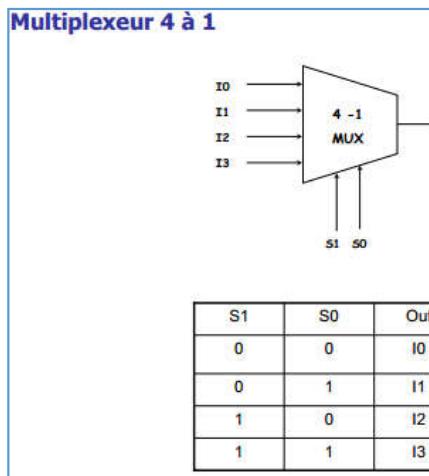


Logique combinatoire et séquentielle

Circuits combinatoires

Les fonctions de sortie s'expriment selon des expressions logiques des seules variables d'entrée.



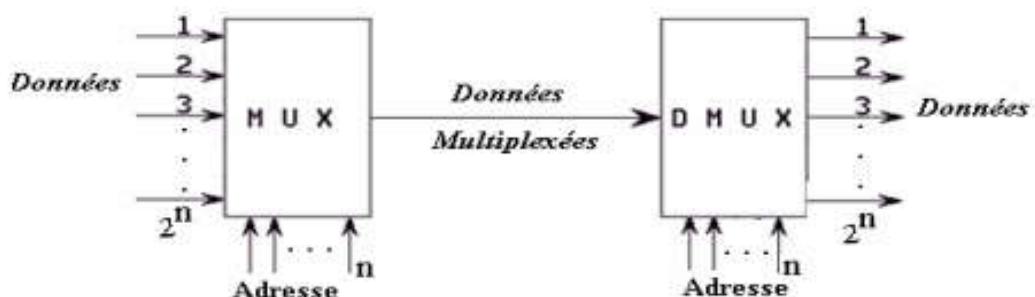
Multiplexeur - Demultiplexeur

Multiplexeur

- 2^n entrées, 1 sortie
- Selon une adresse (n bits), la sortie prend la valeur de l'une des entrées

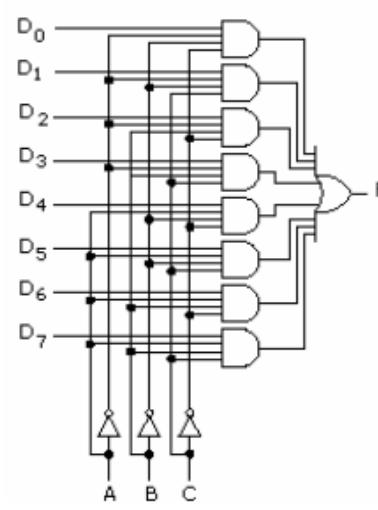
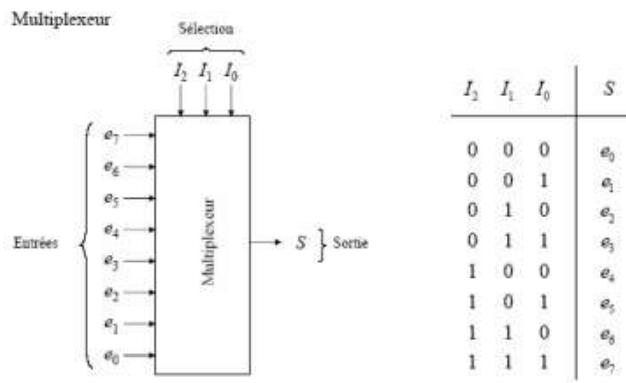
Démultiplexeur

- 1 entrée, X sorties
- Selon une adresse (n bits), une des X sorties prend la valeur de l'entrée

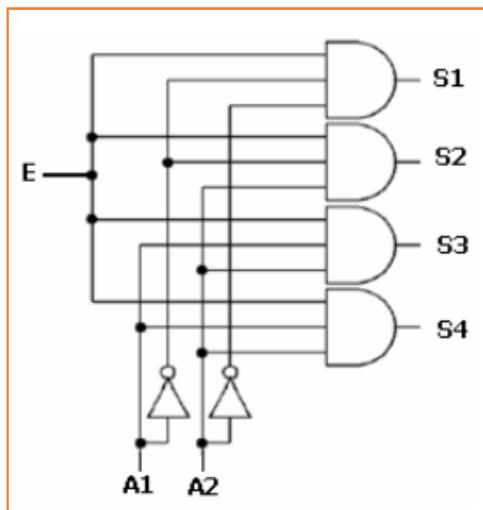
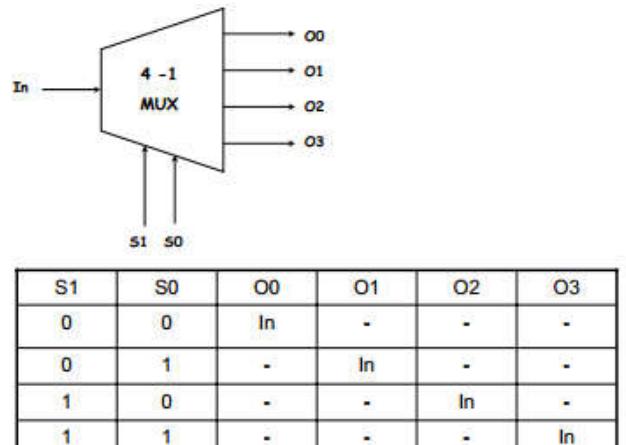


Application: Conversion Série/Parallèle; Parallèle/Série

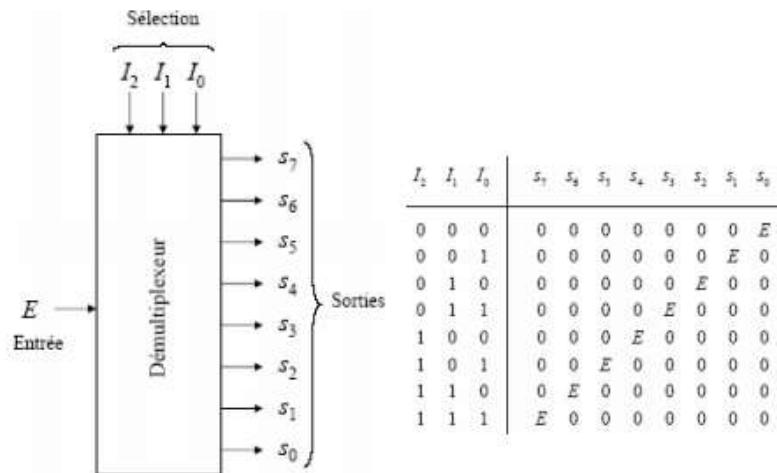
Multiplexeur 8 à 1



Demultiplexeur 1-4



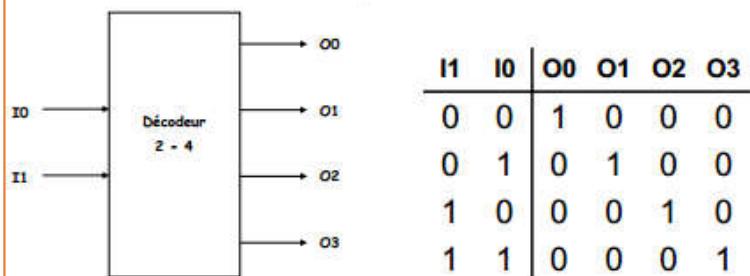
Demultiplexeur 1-8



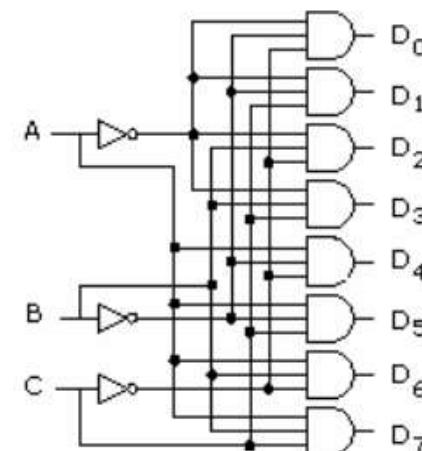
Décodeur

- Active une des X sorties selon un code
- Entrée sur n bits
- Nombre de sorties : 2^n
- Une seule sortie est mise à 1 selon la configuration des entrées
- Application:** Sélection des circuits mémoire

Exemple: Décodeur 2 à 4

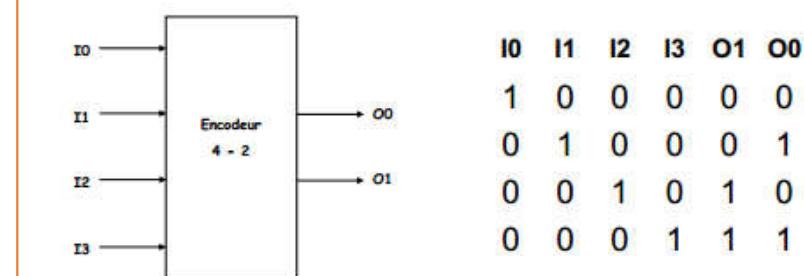


Exemple: Décodeur 3-8



Encodeur

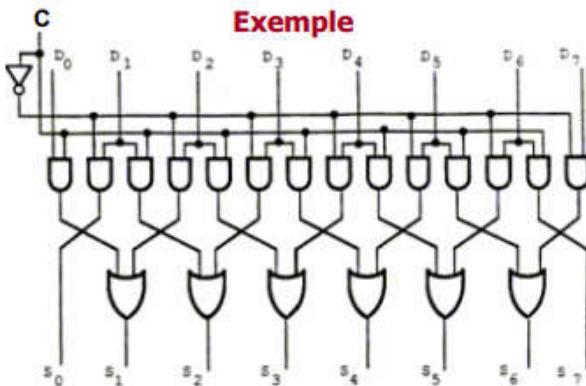
- Active un code selon l'une des X entrées actives
- 2^n entrées, 1 entrée active (valeur 1), les autres sont toutes désactivées (valeur 0)
- Sortie : sur n bits



Exemple: Encodeur 4-2

Circuit de décalage

Décalage de position d'un bit (à droite ou à gauche) sur les n bits



Les lignes de sorties (S_0 à S_7) reflète les 8 bits d'entrée (D_0 à D_7) après décalage d'un bit à droite pour $C=1$ ou à gauche pour $C=0$.

Additionneur

Un additionneur est un circuit capable de faire l'addition de deux nombres de n bits. Une addition génère deux résultats.

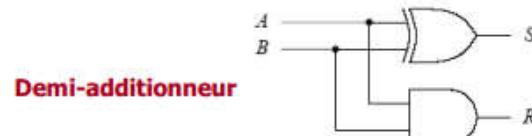
- La somme
- La retenue

Exemple: addition de 2 bits

$$\text{Somme (S)} = A \oplus B$$

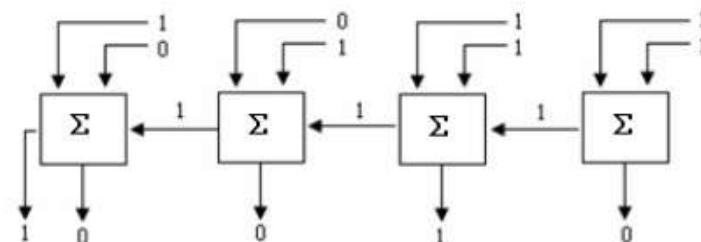
$$\text{Retenue (R)} = AB$$

Entrée		Sortie	
A	B	R	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



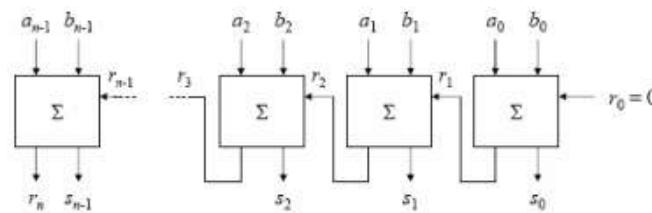
Exemple: Additionneur 4 bits

$$\begin{array}{r}
 A \quad \quad \quad 1 \quad 0 \quad 1 \quad 1 \\
 B \quad \quad \quad 0 \quad 1 \quad 1 \quad 1 \\
 \hline
 1 \quad 0 \quad 0 \quad 1 \quad 0
 \end{array}$$



Additionneur complet

$$\begin{array}{r}
 r_{n-1} \quad \quad \quad \quad r_3 \quad \quad r_2 \quad \quad r_1 \quad \quad r_0 \\
 a_{n-1} \quad \dots \quad a_3 \quad a_2 \quad a_1 \quad a_0 \\
 + \quad b_{n-1} \quad \dots \quad b_3 \quad b_2 \quad b_1 \quad b_0 \\
 \hline
 r_n \quad \dots \quad s_{n-1} \quad s_3 \quad s_2 \quad s_1 \quad s_0
 \end{array}$$



Additionneur complet

r_i	a_i	b_i	s_i	r_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

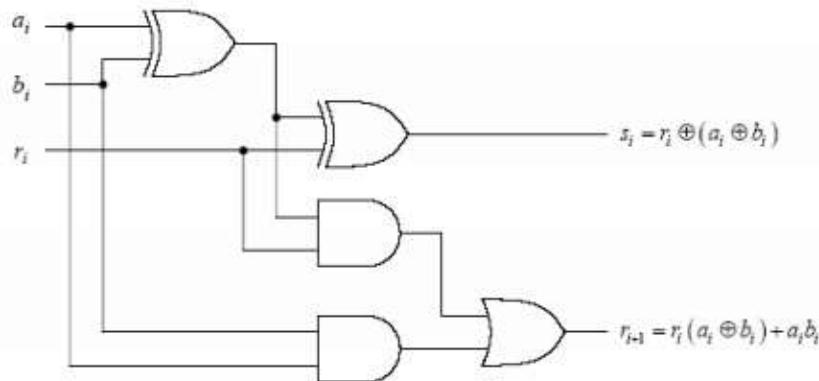
		$a_i b_i$	00	01	11	10
		r_i	0	0	1	0
r_i	0	0	1	0	1	0
	1	1	0	1	0	0

$$\begin{aligned}
 s_i &= r_i \overline{a_i b_i} + \overline{r_i a_i} b_i + r_i a_i b_i + \overline{r_i a_i} \overline{b_i} \\
 &= r_i (\overline{a_i b_i} + a_i b_i) + \overline{r_i} (\overline{a_i b_i} + a_i \overline{b_i}) \\
 &= r_i (\overline{a_i \oplus b_i}) + \overline{r_i} (a_i \oplus b_i) = r_i \oplus (a_i \oplus b_i)
 \end{aligned}$$

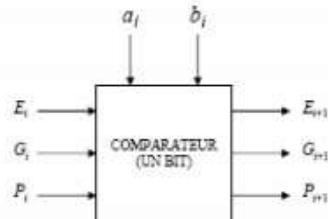
		$a_i b_i$	00	01	11	10
		r_i	0	0	1	0
r_i	0	0	0	1	0	0
	1	0	1	1	1	1

$$\begin{aligned}
 r_{i+1} &= r_i a_i + r_i b_i + a_i b_i \text{ ou } \textcolor{blue}{r_i \overline{a_i} b_i} + \textcolor{green}{r_i a_i \overline{b_i}} + \textcolor{red}{a_i b_i} \\
 &= r_i (\overline{a_i b_i} + a_i \overline{b_i}) + a_i b_i \\
 &= r_i (a_i \oplus b_i) + a_i b_i
 \end{aligned}$$

Additionneur complet

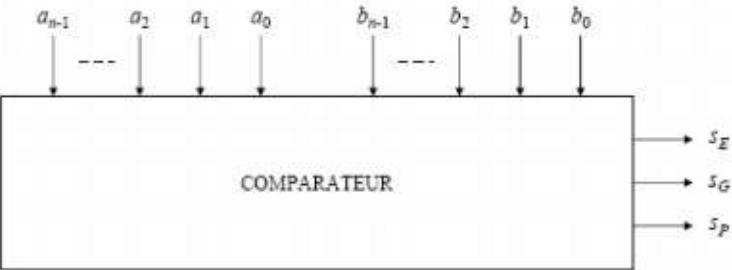


Réalisation avec des comparateurs 1 bit

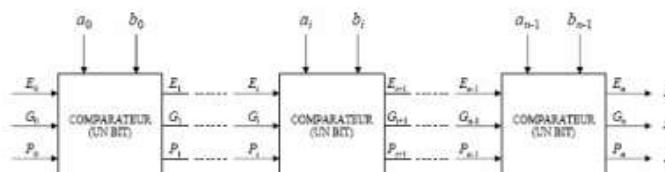


a	b	E	G	P
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Comparateur



Comparateur n bits

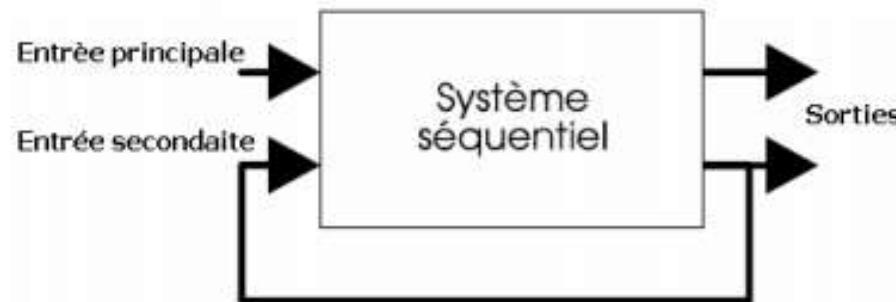


$$S_E S_G S_P = \begin{cases} 100 & \text{si } [a_{n-1} a_{n-2} \dots a_1 a_0] = [b_{n-1} b_{n-2} \dots b_1 b_0] \\ 010 & \text{si } [a_{n-1} a_{n-2} \dots a_1 a_0] > [b_{n-1} b_{n-2} \dots b_1 b_0] \\ 001 & \text{si } [a_{n-1} a_{n-2} \dots a_1 a_0] < [b_{n-1} b_{n-2} \dots b_1 b_0] \end{cases}$$

Circuits séquentiels

Les fonctions de sortie dépendent non seulement de l'état des variables d'entrée mais également de l'état antérieur de certaines variables de sortie (propriétés de mémorisation)

Table de vérité : on trouve en plus des entrées, la valeur de sortie à l'état précédent



Horloge (Clock)

- Les bascules sont généralement commandées par horloge
- Horloge : composant passant indéfiniment et régulièrement d'un niveau haut à un niveau bas (succession de 1 et de 0), chaque transition s'appelle un *top*.



Fréquence = nombre de changement par seconde en hertz (Hz)
Fréquence = 1/période
Une horloge de 1 hertz a une période de 1 seconde
..... 1 megahertz 1 millsec
..... 1 gigaHz 1 nanoSec

Les bascules

- Les circuits séquentiels de base sont les bascules
- Particularité : deux états stables = conservation de l'état de leur sortie même si la combinaison des signaux d'entrée l'ayant provoquée disparaît.
- Une bascule (flip-flop) a pour rôle de mémoriser une information élémentaire (mémoire à 1 bit).
- Une bascule possède deux sorties complémentaires Q et \bar{Q} .
- La mémorisation fait appel à un système de blocage (latch), dont le principe est représenté de la façon suivante.
- **Application:** Registres, Compteurs, etc.

Circuits synchrone et asynchrone

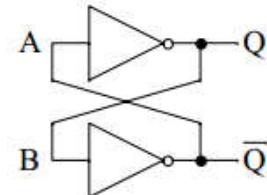
• Circuit synchrone

- Tous les éléments/composants du circuit devant être synchronisés le sont avec le même signal d'horloge

• Circuit asynchrone

- Tous les éléments/composants du circuit devant être synchronisés ne le sont pas avec le même signal d'horloge

Les bascules



$$\begin{cases} (Q = 1) \Rightarrow (B = 1) \Rightarrow (\bar{Q} = 0) \Rightarrow (A = 0) \Rightarrow (Q = 1) \\ (Q = 0) \Rightarrow (B = 0) \Rightarrow (\bar{Q} = 1) \Rightarrow (A = 1) \Rightarrow (Q = 0) \end{cases}$$

Une bascule ne peut donc être que dans deux états:

"1" : $(Q = 1, \bar{Q} = 0)$ "0" : $(Q = 0, \bar{Q} = 1)$

Les interconnexions interdisent les deux autres combinaisons :
 $Q = \bar{Q} = 0$ ou $Q = \bar{Q} = 1$.

Les bascules RS

- Les bascules les plus fréquemment utilisées sont réalisées avec deux portes NOR ou NAND.

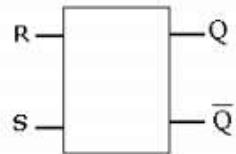
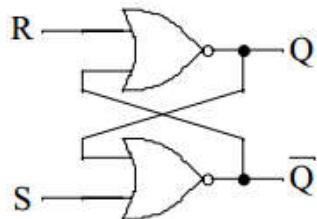


Schéma simplifié

(R) Set: Mise à 1

(S) Reset: Mise à Zéro

Les bascules RS

S	R	Q	\bar{Q}
0	0	Q	\bar{Q}
1	0	1	0
0	1	0	1
1	1	0	0

Sortie inchangée

Set: Mise à 1

Reset: remise à zéro

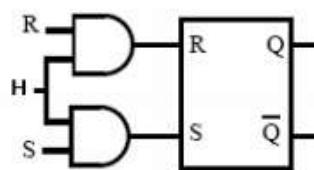
État interdit

$$Q = \overline{R + \bar{Q}}$$

$$\bar{Q} = \overline{S + Q}$$

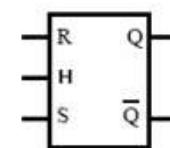
Bascule RS synchrone ou RST

Commandé par un signal horloge



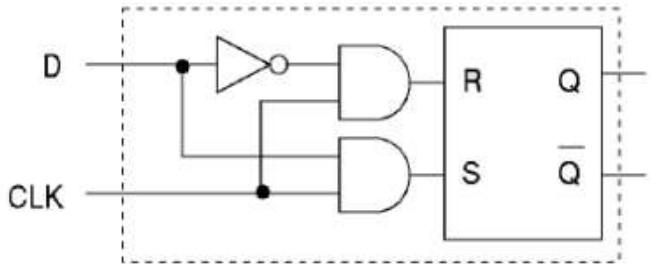
H = 1 → lecture

H = 0 → mémorisation



Bascule D

Pour éliminer l'état interdit $S=R=1 \rightarrow Q=\overline{Q}$



Bascule D

D	H	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

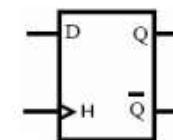
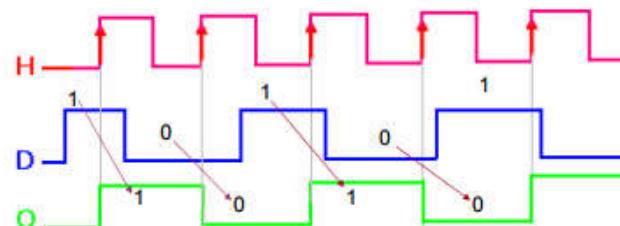
Versions condensées

(H=1)		D	Q_{n+1}
		0	0
		1	1

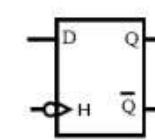
		H	Q_{n+1}
		0	0
		1	D

Bascule sur front d'horloge

Principe: saisir l'information lors du changement d'état de l'horloge



Front montant



Front descendant

Bascule JK asynchrone

- JK = variante de RS
- Similaire à RS mais ajoute le cas R=S=1
- Si $J = K = 1$ alors $Q_{n+1} = \overline{Q_n}$

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JK		00	01	11	10
Q_n		0	0	1	1
		1	1	0	0

J	K	Q_{n+1}
0	0	Q
0	1	0
1	0	1
1	1	Q

$$Q_{n+1} = J\overline{Q_n} + \overline{K}Q_n$$

Utilisation des bascules

Les bascules sont utilisées pour créer des circuits:

- **Compteurs**

- **Registres**

- Mémorisation d'un mot mémoire, décalage vers la droite/gauche du mot ...

- **Mémoires (SRAM)**

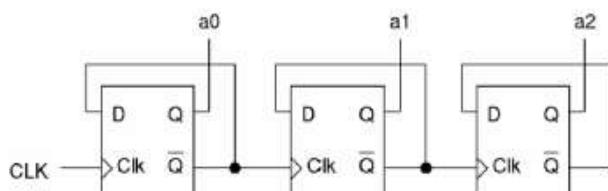
Exemple: compteur cyclique sur 3 bits

- Valeur en décimal sur 3 bits
- Incrémentation de +1 à chaque période d'horloge
- Repasse à 0 après 7

Exemple: Compteur modulo 8

Utilisation de 3 bascules D:

- Principe
 - Chaque bascule prend en entrée D un signal d'horloge
 - Fournit en sortie un signal d'horloge de fréquence divisée par 2
- En mettant en série les 3 bascules
 - 3 signaux d'horloge à 3 fréquences différentes
 - Représente les combinaisons de bits pour les valeurs de 0 à 7



Compteur asynchrone

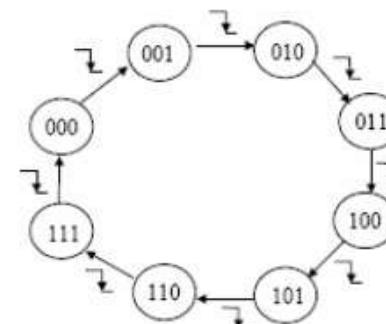
Les compteurs

Ensemble de n bascules interconnectées:

- Peuvent mémoriser des mots de n bits.
- Au rythme d'une horloge ils peuvent décrire une séquence déterminée c'est-à-dire une suite d'états binaires.
- De nombreuses applications industrielles:
 - Comptage du nombre de révolutions d'un moteur,
 - Division de fréquences,
 - Conversions de code, Conversion A/N et N/A, etc.
- Compteur binaire est dit modulo N lorsqu'il peut compter jusqu'à N-1, la Nième impulsion remet le compteur à zéro. $N=2^n$, où n représente le nombre d'étages.
 - Compteurs asynchrones
 - Compteurs synchrones

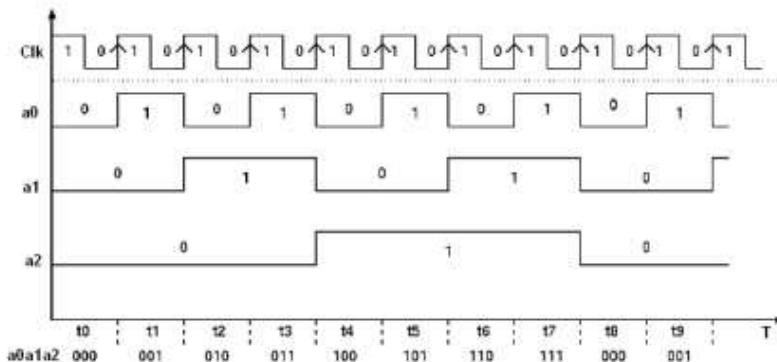
Exemple: Compteur modulo 8

Un compteur modulo 8 démarre à 0 et compte dans l'ordre binaire naturel de 0 à 7.



Chronogramme du compteur 3 bits

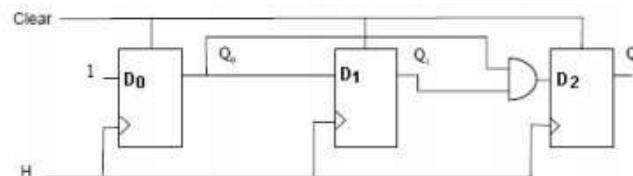
- Version idéale, ne prend pas en compte les temps de propagation à travers les bascules



Compteur synchrone

- Les bascules reçoivent en parallèle le même signal d'horloge.

$$\begin{aligned} D0 &= 1 \\ D1 &= Q0 \\ D2 &= Q0.Q1 \\ Dn &= Q0.Q1...Qn-1 \end{aligned}$$



Exemple: Compteur modulo 8

Registres

- Registres : Mémoires du microprocesseur de X bits (8, 16, 32, etc.)
- Composant localisé dans un processeur pour stocker des informations lors de l'exécution d'un programme par ce processeur (instruction, donnée, état du processeur, etc.)

Exemple: Registre 4 bits

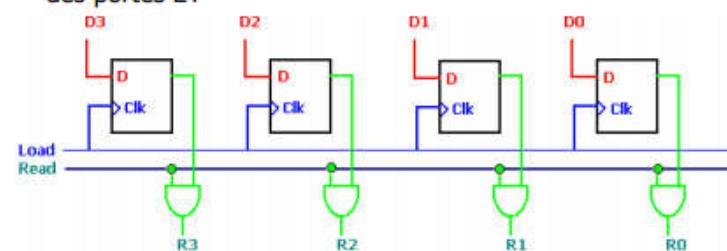
- 4 bascules D stockent les 4 bits
- 4 bits en entrée pour écrire le mot
- 4 bits en sortie pour récupérer la valeur du mot
- Une entrée L (pour « load ») précise si on conserve la valeur du mot stocké (valeur 0) ou écrit le mot avec les 4 bits en entrée (valeur 1).

Exemple: Registre 4 bits

Supposons que l'on ait 4 bits D0, D1, D2, D3 à transférer vers les sorties d'un registre R0, R1, R2, R3.

Le transfert est fait en deux étapes:

- Les valeurs de Di, i = 0, ..., 3 sont transférées vers les sorties Q des 4 bascules D
- Une impulsion sur la ligne Read permet le transfert vers la sortie des portes ET



Registre Parallèle/Parallèle 4 bits

Registre à décalage

Bascules interconnectées de façon à ce que l'état logique de la bascule de rang i puisse être transmis à la bascule de rang $i+1$.

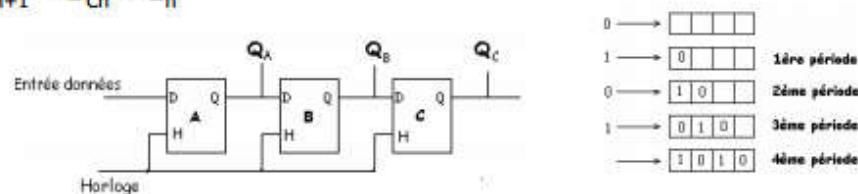
Exemple:

Les bascules sont commandées par le même signal horloge H . Sachant que dans une bascule D , l'état suivant $n+1$ de la sortie Q est égale à l'état présent D_n de l'entrée D , on a donc:

$$A_{n+1} = D_{An} = \text{information à l'entrée}$$

$$B_{n+1} = D_{Bn} = A_n$$

$$C_{n+1} = D_{Cn} = B_n$$



Chargement de la valeur 1010 dans un registre 4 bits