

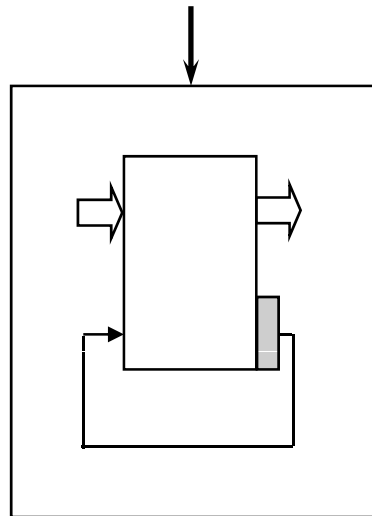
Synthèse des systèmes séquentiels synchrones

Ecole Polytechnique Universitaire de Montpellier
Université Montpellier II
Place Eugène Bataillon, 34095 Montpellier cedex 05, FRANCE

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
UMR 5506 Université Montpellier II / CNRS
161 rue Ada, 34392 Montpellier cedex 05, FRANCE

Architecture générale d'un circuit

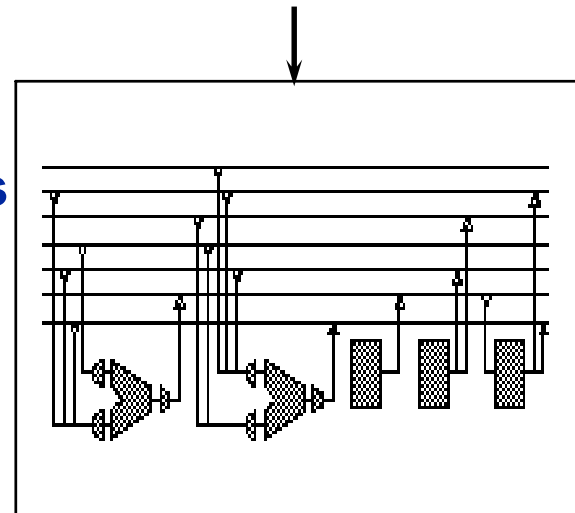
Entrées commandes



Commandes

Statuts

Entrées data



Sorties

CONTROLEUR (PC)

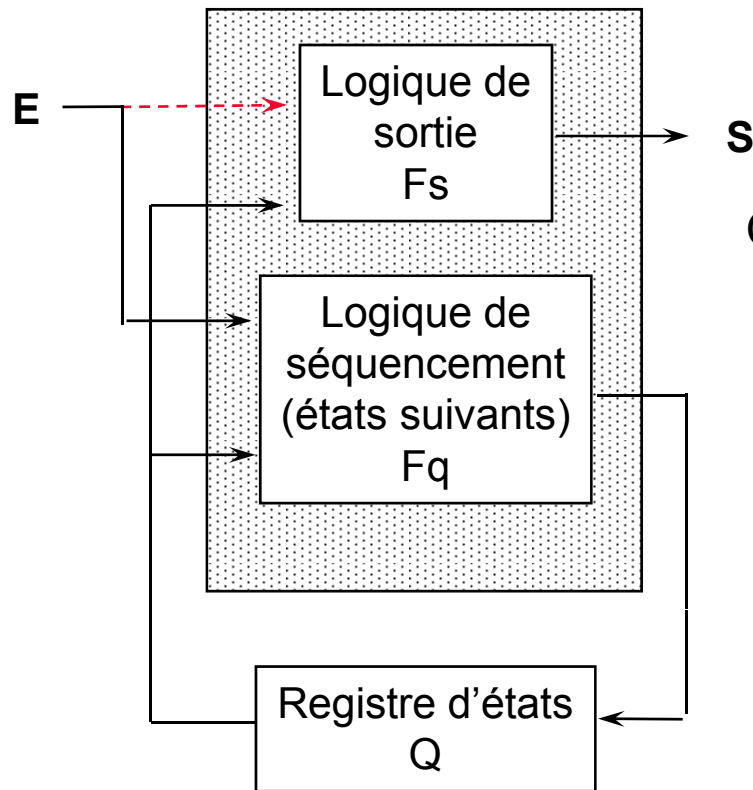
CHEMIN DE DONNEES (PO)

Contrôleur synchrone

Deux blocs combinatoires (calcul de l'état suivant et des sorties)

Un registre d'état à N entrées parallèles

Etats codés sur N bits, $N : \log_2 (Q)$



Quintuplet: (E, S, Q, Fs, Fq)

E : Ensemble des entrées primaires

S : Ensemble des sorties primaires

Q : Ensemble des états

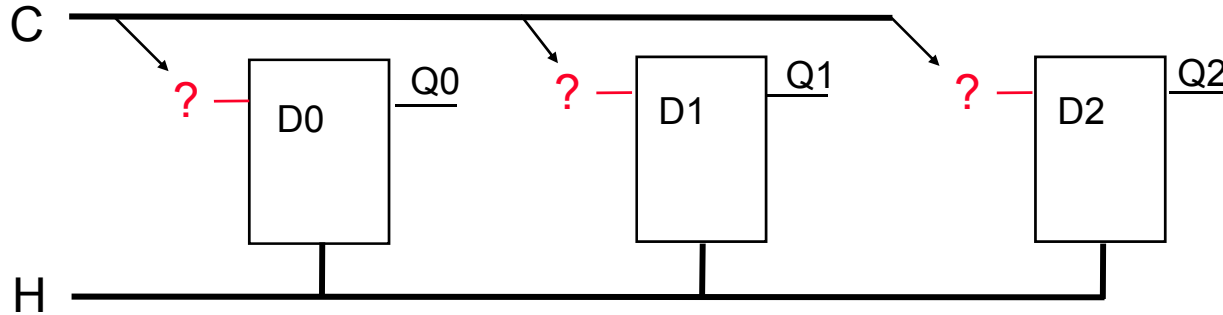
Fs : $S * I \rightarrow S$ Fonction de transition d'états

Fq : $S \rightarrow O$ Fonction de sortie (Moore)

$Fs = f(Q) \Rightarrow Moore$

$Fs = f(Q, E) \Rightarrow Mealy$

Compteur (D)



D	Q(n+1)
0	0
1	1

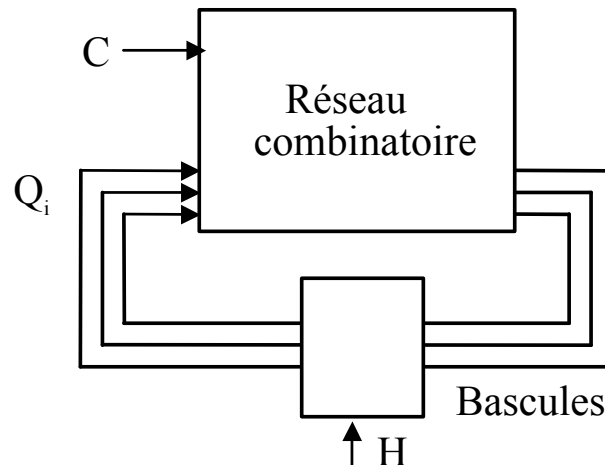
C=0 => Compteur
C=1 => RAZ Synchrone

Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

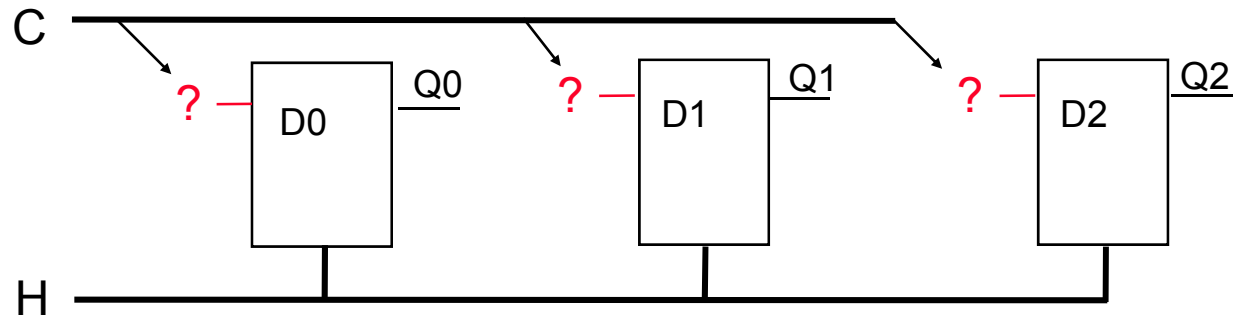
$$D0 = C'.Q0'$$

$$D1 = C'. [Q0.Q1' + Q0'.Q1] = C'. [Q0 \oplus Q1]$$

$$D2 = C'. [Q0.Q1.Q2' + (Q0.Q1)'.Q2] = C'. [Q0.Q1 \oplus Q2]$$



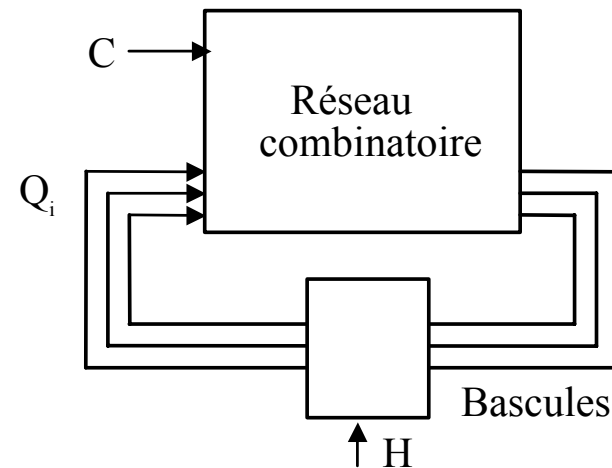
Séquenceur



C=0 => Séquenceur
C=1 => RAZ Synchrone

Q2	Q1	Q0
0	0	0
1	1	1
0	0	1
0	1	1
1	1	0
0	1	0
1	0	0
1	0	1

D0 = ?
D1 = ?
D2 = ?



Principe – Compteur/Séquenceur

Etats	Etats Suivants	
	C = 0	C = 1
000	001	000
001	010	000
010	011	000
011	100	000
100	101	000
101	110	000
110	111	000
111	000	000

Principe – Compteur/Séquenceur

D(n)	Q(n+1)
0	0
1	1

$$Q(n+1) = D(n)$$

Etats $Q_2Q_1Q_0(n)$	Etats Suivants $Q_2Q_1Q_0(n+1)$		C = 0			C = 1		
	C = 0	C = 1	$D_2(n)$	$D_1(n)$	$D_0(n)$	$D_2(n)$	$D_1(n)$	$D_0(n)$
000	001	000	0	0	1	0	0	0
001	010	000	0	1	0	0	0	0
010	011	000	0	1	1	0	0	0
011	100	000	1	0	0	0	0	0
100	101	000	1	0	1	0	0	0
101	110	000	1	1	0	0	0	0
110	111	000	1	1	1	0	0	0
111	000	000	0	0	0	0	0	0

Principe – Compteur/Séquenceur

		Q0C			
Q2Q1		00	01	11	10
	00				
	01				1
	11	1			
	10	1			1

$$D2 = C'.(Q2' \oplus Q1.Q0)$$

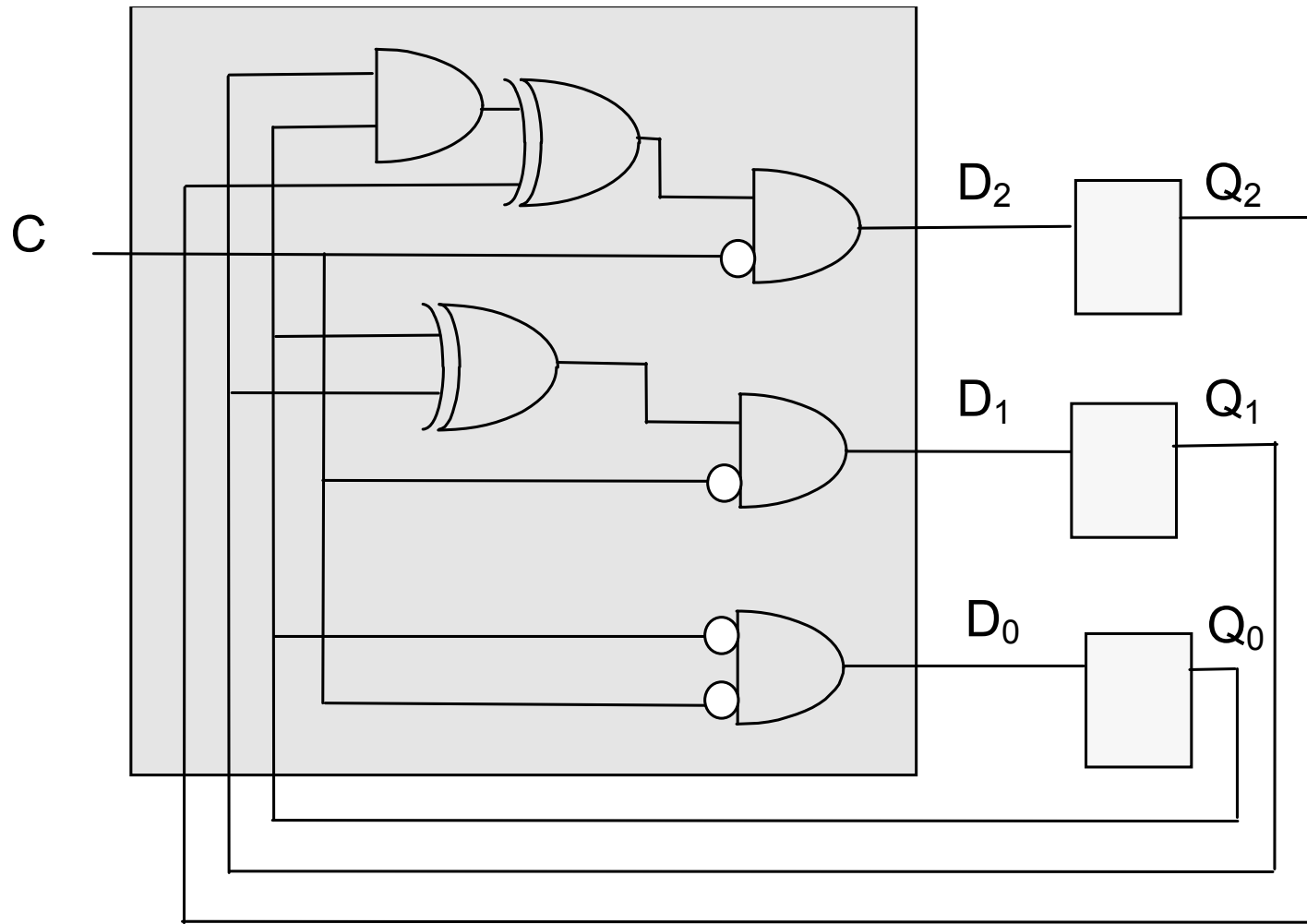
		Q0C			
Q2Q1		00	01	11	10
	00				1
	01	1			
	11	1			
	10				1

$$D1 = C'.(Q1 \oplus Q0)$$

		Q0C			
Q2Q1		00	01	11	10
	00	1			
	01	1			
	11	1			
	10	1			

$$D0 = C'.Q0'$$

Principe – Compteur/Séquenceur



Codage des entrées de bascules (T)

T	Q _{n+1}
0	Q _n
1	Q' _n

Q _n -> Q _{n+1}	T
0 -> 0	0
0 -> 1	1
1 -> 0	1
1 -> 1	0

Etats Q ₂ Q ₁ Q ₀ (n)	Etats Suivants Q ₂ Q ₁ Q ₀ (n+1)		C = 0			C = 1		
	C = 0	C = 1	T ₂	T ₁	T ₀	T ₂	T ₁	T ₀
000	001	000	0	0	1	0	0	0
001	010	000	0	1	1	0	0	1
010	011	000	0	0	1	0	1	0
011	100	000	1	1	1	0	1	1
100	101	000	0	0	1	1	0	0
101	110	000	0	1	1	1	0	1
110	111	000	0	0	1	1	1	0
111	000	000	1	1	1	1	1	1

Codage des entrées de bascules (JK)

JK	Q _{n+1}
00	Q _n
01	0
10	1
11	Q' _n

Q _n -> Q _{n+1}	JK
0 -> 0	0Φ
0 -> 1	1Φ
1 -> 0	Φ1
1 -> 1	Φ0

Etats Q ₂ Q ₁ Q ₀ (n)	Etats Suivants Q ₂ Q ₁ Q ₀ (n+1)		C = 0			C = 1		
	C = 0	C = 1	J ₂ K ₂	J ₁ K ₁	J ₀ K ₀	J ₂ K ₂	J ₁ K ₁	J ₀ K ₀
000	001	000	0Φ	0Φ	1Φ	0Φ	0Φ	0Φ
001	010	000	0Φ	1Φ	Φ 1	0Φ	0Φ	Φ 1
010	011	000	0Φ	Φ 0	1Φ	0Φ	Φ 1	0Φ
011	100	000	1Φ	Φ 1	Φ 1	0Φ	Φ 1	Φ 1
100	101	000	Φ 0	0Φ	1Φ	Φ 1	0Φ	0Φ
101	110	000	Φ 0	1Φ	Φ 1	Φ 1	0Φ	Φ 1
110	111	000	Φ 0	Φ 0	1Φ	Φ 1	Φ 1	0Φ
111	000	000	Φ 1	Φ 1	Φ 1	Φ 1	Φ 1	Φ 1

Généralisation

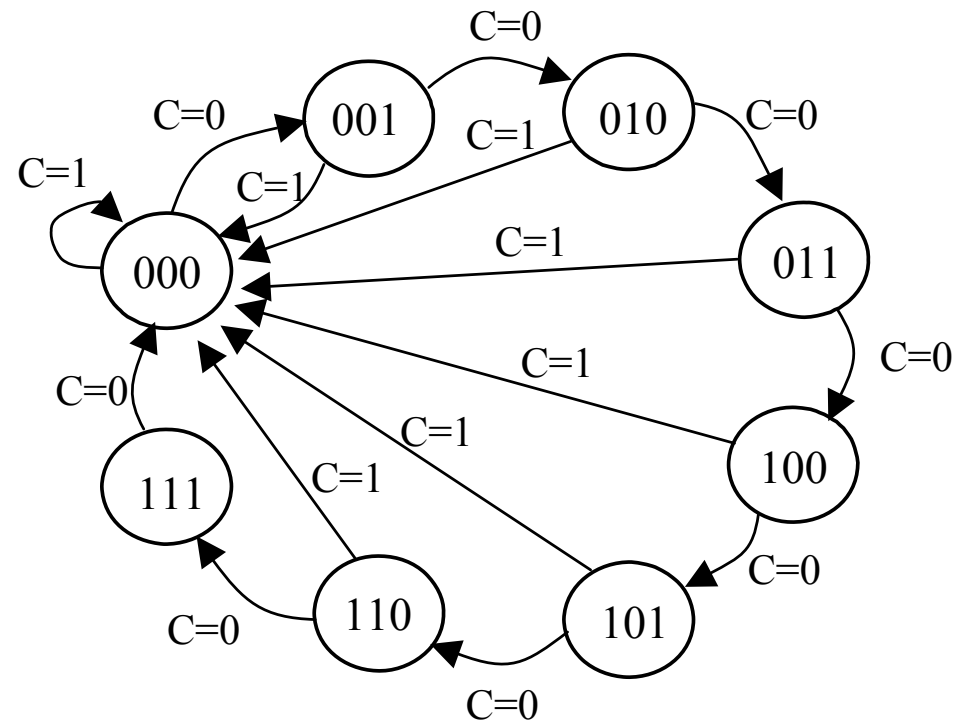
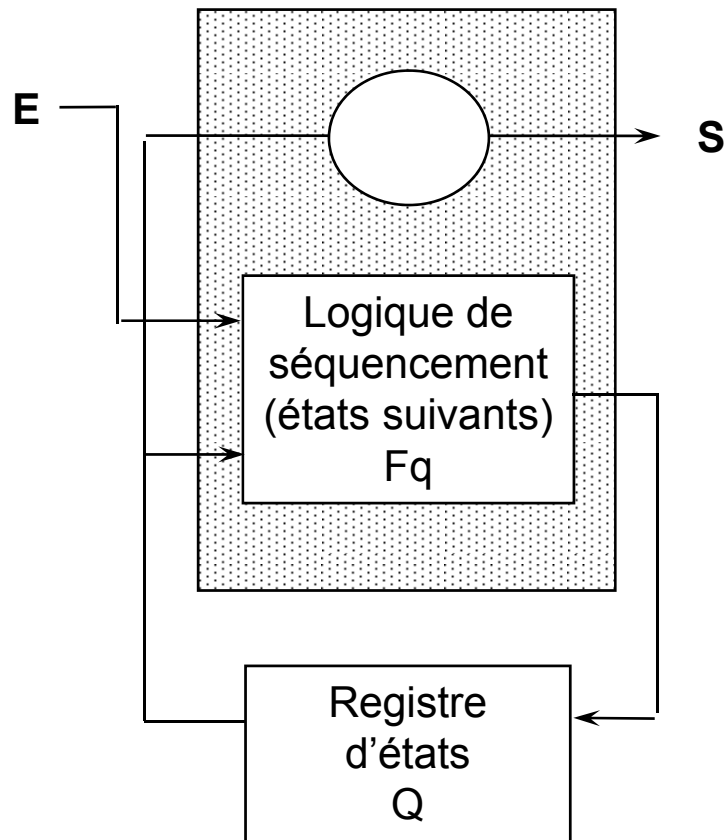
- *Cahiers des charges (Langage) -> Circuit*
 - Exemple de Cahier des charges : *Le système considéré a une entrée (E) et une sortie (S). Il reçoit sur son entrée des bits arrivant en série. La sortie (S) doit passer à 1 chaque fois qu'une séquence 010 apparaît sur l'entrée (E) puis repasser à 0 sur le bit suivant quel que soit sa valeur.*
- *Codes des état non déterminé à priori*
- *Fonctions de sorties $\neq$ états*
- *Optimisation (# bascules, # portes, ...)*

Synthèse - Méthode d'Huffman-Mealy

- ❑ 1: *Modélisation du cahier des charges*
 - Graphe d'état
 - Table d'état
- ❑ 2: *Minimisation du nombre d'états*
 - Règles de minimisation
 - Détermination du nombre de bascules minimum
- ❑ 3: *Codage des états*
 - Codage des états
 - Codage des entrées de bascules
- ❑ 4: *Synthèse*
 - Synthèse des entrées de bascules
 - Synthèse des sorties

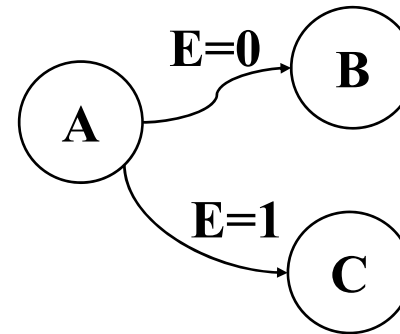
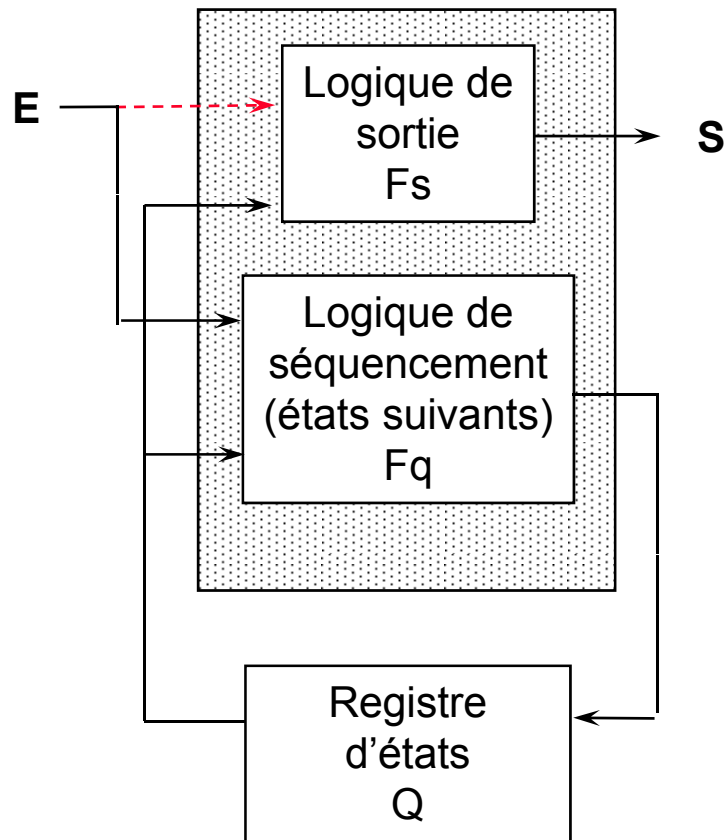
Modélisation du cahier des charges

Compteur / Séquenceur

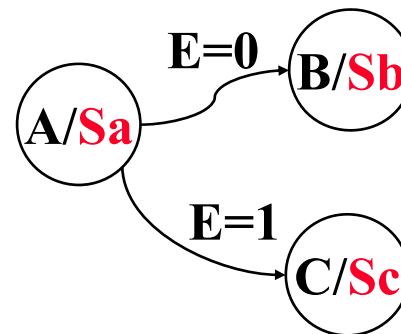


Modélisation du cahier des charges

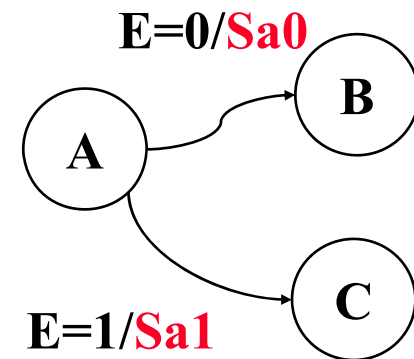
Graphe d'état



Machine
de
Moore

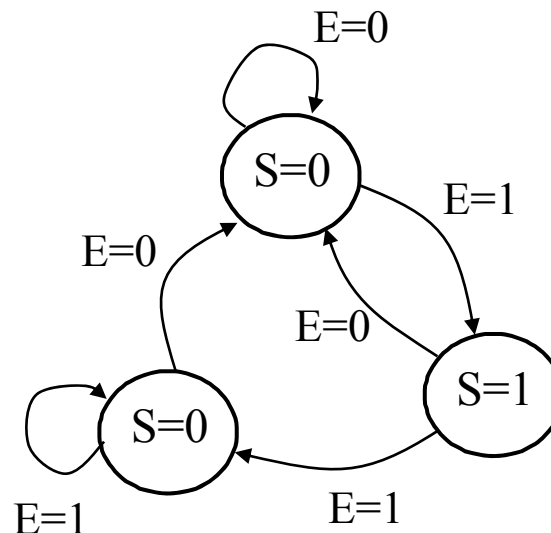
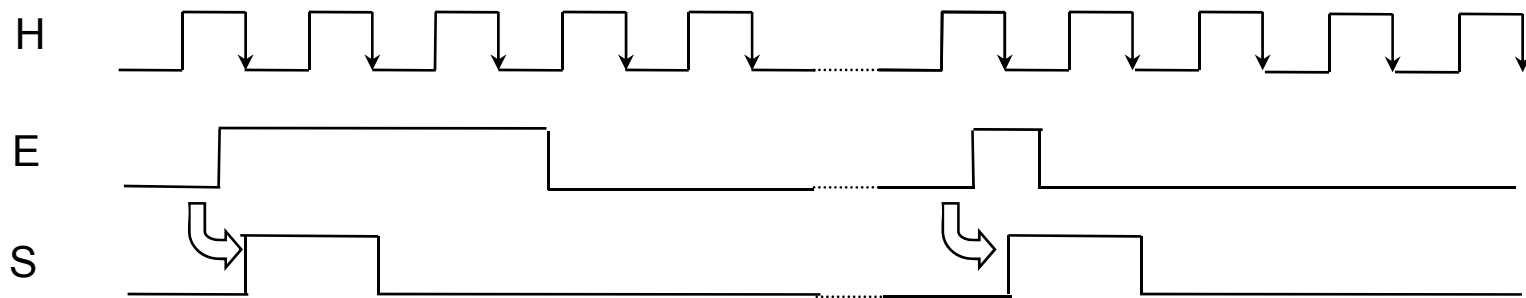


Machine
de
Mealy



Modélisation du cahier des charges (Moore)

Cahier des charges : Le système considéré a une entrée (E) et une sortie (S). La sortie doit générer une impulsion de largeur égale à la période d'horloge chaque fois que l'entrée E passe à 1.

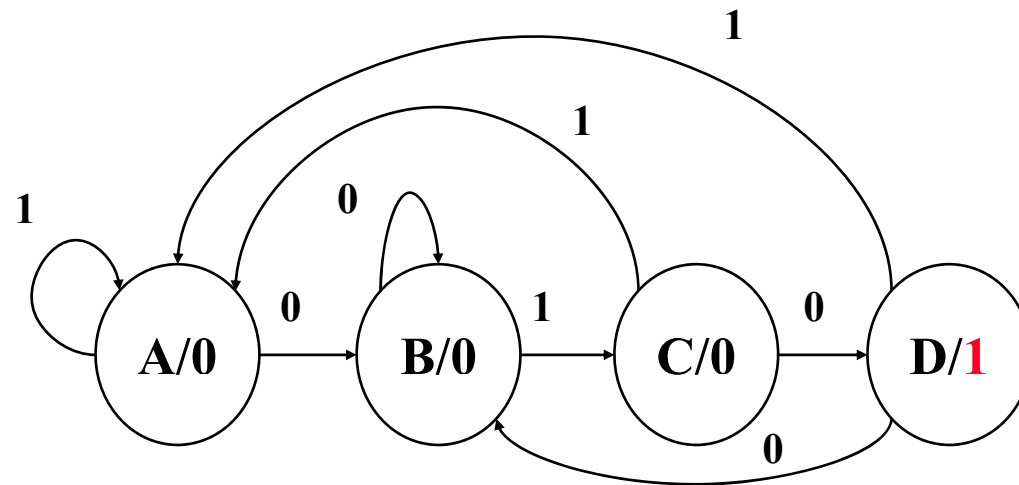


(Moore)

Modélisation du cahier des charges (Moore)

Cahier des charges : Le système considéré a une entrée (E) et une sortie (S). Il reçoit sur son entrée des bits arrivant en série. La sortie (S) doit passer à 1 chaque fois qu'une séquence 010 apparaît sur l'entrée (E) puis repasser à 0 sur le bit suivant quel que soit sa valeur.

Graphe d'état (Moore)



Modélisation du cahier des charges (Moore)

Graphe d'état

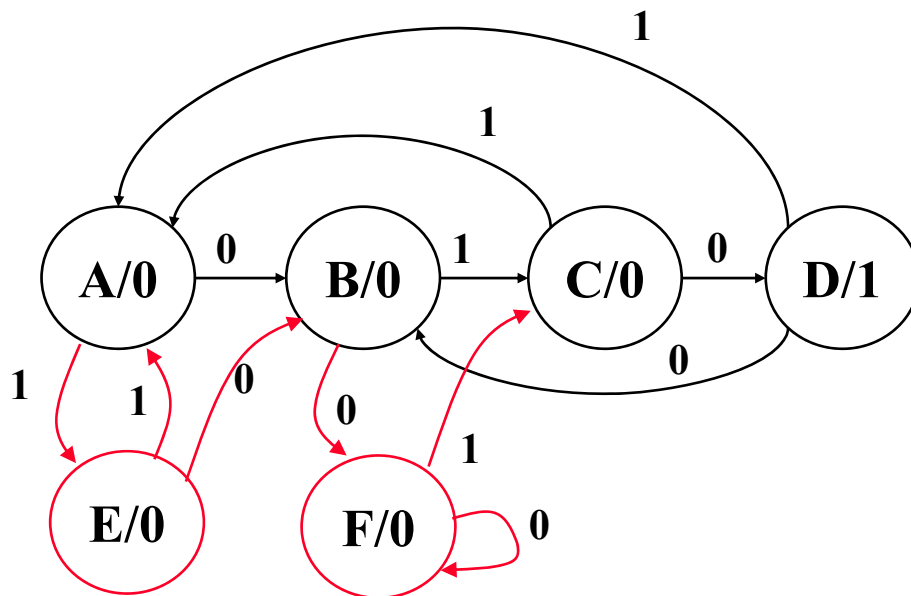


Table d'état

Etats	Etats Suivants		Sortie
	0	1	
A	B	E	0
B	F	C	0
C	D	A	0
D	B	A	1
E	B	A	0
F	F	C	0

Minimisation du nombre d'états

- **Règle 1 (R1):** Deux états sont équivalents si pour chaque combinaison d'entrée, ils ont mêmes sorties et mêmes états suivants.

Table d'état

Etats	Etats Suivants		Sortie
	0	1	
A	B	E	0
B	F	C	0
C	D	A	0
D	B	A	1
E	B	A	0
F	F	C	0

B $\Leftrightarrow$ F

Table d'état réduite

Etats	Etats Suivants		Sortie
	0	1	
A	B	E	0
B	B	C	0
C	D	A	0
D	B	A	1
E	B	A	0

Minimisation du nombre d'états

- **Règle 2 (R2):** Les états sont regroupés en différentes classes selon les valeurs de sorties associées. Ainsi, deux états ayant mêmes sorties (pour chaque combinaison d'entrée) sont dans la même classe. Les états appartenant à une même classe sont équivalents s'il ne peuvent être séparés. Or les états appartenant à une même classe doivent être séparés si les états suivants associés à chacun d'eux sont dans des classes différentes.

Table d'état

Etats	Etats Suivants		Sortie
	0	1	
A	B	E	0
B	B	C	0
C	D	A	0
D	B	A	1
E	B	A	0

2 classes

(1) (2)
(A , B , C, E) (D)

Minimisation du nombre d'états

(1)	(2)	Classe
(A , B , C , E)	(C)	Etats
BE BC AD BA		Etats suivants
11 11 12 11		Classes des états suivants

L'état C doit être exclut de la classe 1 et le processus réitéré.

(1)	(2)	(3)	Classe
(A , B , E)	(C)	(D)	Etats
BE BC BA			Etats suivants
11 12 11			Classes des états suivants

L'état B doit être exclut de la classe 1 et le processus réitéré.

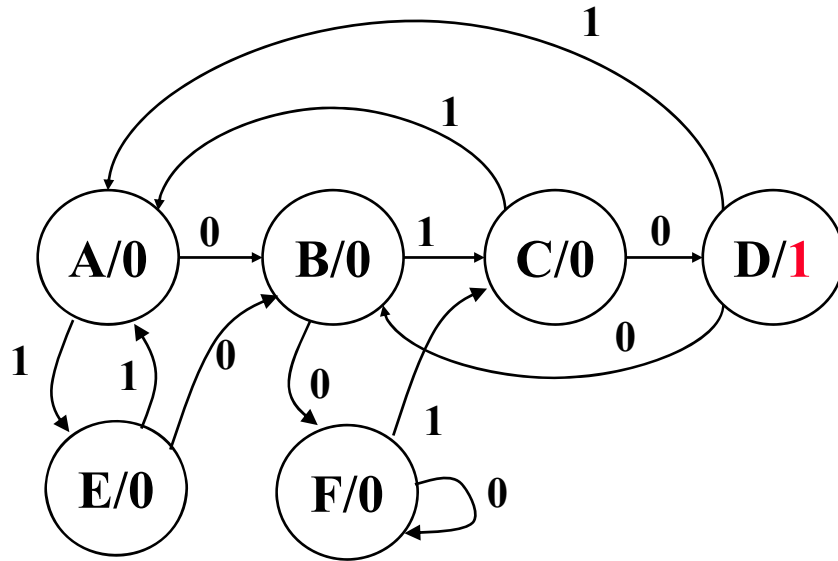
(1)	(2)	(3)	(4)	Classe
(A , E)	(B)	(C)	(D)	Etats
BE BA				Etats suivants
21 12				Classes des états suivants

Les états A et E peuvent rester dans la même classe. Ils sont équivalents.

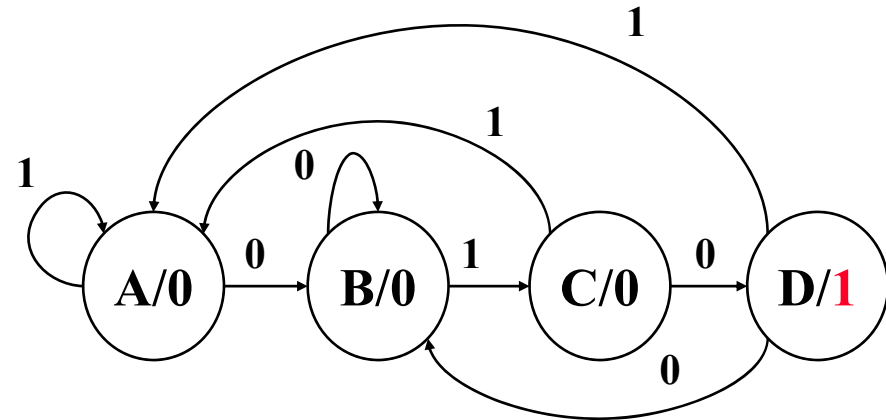
Etats	Etats Suivants		Sortie
	0	1	
A	B	E	0
B	B	C	0
C	D	A	0
D	B	A	1
E	B	A	0

Etats	Etats Suivants		Sortie
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	A	1

Graphe d'états : Interprétation



Etats	Etats Suivants		Sortie
	0	1	
A	B	E	0
B	F	C	0
C	D	A	0
D	B	A	1
E	B	A	0
F	F	C	0



Etats	Etats Suivants		Sortie
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	A	1

Codage des états

Le nombre minimum d'états "p" étant déterminé, on peut en déduire le nombre minimum "n" de variables d'état et par conséquent de bascules nécessaires au codage de ces états à partir de la double inéquation suivante:

$$2^{n-1} < p < 2^n$$

2 états : 1 bascule

3,4 états : 2 bascules

5,6,7,8 états : 3 bascules

...

Etats	Etats Suivants		Sortie
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	A	1

Etats Q1Q2	Etats Suivants		Sortie
	0	1	
00 (A)	01	00	0
01 (B)	01	11	0
11 (C)	10	00	0
10 (D)	01	00	1

Codage des entrées de bascules (D)

D(n)	Q(n+1)
0	0
1	1

$$Q(n+1) = D(n)$$

Etats Q1Q2	Etats Suivants		Sortie	E = 0		E = 1	
	0	1		D1	D2	D1	D2
00 (A)	01	00	0	0	1	0	0
01 (B)	01	11	0	0	1	1	1
11 (C)	10	00	0	1	0	0	0
10 (D)	01	00	1	0	1	0	0

Synthèse

Etats Q1Q2	Etats Suivants		Sortie	E = 0		E = 1	
	0	1		D1	D2	D1	D2
00 (A)	01	00	0	0	1	0	0
01 (B)	01	11	0	0	1	1	1
11 (C)	10	00	0	1	0	0	0
10 (D)	01	00	1	0	1	0	0

		E	
Q1Q2		0	1
	00	0	0
	01	0	1
	11	1	0
	10	0	0
		D1	

		E	
Q1Q2		0	1
	00	1	0
	01	1	1
	11	0	0
	10	1	0
		D2	

$$D1 = E.Q1'.Q2 + E'.Q1.Q2$$

$$D2 = Q1'.Q2. + E'.Q2'$$

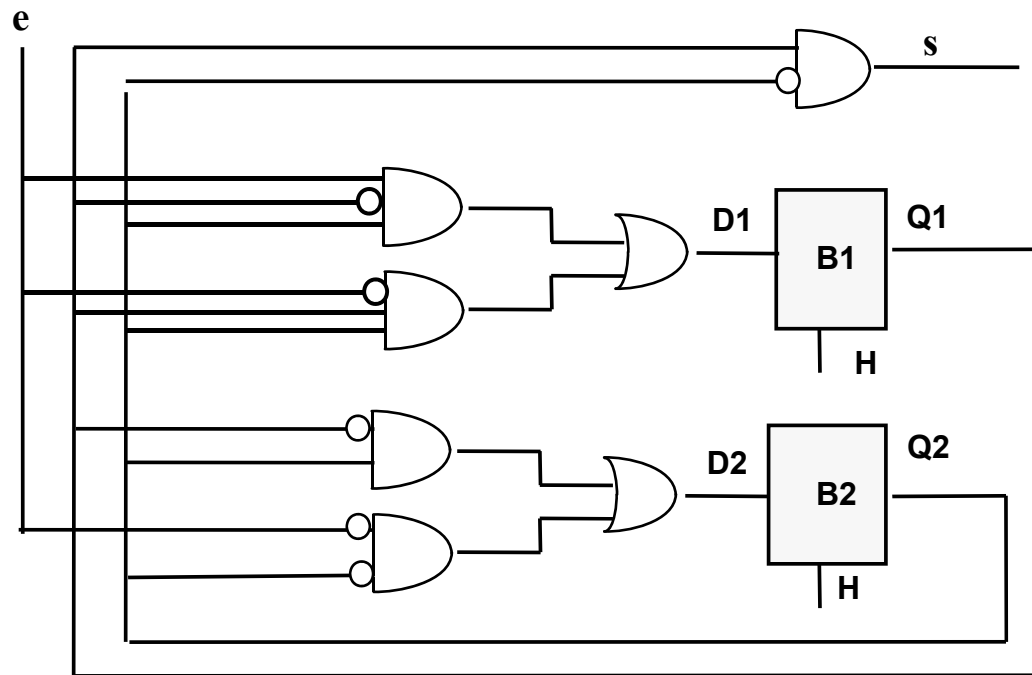
$$S = Q1.Q2'$$

Synthèse

$$S = Q1.Q2'$$

$$D1 = E.Q1'.Q2 + E'.Q1.Q2$$

$$D2 = Q1'.Q2 + E'.Q2'$$



Influence du codage

Etats Q1Q2	Etats Suivants		Sortie	E = 0		E = 1	
	0	1		D1	D2	D1	D2
00 (A)	01	00	0	0	1	0	0
01 (B)	01	10	0	0	1	1	0
10 (C)	11	00	0	1	1	0	0
11 (D)	01	00	1	0	1	0	0

Q1Q2	E		
		0	1
00	0	0	0
01	0	1	1
11	0	0	0
10	1	1	0
D1			

Q1Q2	E		
		0	1
00	1	0	0
01	1	0	0
11	1	0	0
10	1	0	0
D2			

$$D1 = E.Q1'.Q2 + E'.Q1.Q2'$$

$$D2 = E'$$

$$S = Q1.Q2$$

$$S = Q1.Q2'$$

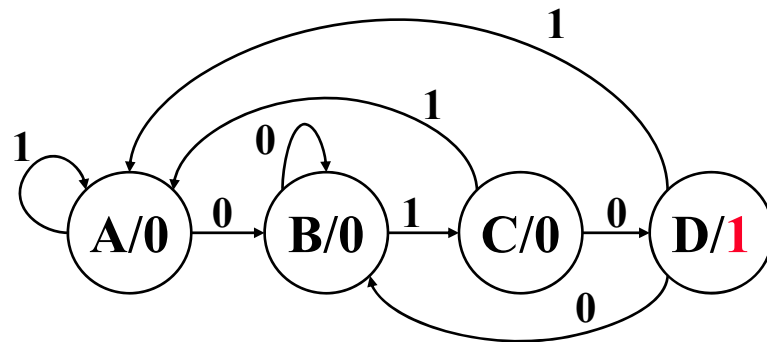
$$D1 = E.Q1'.Q2 + E'.Q1.Q2$$

$$D2 = Q1'.Q2 + E'.Q2'$$

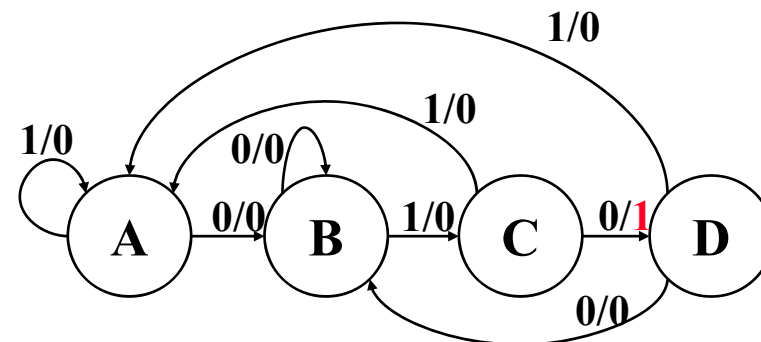
Comparaison Moore / Mealy

Cahier des charges : Le système considéré a une entrée (E) et une sortie (S). Il reçoit sur son entrée des bits arrivant en série. La sortie (S) doit passer à 1 chaque fois qu'une séquence 010 apparaît sur l'entrée (E) puis repasser à 0 sur le bit suivant quel que soit sa valeur.

Graphe d'état (Moore)



Graphe d'état (Mealy)



Comparaison Moore / Mealy

Graphe d'état (Moore)

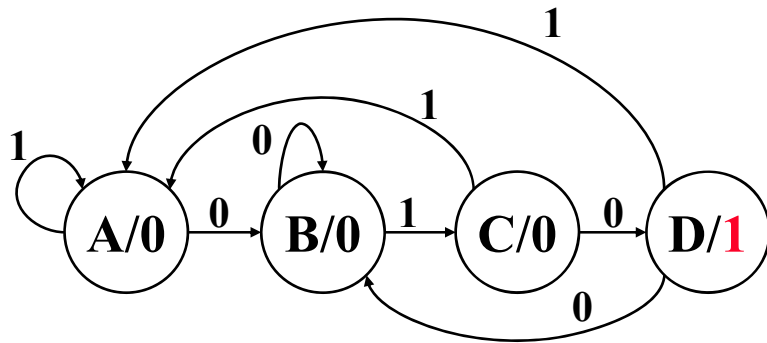


Table d'état

Etats	Etats Suivants		Sortie
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	A	1

Graphe d'état (Mealy)

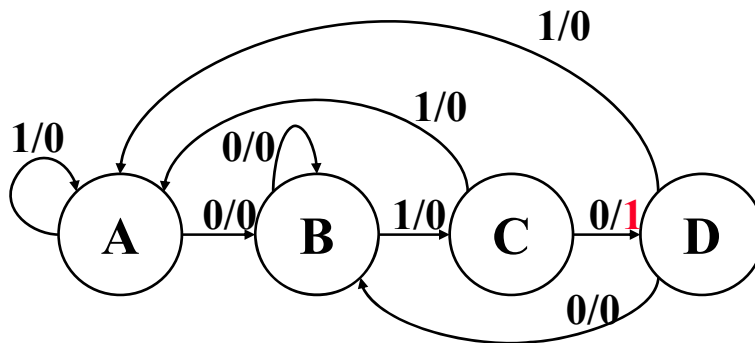


Table d'état

Etats	Etats Suivants		Sortie	
	0	1	0	1
A	B	A	0	0
B	B	C	0	0
C	D	A	1	0
D	B	A	0	0

Minimisation du nombre d'états (Moore)

Etats	Etats Suivants		Sortie
	0	1	
A	B	A	0
B	B	C	0
C	D	A	0
D	B	A	1

R1 : Pas d'états équivalents

R2 : Pas d'états équivalents

(1)
(A , B , C)
BA BC DA
11 11 21
suivants

(2) Classe
(D) Etats
Etats suivants
Classes des états

L'état C doit être exclu de la classe 1 et le processus réitéré.

(1)
(A , B)
BA BC
11 12
suivants

(2) (3) Classe
(C) (D) Etats
Etats suivants
Classes des états

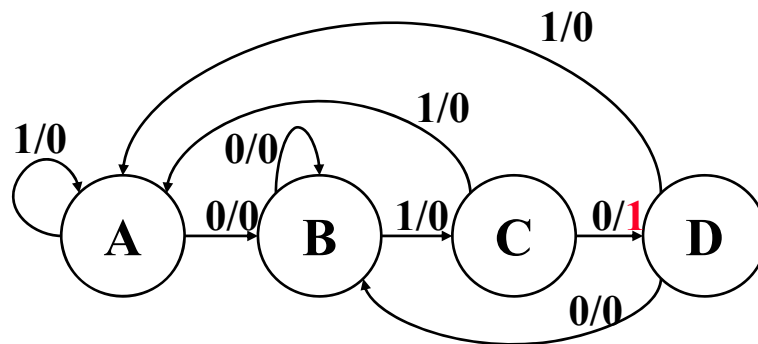
Les états A et B doivent être séparés. Il y a maintenant qu'un seul état par classe. Il n'y a donc pas d'états équivalents. Pour réaliser cette machine, le nombre minimum d'états est de 4.

Minimisation du nombre d'états (Mealy)

R1 : $A \Leftrightarrow D$

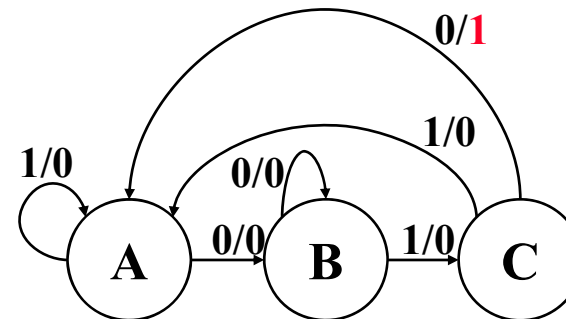
R2 : Pas d'états équivalents

Etats	Etats Suivants		Sortie	
	0	1	0	1
A	B	A	0	0
B	B	C	0	0
C	D	A	1	0
D	B	A	0	0



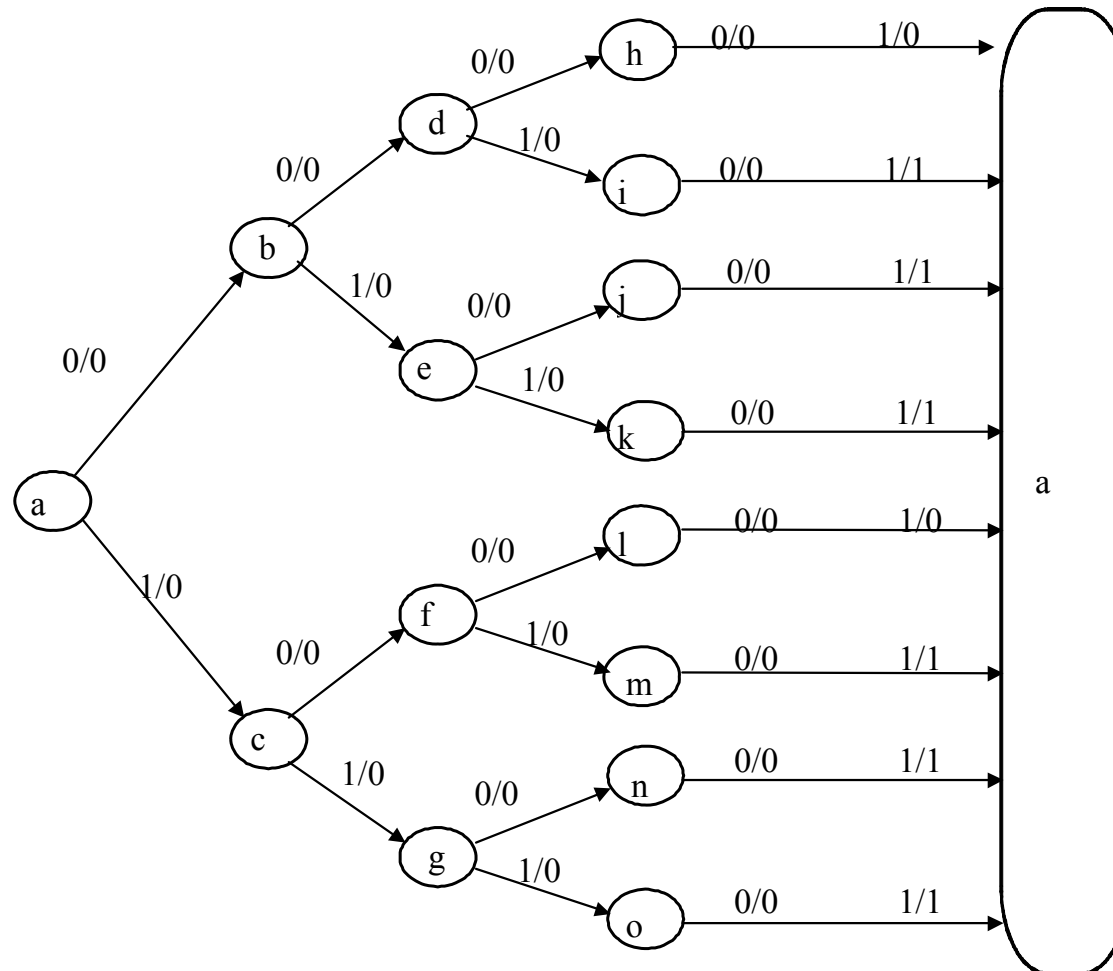
(1) (2)
 (A , B) (C)
 BA BC
 11 12

Etats	Etats Suivants		Sortie	
	0	1	0	1
A	B	A	0	0
B	B	C	0	0
C	A	A	1	0



Modélisation du cahier des charges

Cahier des charges : On désire réaliser une machine recevant sur son entrée E des mots de 4 bits en série (poids faible en tête). Sa sortie S doit passer à 1 chaque fois qu'un mot représente un nombre supérieur à 9 et revenir à 0 sur le premier bit du mot suivant.



Minimisation du nombre d'états

Etats	Etats Suivants		Sortie	
	0	1	0	1
a	b	c	0	0
b	d	e	0	0
c	f	g	0	0
d	h	i	0	0
e	j	k	0	0
f	l	m	0	0
g	n	o	0	0
h	a	a	0	0
i	a	a	0	1
j	a	a	0	1
k	a	a	0	1
l	a	a	0	0
m	a	a	0	1
n	a	a	0	1
o	a	a	0	1

Etats	Etats Suivants		Sortie	
	0	1	0	1
a	b	c	0	0
b	d	e	0	0
c	f	g	0	0
d	h	i	0	0
e	i	i	0	0
f	h	i	0	0
g	i	i	0	0
h	a	a	0	0
i	a	a	0	1

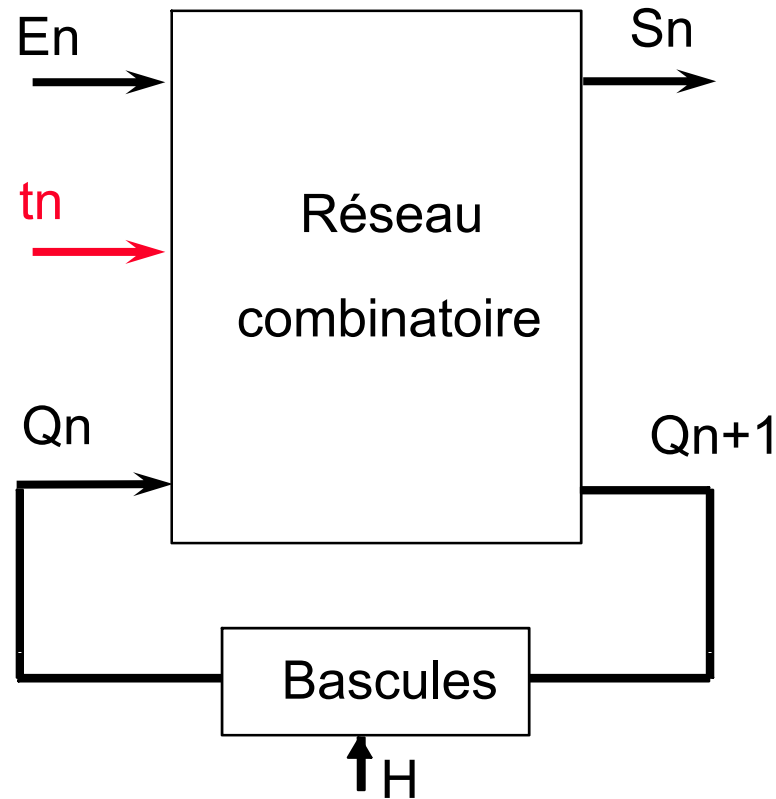
Minimisation du nombre d'états

Etats	Etats Suivants		Sortie	
	0	1	0	1
a	b	c	0	0
b	d	e	0	0
c	d	e	0	0
d	h	i	0	0
e	i	i	0	0
h	a	a	0	0
i	a	a	0	1

Etats	Etats Suivants		Sortie	
	0	1	0	1
a	b	b	0	0
b	d	e	0	0
d	h	i	0	0
e	i	i	0	0
h	a	a	0	0
i	a	a	0	1

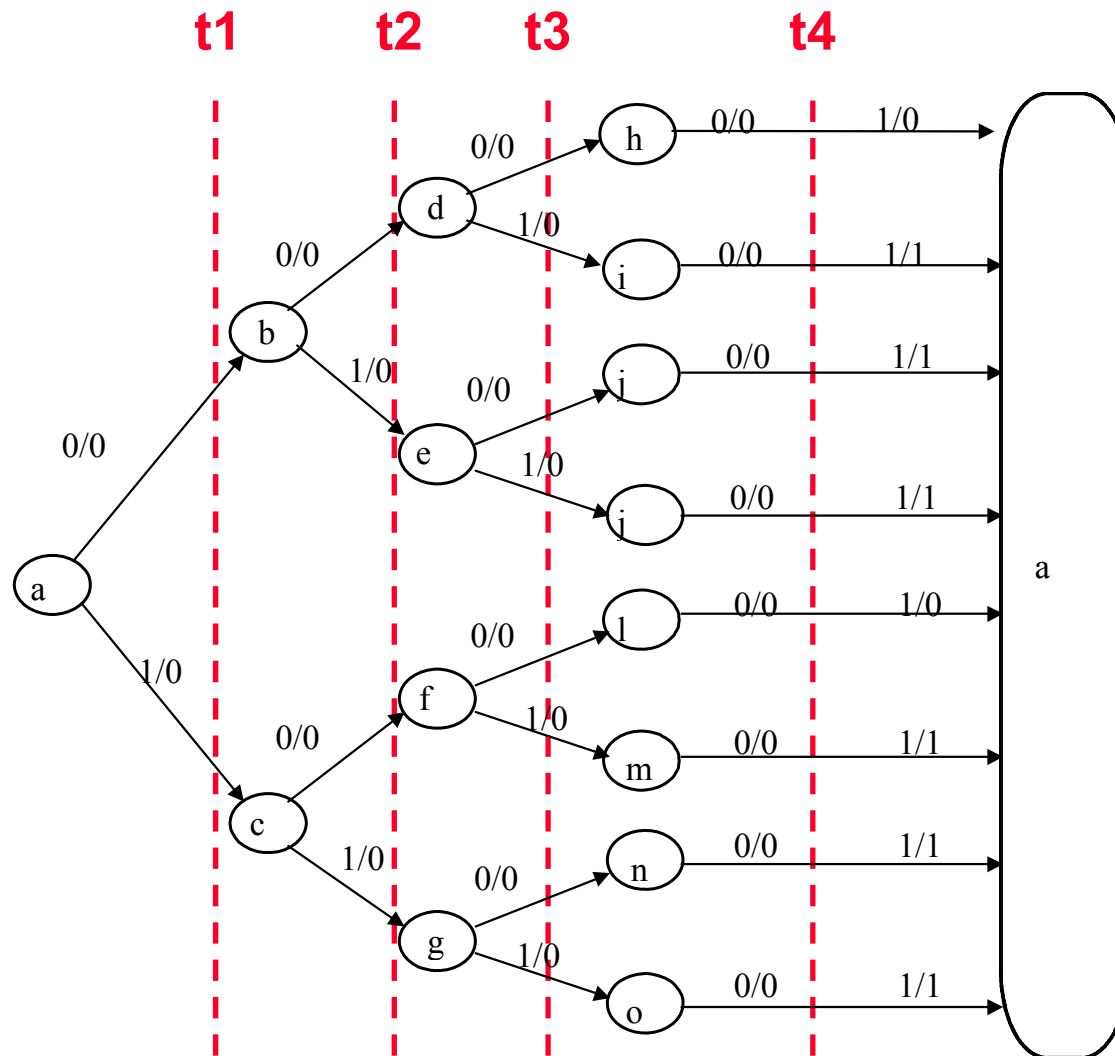
6 états => 3 bascules

Machine à temps explicite



$t_1 \ t_2 \ t_3 \ t_4 = 1000 \Rightarrow$ le premier bit est présent sur les entrées de la machine
= 0100 $\Rightarrow$ le deuxième bit est présent sur les entrées de la machine
= 0010 $\Rightarrow$ le troisième bit est présent sur les entrées de la machine
= 0001 $\Rightarrow$ le quatrième bit est présent sur les entrées de la machine

Machine à temps explicite



Machine à temps explicite

Etats	Etats Suivants								Sortie			
	t1=1		t2=1		t3=1		t4=1		t1	t2	t3	t4
	0	1	0	1	0	1	0	1	01	01	01	01
a	b	c	-	-	-	-	-	-	00	--	--	--
b	-	-	d	e	-	-	-	-	--	00	--	--
c	-	-	f	g	-	-	-	-	--	00	--	--
d	-	-	-	-	h	i	-	-	--	--	00	--
e	-	-	-	-	j	k	-	-	--	--	00	--
f	-	-	-	-	l	m	-	-	--	--	00	--
g	-	-	-	-	n	o	-	-	--	--	00	--
h	-	-	-	-	-	-	a	a	--	--	--	00
i	-	-	-	-	-	-	a	a	--	--	--	01
j	-	-	-	-	-	-	a	a	--	--	--	01
k	-	-	-	-	-	-	a	a	--	--	--	01
l	-	-	-	-	-	-	a	a	--	--	--	00
m	-	-	-	-	-	-	a	a	--	--	--	01
n	-	-	-	-	-	-	a	a	--	--	--	01
o	-	-	-	-	-	-	a	a	--	--	--	01

Machine à temps explicite

Etats	Etats Suivants								Sortie			
	t1=1		t2=1		t3=1		t4=1		t1	t2	t3	t4
	0	1	0	1	0	1	0	1	01	01	01	01
a	b	b	-	-	-	-	-	-	00	--	--	--
b	-	-	d	e	-	-	-	-	--	00	--	--
d	-	-	-	-	h	i	-	-	--	--	00	--
e	-	-	-	-	i	i	-	-	--	--	00	--
h	-	-	-	-	-	-	a	a	--	--	--	00
i	-	-	-	-	-	-	a	a	--	--	--	01

Etats	Etats Suivants								Sortie			
	t1=1		t2=1		t3=1		t4=1		t1	t2	t3	t4
	0	1	0	1	0	1	0	1	01	01	01	01
a	b	b	-	-	a	b	a	a	00	--	00	00
b	-	-	a	b	b	b	a	a	--	00	00	01

Machine à temps explicite

Etats	Etats Suivants								Sortie			
	t1=1		t2=1		t3=1		t4=1		t1	t2	t3	t4
	0	1	0	1	0	1	0	1	01	01	01	01
a	b	b	-	-	a	b	a	a	00	--	00	00
b	-	-	a	b	b	b	a	a	--	00	00	01

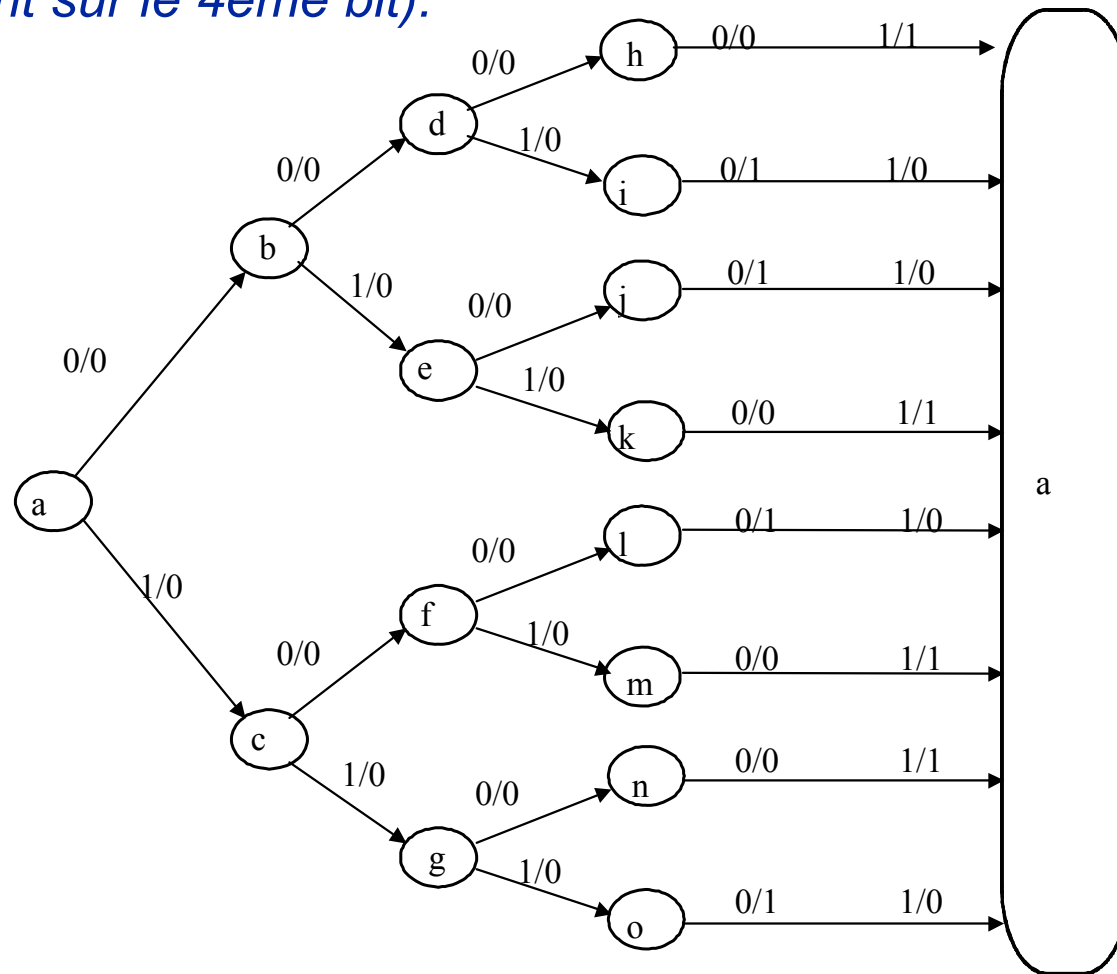
Etats	Etats Suivants								Sortie			
	t1=1		t2=1		t3=1		t4=1		t1	t2	t3	t4
	0	1	0	1	0	1	0	1	01	01	01	01
0	1	1	-	-	0	1	0	0	00	--	00	00
1	-	-	0	1	1	1	0	0	--	00	00	01

$$S = t4.e.Q$$

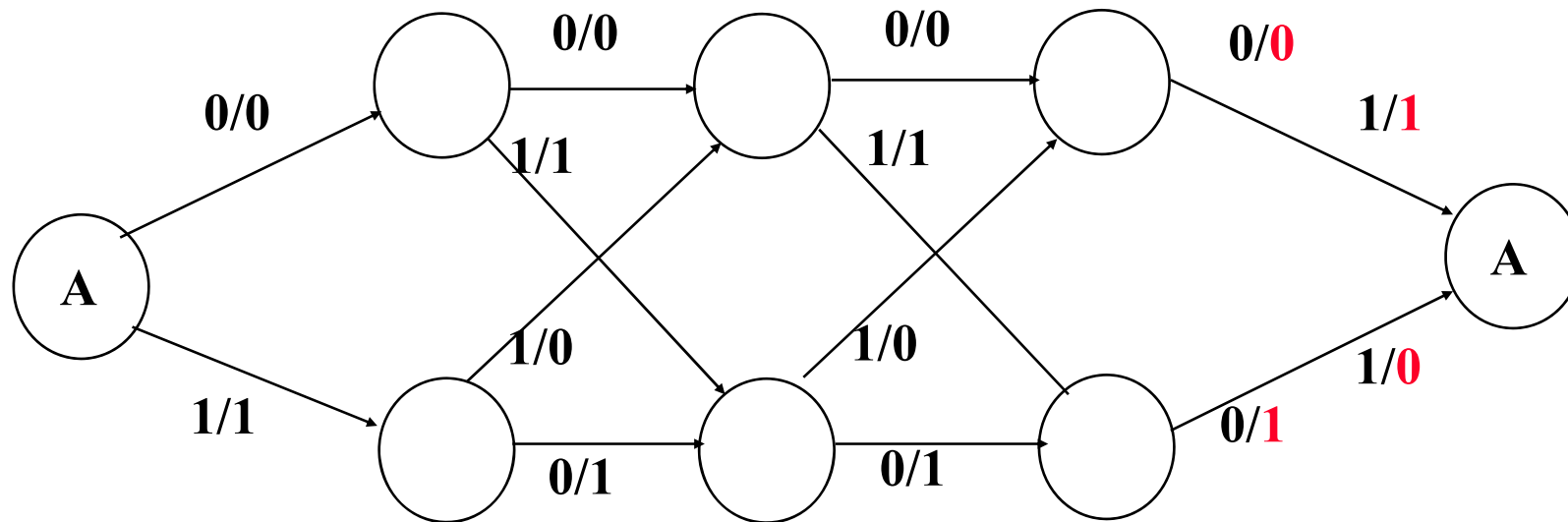
$$D = t1 + t2.e + t3.(e + Q)$$

Modélisation du cahier des charges

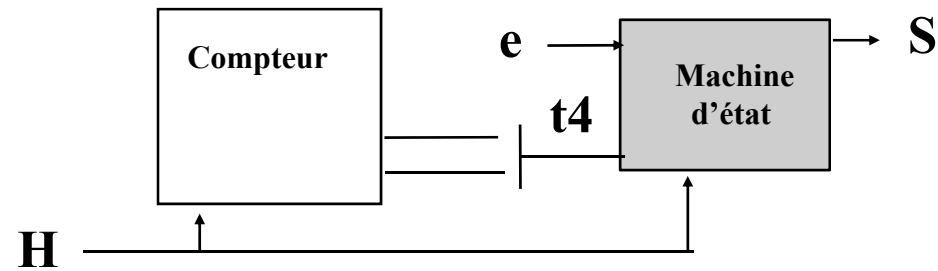
Cahier des charges : On désire réaliser une machine recevant sur son entrée e des mots de 4 bits en série (E). Sa sortie S doit passer à 1 chaque fois qu'un mot contient un nombre impair de 1 (Passage à 1 uniquement sur le 4ème bit).



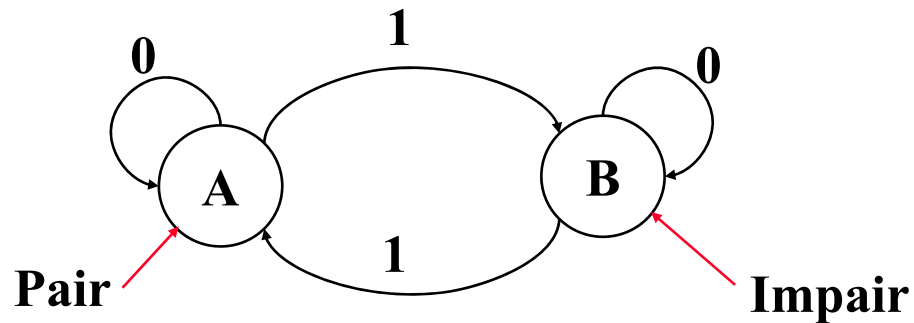
Modélisation du cahier des charges



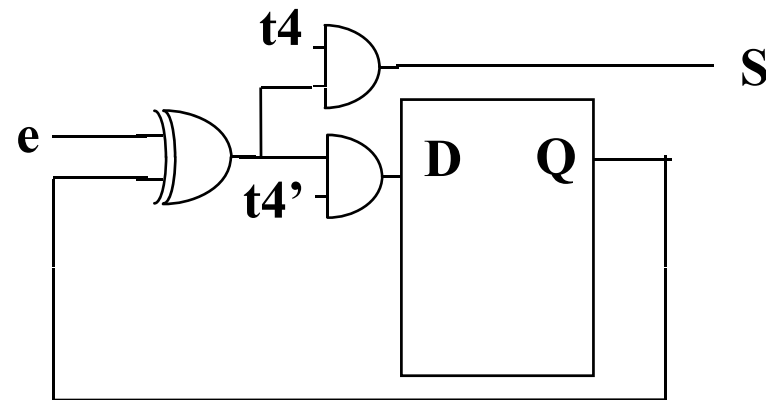
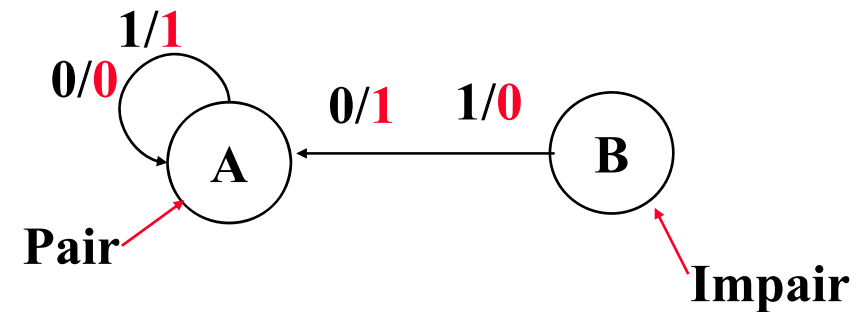
Temps explicite / Partitionnement



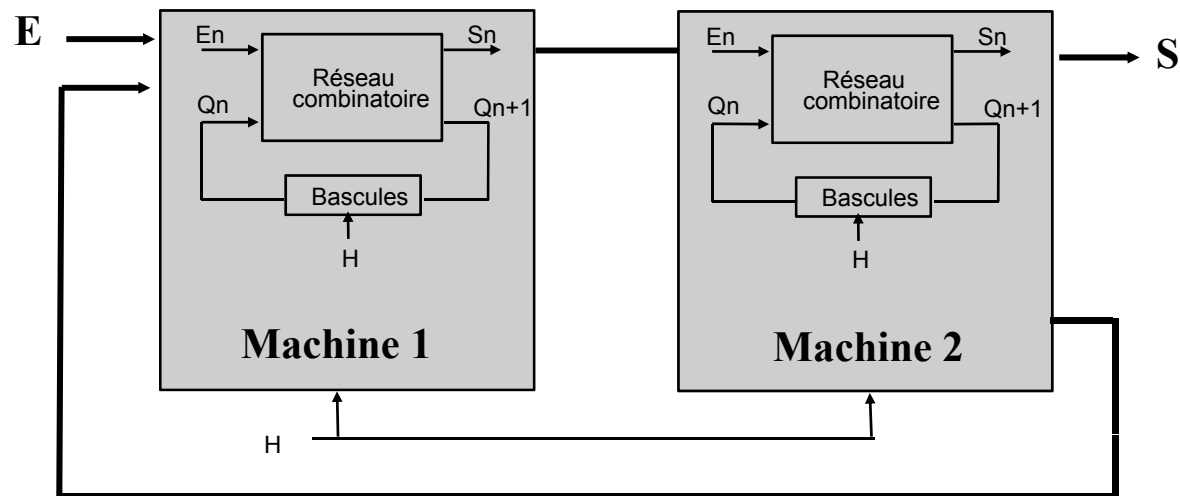
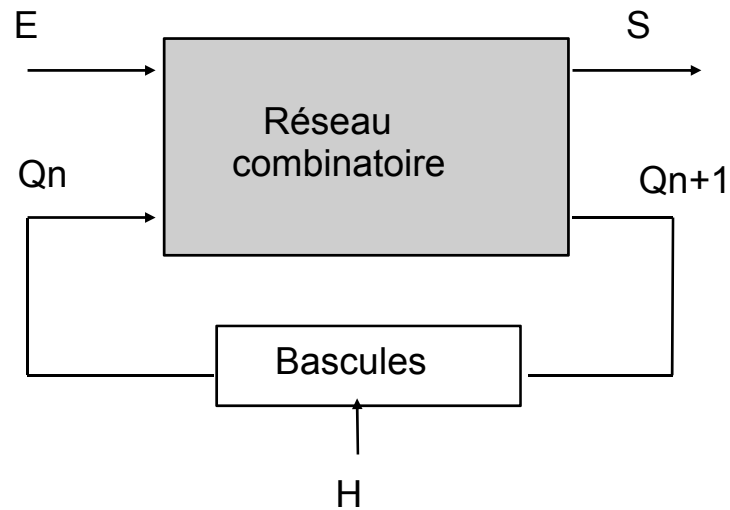
Pour $t4=0$



Pour $t4=1$

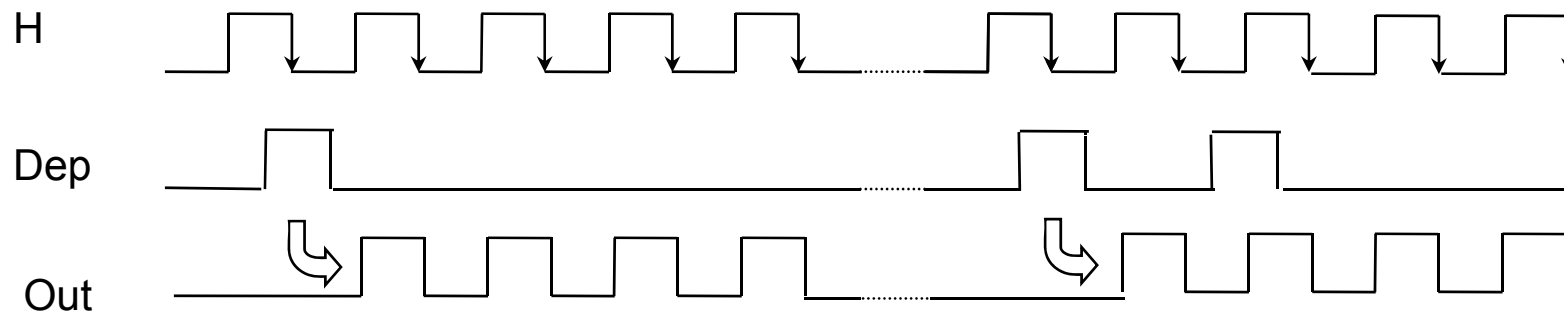


Partitionnement

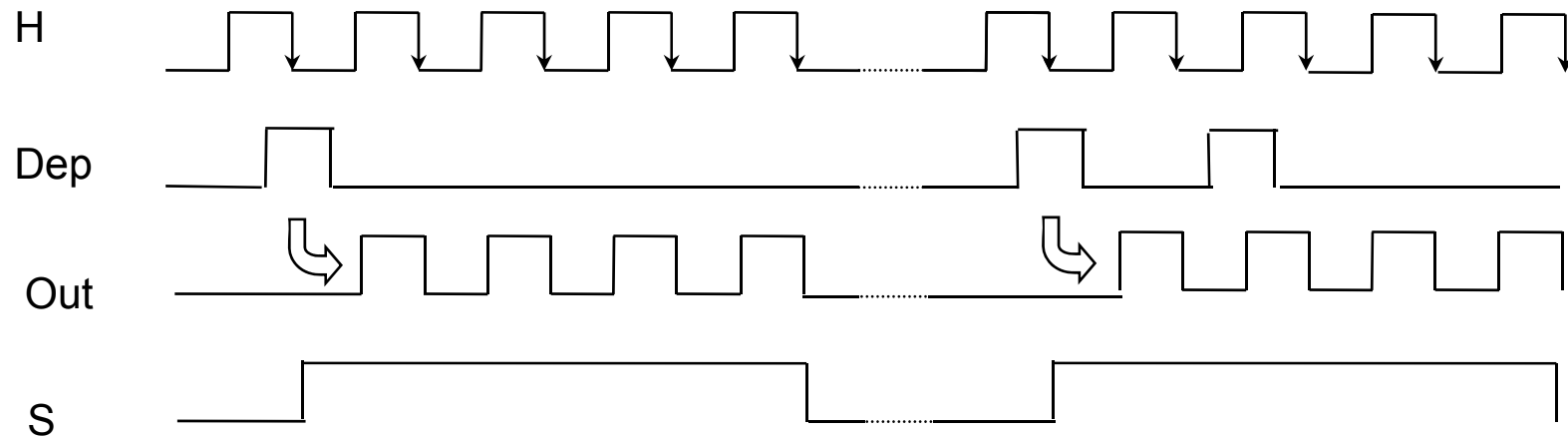
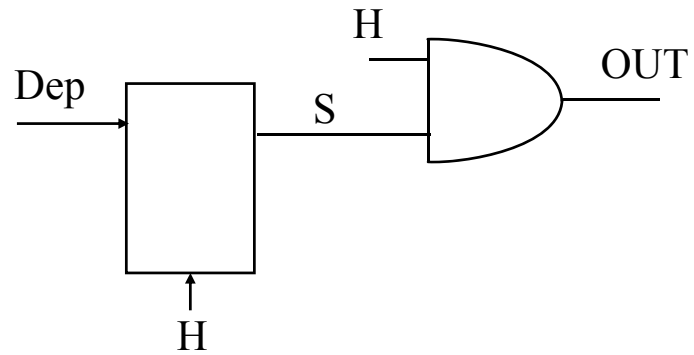


Partitionnement

Cahier des charges : On désire concevoir un système produisant un train de n impulsions à partir d'un signal Dep

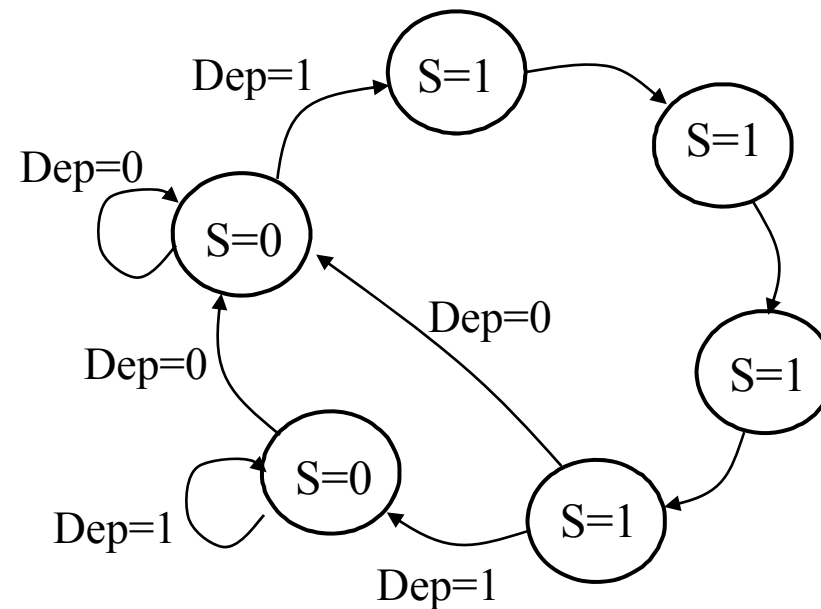
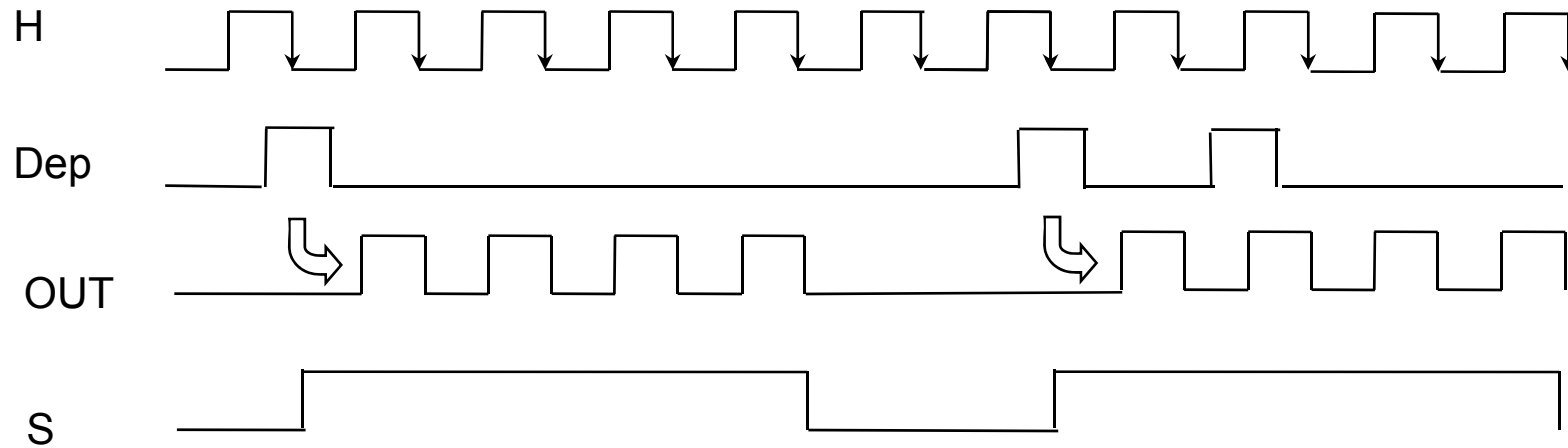


Partitionnement



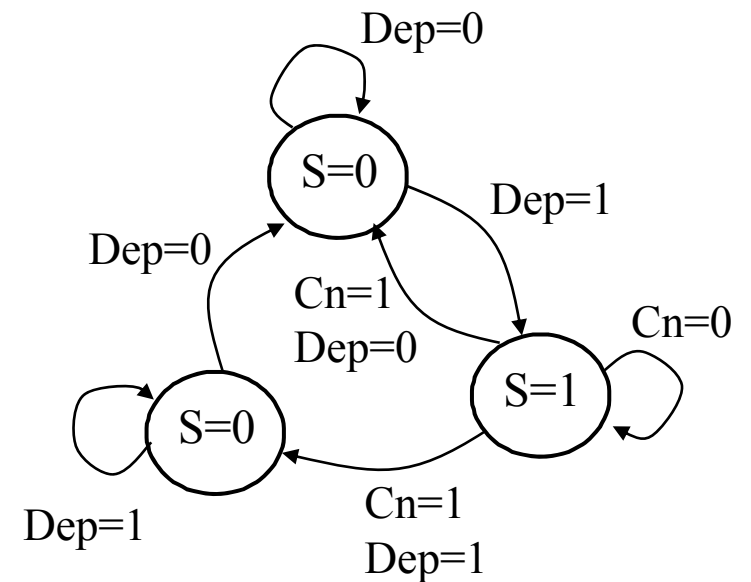
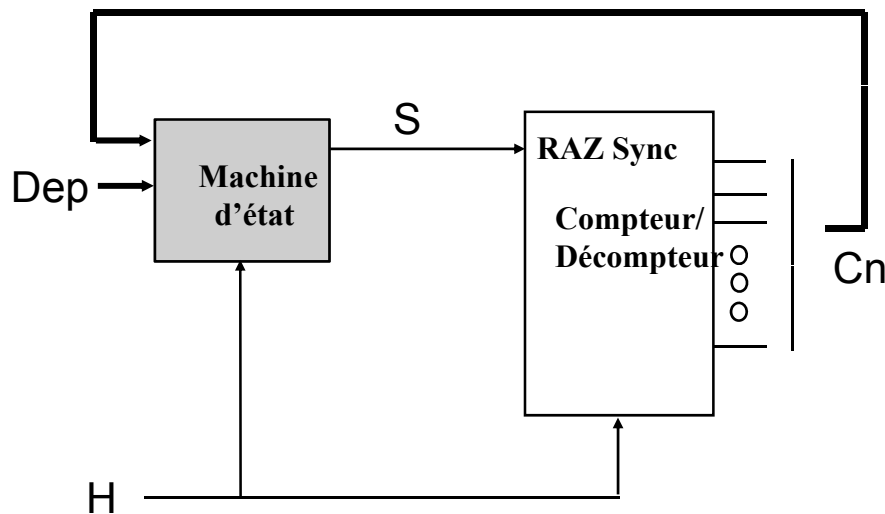
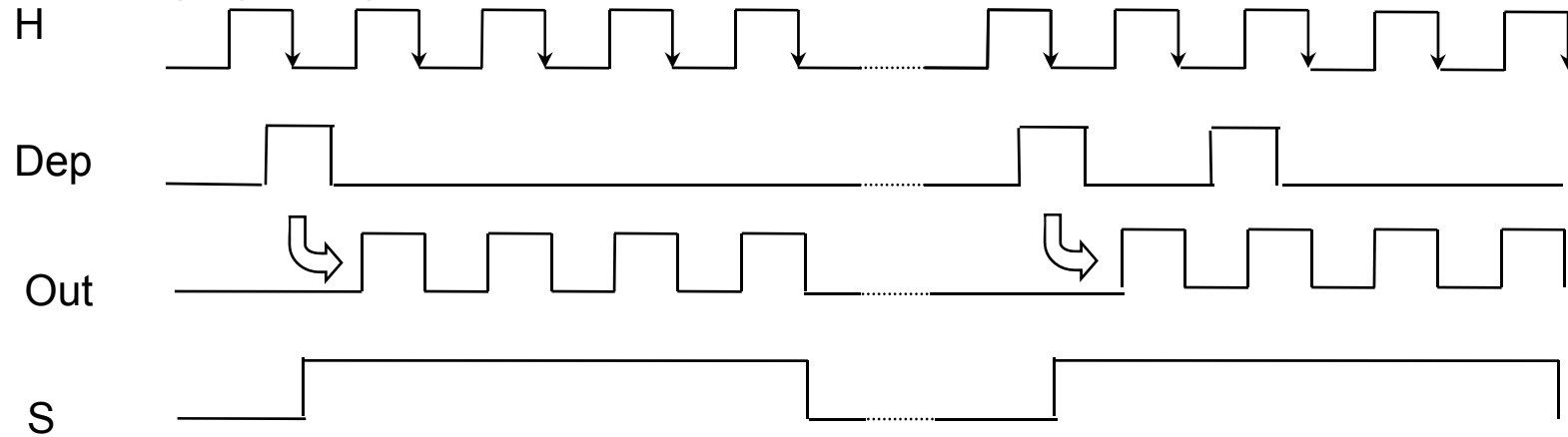
Partitionnement

Solution 1 (n petit)



Partitionnement

Solution 2 (n grand)

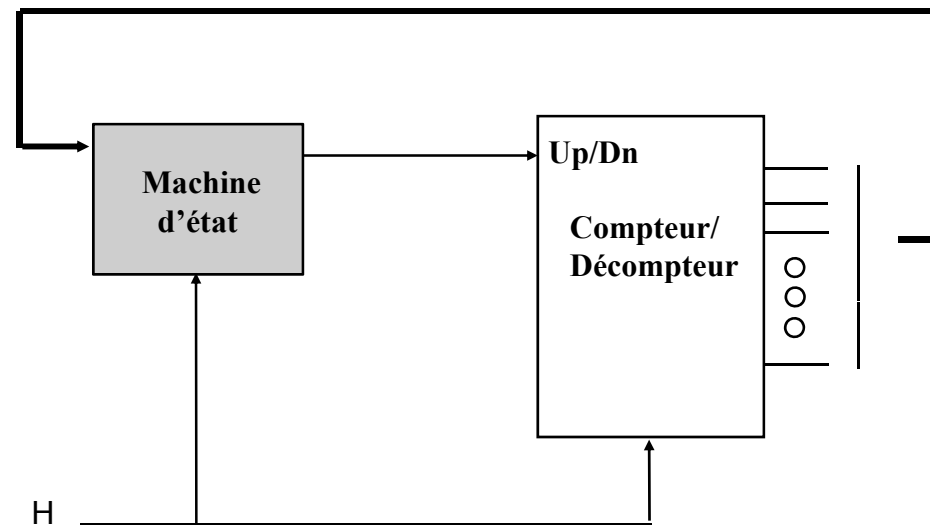


Partitionnement

Cahier des charges : On désire concevoir un système produisant la séquence suivante : 0, 1, 2, 3, ..., n-1, n, n-1, n-2, ..., 3, 2, 1, 0, 1, 2, 3, ...n.

Séquenceur $\Rightarrow 2*n$ Etats

Autre solution : Système à base de Compteur/décompteur



Partitionnement

