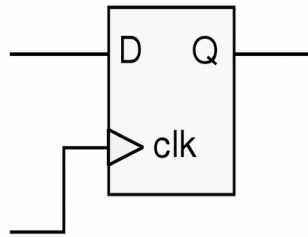
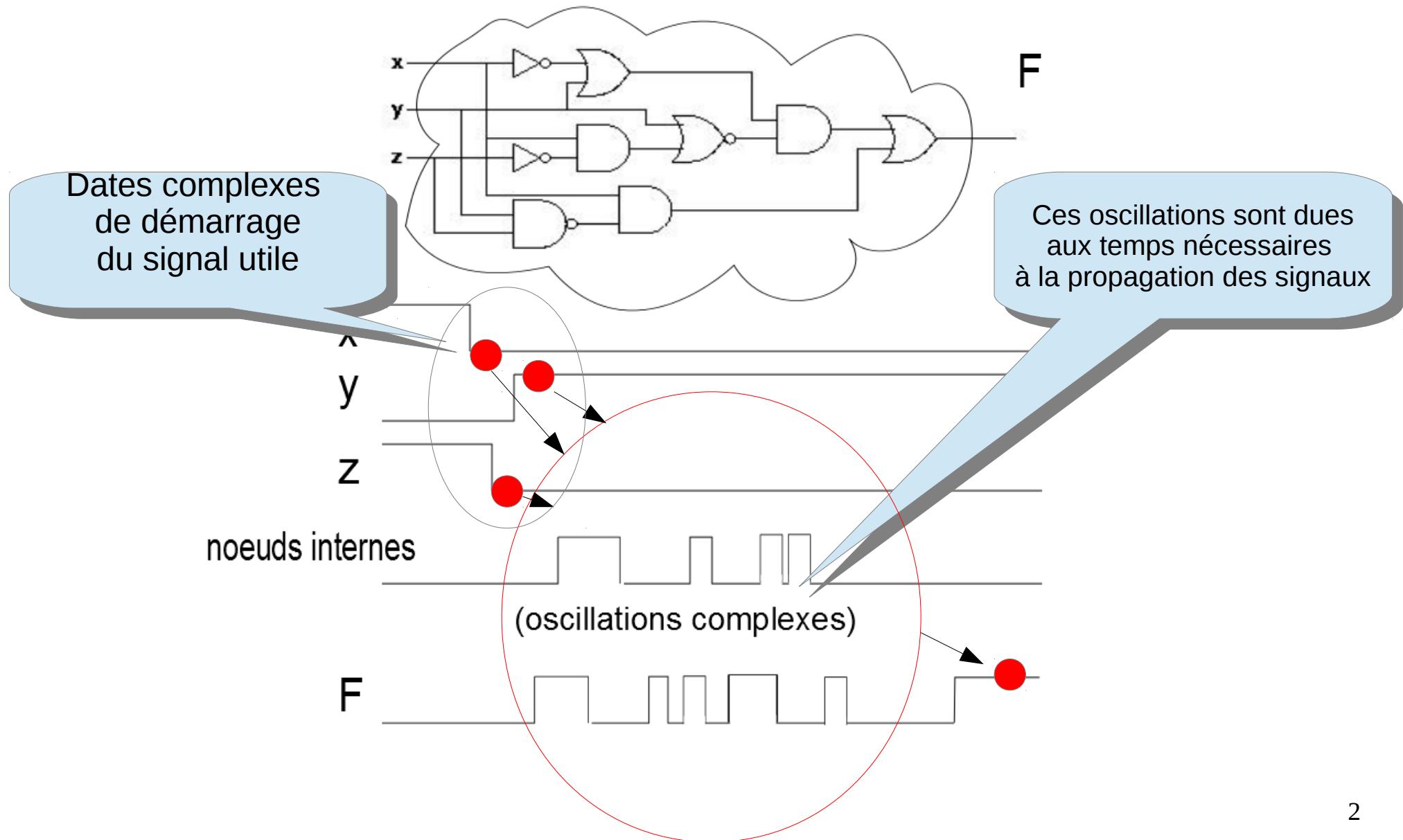


Logique séquentielle



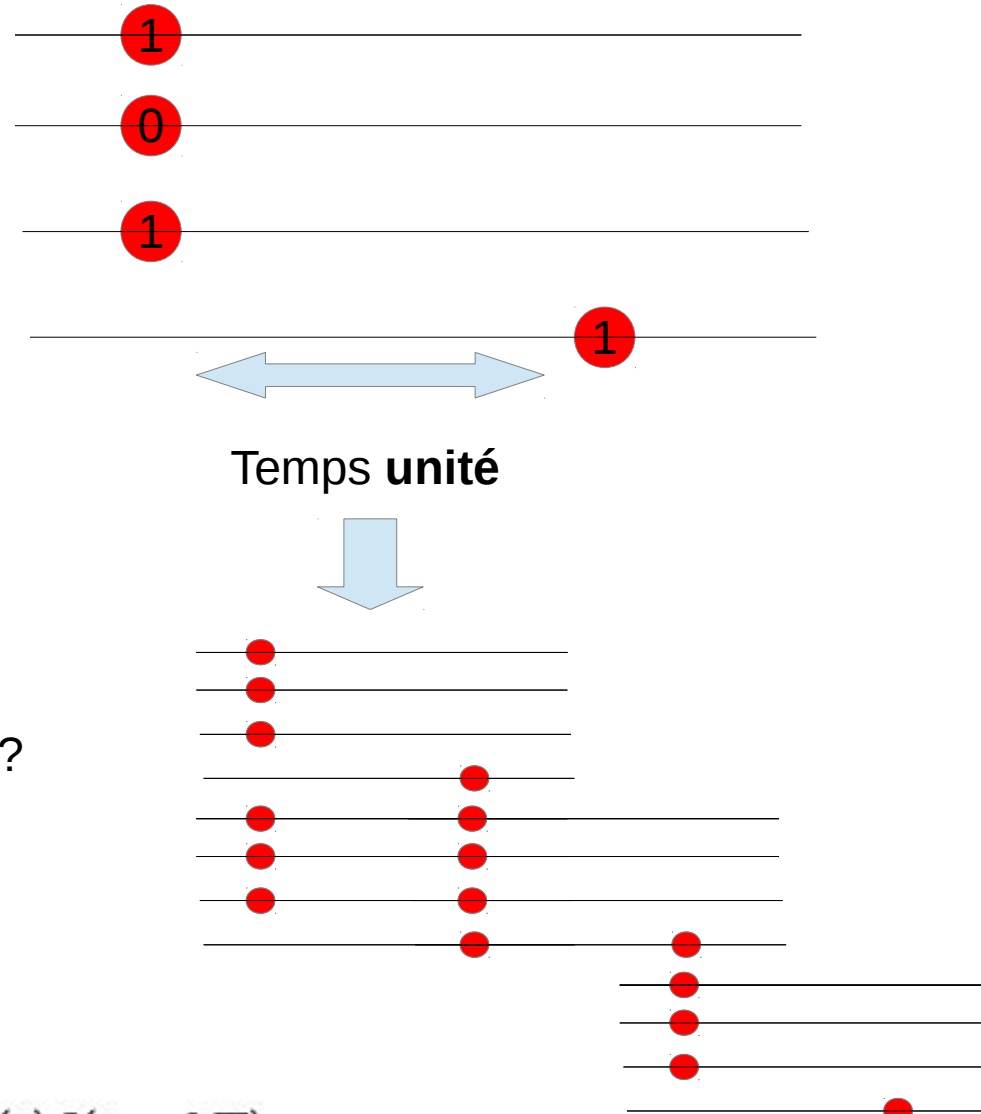
Le problème du combinatoire



Signaux idéaux



Comment assurer ceci ?



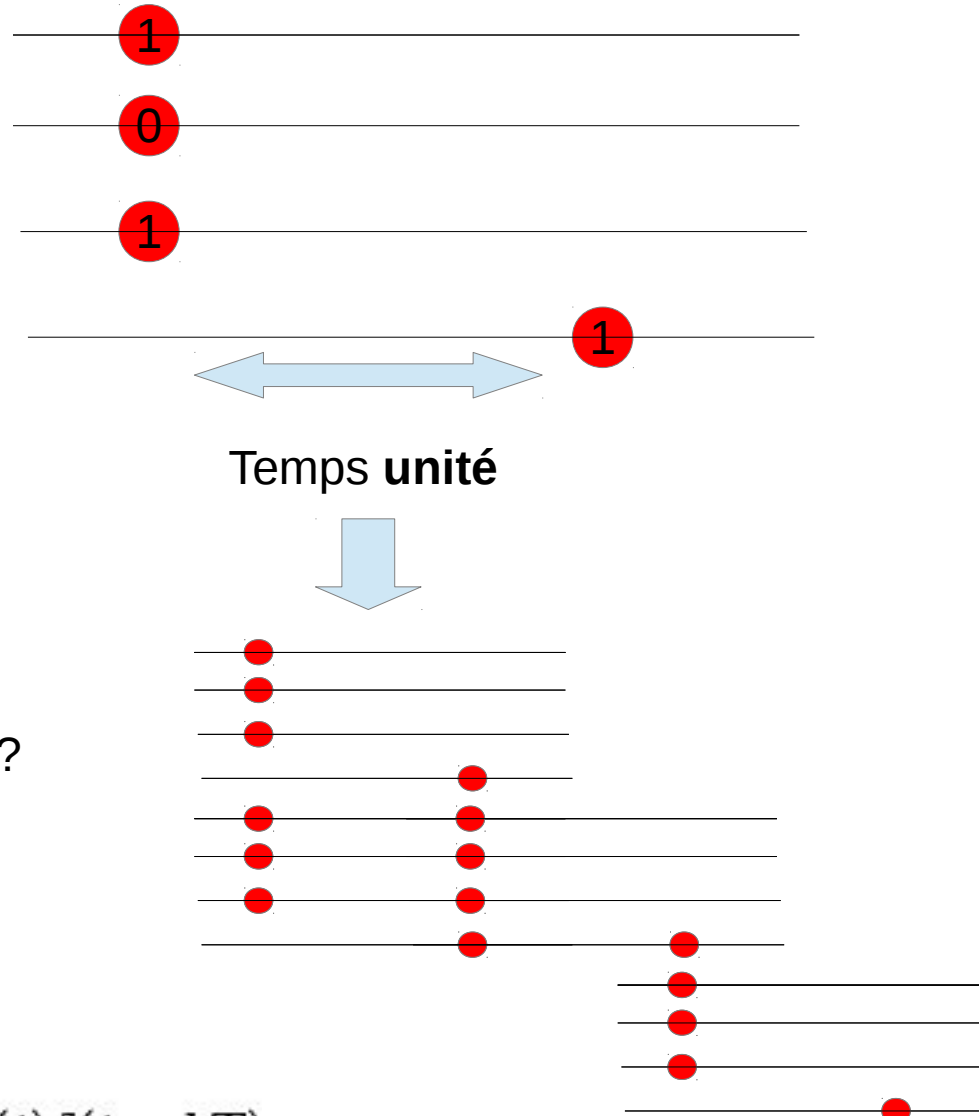
$$x^*[k] = x(kT) = \sum_{k=0}^{\infty} x(t)\delta(t - kT)$$

UV 1.5

Signaux idéaux



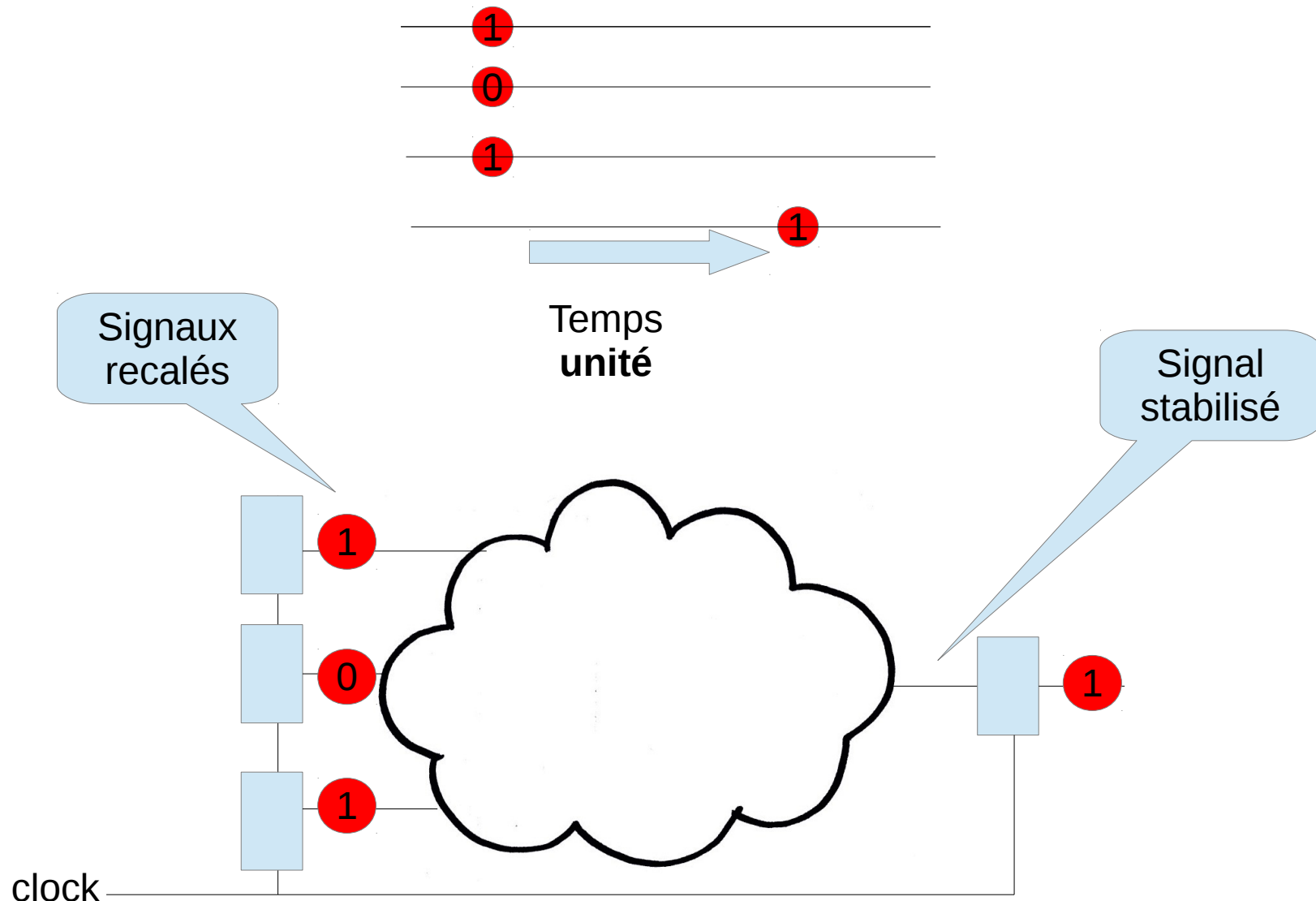
Comment assurer ceci ?



$$x^*[k] = x(kT) = \sum_{k=0}^{\infty} x(t)\delta(t - kT)$$

UV 1.5

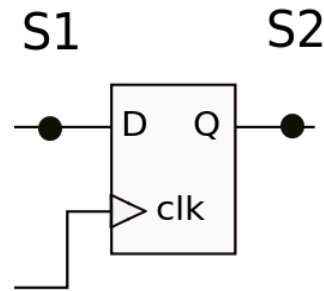
Solution : la bascule D !



On s'autorise à observer les signaux uniquement en sortie de bascule
(ou – ce qui revient au même – après un temps de stabilisation égal à la période d'horloge)

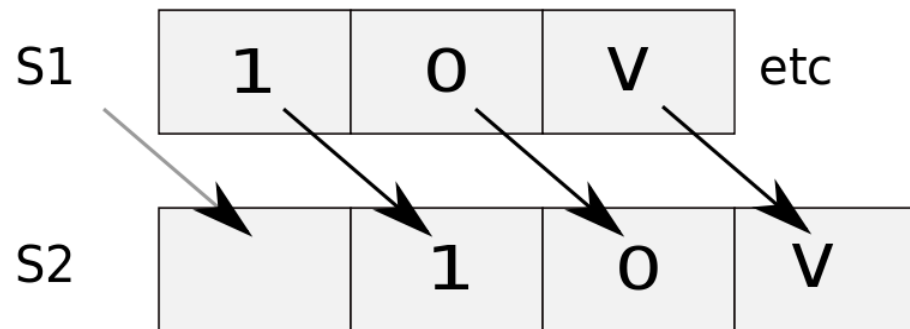
Echantillonnage par bascule D

Vue macroscopique



Aussi appelé :

- Edge triggered flip-flop,
- Flip-flop,
- Registre
- Bascule D



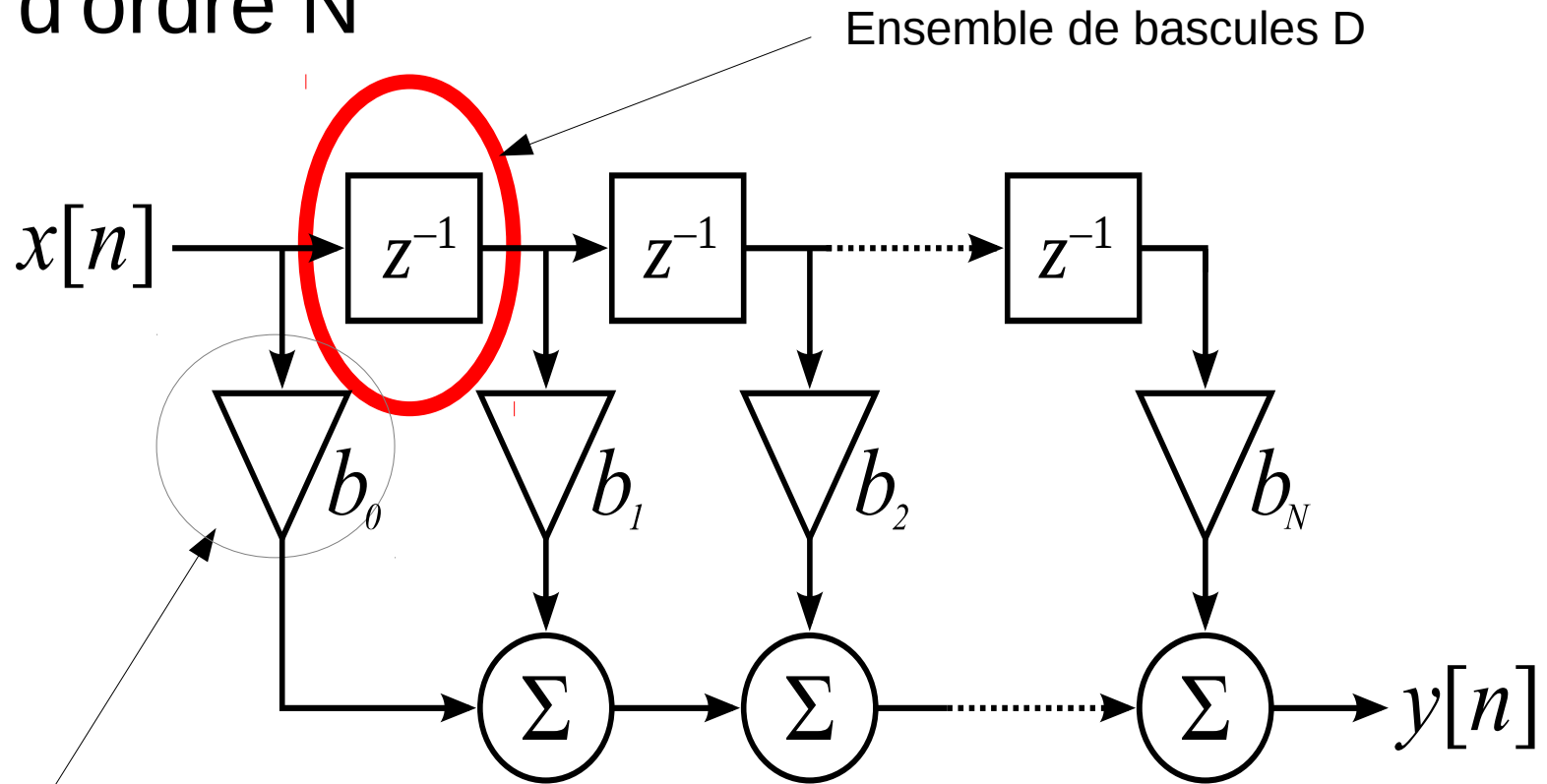
Analogies :

- suites numériques indicées par le numéro du tick d'horloge
- au Z de la transformée en Z

Exemple d'utilisation

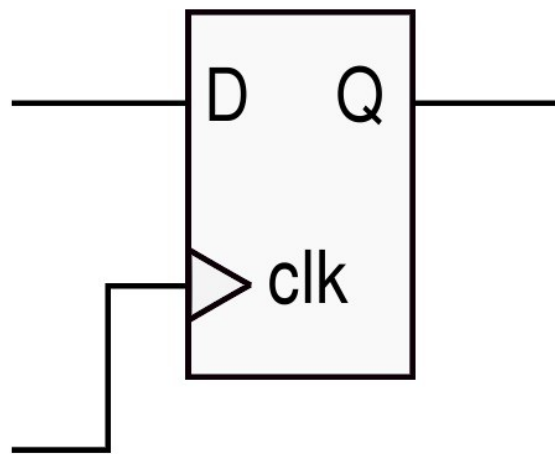
en traitement du signal



- FIR d'ordre N



(Notation maladroite ici)
Il s'agit d'une Multiplication par une constante

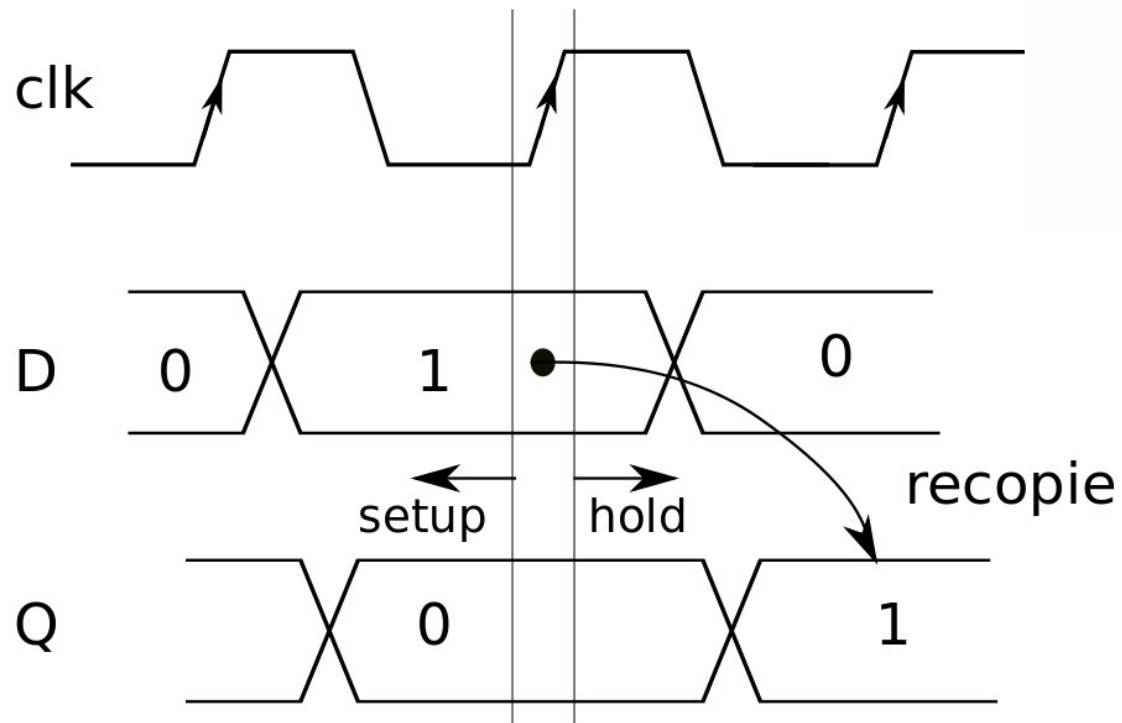
Table de vérité de la bascule D



D	clk	Q_{n+1}
0		0
1		1
X	0 / 1	Q_n

Echantillonnage

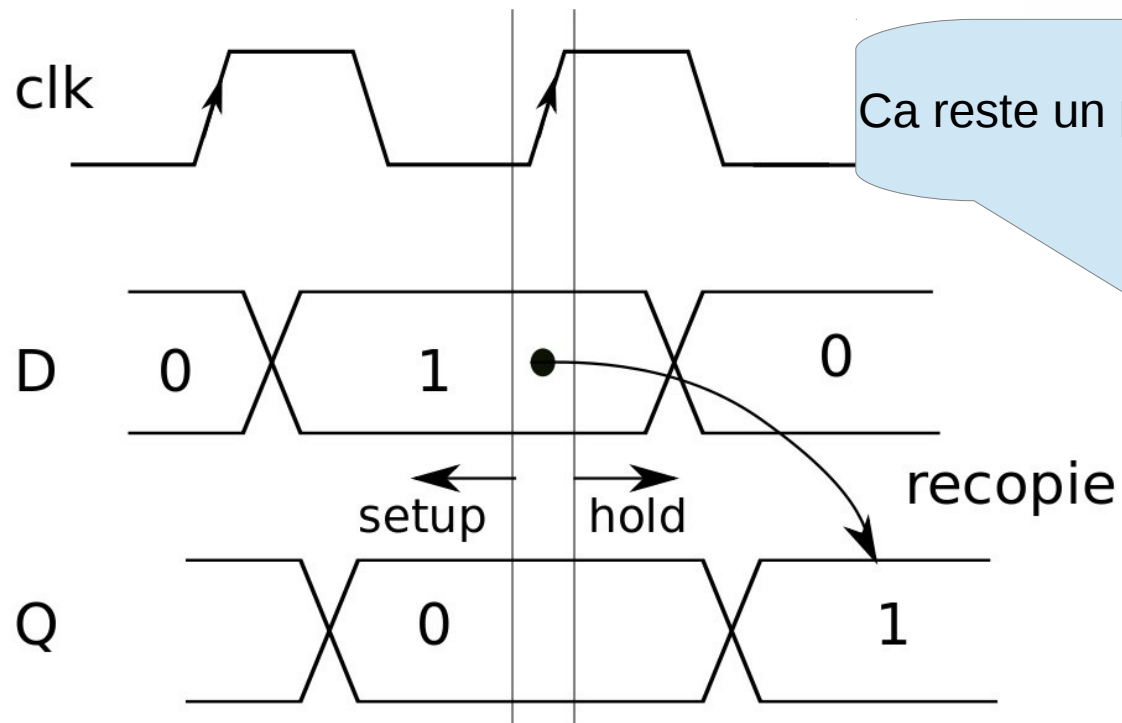
Vue microscopique



Le signal à échantillonner doit être respecter certaines contraintes imposées par l'horloge :
Le signal doit être stable avant et après le front d'échantillonnage

Echantillonnage

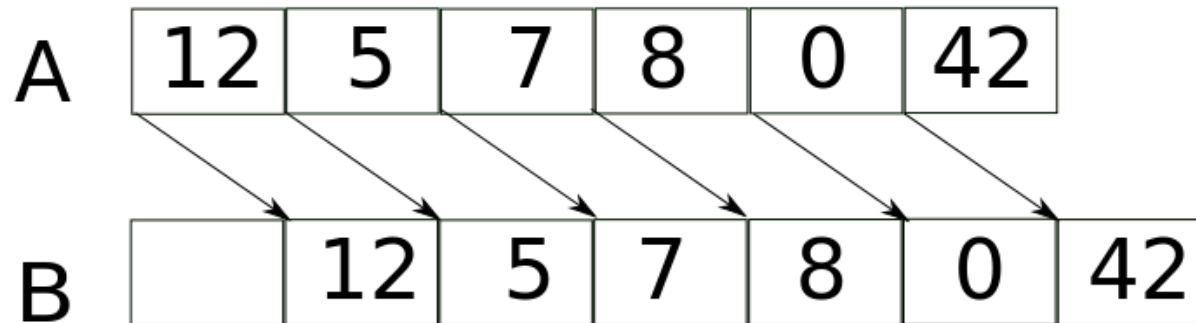
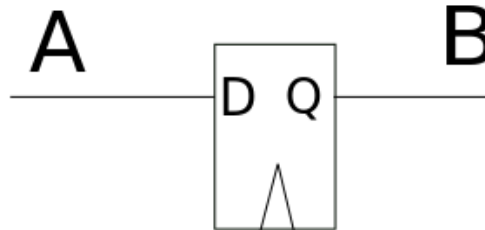
Vue microscopique



Le signal à échantillonner doit être respecter certaines contraintes imposées par l'horloge :
Le signal doit être stable avant et après le front d'échantillonnage

Echantillonnage

Vue macroscopique

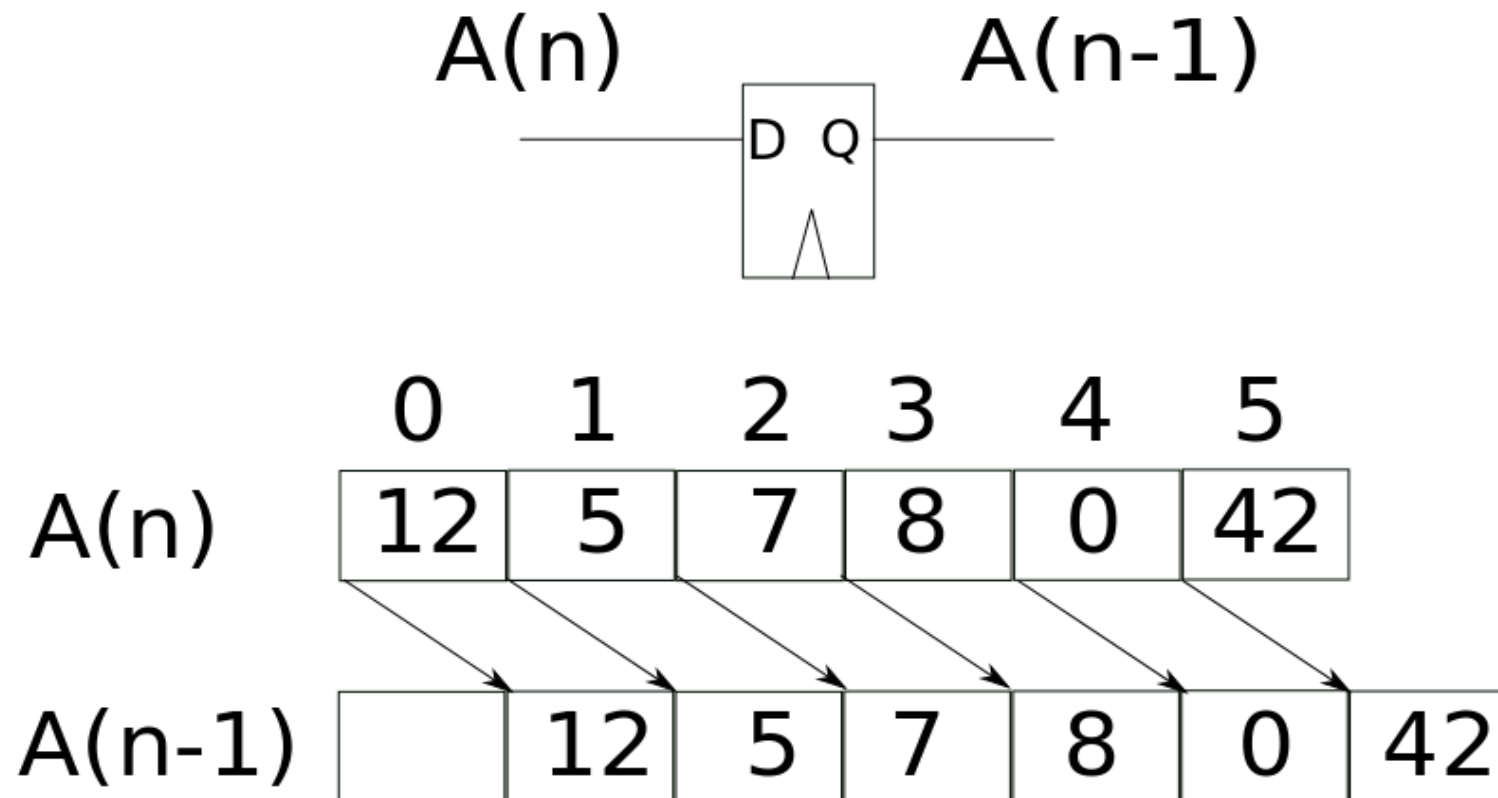


décalage temporel systématique



Vue macroscopique

explicitons l'indice temporel



l'indice n n'est jamais explicite
dans le circuit : ce sont les ticks d'horloge

Expérimentons cette notion d'indice temporel

Retour sur l'exo 3 du TE2

e1	:	0	0	1	1	0	0	0	0	1	1	0	0	1	0	0	0	0
e2	:	0	0	0	0	1	0	0	1	0	1	1	0	1	1	0	0	0
n1	:	0	0	1	1	1	0	0	1	1	1	1	0	1	1	0	0	0
s1	:	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
s2	:	0	0	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1



Retrouver un circuit possible qui se comporte
De cette manière !

Solution possible

e1	:	0	0	1	1	0	0	0	0	1	1	0	0	1	0	0	0	0
e2	:	0	0	0	0	1	0	0	1	0	1	1	0	1	1	0	0	0
n1	:	0	0	1	1	1	0	0	1	1	1	1	0	1	1	0	0	0
s1	:	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
s2	:	0	0	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1

solution :

On voit que :

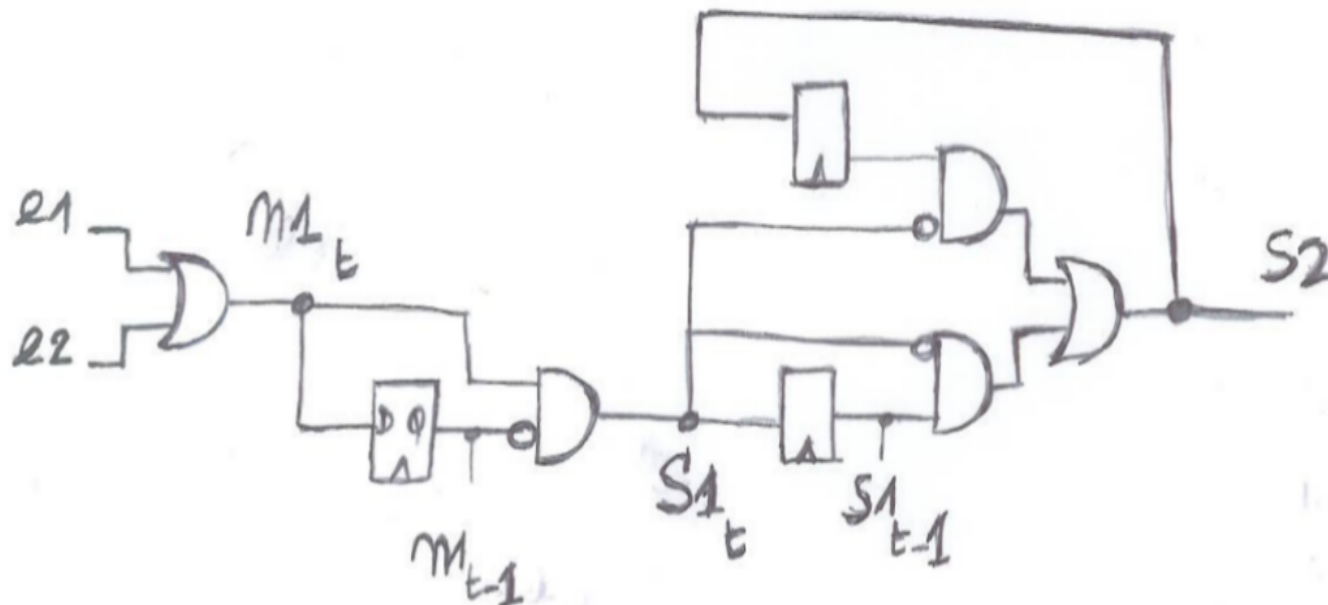
- $n1$ est le OU entre $e1$ et $e2$
- $s1$ marque le debut de la sequence des 1 de $n1$ ($\rightarrow n1_t.\overline{n1_{t-1}}$)
- $s2_t$ vaut 1 quand :
 - $s1_t$ vaut 0 ET $s1_{t-1}$ vaut 1 ($\rightarrow \overline{s1_t}.s1_{t-1}$)
 - OU $s2_{t-1}$ valait déjà 1, sauf dans le cas où $s1_t$ vaut 1 (auquel cas, $s2_t$ vaut 0) ($\rightarrow s2_{t-1}.\overline{s1_t}$)
- donc $s2_t = \overline{s1_t}.s1_{t-1} + s2_{t-1}.\overline{s1_t}$

Solution possible

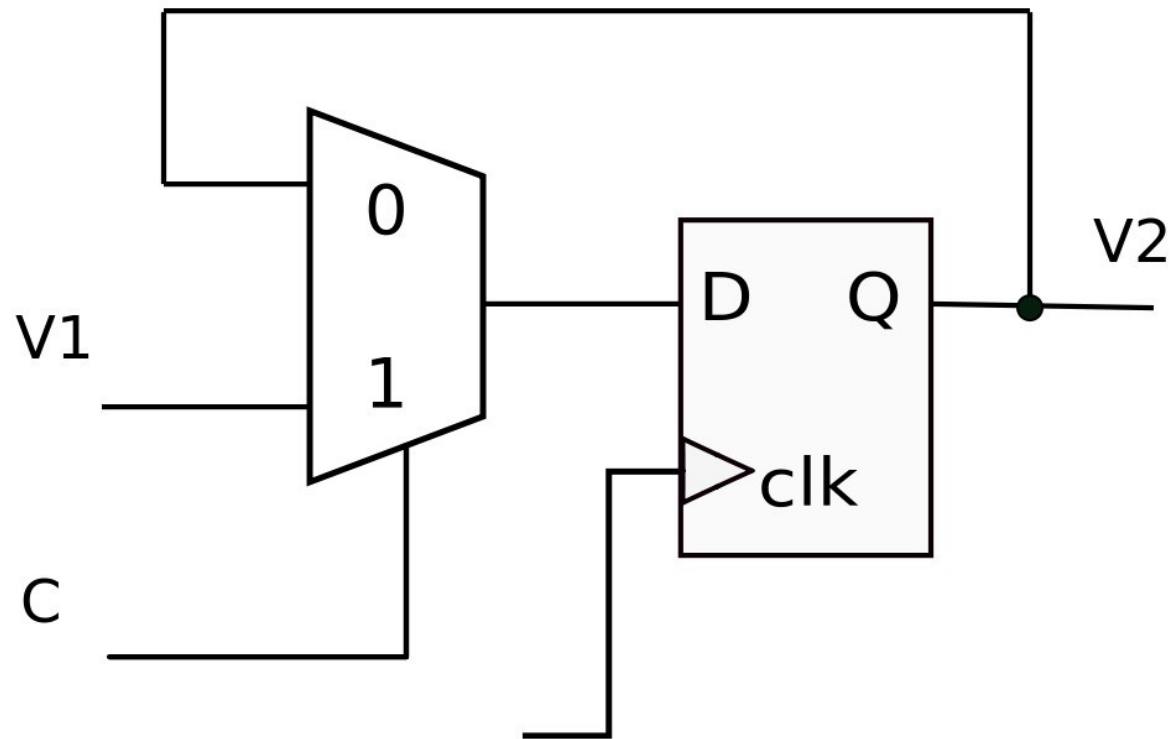
solution :

On voit que :

- $n1$ est le OU entre $e1$ et $e2$
- $s1$ marque le debut de la sequence des 1 de $n1$ ($\rightarrow n1_t \cdot \overline{n1_{t-1}}$)
- $s2_t$ vaut 1 quand :
 - $s1_t$ vaut 0 ET $s1_{t-1}$ vaut 1 ($\rightarrow \overline{s1_t} \cdot s1_{t-1}$)
 - OU $s2_{t-1}$ valait déjà 1, sauf dans le cas où $s1_t$ vaut 1 (auquel cas, $s2_t$ vaut 0) ($\rightarrow s2_{t-1} \cdot \overline{s1_t}$)
- donc $s2_t = \overline{s1_t} \cdot s1_{t-1} + s2_{t-1} \cdot \overline{s1_t}$

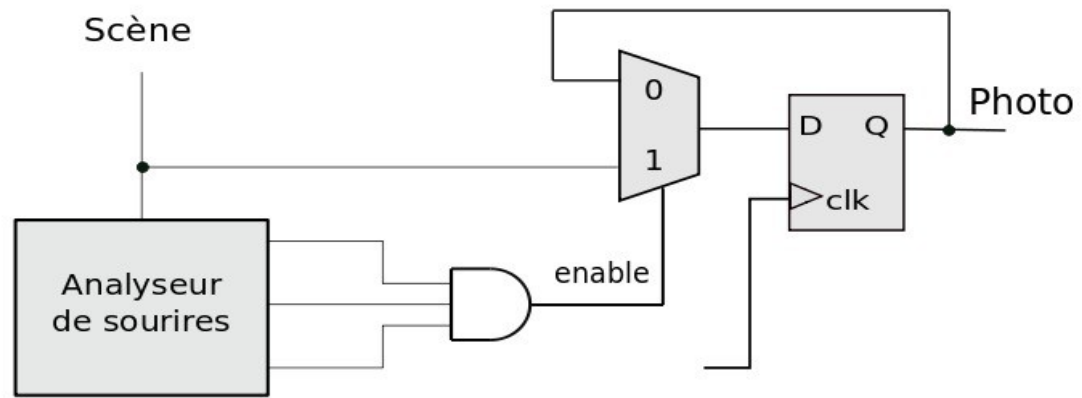


Echantillonnage contrôlé par enable

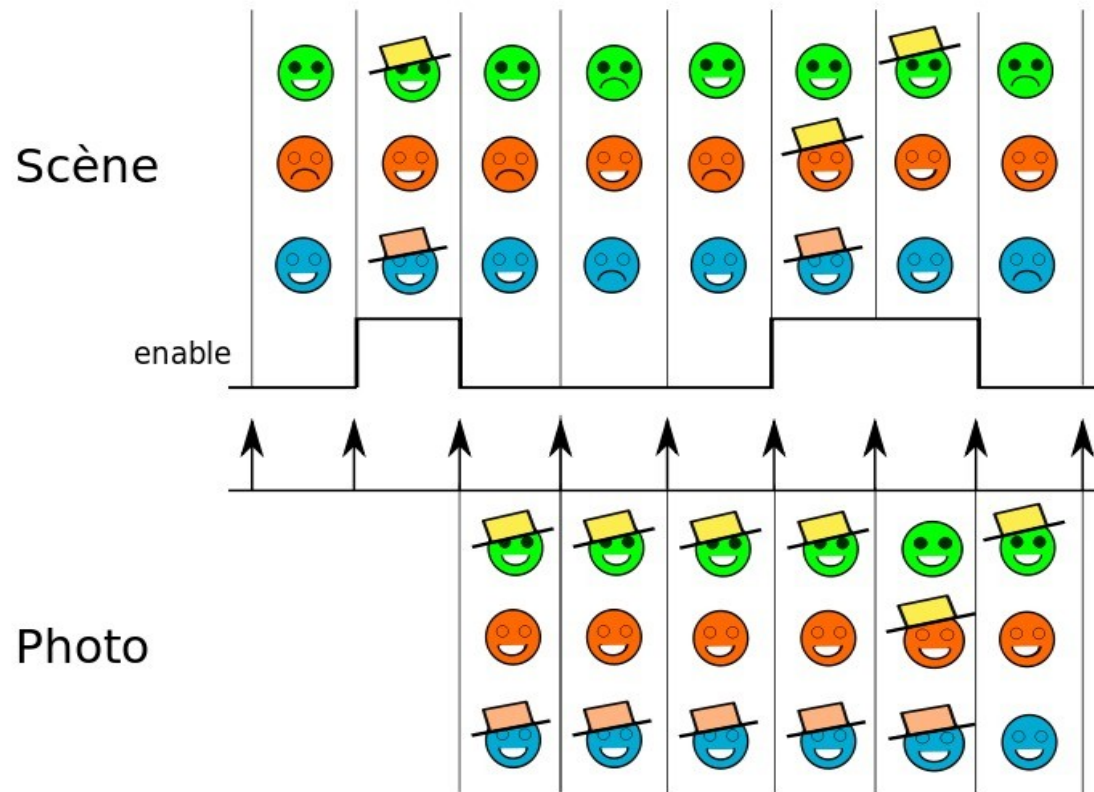


C'est l'utilisation la plus classique
If (condition) {
 V2 = v1 ;
}

UV 1.5



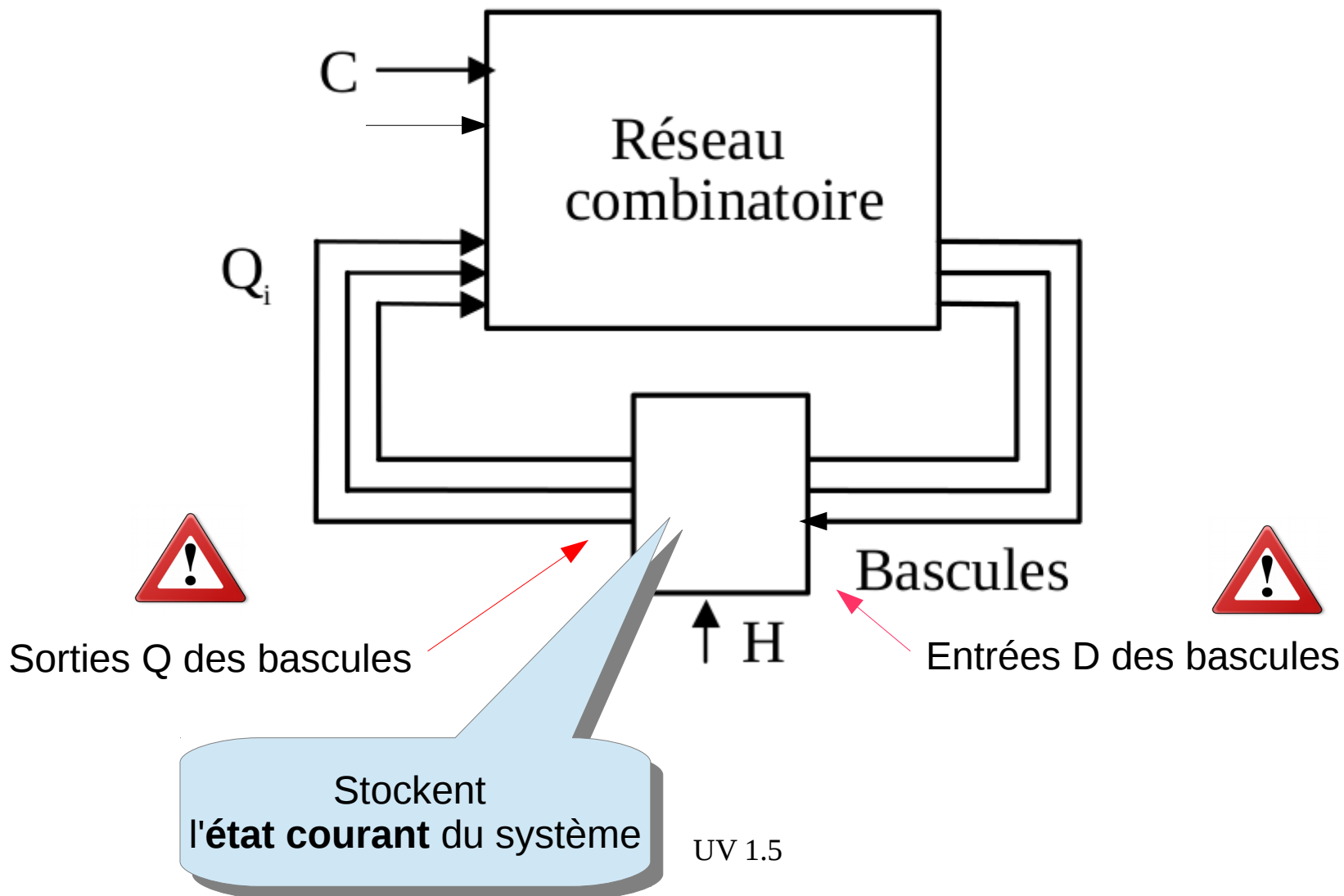
Comb.



Fonctionnement en régime permanent

- Les bascule D effectuent une recopie **systematique** de l'entrée sur la sortie, lors d'un court temps d'échantillonnage
 - Ce temps est quasi-instantané : temps de montée de l'horloge (plus rarement : descente)
 - Le signal recopié ne pourra être vu par une autre bascule qu'au coup d'horloge suivant
- Le régime permanent correspond au fonctionnement normal du circuit...

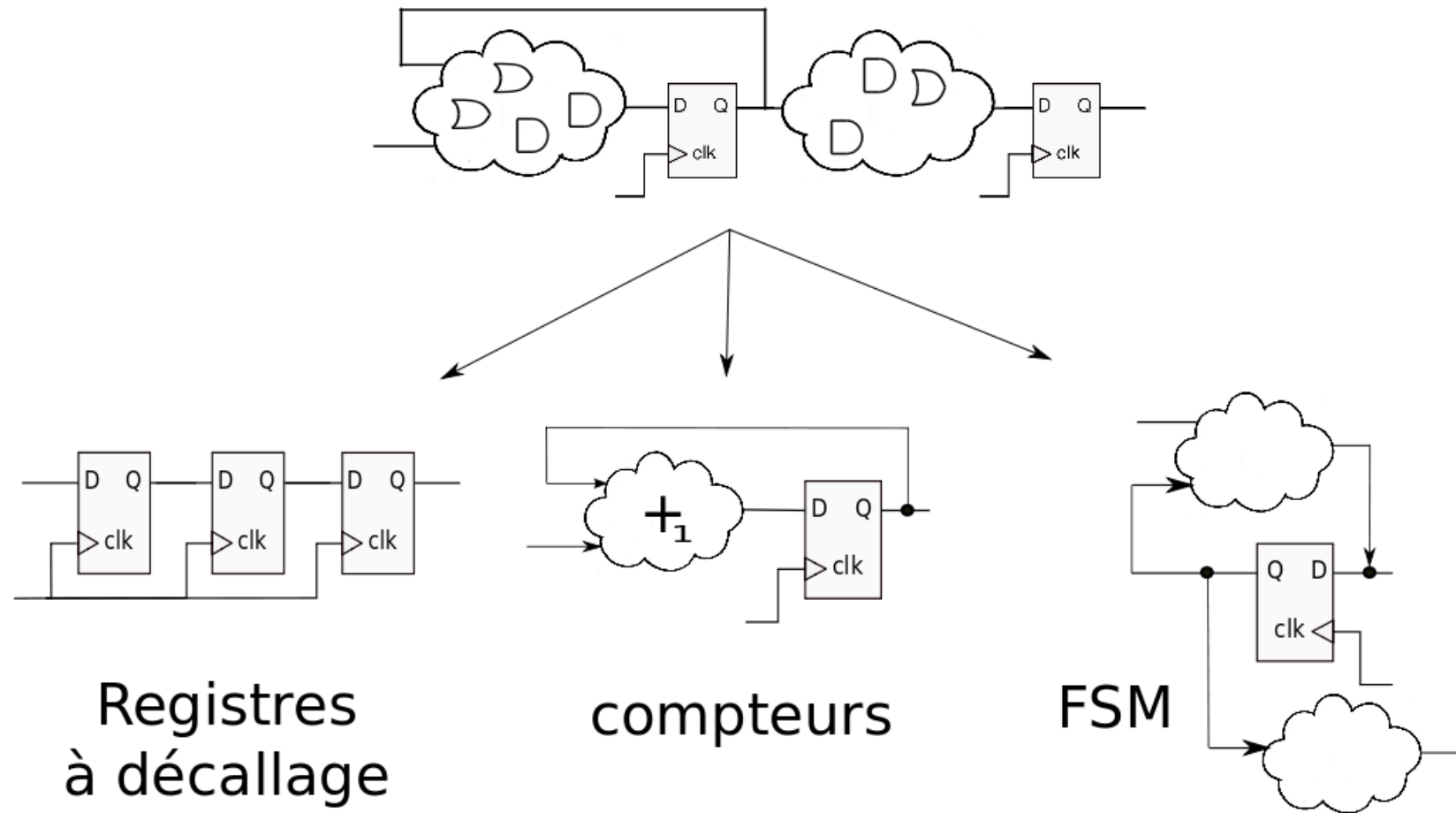
Combinatoire et séquentiel



Un schéma générique

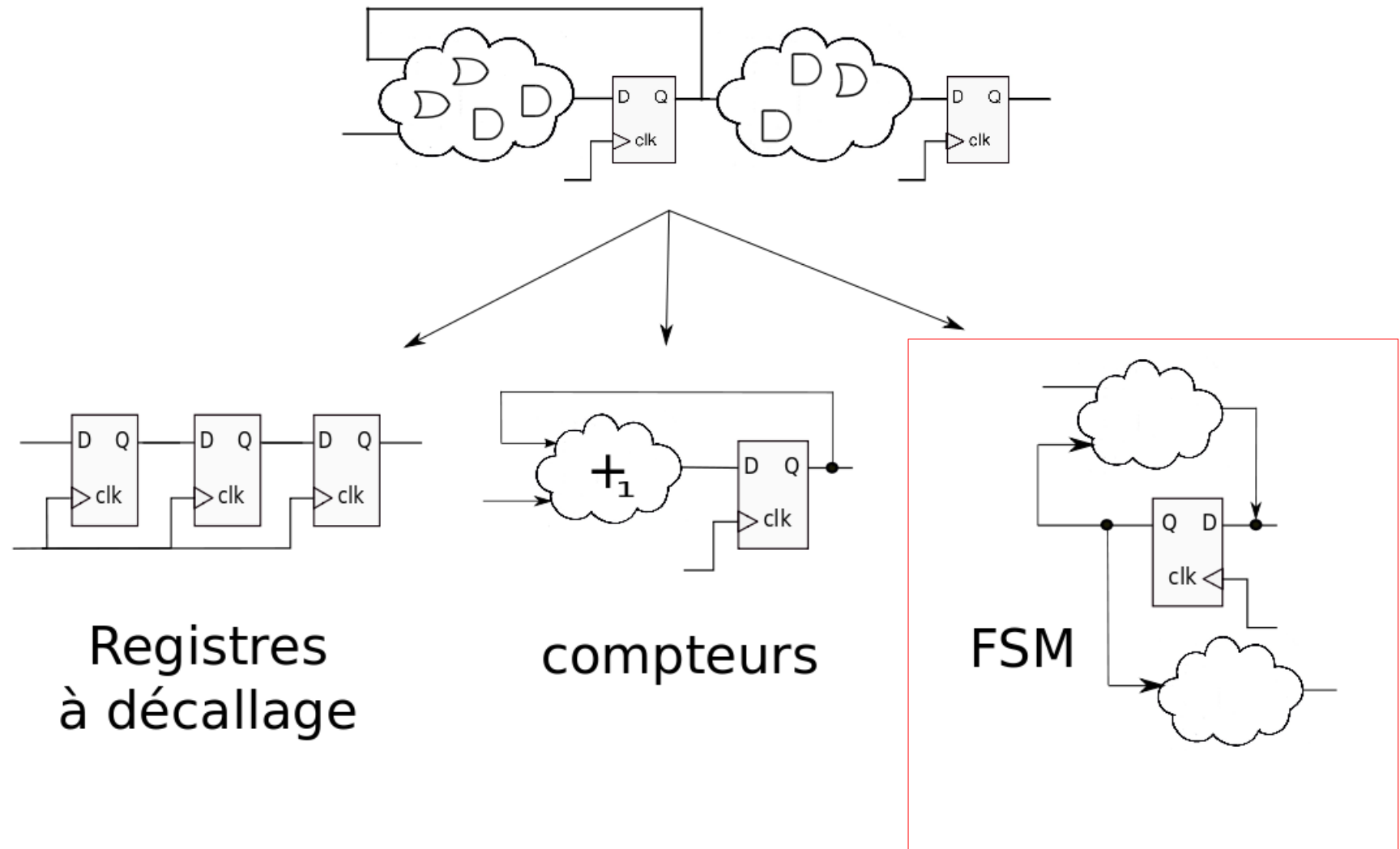
- Ce seul schéma théorique peut représenter n'importe quel système numérique
 - L'état évolue en fonction de ses entrées et de son passé
- En pratique, le concepteur préfère distribuer l'état du système dans plusieurs tels « petits » schémas
 - Exemple dans un slide suivant
- On va l'appliquer à différents cas :
 - Opérateurs séquentiels : $ex \rightarrow$ compteur
 - **Automates** d'états finis ou « **machine d'états finis** »
 - **FSM** en anglais : finite state machine
 - Ce concept est fondamental !!! Il n'y en aura pas beaucoup d'autres !

Dérivation du schéma générique



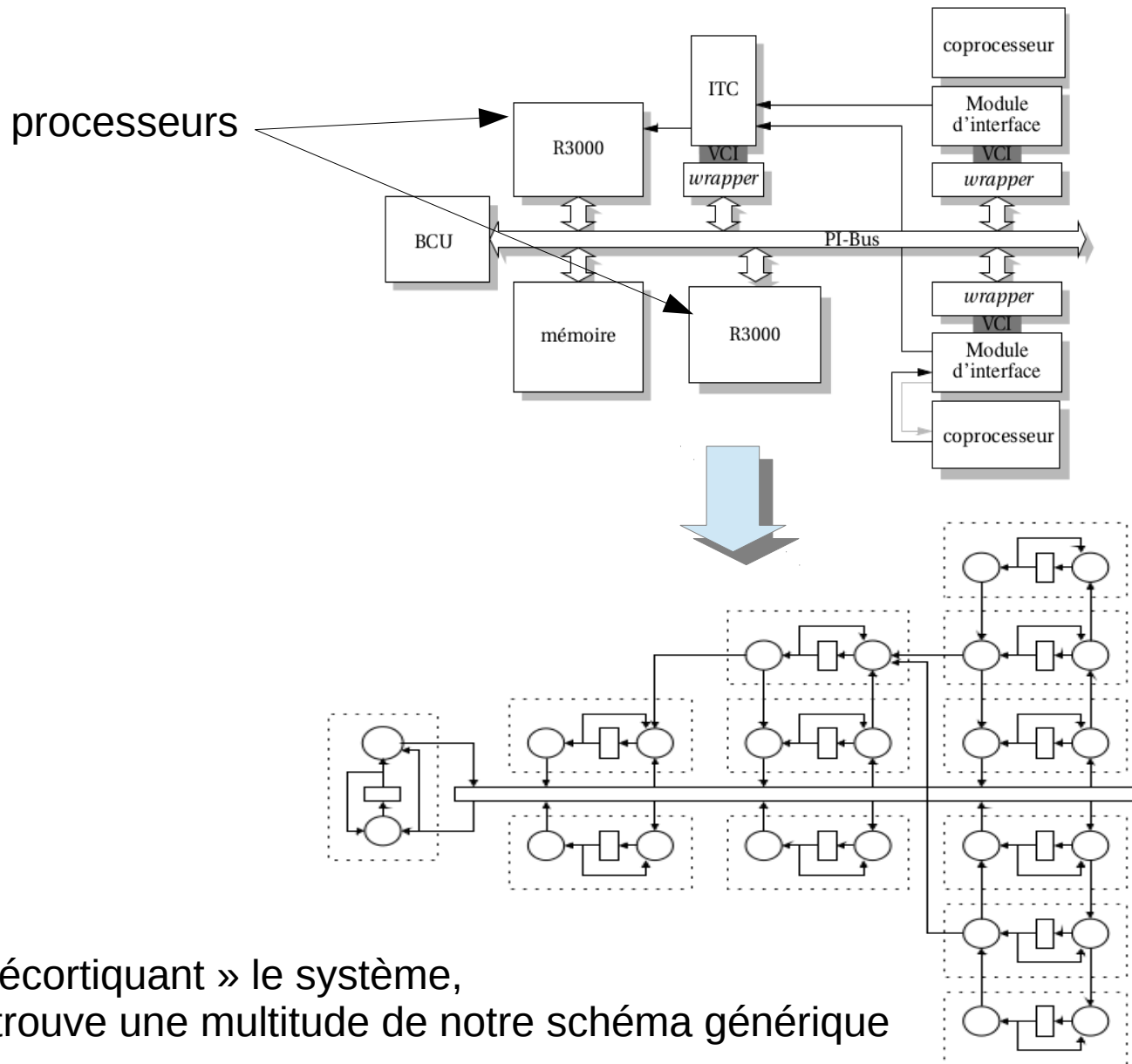
On va désormais s'intéresser aux FSM

Dérivation du schéma générique



On va désormais s'intéresser aux FSM

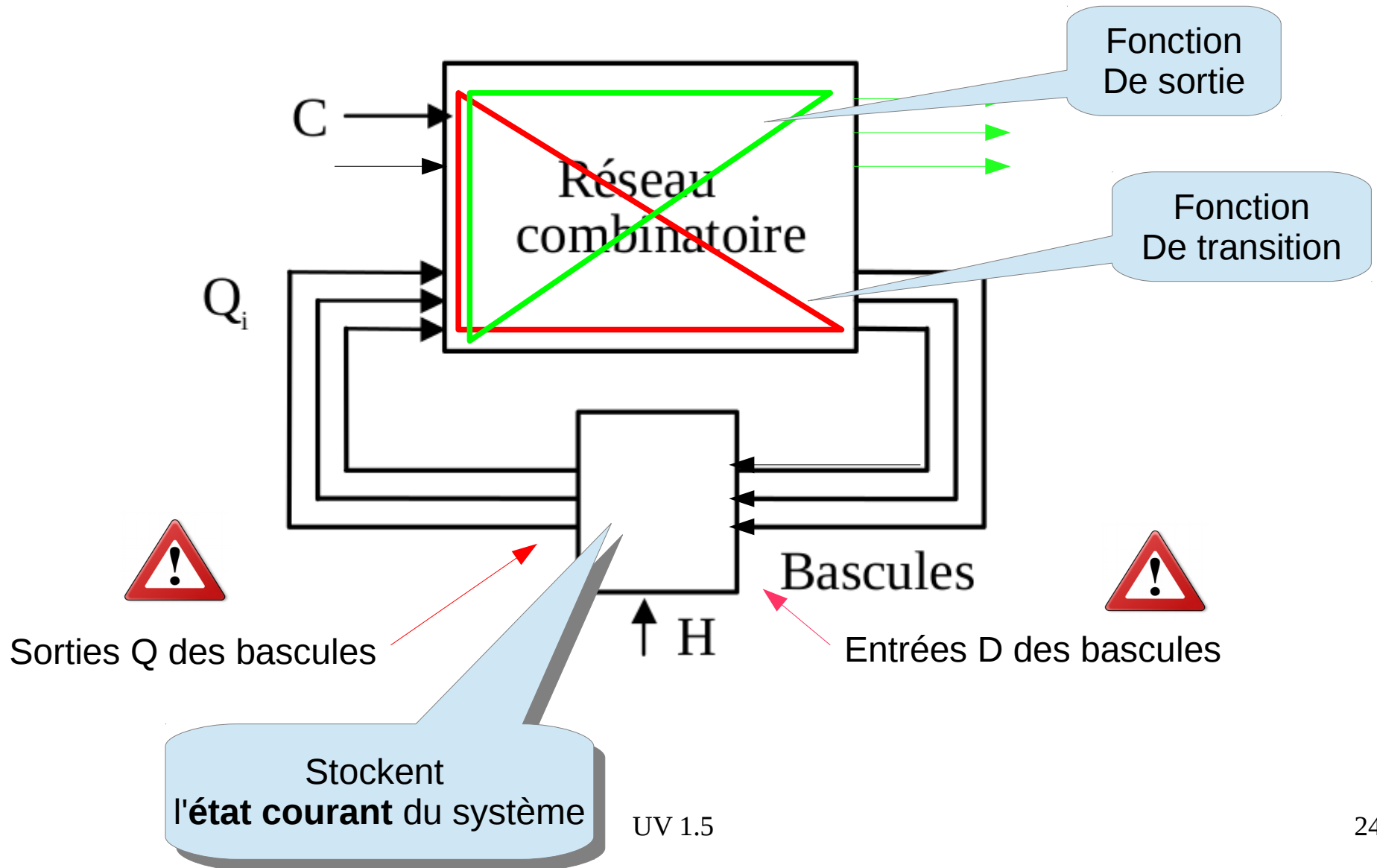
Exemple de systèmes numériques complexes « distribués » sur puce (SoC)



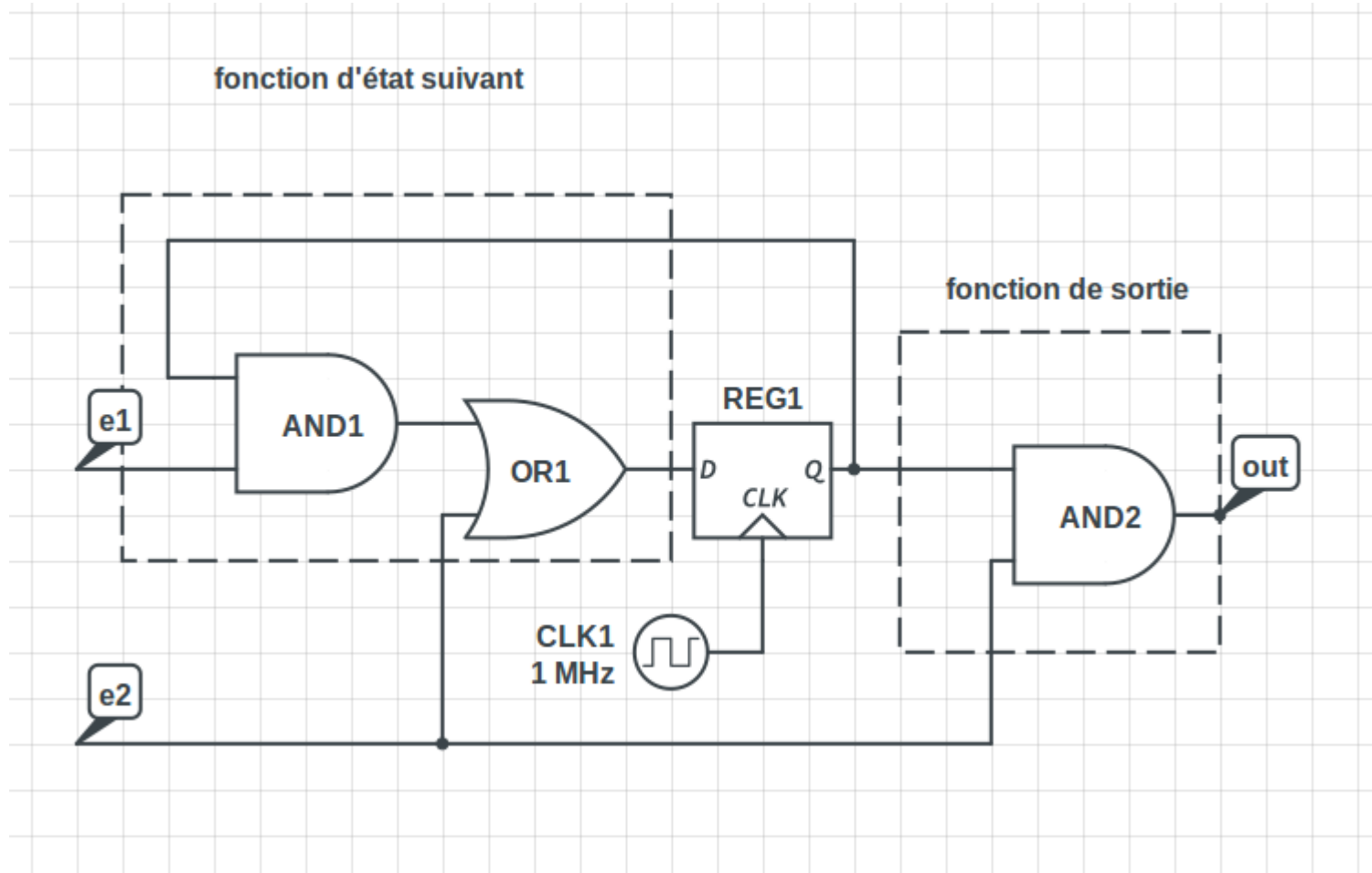
SoC
System-on-Chip

En « décortiquant » le système,
On ne trouve une multitude de notre schéma générique

Combinatoire et séquentiel



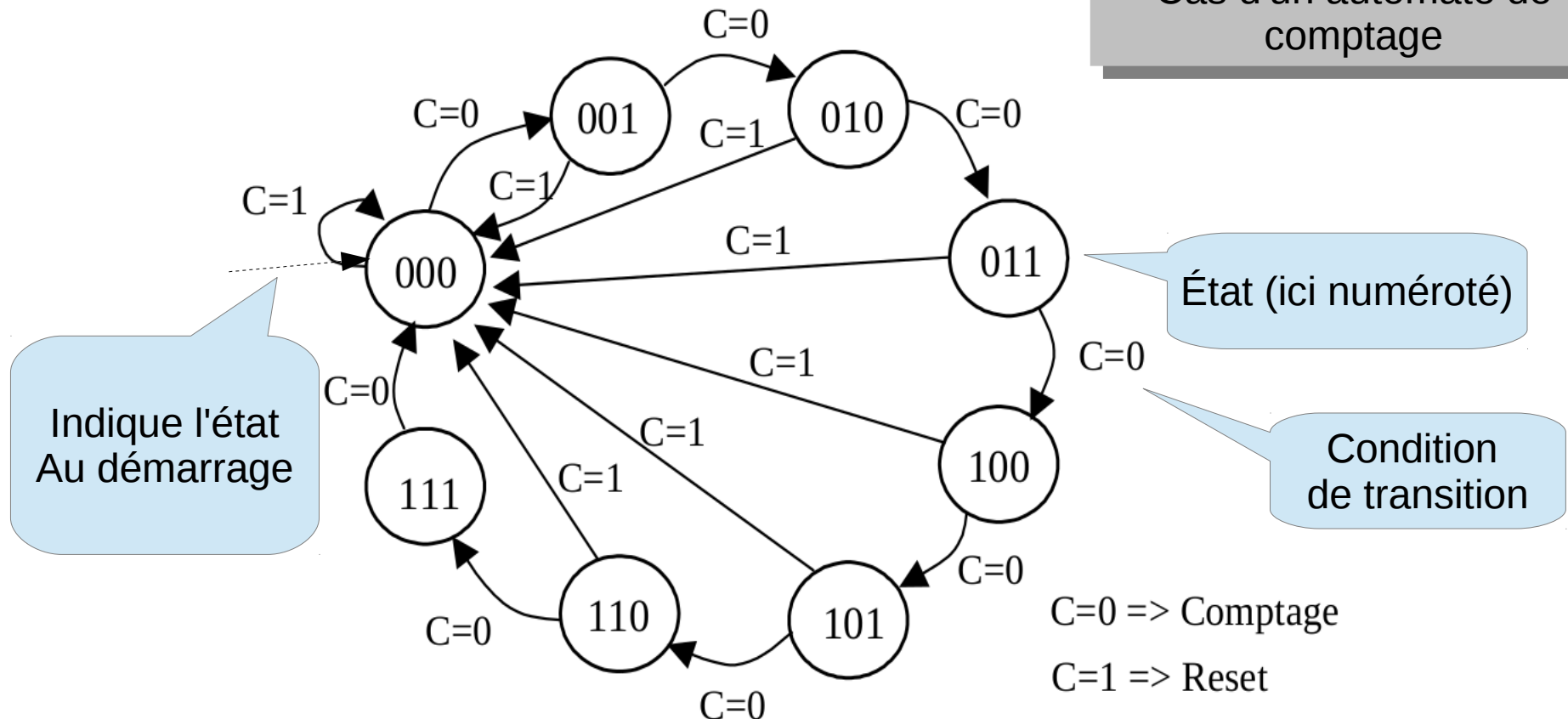
Exemple minimaliste



État codé ici sur 1 bit => 1 bascule

Représentation « diagramme à bulle »

Cas d'un automate de comptage



Représentation **intuitive** de l'évolution de l'état du système

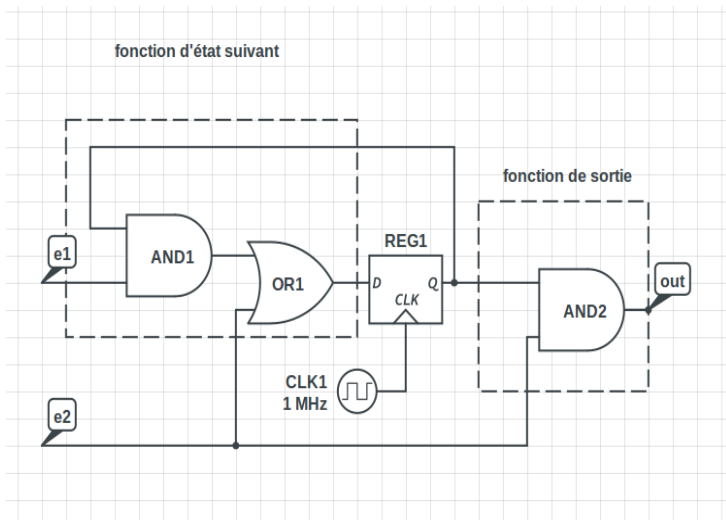


Trouver les équations logiques

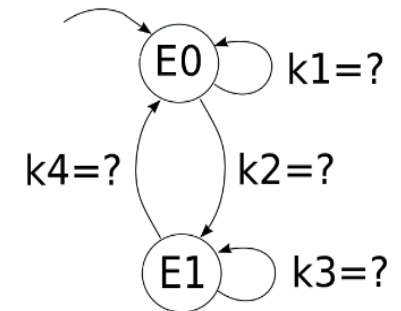
Du circuit au diagramme à bulles ?

Retour sur l'exo 5 du TE2

Le circuit ne possède qu'une seule bascule D donc un seul bit d'état, ce qui va beaucoup nous aider. On écrit la table de vérité de la fonction de transition $f_{trans}(e_1, e_2, Q)$:



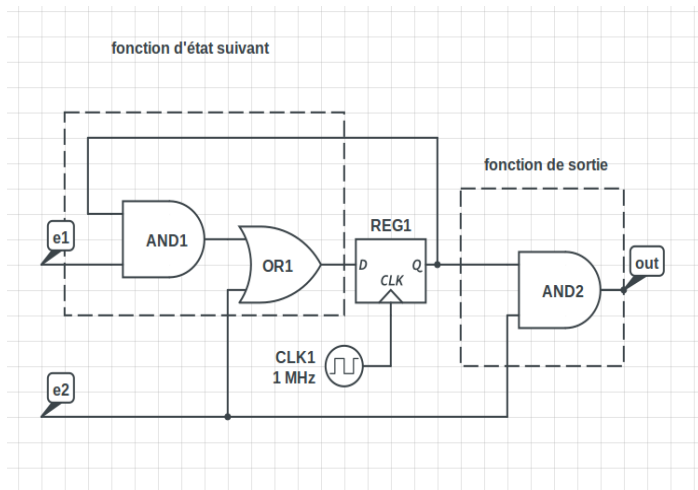
e1	e2	Q	D	ligne
0	0	0	0	0
0	0	1	0	1
0	1	0	1	2
0	1	1	1	3
1	0	0	0	4
1	0	1	1	5
1	1	0	1	6
1	1	1	1	7



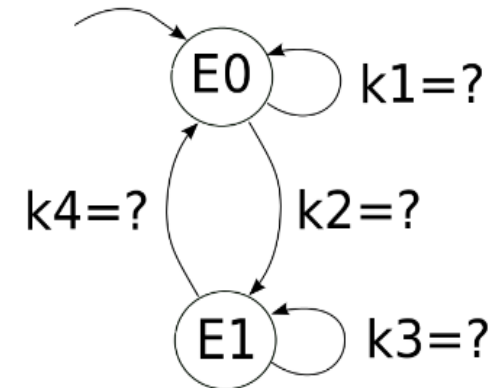
Appelons E cette variable d'état. L'état courant de E est matérialisé par Q et sont état futur par D. Comment évolue cette variable au cours du temps ? On peut dessiner le diagramme à bulle suivant (ou diagramme d'état), et faire donc apparaître 4 conditions booléennes possibles, appelées k_1, k_2, k_3, k_4 , relatives aux transitions.

Du circuit au diagramme à bulles ?

Retour sur l'exo 5 du TE2



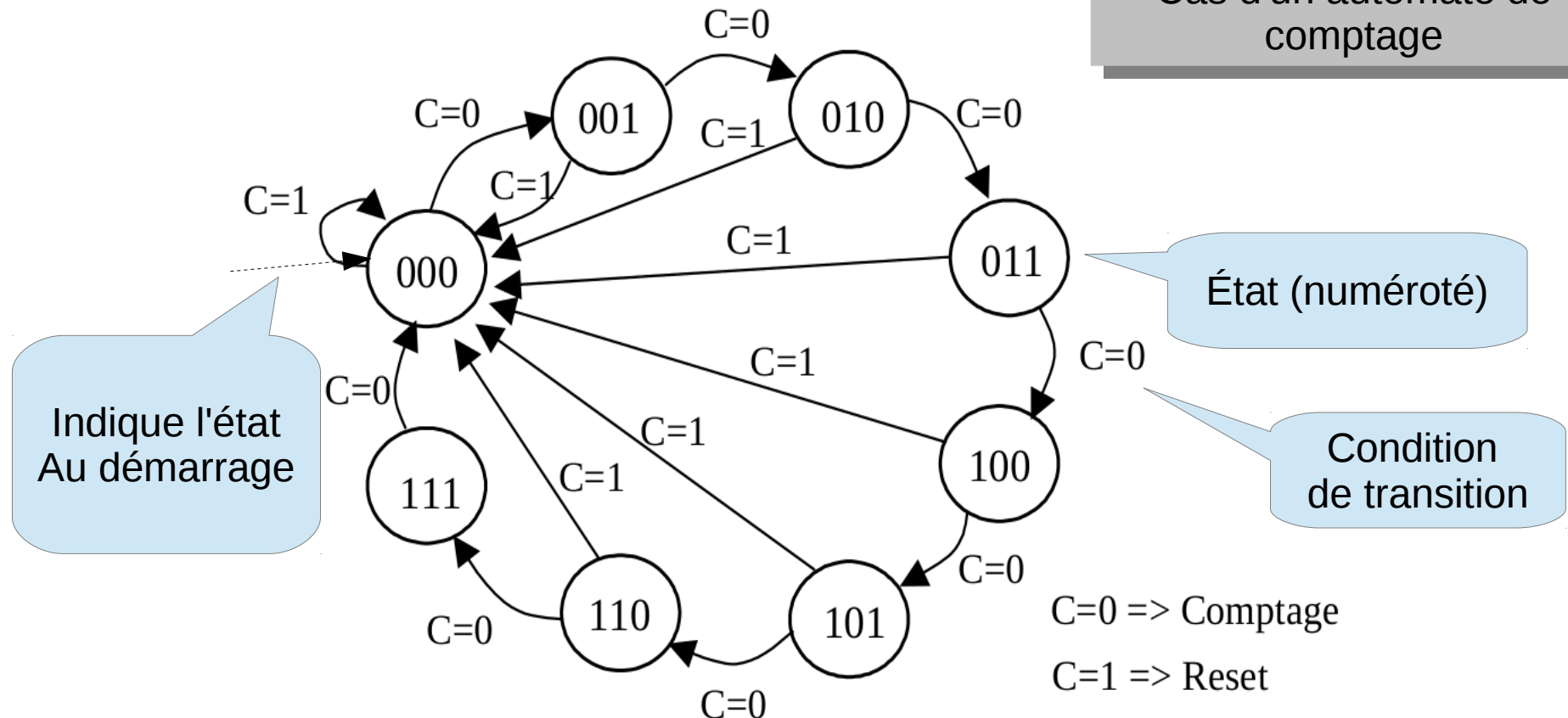
e1	e2	Q	D	ligne
0	0	0	0	0
0	0	1	0	1
0	1	0	1	2
0	1	1	1	3
1	0	0	0	4
1	0	1	1	5
1	1	0	1	6
1	1	1	1	7



- E passe de 0 à 0 sur les lignes 0 et 4 d'où la condition $k_1 = \overline{e_2}$
- E passe de 0 à 1 sur les lignes 2 et 6 d'où la condition $k_2 = e_2$ (soit encore $k_2 = \overline{k_1}$).
- E passe de 1 à 1 sur les lignes 3, 5 et 7 d'où la condition $k_3 = e_1 + e_2$
- E passe de 1 à 0 sur les lignes 1, d'où la condition $k_4 = \overline{e_1} \cdot \overline{e_2}$ (soit encore $k_4 = \overline{k_3}$).

Représentation « diagramme à bulle »

Cas d'un automate de comptage

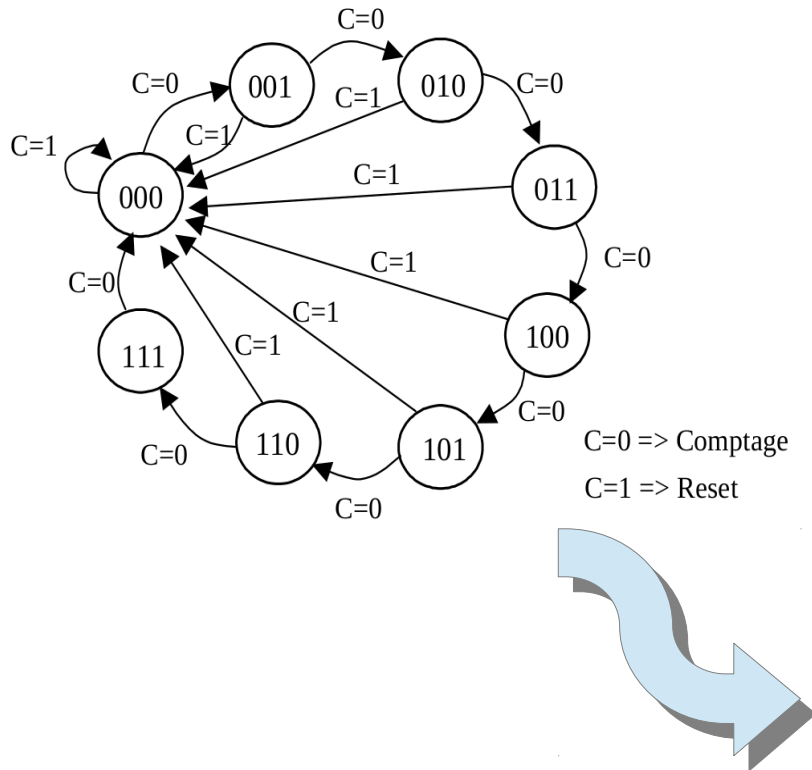


Représentation **intuitive** de l'évolution de l'état du système



Trouver les équations logiques

Table de vérité



Etats	Etats Suivants	
	C = 0	C = 1
000	001	000
001	010	000
010	011 ₅	000
011	100	000
100	101	000
101	110	000
110	111	000
111	000	000

Association Etat \leftrightarrow Bascule

direction : tableaux de Karnaugh !

Etats $Q_2Q_1Q_0(n)$	Etats Suivants $Q_2Q_1Q_0(n+1)$		C = 0			C = 1		
			$D_2(n)$	$D_1(n)$	$D_0(n)$	$D_2(n)$ $D_0(n)$	$D_1(n)$	$D_0(n)$
	C = 0	C = 1						
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

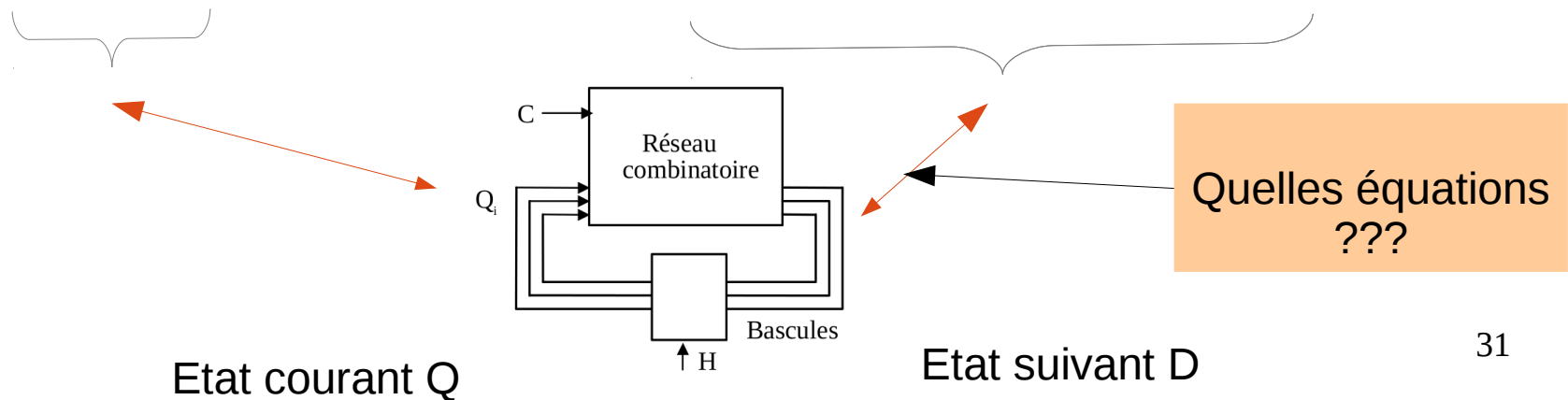


Table de Karnaugh

de la fonction de transition

Etats $Q_2Q_1Q_0(n)$	Etats Suivants $Q_2Q_1Q_0(n+1)$		C = 0			C = 1		
	$Q_2Q_1Q_0(n+1)$		$D_2(n)$	$D_1(n)$	$D_0(n)$	$D_2(n)$	$D_1(n)$	$D_0(n)$
	C = 0	C = 1						
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

Q_0C

Q_2Q_1

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

$$D_2 = C' \cdot (Q_2' \oplus Q_1 \cdot Q_0)$$

Q_0C

Q_2Q_1

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

$$D_1 = C' \cdot (Q_1 \oplus Q_0)$$

Q_0C

Q_2Q_1

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

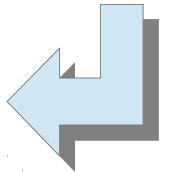
$$D_0 = C' \cdot Q_0'$$

Table de Karnaugh

de la fonction de transition

Etats $Q_2Q_1Q_0(n)$	Etats Suivants $Q_2Q_1Q_0(n+1)$		$C = 0$			$C = 1$		
	$Q_2Q_1Q_0(n+1)$		$D_2(n)$	$D_1(n)$	$D_0(n)$	$D_2(n)$	$D_1(n)$	$D_0(n)$
	$C = 0$	$C = 1$						
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

D_2, D_1, D_0 ?



Q_0C		00	01	11	10
Q_2Q_1	1				
	00				
	01				1
	11	1			
	10	1			1

$$D_2 = C' \cdot (Q_2' \oplus Q_1 \cdot Q_0)$$

Q_0C		00	01	11	10
Q_2Q_1	1				
	00				1
	01	1			
	11	1			
	10				1

$$D_1 = C' \cdot (Q_1 \oplus Q_0)$$

Q_0C		00	01	11	10
Q_2Q_1	1				
	00	1			
	01	1			
	11	1			
	10	1			

$$D_0 = C' \cdot Q_0'$$

State transition diagram for a 3-bit counter with states 000, 001, 010, 011, 100, 101, 110, and 111. Transitions are labeled with C=0 (Comptage) or C=1 (Reset).

Legend:

- C=0 => Comptage
- C=1 => Reset

```

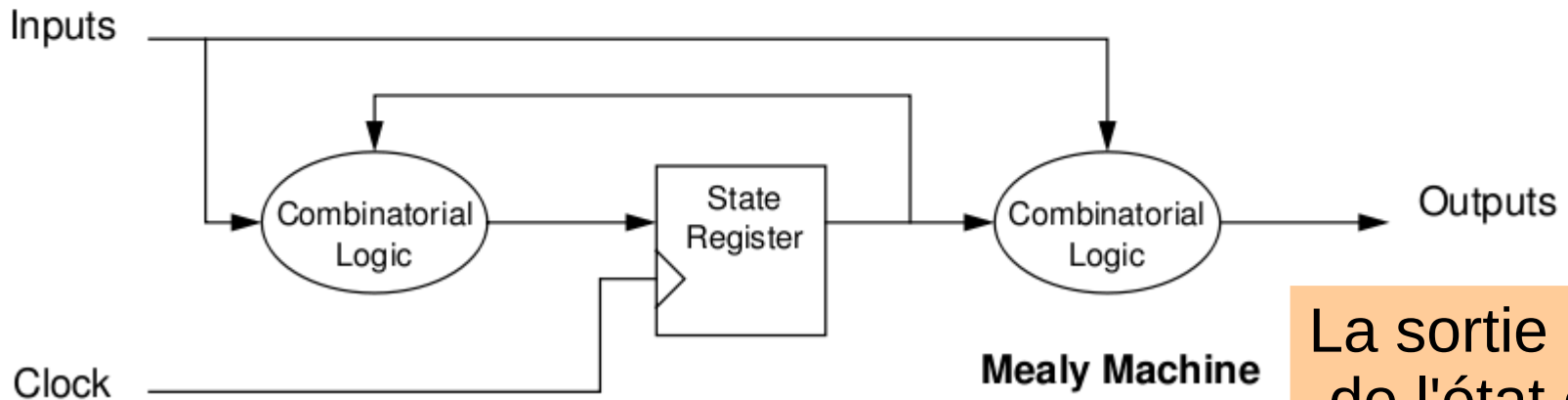
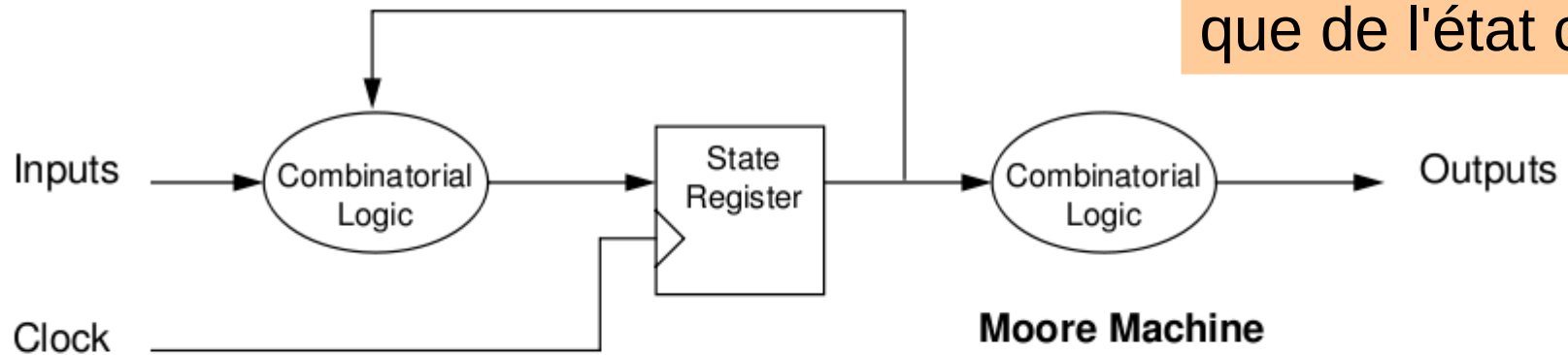
graph TD
    000((000)) -- C=0 --> 001((001))
    001 -- C=0 --> 010((010))
    010 -- C=0 --> 011((011))
    011 -- C=0 --> 100((100))
    100 -- C=0 --> 101((101))
    101 -- C=0 --> 110((110))
    110 -- C=0 --> 111((111))
    111 -- C=0 --> 000
    000 -- C=1 --> 011
    001 -- C=1 --> 010
    010 -- C=1 --> 100
    011 -- C=1 --> 101
    100 -- C=1 --> 110
    101 -- C=1 --> 111
    110 -- C=1 --> 000
    111 -- C=1 --> 001
  
```



FSM

Mealy et Moore

La sortie ne dépend
que de l'état courant

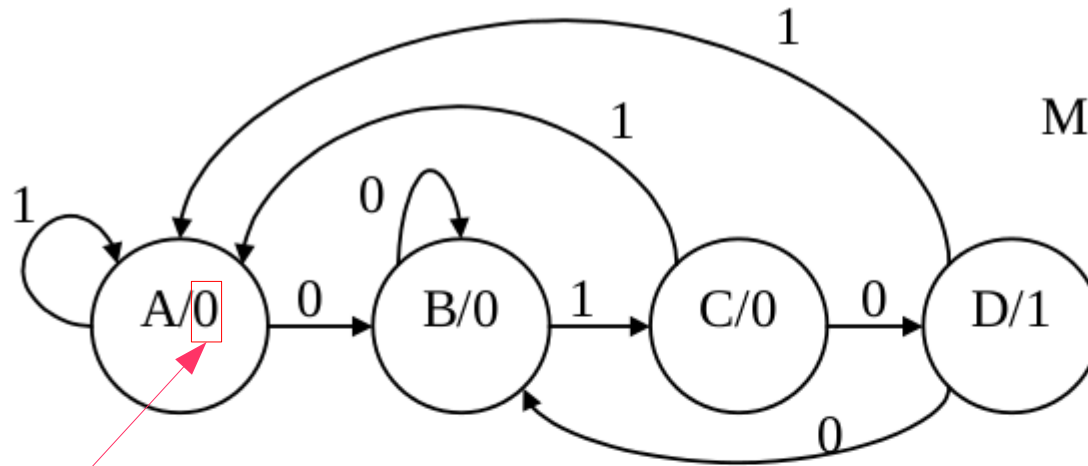


La sortie dépend
de l'état courant
et des entrées

FSM de Mealy / Moore

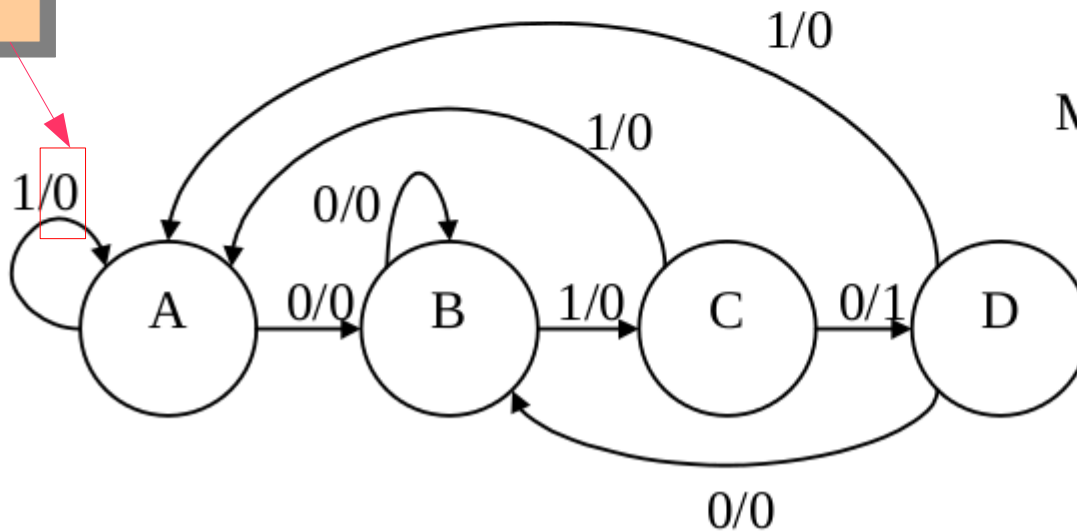
diagramme à bulles

Machine de Moore



sortie

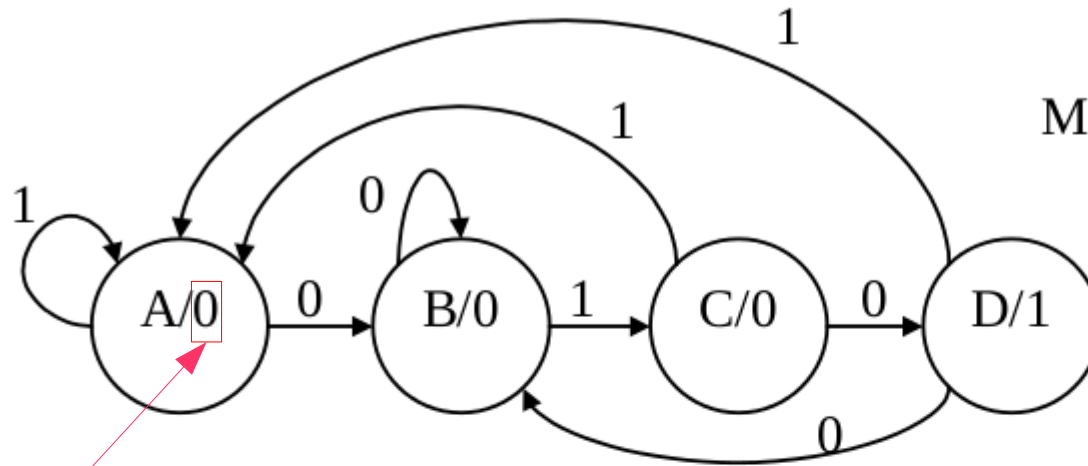
Machine de Mealey



FSM de Mealy / Moore

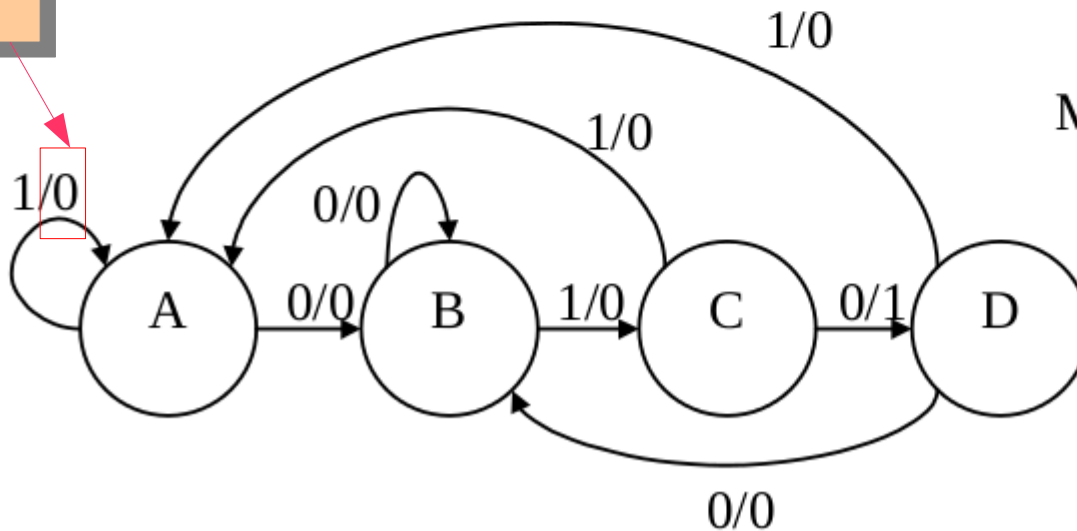
diagramme à bulles

Machine de Moore



sortie

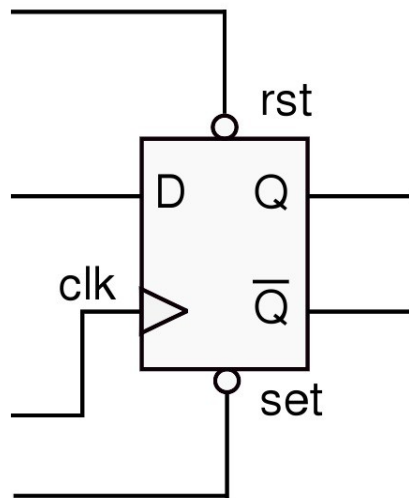
Machine de Mealey



Initialisation d'une bascule D

au démarrage du circuit

Set et reset prioritaire sur l'horloge !



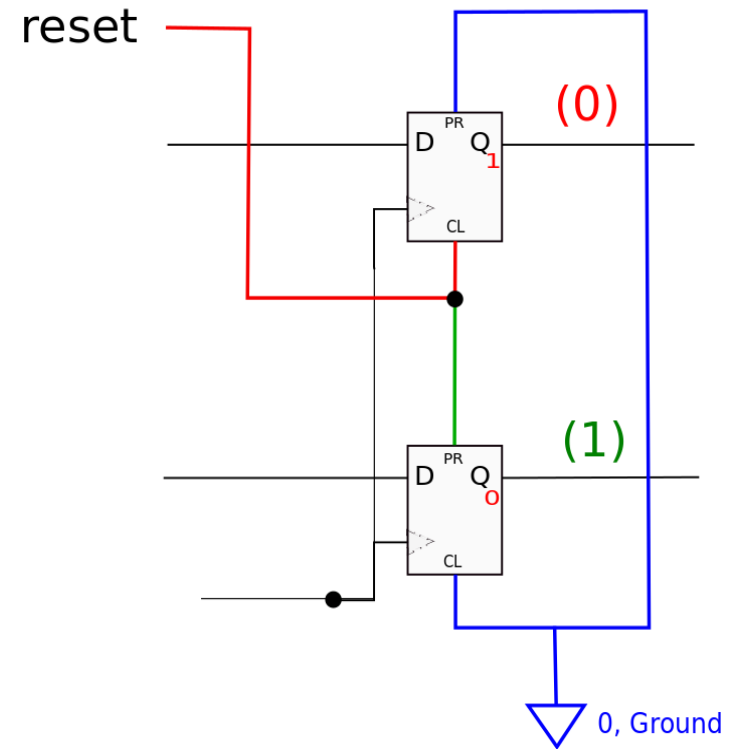
- Le symbol ° ou son absence indique la polarité du signal actif
 - Presence de °
 - Pol = 0
 - Absence de °
 - Pol = 1

Set = POL \Rightarrow Q=1
Rst = POL \Rightarrow Q=0

Dans tous les autres cas,
Q reste **inchangé**

Exemple d'initialisation

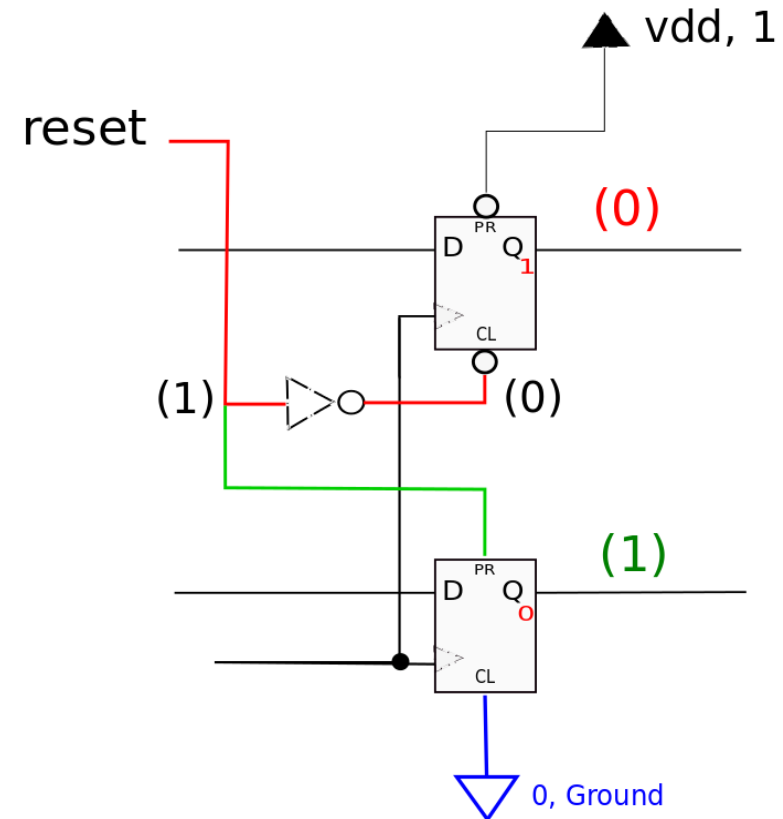
- Cas d'un automate simple
 - État encodé sur 2 bits
 - Stockés dans 2 bascules D
 - Q1 et Q0
 - Etat de démarrage
 - 01 => Q1=0 & Q0=1
 - On suppose ici que le reset = signal **actif haut**



Exemple d'initialisation

autres bascules, autre init !

- Cas d'un automate simple
 - État encodé sur 2 bits
 - Stockés dans 2 bascules D
 - Q1 et Q0
 - Etat de démarrage
 - 01 => Q1=0 & Q0=1
 - On suppose ici que le reset = signal **actif haut**



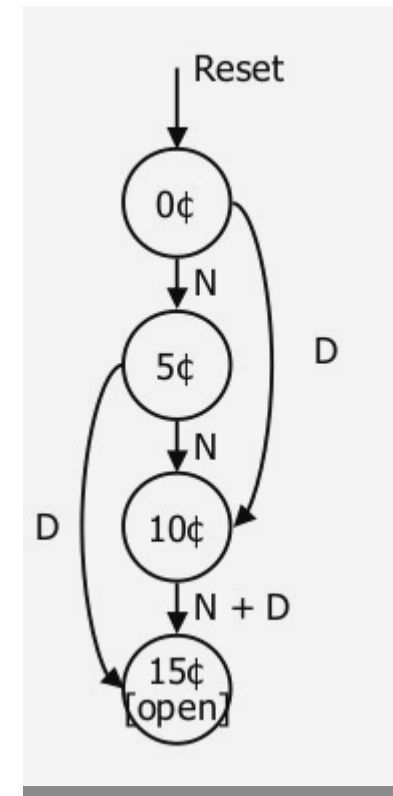
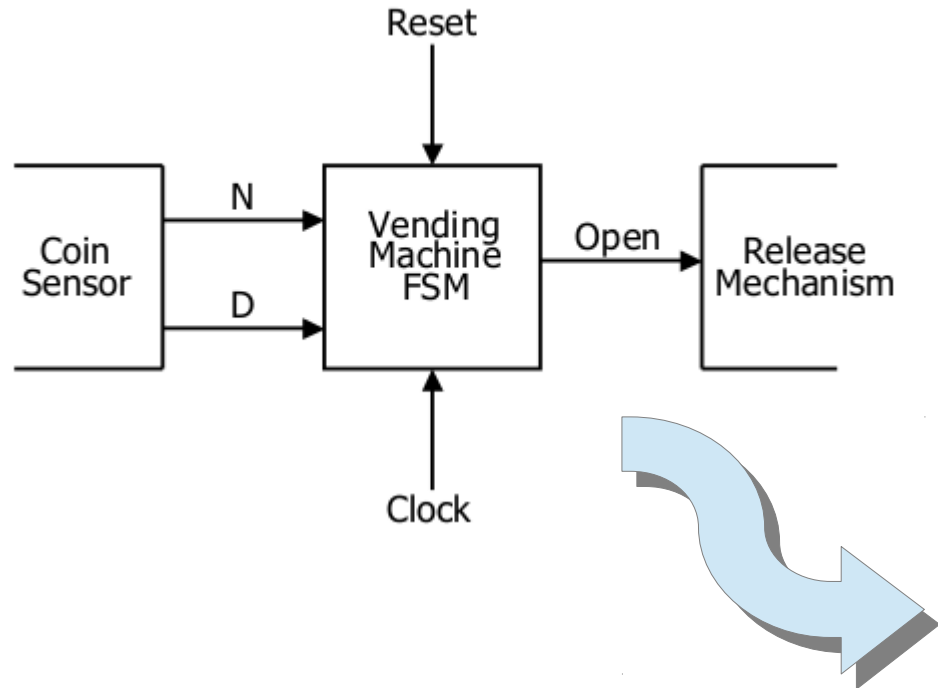
L'encodage des états

- La numérotation des états a des conséquences importantes sur la complexité des équations des fonctions d'état suivant et des sorties
 - Cette numérotation s'appelle l'**encodage des états**
 - Dans l'exemple précédent : $S_0 \rightarrow 0, S_1 \rightarrow 1, S_2 \rightarrow 2..$
 - $N \text{ états} \Rightarrow \text{Ceil}(\log_2(N)) \text{ bits} \Rightarrow \text{encodage dense}$
- Parmi les stratégies possibles, l'une d'entre elle est très répandue : encodage « one-hot »
 - Aussi appelé « 1 bit par état »
 - $N \text{ états} \Rightarrow N \text{ bits} \Rightarrow \text{encodage « one-hot »}$
 - Elle conduit à des circuits rapides
 - ...moyennant plus de bascules que pour d'autres encodages
 - On peut obtenir **directement** le circuit !!!

Encodage One-hot

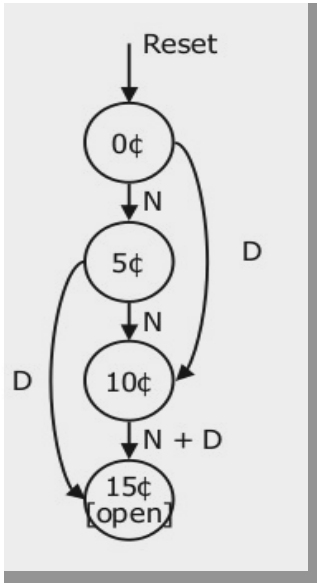
- 1 bit par état :
 - N états => N bascules
 - Exemple
 - S0 = 001
 - S1 = 010
 - S2 = 100
 - Une seule bascule possède le 1 à un instant donné
- Tout se passe comme si, lors d'une transition d'un état à un autre, le '1' se déplaçait d'une bascule à l'autre, comme un jeton...

Exemple : le distributeur



(on précise que le dispositif ne rend pas la monnaie)

Méthode « classique » (sans astuce)



present state				inputs		next state				output
Q_3	Q_2	Q_1	Q_0	D	N	D_3	D_2	D_1	D_0	open
0	0	0	1	0	0	0	0	0	1	0
				0	1	0	0	1	0	0
				1	0	0	1	0	0	0
				1	1	—	—	—	—	—
0	0	1	0	0	0	0	0	1	0	0
				0	1	0	1	0	0	0
				1	0	1	0	0	0	0
				1	1	—	—	—	—	—
0	1	0	0	0	0	0	1	0	0	0
				0	1	1	0	0	0	0
				1	0	1	0	0	0	0
				1	1	—	—	—	—	—
1	0	0	0	—	—	1	0	0	0	1

$$D_0 = Q_0 D' N'$$

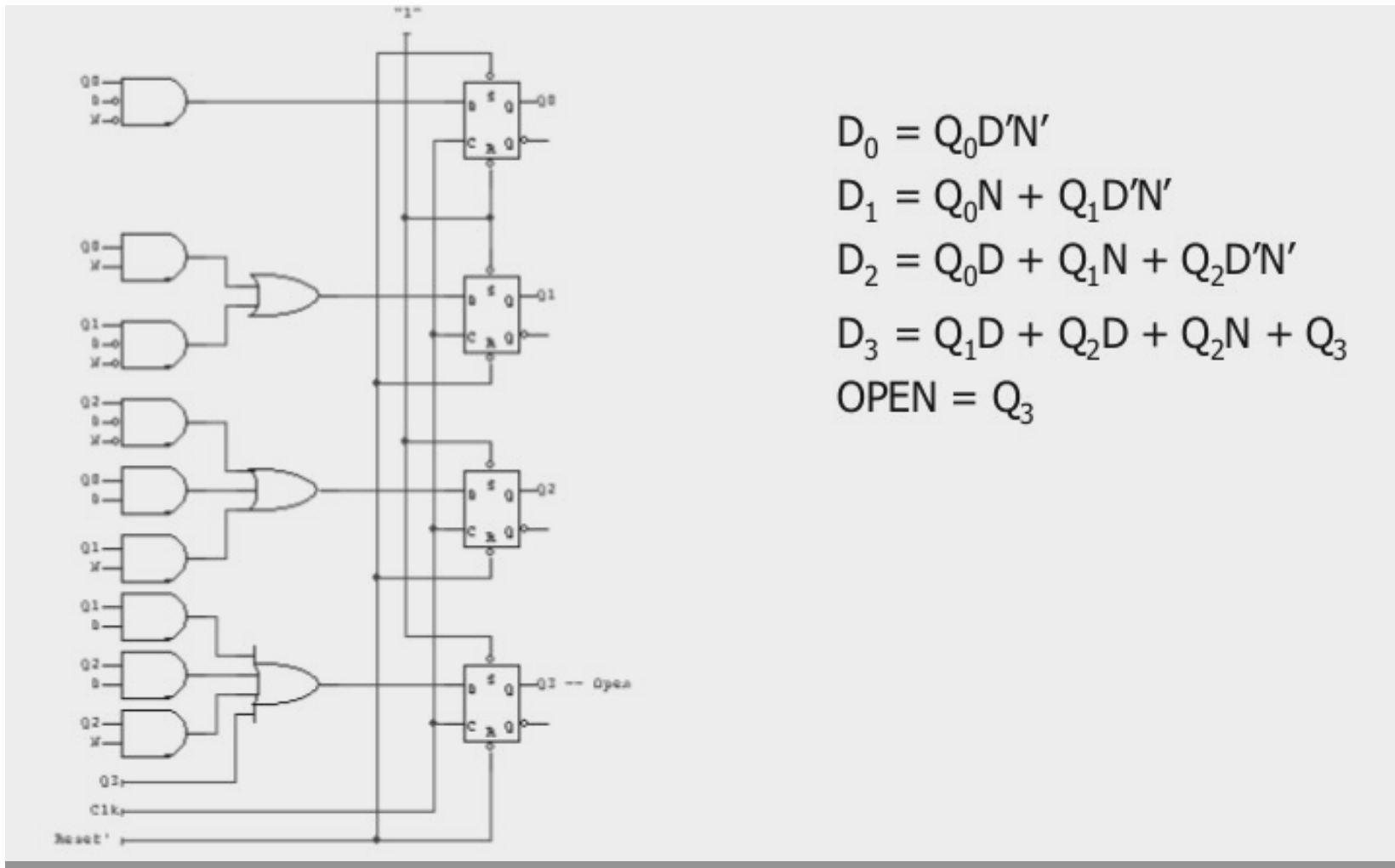
$$D_1 = Q_0 N + Q_1 D' N'$$

$$D_2 = Q_0 D + Q_1 N + Q_2 D' N'$$

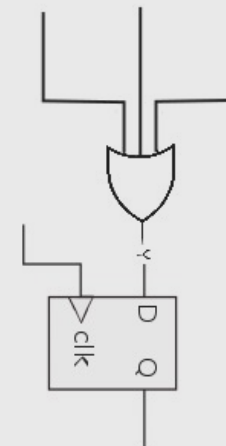
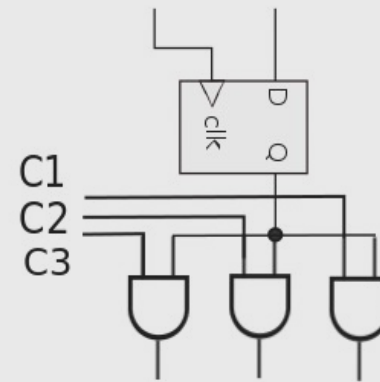
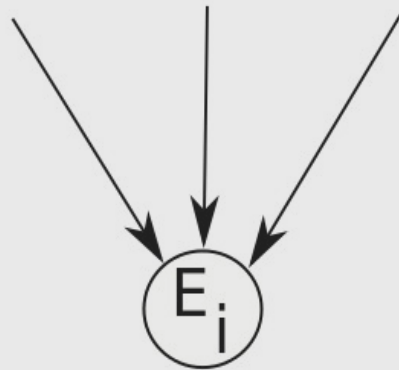
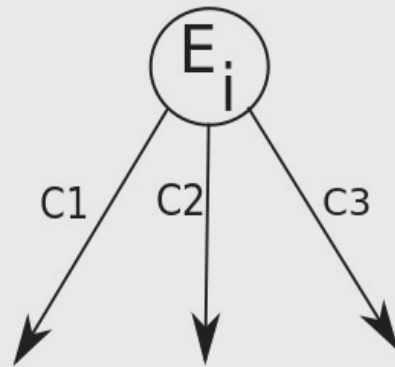
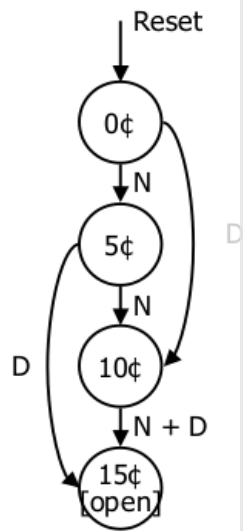
$$D_3 = Q_1 D + Q_2 D + Q_2 N + Q_3$$

$$\text{OPEN} = Q_3$$

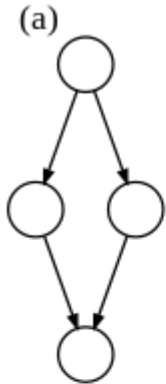
Résultat



Méthode astucieuse, graphique

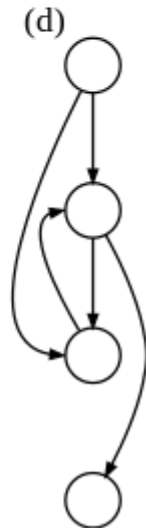
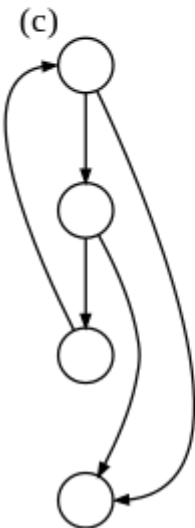


Quelques « patterns » de contrôle connus



a) if then else

b) while cond // ou for....



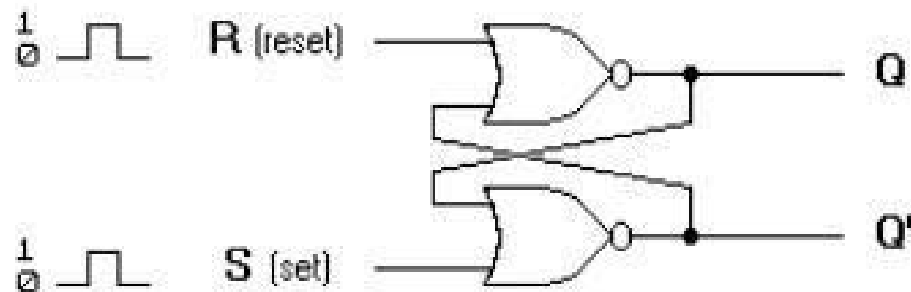
c) while { // ou for....

```
If() {  
    Exit loop ;  
}
```

d) « loop » à deux entrées
(« goto » dans une boucle)....

Constitution interne des bascules

Constitution interne (1/4)



(a) Logic diagram

Exception (cachée) :
ici on autorise
exceptionnellement
un rebouclage
combinatoire

S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

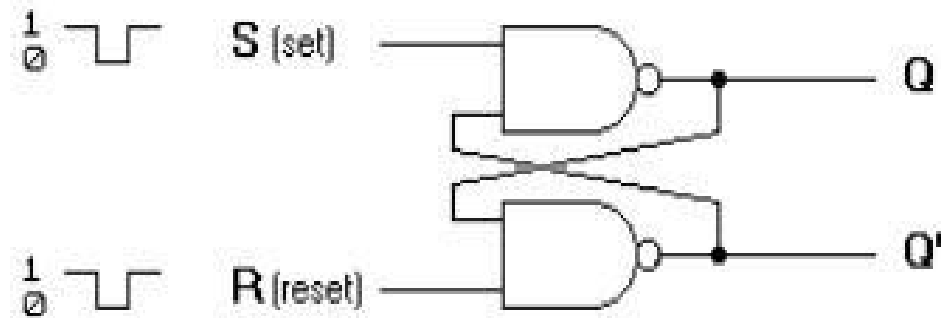
(after S=1, R=0)

(after S=0, R=1)

(b) Truth table

Basic flip-flop circuit with NOR gates

Constitution interne (2/4)



(a) Logic diagram

S	R	Q	Q'
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1

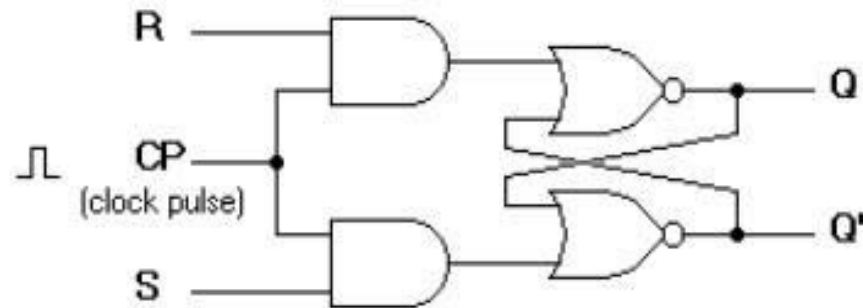
(after S=1, R=0)

(after S=0, R=1)

(b) Truth table

Basic flip-flop circuit with NAND gates

Constitution interne (3/4)



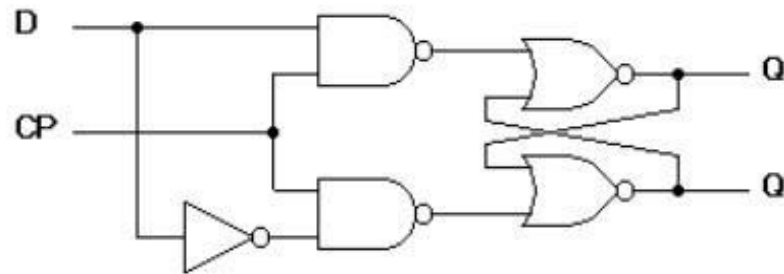
(a) Logic diagram

Q	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

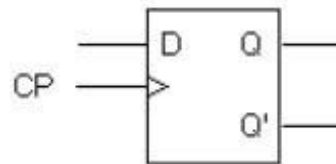
(b) Truth table

Clocked SR flip-flop

Constitution interne (4/4)



(a) Logic diagram with NAND gates



(b) Graphical symbol

Q	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

(c) Transition table