

ENS L3 : "Systèmes numériques : de l'algorithme aux circuits"

Leçon : logique séquentielle synchrone

Sylvain GUILLEY
15 novembre 2016

Notions apprises dans ce cours

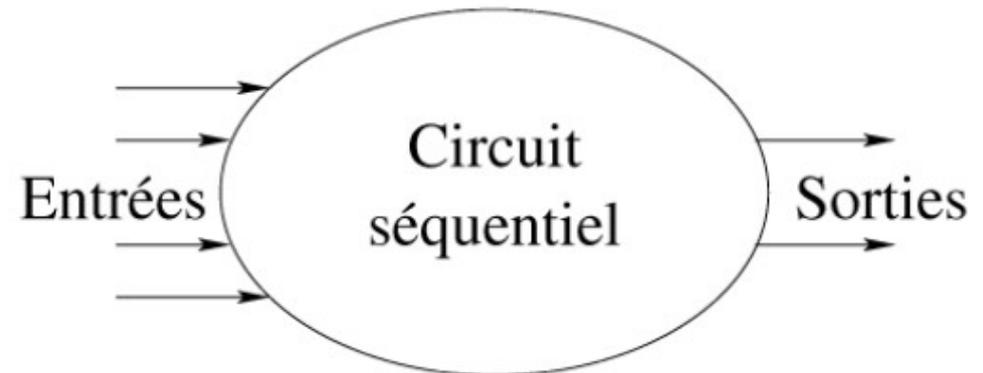
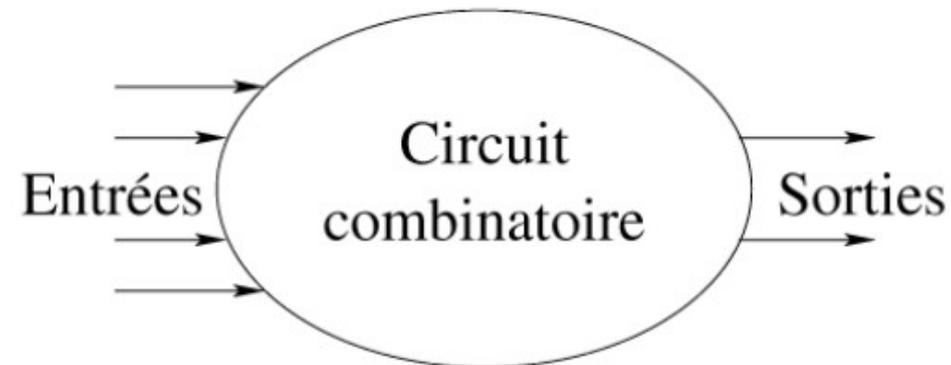
- Logique séquentielle
- Logique synchrone
- Pipeline
- Machine à états finis
 - Modélisation
 - Réalisation
 - Synthèse
- Différents codage des états

Organisation des calculs

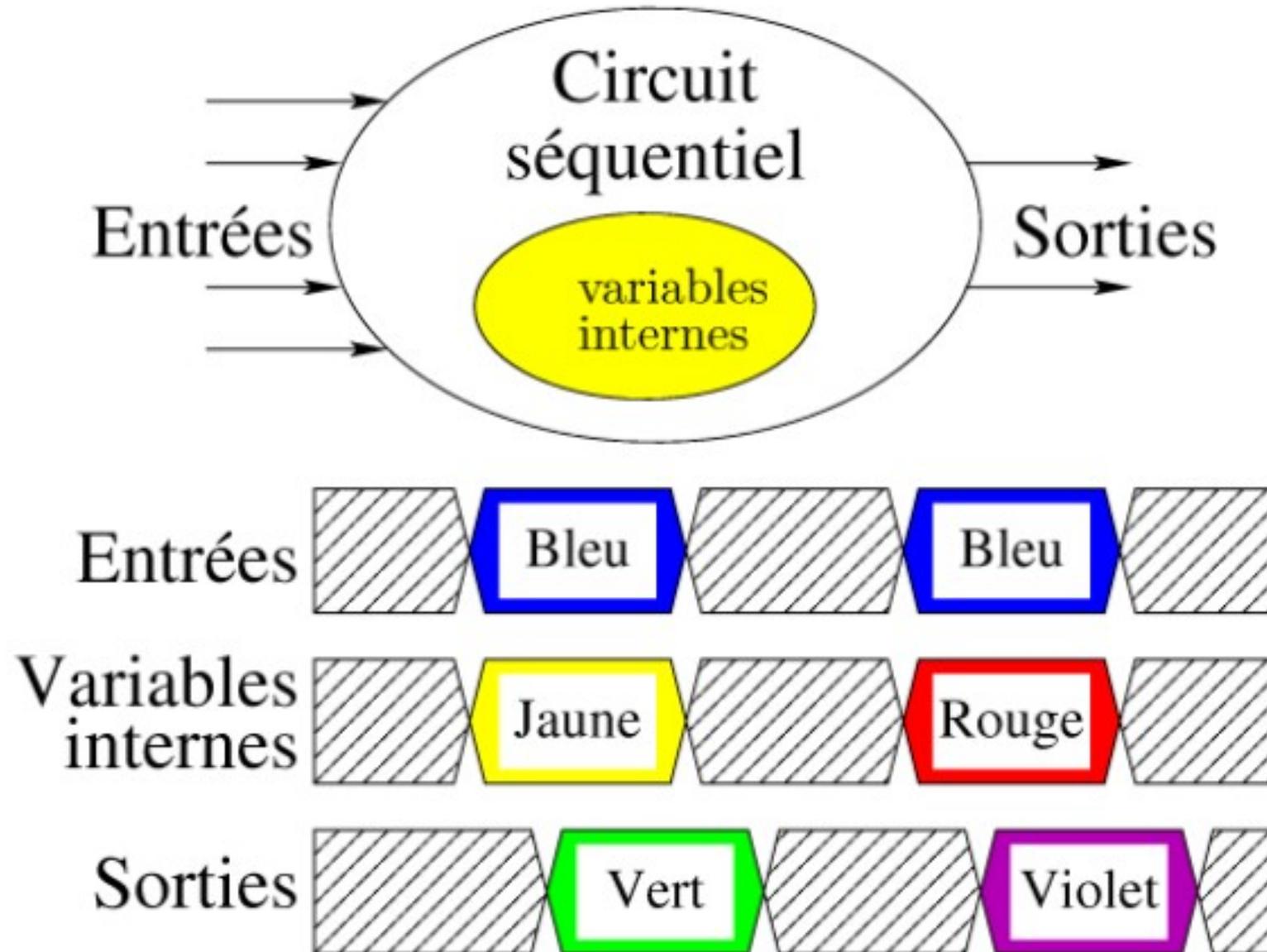
- Comme faire 1000 fois une tâche ?
 - Avec 1000 instances parallèles
 - Avec 1 instance réutilisée 1000 fois
- Espace vs temps
- Temps : mémorisation
 - Registre
 - Mémoire
 - SDRAM
 - FLASH
 - DDR

Reconnaître combi vs séquentiel

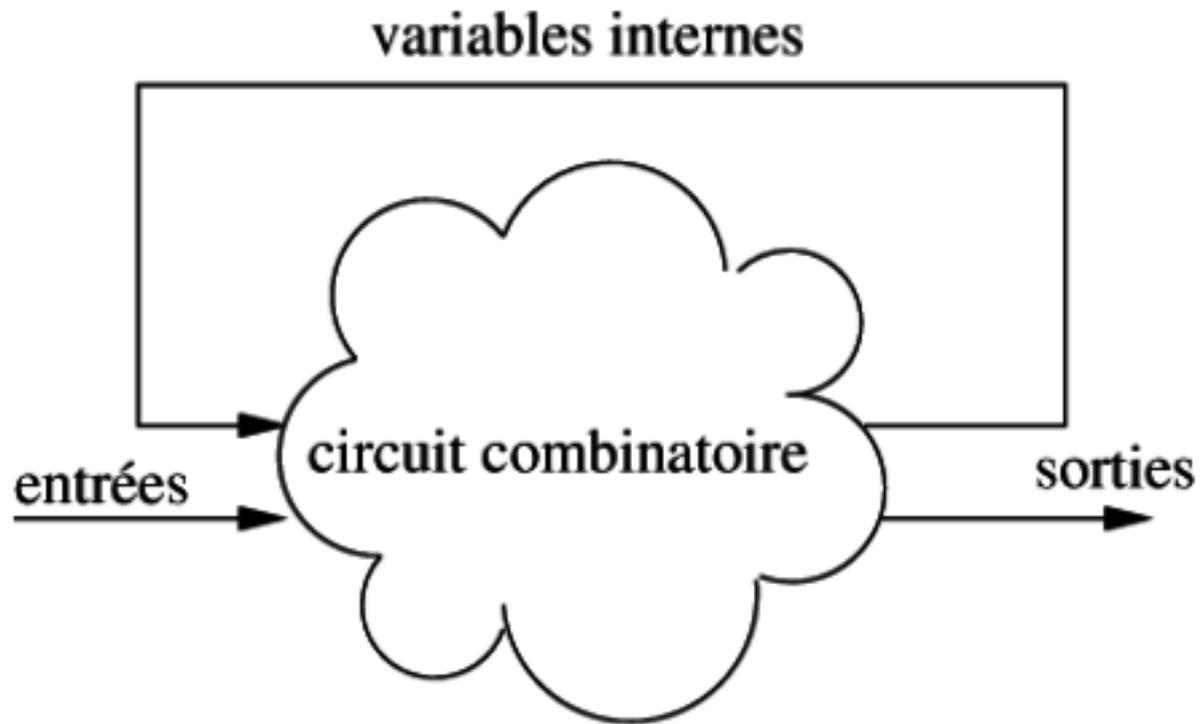
- Chronogrammes



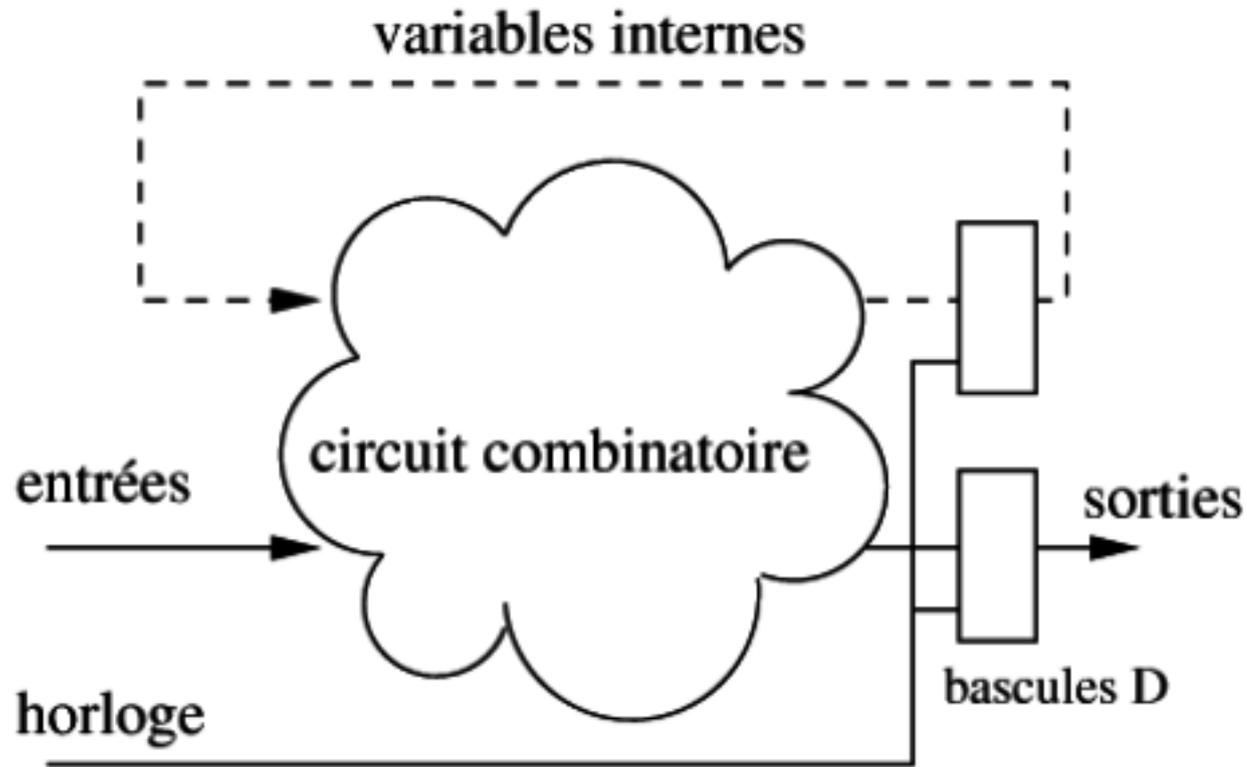
Comment ça marche ?



Comment ça marche ?

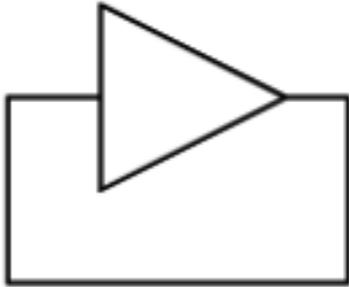


Comment ça marche ?

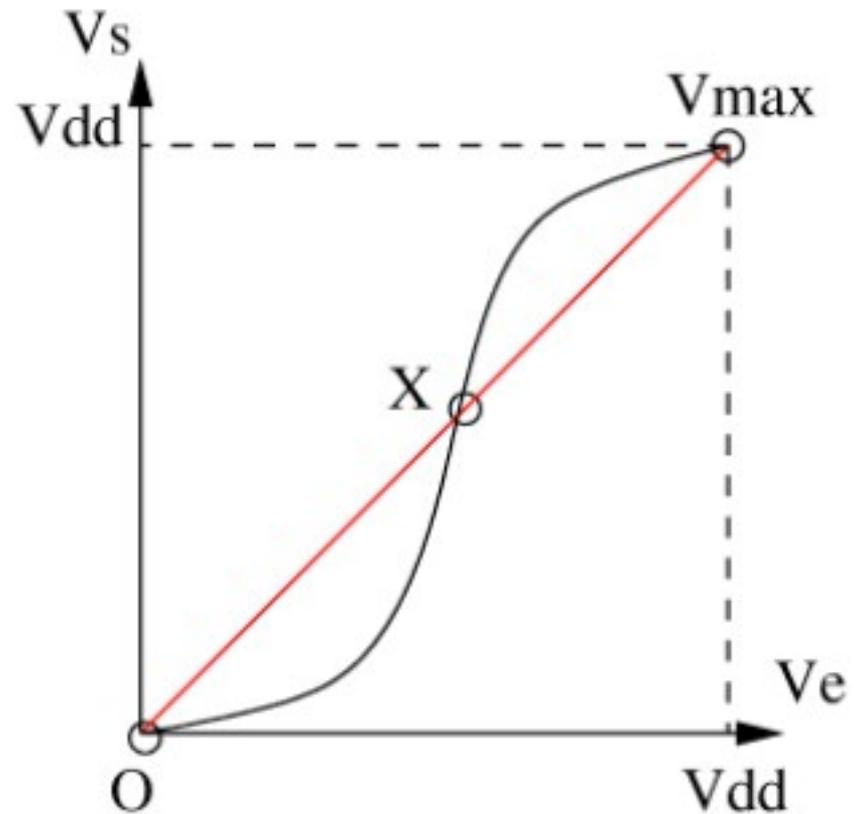


▲ : instant d'échantillonnage — : mémorisation

Mémoriser

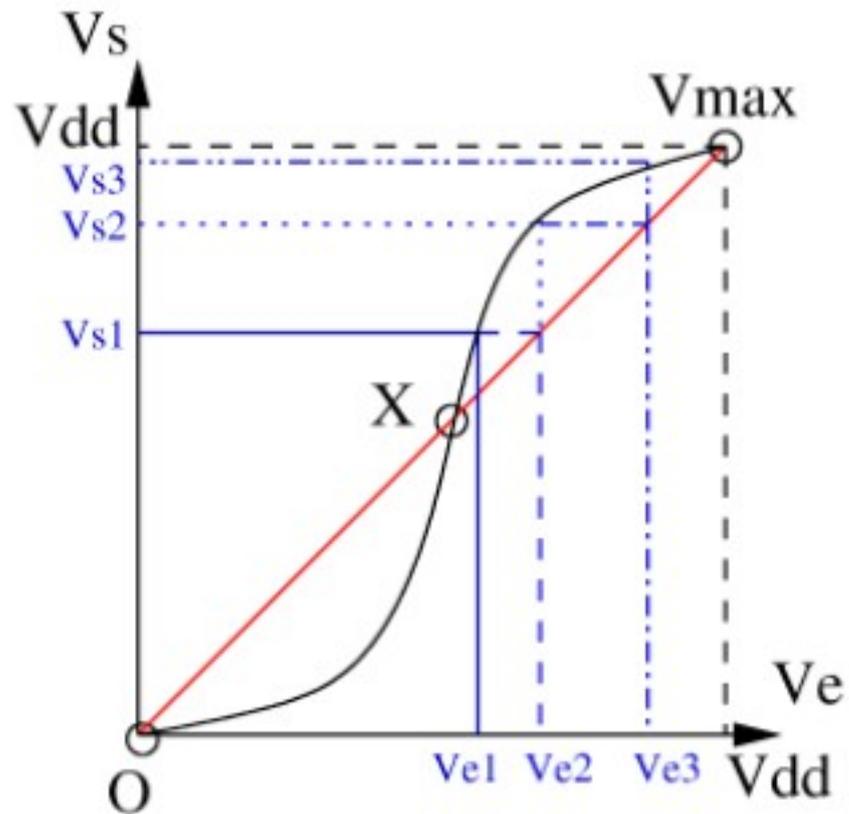


Point mémoire basé sur un amplificateur rebouclé



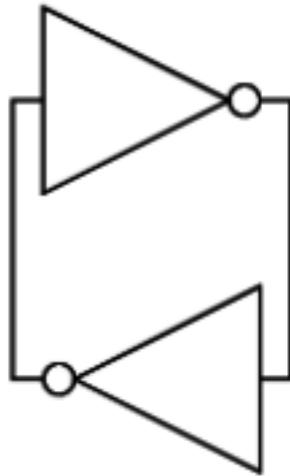
Fonction de transfert de l'amplificateur

Mémoriser

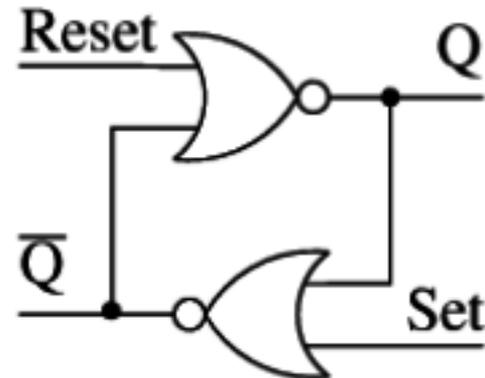
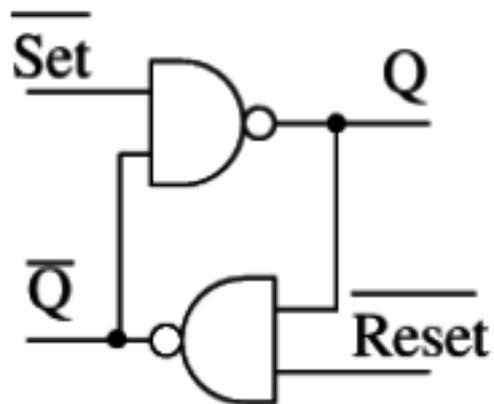


Convergence vers un état stable en ne partant pas de X

Mémoriser

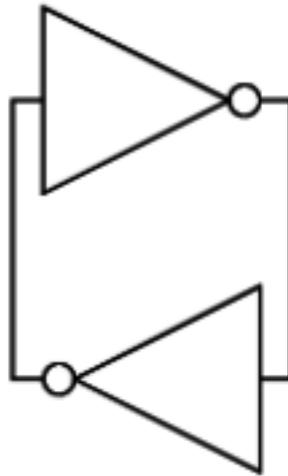


inverseurs en tête bêche pour la mémorisation

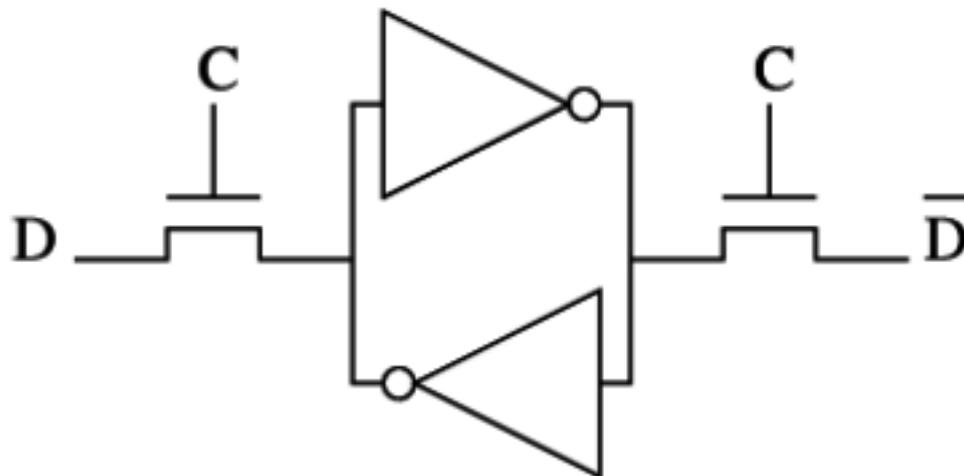


bascule RS avec une structure NAND et NOR

Mémoriser

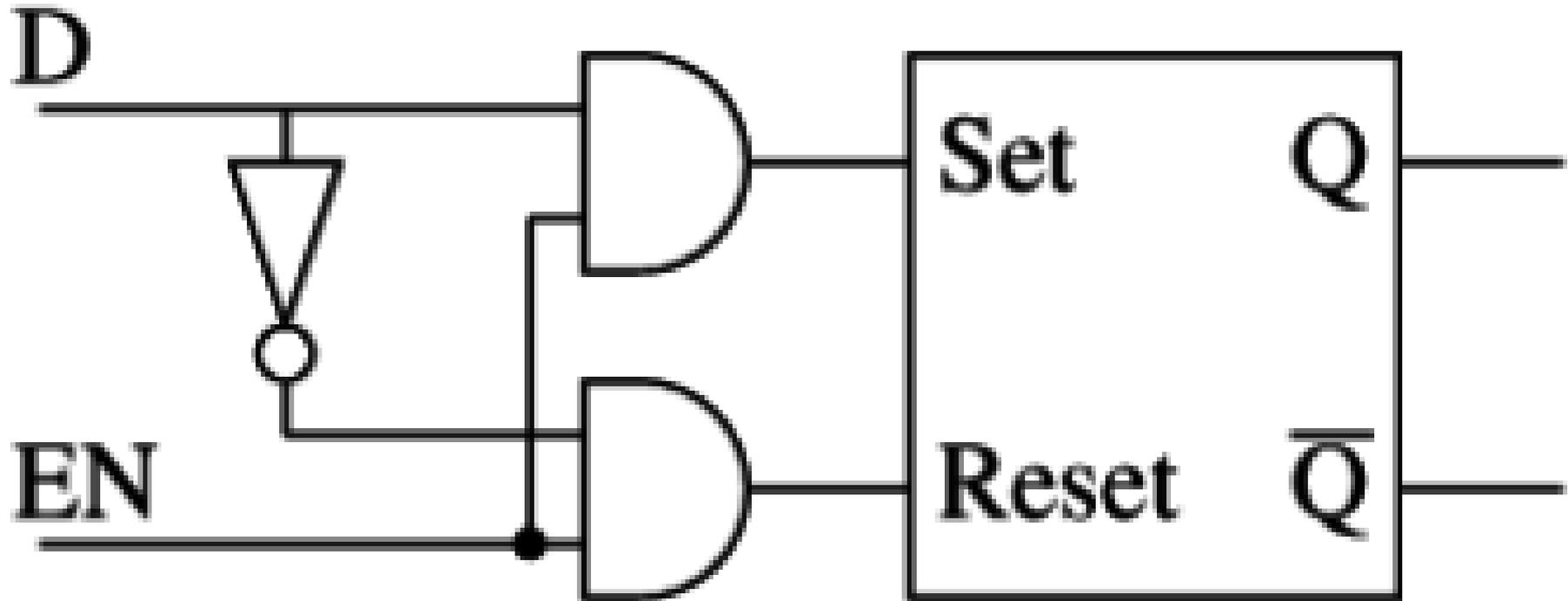


inverseurs en tête bêche pour la mémorisation



Point mémoire RAM statique

Latch (=verrou)



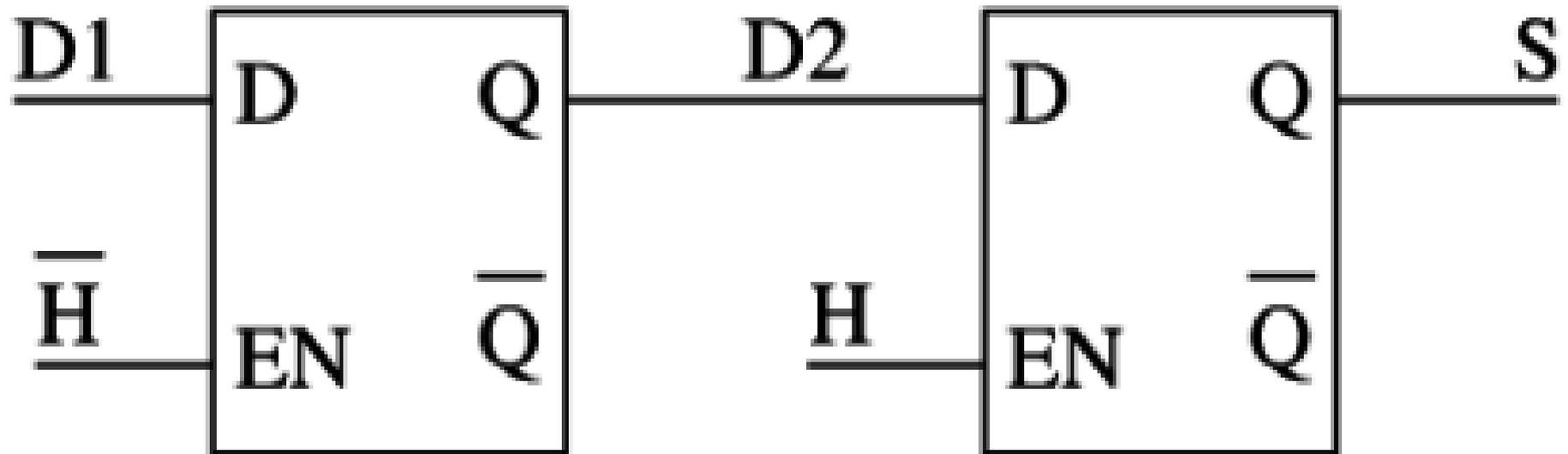
Structure de la bascule D sur état ou latch

Q : imaginer une structure plus simple

Latch

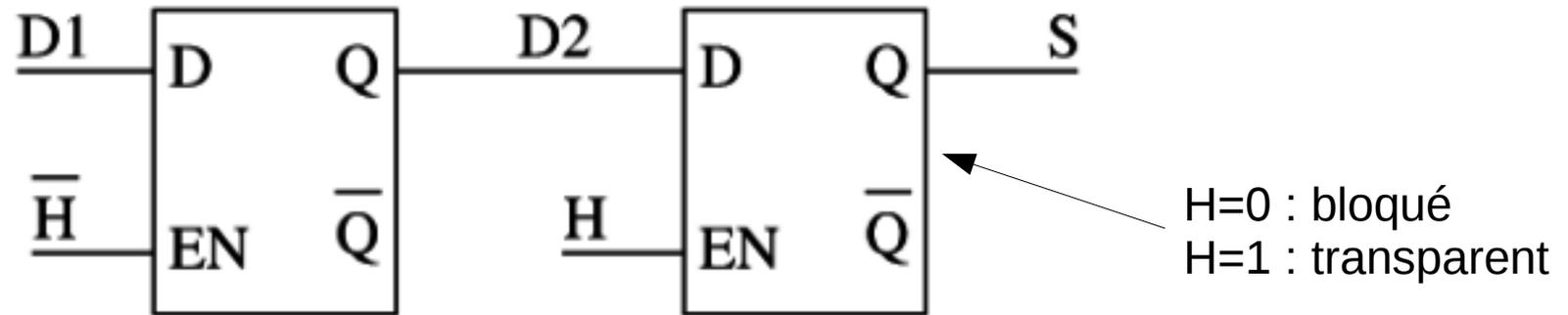


Bascule D

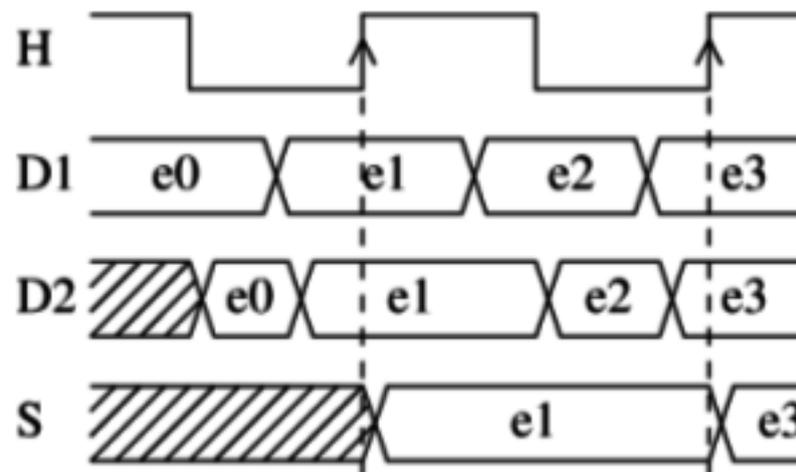


structure de la bascule D à partir de latches

Bascule D

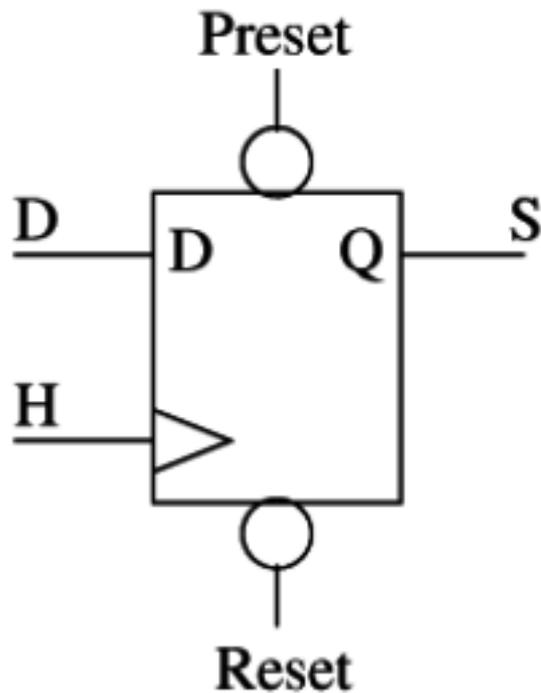


structure de la bascule D à partir de latches



Chronogramme de la bascule D avec 2 latches

Bascule D



Symbole de la bascule D

D	H	Preset	Reset	Q	Etat
0	↑	1	1	0	échantillonnage
1	↑	1	1	1	
X	0	1	1	Q	mémorisation
X	1	1	1	Q	
X	X	0	1	1	forçage à 1
X	X	1	0	0	forçage à 0

Table de vérité de la bascule D

Schéma de la bascule D

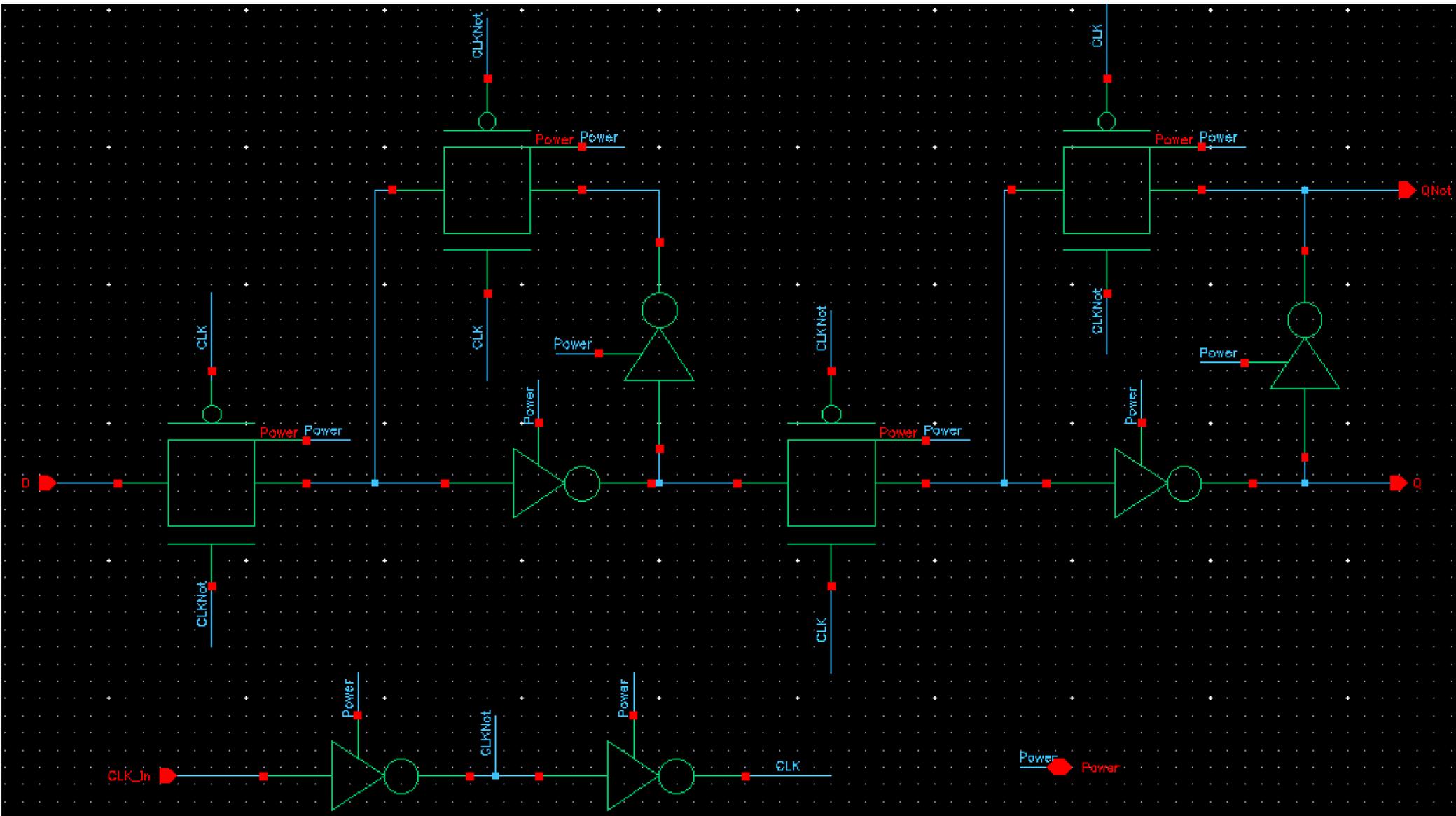
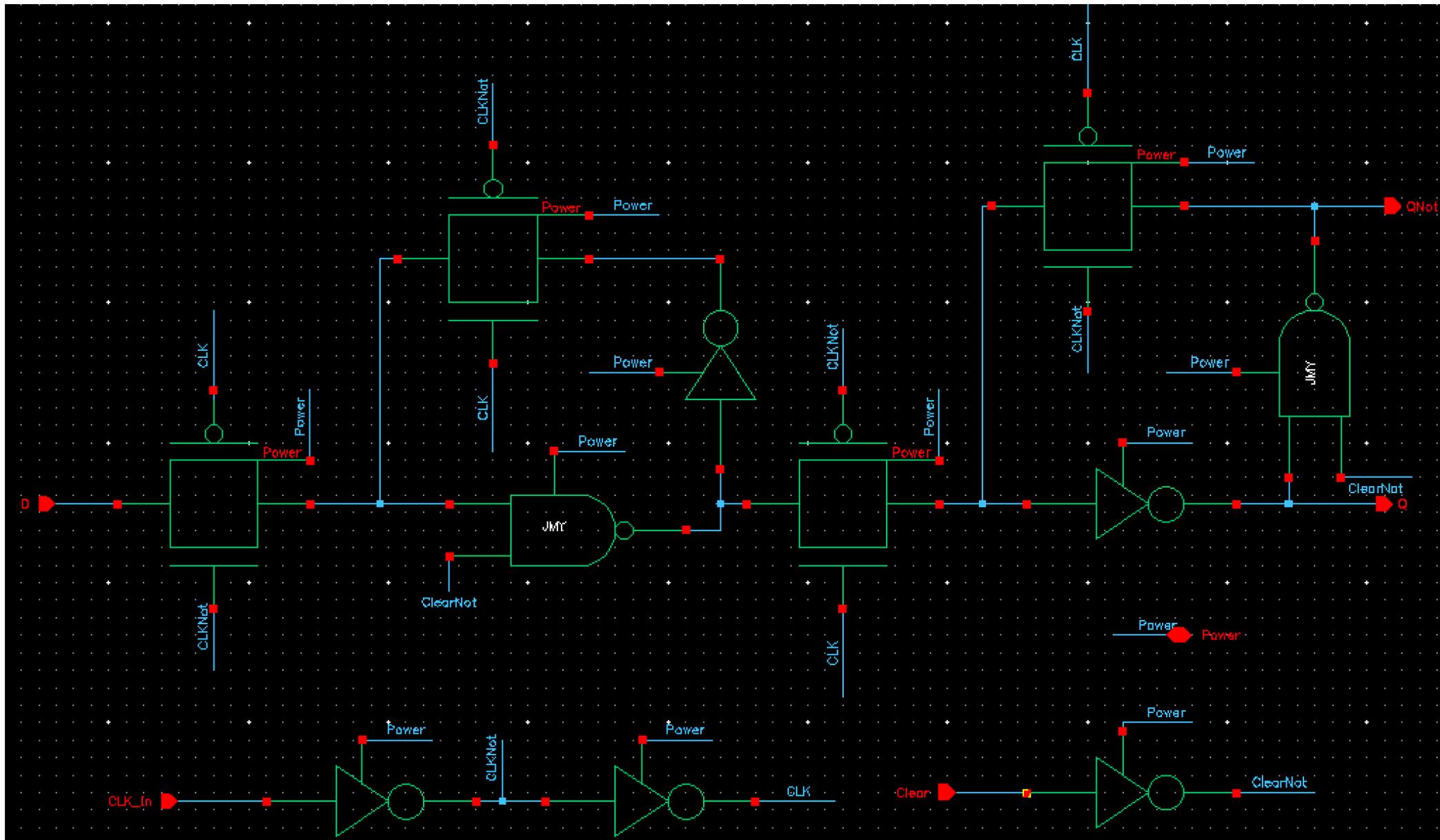
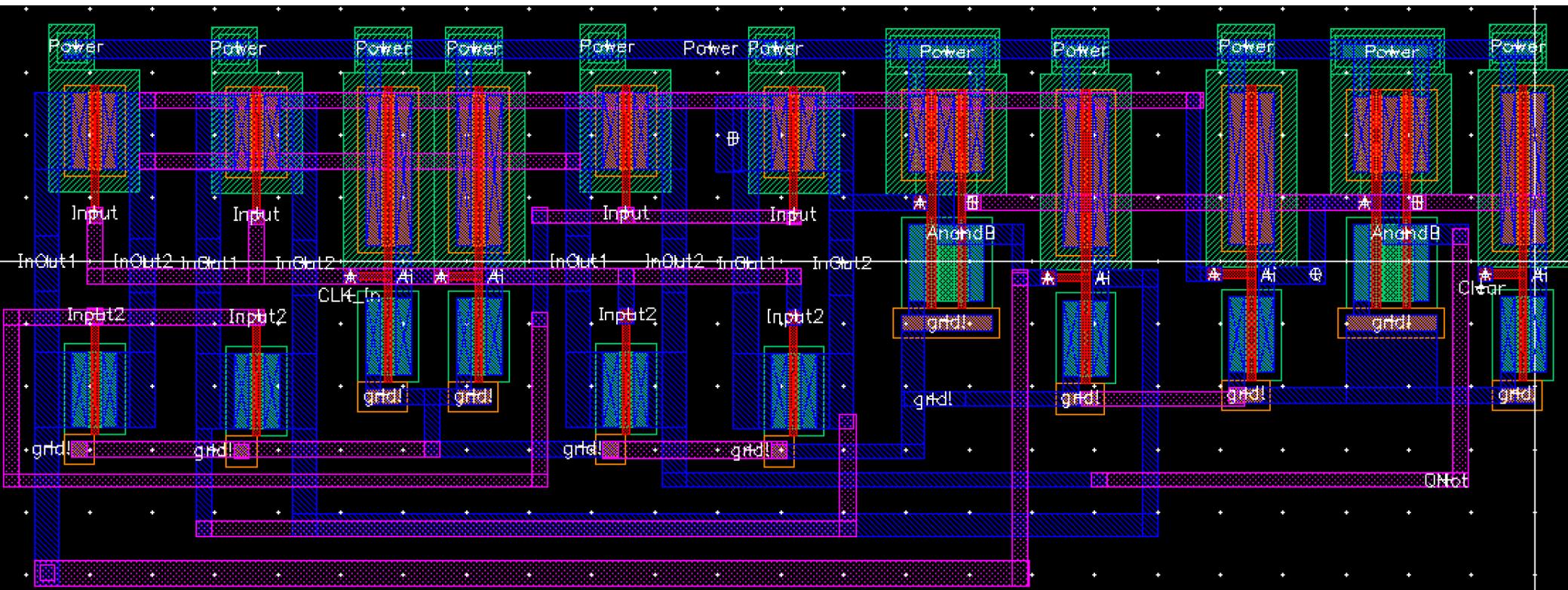


Schéma de la bascule D avec reset



« Layout » de la bascule D



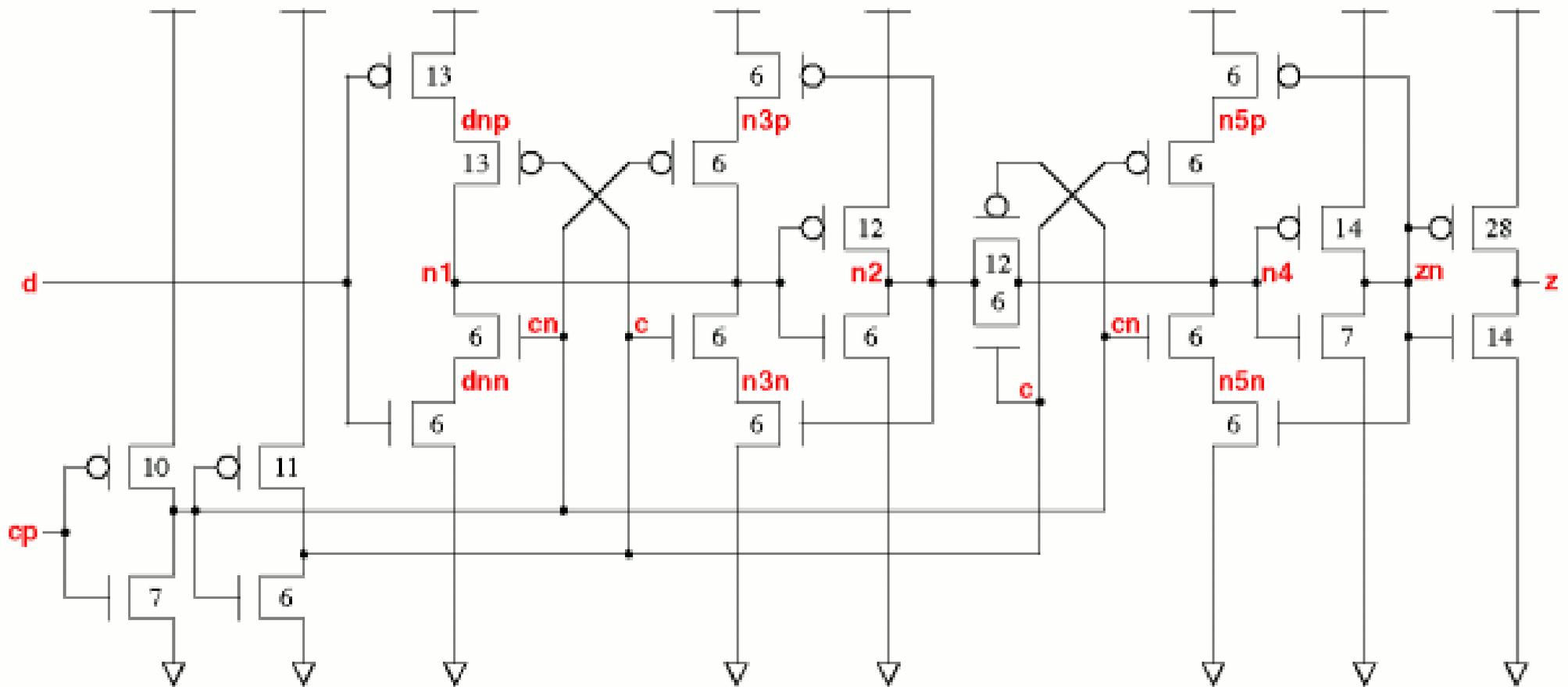
Virtuoso® 6.1.5 - Log: /home/youngj1/CDS.log@csmcluster.ee.unlv.edu

File Tools Options Help

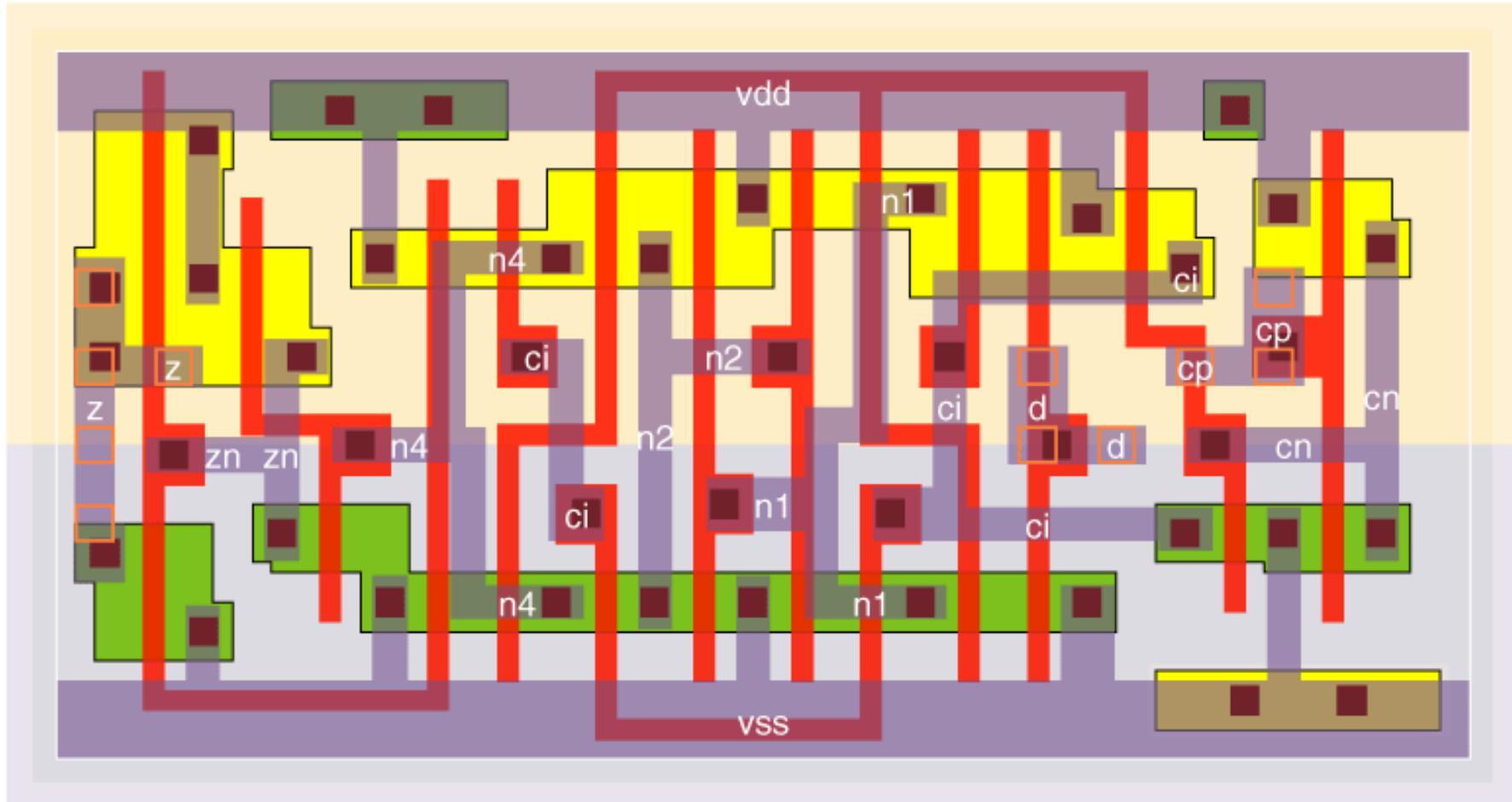
cadence

```
***** Summary of rule violations for cell "DFF_Clear layout" *****  
Total errors found: 0
```

Rising edge flip flop

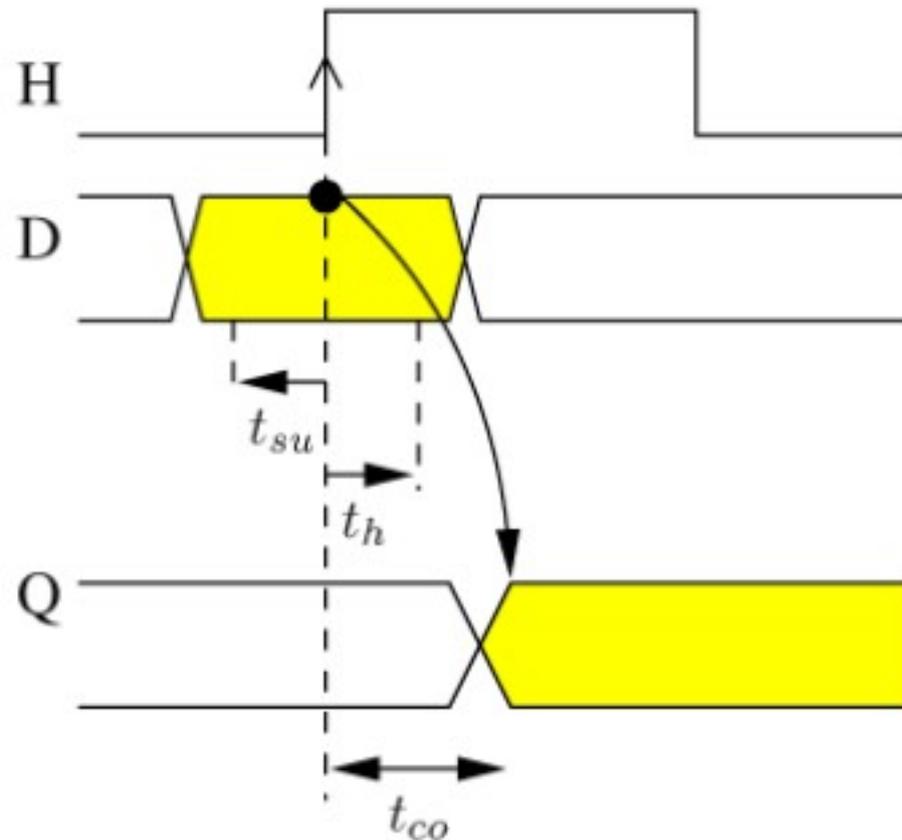


Rising edge flip flop



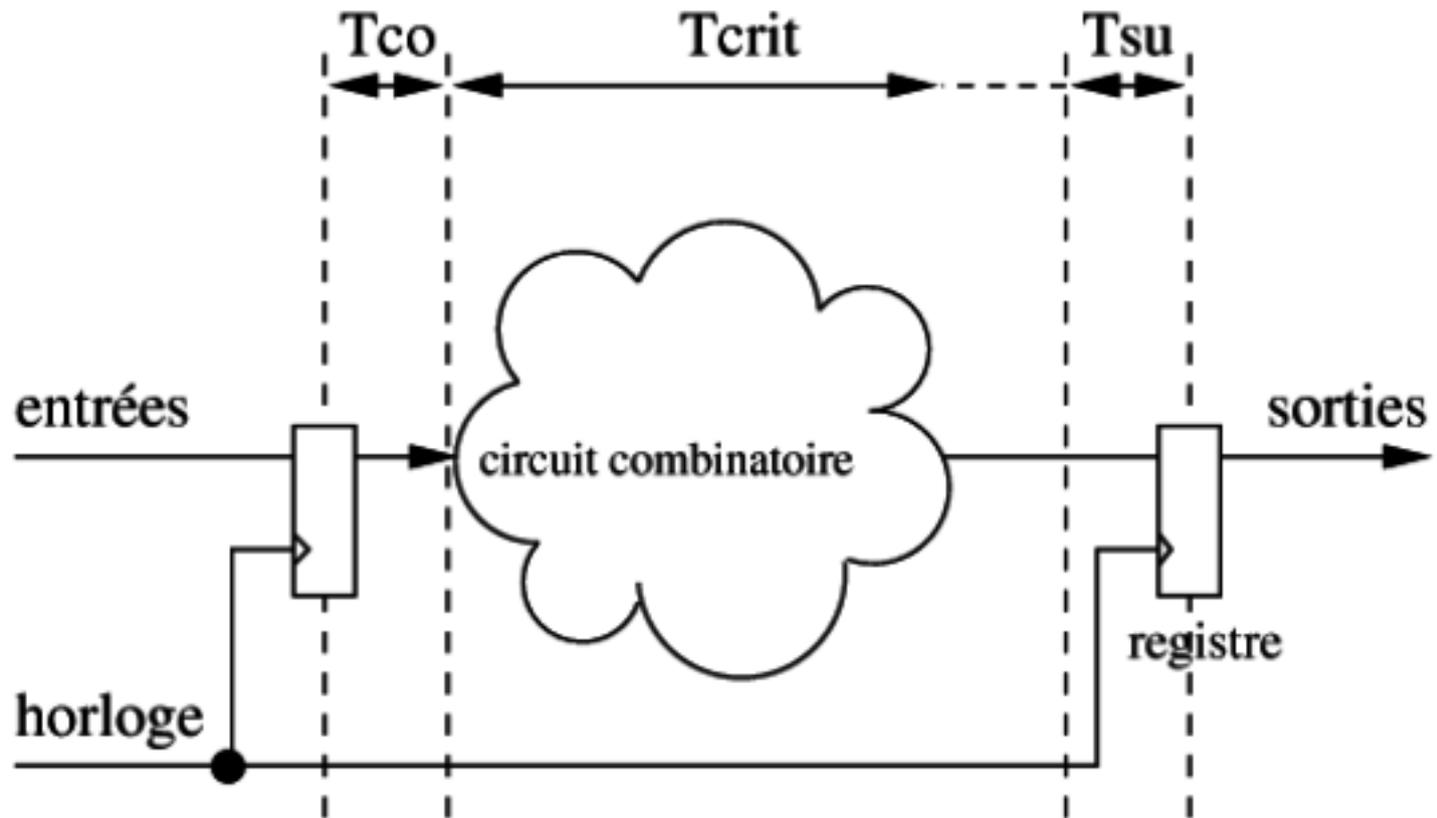
Conditions d'utilisation de la bascule

$$T_h > T_{co} + T_{crit} + T_{su}$$



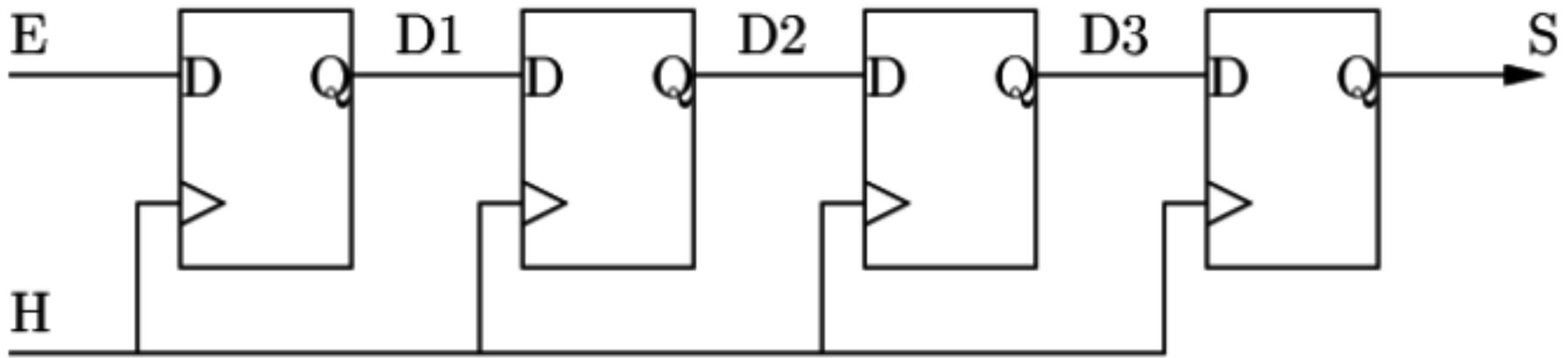
Caractéristiques temporelles de la bascule D

Conditions d'utilisation de la bascule



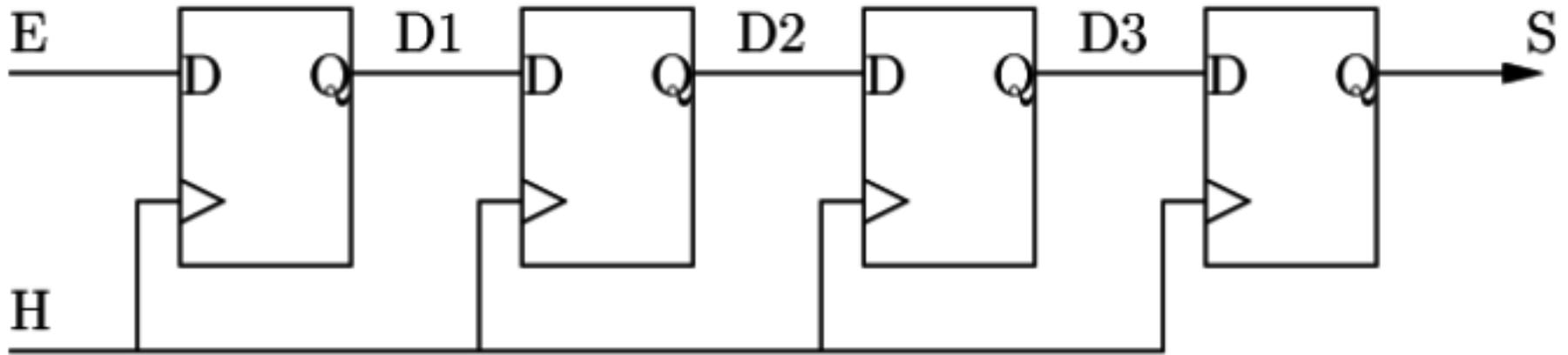
Temps de propagation à considérer en logique séquentielle

Registre à décalage

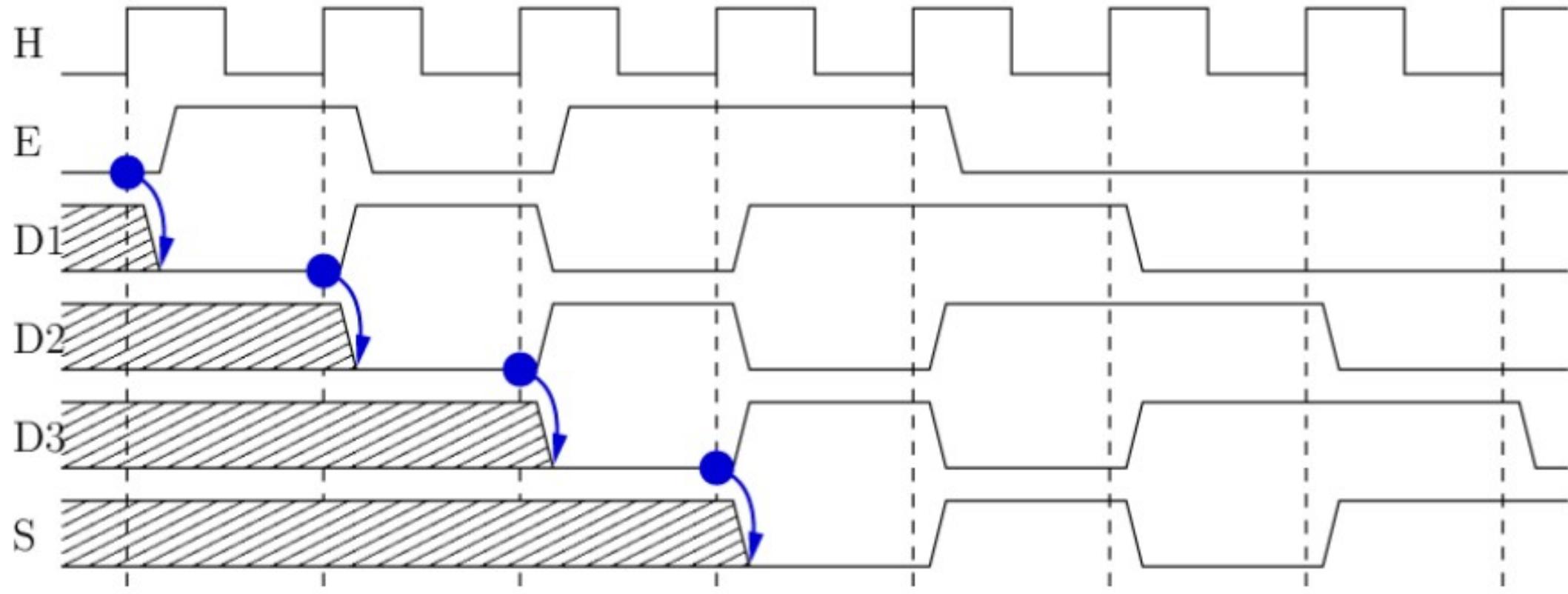


registre à décalage

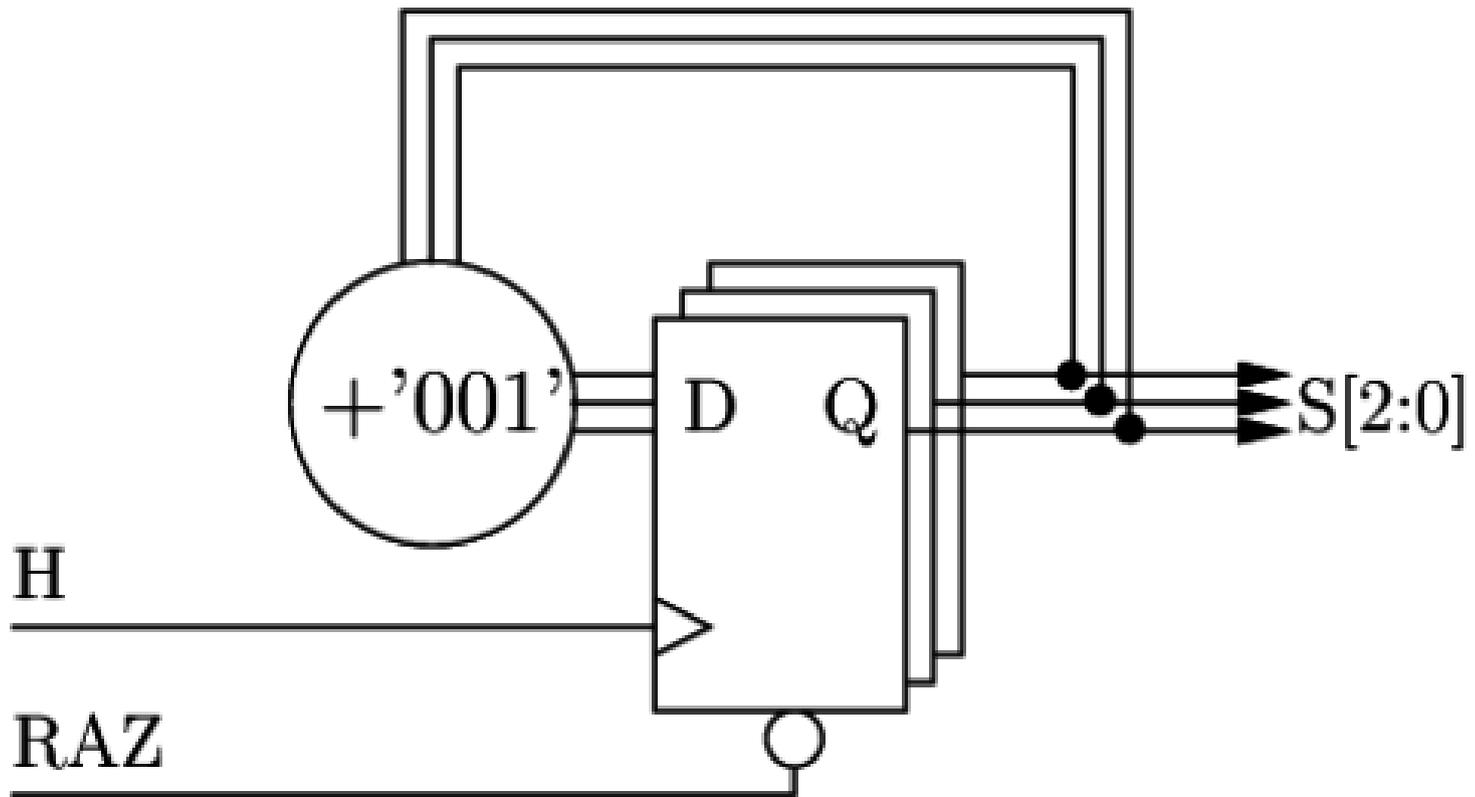
Registre à décalage



Chronogramme du registre à décalage

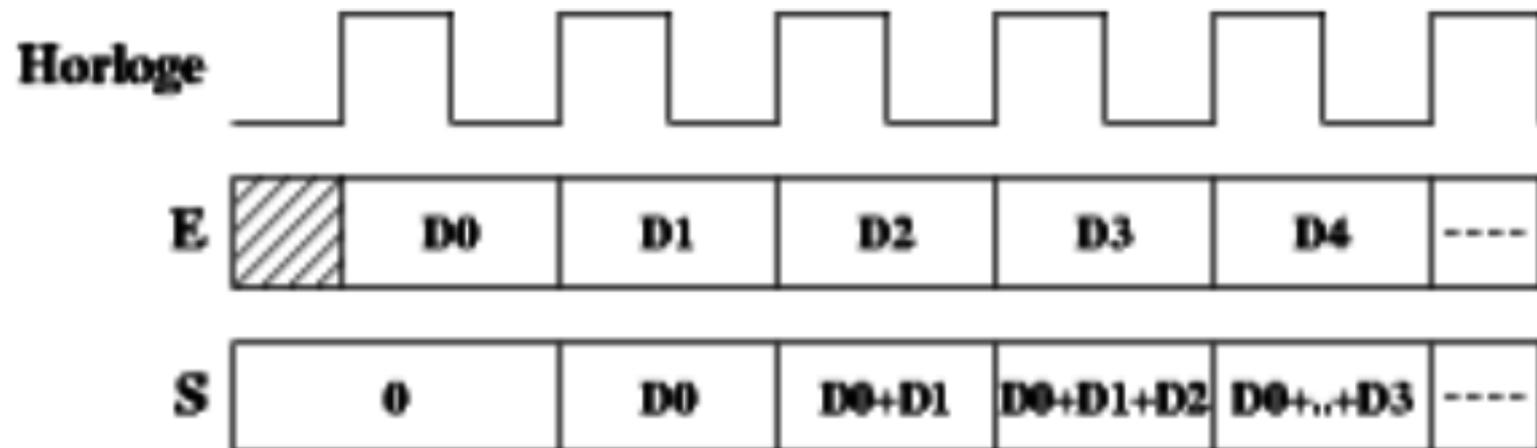


Compteur binaire



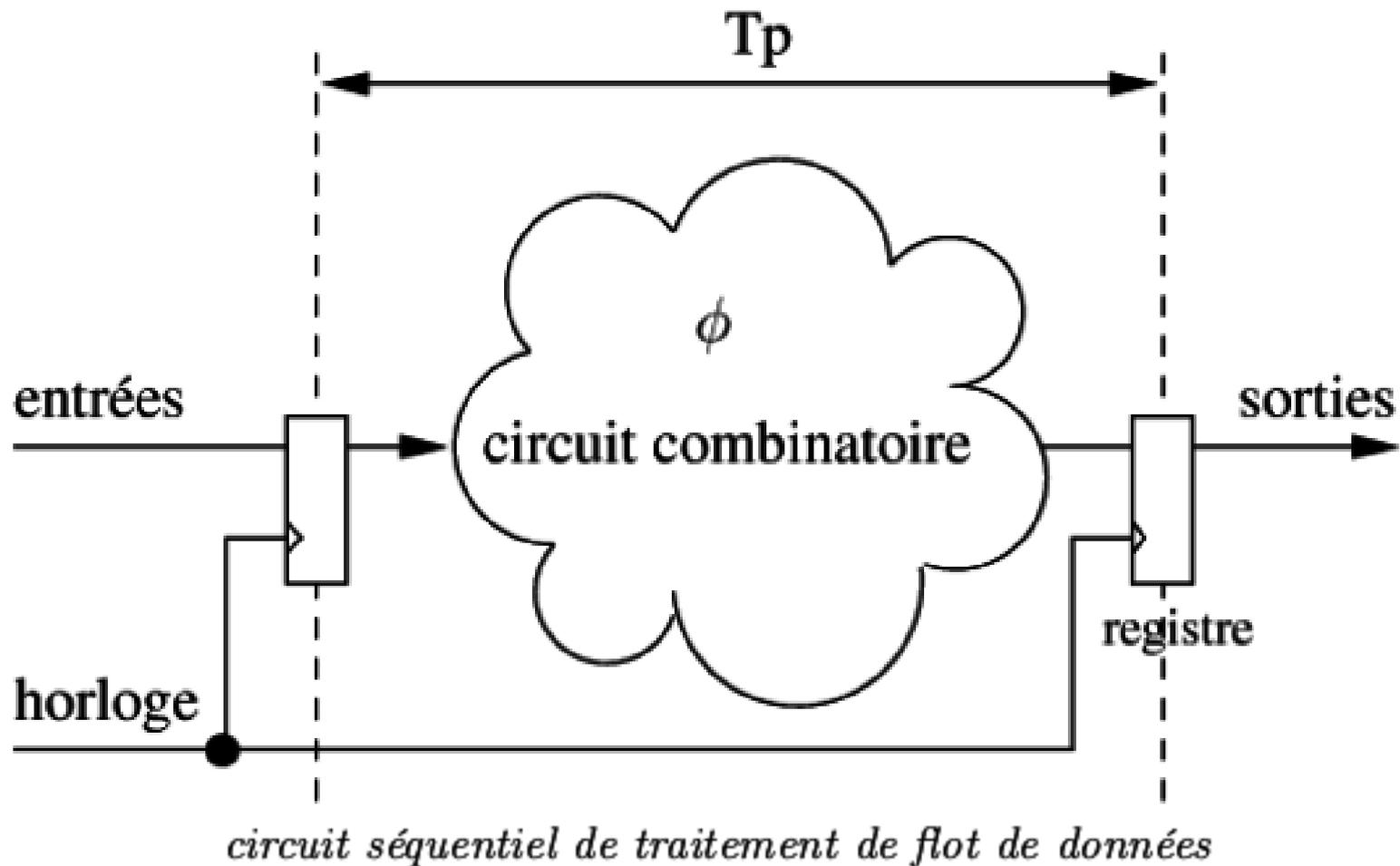
Q : qui a-t'il après 7 ?

Compteur binaire



Chronogramme de l'accumulateur

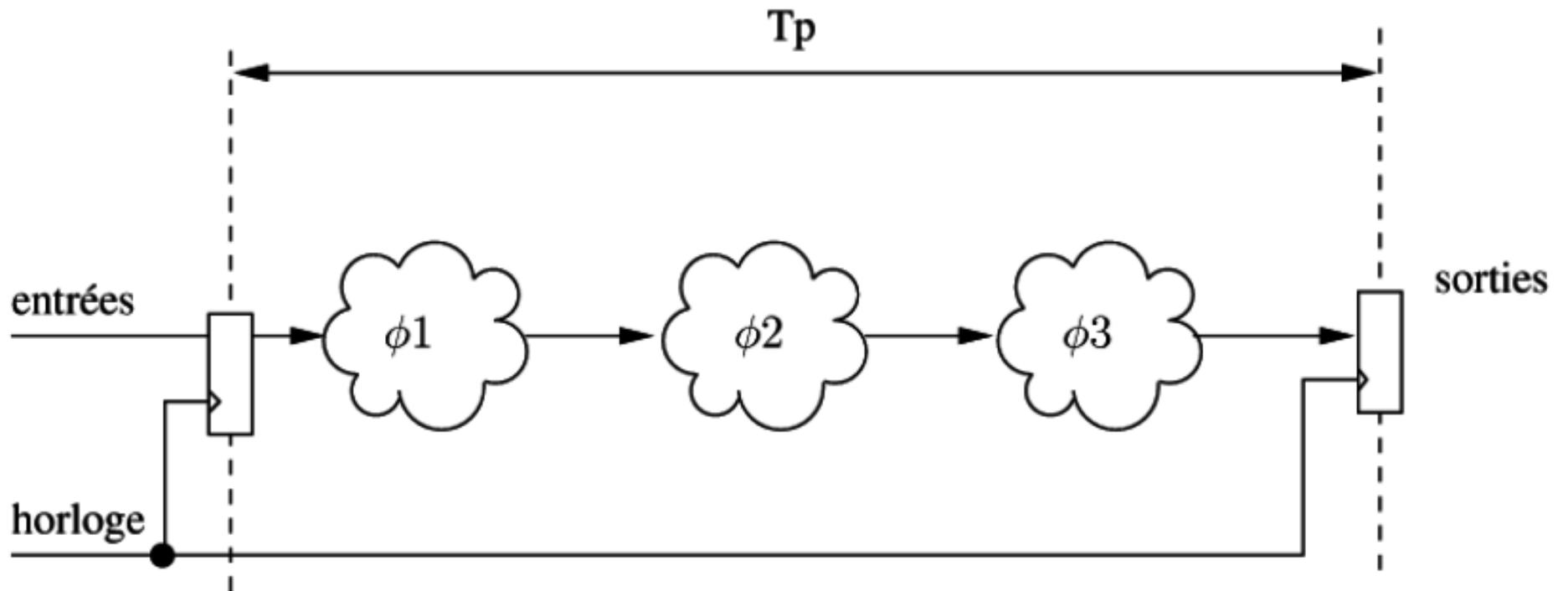
Pipeline



$$T_h = 1/f \text{ et } T_p < 1/f, \text{ où}$$

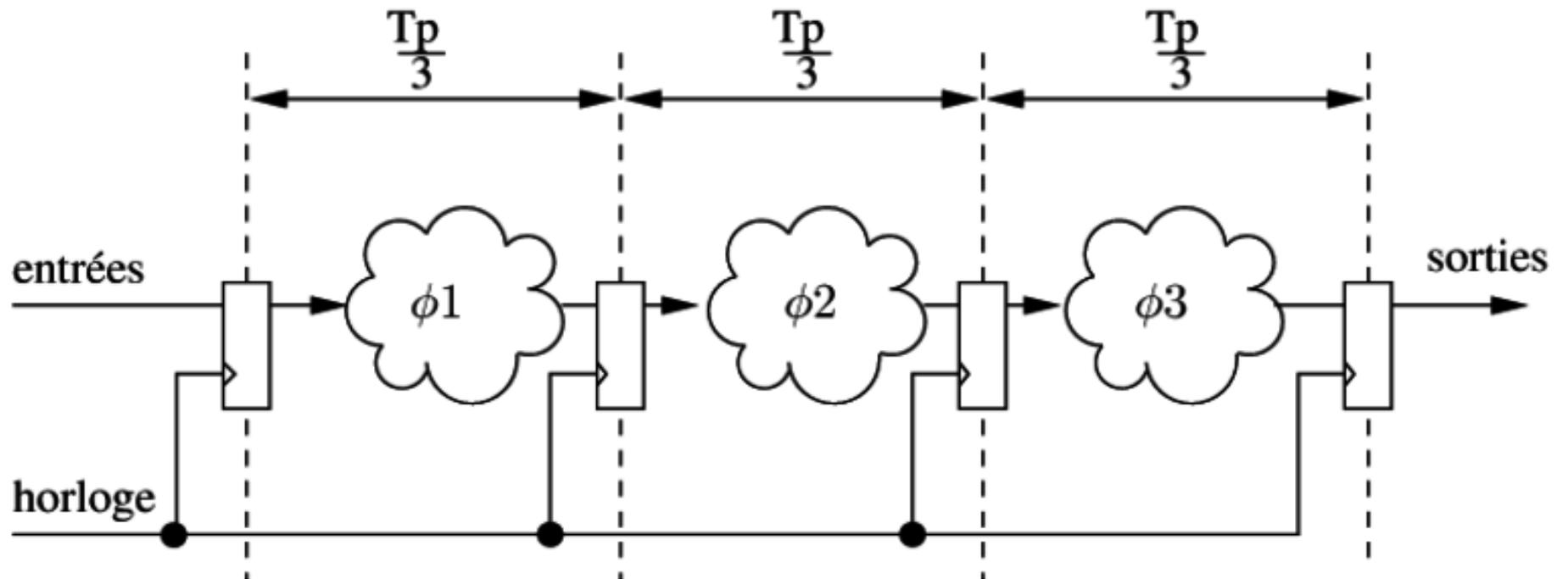
- f est le débit de calcul
- T_h est la période d'horloge
- T_p est le temps de propagation du chemin critique, incluant celui des bascules

Pipeline



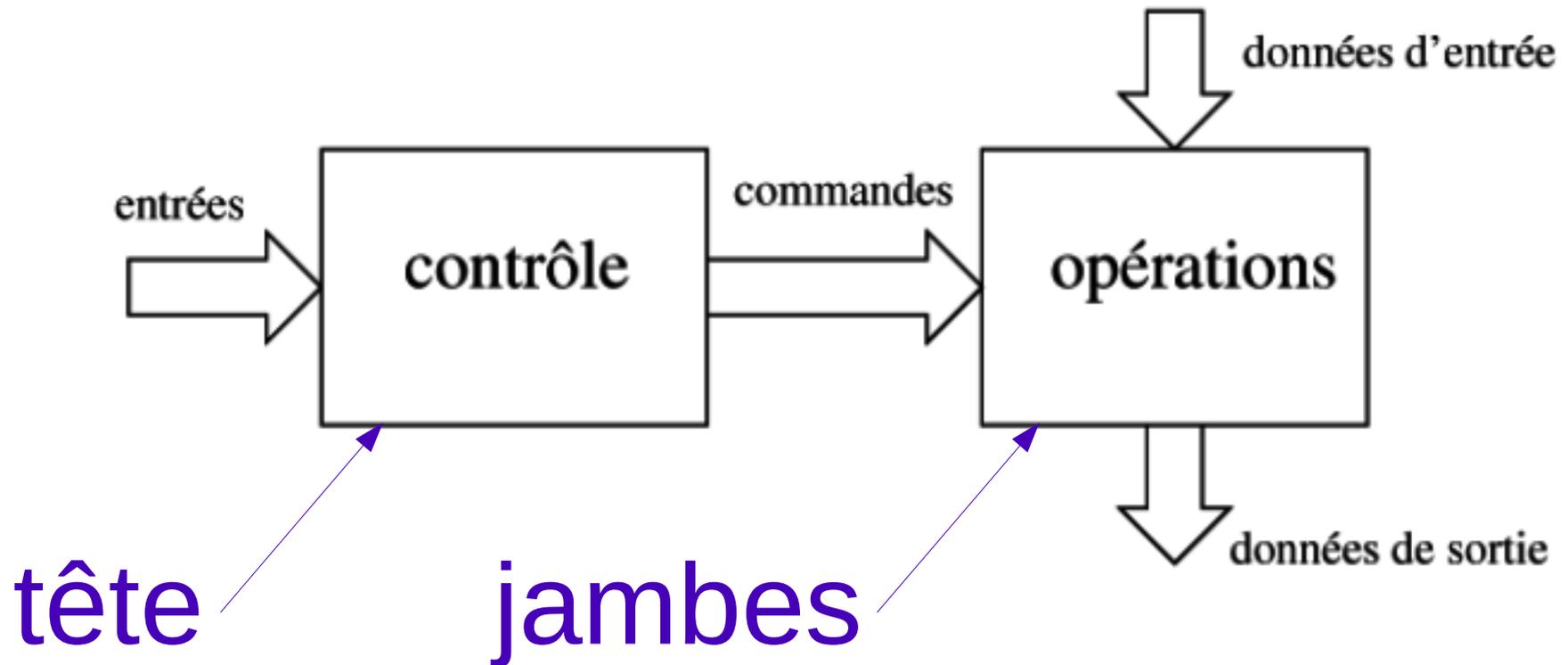
Circuit séquentiel de traitement de flot de données après décomposition en sous fonctions

Pipeline



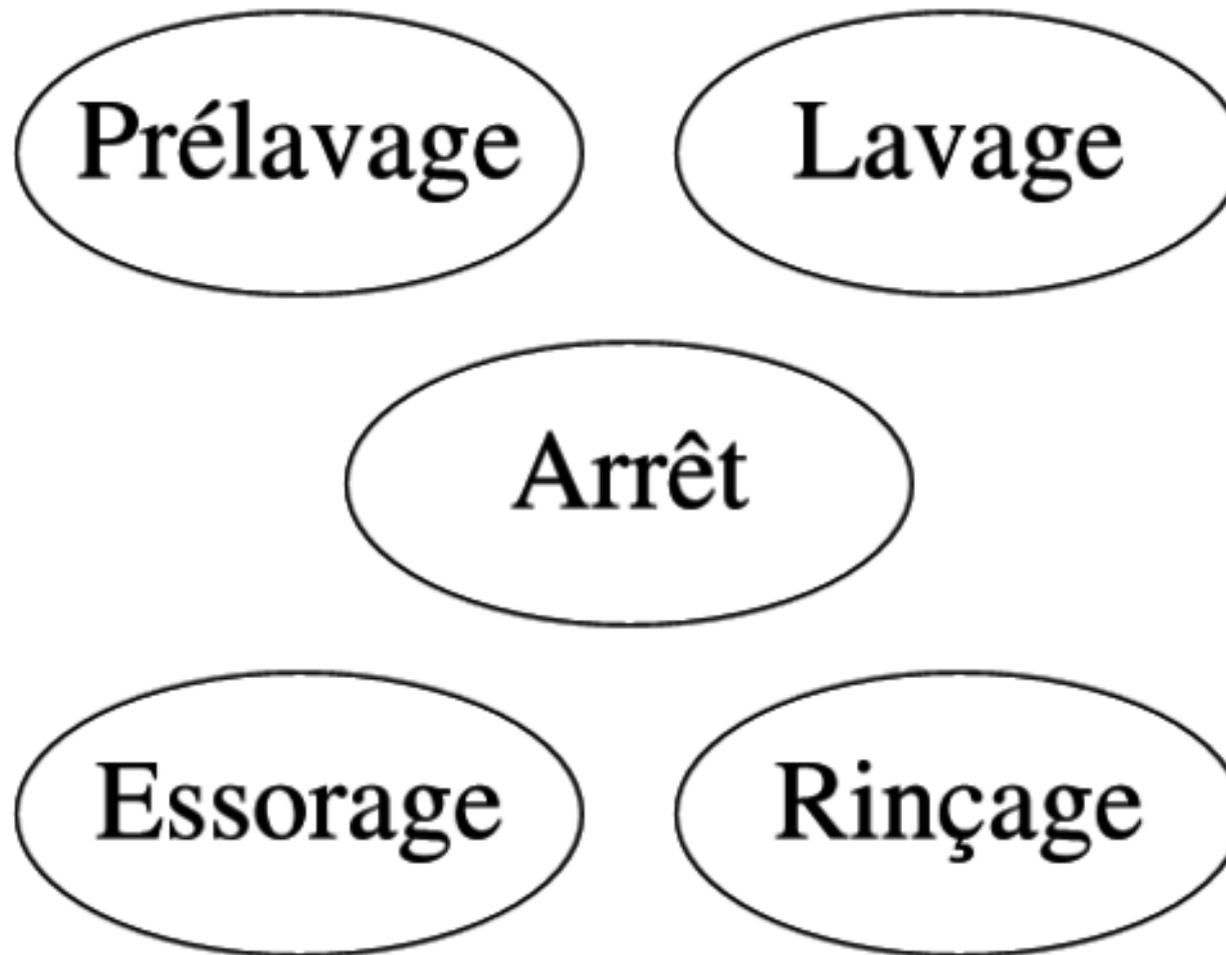
Circuit séquentiel de traitement de flot de données après décomposition en sous fonctions

Machines à états



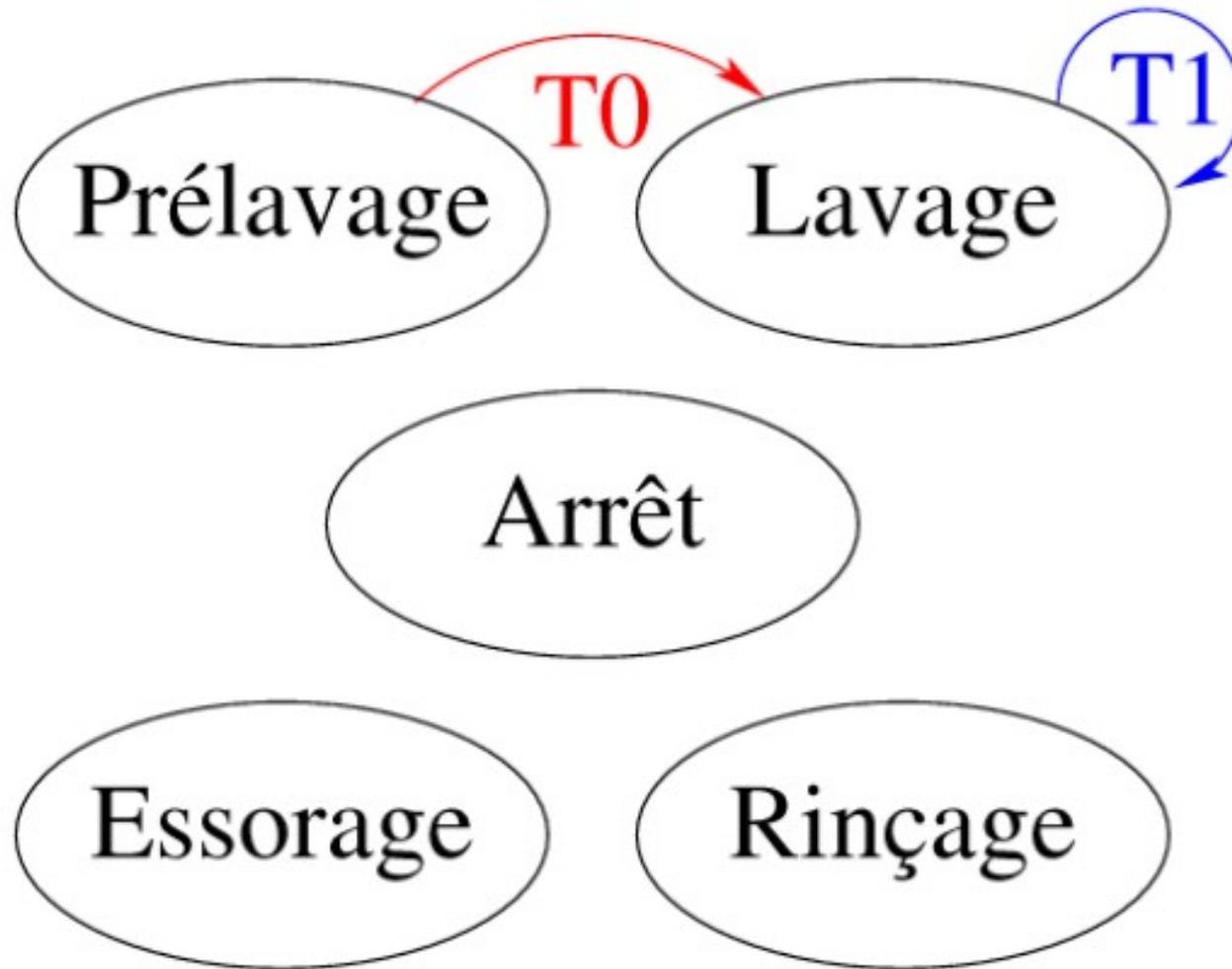
Architecture générique d'un circuit électronique

Exemple



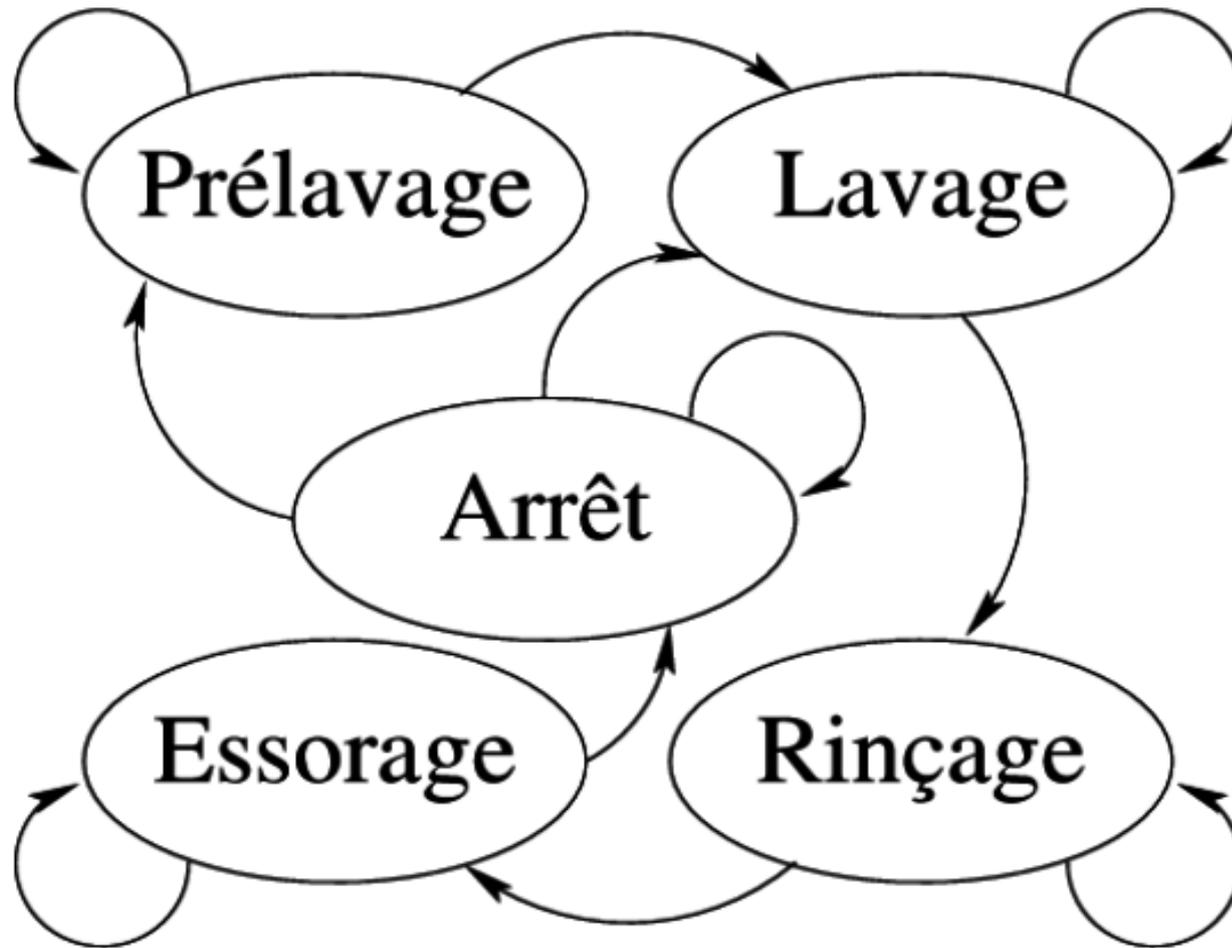
Graphe d'état au départ

Exemple



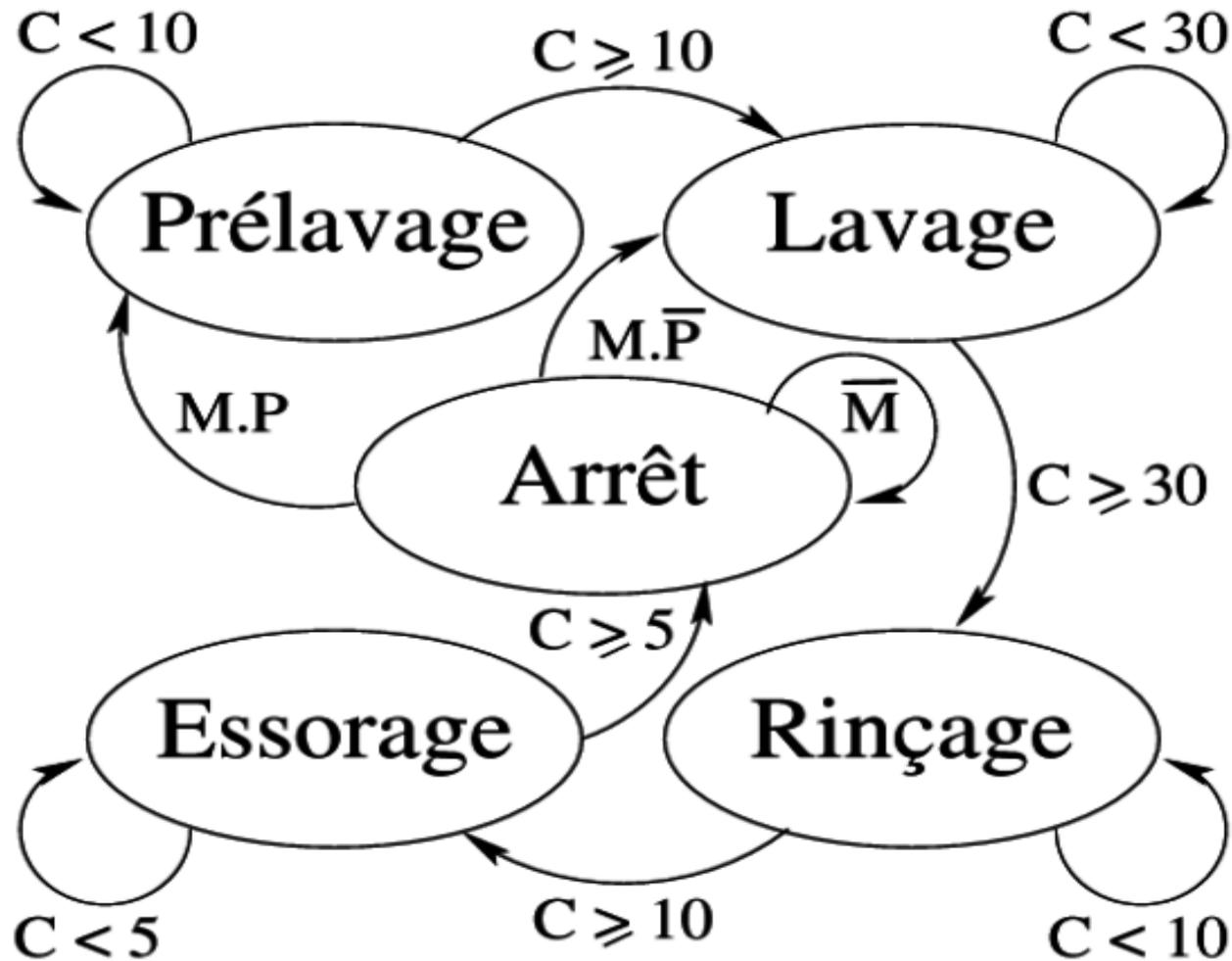
Graphe d'état avec quelques transitions

Exemple



Graphe d'état avec les transitions

Exemple



M :
marche/arrêt

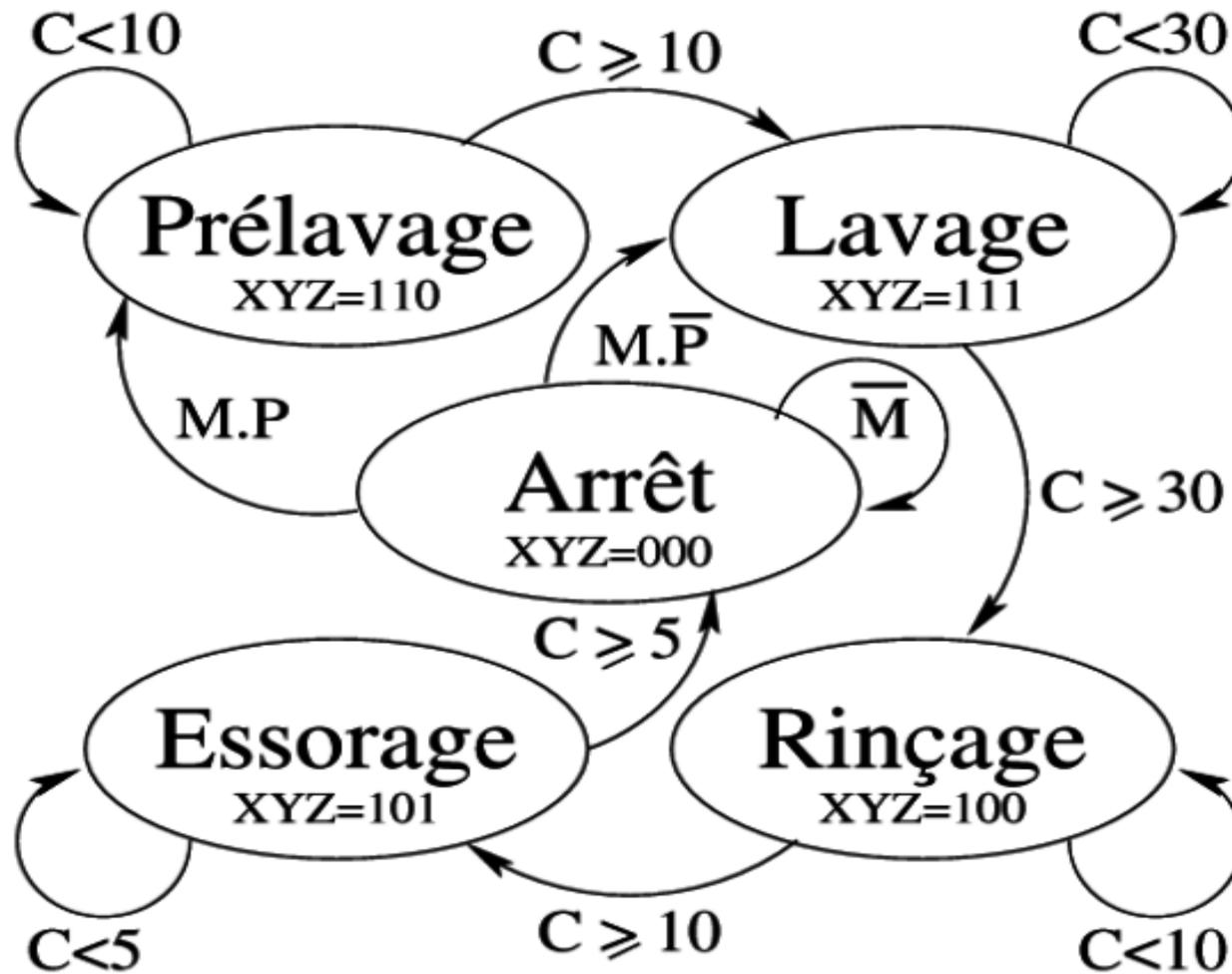
P :
prélavage

C : compteur
réinitialisé au
début de
chaque état

- prélavage : 10 minutes
- lavage : 30 minutes
- rinçage : 10 minutes
- essorage : 5 minutes

Grphe d'état avec les transitions spécifiées

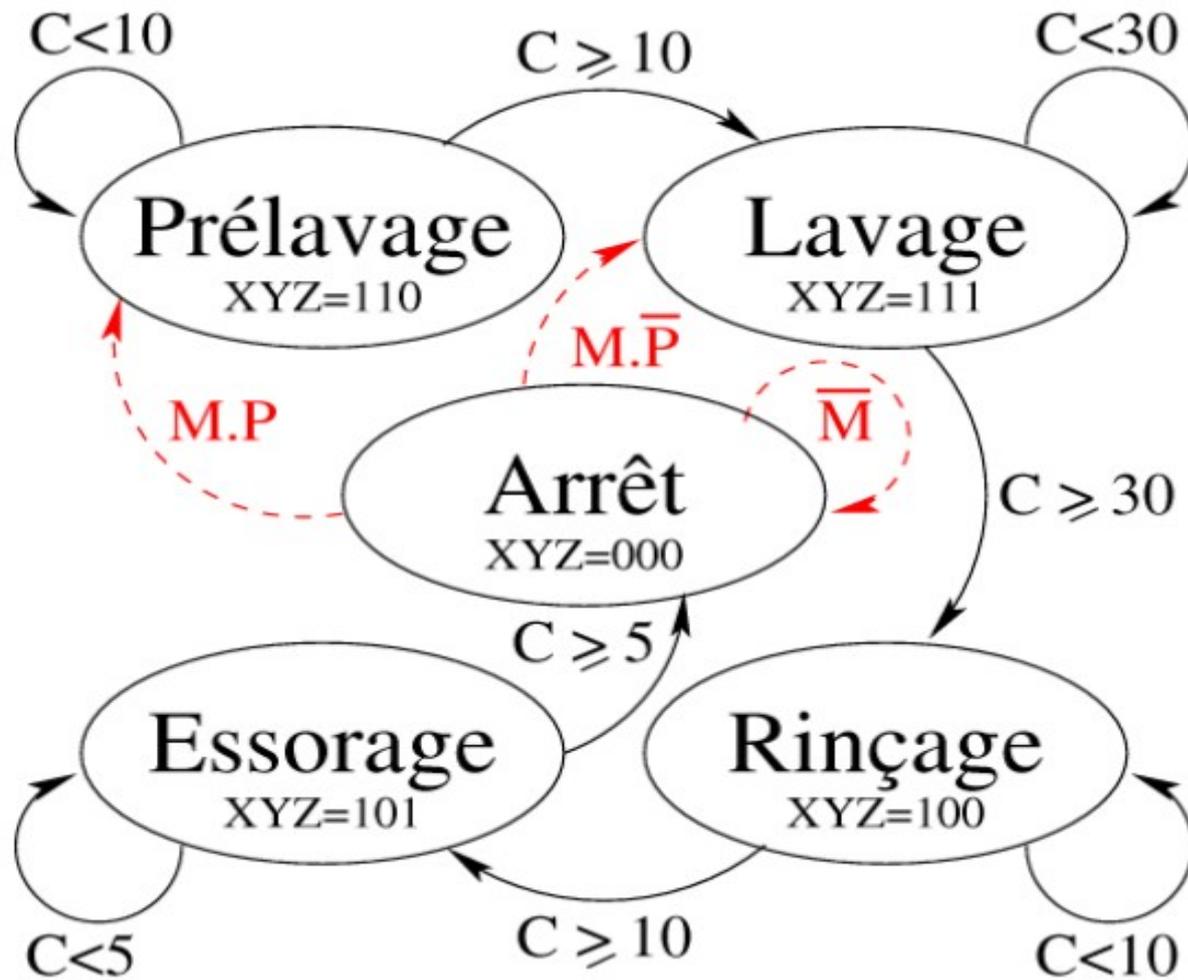
Exemple



Grphe d'état final

X, Y, Z :
Sorties qui
pilotent les
moteurs

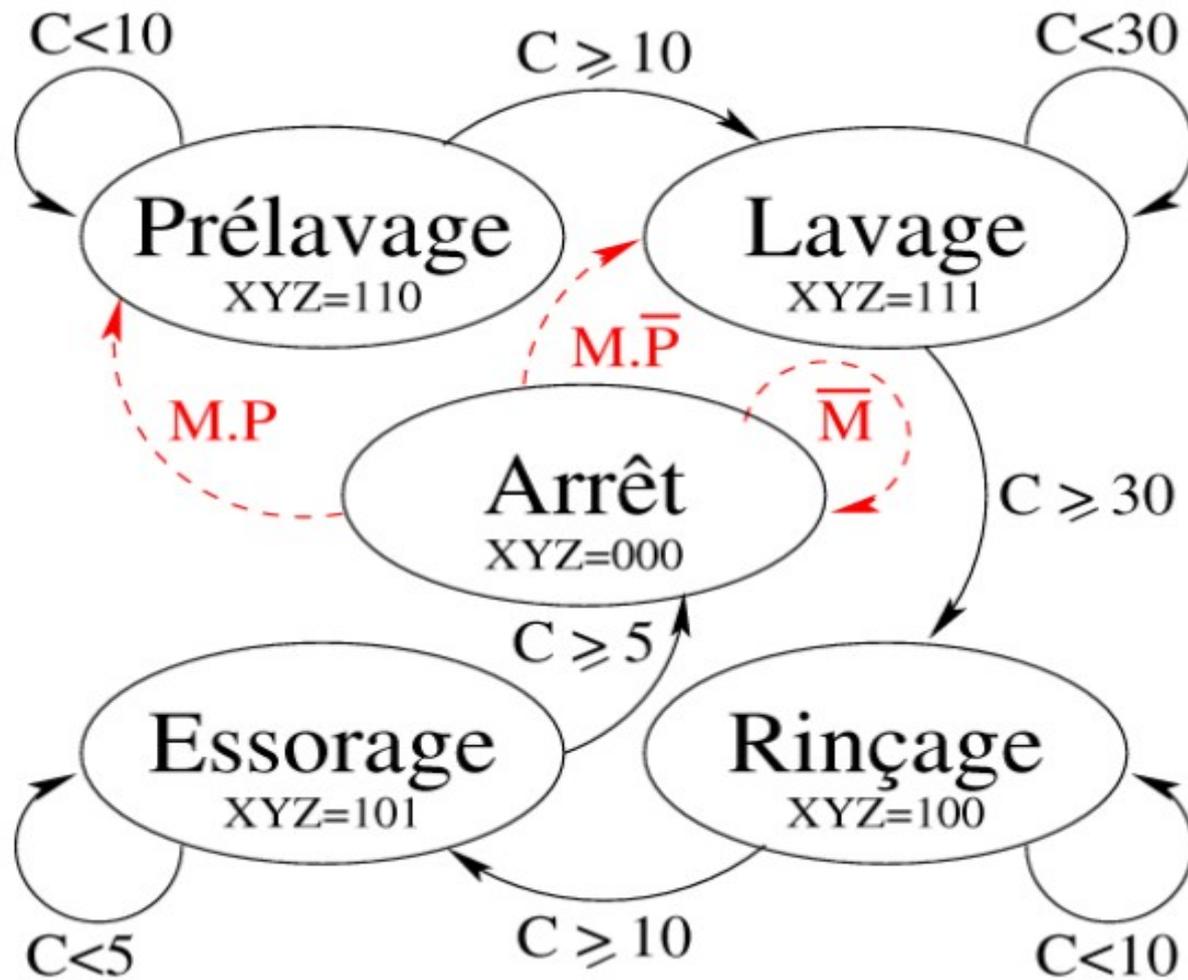
Bonne formation [1/2]



Grphe d'état final

Somme des conditions sortantes = 1

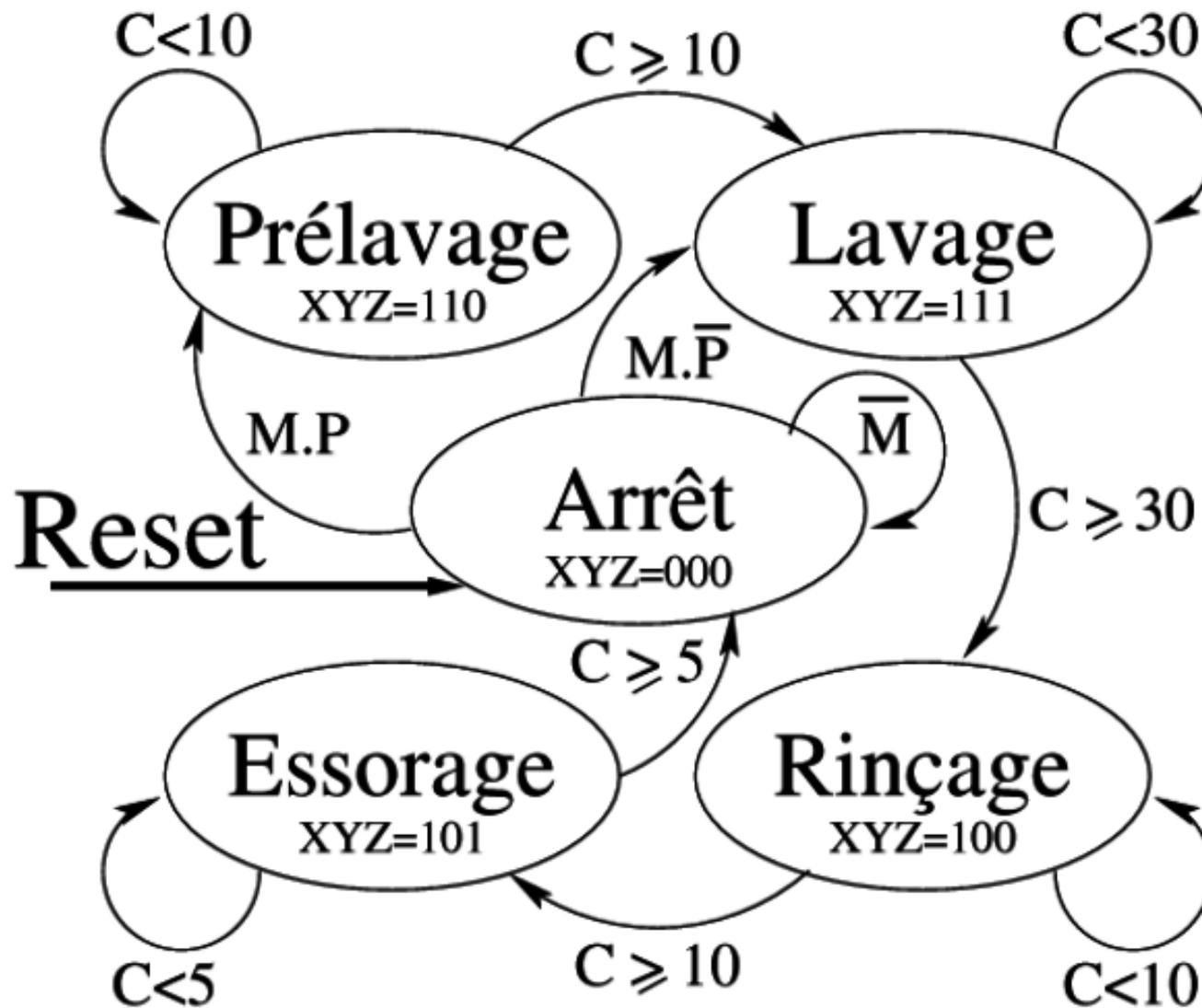
Bonne formation [2/2]



Grphe d'état final

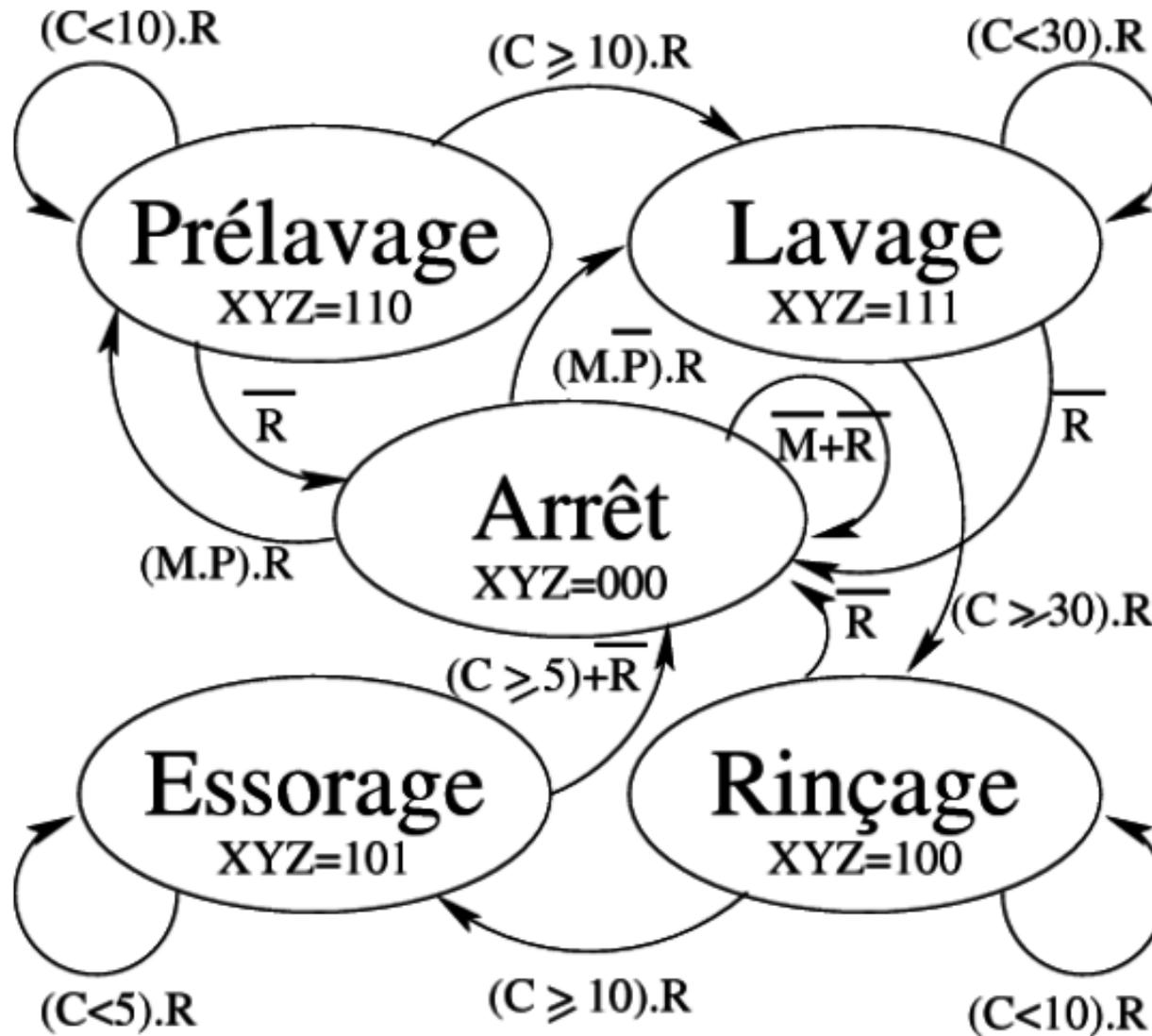
Non-contradictoire.
Q : formaliser ?

Etat initial [1/2]



Grphe d'état avec Reset Asynchrone

Etat initial [2/2]



Graphes d'état avec Reset synchrone

Q : polarité du reset ?

Codage des états

Etat	Trois bits	Quatre bits	Cinq bits	Six 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

Exemples de codage des états

Q1 : est-ce utile d'aller au-delà de 5 bits ?

Q2 : que choisir en technologie FPGA ?

Structure d'une machine à états

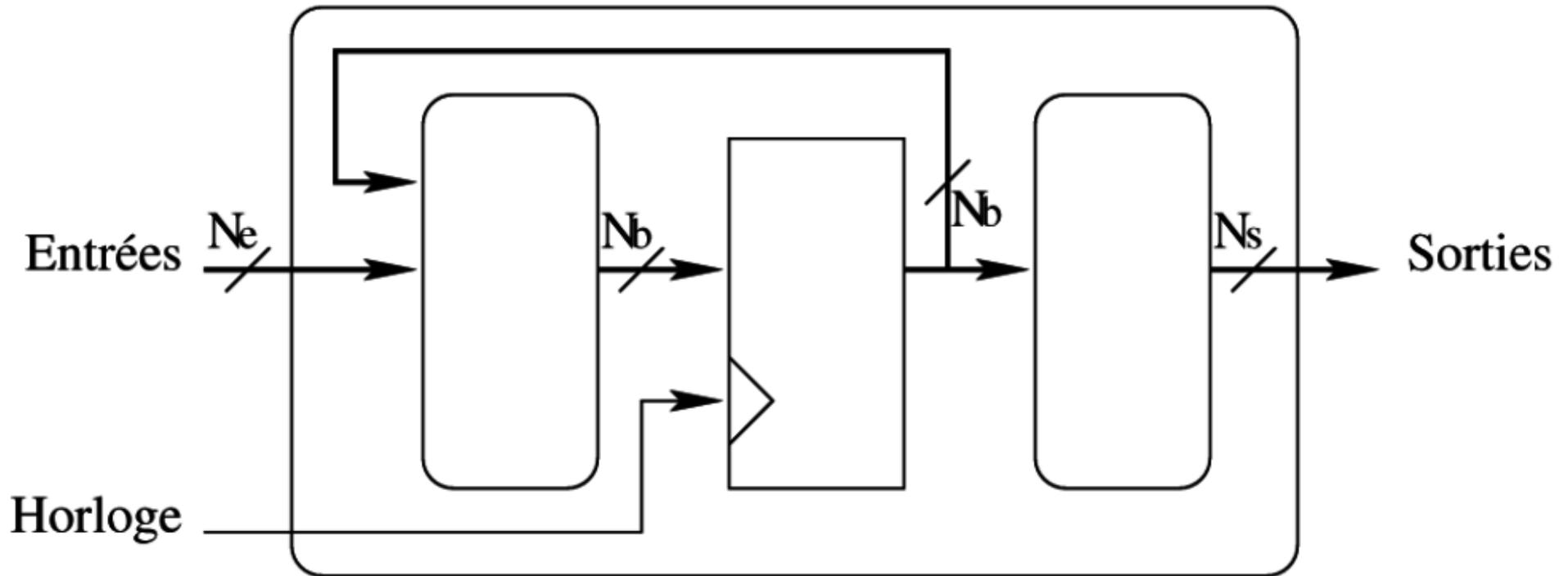
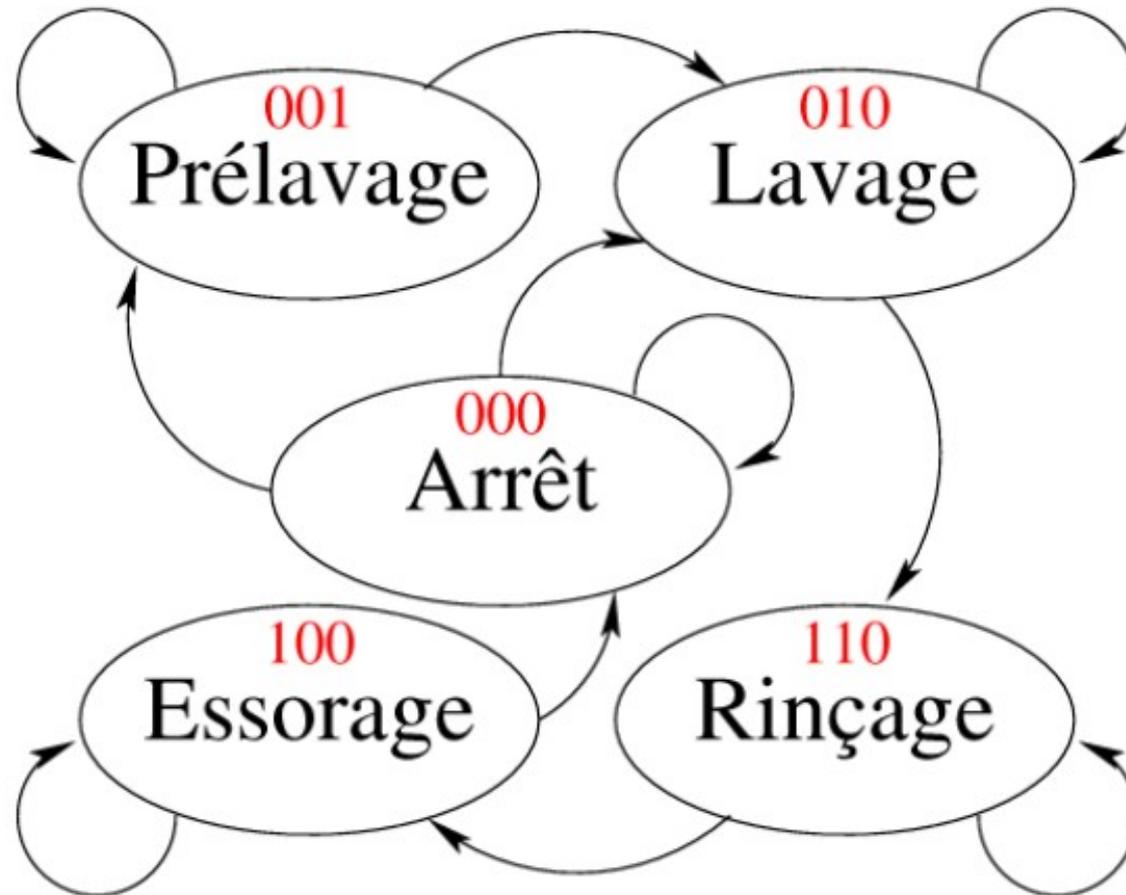


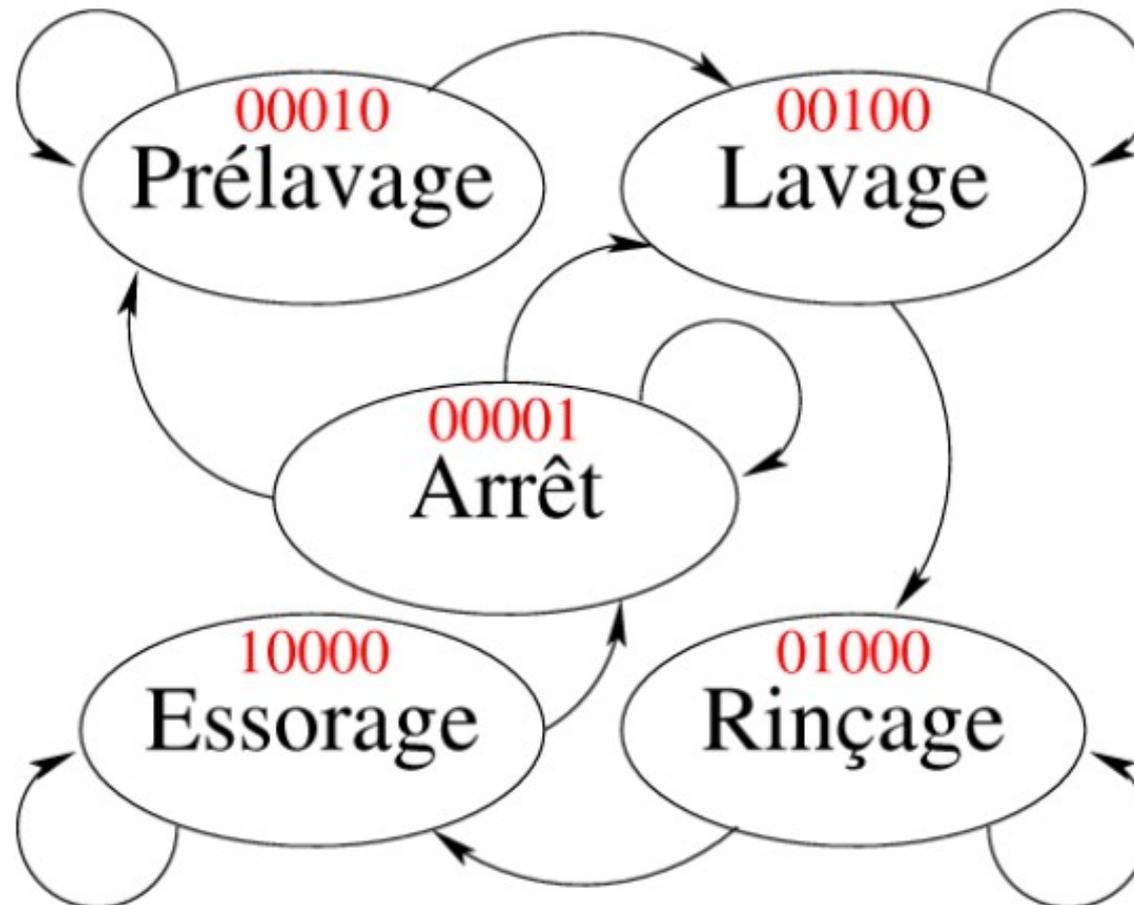
Schéma d'une machine à états avec le nombre de bits nécessaires

Exemple de codages



Graphe avec codage adjacent

Exemple de codages



Graphe avec codage "one-hot"

Table d'évolution

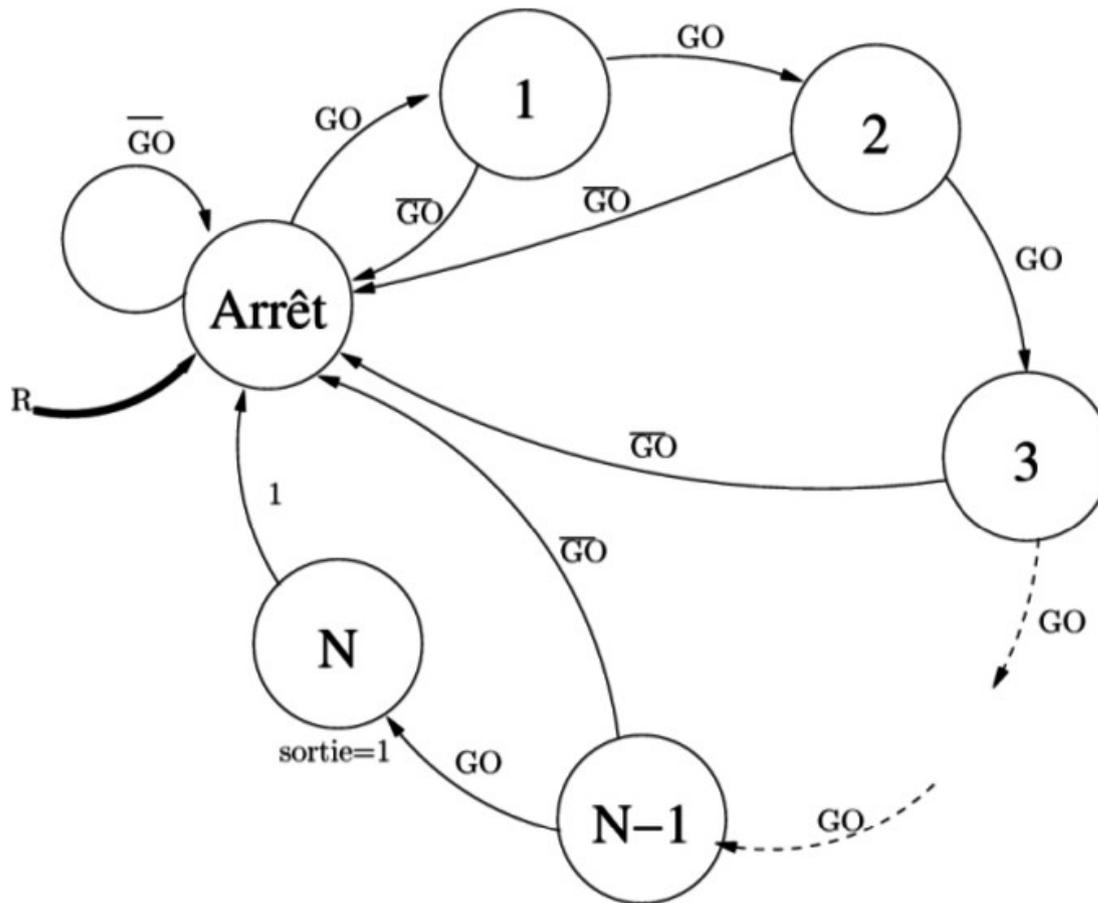
Etat courant			Entrées					Etat futur		
EC2	EC1	EC0	M	P	C5	C10	C30	EF2	EF1	EF0
0	0	0	0	X	X	X	X	0	0	0
0	0	0	1	1	X	X	X	0	0	1
0	0	0	1	0	X	X	X	0	1	0
0	0	1	X	X	X	0	X	0	0	1
0	0	1	X	X	X	1	X	0	1	0
0	1	0	X	X	X	X	0	0	1	0
0	1	0	X	X	X	X	1	1	1	0
1	1	0	X	X	X	0	X	1	1	0
1	1	0	X	X	X	1	X	1	0	0
1	0	0	X	X	0	X	X	1	0	0
1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

$$EF_2 = EC_1.C30 + EC_2.(EC_1 + \overline{C5})$$

$$EF_1 = EC_0.C10 + EC_1.\overline{EC_2}.C10 + \overline{EC_2}.\overline{EC_1}.\overline{EC_0}.M.P$$

$$EF_0 = \overline{EC_2}.\overline{EC_1}.\overline{EC_0}.M.P + EC_0.\overline{C10}$$

Minuteur



Graphe avec codage choisi pour la conception

N = 5, 10 ou 30

Minuteur

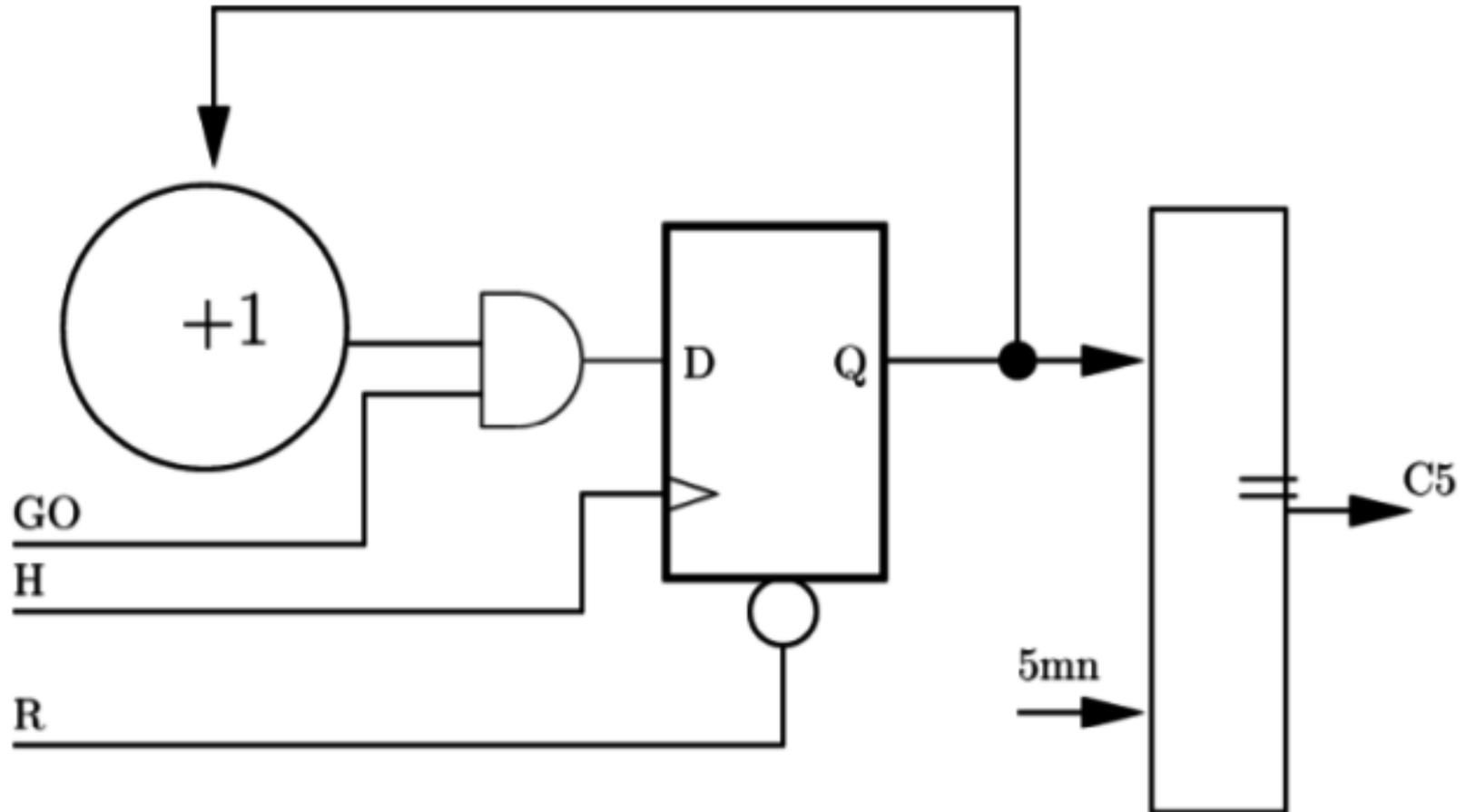


Schéma d'un minuteur

Programmation du minuteur

SEL1	SEL0	Mode de programmation
0	0	5mn
1	x	10mn
0	1	30mn

Minuteur automatique

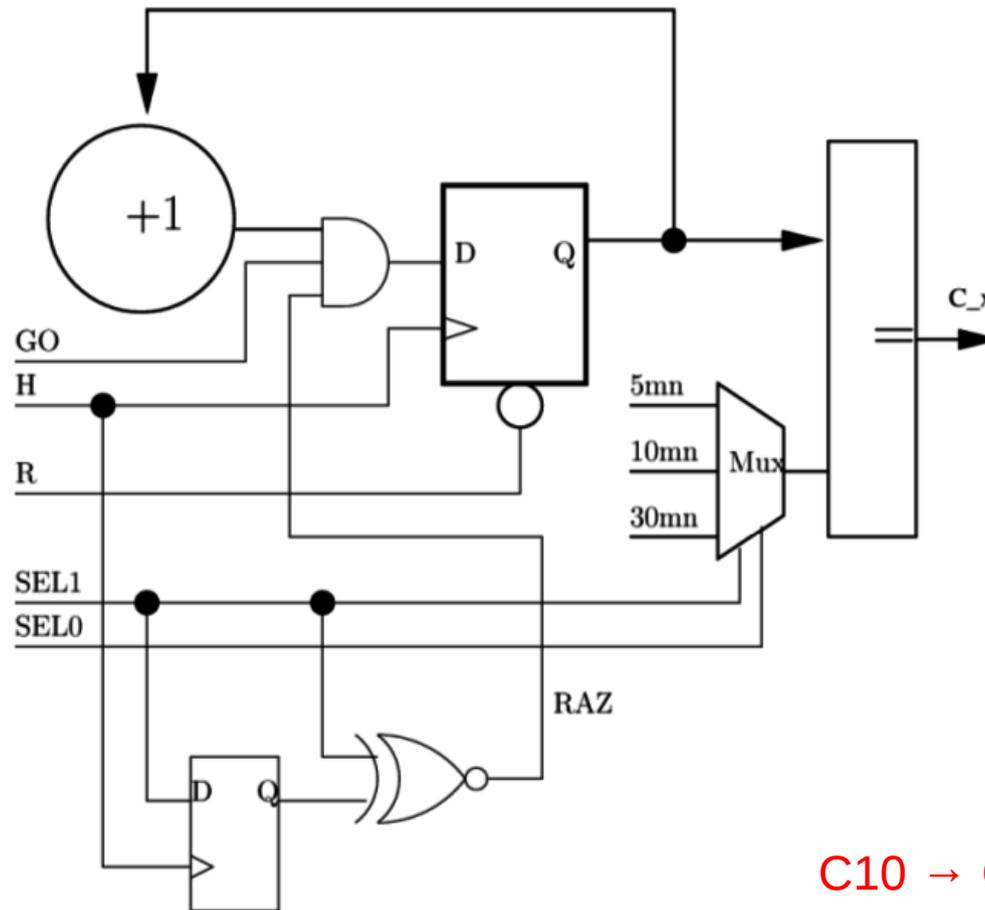


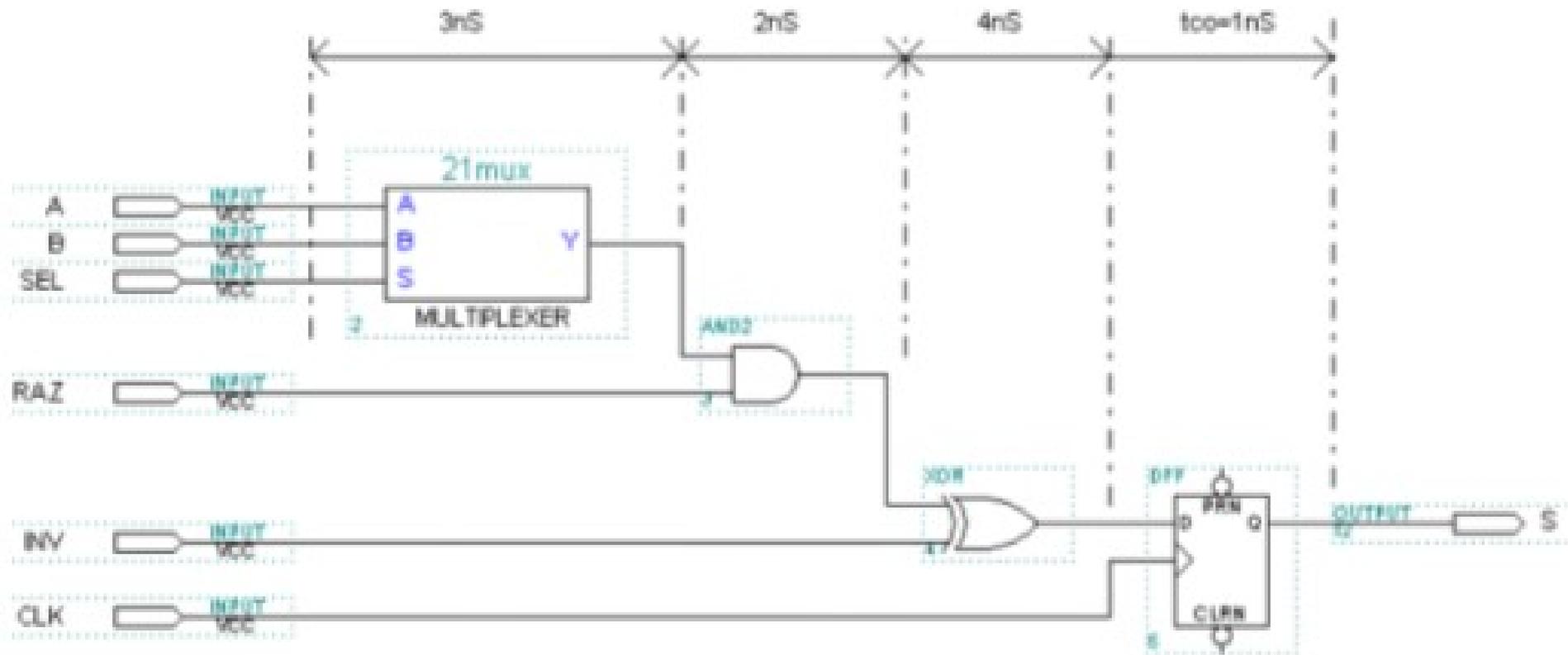
Schéma d'un minuteur générique avec RAZ automatique

TD1 : utilisation des bascules

Programme :

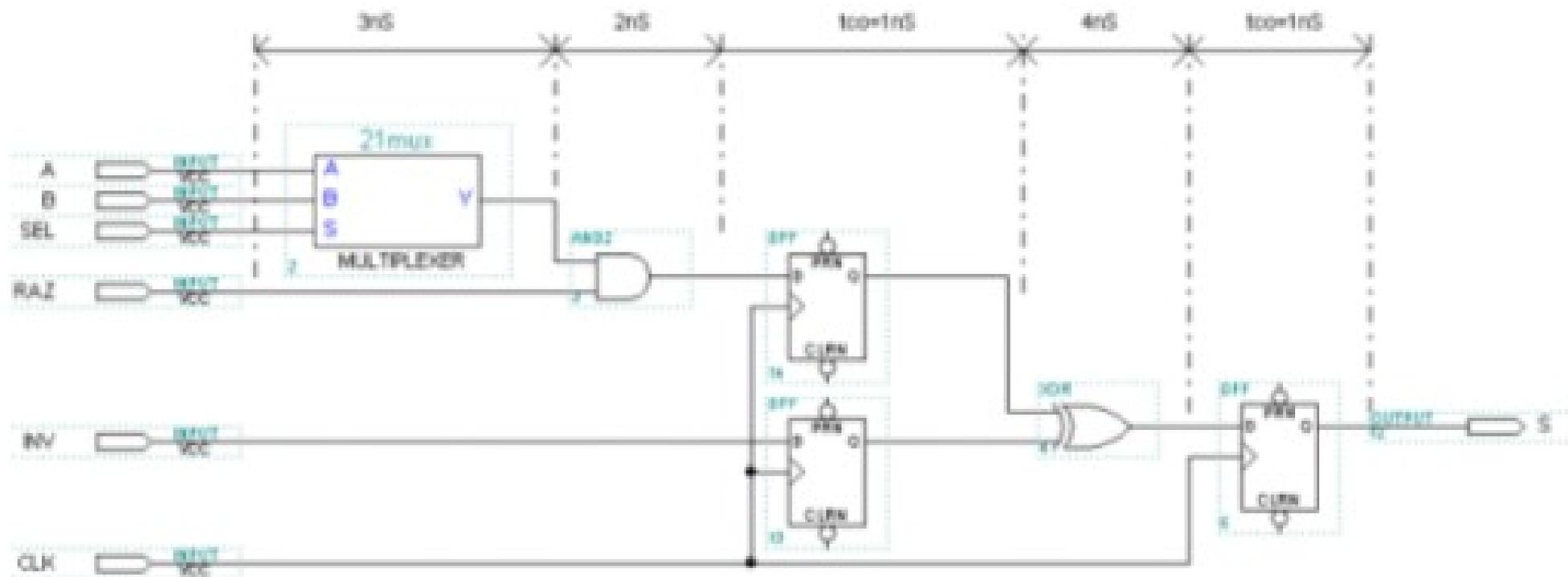
1. Mise en pipeline d'une fonction combinatoire
2. Changement de format série ↔ parallèle
3. Calcul de parité

TD1.1 : Mise en pipeline



1. Quelles sont les rôles respectifs des entrées SEL, RAZ et INV ?
2. On considère que toutes les entrées sont issues de bascules ayant un temps de propagation t_{co} de 1 ns et que les temps de prépositionnement t_{su} et t_{ho} des bascules sont négligeables. Quelle est la fréquence d'échantillonnage maximum f_{max} de la fonction ?

TD1.1 : Mise en pipeline

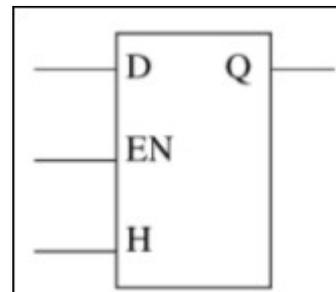


1. Quelle est la nouvelle fréquence maximum f_{max} de fonctionnement ?
2. Quels sont le retard et la latence du signal de sortie ?

TD1.2 : Série ↔ Parallèle

EN=0	Gel de la sortie Q
EN=1	bascule D avec entrées sur D et sortie sur Q

Spécifications de D-EN

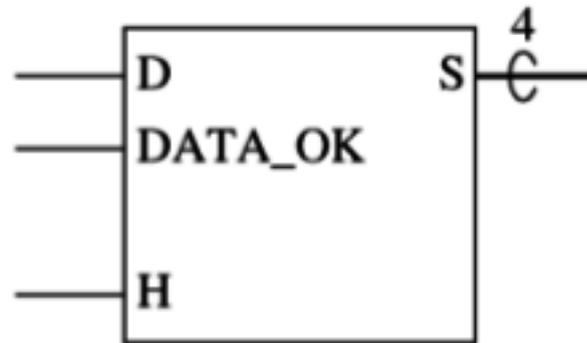


Bascule D-EN

TD1.2 : Série ↔ Parallèle

DATA_OK=0	Gel de la sortie S
DATA_OK=1	La sortie S prend les 4 derniers bits de D

Spécifications de SER-PAR

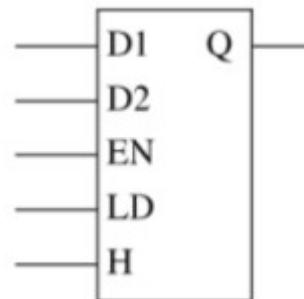


Composant SER-PAR

TD1.2 : Série ↔ Parallèle

EN=0	gel de la sortie Q
EN=1 et LD =0	bascule D avec entrée sur D1 et sortie sur Q
EN=1 et LD =1	bascule D avec entrée sur D2 et sortie sur Q

Spécifications de D-EN-LD

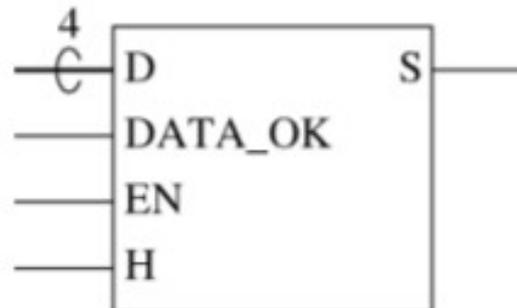


Bascule D-EN-LD

TD1.2 : Série ↔ Parallèle

EN=0	Gel de la sortie S
EN=1 et DATA_OK=0	la sortie S sort d'une façon cyclique les 4 bits de D enregistrées
EN=1 et DATA_OK=1	Les entrées D sont enregistrées

Spécifications de PAR-SER



Composant PAR-SER

TD1.3

a) Calcul de parité sur un mot parallèle

1. Concevez un circuit calculant la parité d'un mot de 8 bits en parallèle, à partir exclusivement de portes XOR et ayant un temps de propagation minimal.

b) Calcul de parité sur un mot série

Les bits du mot arrivent maintenant en série et d'une manière synchrone avec une horloge CLK. Le signal DEB est actif juste avant le premier bit. Le chronogramme de la donnée et de DEB est donné en figure 13.7.

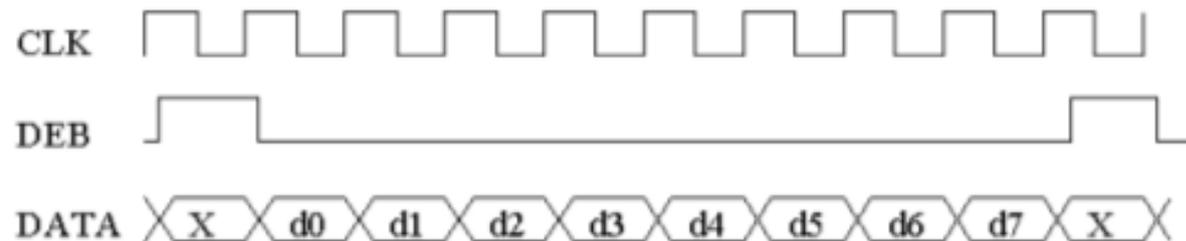


FIG. 13.7: Chronogramme des entrées du calculateur

Comme les bits arrivent séquentiellement, on peut utiliser une structure simple calculant la parité bit après bit et stockant le résultat dans une bascule D. La bascule doit être initialisée à l'aide du signal DEB et contient le résultat au bout des 8 bits.

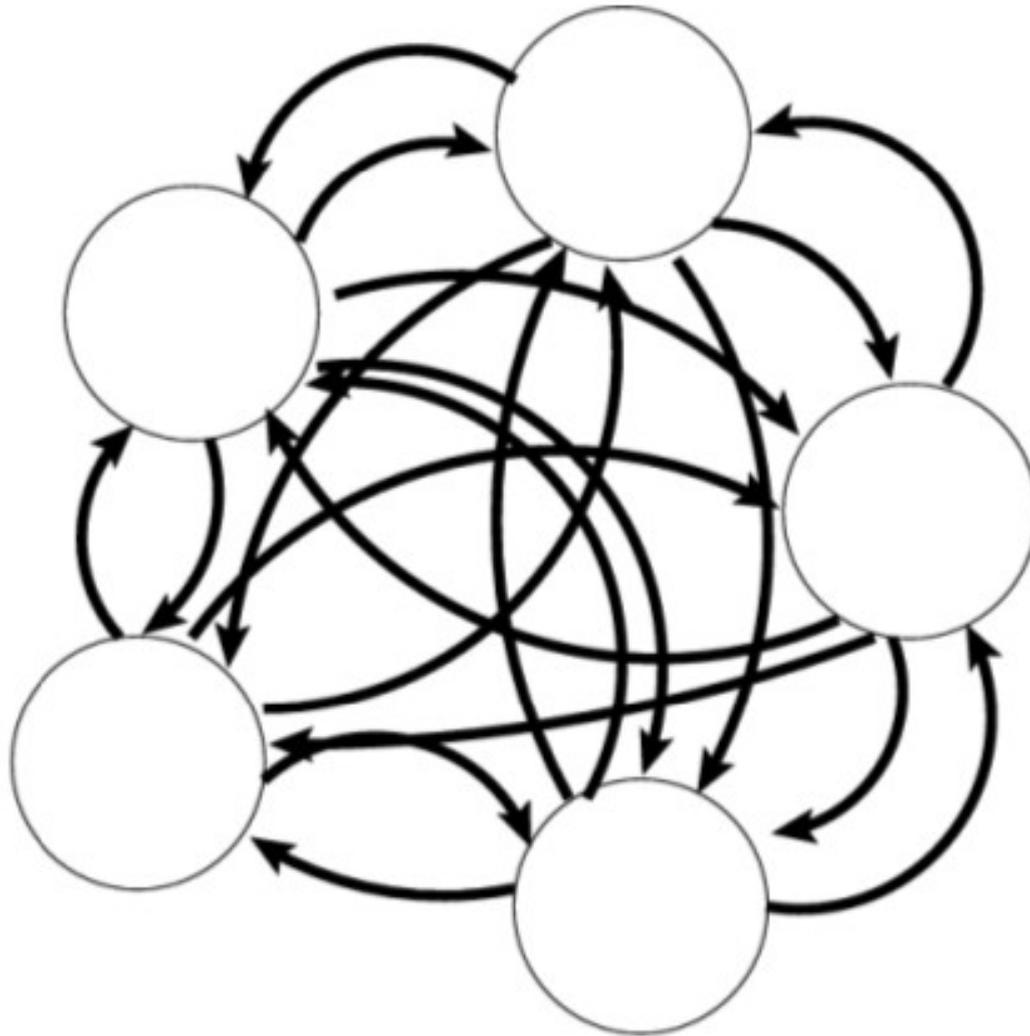
1. Donnez l'équation de l'entrée D de la bascule par rapport à DATA, DEB et la sortie de la bascule.
2. Déduisez la structure du circuit séquentiel correspondant.

TD2 : synthèse et utilisation des machines à état synchrones

Programme :

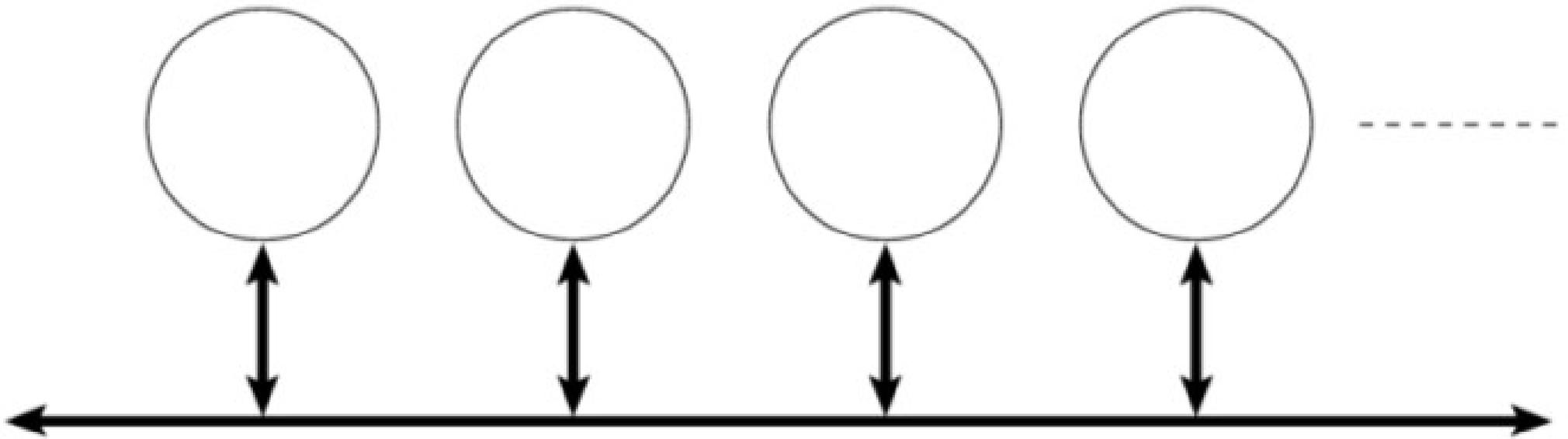
1. Etude et conception d'un contrôleur de bus simple
2. Prise en compte du problème de l'équité

Pourquoi un bus de communication ?



Liaisons point à point

Pourquoi un bus de communication ?

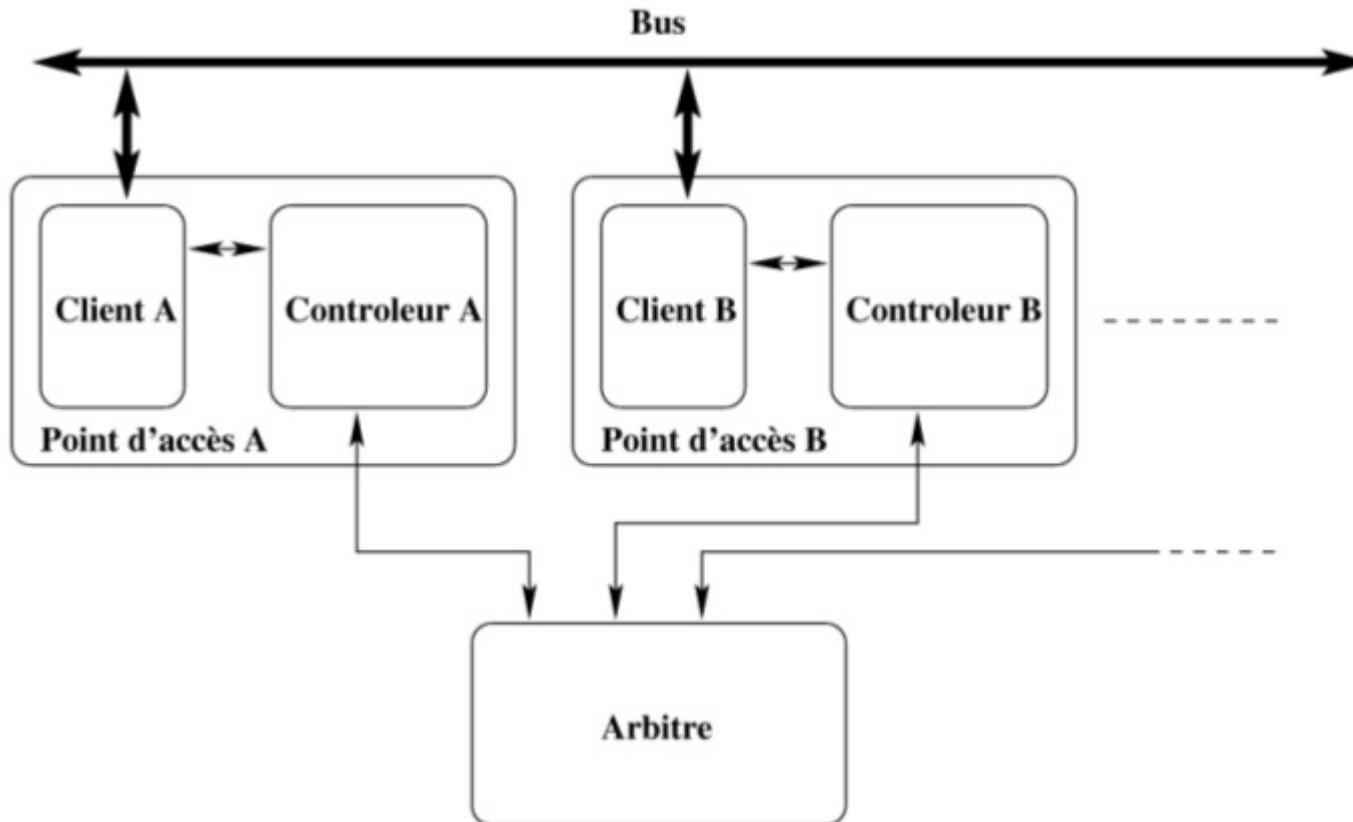


Bus central partagé

Architecture

Systeme :

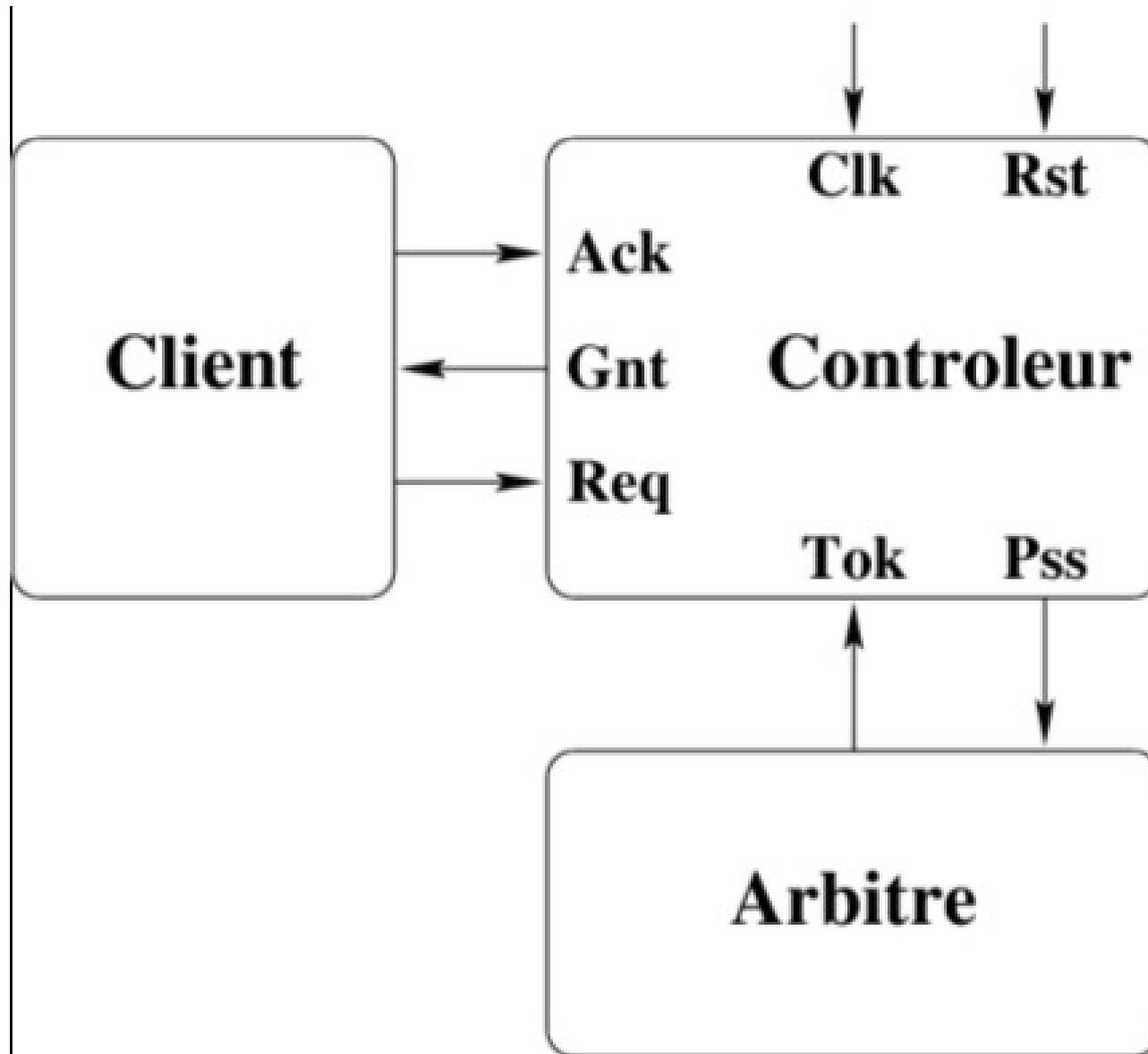
- Un arbitre de bus
- Un nombre indéterminé de points d'accès au bus
 - Chacun comporte un contrôleur et un client



Stratégie d'arbitrage

- Existence d'un unique jeton
 - Seul le module en possession du jeton peut écrire sur le bus
 - Le jeton est rendu après usage du bus
 - Absence de contention garantie par unicité du jeton
- Les clients sont les utilisateurs du bus
 - Ils demandent un jeton et l'attendent avant d'écrire

Contrôleur de communication



Nom	Direction	Description
CLK	Entrée	Horloge pour la synchronisation du contrôleur
RST	Entrée	Signal de reset asynchrone, actif à '0'. Lorsque ce signal est à état bas ('0') le contrôleur est entièrement réinitialisé.
TOK	Entrée	Ce signal provient de l'arbitre et indique que le contrôleur peut disposer du bus. Il signifie donc que l'arbitre offre le jeton au contrôleur. Il n'est actif que pendant une période d'horloge. Si le contrôleur n'a pas besoin du jeton il le rend (voir le signal PSS). Sinon il le garde jusqu'à ce qu'il n'en ait plus l'utilité.
REQ	Entrée	Ce signal est émis par le client et indique que ce dernier demande à disposer du bus. Le client maintient ce signal jusqu'à la fin de sa transaction sur le bus. Il ne le relâche que lorsqu'il n'a plus besoin du bus.
ACK	Entrée	Ce signal provient du client et indique que le client a pris le bus et commence sa transaction. Il n'est actif que pendant une période d'horloge.
PSS	Sortie	Ce signal est destiné à l'arbitre et l'informe que le contrôleur rend le bus, soit parce que l'arbitre le lui a proposé alors qu'il n'en a pas besoin, soit parce que la transaction du client est terminée. Il signifie donc que le contrôleur rend le jeton à l'arbitre qui pourra ensuite en disposer et l'attribuer à un autre contrôleur, voire au même. Il n'est actif que pendant une période d'horloge.
GNT	Sortie	Ce signal est destiné au client et l'informe qu'il peut disposer du bus. Il est maintenu tant que le client n'a pas répondu (par le signal ACK) qu'il a pris le bus.

Travail

- Dessinez un chronogramme représentant une ou plusieurs transactions complètes entre un contrôleur, son client et l'arbitre.
- Le contrôleur est une machine à états de Moore. Imaginez et dessinez son graphe.
- Vérifiez la cohérence du graphe.
- Vérifiez que les spécifications du contrôleur sont respectées par votre graphe.

Optimisation

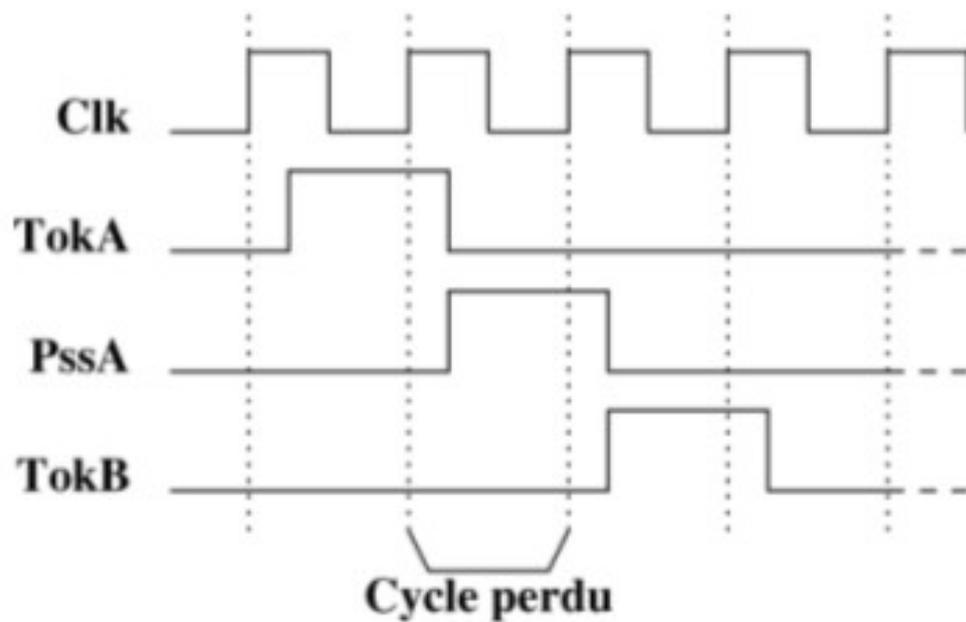
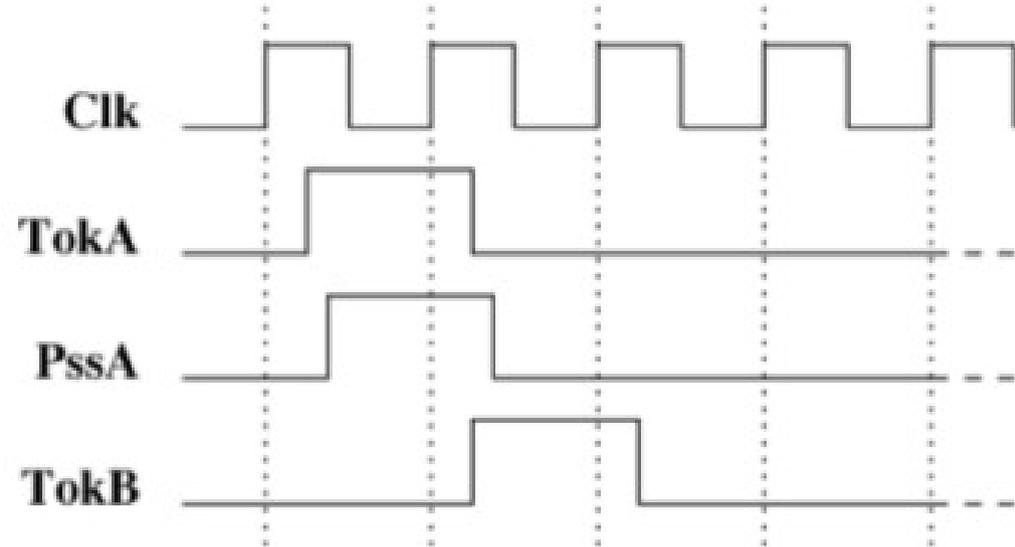


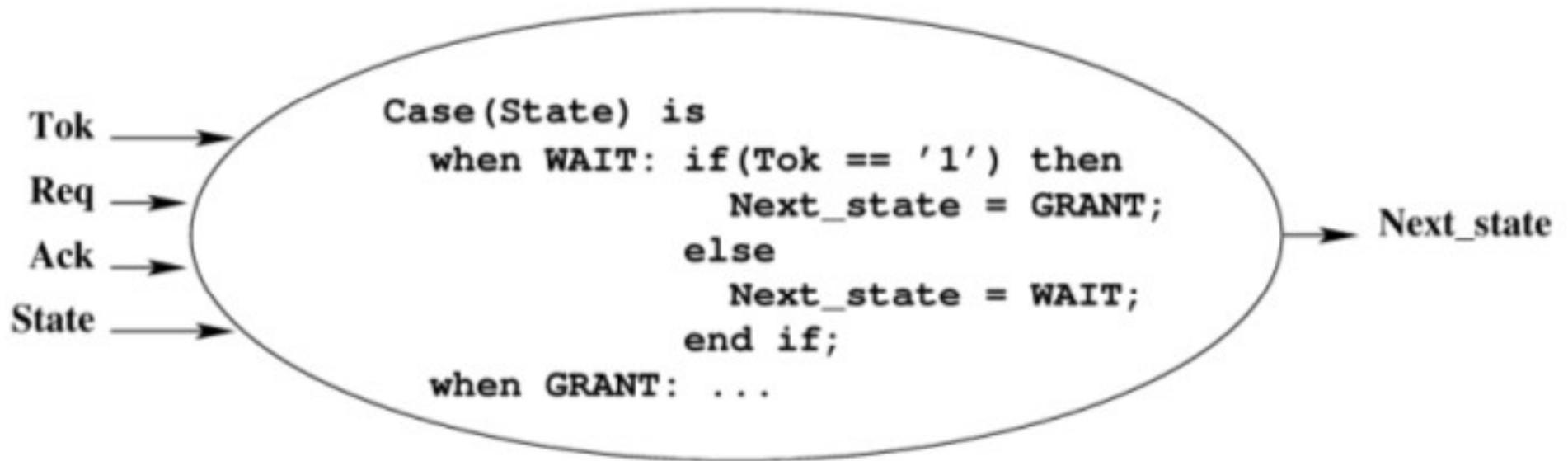
Illustration de la perte d'un cycle



Chronogrammes optimisés

- Proposer des modifications au contrôleur
- Discuter leurs mérites
- Le contrôleur est-il toujours une machine de Moore ?

Description fonctionnelle symbolique



Équité

- Le contrôleur équitable

1. Imaginez des solutions afin de rendre équitable le contrôleur optimisé du premier exercice.
2. Décrivez, sans entrer dans les détails, la structure de ce nouveau contrôleur. Vous séparerez les registres et les parties combinatoires. Vous donnerez une description fonctionnelle symbolique des parties combinatoires et vous explicitez le comportement des registres.

- L'arbitre équitable

1. Pour obtenir que l'ensemble du système soit équitable, la modification du contrôleur seul ne suffit pas. L'arbitre doit, lui aussi, adopter un comportement particulier. Pourquoi ? Donnez un exemple de comportement non équitable possible de l'arbitre et ses conséquences.
2. Imaginez et décrivez des comportements possibles de l'arbitre équitable.
3. Comme précédemment, décrivez la structure de l'arbitre équitable.