Circuits numériques synchrones

et synthèse des automates

version 1.0

Alain Greiner

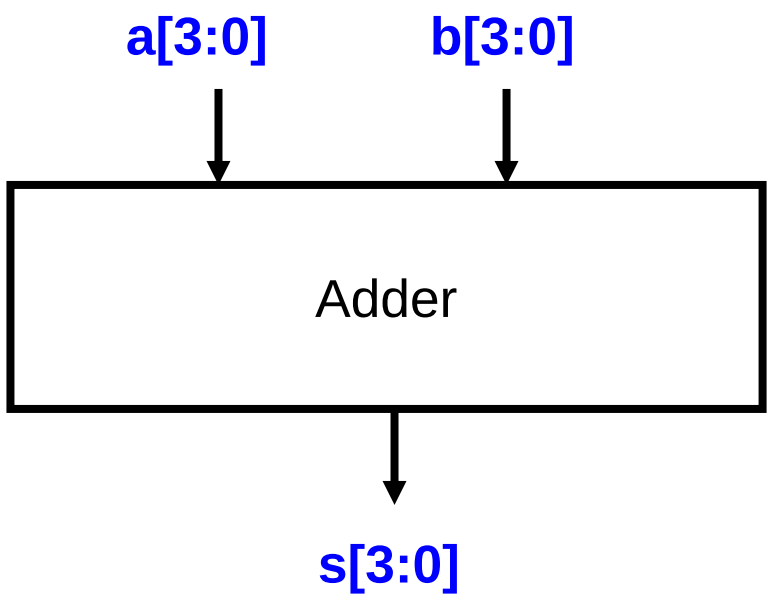
Plan

- **Opérateurs combinatoires**
- **Circuits numériques synchrones**
- **Eléments mémorisants**
- **Modèle des automates synchrones**
- **Synthèse des automates**

Opérateur combinatoire : Additionneur 4 bits

$$A = \Sigma\ a_i * 2^i = a_3*2^3 + a_2* 2^2 + a_1* 2^1 + a_0* 2^0$$

$$B = \Sigma\ b_i * 2^i = b_3*2^3 + b_2* 2^2 + b_1* 2^1 + b_0* 2^0$$



Additionneur 1 bit

$$
\begin{array}{rl}
 & a_i \\
+ & b_i \\
+ & c_i \\
\hline
 & r_i \;\; s_i
\end{array}
$$

La somme de trois bits de même poids numérique peut prendre trois valeurs : 0 , 1 , 2 , 3
Cette somme peut se recoder sur deux bits r_i et s_i
(r_i a un poids numérique double de s_i)

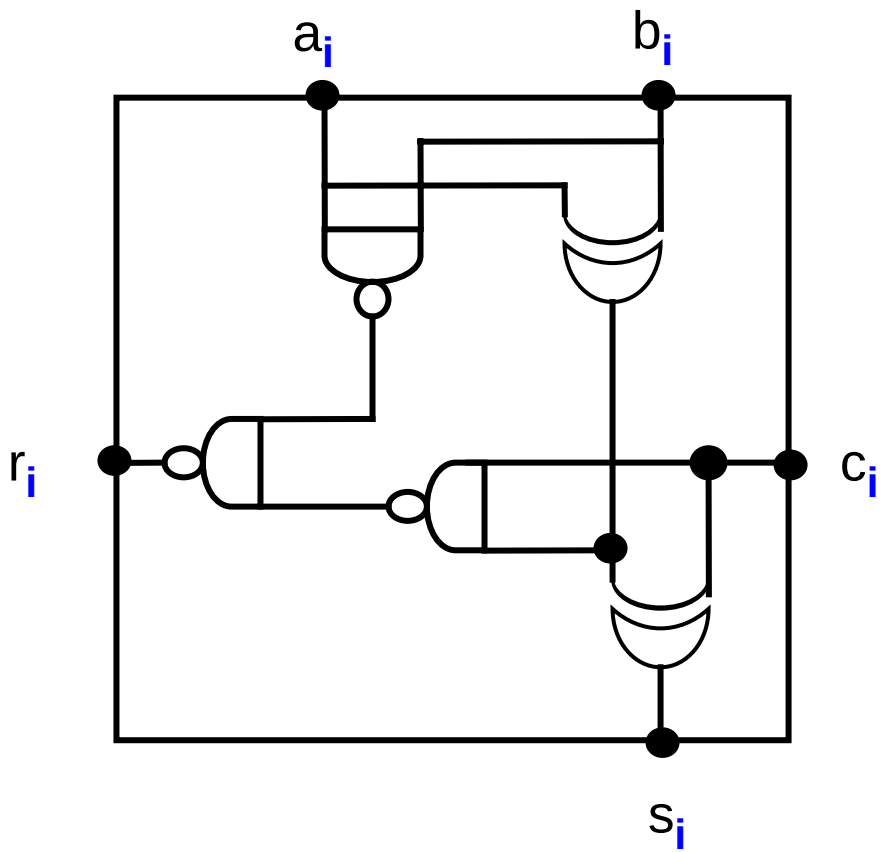
a_i	b_i	c_i	r_i	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Table de vérité

Schéma possible pour l'additionneur 1 bit

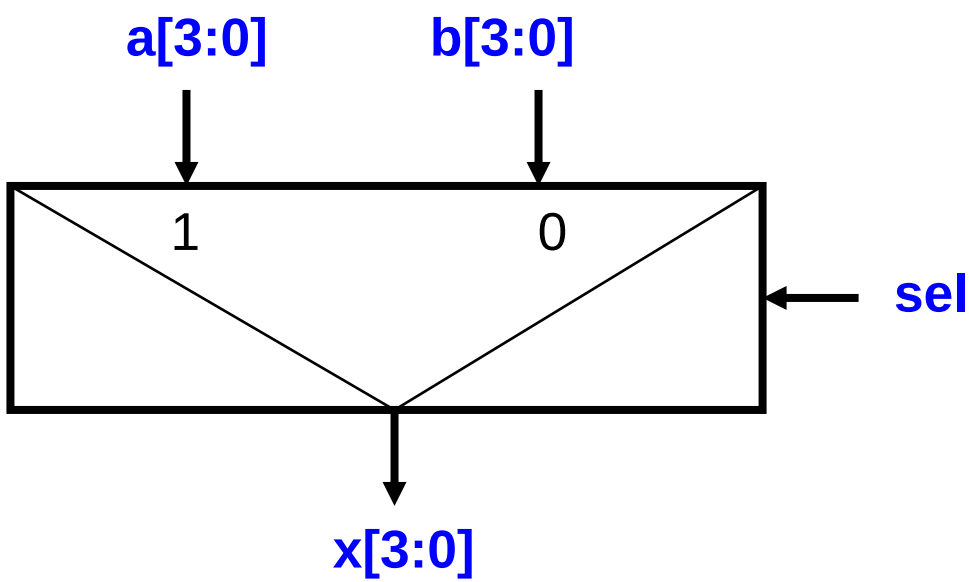
$s_i = a_i$ xor b_i xor c_i

$r_i = (a_i$ and $b_i)$ or $(b_i$ and $c_i)$ or $(a_i$ and $c_i)$



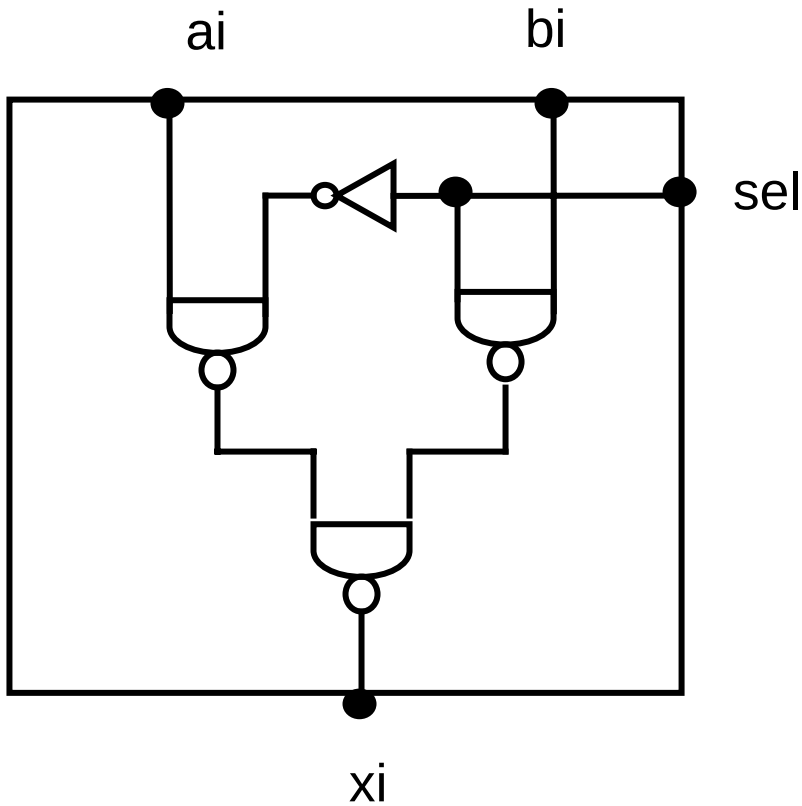
Opérateur combinatoire : Multiplexeur 4 bits

Un multiplexeur permet de sélectionner une donnée parmi 2



Pour tous les bits x_i : x_i = (sel and a_i) or ((not sel) and b_i)

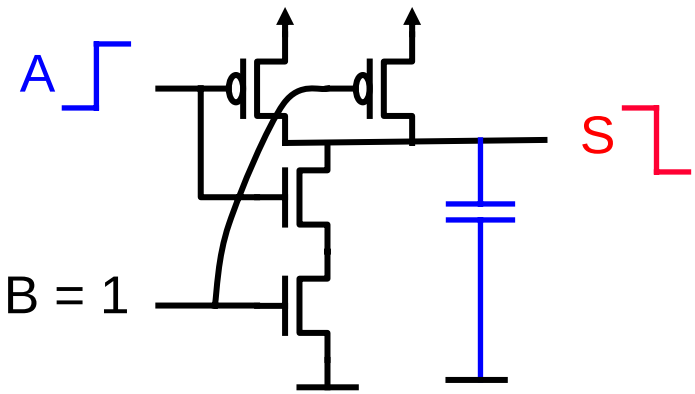
Schéma possible de multiplexeur 1 bit

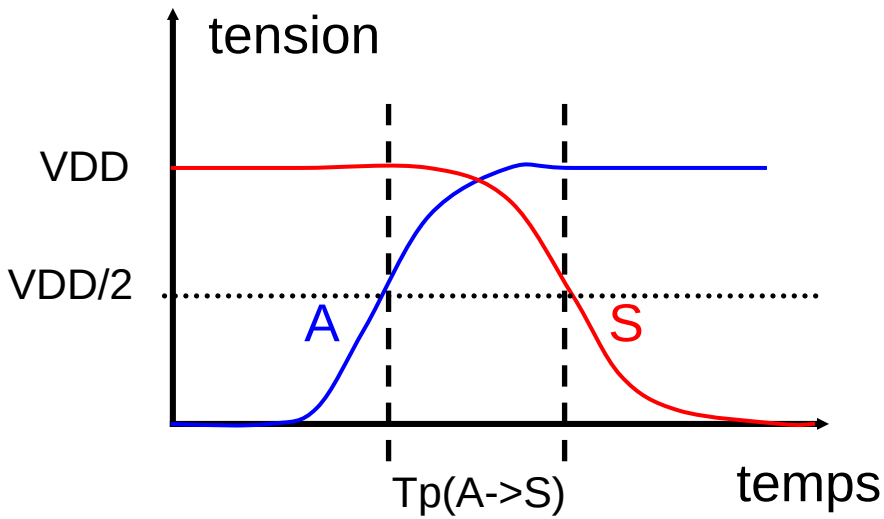


x_i = (sel and a_i) or ((not sel) and b_i)

Temps de propagation

Les portes logiques sont caractérisées par des temps de propagation : un événement sur une entrée peut créer un événement sur la sortie après un temps Tp(A->S)

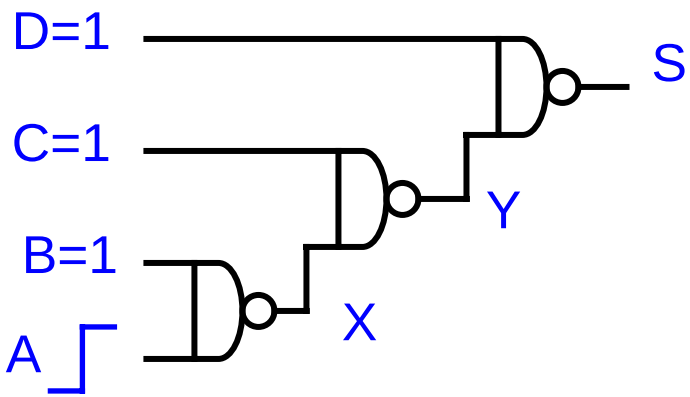


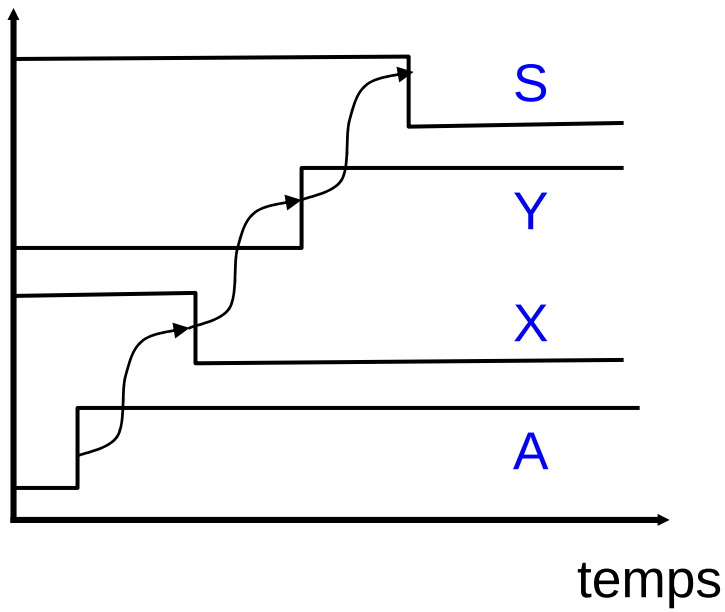


Le temps de propagation dépend de la capacité de charge et de la puissance de la porte.

Propagation des événements

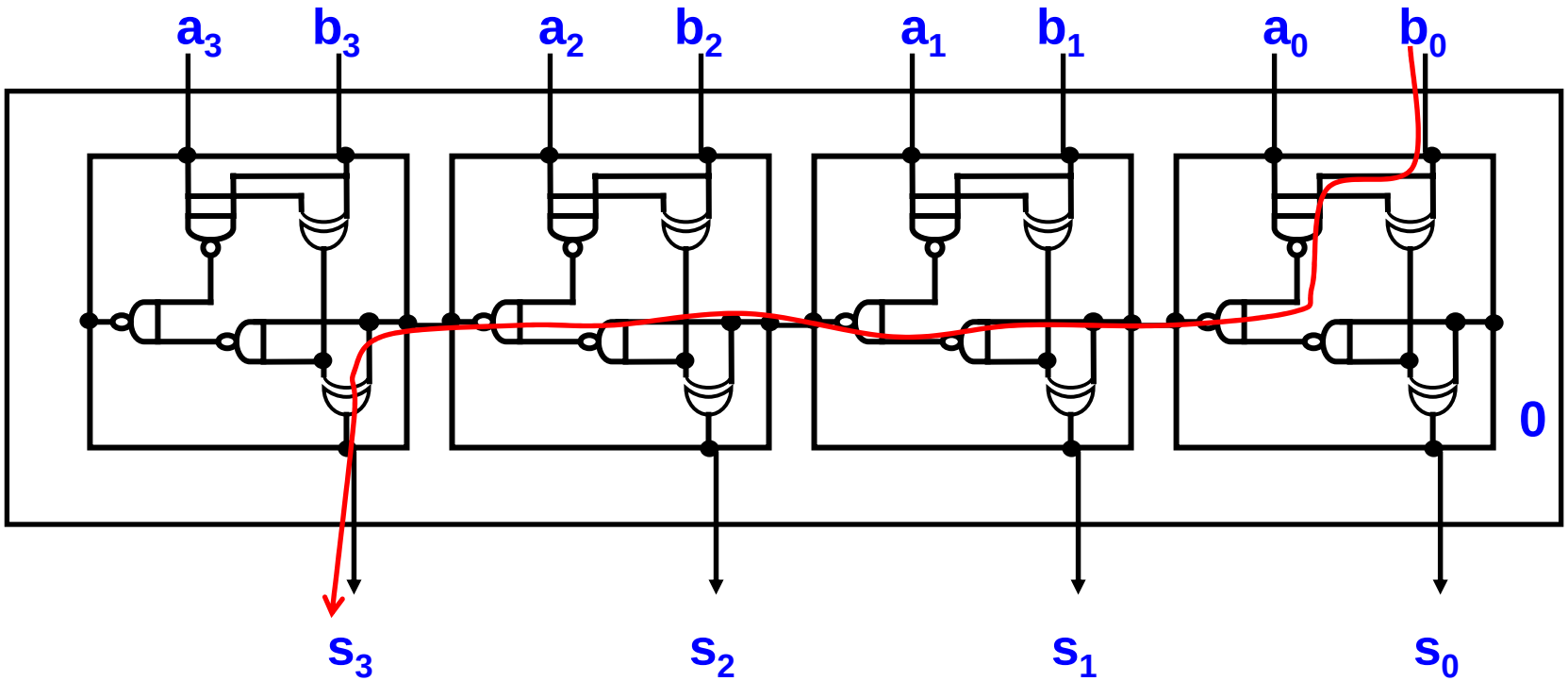
Un opérateur combinatoire est une structure orientée :
Les événements se propagent des entrées vers les sorties !





=> Il ne doit pas y avoir de rebouclage dans un bloc combinatoire

Chaîne longue additionneur 4 bits



La chaîne longue traverse 6 portes « nand » et une porte « xor »

Graphe de causalité

- Les nœuds représentent les événements sur les signaux
- Les arcs représentent les relations de causalité entre événements

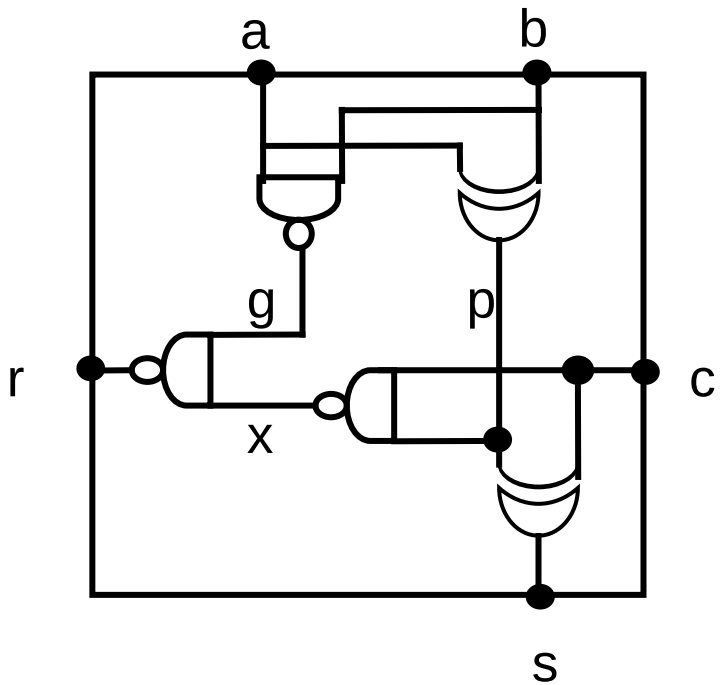
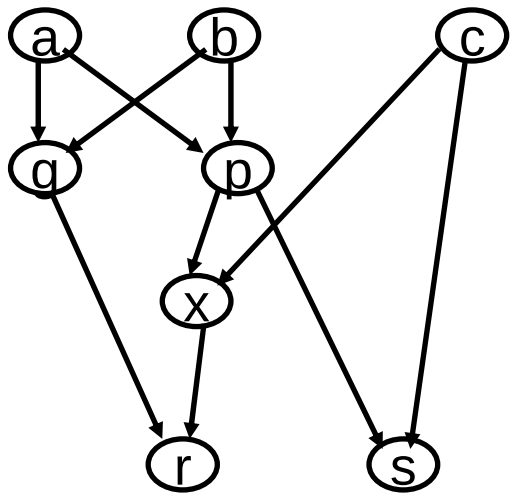


Schéma en portes logiques
(opérateur combinatoire)

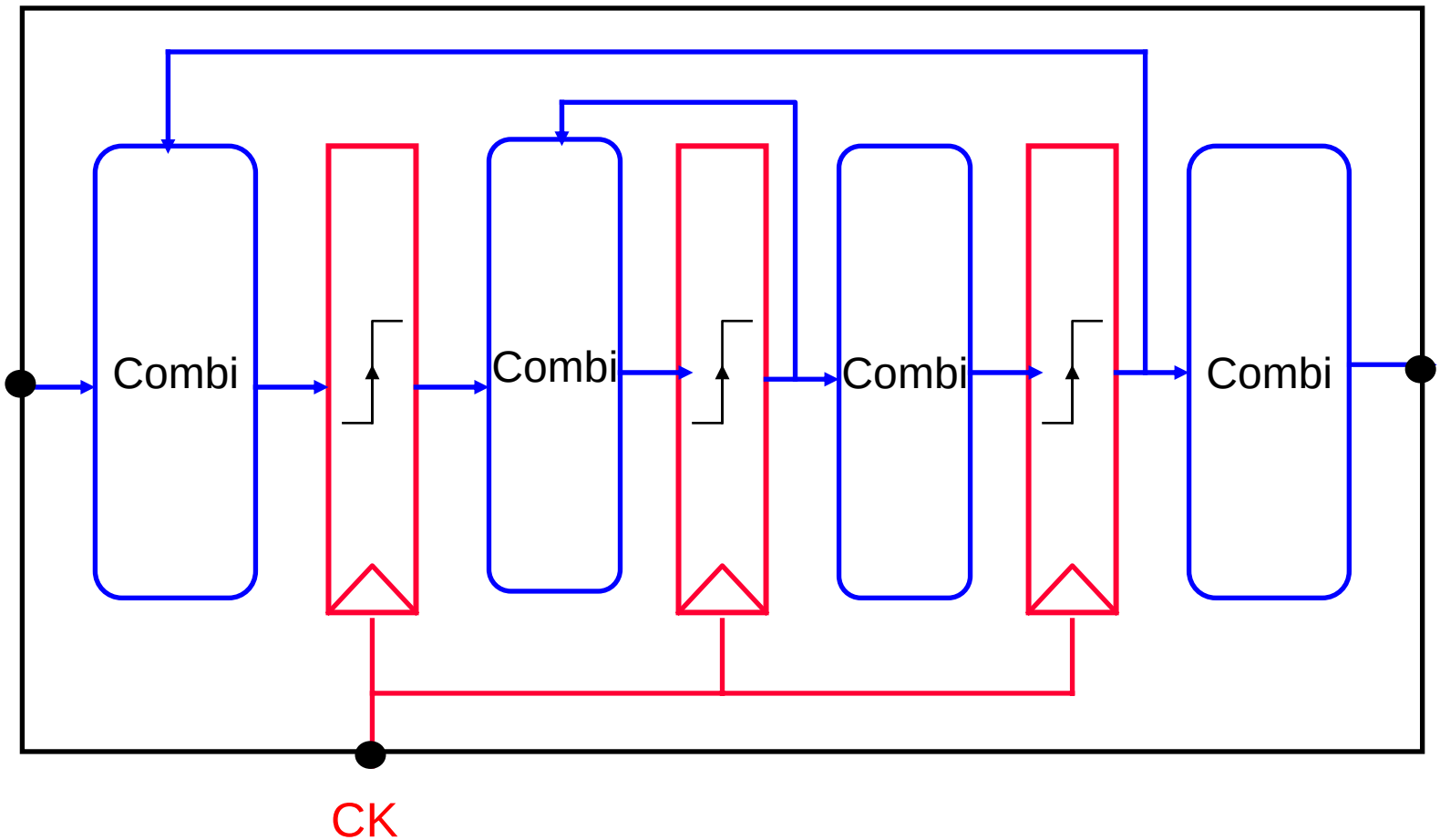


Graphe de causalité
(orienté et acyclique)

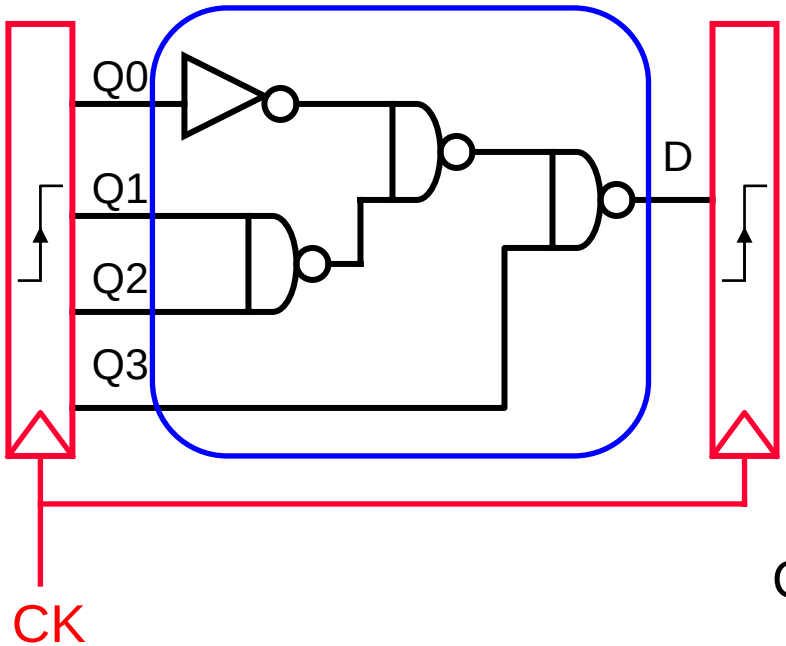
Plan

- **Opérateurs combinatoires**
- **Circuits numériques synchrones**
- **Eléments mémorisants**
- **Modèle des automates synchrones**
- **Synthèse des automates**

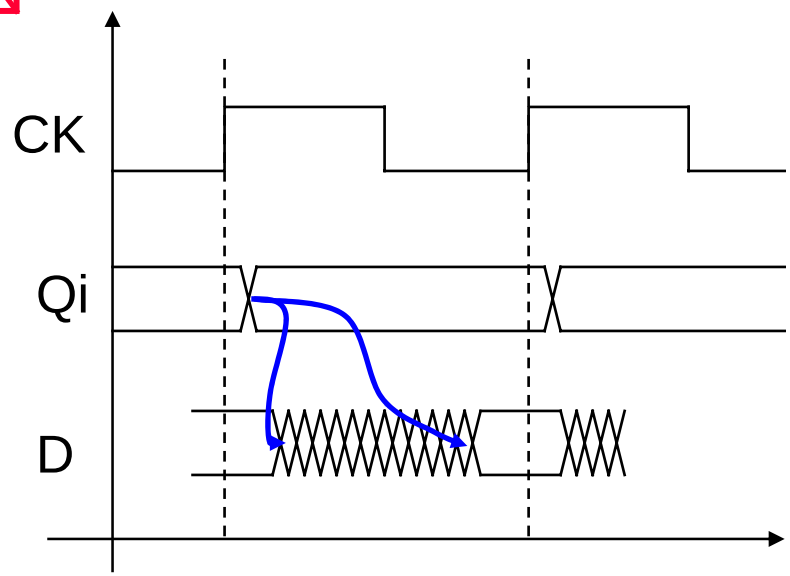
Circuits numériques synchrones



Le rôle stabilisateur des registres

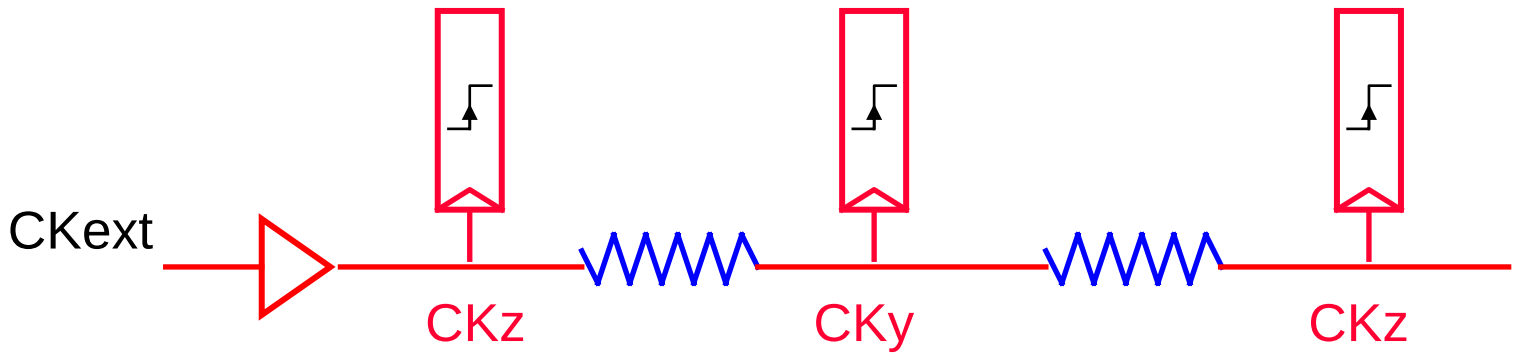


Dans l'opérateur combinatoire, les temps de propagation entre les signaux Q1,Q2,Q3,Q4 et le signal D sont très différents



- Les signaux de sortie des registres (Qi) sont stables pendant tout le cycle
- Les signaux d'entrée des registres ne doivent être stables qu'au moment du front montant d CK.

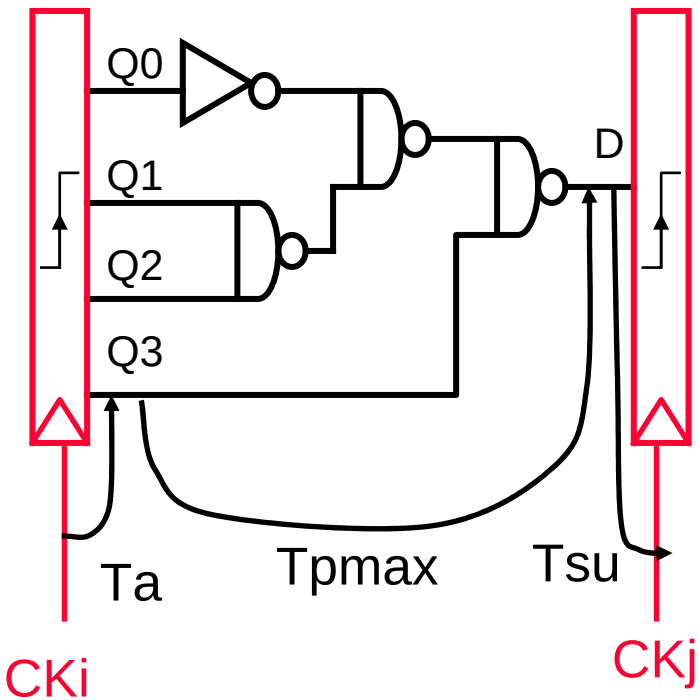
Le skew



Le signal de synchronisation CK est une abstraction.
Le réseau physique de distribution du signal CK n'est pas parfait : les résistances et capacités intrinsèques des fils, ainsi que les amplificateurs intermédiaires introduisent des déphasages entre les signaux Cki qui parviennent aux registres.

Définition : le skew est un majorant de la valeur absolue du déphasage entre deux signaux d'horloge Cki et CKj

Chaînes longues



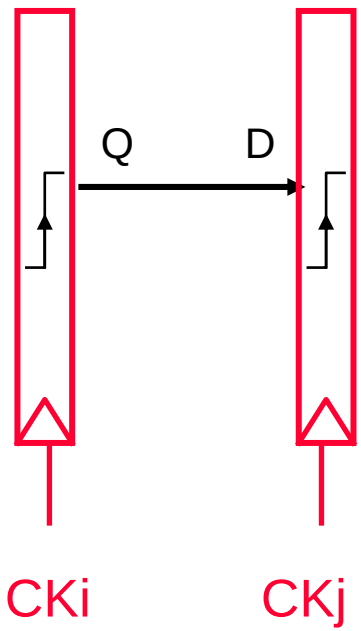
Pour qu'un circuit numérique synchrone fonctionne correctement, il faut que le temps de propagation le la chaîne combinatoire la plus longue Tpmax entre deux registres soit inférieur au temps de cycle TC.

Il faut tenir compte du temps d'accès Ta et du temps de pré-établissement Tsu des registres, ainsi que du « skew »

Tout opérateur combinatoire doit respecter la condition suivante :

$$\boxed{Ta(CK\text{->}Qi) + Tpmax(Qi\text{->}D) + Tsu(D\text{->}CKj) < TC - skew}$$

Chaînes courtes



Un dysfonctionnement de type « chaîne courte » se produit lorsque le temps de propagation minimal entre deux registres est plus court que le déphasage entre les deux signaux d'horloge (skew).

Il faut ici également tenir compte du temps d'accès Ta du registre source et du temps de maintien Th du registre destination.

Tout opérateur combinatoire doit respecter la condition suivante :

$$\boxed{Ta(CK\text{->}Q) + Tpmin(Q\text{->}D) > Th(D\text{->}CKj + skew}$$

Le triple rôle des registres

Les registres ont une triple fonction :

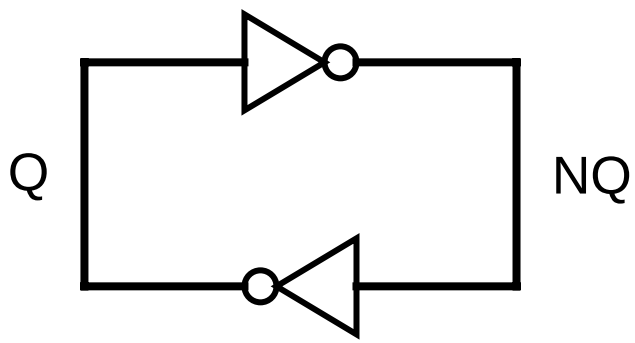
- Mémorisation : stocker une valeur qui sera utilisée plus tard. Ceci impose que l'écriture dans les registres soit conditionnelle.
- Synchronisation : assurer que toutes les données d'un même opérateur combinatoire sont simultanément disponibles
- Stabilisation : garantir que les signaux d'entrée des opérateurs combinatoires sont stables pendant toute la durée du cycle.

Plan

- **Principe des circuits numériques synchrones**
- **Opérateurs combinatoires**
- **Eléments mémorisants**
- **Modèle des automates synchrones**
- **Synthèse des automates**

Bi-stable

Tout dispositif de mémorisation statique (durée de mémorisaton illimitée) contient une structure rebouclée possédant deux états stables.



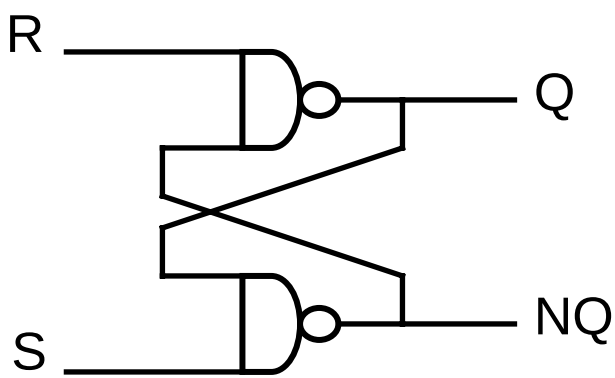
NQ = not Q

Q = not NQ

Les deux états sont :

- Q = 0 / NQ = 1
- Q = 1 / NQ = 0

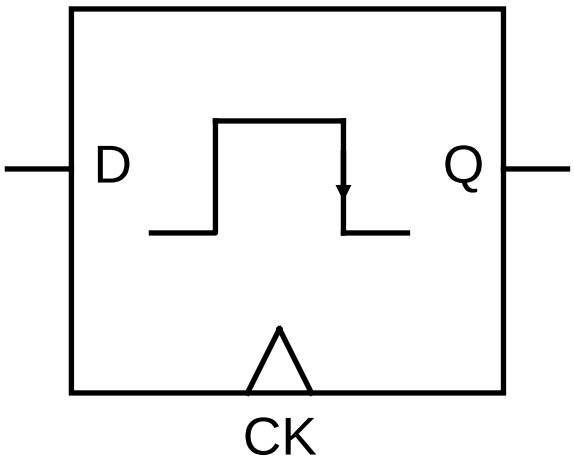
Bascule R/S



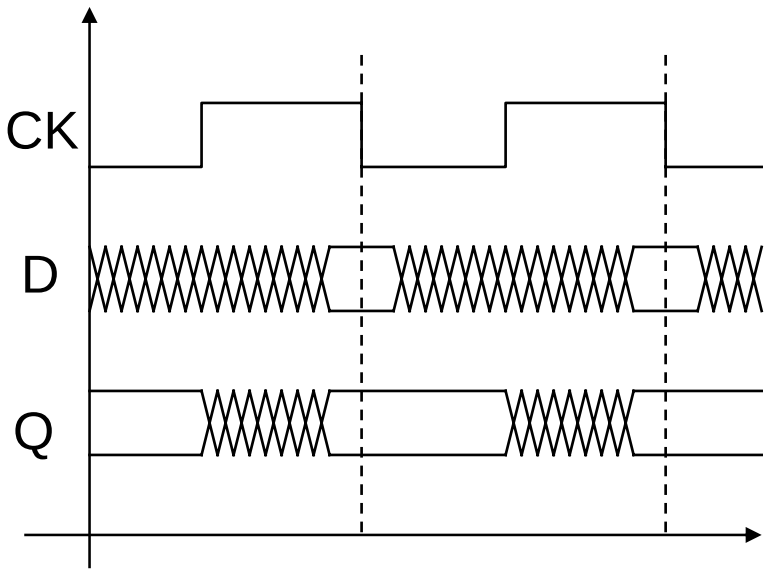
R	S	Q	NQ	
1	1	Q_{-1}	NQ_{-1}	Mémorisation
0	1	1	0	Écriture d'un 1
1	0	0	1	Écriture d'un 0
0	0	1	1	Inutilisé

Une « bascule » R/S est un bistable inscriptible

Définition du Latch

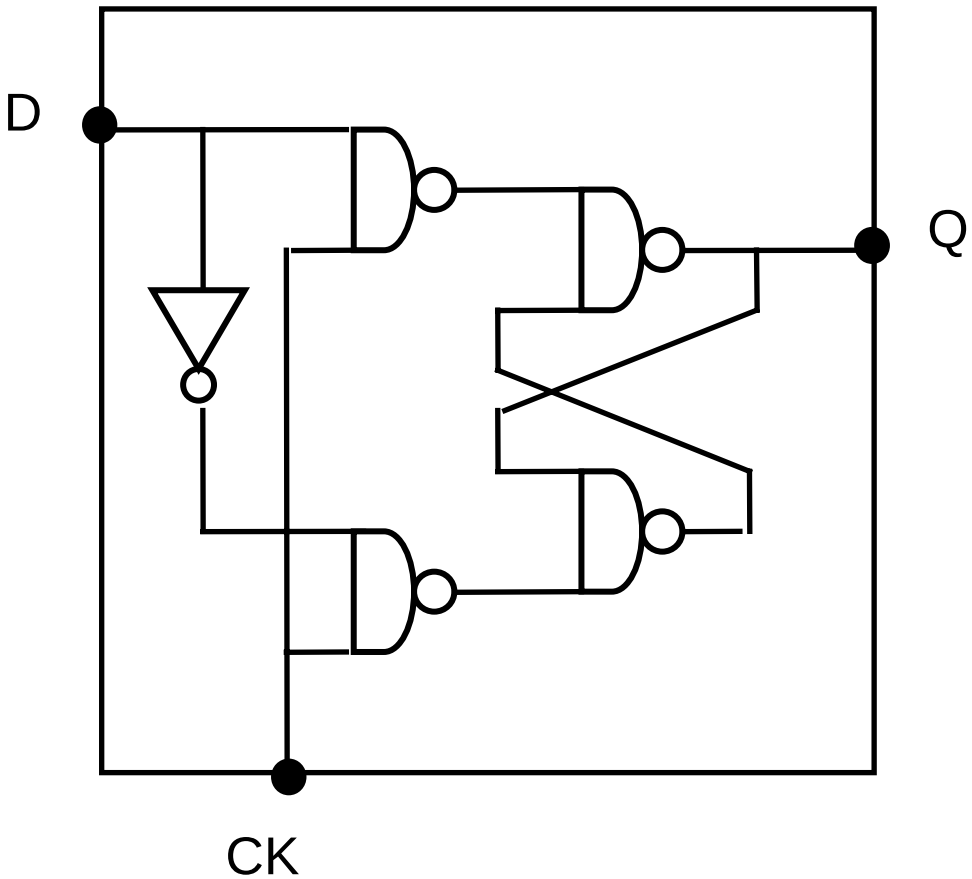


CK	Q	
0	Q_{-1}	Mémorisation
1	D	Transparence



Un latch est un
« demi-registre »

Schéma possible d'un latch

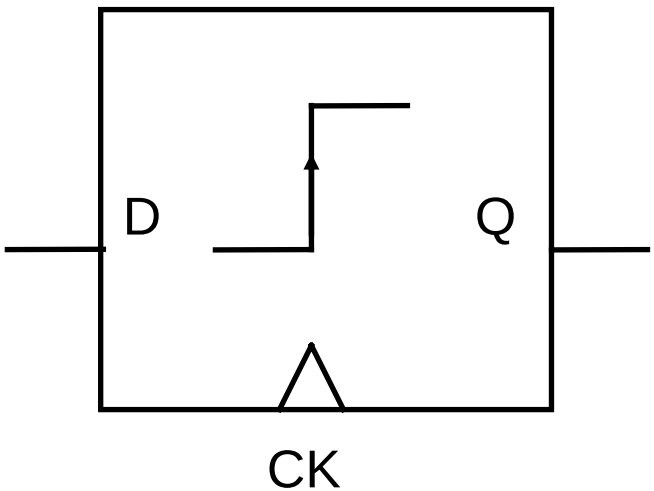


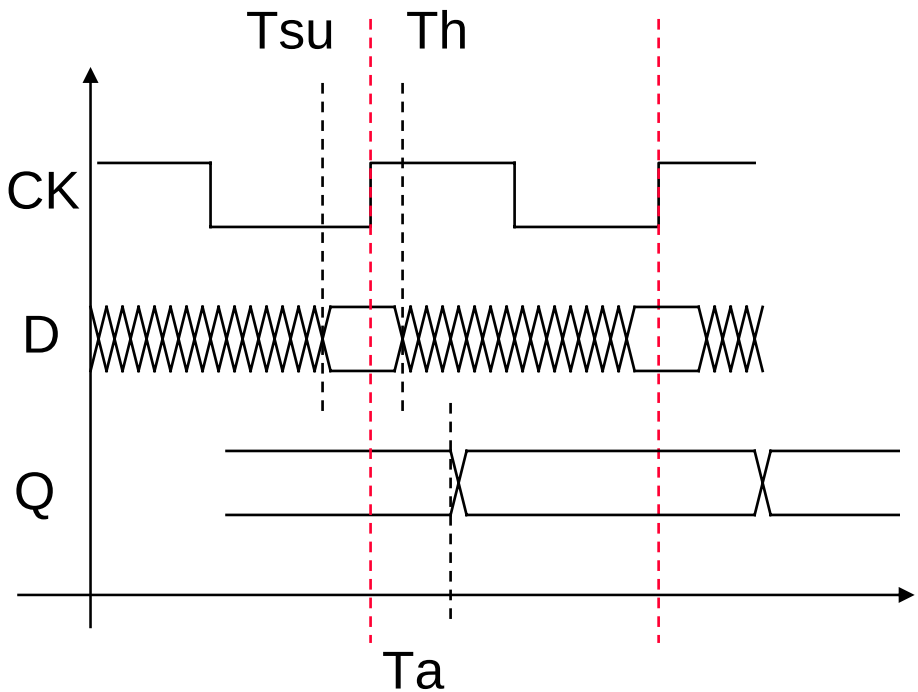
CK	Q	
0	Q_{-1}	Mémorisation
1	D	Transparence

Définition de la bascule D (D flip-flop)

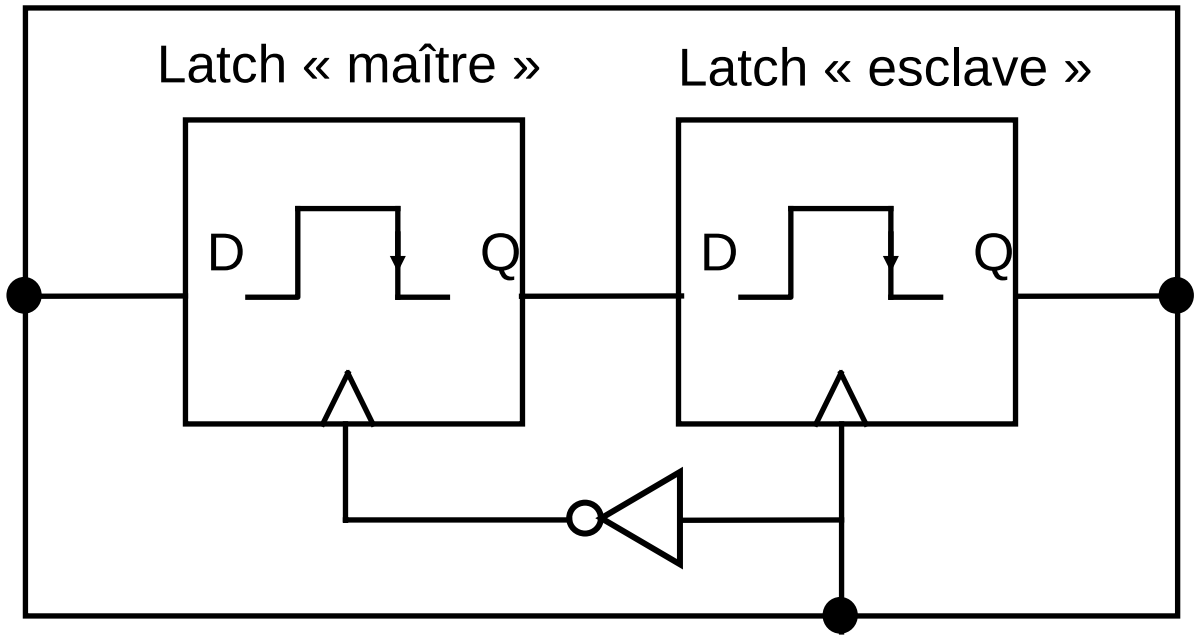
Une bascule D permet de « mémoriser » 1 bit

- L'écriture d'une nouvelle valeur a lieu lors du front montant du signal CK
- L'entrée D doit être stable un peu avant (Tsu) et un peu après (Th) le front
- La sortie Q change de valeur au plus une fois par cycle après le front (Ta)



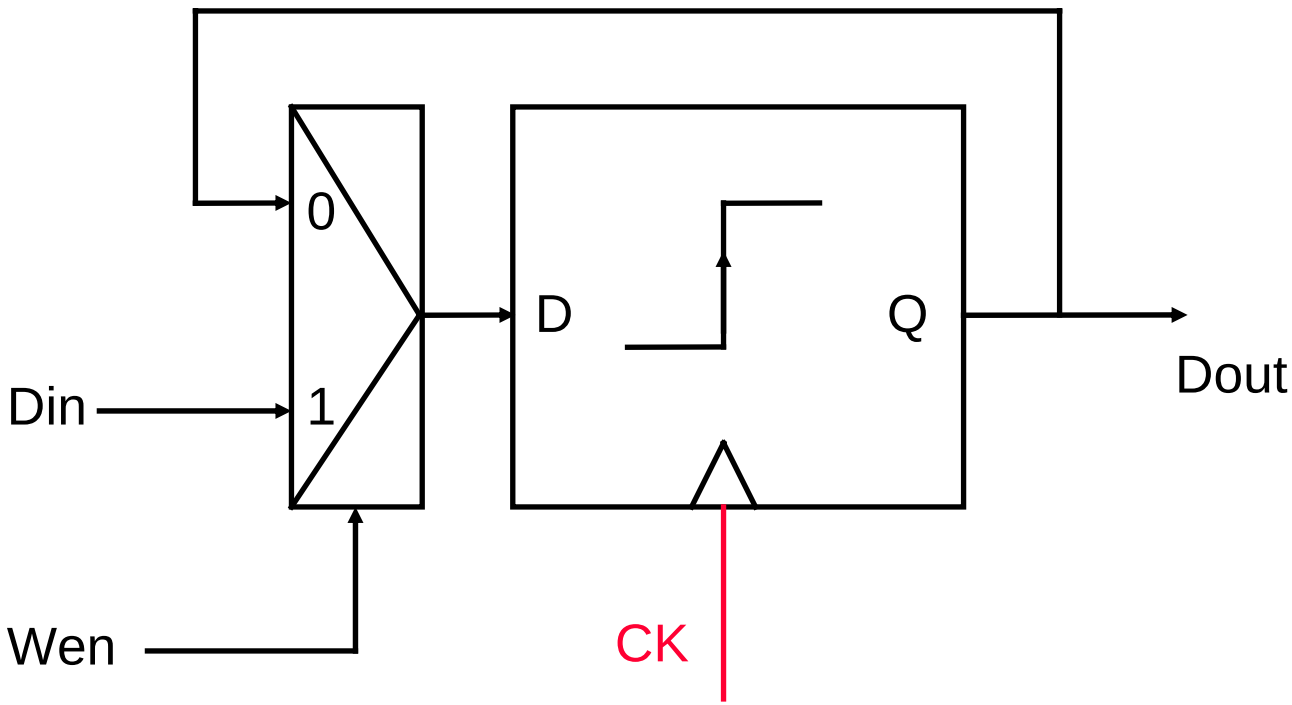


Réalisation d'une Bascule D



On réalise une bascule D à échantillonnage sur front en utilisant deux latchs en série.

Ecriture conditionnelle

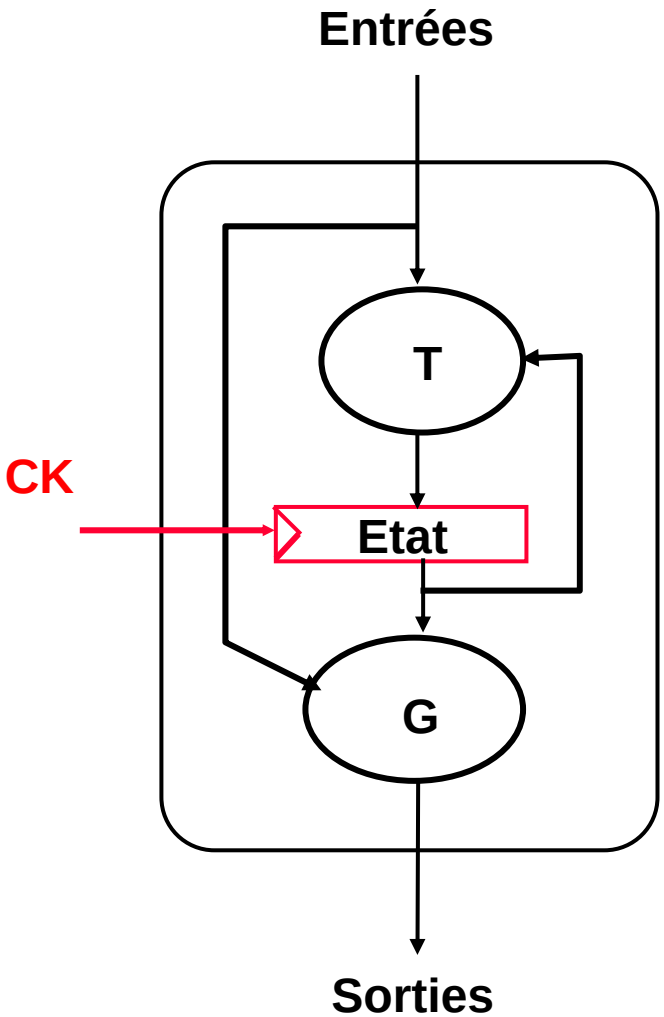


Lorsque le signal d'autorisation d'écriture vaut 0,
on re-écrit la valeur déjà pstockée dans le registre.

Plan

- **Principe des circuits numériques synchrones**
- **Opérateurs combinatoires**
- **Réalisation des éléments mémorisants**
- **Modèle des automates synchrones**
- **Synthèse des automates**

Automates d'états synchrones



Tout système numérique synchrone peut être modélisé par un automate :

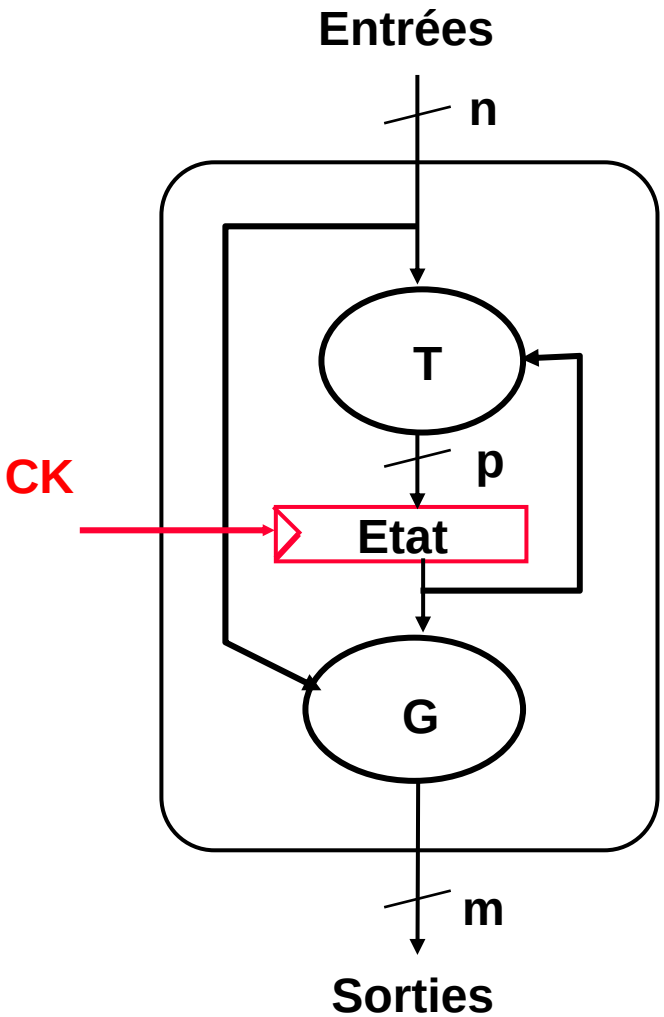
Fonction de transition :

NextEtat <= T(Etat, Entrées)

Fonction de génération :

Sorties <= G(Etat, Entrées)

Automates d'états synchrones



Si on a :
- n bits d'entrée Ei
- m bits de sortie Sj
- p bits mémorisés Rk

Il faut définir :

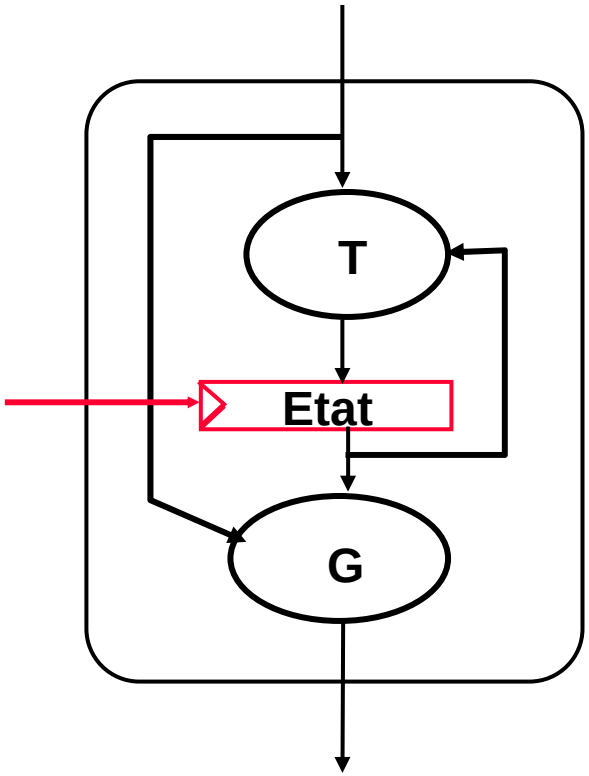
- p fonctions booléennes dépendant de (n+p) variables pour la transition

$$NR_k = T_k(E_i, R_k)$$

- m fonctions booléennes dépendant de (n+p) variables la génération

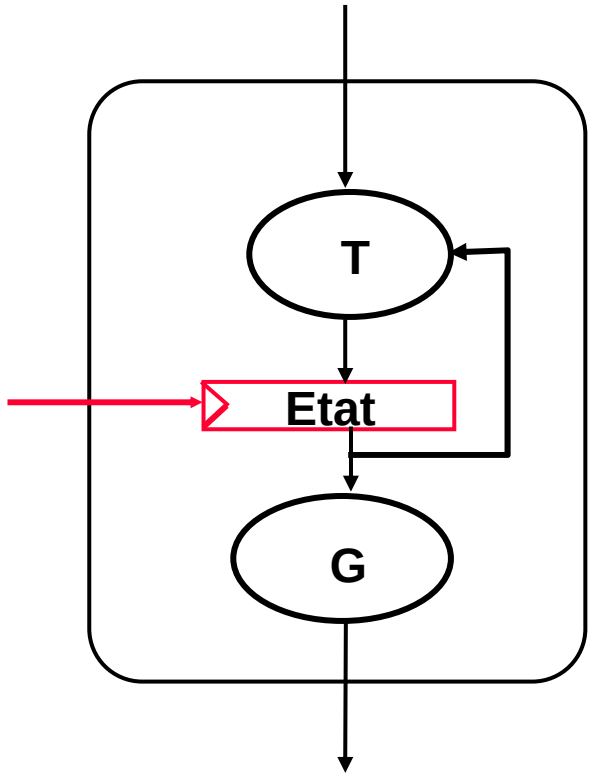
$$S_j = \Gamma j(E_i, R_k)$$

Automates de Moore et de Mealy



Cas général :

Automate de Mealy



Cas particulier : La fonction de génération ne dépend que de l'état !

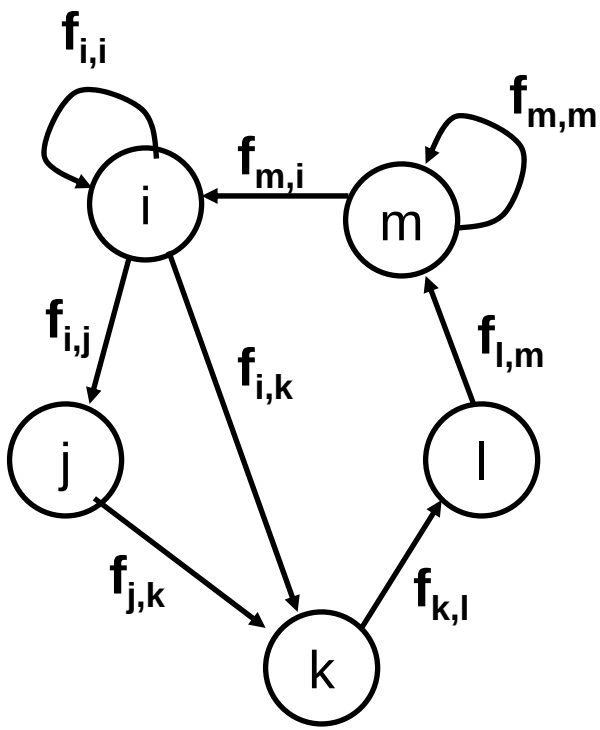
Automate de Moore

Représentation graphique d'un automate

- Les nœuds {i} représentent les états
- Les arcs (i,j) représentent les transitions

Chaque arc (i,j) est étiqueté par une expression Booléenne $\mathbf{f_{i,j}}$ ne dépendant que des signaux d'entrée, et définissant la condition de transition.

Dans le cas d'un automate de Moore, les signaux de sortie ne dépendent que de l'état : chaque nœud est donc étiqueté par la valeur des signaux de sortie.



Automate déterministe

Pour qu'un automate possède un comportement déterministe, il faut respecter les conditions suivantes :

• Existence d'un mécanisme matériel d'initialisation permettant de forcer l'automate dans un état initial connu.

• Condition d'orthogonalité : pour tout état i, et pour toute configuration des entrées, il y a un seul état successeur de l'état i.

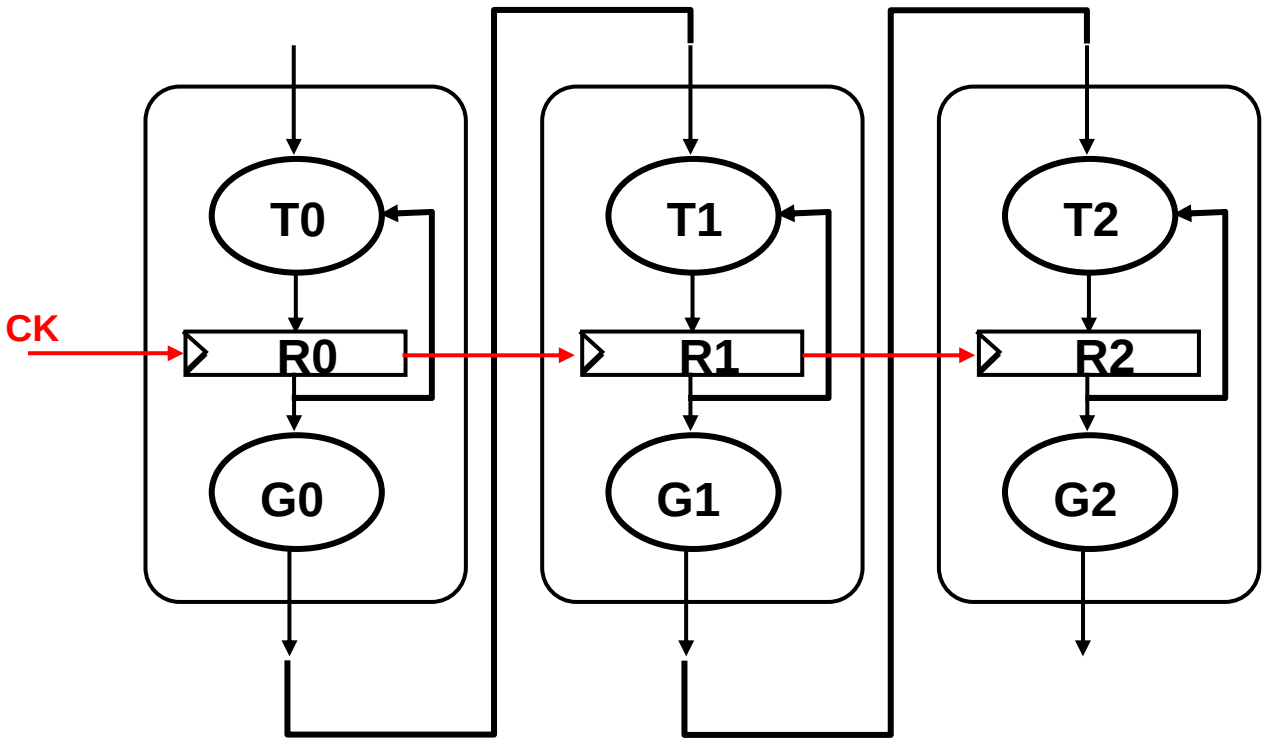
$$\mathbf{f}_{i,j} \cdot \mathbf{f}_{i,k} = 0 \quad \text{si j différent de k}$$

• Condition de complétude : pour toute configuration des entrées, il y a toujours un état successeur de l'état i.

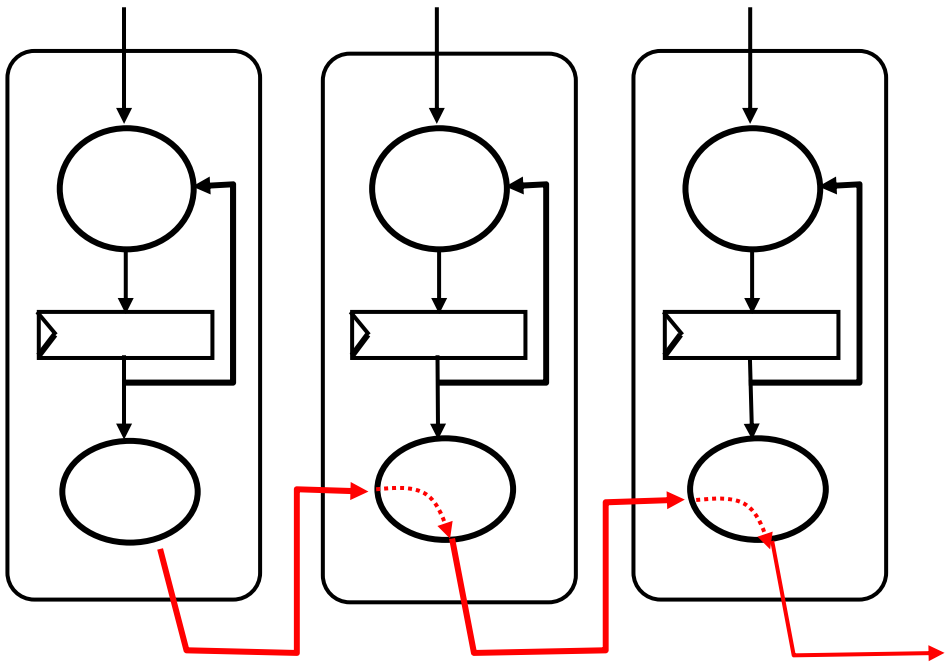
$$\Sigma_j \ \mathbf{f}_{i,j} = 1$$

Automates communicants

Un système numérique synchrone est souvent conçu comme un ensemble d'automates fonctionnant en parallèle, et communicant entre eux.



Inconvénient des Automates de mealy



Dans un automate de mealy,

Il existe une dépendance combinatoire entre les entrées et les sorties !

⇒ ceci introduit des chaînes longues traversant plusieurs composants.

Il devient impossible de caractériser le comportement temporel de chaque automate indépendamment des autres.

Plan

- **Principe des circuits numériques synchrones**
- **Opérateurs combinatoires**
- **Réalisation des éléments mémorisants**
- **Modèle des automates synchrones**
- **Synthèse des automates**

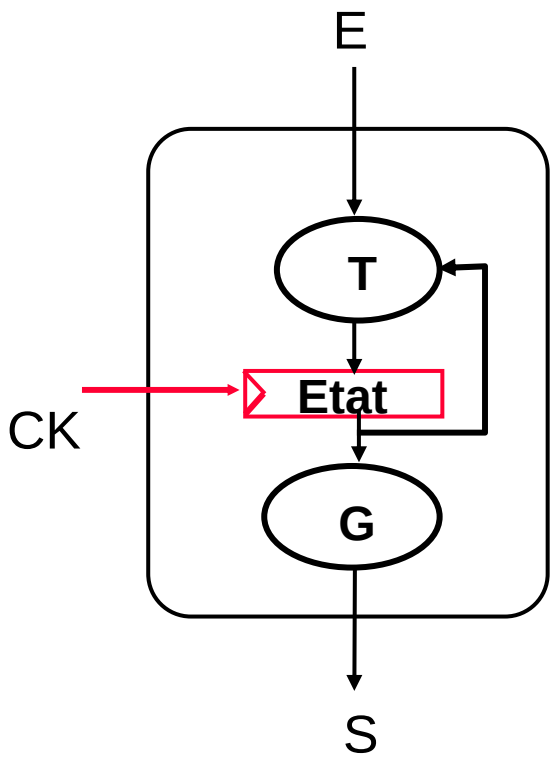
Exemple : Détection d'une séquence

Problème à résoudre :

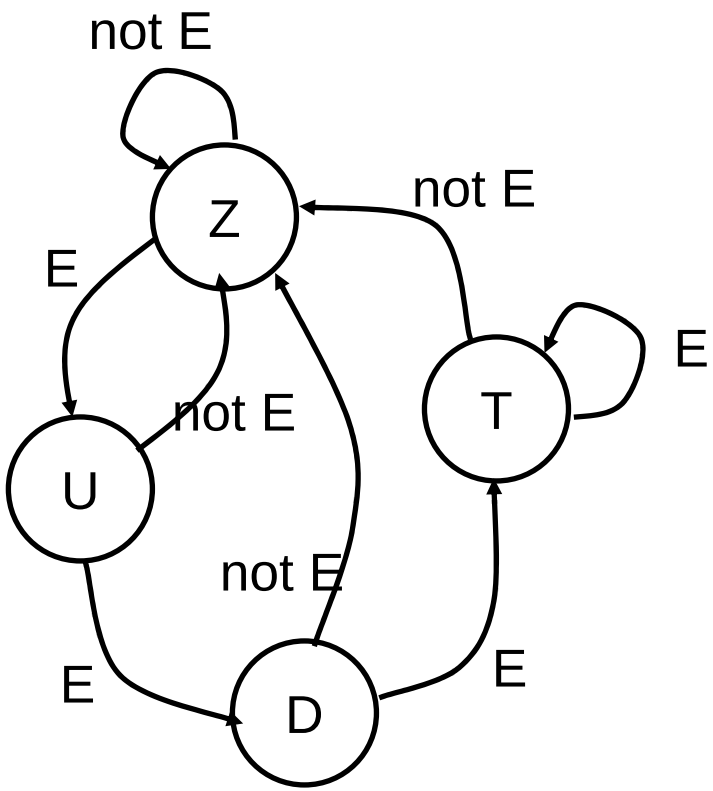
On cherche a réaliser un dispositif matériel possédant un seul bit d' entrée E et une seul bit de sortie S.

L'entrée E est échantillonnée à chaque cycle.

La sortie S passe à 1 pendant un cycle chaque fois que l'entrée E a la valeur 1 pendant 3 cycles consécutifs.



1) construction du graphe de l'automate



Signification des états :

- Z : le dernier bit reçu valait 0
- U : les 2 derniers bits reçus valaient 0,1
- D : les 3 derniers bits reçus valaient 0,1,1
- T : les 3 derniers bits reçus valaient 1,1,1

2) Choix du codage des états

Etat	R1	R0	X0	X1	X2	X3
« Z »	0	0	1	0	0	0
« U »	0	1	0	1	0	0
« D »	1	0	0	0	1	0
« T »	1	1	0	0	0	1

codage binaire (sur 2 bits)

codage « one hot » (un bit par état)

Dans la suite de l'exemple, on utilisera le codage binaire

3) Construction de la table de vérité

R1	R0	E	NR1	NR0	S
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Fonction de transition :

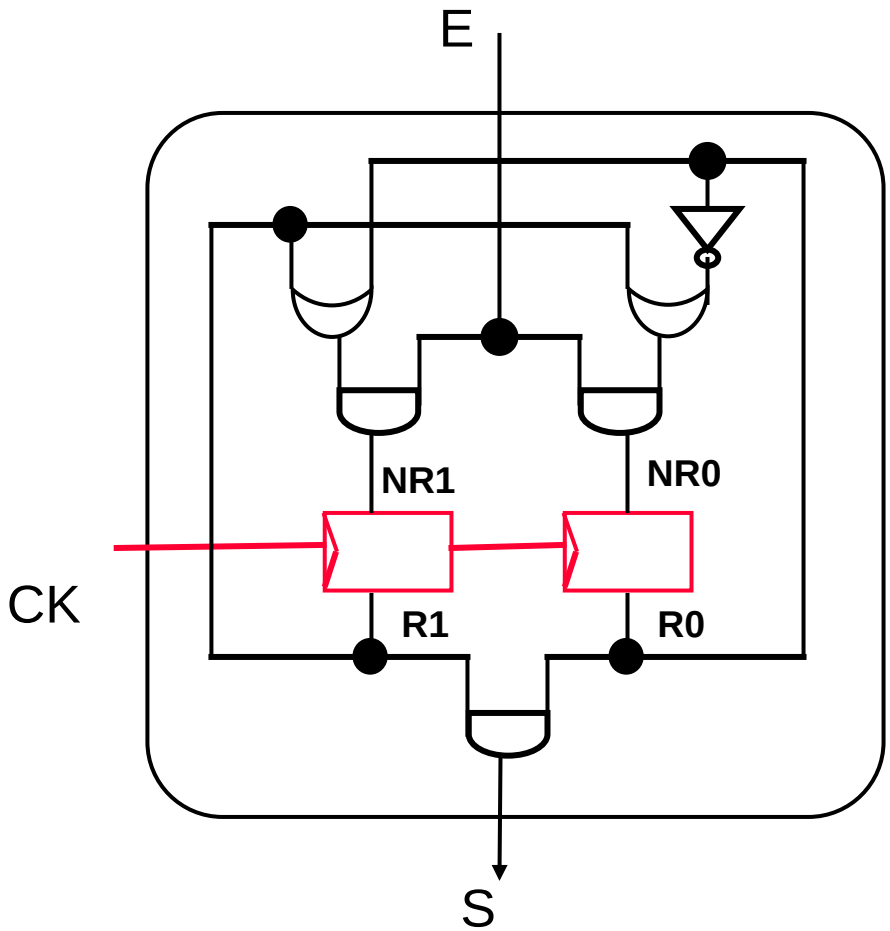
NR1 = E and (R1 or R0)

NR0 = E and (R1 or (not R0))

Fonction de génération :

S = R1 and R0

4/ Synthèse logique



On traduit les expressions Booléennes représentant la fonction de transition et la fonction de génération en un schéma en portes.

Pour cet exemple, il faut 3 portes « and », 2 portes « or », un inverseur et 2 bascules « D ».

Généralisation

– Cette méthode de synthèse est systématique et a été automatisée dans des outils de CAO/VLSI.

– On peut générer le schéma en portes pour des automates possédant plusieurs centaines d'états.

– Le facteur limitant est la taille de la table de vérité dont la complexité est proportionnelle à 2^{n+p} (n nombre de bits d'entrée, p nombre de bascules)