

---

**ITI 1500**

**Hiver 2013**

# **Systemes Numériques I**

Professeur : Dr A. Karmouch, Bureau: CBY-A508

---

# Chapitre 6

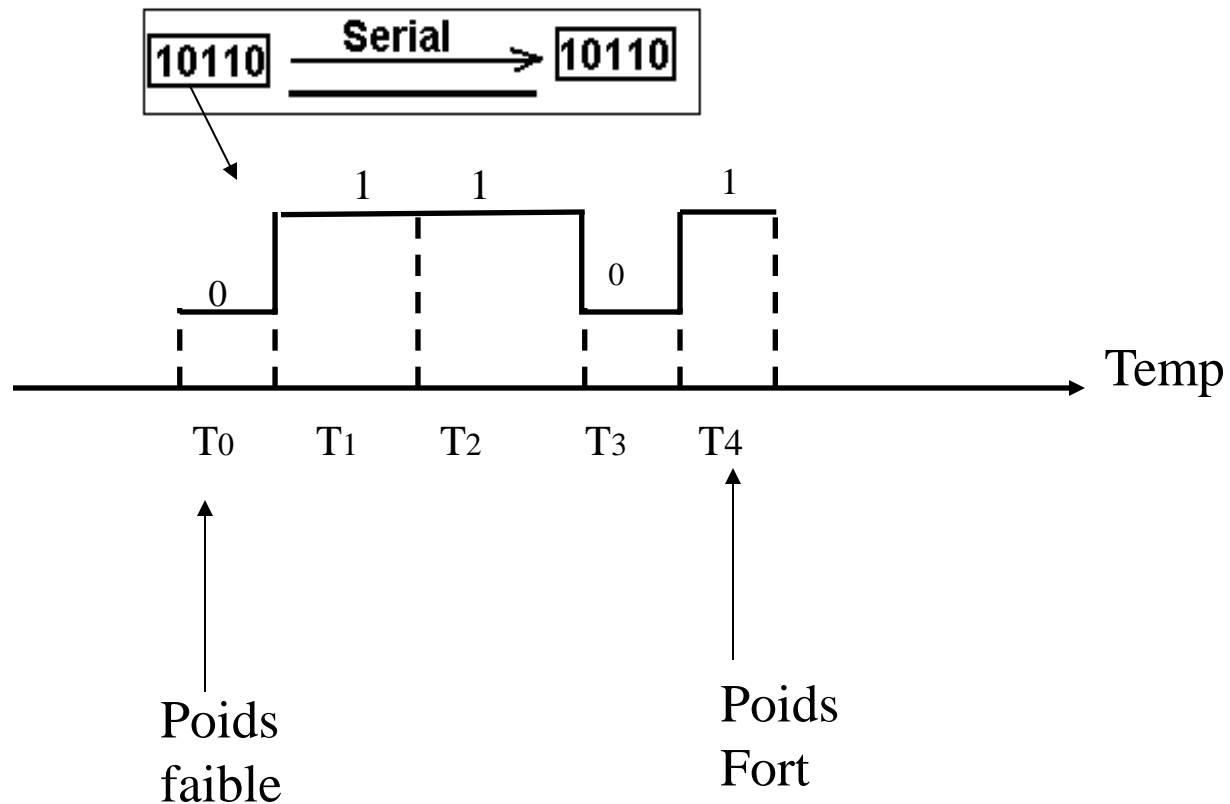
## Registres et Compteurs

# Registres

- Plusieurs bascules peuvent être combinés pour former un registre de données
- Le registre a décalage permet de transférer les données un bit a la fois
- aussi de transférer les données en parallèle.
  - plusieurs bits sont transférés en même temps
- sont des composant de base de plusieurs ordinateur.

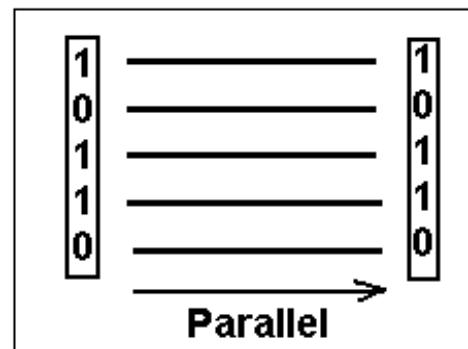
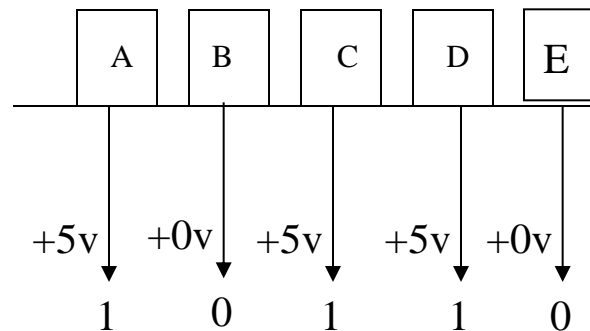
# Parallèle versus Série

- Une communication en série transfère une donnée comme une séquence bit un après l'autre au travers d'une ligne de données.



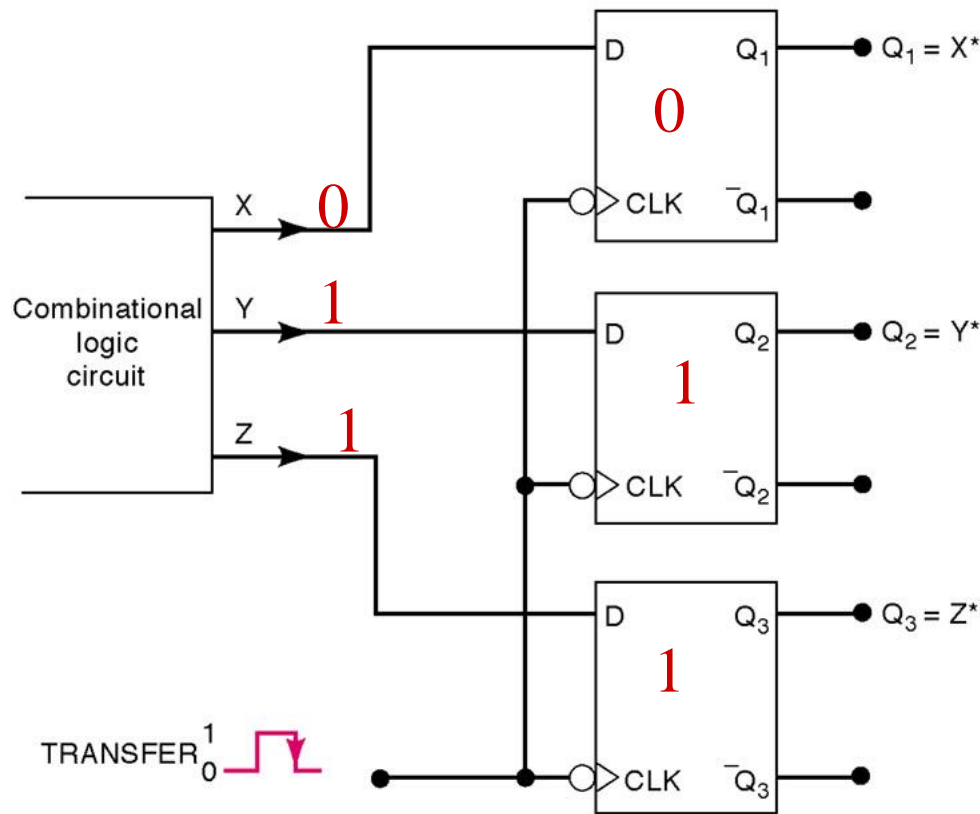
# Parallèle versus Série

- **Communications Parallèle** transfère une donnée comme une séquence bit au travers de plusieurs lignes de données en même temps



# Transfert Parallèle de Données

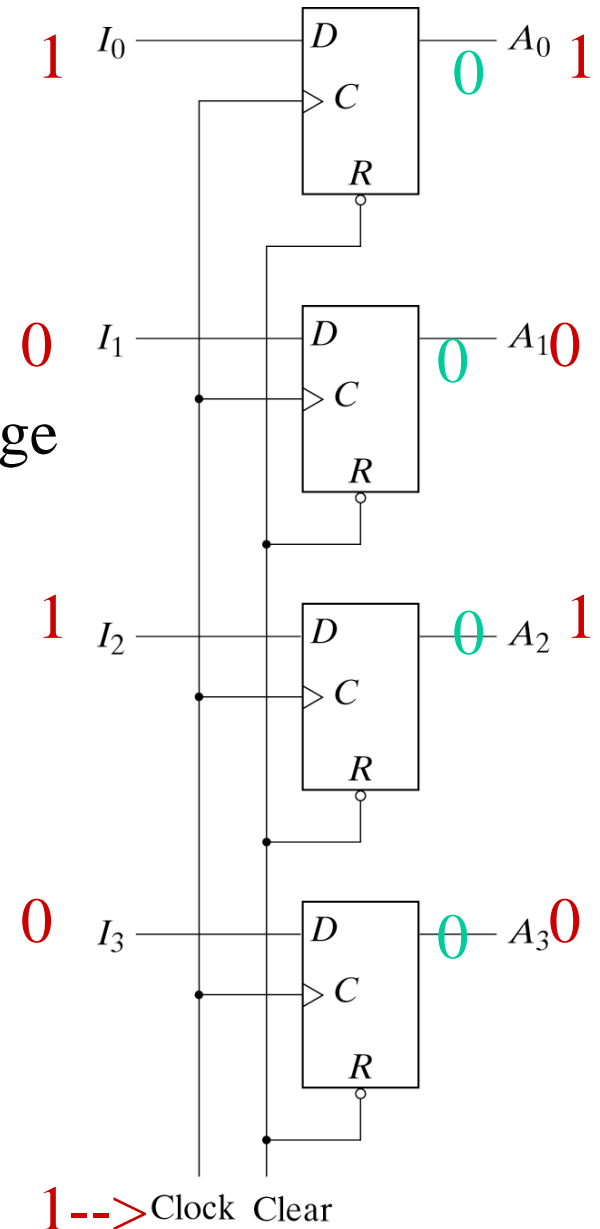
- Dans cet exemple la bascule D stocke les sorties du circuit combinatoire qui possède trois sorties.
- Plusieurs flip flops peuvent stocker une suite de données



\*After occurrence of NGT

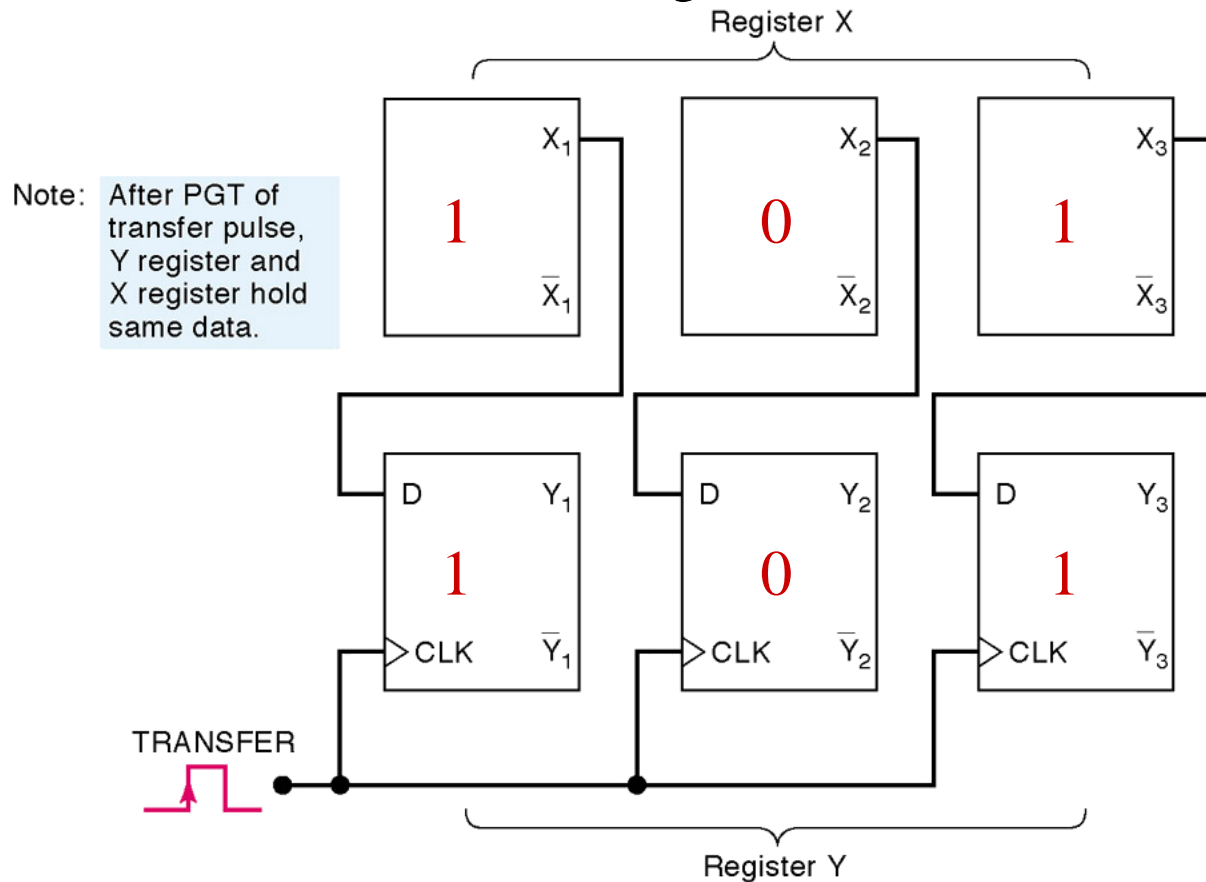
# Registre avec Chargement Parallèle

- Registre: Groupe de bascules
- Ex: bascule D
- contient 4 bits de données
- Charge en Parallèle sur la Transition hologe
- Asynchrone Clear (Reset)

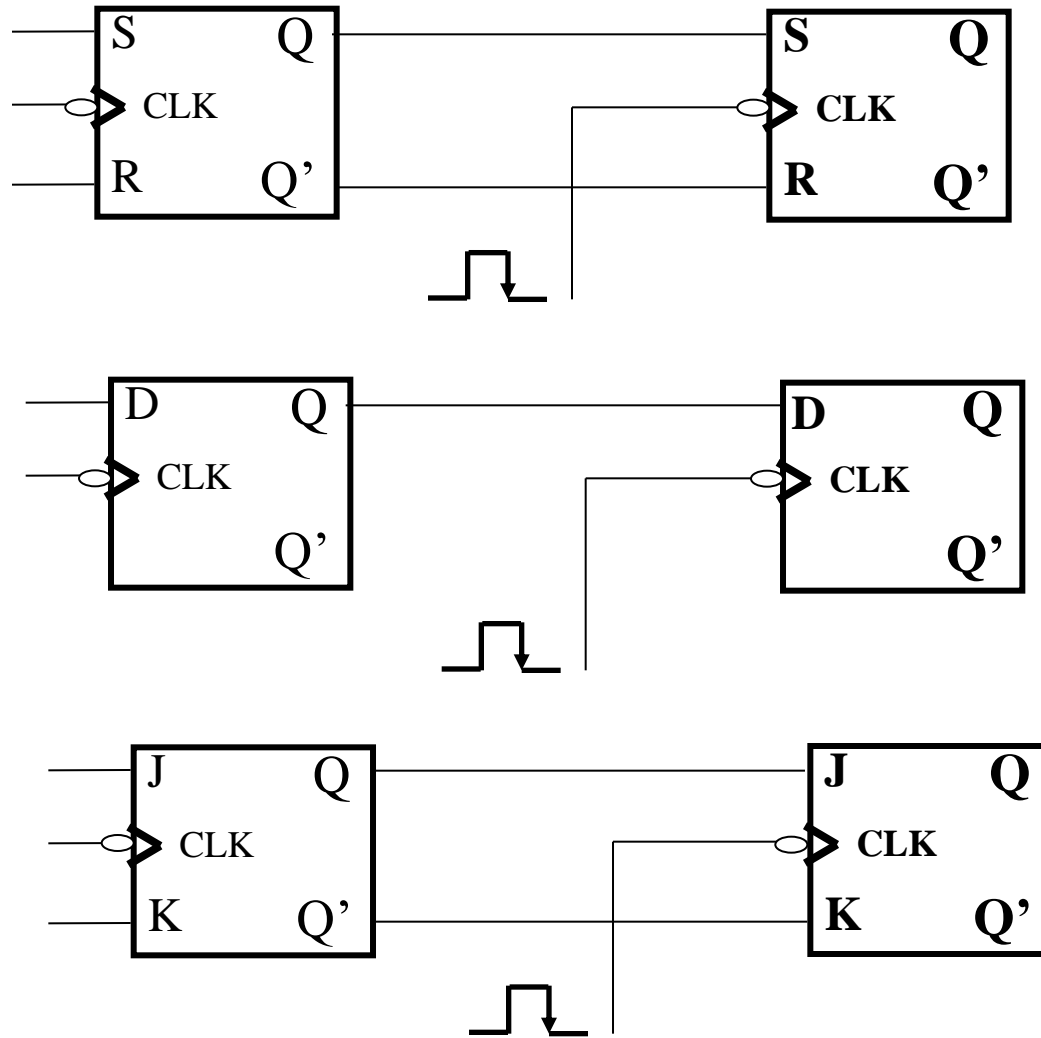


# Transfert de Données en Parallèle

- ° Toutes les données sont transférées sur un front montant de l'horloge
- ° Données stockées dans le registre Y



# Transfert Série



# Registres a décalage

- Une cascade de bascules
- Les bits sont décalés sur un front montant ( dans cet exemple)
- Entrées en série et sorties en série

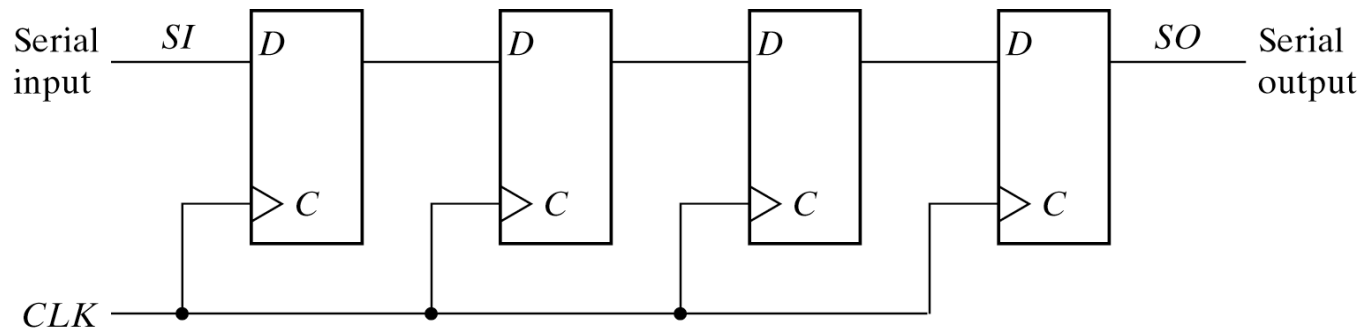
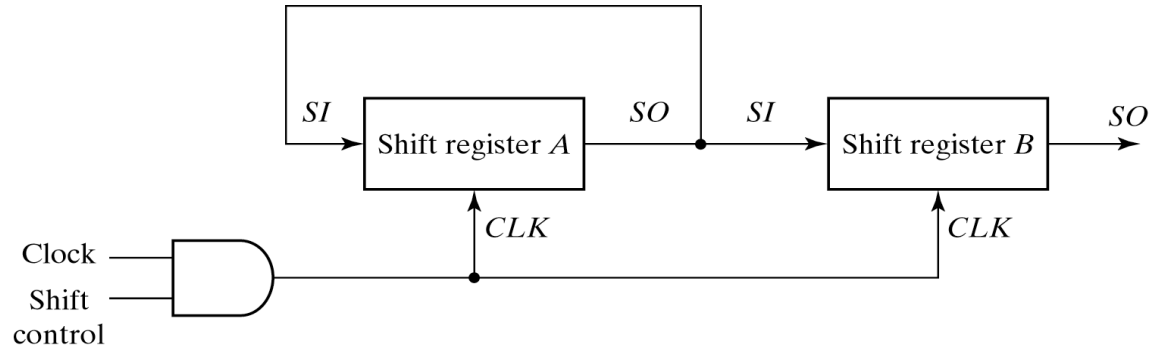


Fig. 6-3 4-Bit Shift Register

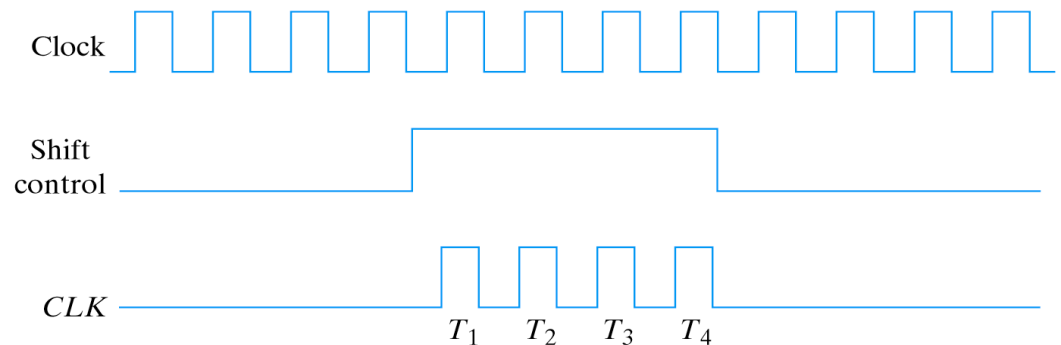
# Transfert Série

- Les données sont transférées un bit à la fois
- Noter la boucle sur le registre A.



(a) Block diagram

| Time | A    | B    |
|------|------|------|
| T0   | 1011 | 0011 |
| T1   |      |      |
| T2   |      |      |
| T3   |      |      |
| T4   |      |      |

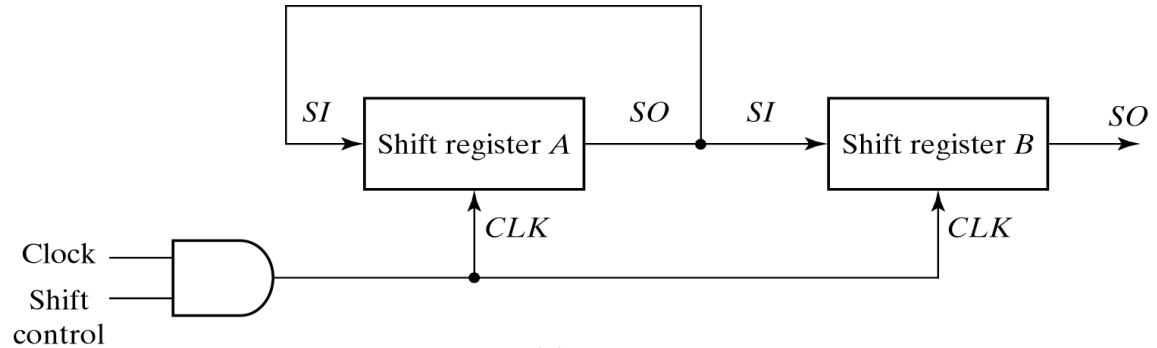


(b) Timing diagram

Fig. 6-4 Serial Transfer from Register A to register B

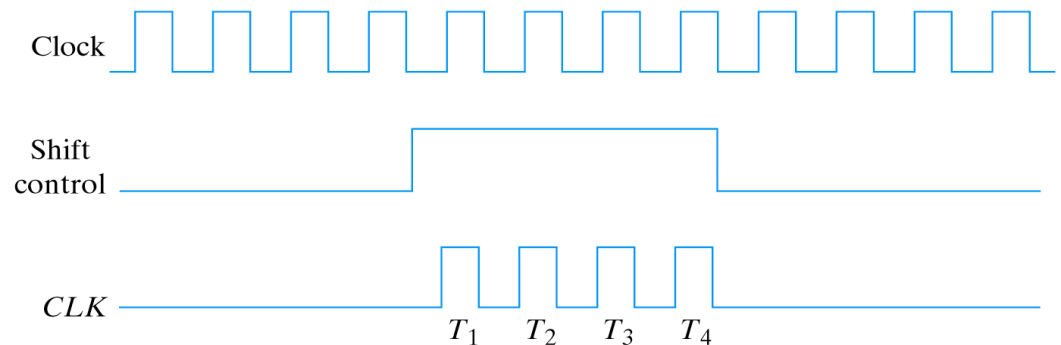
# Transfert Série

- Les données sont transférées un bit à la fois
- Noter la boucle sur le registre A.



(a) Block diagram

| Time | A    | B    |
|------|------|------|
| T0   | 1011 | 0011 |
| T1   | 1101 | 1001 |
| T2   | 1110 | 1100 |
| T3   | 0111 | 0110 |
| T4   | 1011 | 1011 |

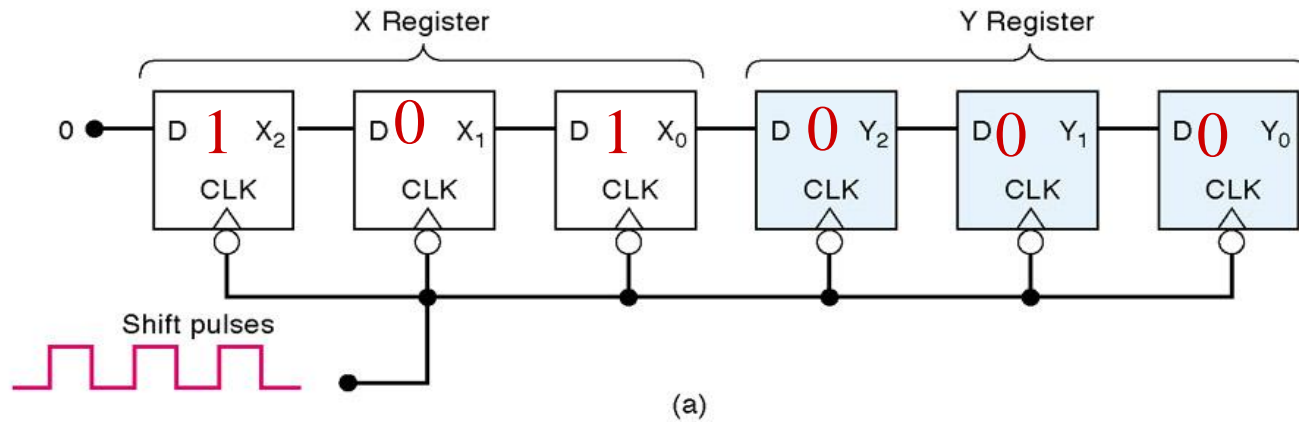


(b) Timing diagram

Fig. 6-4 Serial Transfer from Register A to register B

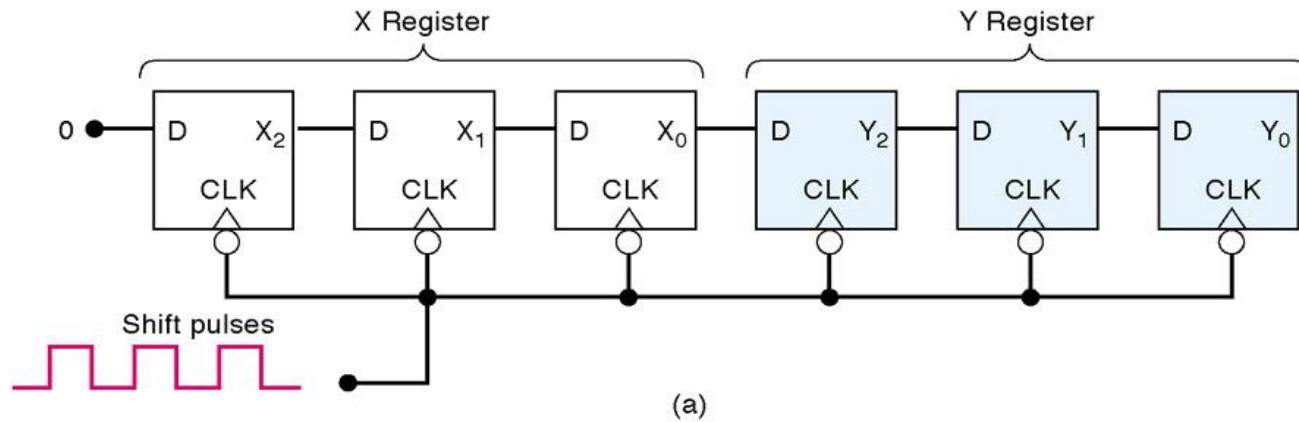
# Transfert de Données en Série

## Transfert du registre X au registre Y



# Transfert de Données en Série

## Transfert du registre X au registre Y



| X <sub>2</sub> | X <sub>1</sub> | X <sub>0</sub> | Y <sub>2</sub> | Y <sub>1</sub> | Y <sub>0</sub> |                         |
|----------------|----------------|----------------|----------------|----------------|----------------|-------------------------|
| 1              | 0              | 1              | 0              | 0              | 0              | ← Before pulses applied |
| 0              | 1              | 0              | 1              | 0              | 0              | ← After first pulse     |
| 0              | 0              | 1              | 0              | 1              | 0              | ← After second pulse    |
| 0              | 0              | 0              | 1              | 0              | 1              | ← After third pulse     |

(b)

# Additionneur Complet

**Table de vérité**

| <b>x</b> | <b>y</b> | <b>z</b> | <b>C<sub>o</sub></b> | <b>S</b> |
|----------|----------|----------|----------------------|----------|
| 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        |



# Additionneur Complet- Diagramme de K

|   |   | yz |    | y  |    |
|---|---|----|----|----|----|
|   |   | 00 | 01 | 11 | 10 |
| x | 0 |    | 1  |    | 1  |
|   | 1 | 1  |    | 1  |    |

z

$$S = x'y'z + x'yz' + xy'z' + xyz$$

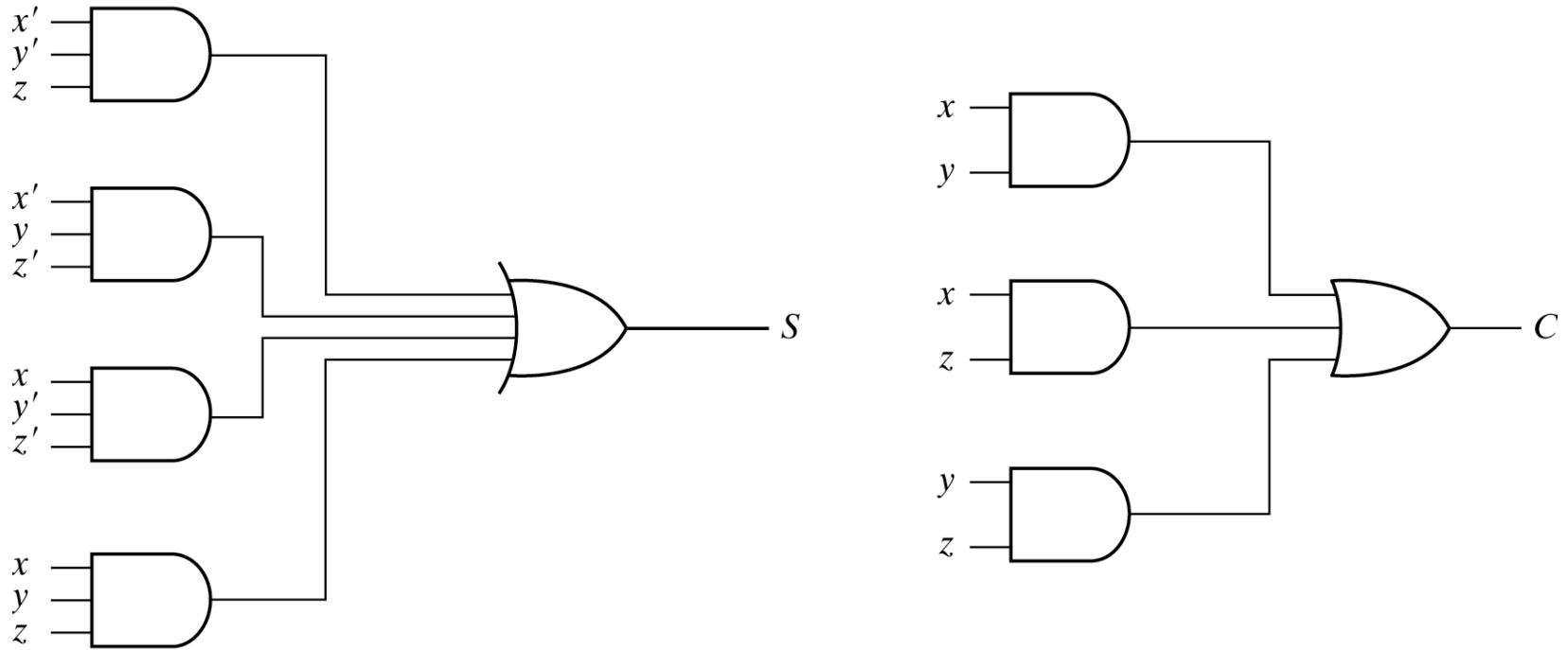
|   |   | yz |    | y  |    |
|---|---|----|----|----|----|
|   |   | 00 | 01 | 11 | 10 |
| x | 0 |    |    | 1  |    |
|   | 1 |    | 1  | 1  | 1  |

z

$$\begin{aligned} S &= xy + xz + yz \\ &= xy + xy'z + x'yz \end{aligned}$$

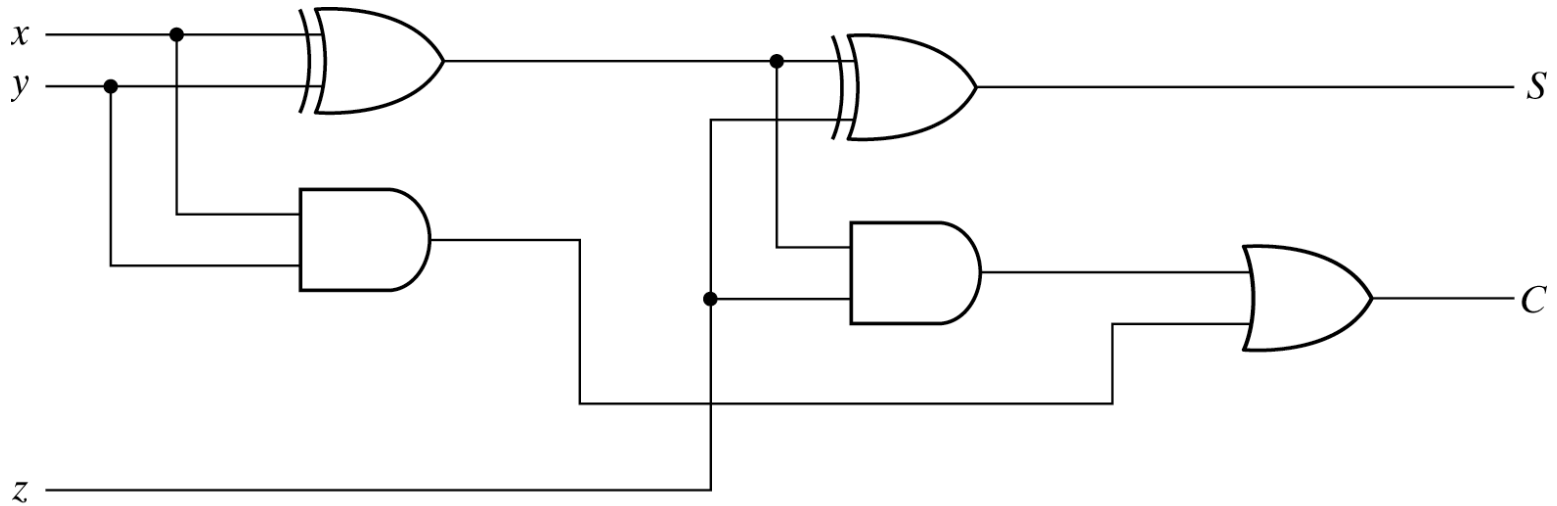
**Diagramme pour l'additionneur complet**

# Additionneur Complet- réalisation: voir d'autres chapitre 2)



Réalisation somme de produits

# Additionneur Complet- réalisation: voir d'autres chapitre 2)



Réalisation avec deux demi- additionneur et une porte OU

# Additionneur en série avec bascule D

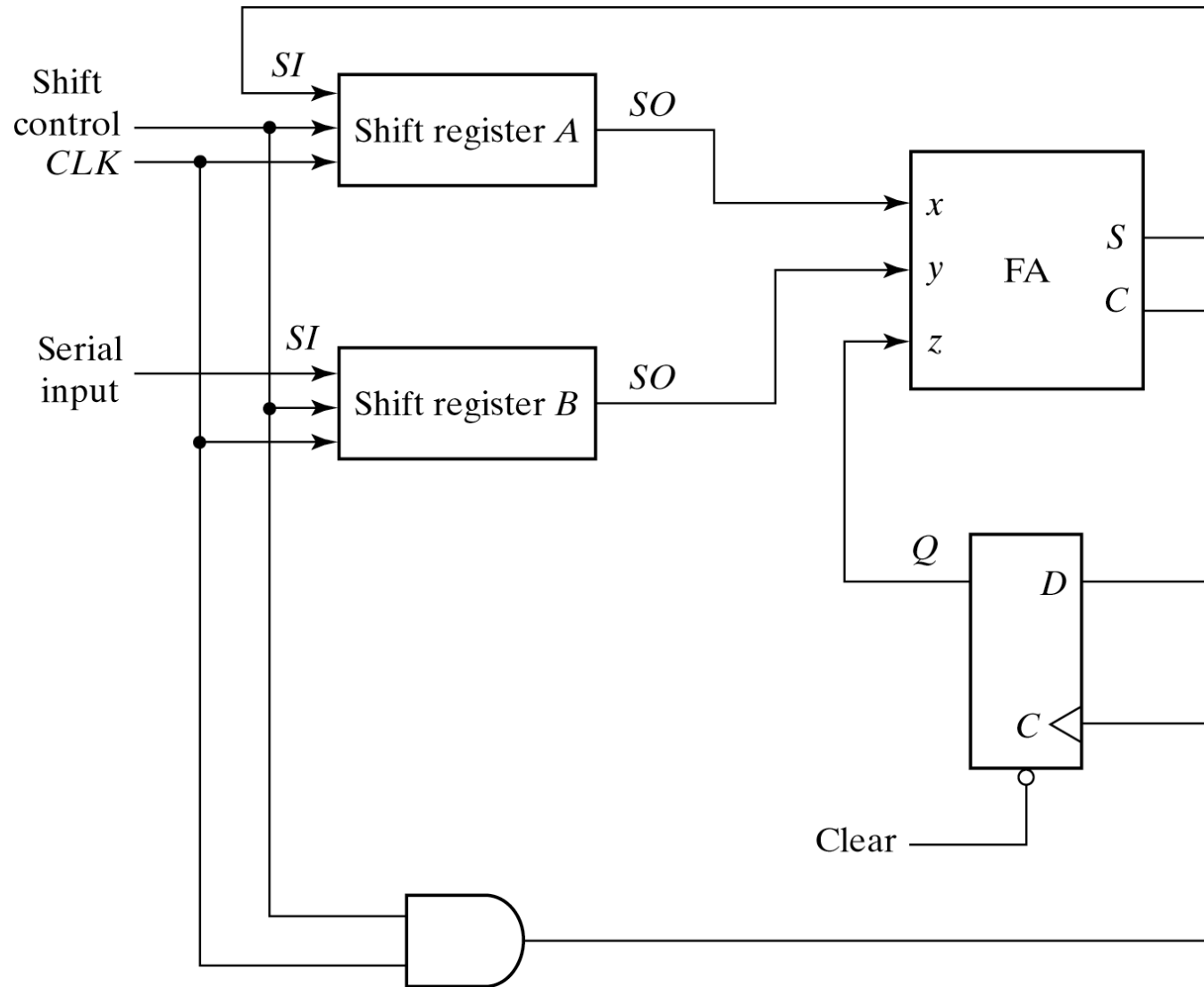


Fig. 6-5 Serial Adder

# Tables de Transition: exemple avec JK

## Table de transition pour J-K

- on indique l'état présent et suivant pour les sorties  $Q$ .
- On doit aussi déterminer les deux entrées J et K
- Le "X" indique l'entrée "indéterminées" (peut être l'entrée '1' ou '0').

| J | K | $Q_{n+1}$ | Function   |
|---|---|-----------|------------|
| 0 | 0 | $Q_n$     | inchangée  |
| 0 | 1 | 0         | RESET      |
| 1 | 0 | 1         | SET        |
| 1 | 1 | $Q'_n$    | complément |

| J-K FLIP-FLOP |      |              |
|---------------|------|--------------|
| Present       | Next | Input<br>J K |
| 0             | → 0  | 0 X          |
| 0             | → 1  | 1 X          |
| 1             | → 0  | X 1          |
| 1             | → 1  | X 0          |

# Additionneur en série avec bascule JK

- $J_Q = x \cdot y,$
- $K_Q = x' \cdot y' = (x+y)',$
- $S = x \oplus y \oplus Q$

| Present State | Inputs |   | Next State | Output | Flip_Flop Inputs |       |
|---------------|--------|---|------------|--------|------------------|-------|
| Q             | x      | y | Q          | S      | $J_Q$            | $K_Q$ |
| 0             | 0      | 0 | 0          | 0      | 0                | x     |
| 0             | 0      | 1 | 0          | 1      | 0                | x     |
| 0             | 1      | 0 | 0          | 1      | 0                | x     |
| 0             | 1      | 1 | 1          | 0      | 1                | x     |
| 1             | 0      | 0 | 0          | 1      | x                | 1     |
| 1             | 0      | 1 | 1          | 0      | x                | 0     |
| 1             | 1      | 0 | 1          | 0      | x                | 0     |
| 1             | 1      | 1 | 1          | 1      | x                | 0     |

# Additionneur en série avec bascule JK

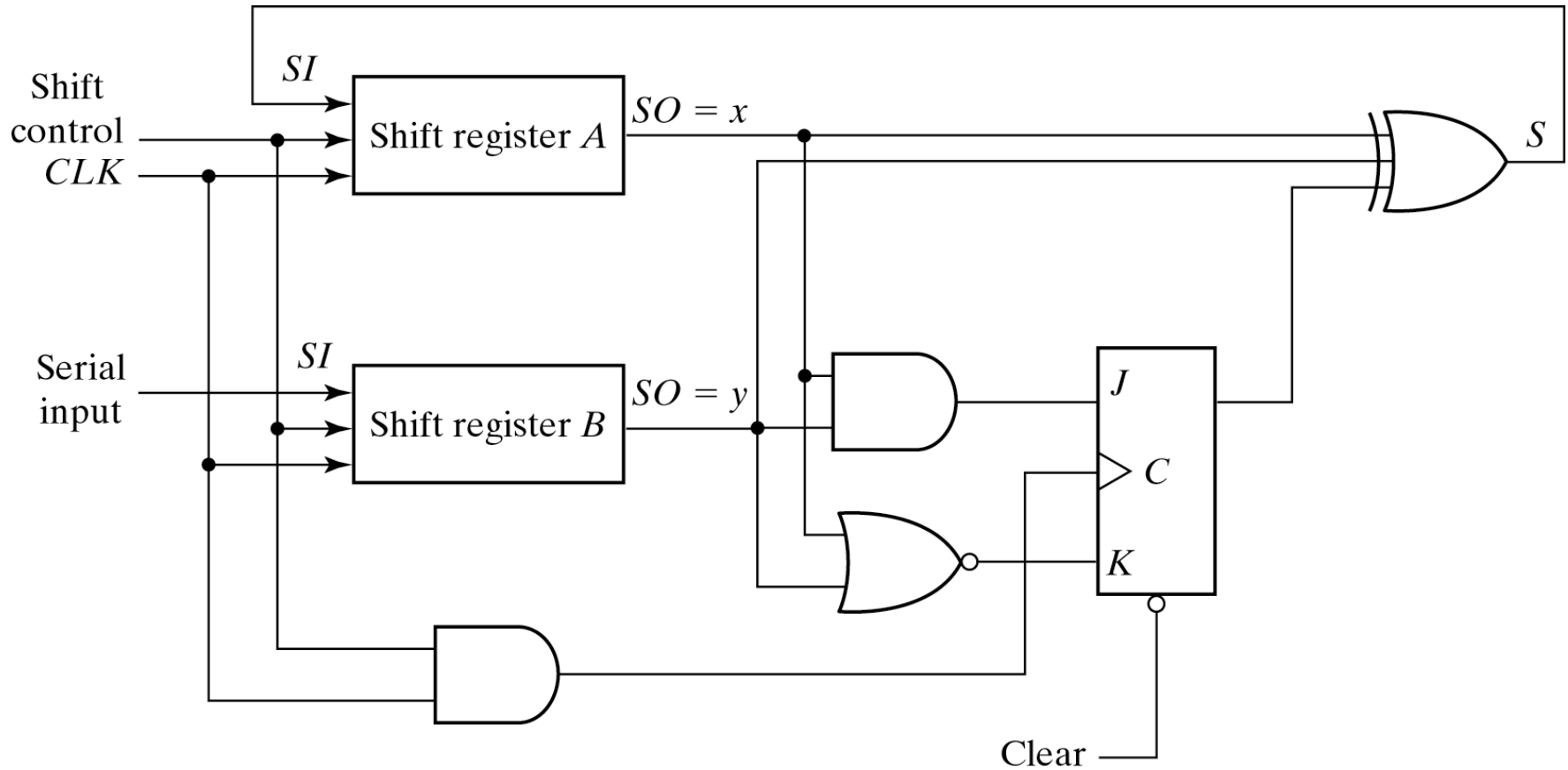


Fig. 6-6 Second form of Serial Adder

# Compteurs

- Compteur: un registre qui passe par une série d'états prédéfinies
- deux types principaux de compteur
  - **Compteurs asynchrones**: en anglais « Ripple counter »
  - **Compteurs synchrones**
- *Compteurs asynchrones*
  - Les sorties de la bascule servent d'horloge pour les bascules qui leurs sont connectées
- *Compteurs synchrones*
  - Toutes les bascules sont synchronisées avec la même horloge et en même temps
- Les compteurs synchrones sont les plus utilisées

# Compteur binaire asynchrone (Ripple)

- **Le signal reset: mise a 0 des bascules**
- **Count signal fait basculer (complément)  $A_0$**
- **La bascules de niveaux inférieur fournie le signale pour l bascule adjacente**
- **Pas toutes les bascules qui change d'états en même temps**
  - **La bascules de niveau inférieur change en premier.**

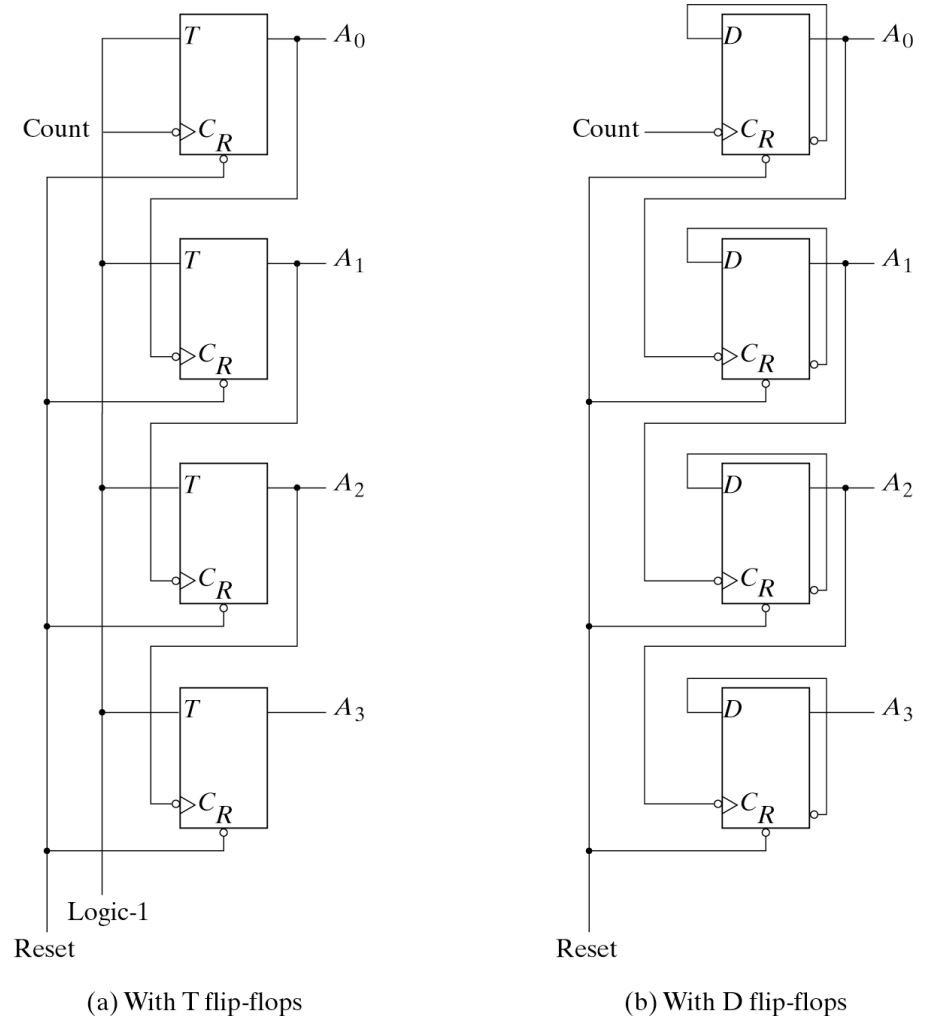
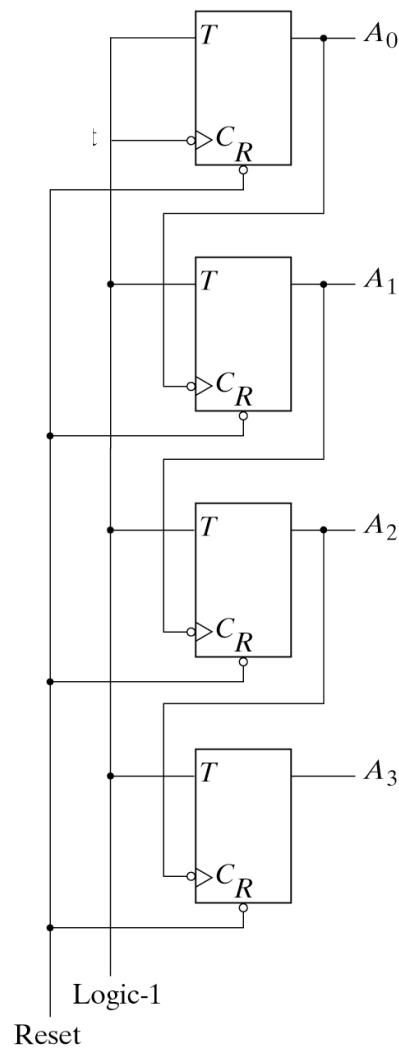
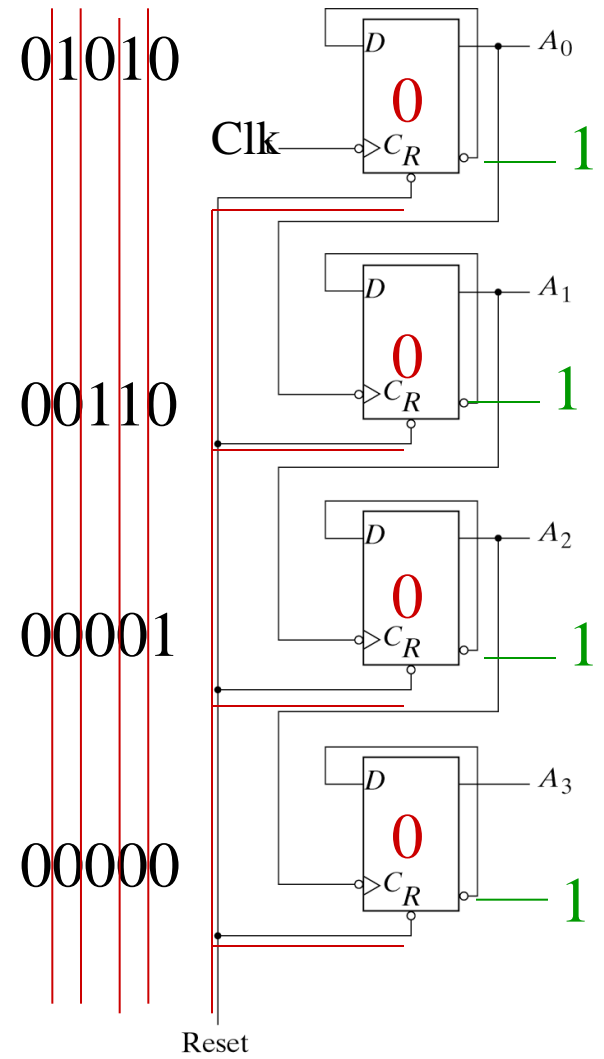


Fig. 6-8 4-Bit Binary Ripple Counter

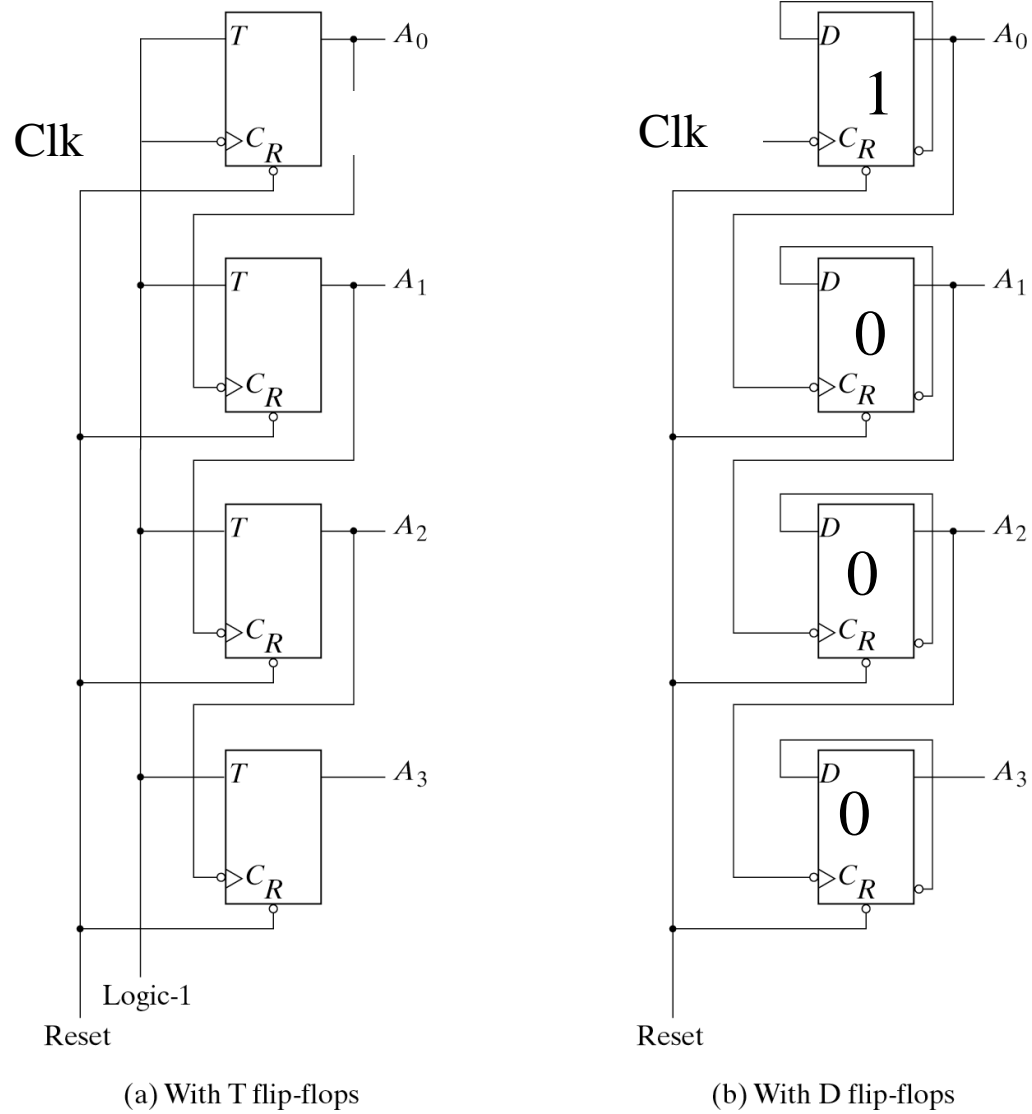


(a) With T flip-flops



(b) With D flip-flops

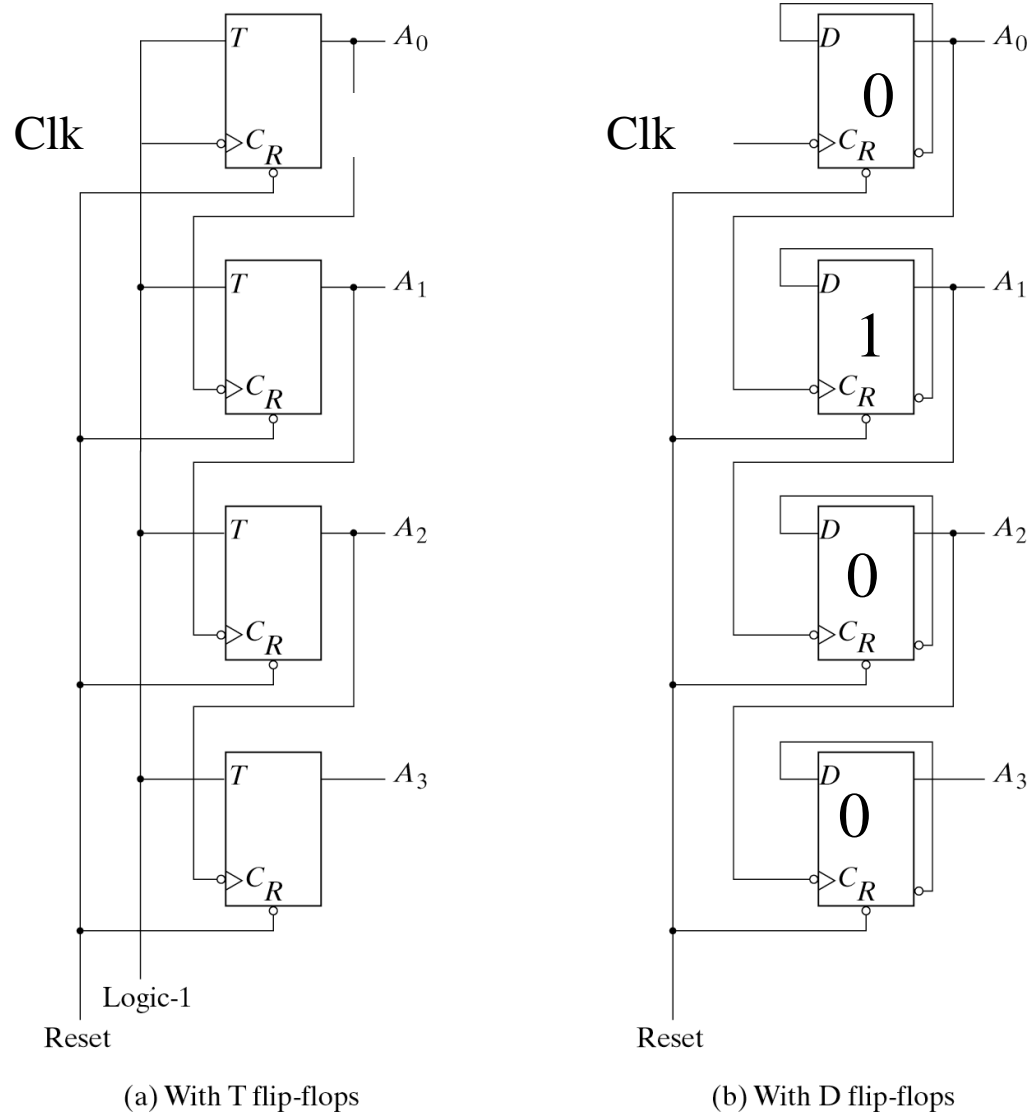
01010  
 00110  
 00001  
 00000



(a) With T flip-flops

(b) With D flip-flops

Fig. 6-4  
ITI1500 A. Karmouch



(a) With T flip-flops

(b) With D flip-flops

Fig. 6-4  
ITI1500 A. Karmouch

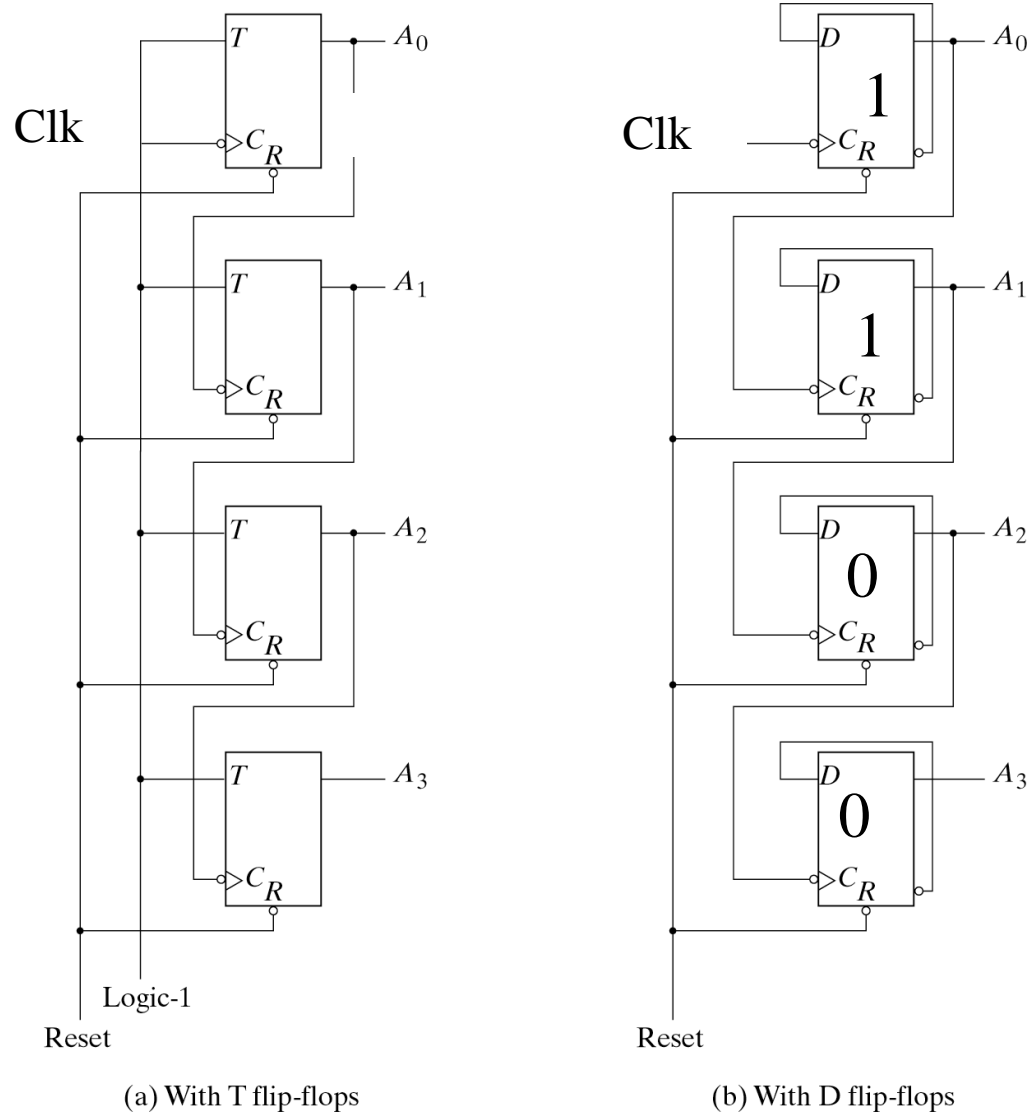
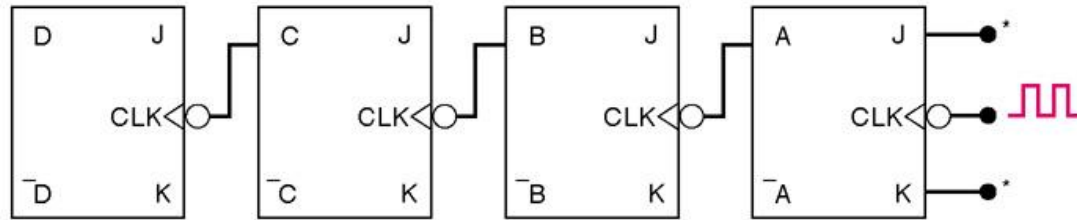
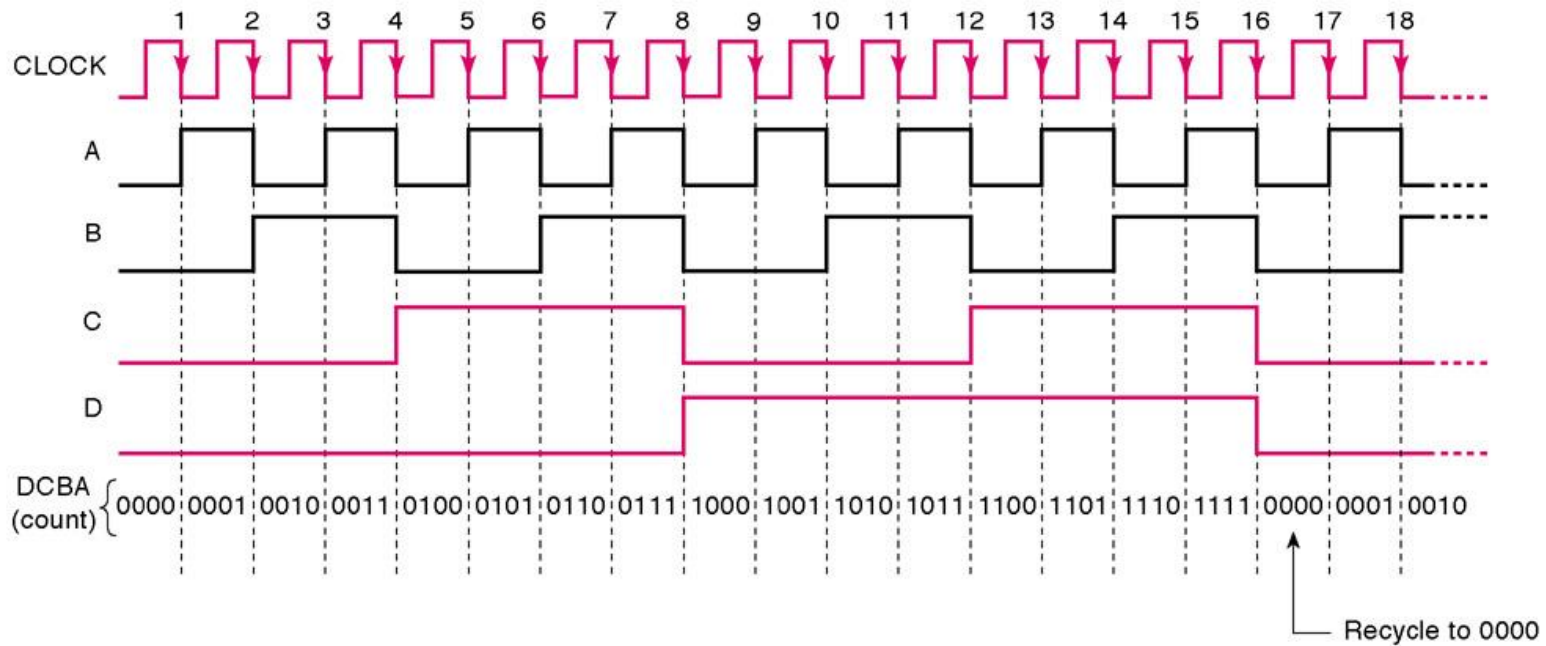


Fig. 6-4  
ITI1500 A. Karmouch

# Exemples de compteur Asynchrone



\*All J and K inputs assumed to be 1.



# Compteurs synchrones

- Les compteurs synchrones
  - Toutes les bascules reçoivent un signal en même temps
  - Toutes les bascules changent en même temps
- dans JK flip Flop
  - Si  $J=K=0$ , bascule garde sa valeur
  - Si  $J=K=1$ , bascules complémente
- Valeur incrémentée sur front montant

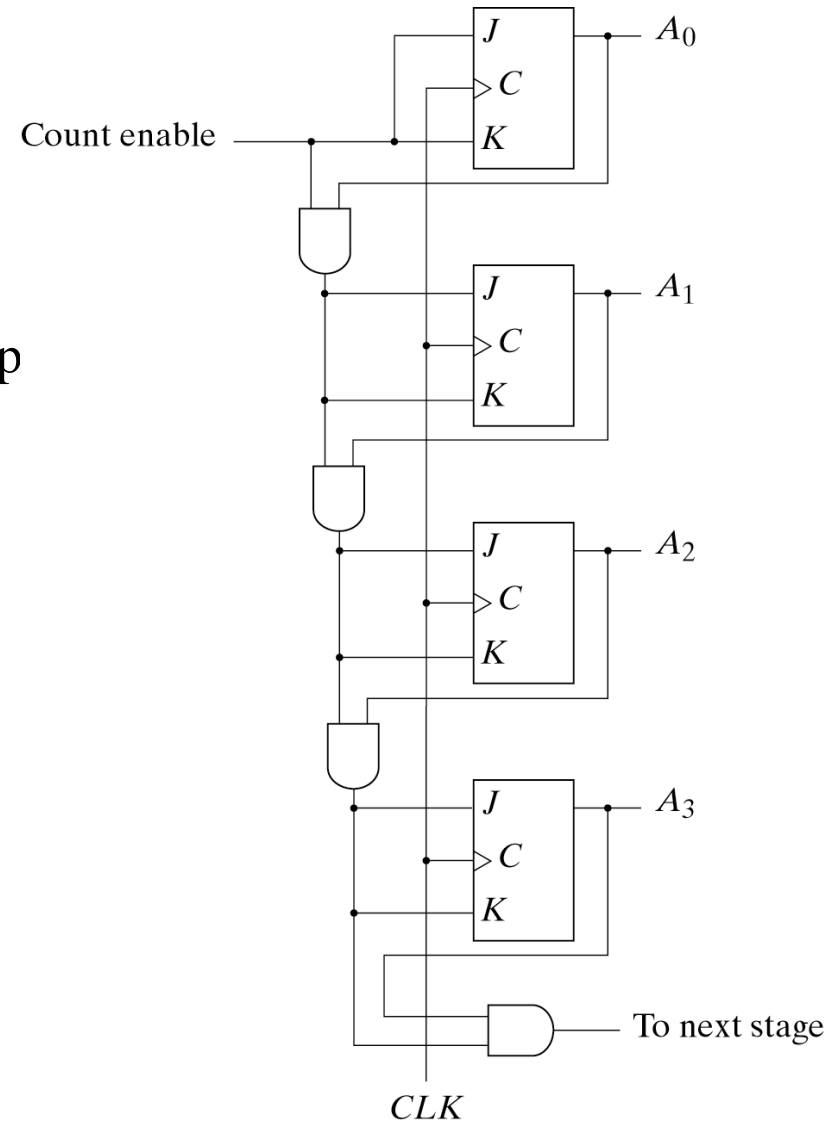
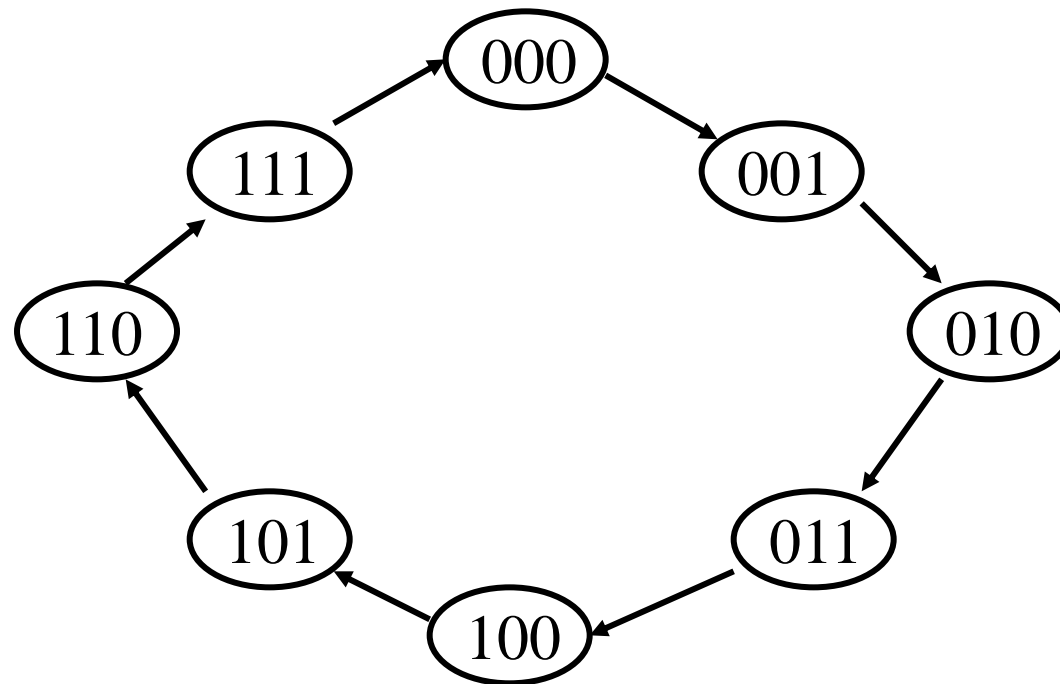


Fig. 6-12 4-Bit Synchronous Binary Counter

# Compteurs binaires

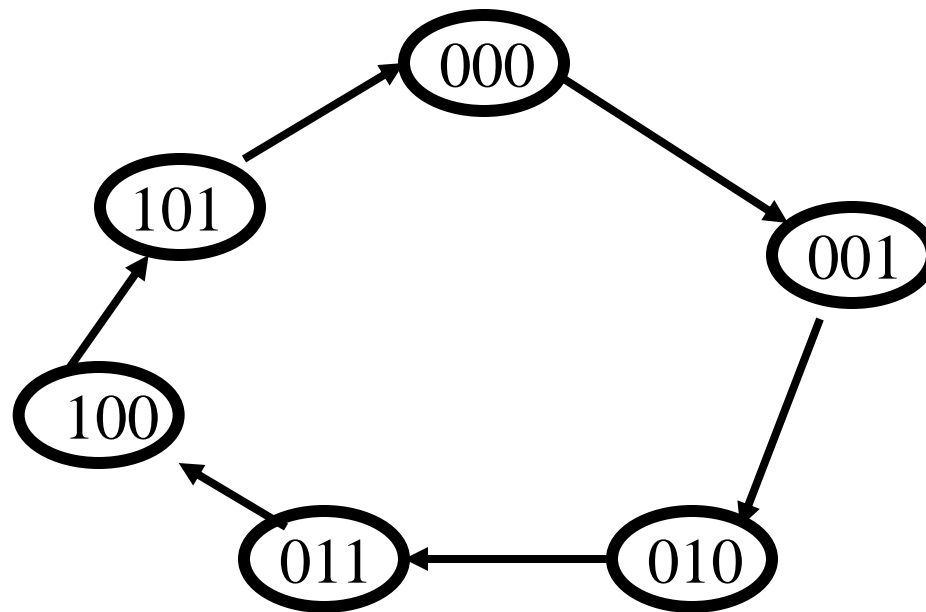
- Compteurs produisent un nombre fixe de séquences binaires
- Compteurs sont conçus pour recommencer le comptage: recyclage

Exemple:



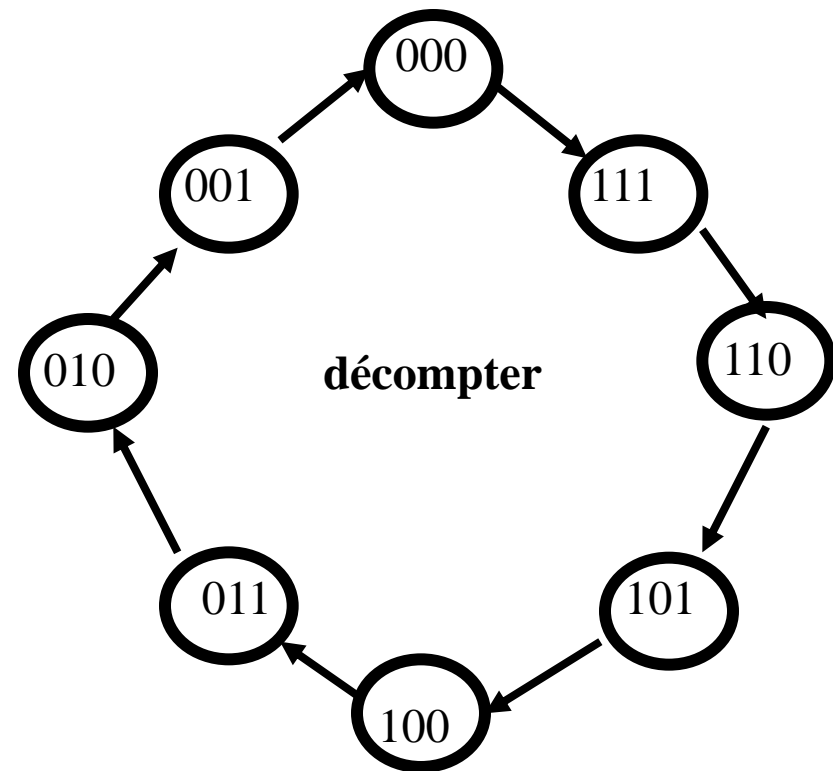
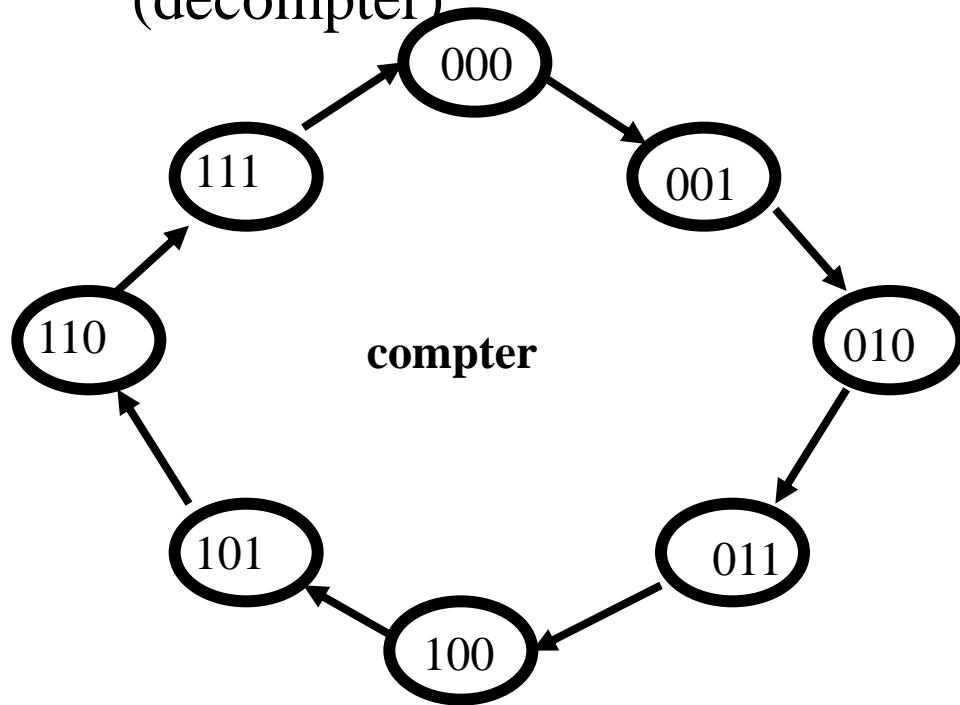
# Compteurs Binaire: Modulo

- Le nombre d'états que traverse un compteur est appelée Modulo, ou MOD.
  - Par exemple, un compteur modulo-6 possède 6 états .



# Compteur binaire

- Un compteur séquentiel représente un comptage naturel binaire.
- Le comptage peut être croissant (compter) ou décroissant (décompter)



# Diagramme d'États

- Les états du compteur sont séquentiels. Chaque état suit l'autre en séquence
- Diagramme d'états est utilisé pour représenter ces séquences.

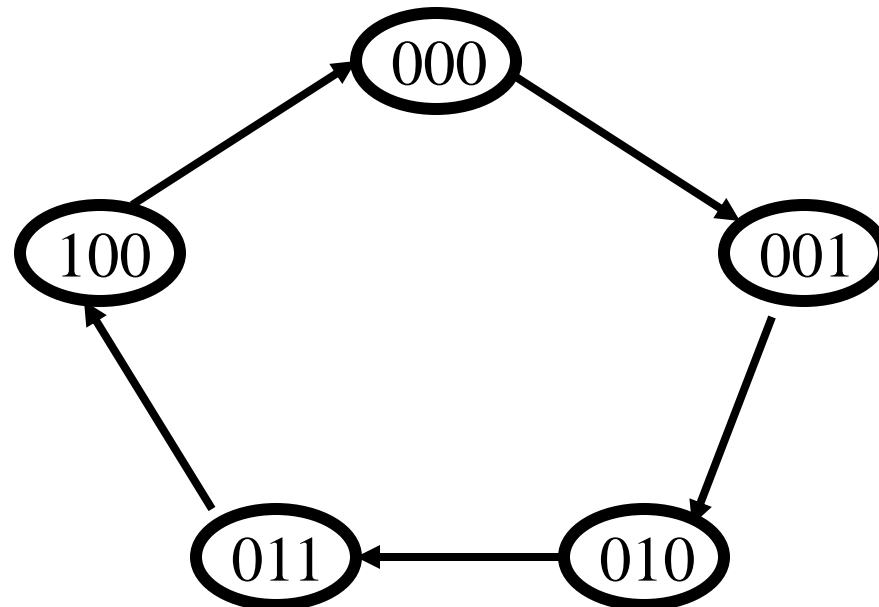


Diagramme d'états

# Table des états

- la table des états (aussi dite de transitions) est un moyen de représenter les séquences d'états
- Comme pour le diagramme d'états la table représente l'état suivant basé sur l'état présent.

Table d'Etat

| Present |       |       | Suivant |       |       |
|---------|-------|-------|---------|-------|-------|
| $Q_c$   | $Q_b$ | $Q_a$ | $Q_c$   | $Q_b$ | $Q_a$ |
| 0       | 0     | 0     | 0       | 0     | 1     |
| 0       | 0     | 1     | 0       | 1     | 0     |
| 0       | 1     | 0     | 0       | 1     | 1     |
| 0       | 1     | 1     | 1       | 0     | 0     |
| 1       | 0     | 0     | 0       | 0     | 0     |

# Les séquences du compteur

## → *Compteur complet de séquences*

Toutes les séquences du compteur binaire sont traversées. Le modulo est le même que le maximum du modulo.

## → *Compteur à séquences tronquées*

C'est le cas où le modulo du compteur est moins que le modulo maximum. Le compteur passe par moins d'états qu'il est possible de faire avec le nombre de bascules utilisées.

# Procédure de conception d'un compteur synchrone

- 1- a partir d'une table d'états donnée, obtenir (i) le diagramme d'états (ii) table de transition pour le compteur
- 2- Fournir les entrées pour chaque bascule en utilisant les sorties des autres bascules.
- 3- Utiliser les portes logiques externes pour détecter les séquences spécifique pour chaque bascules de manière obtenir la séquences suivante des entrées.
- 4- Utiliser le diagramme de K pour simplifier les sorties disponibles et créer les entrées.
- 5- Réaliser le circuit

# Tables de Transitions

---

- On doit établir comment les bascules réagissent aux différentes données en entrée, en fonction de leur **état présent**.

# Construction de la Table de Transition

- La construction de la table de transition est obtenu en utilisant la table de vérité (ou table de fonction) des bascules.
- Quand la table de vérité indique “Q”, “Q’” ou “pas de Changement”, nous indiquons les valeur binaires.

# Table de transition : Exemple avec Bascules D

| D-FLIP-FLOP |      |       |
|-------------|------|-------|
| Present     | Next | Input |
| 0           | → 0  | 0     |
| 0           | → 1  | 1     |
| 1           | → 0  | 0     |
| 1           | → 1  | 1     |

- On indique pour les sorties l'état présent et l'état suivant et l'entrée de D nécessaire pour obtenir l'état suivant

# Tables de Transition: exemple avec JK

## Table de transition pour J-K

- on indique l'état présent et suivant pour les sorties Q.
- On doit aussi déterminer les deux entrées J et K
- Le "X" indique l'entrée "indéterminées" (peut être l'entrée '1' ou '0').

| J-K FLIP-FLOP |      |              |
|---------------|------|--------------|
| Present       | Next | Input<br>J K |
| 0             | → 0  | 0 X          |
| 0             | → 1  | 1 X          |
| 1             | → 0  | X 1          |
| 1             | → 1  | X 0          |

# Table de transition: JK (suite)

| J-K FLIP-FLOP |      |              |
|---------------|------|--------------|
| Present       | Next | Input<br>J K |
| 0             | → 0  | 0 X          |
| 0             | → 1  | 1 X          |
| 1             | → 0  | X 1          |
| 1             | → 1  | X 0          |

## États

Pas de changement ou or Reset

Complément ou Set

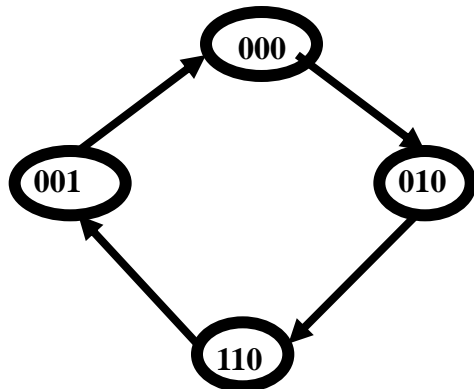
complément ou Reset

Pas de changement ou or Set

→ chaque sortie de la Bascules J-K est le résultat de 2 états possibles

# Construction de Karnaugh avec les bascules D

- Pour construire le diagramme, nous avons besoins:
  - 1- du diagramme d'états pour les séquences en sorties
  - 2- la table d'états pour les séquences en sortie.
  - 3- La table de transition de la bascules que nous voulons utiliser



| Present       | Next          |
|---------------|---------------|
| $Q_C Q_B Q_A$ | $Q_C Q_B Q_A$ |
| 0 0 0         | 0 1 0         |
| 0 1 0         | 1 1 0         |
| 1 1 0         | 0 0 1         |
| 0 0 1         | 0 0 0         |

| D-FLIP-FLOP |      |       |
|-------------|------|-------|
| Present     | Next | Input |
| 0           | → 0  | 0     |
| 0           | → 1  | 1     |
| 1           | → 0  | 0     |
| 1           | → 1  | 1     |

# Construction de Karnaugh avec les bascules D :

## Etape 1

- Établir la table des états et le diagramme d'états pour les séquences du compteur

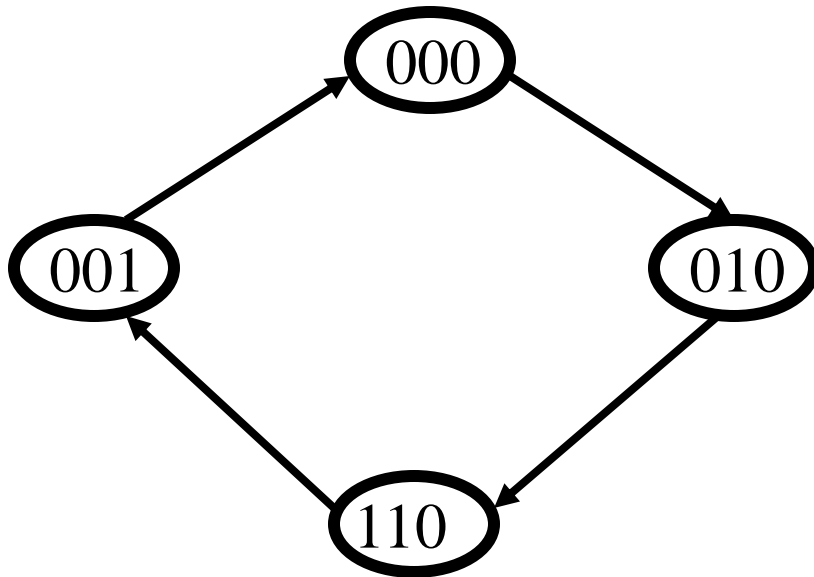


Diagramme d'états

| Present       | Next          |
|---------------|---------------|
| $Q_C Q_B Q_A$ | $Q_C Q_B Q_A$ |
| 0 0 0         | 0 1 0         |
| 0 1 0         | 1 1 0         |
| 1 1 0         | 0 0 1         |
| 0 0 1         | 0 0 0         |

Table d'états

# Construction de Karnaugh avec les bascules D :

## étape 2

- choisir la bascules et sa table de transition. **Pour cet exemple on utilise la bascule D**

| D-FLIP-FLOP |      |       |
|-------------|------|-------|
| Present     | Next | Input |
| 0           | → 0  | 0     |
| 0           | → 1  | 1     |
| 1           | → 0  | 0     |
| 1           | → 1  | 1     |

Transition Table

# Construction de Karnaugh avec les bascules **D**:

## étape 3

- **Construire la table d'états pour les bascules**

| D-FLIP-FLOP |      |       |
|-------------|------|-------|
| Present     | Next | Input |
| 0           | → 0  | 0     |
| 0           | → 1  | 1     |
| 1           | → 0  | 0     |
| 1           | → 1  | 1     |

| Présent |       |       | Suivant |       |       | Entrée |       |       |
|---------|-------|-------|---------|-------|-------|--------|-------|-------|
| $Q_C$   | $Q_B$ | $Q_A$ | $Q_C$   | $Q_B$ | $Q_A$ | $D_C$  | $D_B$ | $D_A$ |
| 0       | 0     | 0     | 0       | 1     | 0     | 0      | 1     | 0     |
| 0       | 1     | 0     | 1       | 1     | 0     | 1      | 1     | 0     |
| 1       | 1     | 0     | 0       | 0     | 1     | 0      | 0     | 1     |
| 0       | 0     | 1     | 0       | 0     | 0     | 0      | 0     | 0     |

**D-FLIP-FLOP**

| Present | Next | Input |
|---------|------|-------|
| 0       | → 0  | 0     |
| 0       | → 1  | 1     |
| 1       | → 0  | 0     |
| 1       | → 1  | 1     |

| Present |       |       | Next  |       |       | Entree |       |       |
|---------|-------|-------|-------|-------|-------|--------|-------|-------|
| $Q_C$   | $Q_B$ | $Q_A$ | $Q_C$ | $Q_B$ | $Q_A$ | $D_C$  | $D_B$ | $D_A$ |
| 0       | 0     | 0     | 0     | 1     | 0     | 0      | 1     | 0     |
| 0       | 1     | 0     | 1     | 1     | 0     | 1      | 1     | 0     |
| 1       | 1     | 0     | 0     | 0     | 1     | 0      | 0     | 1     |
| 0       | 0     | 1     | 0     | 0     | 0     | 0      | 0     | 0     |

Animation

| Present |       |       | Next  |       |       | Input |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|
| $Q_C$   | $Q_B$ | $Q_A$ | $Q_C$ | $Q_B$ | $Q_A$ | $D_C$ | $D_B$ | $D_A$ |
| 0       | 0     | 0     | 0     | 1     | 0     | 0     | 1     | 0     |
| 0       | 1     | 0     | 1     | 1     | 0     | 1     | 1     | 0     |
| 1       | 1     | 0     | 0     | 0     | 1     | 0     | 0     | 1     |
| 0       | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     |

000

010

|           |       |   |  |
|-----------|-------|---|--|
|           | $Q_A$ |   |  |
| $Q_C Q_B$ | 0     | 1 |  |
| 00        | 0     | 0 |  |
| 01        | 1     | X |  |
| 11        | 0     | X |  |
| 10        | X     | X |  |

$D_C$

|           |       |   |  |
|-----------|-------|---|--|
|           | $Q_A$ |   |  |
| $Q_C Q_B$ | 0     | 1 |  |
| 00        | 1     | 0 |  |
| 01        | 1     | X |  |
| 11        | 0     | X |  |
| 10        | X     | X |  |

$D_B$

|           |       |   |  |
|-----------|-------|---|--|
|           | $Q_A$ |   |  |
| $Q_C Q_B$ | 0     | 1 |  |
| 00        | 0     | 0 |  |
| 01        | 0     | X |  |
| 11        | 1     | X |  |
| 10        | X     | X |  |

$D_A$

A  
N  
I  
M  
A  
T  
E  
D

Faire un Karnaugh pour chaque sortie:

| Présent |       |       | Suivant |       |       | Entrée |       |       |
|---------|-------|-------|---------|-------|-------|--------|-------|-------|
| $Q_C$   | $Q_B$ | $Q_A$ | $Q_C$   | $Q_B$ | $Q_A$ | $D_C$  | $D_B$ | $D_A$ |
| 0       | 0     | 0     | 0       | 1     | 0     | 0      | 1     | 0     |
| 0       | 1     | 0     | 1       | 1     | 0     | 1      | 1     | 0     |
| 1       | 1     | 0     | 0       | 0     | 1     | 0      | 0     | 1     |
| 0       | 0     | 1     | 0       | 0     | 0     | 0      | 0     | 0     |

| $Q_C \backslash Q_B \backslash Q_A$ | $Q_A$ |   |
|-------------------------------------|-------|---|
|                                     | 0     | 1 |
| $Q_C \backslash Q_B$                |       |   |
| 00                                  | 0     | 0 |
| 01                                  | 1     | X |
| 11                                  | 0     | X |
| 10                                  | X     | X |

**$D_C$**

| $Q_C \backslash Q_B \backslash Q_A$ | $Q_A$ |   |
|-------------------------------------|-------|---|
|                                     | 0     | 1 |
| $Q_C \backslash Q_B$                |       |   |
| 00                                  | 1     | 0 |
| 01                                  | 1     | X |
| 11                                  | 0     | X |
| 10                                  | X     | X |

**$D_B$**

| $Q_C \backslash Q_B \backslash Q_A$ | $Q_A$ |   |
|-------------------------------------|-------|---|
|                                     | 0     | 1 |
| $Q_C \backslash Q_B$                |       |   |
| 00                                  | 0     | 0 |
| 01                                  | 0     | X |
| 11                                  | 1     | X |
| 10                                  | X     | X |

**$D_A$**

# Construction de Karnaugh avec les bascules D :

## étape 5

déterminer somme de produits simplifiées pour les entrées

| $Q_C Q_B \backslash Q_A$ | 0 | 1 |
|--------------------------|---|---|
| 00                       | 0 | 0 |
| 01                       | 1 | X |
| 11                       | 0 | X |
| 10                       | X | X |

$D_C$

$$D_C = Q'_C Q_B$$

| $Q_C Q_B \backslash Q_A$ | 0 | 1 |
|--------------------------|---|---|
| 00                       | 1 | 0 |
| 01                       | 1 | X |
| 11                       | 0 | X |
| 10                       | X | X |

$D_B$

$$D_B = Q'_C Q'_A$$

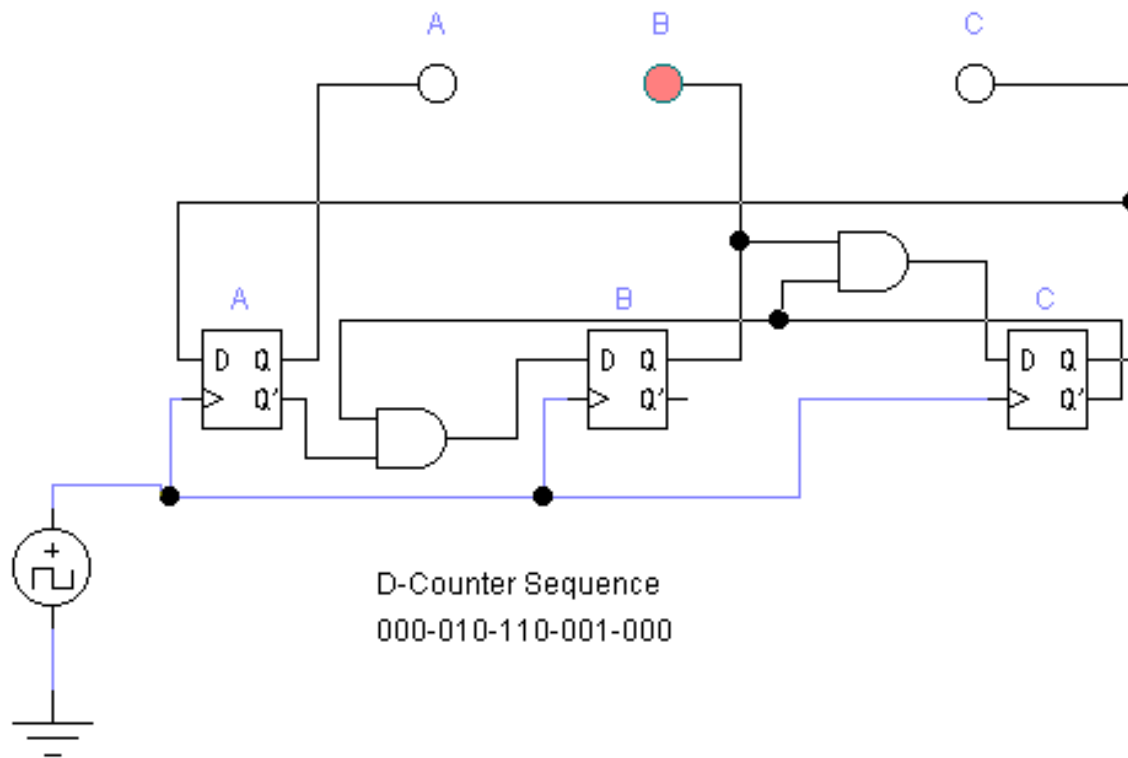
| $Q_C Q_B \backslash Q_A$ | 0 | 1 |
|--------------------------|---|---|
| 00                       | 0 | 0 |
| 01                       | 0 | X |
| 11                       | 1 | X |
| 10                       | X | X |

$D_A$

$$D_A = Q_C$$

# Diagramme Implémentation

- Construire le diagramme du circuits
- Vérifier ses opérations.



# Exemple de Compteur Binaire

---

- Exemples de compteur binaires avec la bascule JK
  - 1- compteur complet sans états manquant
  - 2- compteur avec des états manquants

# Exemple de Compteur Binaire Avec JK

## Table de transition pour JK

(a) JK flip-flop truth table

| 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         |

(b) Transition table for JK flip-flops

| $Q_n$ | $Q_{n+1}$ | J | K |
|-------|-----------|---|---|
| 0     | 0         | 0 | x |
| 0     | 1         | 1 | x |
| 1     | 0         | x | 1 |
| 1     | 1         | x | 0 |

# Exemple de Compteur Binaire Avec JK

---

- Construire le diagramme d'états
- Construire la table des états qui comprend
  - \* Etat Présent
  - \* Etat suivant
  - \* Les entrées JK pour chaque bascule
- Utiliser la table K-maps pour simplifier les expressions pour les entrées JK
- Construire le circuit pour le compteur

# Exemple de Compteur Binaire Avec JK

---

**0 → 1 → 2 → 3 → 4 → 5 → 6 → 7**

- compteur binaire a 3-bit
- 3 bascules JK
- Etat présent et Etat suivant ont chacun trois bits
- 3 pairs d'entrées JK

# Exemple de Compteur Binaire Avec JK

Table d'états pour le compteur binaire

| Present state |   |   | Next state |   |   | JK flip-flop inputs |                |                |                |                |                |
|---------------|---|---|------------|---|---|---------------------|----------------|----------------|----------------|----------------|----------------|
| A             | B | C | A          | B | C | J <sub>A</sub>      | K <sub>A</sub> | J <sub>B</sub> | K <sub>B</sub> | J <sub>C</sub> | K <sub>C</sub> |
| 0             | 0 | 0 | 0          | 0 | 1 | 0                   | X              | 0              | X              | 1              | X              |
| 0             | 0 | 1 | 0          | 1 | 0 | 0                   | X              | 1              | X              | X              | 1              |
| 0             | 1 | 0 | 0          | 1 | 1 | 0                   | X              | X              | 0              | 1              | X              |
| 0             | 1 | 1 | 1          | 0 | 0 | 1                   | X              | X              | 1              | X              | 1              |
| 1             | 0 | 0 | 1          | 0 | 1 | X                   | 0              | 0              | X              | 1              | X              |
| 1             | 0 | 1 | 1          | 1 | 0 | X                   | 0              | 1              | X              | X              | 1              |
| 1             | 1 | 0 | 1          | 1 | 1 | X                   | 0              | X              | 0              | 1              | X              |
| 1             | 1 | 1 | 0          | 0 | 0 | X                   | 1              | X              | 1              | X              | 1              |

# Exemple de Compteur Binaire Avec JK

Utilisation de K-maps pour simplifier les expressions des entrées JK

$d = X$

|   |  |    |    |    |    |
|---|--|----|----|----|----|
|   |  | BC |    |    |    |
| A |  | 00 | 01 | 11 | 10 |
| 0 |  | 0  | 0  | 1  | 0  |
| 1 |  | d  | d  | d  | d  |

$$J_A = B C$$

|   |  |    |    |    |    |
|---|--|----|----|----|----|
|   |  | BC |    |    |    |
| A |  | 00 | 01 | 11 | 10 |
| 0 |  | d  | d  | d  | d  |
| 1 |  | 0  | 0  | 1  | 0  |

$$K_A = B C$$

|   |  |    |    |    |    |
|---|--|----|----|----|----|
|   |  | BC |    |    |    |
| A |  | 00 | 01 | 11 | 10 |
| 0 |  | 0  | 1  | d  | d  |
| 1 |  | 0  | 1  | d  | d  |

$$J_B = C$$

|   |  |    |    |    |    |
|---|--|----|----|----|----|
|   |  | BC |    |    |    |
| A |  | 00 | 01 | 11 | 10 |
| 0 |  | d  | d  | 1  | 0  |
| 1 |  | d  | d  | 1  | 0  |

$$K_B = C$$

|   |  |    |    |    |    |
|---|--|----|----|----|----|
|   |  | BC |    |    |    |
| A |  | 00 | 01 | 11 | 10 |
| 0 |  | 1  | d  | d  | 1  |
| 1 |  | 1  | d  | d  | 1  |

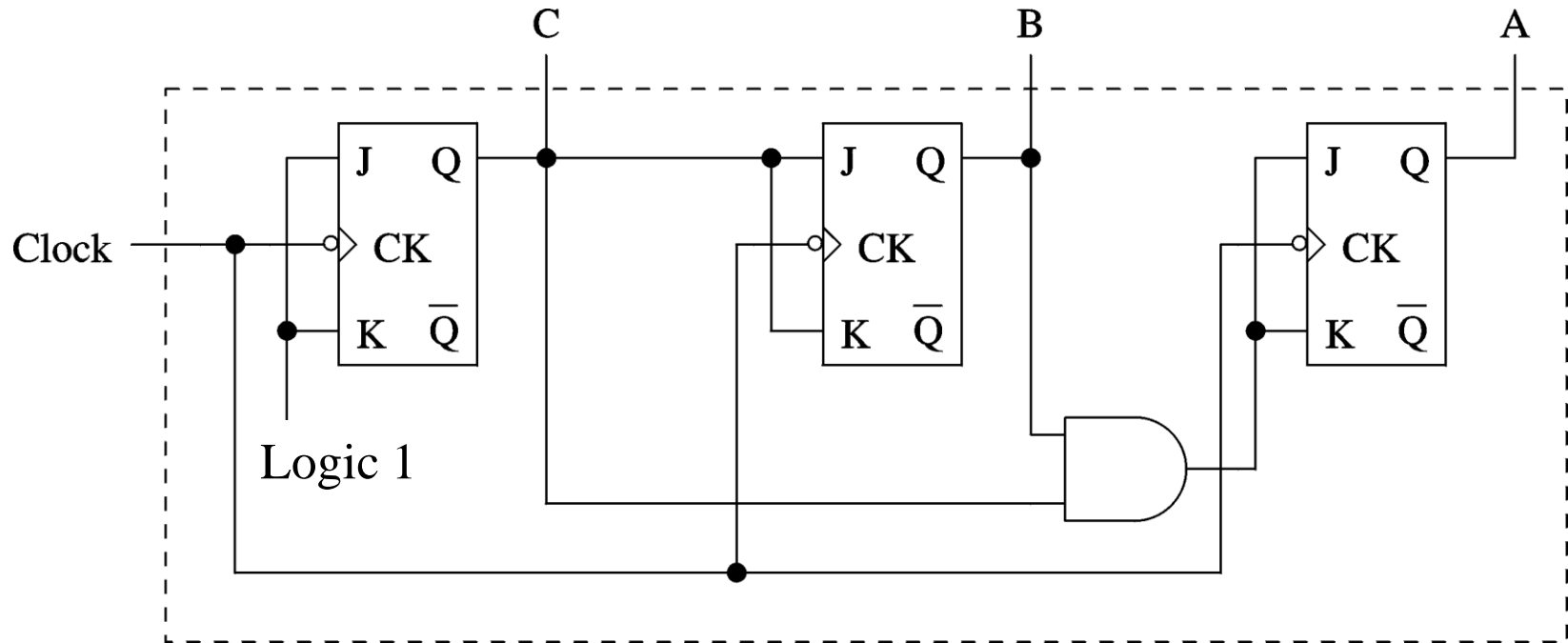
$$J_C = 1$$

|   |  |    |    |    |    |
|---|--|----|----|----|----|
|   |  | BC |    |    |    |
| A |  | 00 | 01 | 11 | 10 |
| 0 |  | d  | 1  | 1  | d  |
| 1 |  | d  | 1  | 1  | d  |

$$K_C = 1$$

# Exemple de Compteur Binaire Avec JK

- Le circuit final pour le compteur binaire



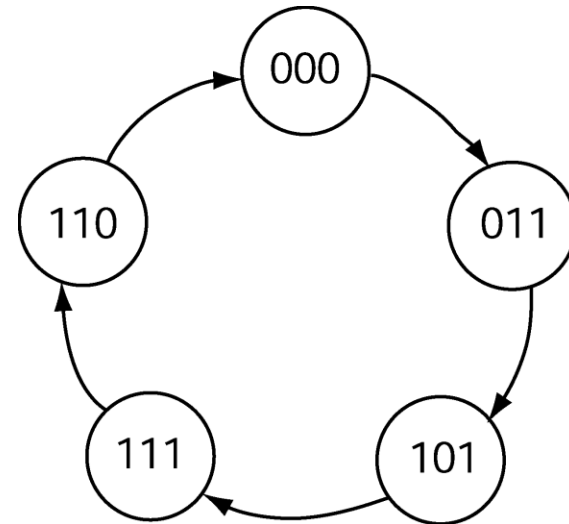
# Exemple de compteur avec JK ( avec états manquants )

---

→ Exemple avec états manquants

0 → 3 → 5 → 7 → 6 → 0

- Même procédure de conception que précédemment
- Un changement:
  - \* **Etats manquants**
    - » 1, 2, et 4
    - » utilisation des (don't care) indéterminés pour ces états



# Exemple de compteur avec JK ( avec états manquants )

Table d'états pour le compteur binaire

| Present state |   |   | Next state |   |   | JK flip-flop inputs |                |                |                |                |                |
|---------------|---|---|------------|---|---|---------------------|----------------|----------------|----------------|----------------|----------------|
| A             | B | C | A          | B | C | J <sub>A</sub>      | K <sub>A</sub> | J <sub>B</sub> | K <sub>B</sub> | J <sub>C</sub> | K <sub>C</sub> |
| 0             | 0 | 0 | 0          | 1 | 1 | 0                   | X              | 1              | X              | 1              | X              |
| 0             | 0 | 1 | X          | X | X | X                   | X              | X              | X              | X              | X              |
| 0             | 1 | 0 | X          | X | X | X                   | X              | X              | X              | X              | X              |
| 0             | 1 | 1 | 1          | 0 | 1 | 1                   | X              | X              | 1              | X              | 0              |
| 1             | 0 | 0 | X          | X | X | X                   | X              | X              | X              | X              | X              |
| 1             | 0 | 1 | 1          | 1 | 1 | X                   | 0              | 1              | X              | X              | 0              |
| 1             | 1 | 0 | 0          | 0 | 0 | X                   | 1              | X              | 1              | 0              | X              |
| 1             | 1 | 1 | 1          | 1 | 0 | X                   | 0              | X              | 0              | X              | 1              |

# Exemple de compteur avec JK ( avec états manquants )

K-maps to simplify JK input expressions

X=d

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | BC |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| A | 0 | 0  | d  | 1  | d  |
|   | 1 | d  | d  | d  | d  |

$$J_A = B$$

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | BC |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| A | 0 | d  | d  | d  | d  |
|   | 1 | d  | 0  | 0  | 1  |

$$K_A = \bar{C}$$

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | BC |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| A | 0 | 1  | d  | d  | d  |
|   | 1 | d  | 1  | d  | d  |

$$J_B = 1$$

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | BC |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| A | 0 | d  | d  | 1  | d  |
|   | 1 | d  | d  | 0  | 1  |

$$K_B = \bar{A} + \bar{C}$$

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | BC |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| A | 0 | 1  | d  | d  | d  |
|   | 1 | d  | d  | d  | 0  |

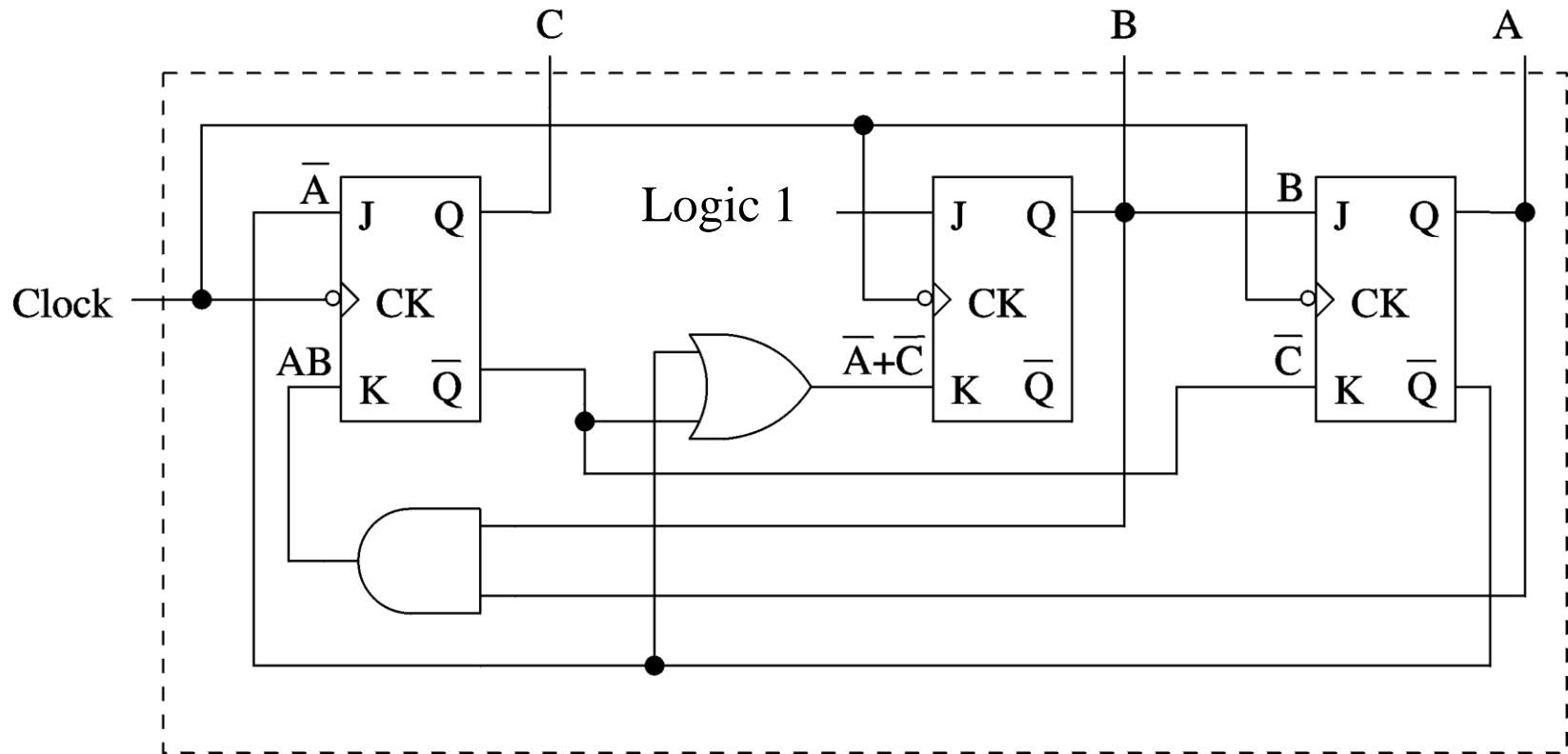
$$J_C = \bar{A}$$

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | BC |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| A | 0 | d  | d  | 0  | d  |
|   | 1 | d  | 0  | 1  | d  |

$$K_C = A B$$

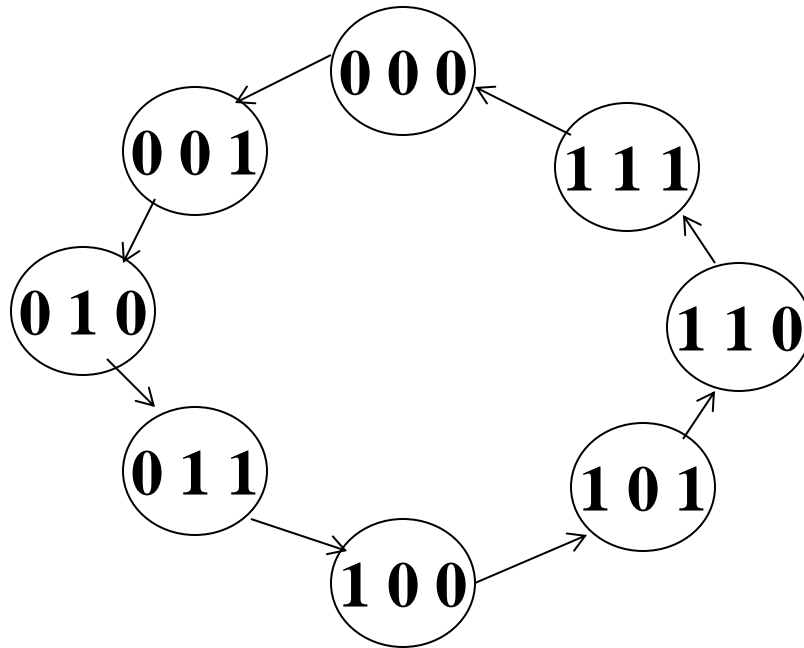
# Exemple de compteur avec JK ( avec états manquants )

circuit final



# Exemple de compteur binaire a 3 bits avec la Bascule T

---



# Exemple de compteur binaire a 3 bits avec la Bascule T

---

| <b>Etat Present</b> | <b>Etat Suivant</b> | <b>Entrees des bascules</b> |            |            |
|---------------------|---------------------|-----------------------------|------------|------------|
| <b>A B C</b>        | <b>A B C</b>        | <b>T A</b>                  | <b>T B</b> | <b>T C</b> |
| <b>0 0 0</b>        | <b>0 0 1</b>        | <b>0</b>                    | <b>0</b>   | <b>1</b>   |
| <b>0 0 1</b>        | <b>0 1 0</b>        | <b>0</b>                    | <b>1</b>   | <b>1</b>   |
| <b>0 1 0</b>        | <b>0 1 1</b>        | <b>0</b>                    | <b>0</b>   | <b>1</b>   |
| <b>0 1 1</b>        | <b>1 0 0</b>        | <b>1</b>                    | <b>1</b>   | <b>1</b>   |
| <b>1 0 0</b>        | <b>1 0 1</b>        | <b>0</b>                    | <b>0</b>   | <b>1</b>   |
| <b>1 0 1</b>        | <b>1 1 0</b>        | <b>0</b>                    | <b>1</b>   | <b>1</b>   |
| <b>1 1 0</b>        | <b>1 1 1</b>        | <b>0</b>                    | <b>0</b>   | <b>1</b>   |
| <b>1 1 1</b>        | <b>0 0 0</b>        | <b>1</b>                    | <b>1</b>   | <b>1</b>   |

# Exemple de compteur binaire a 3 bits avec la Bascule T

- Kmap

|   |   |   |   |
|---|---|---|---|
|   | B |   |   |
|   | 0 | 1 | 0 |
| A | 0 | 1 | 0 |

TA = BC      C

|   |   |   |   |
|---|---|---|---|
|   | B |   |   |
|   | 0 | 1 | 0 |
| A | 0 | 1 | 0 |

C      TB = C

|   |   |   |   |
|---|---|---|---|
|   | B |   |   |
|   | 0 | 1 | 0 |
| A | 1 | 1 | 1 |

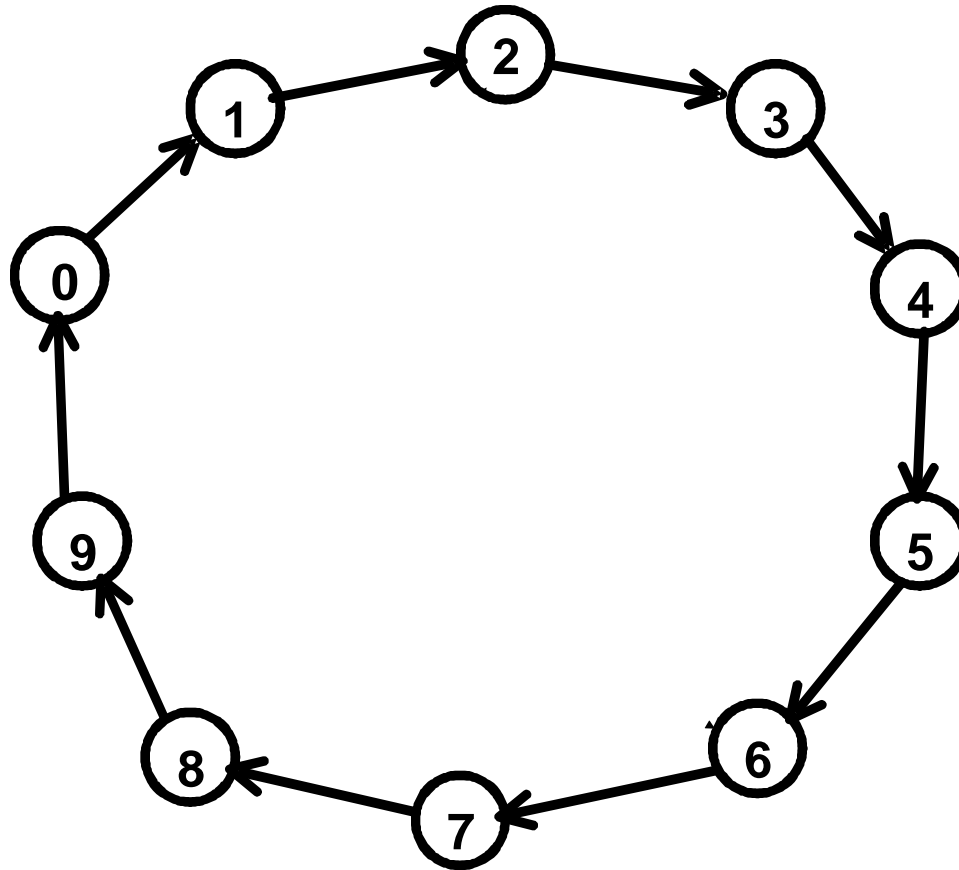
TC = 1



# Compteur Décimal Code Binaire (BCD)

---

- Avec la bascule T



# Compteur Décimal Code Binaire (BCD)

---

- Table des Etats

| <b>Etat Present</b><br><b>Q<sub>8</sub> Q<sub>4</sub> Q<sub>2</sub> Q<sub>1</sub></b> | <b>Etat Suivant</b><br><b>Q<sub>8</sub> Q<sub>4</sub> Q<sub>2</sub> Q<sub>1</sub></b> | <b>T-Entrees</b><br><b>T<sub>8</sub> T<sub>4</sub> T<sub>2</sub> T<sub>1</sub></b> |
|---|---|--|
| <b>0 0 0 0</b>  | <b>0 0 0 1</b>  |  |
| <b>0 0 0 1</b>  | <b>0 0 1 0</b>  |  |
| <b>0 0 1 0</b>  | <b>0 0 1 1</b>  |  |
| <b>0 0 1 1</b>  | <b>0 1 0 0</b>  |  |
| <b>0 1 0 0</b>  | <b>0 1 0 1</b>  |  |
| <b>0 1 0 1</b>  | <b>0 1 1 0</b>  |  |
| <b>0 1 1 0</b>  | <b>0 1 1 1</b>  |  |
| <b>0 1 1 1</b>  | <b>1 0 0 0</b>  |  |
| <b>1 0 0 0</b>  | <b>1 0 0 1</b>  |  |
| <b>1 0 0 1</b>  | <b>0 0 0 0</b>  |  |

# Compteur Décimal Code Binaire (BCD synchrone)

Table des Etats (avec les entrées de la bascule T)

- Don't care ne figurent pas ici

| Etat Present   |                |                |                | Etat suivant   |                |                |                | T-Entrees      |                |                |                |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Q <sub>8</sub> | Q <sub>4</sub> | Q <sub>2</sub> | Q <sub>1</sub> | Q <sub>8</sub> | Q <sub>4</sub> | Q <sub>2</sub> | Q <sub>1</sub> | T <sub>8</sub> | T <sub>4</sub> | T <sub>2</sub> | T <sub>1</sub> |
| 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 1              |
| 0              | 0              | 0              | 1              | 0              | 0              | 1              | 0              | 0              | 0              | 1              | 1              |
| 0              | 0              | 1              | 0              | 0              | 0              | 1              | 1              | 0              | 0              | 0              | 1              |
| 0              | 0              | 1              | 1              | 0              | 1              | 0              | 0              | 0              | 1              | 1              | 1              |
| 0              | 1              | 0              | 0              | 0              | 1              | 0              | 1              | 0              | 0              | 0              | 1              |
| 0              | 1              | 0              | 1              | 0              | 1              | 1              | 0              | 0              | 0              | 1              | 1              |
| 0              | 1              | 1              | 0              | 0              | 1              | 1              | 1              | 0              | 0              | 0              | 1              |
| 0              | 1              | 1              | 1              | 1              | 0              | 0              | 0              | 1              | 1              | 1              | 1              |
| 1              | 0              | 0              | 0              | 1              | 0              | 0              | 1              | 0              | 0              | 0              | 1              |
| 1              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 1              |

# Compteur Décimal Code Binaire (BCD, Synchrones)

- simplification avec K-Maps

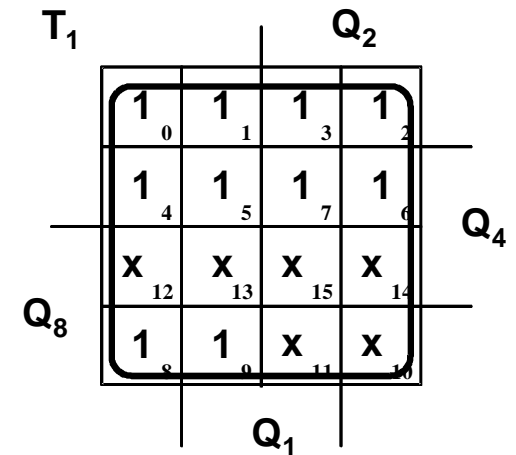
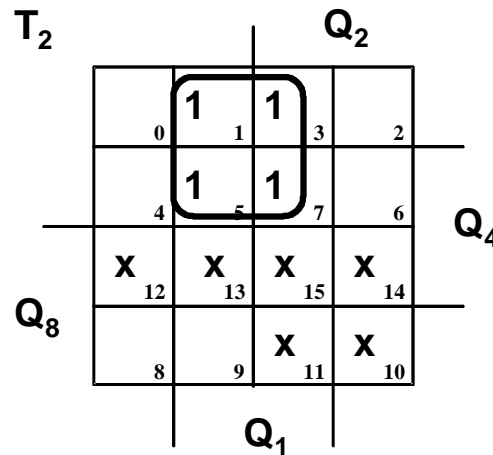
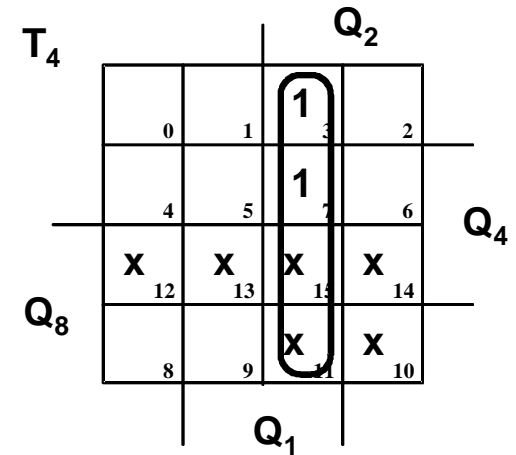
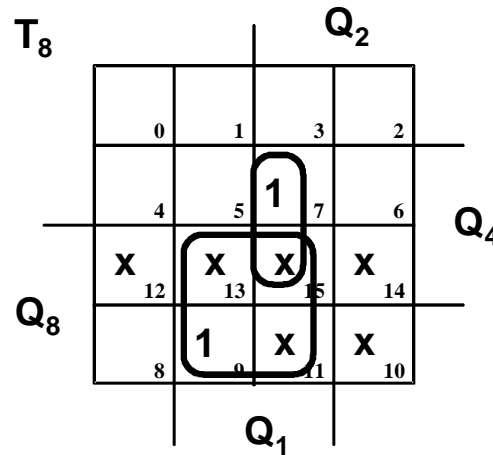
$$T_8 = Q_8 \bullet Q_1 + Q_4 \bullet Q_2 \bullet Q_1$$

$$T_4 = Q_2 \bullet Q_1$$

$$T_2 = Q_8' \bullet Q_1$$

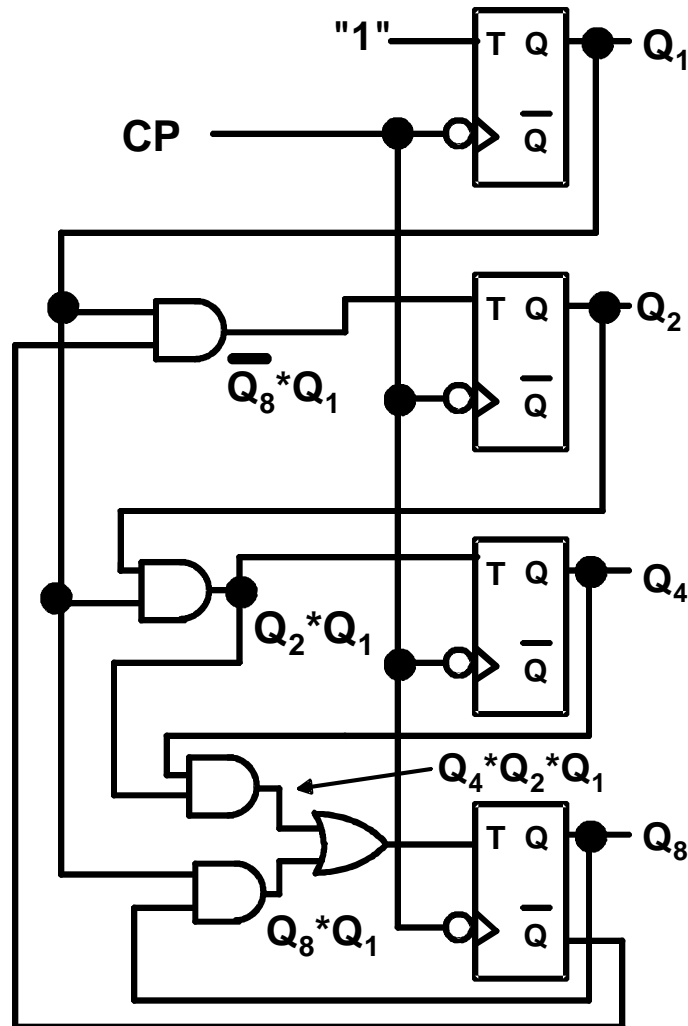
$$T_1 = "1"$$

Note: Don't Cares sont inclus ici



# Compteur Décimal Code Binaire (BCD, Synchrones)

- circuit:



# Compteur décimal codé binaire (asynchrone, Ripple counter)

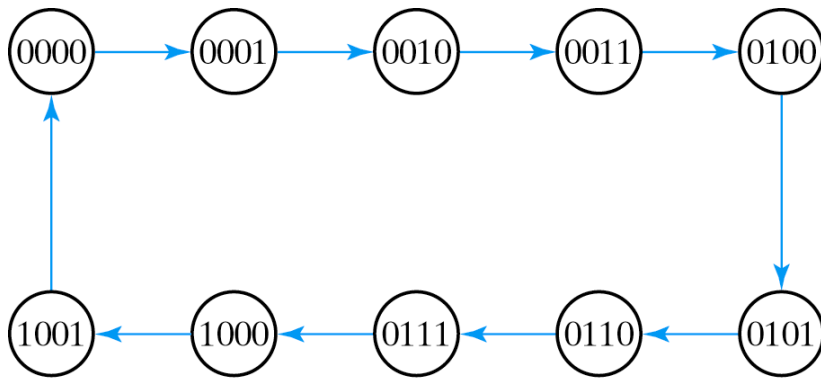
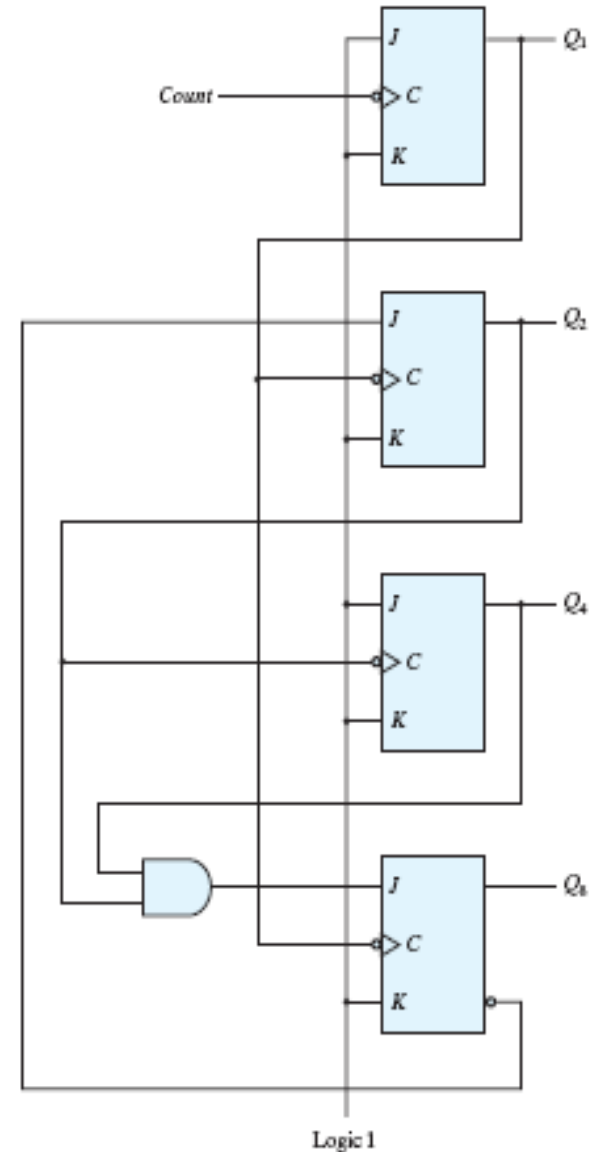


Fig. 6-9 State Diagram of a Decimal BCD-Counter



# Concevoir un compteur décimal codé binaire a étage

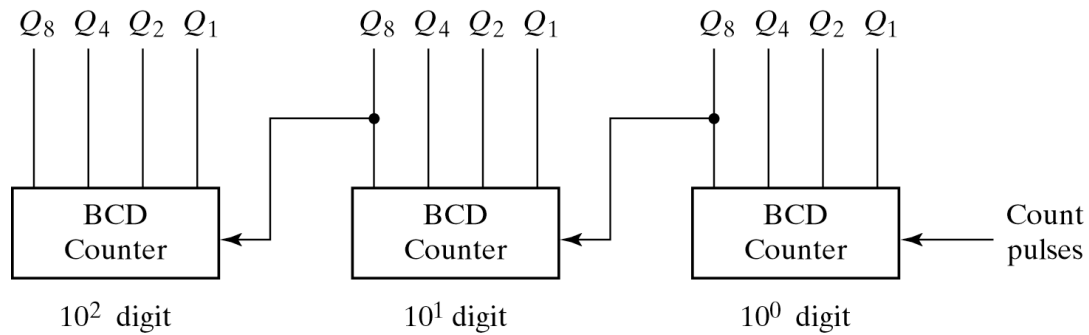


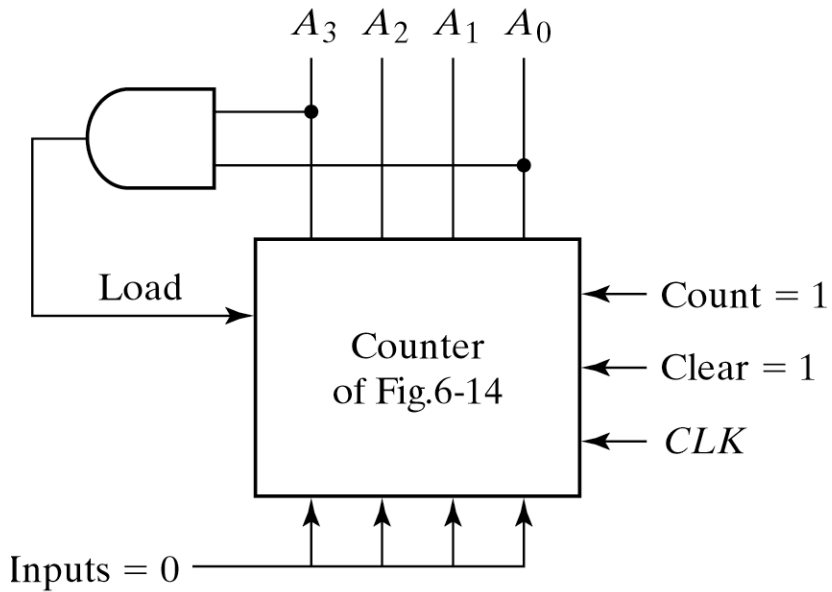
Fig. 6-11 Block Diagram of a Three-Decade Decimal BCD Counter

# Concevoir un compteur décimal codé binaire a étages

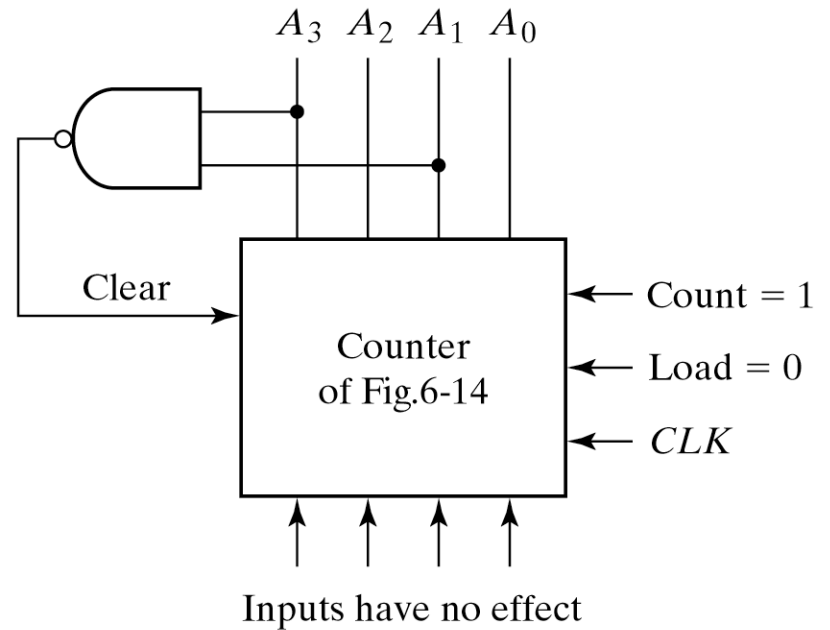
| Present State  |                |                |                | Next State     |                |                |                | Output | Input           |                 |                 |                 |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------|-----------------|-----------------|-----------------|-----------------|
| Q <sub>8</sub> | Q <sub>4</sub> | Q <sub>2</sub> | Q <sub>1</sub> | Q <sub>8</sub> | Q <sub>4</sub> | Q <sub>2</sub> | Q <sub>1</sub> | y      | TQ <sub>8</sub> | TQ <sub>4</sub> | TQ <sub>2</sub> | TQ <sub>1</sub> |
| 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0      | 0               | 0               | 0               | 1               |
| 0              | 0              | 0              | 1              | 0              | 0              | 1              | 0              | 0      | 0               | 0               | 1               | 1               |
| 0              | 0              | 1              | 0              | 0              | 0              | 1              | 1              | 0      | 0               | 0               | 0               | 1               |
| 0              | 0              | 1              | 1              | 0              | 1              | 0              | 0              | 0      | 0               | 1               | 1               | 1               |
| 0              | 1              | 0              | 0              | 0              | 1              | 0              | 1              | 0      | 0               | 0               | 0               | 1               |
| 0              | 1              | 1              | 0              | 0              | 1              | 1              | 0              | 0      | 0               | 0               | 1               | 1               |
| 0              | 1              | 0              | 1              | 0              | 1              | 1              | 1              | 0      | 0               | 0               | 0               | 1               |
| 0              | 1              | 1              | 1              | 1              | 0              | 0              | 0              | 0      | 1               | 1               | 1               | 1               |
| 1              | 0              | 0              | 0              | 1              | 0              | 0              | 1              | 0      | 0               | 0               | 0               | 1               |
| 1              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 1      | 1               | 0               | 0               | 1               |

$$TQ_1 = 1, \quad TQ_2 = Q'_8 Q_1, \quad TQ_4 = Q_2 Q_1, \quad TQ_8 = Q_8 Q_1 + Q_4 Q_2 Q_1, \quad y = Q_8 Q_1$$

# Compteur DCB avec chargement parallèle



(a) Using the load input



(b) Using the clear input

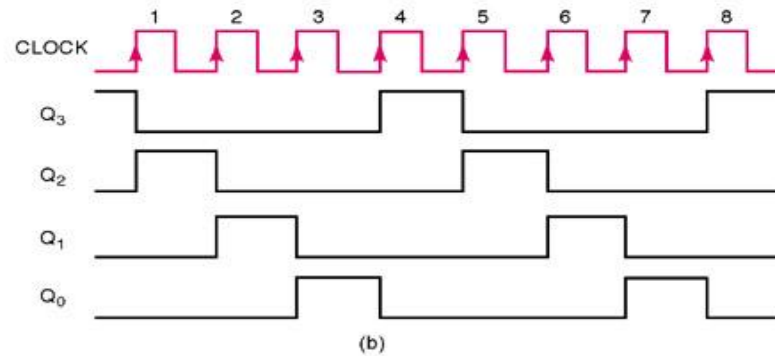
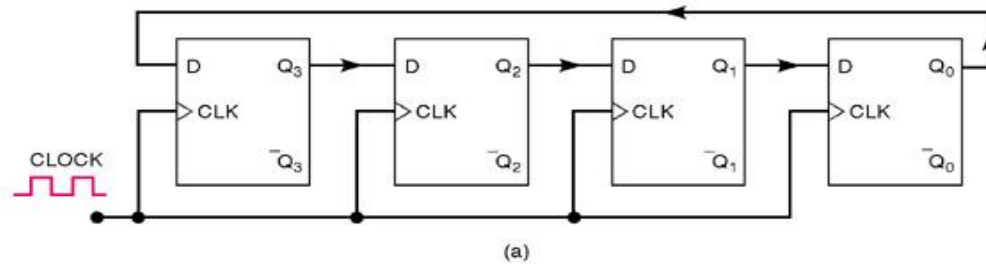
Fig. 6-15 Two ways to Achieve a BCD Counter Using a Counter with Parallel Load

# Compteur en Anneau

---

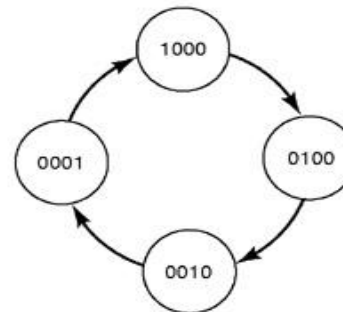
- Dans le compteur en anneau on récupère la sortie en série de la dernière bascule du registre a décalage et la fournit a l'entrée série de la première bascule.

# Compteur en Anneau



| Q <sub>3</sub> | Q <sub>2</sub> | Q <sub>1</sub> | Q <sub>0</sub> | CLOCK pulse |
|----------------|----------------|----------------|----------------|-------------|
| 1              | 0              | 0              | 0              | 0           |
| 0              | 1              | 0              | 0              | 1           |
| 0              | 0              | 1              | 0              | 2           |
| 0              | 0              | 0              | 1              | 3           |
| 1              | 0              | 0              | 0              | 4           |
| 0              | 1              | 0              | 0              | 5           |
| 0              | 0              | 1              | 0              | 6           |
| 0              | 0              | 0              | 1              | 7           |
| .              | .              | .              | .              | .           |
| .              | .              | .              | .              | .           |

(c)

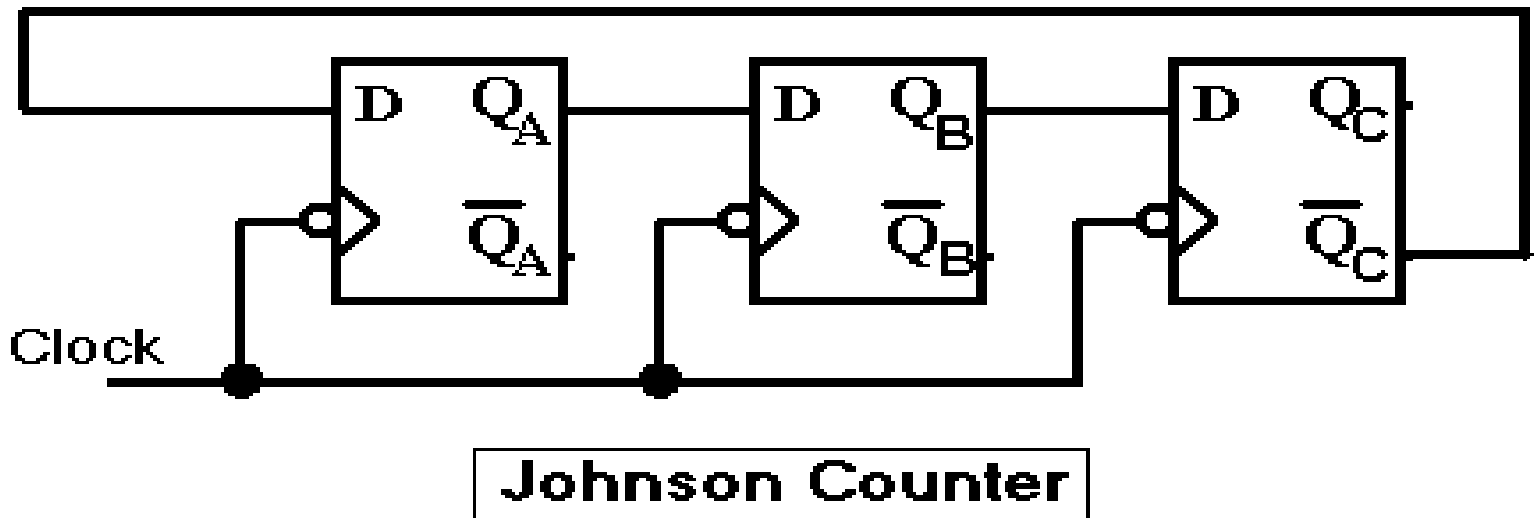


# Compteur de Johnson (1)

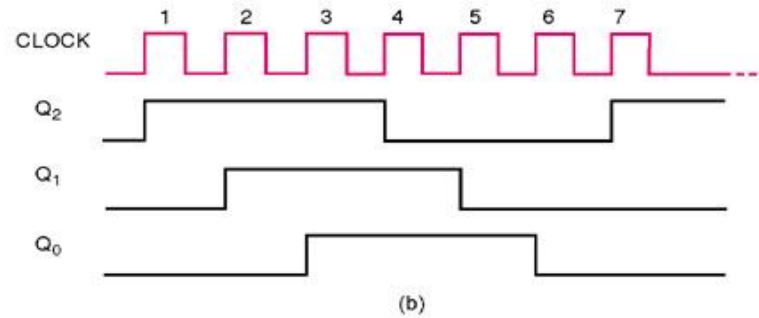
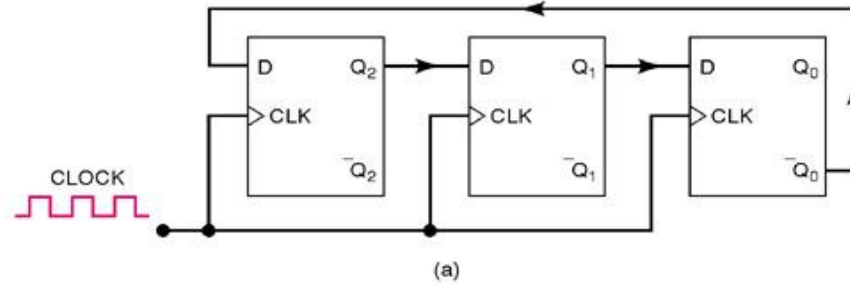
- le compteur de Johnson est le même que le compteur en anneau sauf que la sortie  $Q'$  (inversée) est considérée au lieu de  $Q$

000, 100, 110, 111, 011, 001, 000, 100, 110, ....

Etat Initial

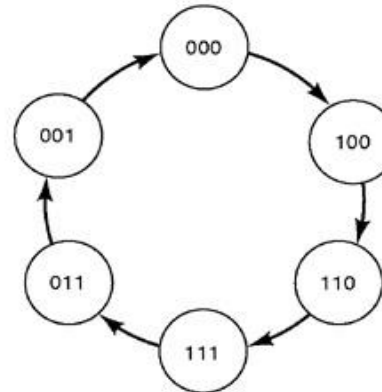


# Compteur de Johnson (2)



| Q <sub>2</sub> | Q <sub>1</sub> | Q <sub>0</sub> | CLOCK pulse |
|----------------|----------------|----------------|-------------|
| 0              | 0              | 0              | 0           |
| 1              | 0              | 0              | 1           |
| 1              | 1              | 0              | 2           |
| 1              | 1              | 1              | 3           |
| 0              | 1              | 1              | 4           |
| 0              | 0              | 1              | 5           |
| 0              | 0              | 0              | 6           |
| 1              | 0              | 0              | 7           |
| 1              | 1              | 0              | 8           |
| ·              | ·              | ·              | ·           |
| ·              | ·              | ·              | ·           |
| ·              | ·              | ·              | ·           |

(c)



(d)

---

**Fin du chapitre 6 et du Cours!!**