

Chapitre III

Etude des circuits combinatoires, codeurs, décodeurs, Multiplexeur

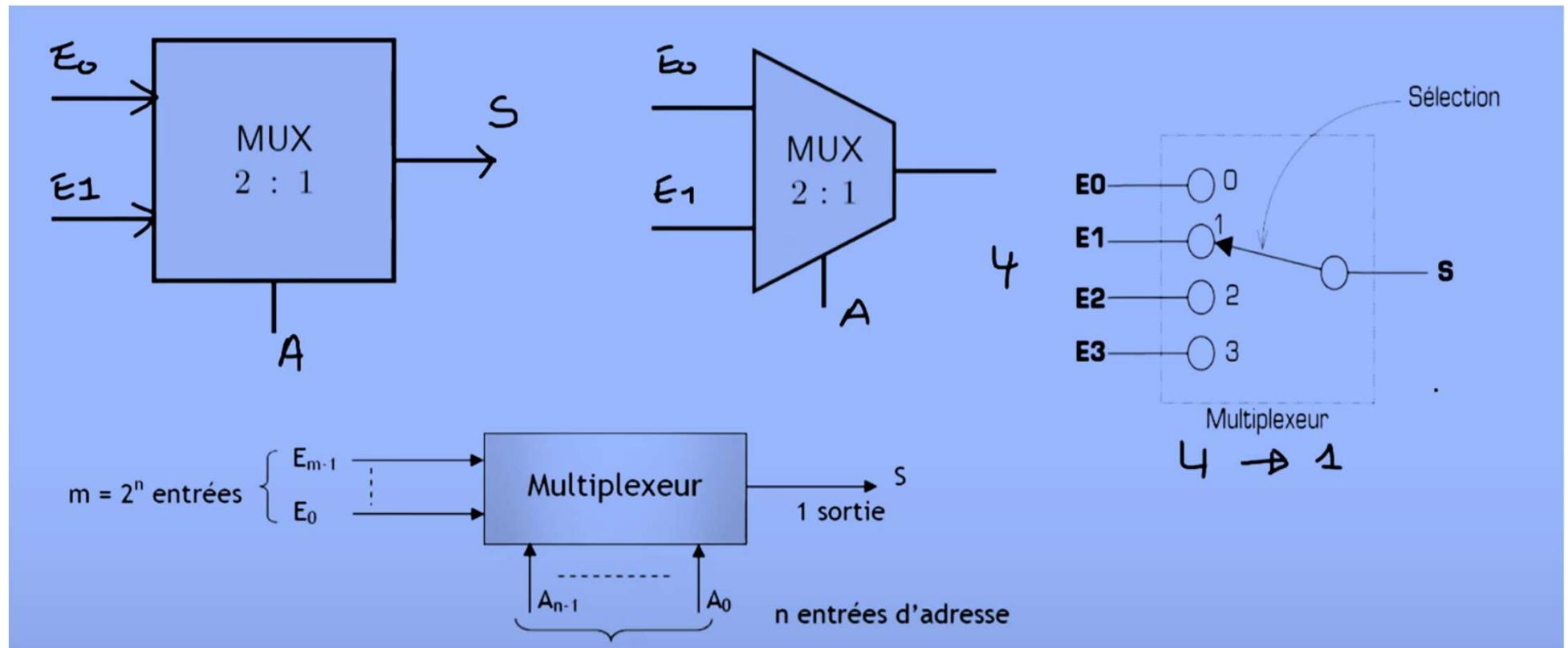
Multiplexeur et Démultiplexeur

1- Multiplexeur :

Définition :

- . Un multiplexeur possède plusieurs entrées et une seule sortie.
- . Il agit comme un sélecteur de données en orientant vers sa sortie la donnée présente sur l'une de ses entrées .
- . La sélection de l'entrée se fait alors à l'aide de n lignes d'adressage ou lignes de sélection
- . Habituellement, on a 2^n entrées et n bits de sélection, et une seule sortie sur un multiplexeur

Symbole



Application d'un Multiplexeur :

Les multiplexeurs ont de nombreuses applications. Ils peuvent par exemple être utilisés comme :

- sélecteur de données.
- convertisseur parallèle-série. Le multiplexeur reçoit en parallèle des données qu'il peut transmettre l'une après l'autre sur sa sortie en série.
- générateur de fonctions logiques.

Exemple 1 : Multiplexeur 4 vers 1

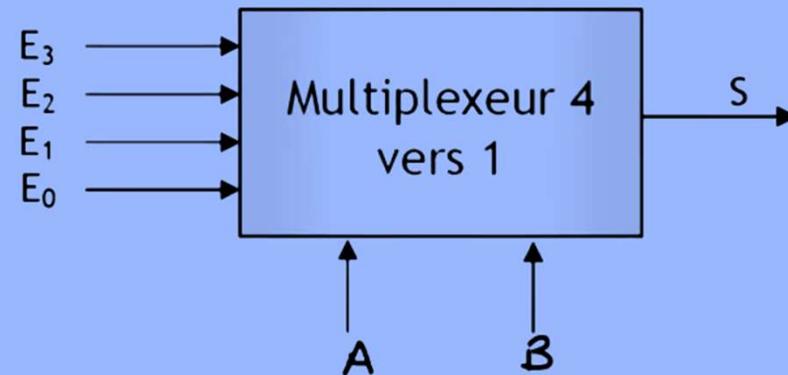
C'est un multiplexeur 4 entrées et 2 lignes d'adresse.

Table de vérité		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

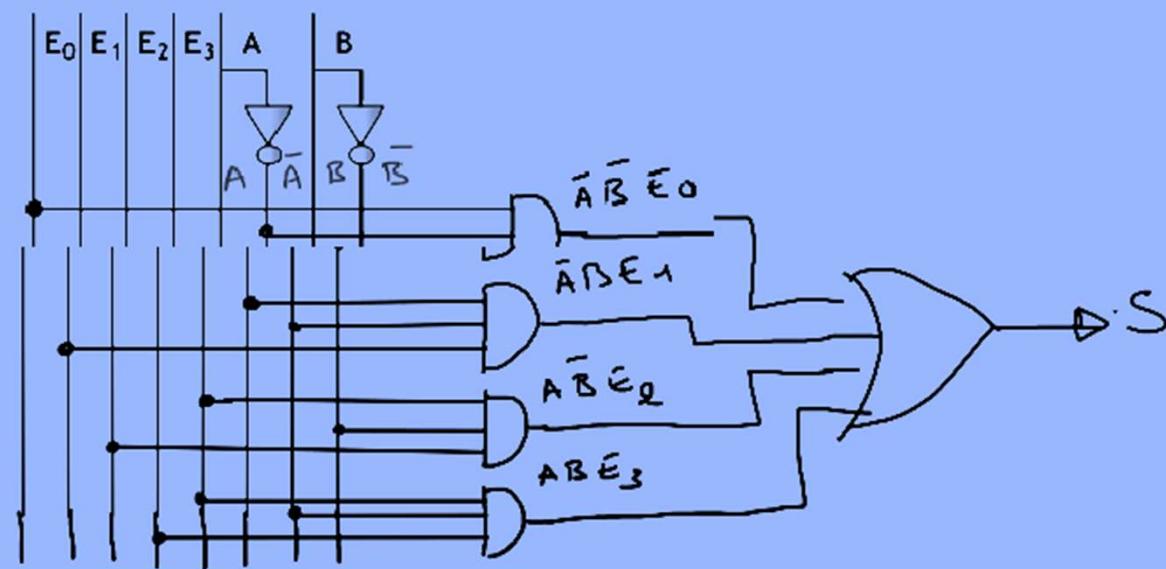
Equations de la sortie :

$$S = \dots$$

Schéma synoptique :



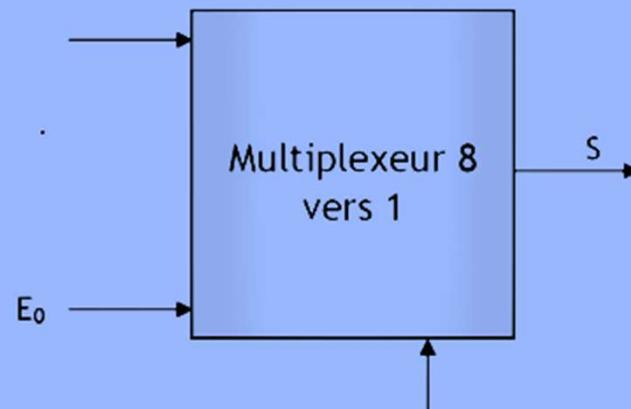
Logigramme :



Exemple 2 : Multiplexeur 8 vers 1

C'est un multiplexeur 8 entrées et 3 lignes d'adresse.

Schéma synoptique :



Equations de la sortie :

S =

Réalisation des fonctions logiques avec un Multiplexeur

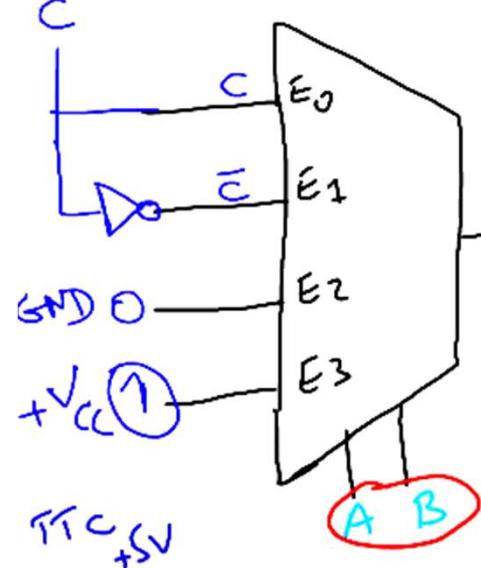
	A	B	C	S
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Travail demandé :

Réaliser la fonction logique de la sortie S avec

- 1- un multiplexeur 8 vers 1
- 2- un multiplexeur 4 vers 1
- 3- un multiplexeur 16 vers 1

2) Mux 4 → 1



A B	0 0	0 0	0 1	1 0	0 1	1 1	s
0 0	0 1	0 0	1 1	0 0	1 1	0 1	$E_0 \neq S = C$
0 1	0 0	1 0	0 1	0 1	0 1	1 0	$E_1 \neq S = \bar{C}$
1 0	1 0	0 0	0 1	0 1	0 1	0 0	$E_2 \neq S = 0$
1 1	1 1	0 1	1 1	1 1	1 1	1 1	$E_3 \neq S = 1$

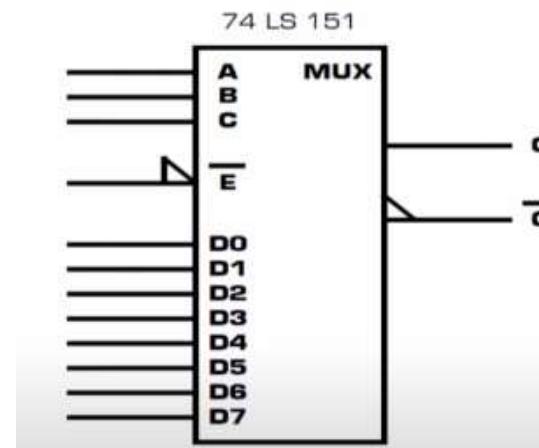
Exemple

Exemple de multiplexeur : le circuit intégré 74 LS 151

Le circuit 74 LS 151 est un multiplexeur 8 vers 1. Il comporte :

- 8 entrées de données D0 à D7
- 3 entrée de sélection A, B et C
- 1 entrée de validation E
- 2 sorties complémentaires Q et Q̄

Si le boîtier est validé (entrée de validation E activée à 0),
on retrouve sur la sortie du multiplexeur les informations
présentes sur l'entrée de donnée qui est sélectionnée avec les
entrées de sélection A, B et C.

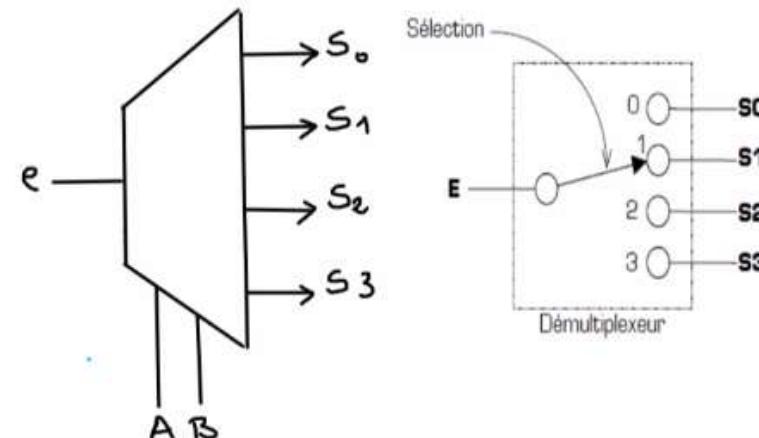
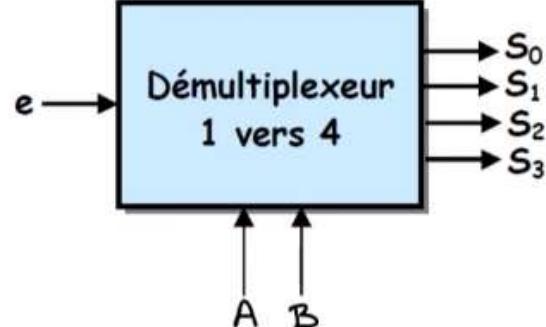


Le Démultiplexeur

Définition :

- Le démultiplexeur effectue l'opération inverse d'un multiplexeur, c'est-à-dire qu'il permet d'aiguiller l'information présente à l'entrée unique vers l'une des 2^n sorties. La sélection de la sortie se fait à l'aide de n lignes d'adressage.

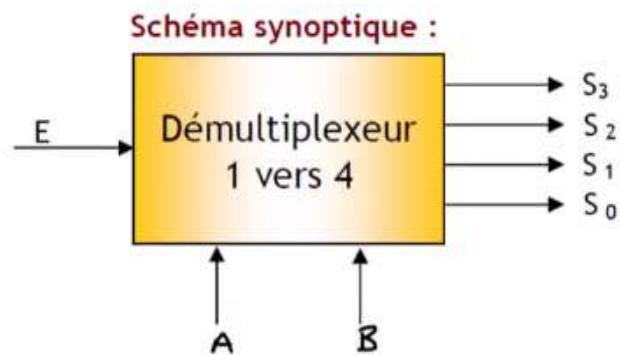
Symbole :



- Pour comprendre le principe, considérons un démultiplexeur à quatre sorties, donc deux lignes d'adressage et une ligne d'entrée (Démultiplexeur 1 vers 4).

C'est un démultiplexeur 4 sorties et 2 lignes d'adresse.

Table de vérité						
A	B	S ₀	S ₁	S ₂	S ₃	



Equations des sorties :

$$S_0 = \bar{A}\bar{B}E$$

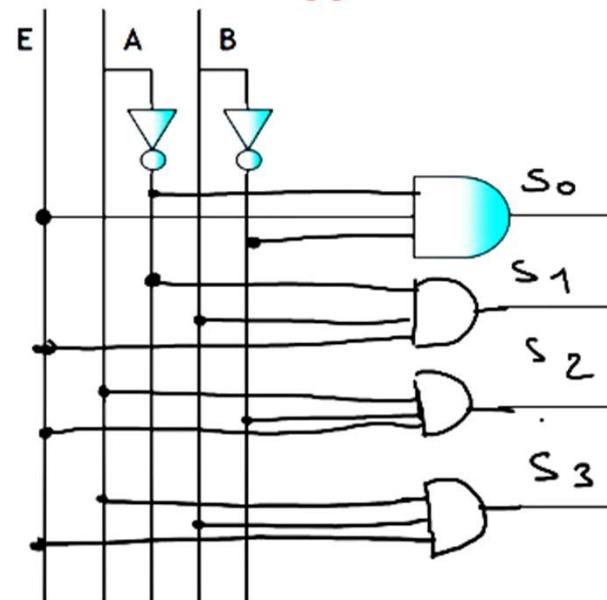
$$S_1 = \bar{A}BE$$

$$S_2 = A\bar{B}E$$

$$S_3 = ABE$$



Logigramme :



Démultiplexeur 1 vers 8

Table de vérité				S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
A	B	C	E	0	0	0	0	0	0	0	0
0	0	1	E	0	0	0	0	0	0	0	0
0	1	0	E	0	0	0	0	0	0	0	0
0	1	1	E	0	0	0	0	0	0	0	0
1	0	0	E	0	0	0	0	0	0	0	0
1	0	1	E	0	0	0	0	0	0	0	0
1	1	0	E	0	0	0	0	0	0	0	0
1	1	1	E	0	0	0	0	0	0	0	0

Équations des sorties:

$$S_0 = A\bar{B}\bar{C} \cdot E$$

$$S_1 = A\bar{B} \cdot C \cdot E$$

$$S_2 = \dots \cdot E$$

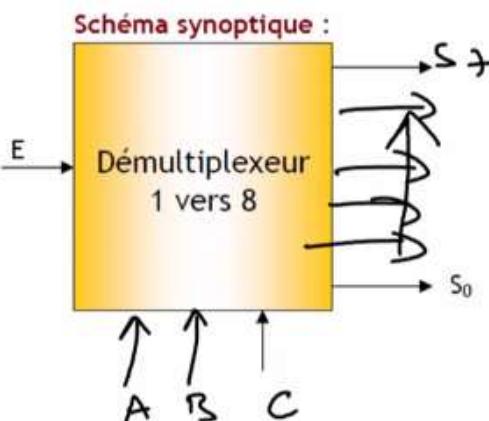
$$S_3 = \dots \cdot E$$

$$S_4 = \dots \cdot E$$

$$S_5 = \dots \cdot E$$

$$S_6 = A\bar{B}\bar{C} \cdot E$$

$$S_7 = \dots \cdot E$$



Application d'un Démultiplexeur

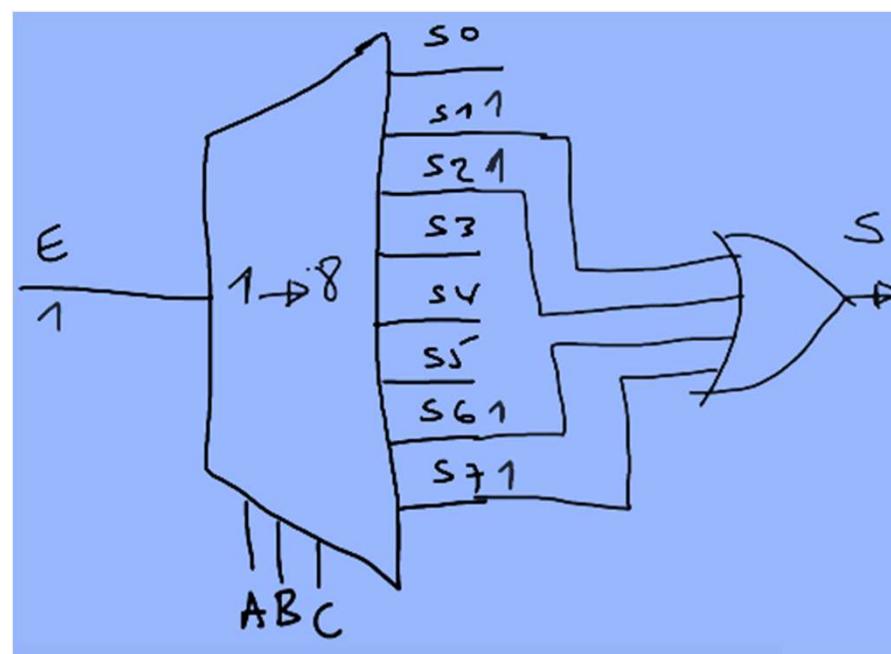
- Un démultiplexeur est un aiguilleur à une entrée de donnée E, n entrées d'adresse et m sorties.
Dans cette fonction le circuit joue le rôle inverse du multiplexeur.
- Convertisseur série-parallèle. Le multiplexeur reçoit en série des données une après l'autre qu'il peut transmettre sur sa sortie en parallèle.
- Un démultiplexeur dont l'entrée E est maintenue active au niveau 1 peut également servir de générateur de fonctions logiques.
En effet, puisque l'on retrouve sur les sorties toutes les combinaisons possibles des entrées d'adresse, une porte OU suffit pour fabriquer une fonction logique sous sa première forme canonique.

Réalisation des fonction logiques avec un Démultiplexeur

	A	B	C	S
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Travail demandé :

Réaliser la fonction logique de la sortie S avec un Démultiplexeur



Exemple

Description du circuit 74 LS 138

Le circuit 74 LS 138 est un démultiplexeur 1 parmi 8. Il comporte :

8 sorties de données Q0 à Q7

3 entrées de sélection A, B et C

3 entrées de validation G0, G1 et G2

Si le boîtier est validé ($G0=1$ ET $G1=0$ ET $G2=0$),

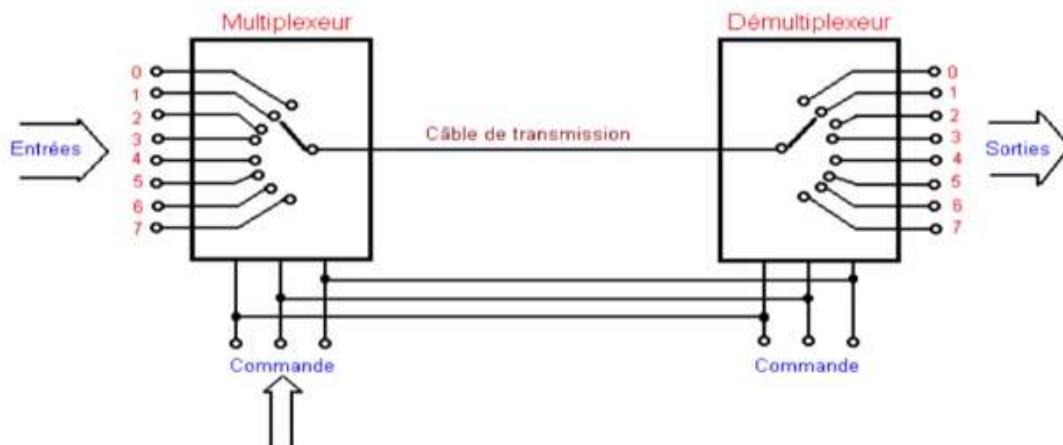
la sortie sélectionnée avec les entrées de

sélection A, B et C est à 0, les 7 autres sorties étant à 1.

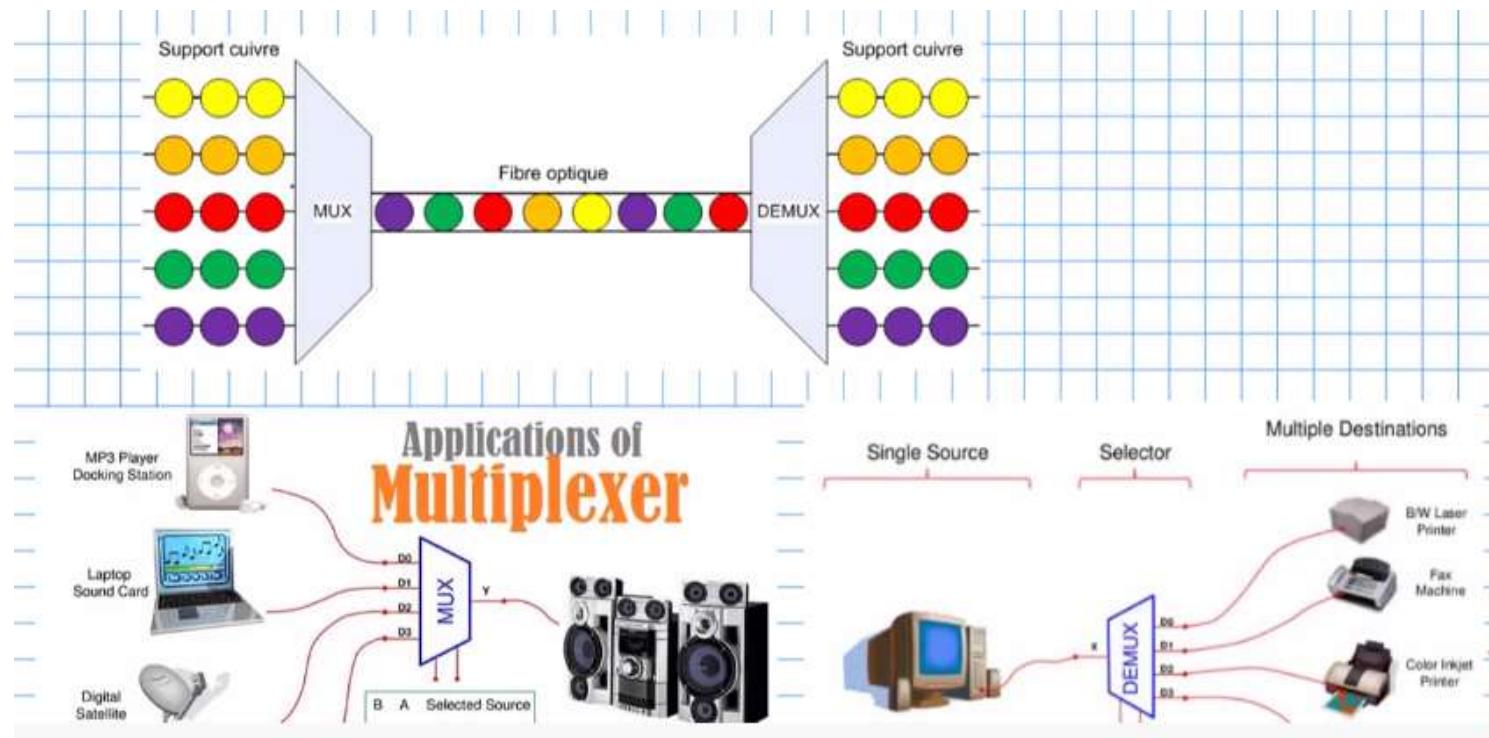
Symbol du démultiplexeur 74 LS 138 :

Symbole du démultiplexeur 74 LS 138 :

Exemple d'application d'un Multipléxeur et Démultiplexeur dans la transmission en Technologie Numérique



Représentation schématique de l'utilisation d'un multiplexeur et d'un démultiplexeur pour la transmission de plusieurs signaux sur un seul câble.



Logique Combinatoire (Décodeur, Codeur , Transcodeur)

Décodeur - codeur - transcodeur

Généralités

Les circuits de transformation des codes font la transposition des données d'un code à un autre.

Ils jouent le rôle d'interprète :
entre l'homme et la machine (codeur)
entre la machine et l'homme (décodeur)
entre machine et machine (transcodeur).

Décodeur

La fonction de décodage consiste à faire correspondre à un code présent en entrée sur m lignes, un autre code en sortie sur n lignes avec en général $m \neq n$.



Application :

Suivant le type de décodeur, la sortie peut traduire deux fonctions:

- Convertisseur de code d'entrée à un code de sortie

Exemple: Un décodeur binaire octal possède 3 bits d'entrées permettant

$2^3=8$ combinaisons pour activer chacun des 8 sorties de l'octal.

- Sélecteur de sortie: Une seule sortie parmi les M disponibles est activée à la fois en fonction de la valeur binaire affichée à l'entrée. (aiguillage raille Train - MAL)
- Ces fonctions permettent d'activer (électionner) un circuit intégré parmi plusieurs.

Exemple1: décodeur 1 parmi 4 ou 2 vers 4

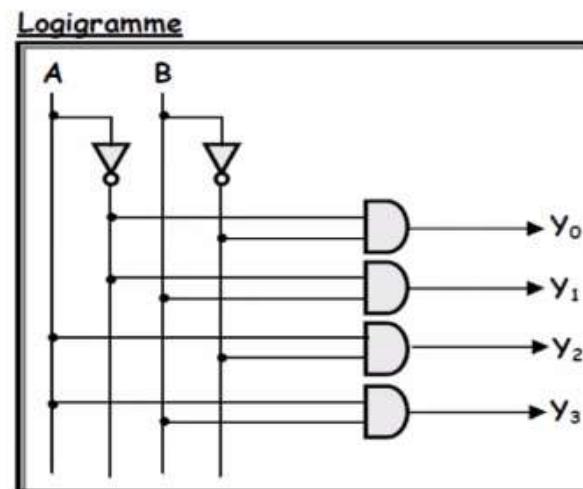


Table de vérité

Entrées		Sorties			
A	B	y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Équations logiques

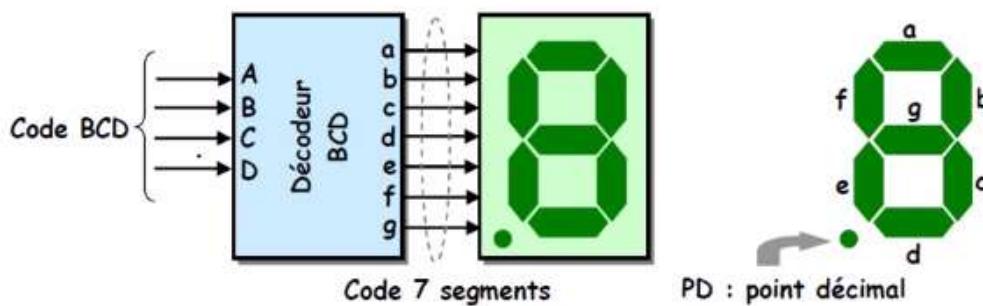
$y_0 = \bar{A}\bar{B}$
$y_1 = \bar{A}B$
$y_2 = A\bar{B}$
$y_3 = AB$



Exemple 2: Décodeur BCD – 7 Segments

Ce décodeur permet de convertir le code BCD, présent à son entrée sous 4 bits, en un code 7 segments disponible à sa sortie.

Il est utilisé pour commander un afficheur 7 segments afin d'écrire des chiffres de 0 à 9, certaines lettres et aussi quelques symboles.



On souhaite concevoir un décodeur BCD pour piloter un afficheur 7 segments à cathode commune.



Table de vérité

Nombre BCD	Entrées				Sorties						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

a

	CD	AB	00	01	11	10
00			1	1	X	0
01			1	X	X	1
11			1	X	X	1
10			0	1	X	1

e

	CD	AB	00	01	11	10
00			1	1	X	0
01			1	X	X	1
11			0	X	X	0
10			0	0	X	0

b

	CD	AB	00	01	11	10
00			1	1	X	1
01			1	X	X	0
11			1	X	X	1
10			1	1	X	0

f

	CD	AB	00	01	11	10
00			1	1	X	1
01			0	X	X	1
11			0	X	X	0
10			0	1	X	1

c

	AB	CD	00	01	11	10
00	1	1	X	1		
01	0	X	X	1		
11	1	X	X	1		
10	1	1	X	1		

g

	AB	CD	00	01	11	10
00	0	1	X	1		
01	1	X	X	1		
11	1	X	X	0		
10	0	1	X	1		

d

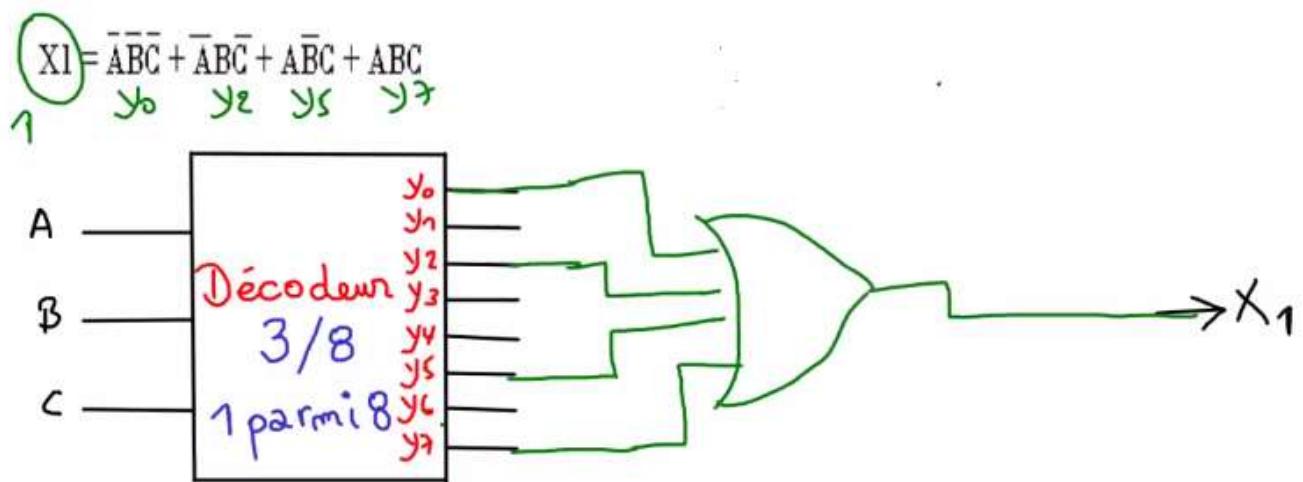
	AB	CD	00	01	11	10
00	1	1	X	0		
01	1	X	X	1		
11	1	X	X	0		
10	0	1	X	1		

Equations logiques

a = B + D + AC + $\bar{A}\bar{C}$	e = $\bar{A}B + \bar{A}\bar{C}$
b = $\bar{C} + AB + \bar{A}\bar{B}$	f = D + $\bar{A}\bar{B} + \bar{B}C + C\bar{A}$
c = A + C + \bar{B}	g = D + $\bar{A}C + C\bar{B} + B\bar{C}$
d = D + $\bar{A}B + B\bar{C} + \bar{C}\bar{A} + A\bar{B}C$	

Exemple 3 : Réalisation des fonctions logiques

$$X1 = \overline{ABC} + \overline{AB}\overline{C} + A\overline{B}\overline{C} + ABC$$



Codeur ou encodeur

C'est un dispositif qui effectue l'opération inverse du décodeur:
Une seule entrée parmi M est activée à la fois, ce qui correspond
à un nombre binaire en sortie.
On l'appelle aussi encodeur.

Principe d'un codeur 4 voies d'entrées et 2 bits de sortie

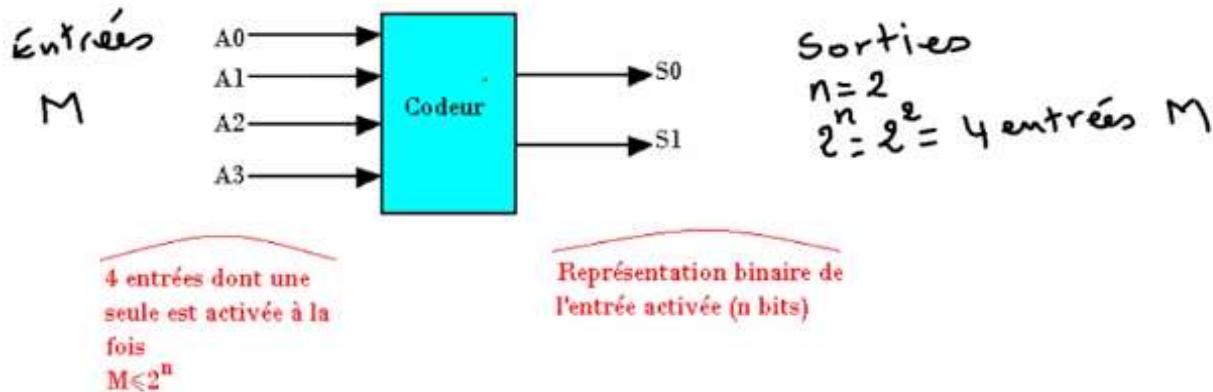


Table de vérité:

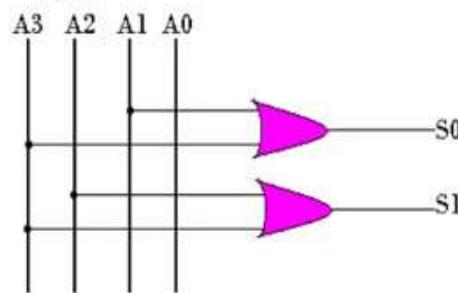
ENTREES				SORTIE	
Codage 1 parmi 2^n				Nombre binaire de n bits	
A ₃	A ₂	A ₁	A ₀	S ₁	S ₀
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Equation des sorties

$$S_1 = 1 \text{ si } (A_2 = 1) \text{ ou } (A_3 = 1); S_1 = A_2 + A_3$$

$$S_0 = 1 \text{ si } (A_1 = 1) \text{ ou } (A_2 = 1); S_0 = A_1 + A_2$$

Logigramme:



Codeur de priorité :

Si nous activons simultanément les entrées A1 et A2 du codeur ci-dessus, les sorties S1S0 présente le nombre 11 qui ne correspond pas au code de l'une ou de l'autre des entrées activés.

C'est plutôt le code qui représente l'activation de A3

Pour résoudre ce problème on utilise un codeur de priorité qui choisit le plus grand nombre lorsque plusieurs entrées sont activées à la fois.

Exemple lorsque A1 et A2 sont activées simultanément S1S0 sera égale à 10 qui représente l'activation de A2

A_3	A_2	A_1	A_0	S_1	S_0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

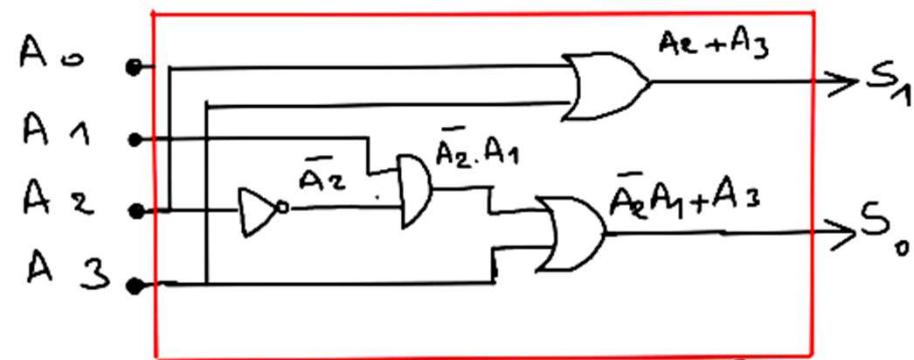
$$\begin{aligned} A_0 &\rightarrow 00 \\ A_1 &\rightarrow 01 \\ A_2 &\rightarrow 10 \\ A_3 &\rightarrow 11 \end{aligned}$$

$A_3 A_2$	00	01	11	10
$A_1 A_0$	00	1	1	1
00	0	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

$A_3 A_2$	00	01	11	10
$A_1 A_0$	00	0	1	1
00	0	0	1	1
01	0	0	1	1
11	1	0	1	1
10	1	0	1	1

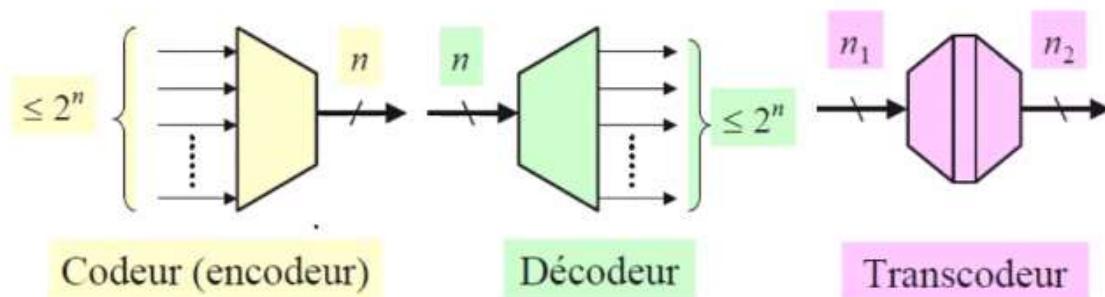
$$S_1 = A_2 + A_3$$

$$S_0 = \bar{A}_2 A_1 + A_3$$



Transcodeur

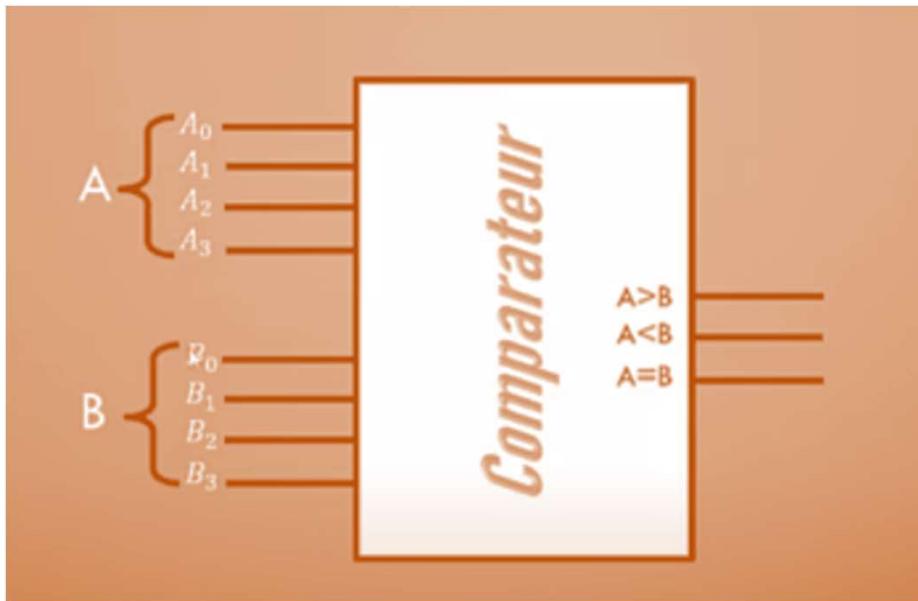
Un transcodeur transforme une information disponible en entrée sous forme donnée (généralement un code) en la même information, mais sous une autre forme (généralement un autre code).



Comparateur

Comparateur

Un comparateur permet de comparer 2 mots binaires, c'est à dire d'indiquer si ces 2 mots sont égaux, mais également, si ça n'est pas le cas, lequel est le plus grand. Il est basé sur l'utilisation d'un comparateur 1 bit.



Comparateur de 1 bit

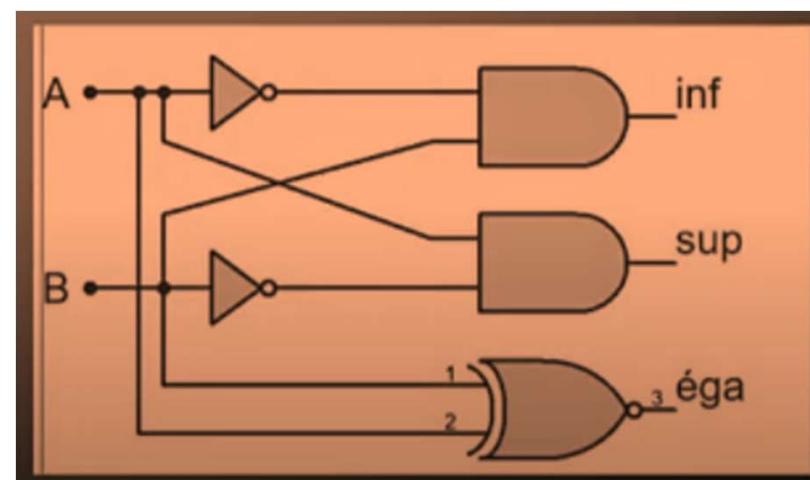
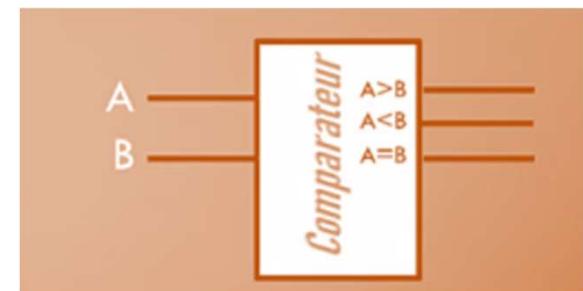
A	B	Sup	inf	éga
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

$$Sup = A\bar{B}$$

$$Inf = \bar{A}B$$

$$\text{éga} = \bar{A}\bar{B} + AB$$

$$\text{éga} = \overline{A \oplus B}$$



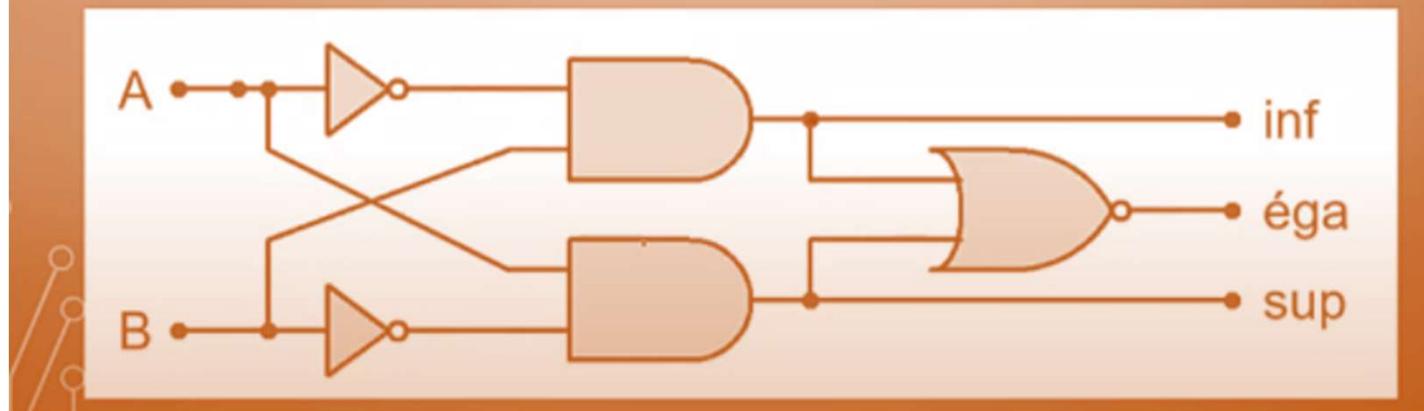
Comparateur de 1 bit



$$Sup = A\bar{B} \quad Inf = \bar{A}B \quad \text{éga} = \overline{A \oplus B}$$

$$Sup + Inf = \bar{A}B + A\bar{B} = A \oplus B$$

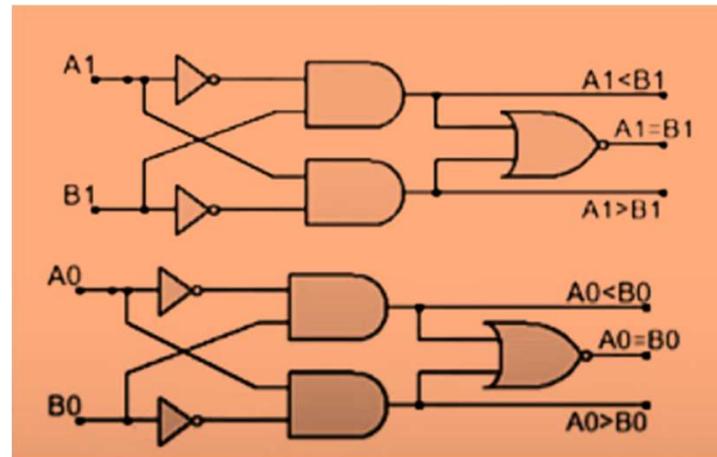
$$\overline{Sup + Inf} = \text{éga} = \overline{A \oplus B}$$



Comparateur de 2 bit

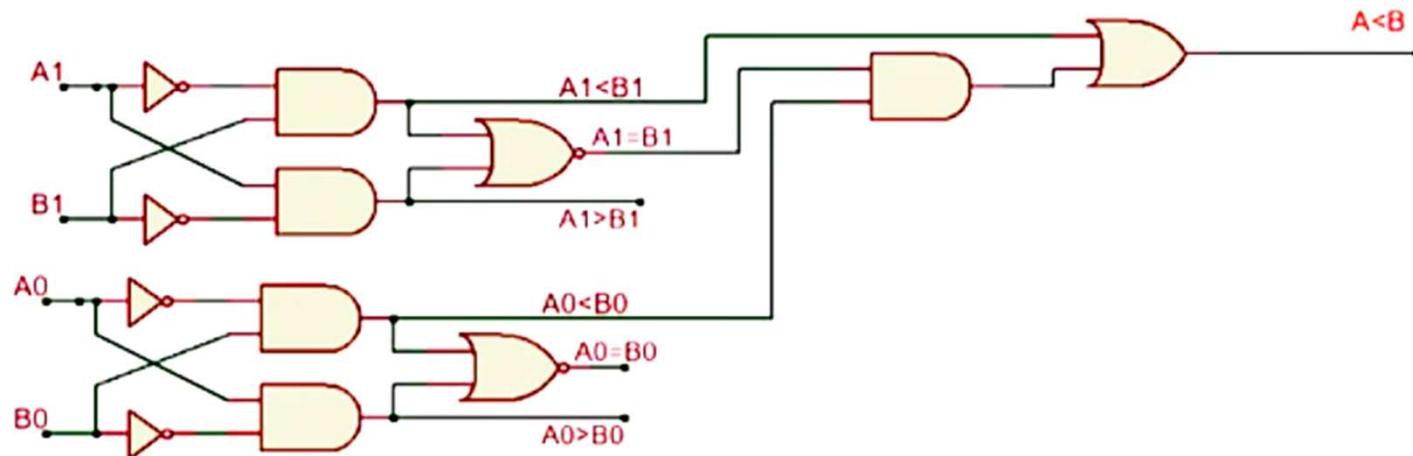
$$A = A_1 \ A_0 \quad B = B_1 \ B_0$$

$A > B \Leftrightarrow A_1 > B_1 \text{ ou } A_1 = B_1 \text{ et } A_0 > B_0$
 $A < B \Leftrightarrow A_1 < B_1 \text{ ou } A_1 = B_1 \text{ et } A_0 < B_0$
 $A = B \Leftrightarrow A_1 = B_1 \text{ et } A_0 = B_0$



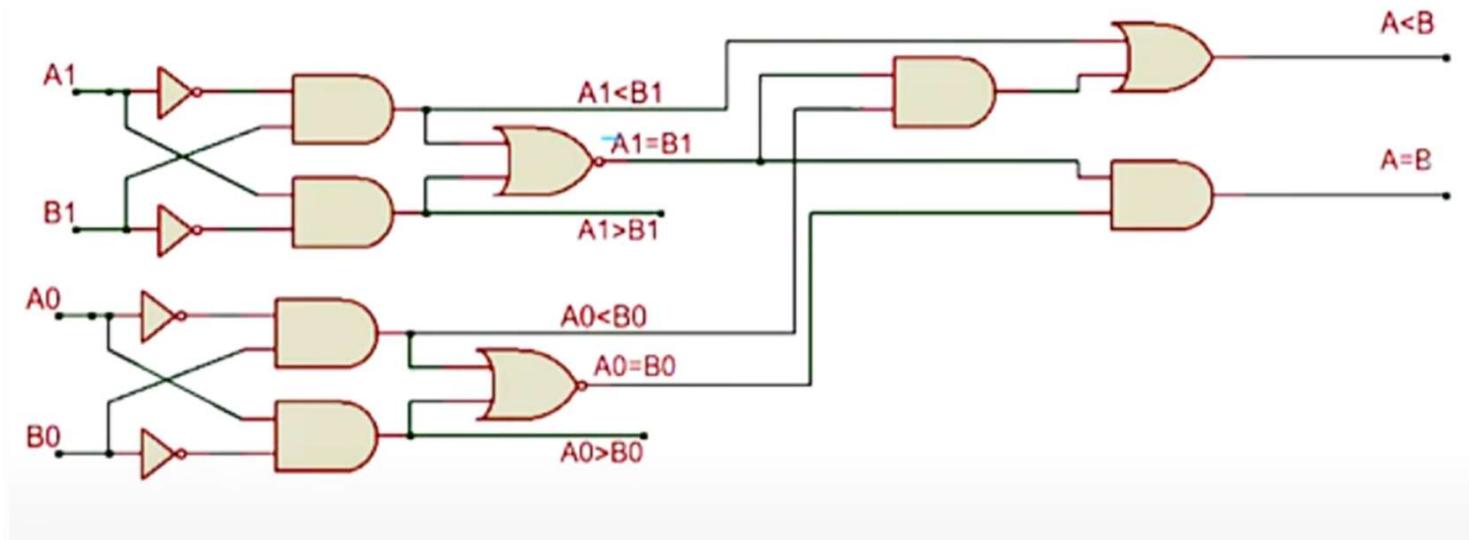
Comparateur de 2 bit

$$A \lessdot B \Rightarrow A_1 < B_1 \text{ ou } A_1 = B_1 \text{ et } A_0 < B_0$$



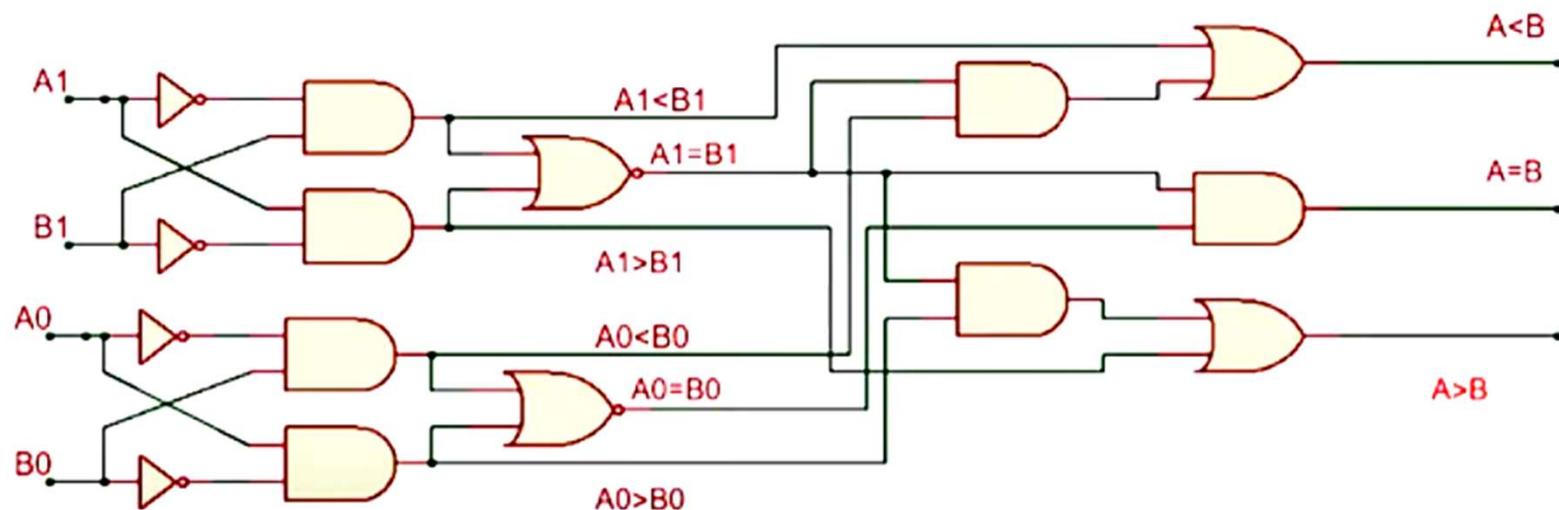
Comparateur de 2 bit

$$A \neq B \Rightarrow A_1 = B_1 \text{ et } A_0 = B_0$$



Comparateur de 2 bit

$$A \geq B \Rightarrow A_1 > B_1 \text{ ou } A_1 = B_1 \text{ et } A_0 > B_0$$



Comparateur de 2 bit

- A < B

$A_0A_1 \backslash B_0B_1$	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	1	0	1
10	1	0	0	0

- A = B

$A_0A_1 \backslash B_0B_1$	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

- A > B

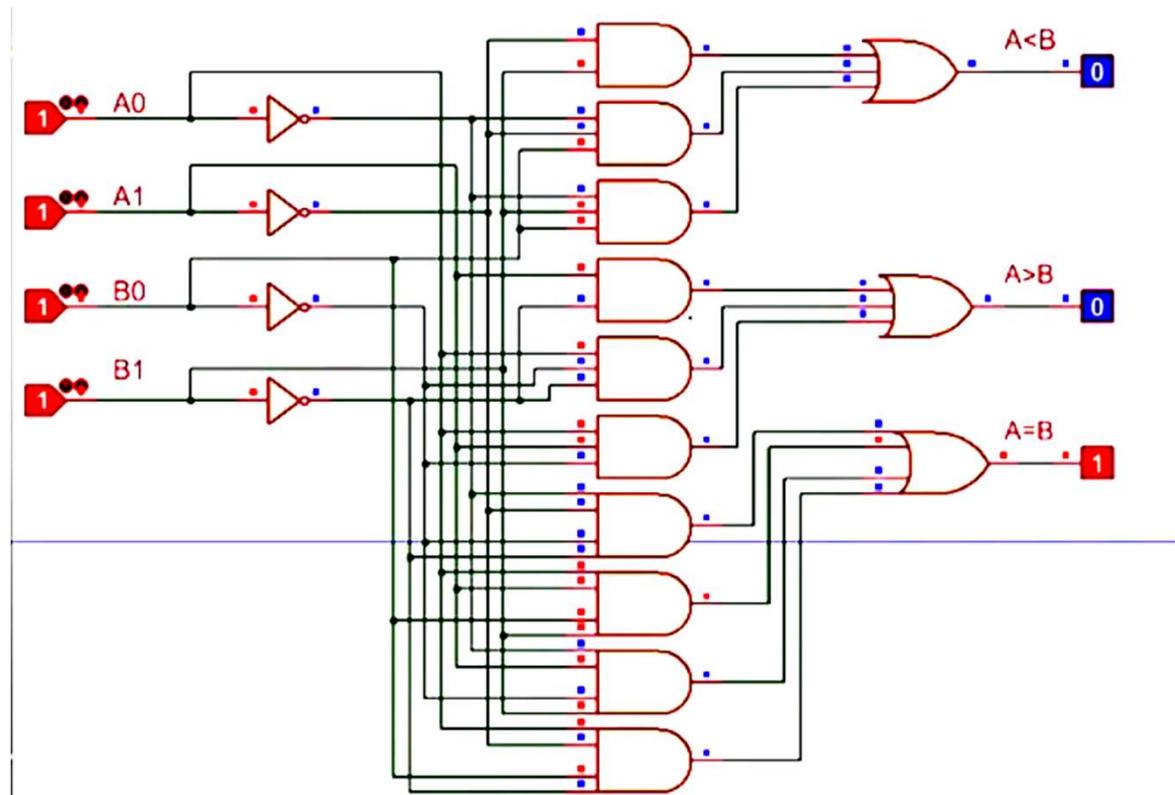
$A_0A_1 \backslash B_0B_1$	00	01	11	10
00	0	1	1	1
01	0	0	1	0
11	0	0	0	0
10	0	1	1	0

$$A < B = \overline{A}_1 B_1 + \overline{A}_0 \overline{A}_1 B_0 + \overline{A}_0 B_0 B_1$$

$$A = B = \overline{A}_0 \overline{A}_1 \overline{B}_0 \overline{B}_1 + \overline{A}_0 A_1 \overline{B}_0 B_1 \\ + A_0 A_1 B_0 B_1 + A_0 \overline{A}_1 B_0 \overline{B}_1$$

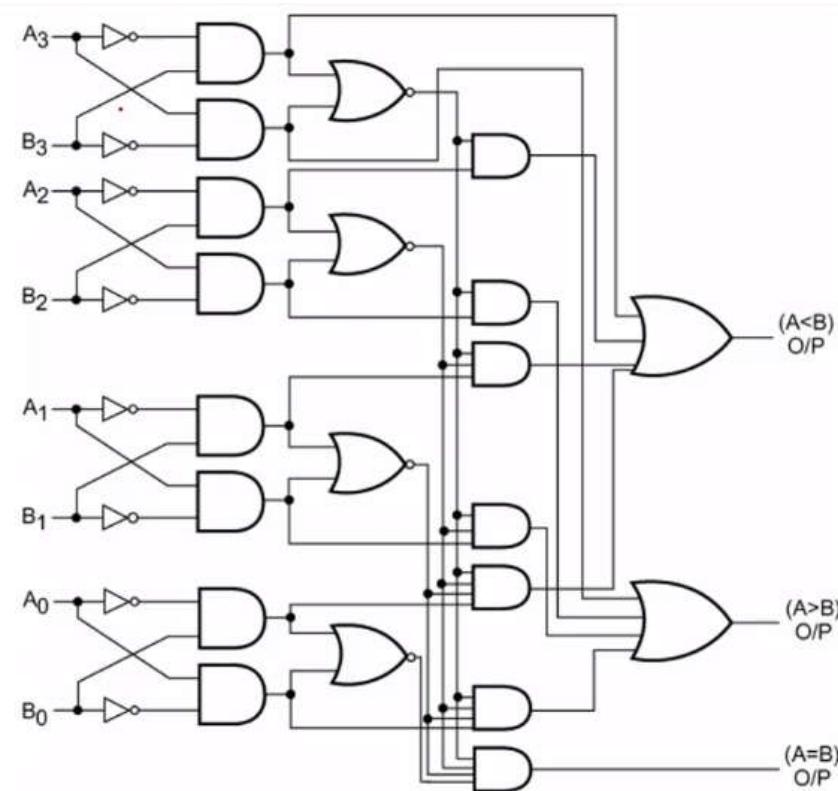
$$A > B = A_0 \overline{B}_0 \overline{B}_1 + A_0 A_1 \overline{B}_0 + A_1 \overline{B}_1$$

Comparateur de 2 bit



Famille TTL : 74 85

Famille CMOS 4000 : 4585



Chapitre IV

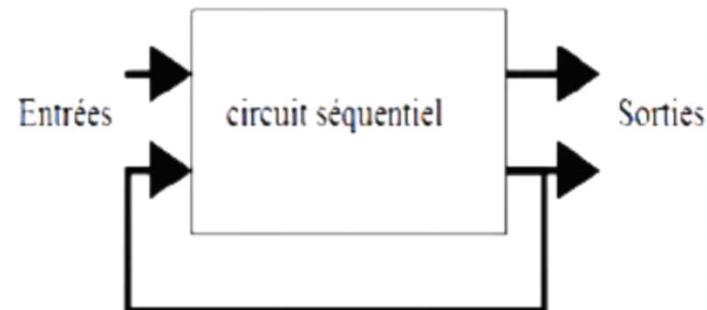
Etude des bascules et les systèmes séquentiels

Définitions

Circuits séquentiels:

Un circuit séquentiel est un circuit dont les sorties dépendent non seulement de l'état des entrées mais aussi de l'état antérieur de certaines sorties (notion de mémorisation).

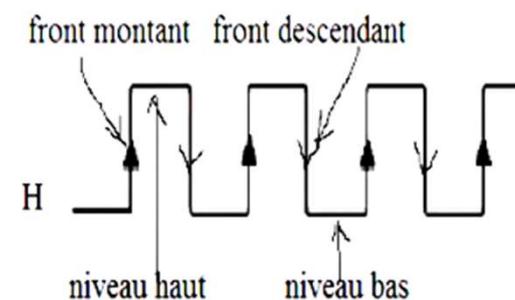
C'est un circuit dont le temps intervient dans la définition des sorties



Définitions

Horloge:

Composant passant régulièrement et indéfiniment d'un niveau haut (1) à un niveau bas (0) par la génération simultanée des impulsions des signaux électriques



Définitions

Logique synchrone:

Les signaux périodiques d'une horloge servent à synchroniser tous les changements d'état.

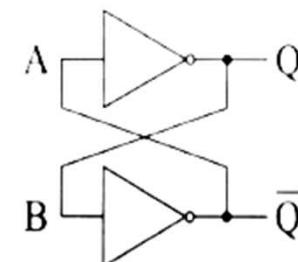
Logique asynchrone:

Les changements d'état des composants ne dépendent que du temps de réaction des composants et du temps de propagation des signaux.

Les Bascules

Système bistable (deux états ‘0’ et ‘1’) permettant de mémoriser une information élémentaire, il représente une mémoire à 1 bit ayant 2 états : Q et \bar{Q} , il utilise un mécanisme de verrou (latch)

➤ On peut vérifier que les états sont cohérents
 $(Q = 1) \Rightarrow (B = 1) \Rightarrow (\bar{Q} = 0) \Rightarrow (A = 0) \Rightarrow (Q = 1)$ Et $(Q = 0) \Rightarrow (B = 0) \Rightarrow (\bar{Q} = 1) \Rightarrow (A = 1) \Rightarrow (Q = 0)$



La Bascule RS

C'est une bascule à deux entrées R (Reset : remise à zéro) et S (Set : mise à un) tels que :

Si les deux entrées sont inactives $\rightarrow Q_+ = Q$

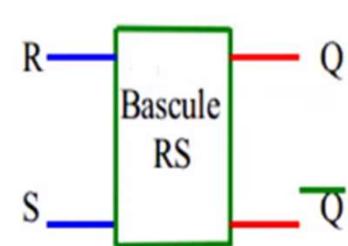
Si S est active seule $Q_+ = 1$

Si R est active seule $\rightarrow Q_+ = 0$

Un cas à éviter si les deux entrées sont actives à

la fois parce dans ce cas on obtient :

$$Q_{t+1} = \overline{Q_{t+1}} = 0$$



La Bascule RS

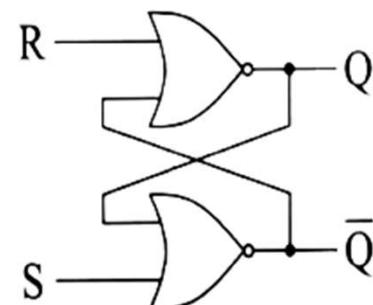
R	S	Q_+	\bar{Q}_+
0	0	Q	\bar{Q}
0	1	1	0
1	0	0	1
1	1	X	X

sorties inchangées

Set : remise à 1

Reset : remise à 0

à proscrire



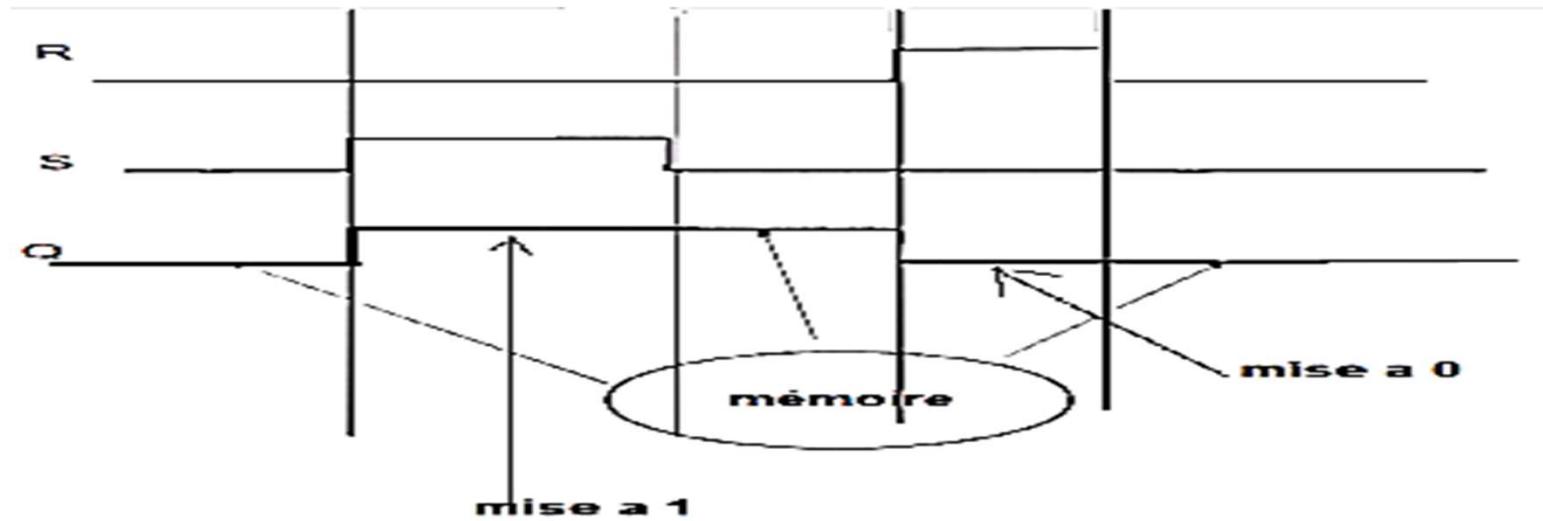
La Bascule RS

Table de transition (excitation)

Q	Q+	R	S
0	0		
0	1		
1	0		
1	1		

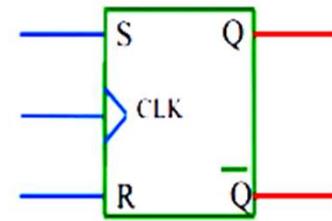
Q	Q+	R	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

La Bascule RS



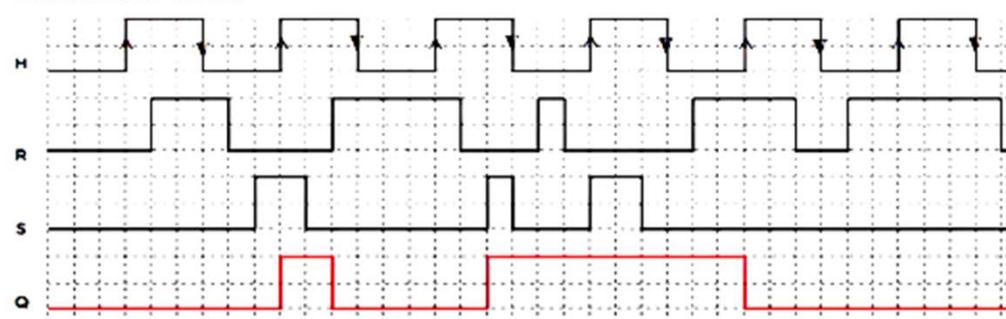
La Bascule RSH

C'est une bascule RS synchronisée par un signal d'horloge H (niveau haut en général).
Cette bascule souffre toujours d'un état interdit et elle a un état de mémorisation en plus.



La Bascule RSH en niveau haut (H=1)

- Lorsque H=0, la bascule est dans l'état mémoire.
- Lorsque H =1, la bascule fonctionne comme une bascule RS.

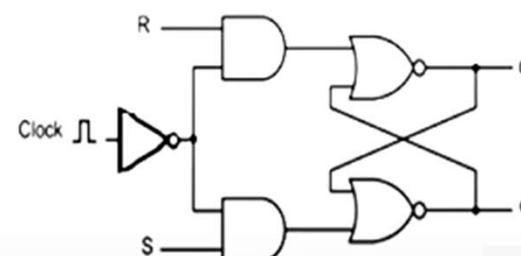
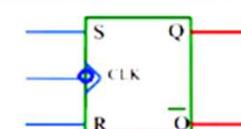


H	R	S	Q +	Q +
0	X	X	Q	Q
1	0	0	Q	Q
1	0	1	1	0
1	1	0	0	1
1	1	1	x	x

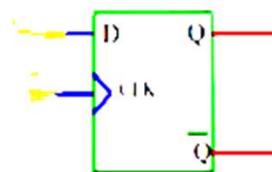
La Bascule RSH en niveau bas (H=0)

- Lorsque H=1, la bascule est dans l'état mémoire.
- Lorsque H =0, la bascule fonctionne comme une bascule RS.

H	R	S	Q +	\bar{Q} +
1	X	X	Q	\bar{Q}
0	0	0	Q	\bar{Q}
0	0	1	1	0
0	1	0	0	1
0	1	1	x	x



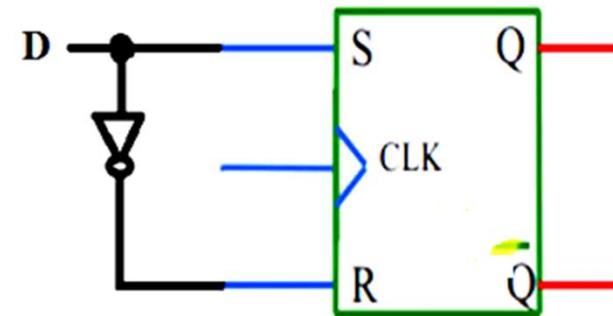
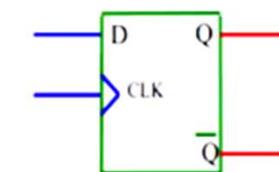
La Bascule D (Data)



Elle est conçue sur le même principe que la RSH. Obtenu à partir d'une bascule RSH en ne considérant que les deux combinaisons $(R,S) = (0,1)$ et $(1,0)$. Les entrées R et S sont remplacé par \bar{D} et D , d'où elle ne traite pas les cas $(0,0)$ et $(1,1)$, d'où elle élimine le deuxième cas de mémorisation et le cas interdit de la bascule RSH. Il existe deux types de bascule D selon le mode de synchronisation :

- Bascule D active sur le niveau (Bascule D-Latch)
- Bascule D active sur le front (Bascule Normale)

La Bascule D (Data)

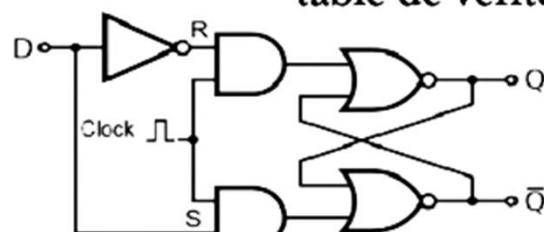


Bascule D-Latch: fonctionne sur le niveau (haut H=1).

H	D	$Q +$	$\bar{Q} +$
0	x	Q	\bar{Q}
1	0	0	1
1	1	1	0

Cette table de vérité peut être interprétée comme suit :

- Si $H=0$ alors $Q+=Q$ (mémorisation)
- Si $H=1$ alors $Q+=D$, donc on peut représenter sa table de vérité de la manière suivante :



H	$Q +$	$\bar{Q} +$
0	Q	\bar{Q}
1	D	\bar{D}

**Bascule D-Latch: fonctionne sur le niveau
(haut H=1).**

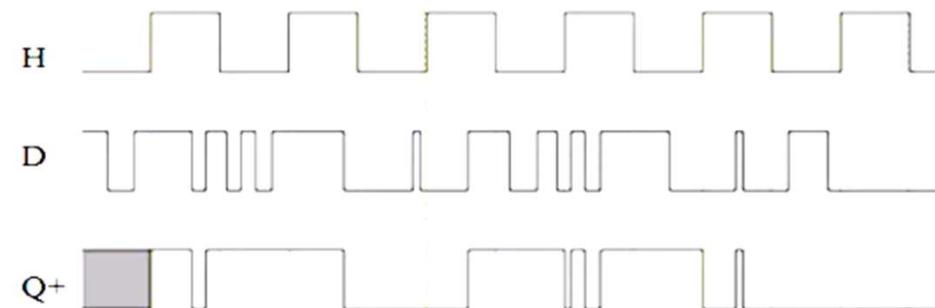


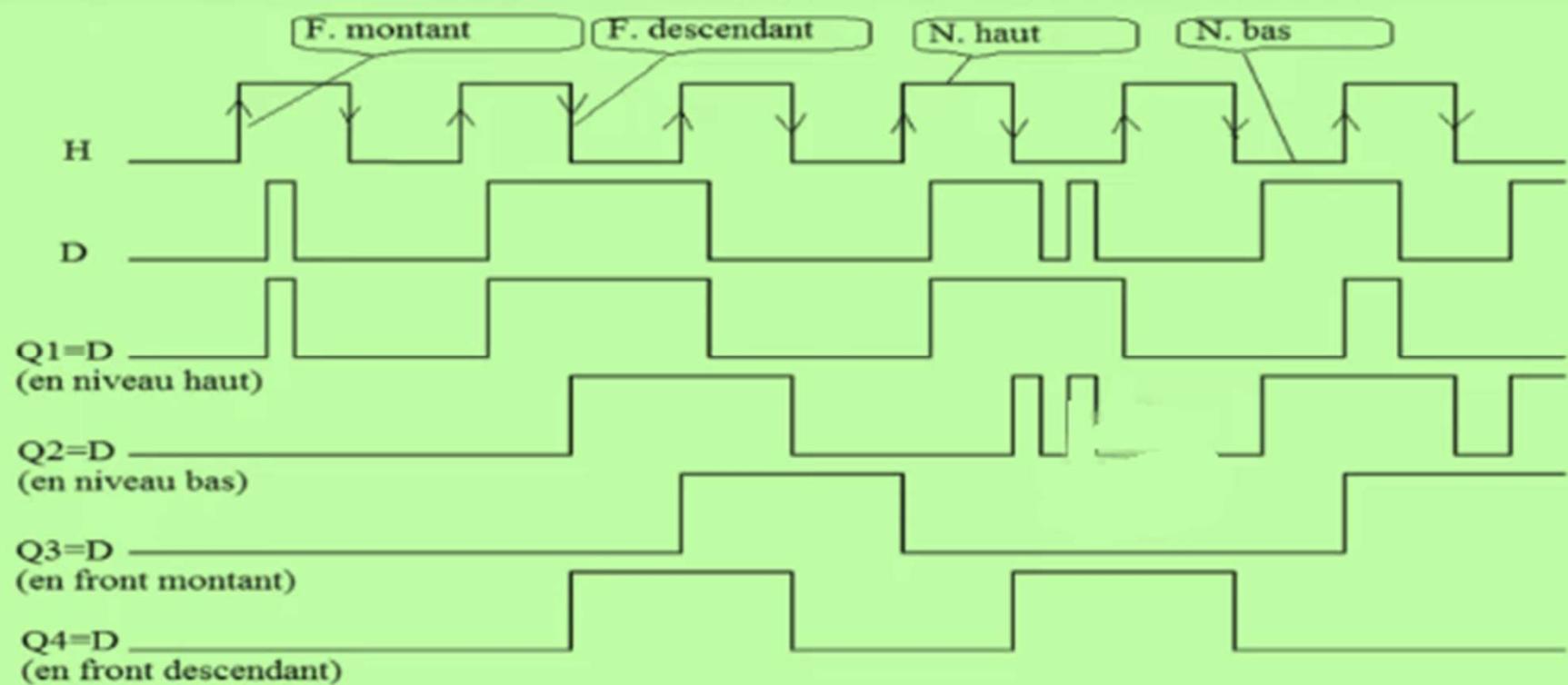
Table de transition (excitation)

Q	Q ⁺	D
0	0	0
0	1	1
1	0	0
1	1	1

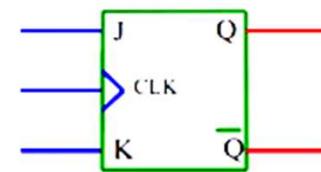
Bascule D Normale

La sortie Q est calculée en front, donc
l'information D doit être stable juste avant
l'arrivée du front de l'horloge

Chronogramme des différentes Bascules D



La Bascule JK



C'est une bascule synchrone (généralement en front d'horloge), offrant les fonctions : mémorisation, mise à 0, mise à 1, et assurant en plus la fonction basculement (inversement des sorties).

La bascule JK est la bascule la plus complète, offrant tous les modes de fonctionnement que l'on peut demander à une bascule.

La Bascule JK

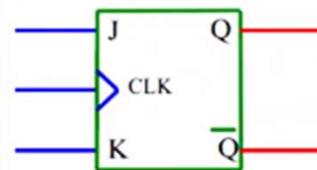
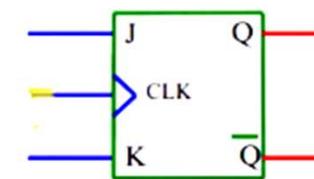


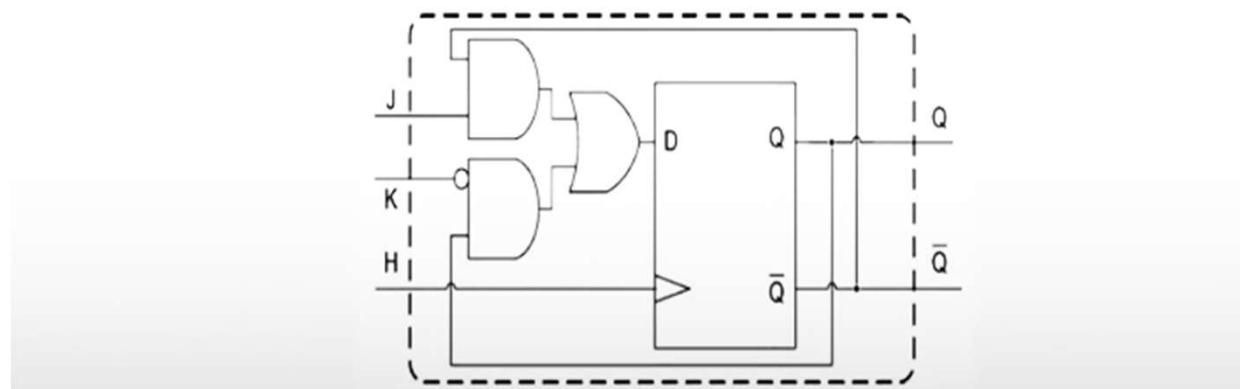
Table de vérité d'une Bascule JK en front montant

H	J	K	Q_+	\bar{Q}_+	Observation
0/1	x	x	Q	\bar{Q}	Mémorisation
↑	0	0	Q	\bar{Q}	Mémorisation
↑	0	1	0	1	Mise à 0
↑	1	0	1	0	Mise à 1
↑	1	1	\bar{Q}	Q	Basculement

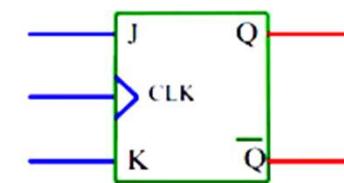
La Bascule JK



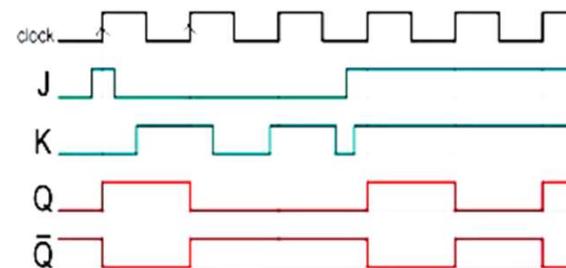
Réalisation d'une Bascule JK à l'aide d'une bascule D



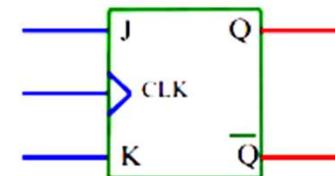
La Bascule JK



Chronogramme d'une bascule JK en front montant



La Bascule JK



Pour représenter Q_+ en fonction de J et K on doit établir sa table de vérité détaillée comme suit :

Entrées		Sortie	
J	K	Q	Q_+
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

Tableau de Karnaugh

$\backslash Q$	0	1
00	0	1
01	0	0
11	1	0
10	1	1

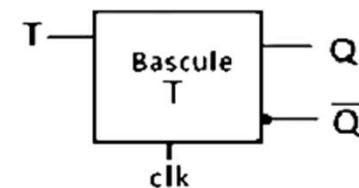
$$Q_+ = J\bar{Q} + \bar{K}Q$$

Table de transition (excitation)

Q	Q+	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

0	0
0	1
1	1
1	0
1	1
0	1
0	0
0	0

La Bascule T (toggle)

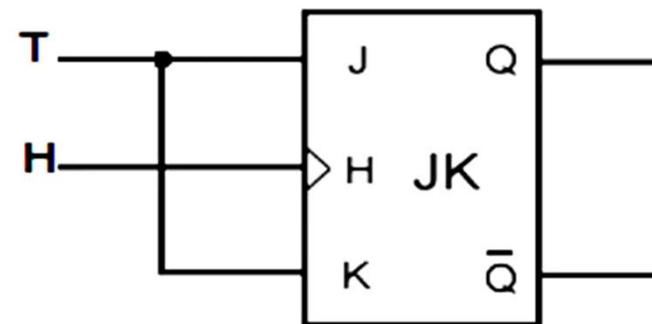


C'est une bascule qui fonctionne généralement en front de l'horloge avec une entrée T, elle a seulement deux fonctions : mémorisation ($Q_+ = Q$) si $T=0$ et basculement ($Q_+ = \overline{Q}$) si $T=1$

Table de vérité

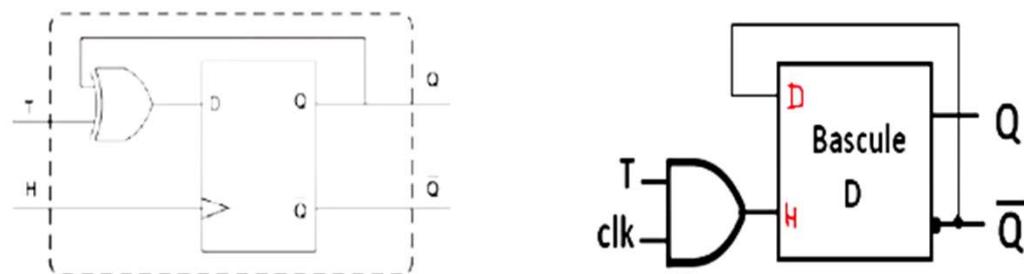
H	T	Q_+	\overline{Q}_+	observation
0/1	X	Q	\overline{Q}	Etat mémoire
\uparrow	0	Q	\overline{Q}	Mémorisation
\uparrow	1	\overline{Q}	Q	Basculement

Réalisation d'un bascule T



A l'aide d'une bascule JK

Réalisation d'un bascule T



A l'aide d'une bascule D

Chronogramme d'une bascule T en front montant

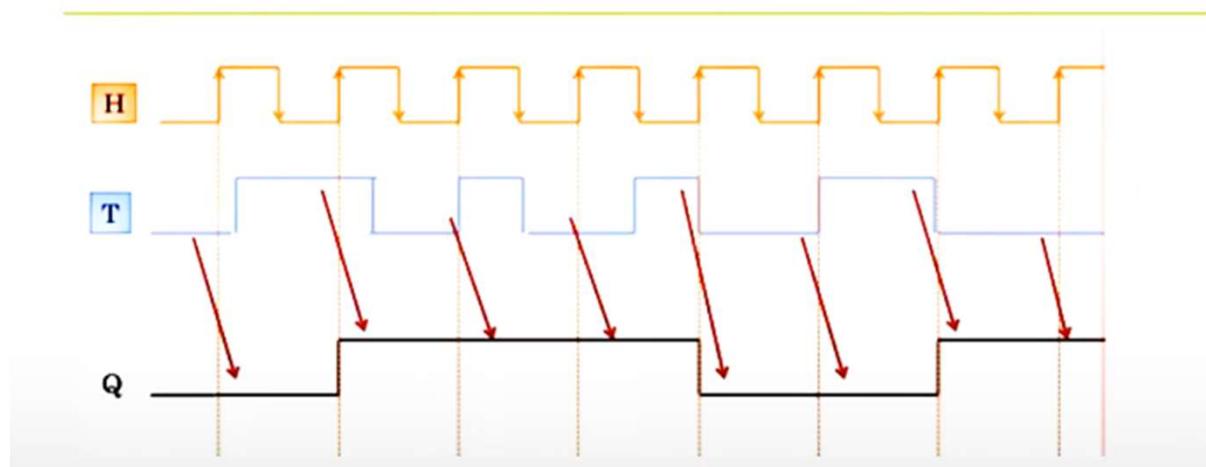
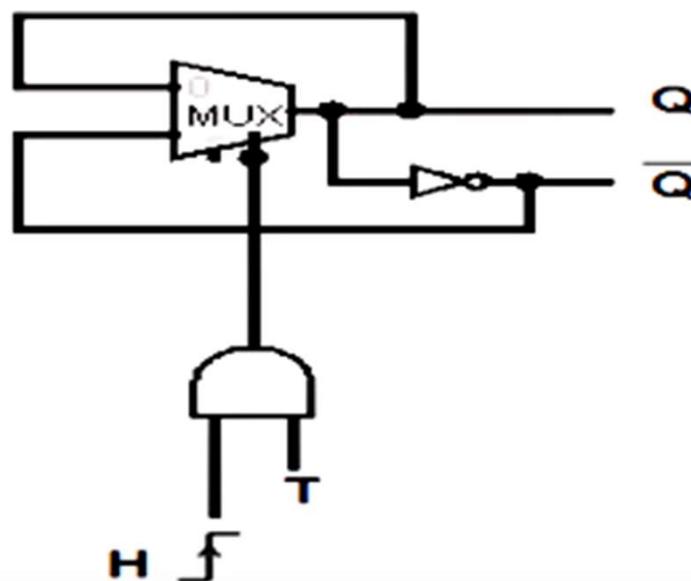


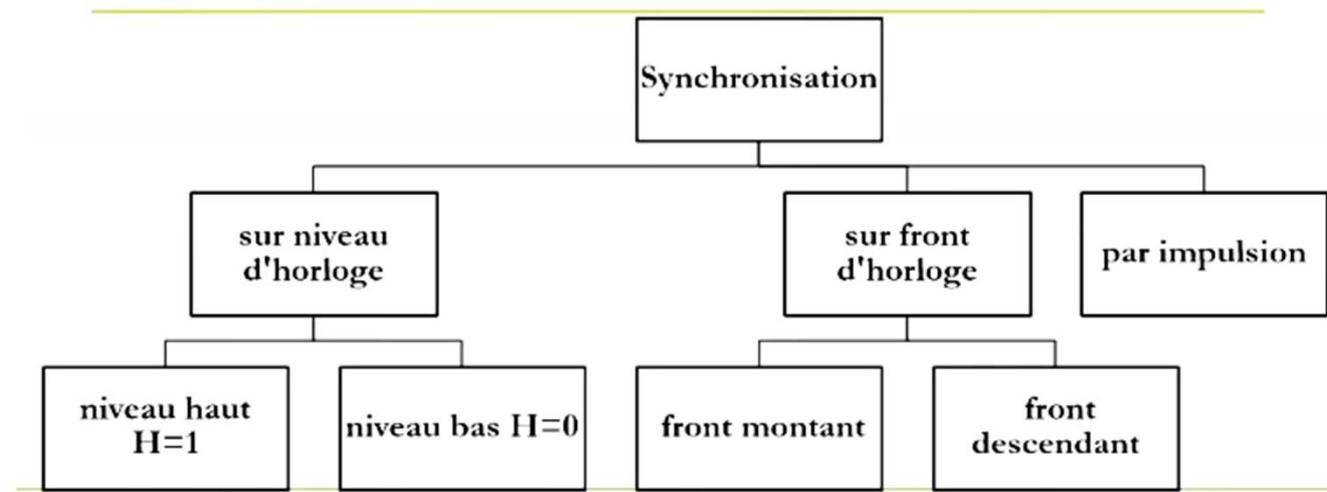
Table de transition (excitation)

Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0

Exercice : réalisez une bascule T à l'aide d'un
Mux 2→1 en front montant



Déclenchement d'une bascule



Déclenchement sur niveau

Les entrées sont prises en compte pendant un niveau fixé de l'horloge ($H = 1$ ou $H = 0$).

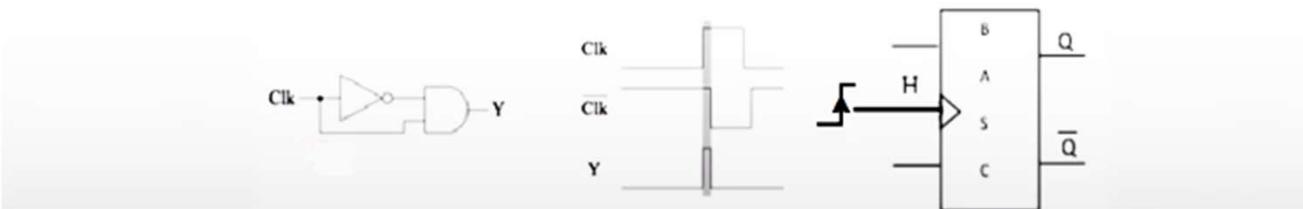
- Niveau haut : le signale H est utilisé directement
- Niveau bas : le signale H est utilisé avec un inverseur (généralement)

Déclenchement sur front

Les entrées sont prises en compte pendant un front de l'horloge qui est le moment de passage d'un niveau à autre (quelques nanosecondes).

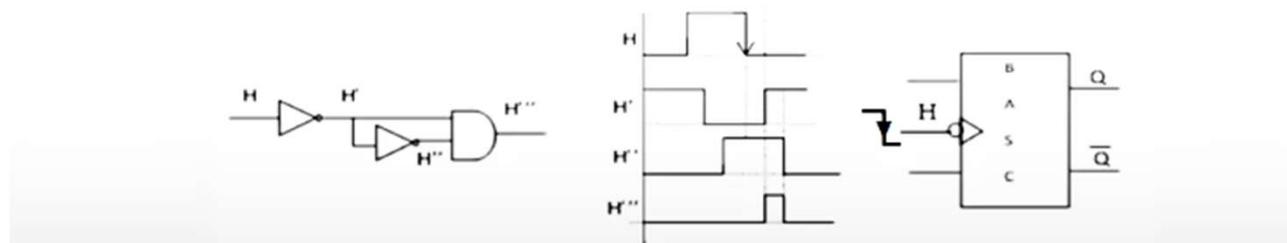
Front montant

Le moment du passage du niveau bas au niveau haut, un circuit est utilisé pour détecter ce passage appelé détecteur du front montant. L'inverseur introduit un retard de quelques nanosecondes de sorte que le front de \overline{CLK} a lieu légèrement plus tard que celui de CLK.



Front descendant

le moment du passage du niveau haut au niveau bas, un circuit est utilisé pour détecter ce passage appelé détecteur du front descendant.



Déclenchement sur impulsion

Ce type de synchronisation est utilisé dans les bascules maître-esclaves, où le changement des sorties a eu lieu pendant la transition complète d'une impulsion d'horloge.

Les registres

- ▶ Un registre est un ensemble de cellules mémoires constituées par des bascules.
- ▶ Le contenu d'un registre peut donc être considéré comme un nombre binaire ou un "mot" de n bits.

Applications

Les applications des registres sont nombreuses, parmi lesquelles :

- ▶ Stockage temporel de l'information
- ▶ Conversion série – parallèle
- ▶ Multiplication par une puissance de 2 (décalage vers la gauche).
- ▶ Division par une puissance de 2 (décalage vers la droite).

Types de registres

- ▶ **Registre de mémorisation**
- ▶ **Registre à décalage**

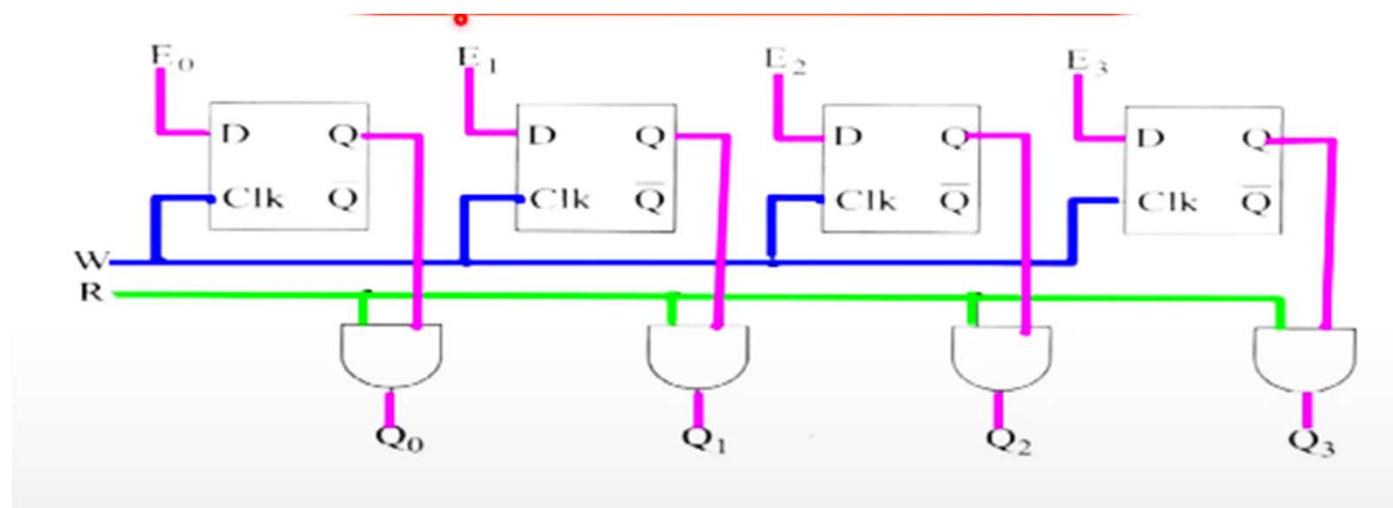
Registre de mémorisation

La fonction d'un tel registre est de "stocker / mémoriser" un mot de n bits.

Exemple de réalisation (à l'aide de bascules D) : chacune des cellules est une bascule D.

La sortie du registre mémorise le mot d'entrée tant que $H = 0$ ou 1 . Lorsque l'horloge présente un front montant, les données en sortie sont actualisées. Le registre peut être initialisé grâce aux entrées de forçage asynchrone qui peuvent forcer les sorties des bascules à 0 ou à 1 .

Exemple



Registre à décalage

Ce type de registre sert à décaler tous les bits d'un mot d'un ou plusieurs crans vers la droite ou vers la gauche. Il est constitué d'une série de bascules en cascade, où la sortie d'une bascule est branchée à l'entrée de la bascule suivante. Ces systèmes peuvent être utilisés pour effectuer des multiplications ou divisions par une puissance de 2, ou encore pour effectuer une conversion série – parallèle.

Exemple d'application : la multiplication par 2^n

Soit $N = (3)_{10} = (0011)_2$

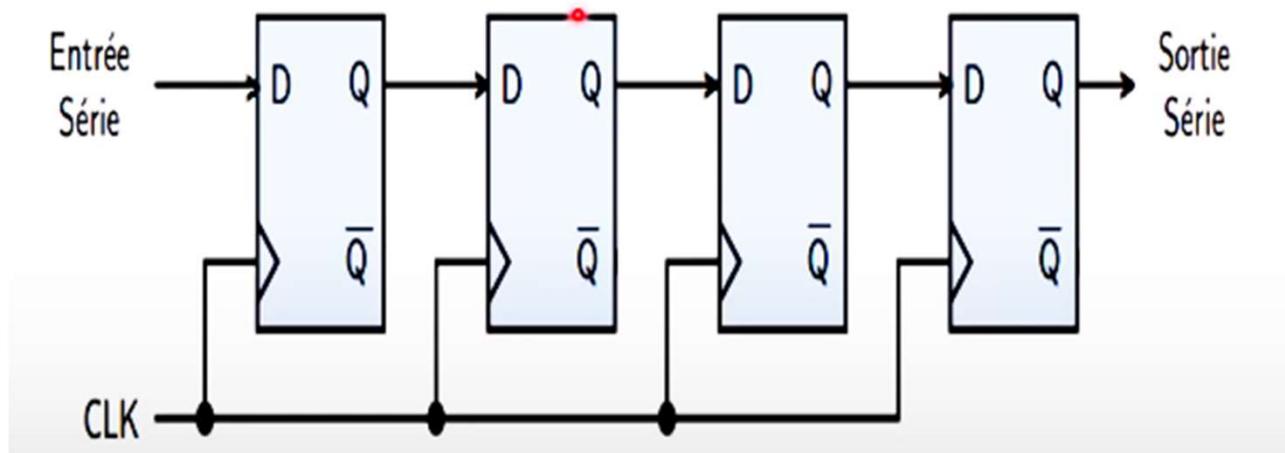
$2 \times N = 6 = 0110$

$2 \times (2 \times N) = 12 = 1100$

On constate que pour effectuer la multiplication d'un nombre par 2 il suffit de décaler tous les bits du nombre de 1 cran vers la gauche (vers les bits de poids fort).

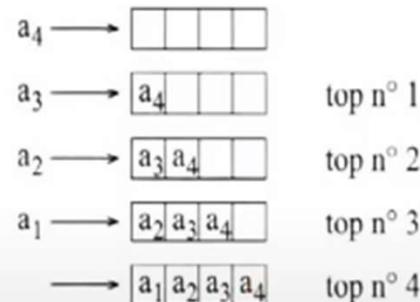
De la même façon, pour réaliser la division d'un nombre par 2 il suffit de décaler tous les bits du nombre de 1 cran vers la droite (vers les bits de poids faible).

Exemple



Types d'entrées

- **Série : l'information est présentée séquentiellement bit après bit à la 1^{ère} bascule.** À chaque coup d'horloge, un nouveau bit est présenté et ceux déjà chargés sont décalés d'un rang.



- **Parallèle : comme dans un registre de mémorisation.**

Types de sortie

L'information stockée dans un registre à décalage peut être lue de la même manière, (en série ou en parallèle).

Sens de décalage

Le décalage peut se faire vers la droite ou vers la gauche. Certains registres sont capables de décaler à droite ou à gauche selon le choix (registres à décalage universel).