de plusieurs transitions

La transition 3 pour $ab=11$ est utilisée pour passer de 1 à 3, mais aussi pour passer de 2 à 3.

On commence par fixer la sortie de passage de 2 à 3, c'est elle qui fixera la sortie de la transition 3.

	ab			
	00	01	11	10
1	1/1		2	1/1
2		2/0	3	
3			3/0	
4				



	ab			
z	00	01	11	10
1	1		—	1
2		0	0	
3			0	
4				

II. Synthèse de la machine de Meally

La fonction de sortie pour l'exemple : on recopie la sortie pour les états stables et on fixe la sortie pour les deux transitions.

ab

	00	01	11	10
1	1/0	1/1	2	1/0
2	2/1	2/0	2/1	1

ab

Y_1	00	01	11	10
0	0	0	1	0
1	1	1	1	0

$$Y_1 = y_1 a' + ab$$

ab

Z	00	01	11	10
0	0	1	1	0
1	1	0	1	0

$$Z = y_1 a' b' + y_1' b + ab$$

Synthèse de la machine de Moore et de Meally

Comparaison de deux résultats :

Machine de Moore

$$Y_1 = a' y_1 + ab$$

$$Y_2 = ab + b' y_1 + a' b y_1'$$

$$Z = y_2$$

Machine de Meally

$$Y_1 = y_1 a' + ab$$

$$Z = y_1 a' b' + y_1' b + ab$$

Fautes graves

1. Ne pas résoudre les transitions pour la(es) fonction(s) de sortie. (Considérer que la fonction de sortie est fonction uniquement des variables d'état.)
2. Dans le cas d'une machine de Meally, fusionner deux états stables de la même colonne avec des sorties différentes.
3. Lorsque l'on demande de construire une machine de Meally (de faire les fusionnements sans respecter les sorties), faire une machine de Moore (deux solutions sont fondamentalement différentes).