

2E200 : Electronique Numérique, Combinatoire et Séquentielle

Bertrand Granado

Licence E²A

Hiver 2019



Plan

- 1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
- 2 Algèbre de Boole
- 3 Codage
- 4 Les composants combinatoire simples
- 5 Les composants combinatoires complexes
- 6 Les composants séquentiels
- 7 Interface avec l'environnement continu : Conversion Analogique vers Numérique et Numérique vers Analogique

Plan

- 1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
- 2 Algèbre de Boole
- 3 Codage
- 4 Les composants combinatoire simples
- 5 Les composants combinatoires complexes
- 6 Les composants séquentiels
- 7 Interface avec l'environnement continu : Conversion Analogique vers Numérique et Numérique vers Analogique

Plan

- 1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
- 2 Algèbre de Boole
- 3 Codage
- 4 Les composants combinatoire simples
- 5 Les composants combinatoires complexes
- 6 Les composants séquentiels
- 7 Interface avec l'environnement continu : Conversion Analogique vers Numérique et Numérique vers Analogique

Plan

- 1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
- 2 Algèbre de Boole
- 3 Codage**
- 4 Les composants combinatoire simples
- 5 Les composants combinatoires complexes
- 6 Les composants séquentiels
- 7 Interface avec l'environnement continu : Conversion Analogique vers Numérique et Numérique vers Analogique

Plan

- 1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
- 2 Algèbre de Boole
- 3 Codage
- 4 Les composants combinatoire simples**
- 5 Les composants combinatoires complexes
- 6 Les composants séquentiels
- 7 Interface avec l'environnement continu : Conversion Analogique vers Numérique et Numérique vers Analogique

Plan

- 1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
- 2 Algèbre de Boole
- 3 Codage
- 4 Les composants combinatoire simples
- 5 Les composants combinatoires complexes
- 6 Les composants séquentiels
- 7 Interface avec l'environnement continu : Conversion Analogique vers Numérique et Numérique vers Analogique

Plan

- 1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
- 2 Algèbre de Boole
- 3 Codage
- 4 Les composants combinatoire simples
- 5 Les composants combinatoires complexes
- 6 Les composants séquentiels**
 - les bascules
 - les registres
 - les compteurs
 - Le contrôle, le traitement Pipeliné
 - What else ?

Et après ?

Les registres

Les registres

- Taille des données 1 bits

Les registres

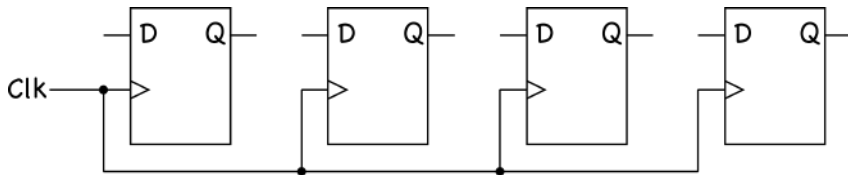
- Taille des données 1 bits
- Associer des Bascules pour augmenter la taille

Les registres

- Taille des données 1 bits
- Associer des Bascules pour augmenter la taille
- Éléments importants dans les micro-processeurs : permet de réaliser un pipeline

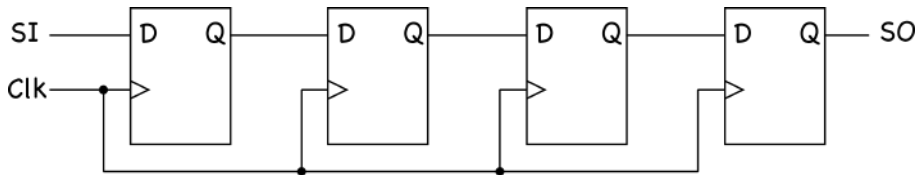
Registre bascules D

Registre



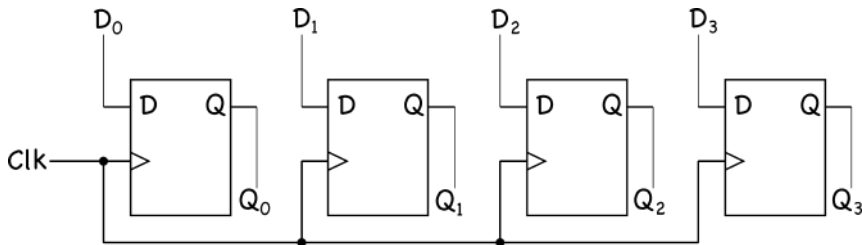
Registre bascules D

Registre à Décalage



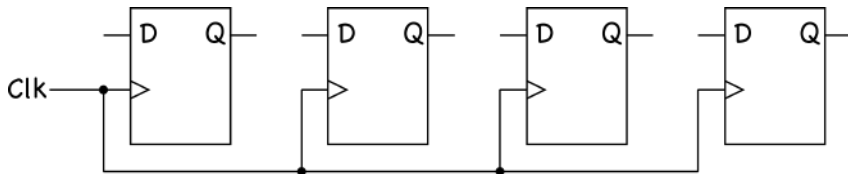
Registre bascules D

Registre à chargement parallèle



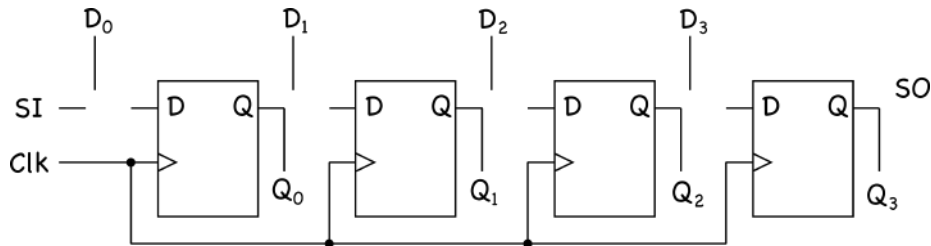
Registre bascules D

Registre Universel



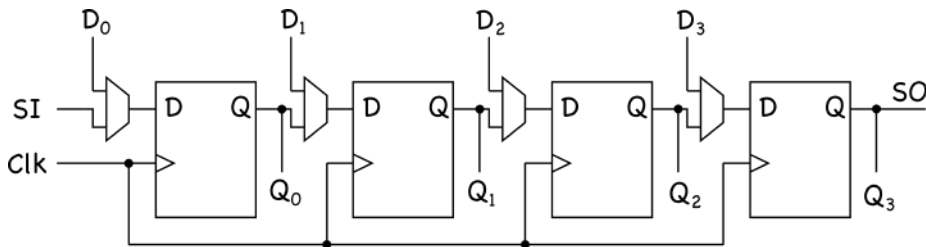
Registre bascules D

Registre Universel



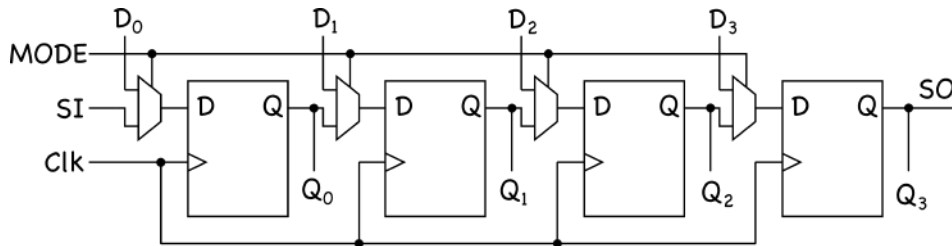
Registre bascules D

Registre Universel



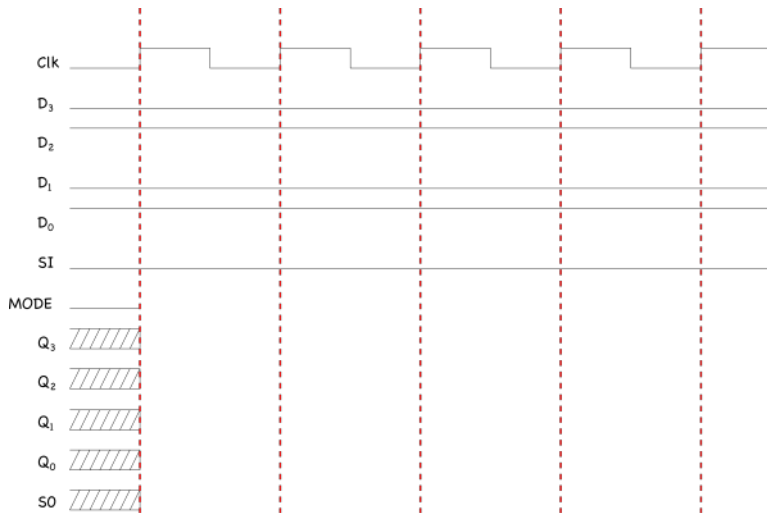
Registre bascules D

Registre Universel



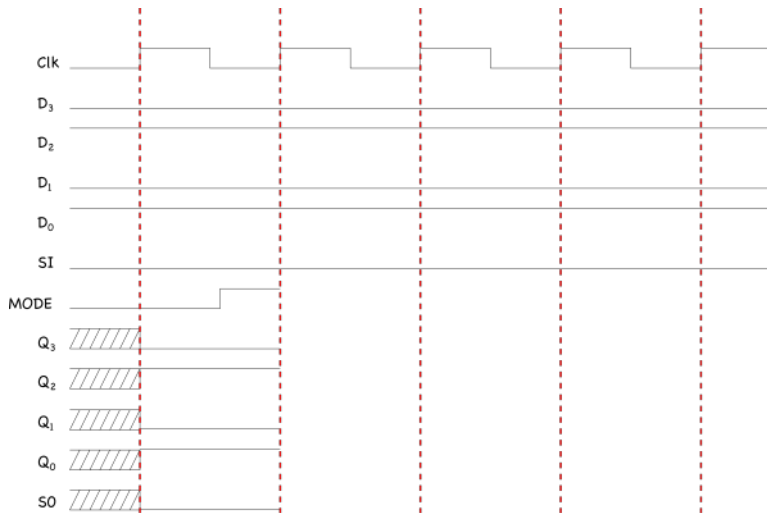
Registre bascules D

Registre Universel



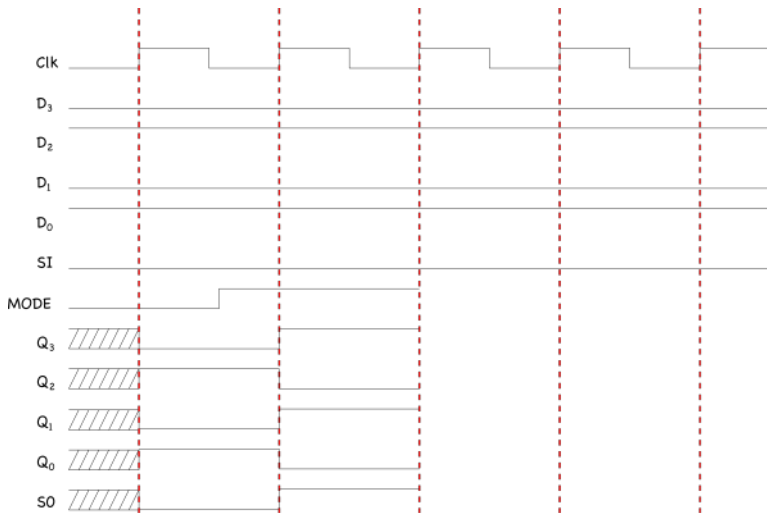
Registre bascules D

Registre Universel



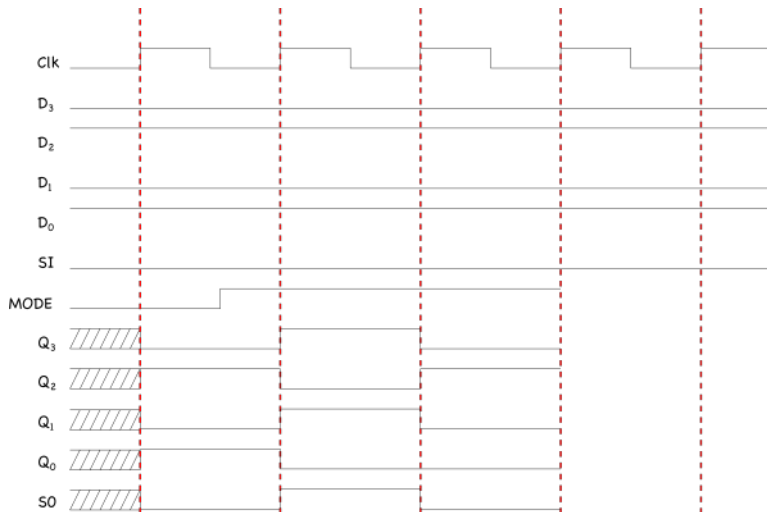
Registre bascules D

Registre Universel



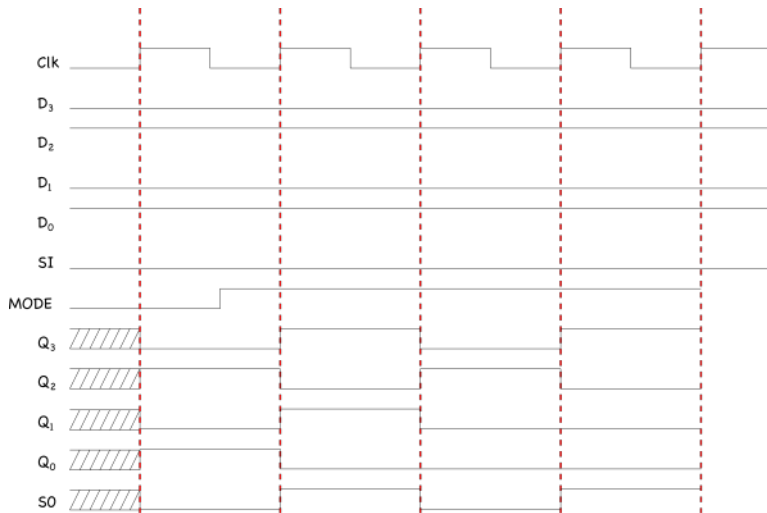
Registre bascules D

Registre Universel



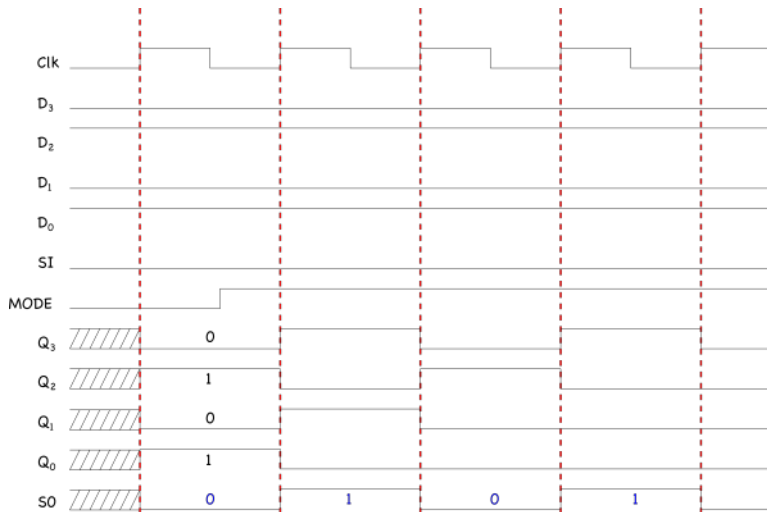
Registre bascules D

Registre Universel



Registre bascules D

Registre Universel



Vhdl : registre universel

```
library ieee;
use ieee.std_logic_1164.all;

entity registre_universel is
port (mode, si, clk : in std_logic;
      d : in std_logic_vector(3 downto 0);
      q : out std_logic_vector(3 downto 0);
      so : out std_logic);
end entity registre_universel;

architecture struct of registre_universel is
signal dint, qint : std_logic_vector(4 downto 0);
begin
  reg: for i in 0 to 3 generate
    basci: entity work.basculer(comport)
      port map (clk => clk, d => dint(i), q => qint(i+1));
    dint(i) <= d(i) when mode = '0' else qint(i);
  end generate;

  qint(0) <= si;
  so <= qint(4);
  q <= qint(4 downto 1);
end architecture struct;
```

Registre N bits

- Un registre à chargement parallèle a le même fonctionnement quelque soit la taille des données qu'il manipule
- En VHDL, il existe une construction, la construction **generic**, qui permet de décrire des composants paramétrables
- Par exemple un registre à chargement parallèle avec un paramètre N variable qui indique la taille des données que manipule le registre

Vhdl : registre générique

```
library ieee;
use ieee.std_logic_1164.all;

entity reg8generic is
generic (N : natural := 8);
port ( d :in std_logic_vector(N-1 downto 0);
      clk : in std_logic;
      q: out std_logic_vector(N-1 downto 0) );
end entity reg8generic;

architecture comport of reg8generic is
begin
  stockage : process (clk,d) is
  begin
    if (clk='1' and clk'event) then
      q <= d;
    end if;
  end process stockage;
end architecture comport;
```

Vhdl : registre générique utilisation

```
library ieee;
use ieee.std_logic_1164.all;

entity doubleregiste is
port ( d :in std_logic_vector(15 downto 0);
      clk : in std_logic;
      q: out std_logic_vector(15 downto 0) );
end entity doubleregiste;

architecture struct of doubleregiste is
signal qint : std_logic_vector(15 downto 0);
begin;
  reg1 : entity work.reg8generic(comport)
        generic map(N=>16)
        port map(d=>d,clk=>clk,q=>qint);

  reg2 : entity work.reg8generic(comport)
        generic map(N => 16)
        port map(d=>qint,clk=>clk,q=>q);
end architecture struct;
```

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états

Les Compteurs

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D

Les Compteurs

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D
- Plus ou Moins Complexes

Les Compteurs

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D
- Plus ou Moins Complexes
 - ▶ Simple (Bête) compte de 0 à $N - 1$ en boucle

Les Compteurs

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D
- Plus ou Moins Complexes
 - ▶ Simple (Bête) compte de 0 à $N - 1$ en boucle
 - ▶ $\log_2(N)$ bascules

Les Compteurs

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D
- Plus ou Moins Complexes
 - ▶ Simple (Bête) compte de 0 à $N - 1$ en boucle
 - ▶ $\log_2(N)$ bascules
 - ▶ Complexes

Les Compteurs

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D
- Plus ou Moins Complexes
 - ▶ Simple (Bête) compte de 0 à $N - 1$ en boucle
 - ▶ $\log_2(N)$ bascules
 - ▶ Complexes
 - ★ Initialisation

Les Compteurs

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D
- Plus ou Moins Complexes
 - ▶ Simple (Bête) compte de 0 à $N - 1$ en boucle
 - ▶ $\log_2(N)$ bascules
 - ▶ Complexes
 - ★ Initialisation
 - ★ Arrêt - Reprise

Les Compteurs

- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D
- Plus ou Moins Complexes
 - ▶ Simple (Bête) compte de 0 à $N - 1$ en boucle
 - ▶ $\log_2(N)$ bascules
 - ▶ Complexes
 - ★ Initialisation
 - ★ Arrêt - Reprise
 - ★ Compteur - Décompteur

Les Compteurs

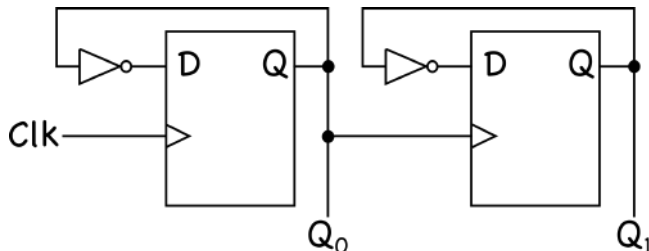
- Système séquentiel dont les sorties sont une suite pré-déterminée d'états
- Bascules D
- Plus ou Moins Complexes
 - ▶ Simple (Bête) compte de 0 à $N - 1$ en boucle
 - ▶ $\log_2(N)$ bascules
 - ▶ Complexes
 - ★ Initialisation
 - ★ Arrêt - Reprise
 - ★ Compteur - Décompteur
 - ★ Fonctions nécessaires à l'application

Les compteurs

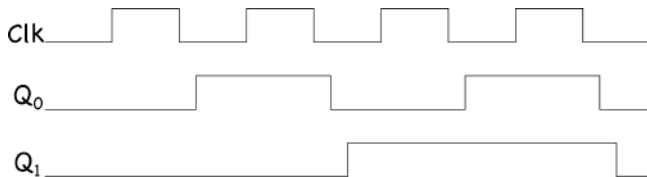
- Il existe deux types de compteurs :
 - ▶ Les compteurs Asynchrones qui utilisent des bascules qui ne partagent pas toutes le même signal d'horloge
 - ▶ Les compteurs Synchrones qui utilisent des bascules qui partagent toutes le même signal d'horloge

Les compteurs Asynchrone

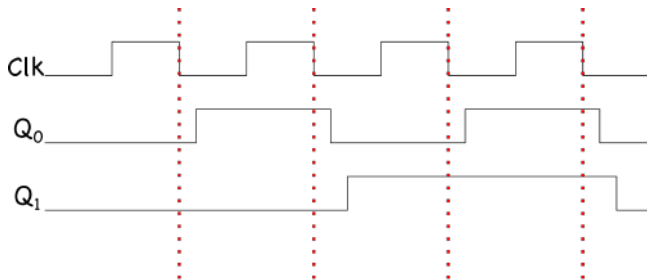
- Compteur par 3 avec des bascules actives sur front descendant



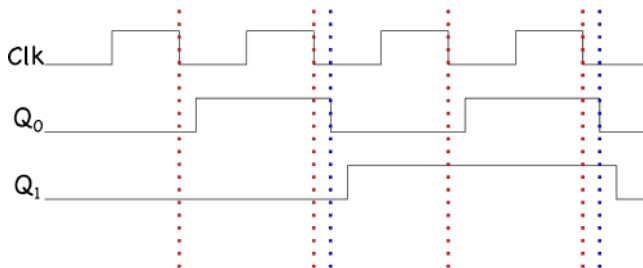
Les compteurs Asynchrone



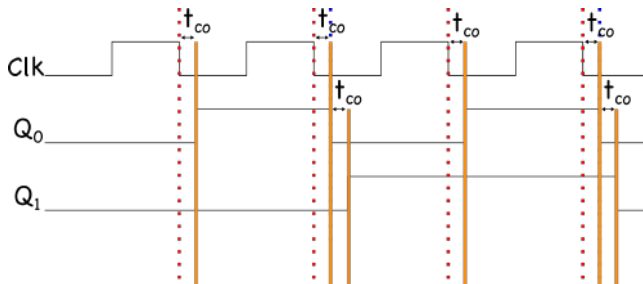
Les compteurs Asynchrone



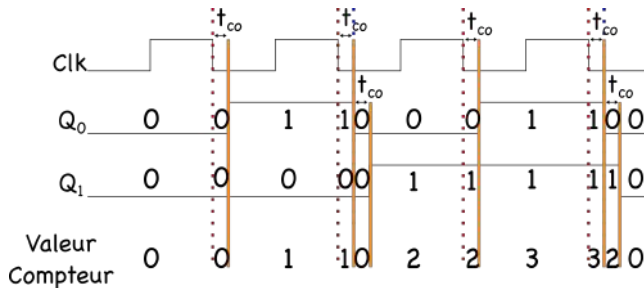
Les compteurs Asynchrone



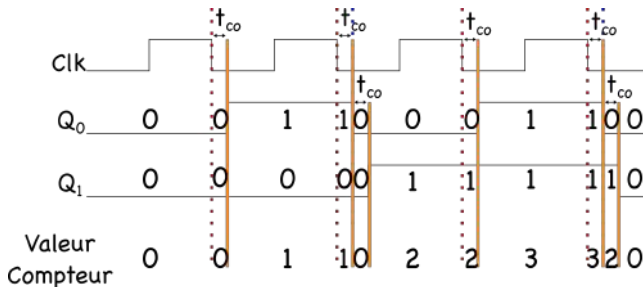
Les compteurs Asynchrone



Les compteurs Asynchrone

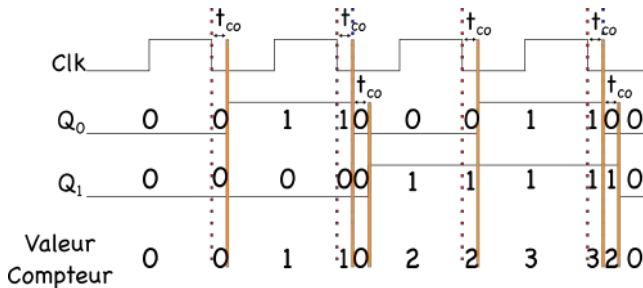


Les compteurs Asynchrone



- Présence d'aléas du à l'asynchronisme du compteur : les bascules n'ont pas la même horloge et recopient leur entrée a des instants décalés

Les compteurs Asynchrone



- Présence d'aléas du à l'asynchronisme du compteur : les bascules n'ont pas la même horloge et recopient leur entrée a des instants décalés
- Possibilité de resynchroniser la sortie du compteur

Les compteurs Synchrones

- Pour réaliser un compteur synchrone, c'est ce que nous appelons la synthèse d'un compteur, il faut :
 - ▶ Déterminer le nombre de bascules nécessaire. Pour un compteur par N il faut $\lceil \log_2(N) \rceil$ bascules.
 - ▶ Etablir une table des transitions
 - ▶ Etablir les équations booléennes des entrées des bascules à partir de la table des transitions
 - ▶ Exemple : faire la synthèse d'un compteur synchrone par 4, c'est à dire un compteur qui a pour cycle 0, 1, 2, 3.

Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 4

- ▶ Nombre de bascules $\lceil \log_2(4) \rceil = 2$. Une bascule D_0 et une bascule D_1 .
- ▶ Table des transitions :

Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Instant n Instant $n+1$

- ▶ Equations :

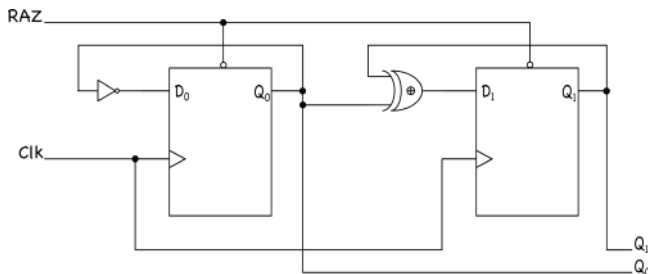
- ★ $Q_0^{n+1} = \overline{Q_0^n}$

- ★ $Q_1^{n+1} = \overline{Q_1^n} \cdot Q_0^n + Q_1^n \cdot \overline{Q_0^n} = Q_1^n \oplus Q_0^n$

- ▶ Q_0^{n+1} est l'entrée D de la bascule D_0 , Q_0^n en est sa sortie Q
- ▶ Q_1^{n+1} est l'entrée D de la bascule D_1 , Q_1^n en est sa sortie Q

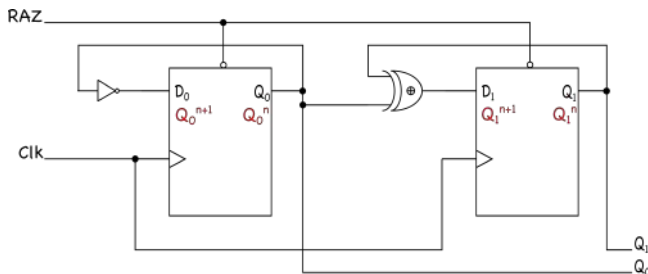
Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 4, schéma



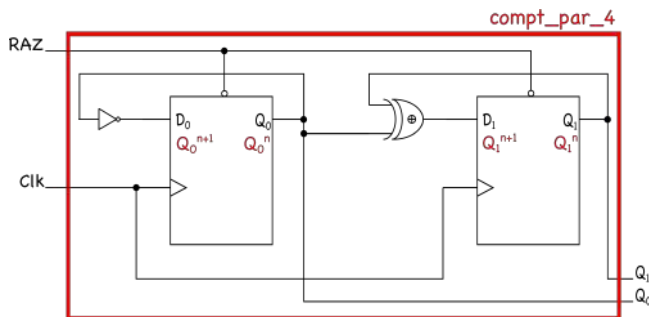
Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 4, schéma



Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 4, schéma



Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 4, description VHDL. Utilisation de bascules avec entrées asynchrones actives à 0.

```
library ieee;
use ieee.std_logic_1164.all;

entity compt_par_4 is
port (clk, raz : in std_logic;
      q : out std_logic_vector(1 downto 0));
end entity compt_par_4;

architecture struct of compt_par_4 is
signal rau : std_logic;
signal dint, qint : std_logic_vector(1 downto 0);
begin
    rau <= '1';
    basc0 : entity work.basculer_raz_rau(comport)
        port map(clk=>clk, raz => raz, rau => rau, d => dint(0), q => qint(0));
    basc1 : entity work.basculer_raz_rau(comport)
        port map(clk=>clk, raz => raz, rau => rau, d => dint(1), q => qint(1));

    dint(0) <= not(qint(0));
    dint(1) <= qint(1) xor qint(0);
    q <= qint;
end architecture struct;
```


Les compteurs Synchrones

- Autre exemple : synthèse d'un compteur synchrone par 5

- ▶ Nombre de bascules $\lceil \log_2(5) \rceil = 3$. Une bascule D_0 , une bascule D_1 et une bascule D_2 .
- ▶ Table des transitions :

Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
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

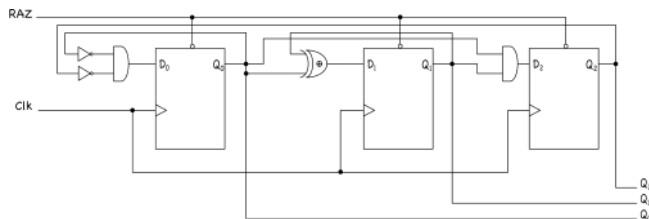
Instant n

Instant $n+1$

- ▶ Equations, il est possible d'utiliser les tableaux de Karnaugh pour obtenir ces équations :
 - ★ $Q_0^{n+1} = \overline{Q_2^n} \cdot \overline{Q_0^n}$
 - ★ $Q_1^{n+1} = Q_1^n \oplus Q_0^n$
 - ★ $Q_2^{n+1} = Q_1^n \cdot Q_0^n$
- ▶ Q_0^{n+1} est l'entrée D de la bascule D_0 , Q_0^n en est sa sortie Q
- ▶ Q_1^{n+1} est l'entrée D de la bascule D_1 , Q_1^n en est sa sortie Q
- ▶ Q_2^{n+1} est l'entrée D de la bascule D_2 , Q_1^n en est sa sortie Q

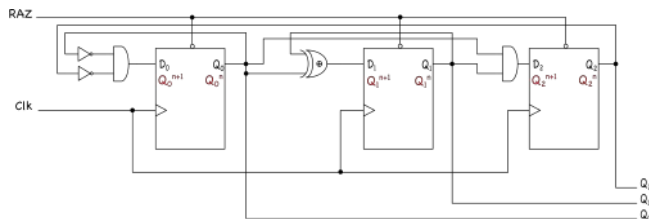
Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 5, schéma



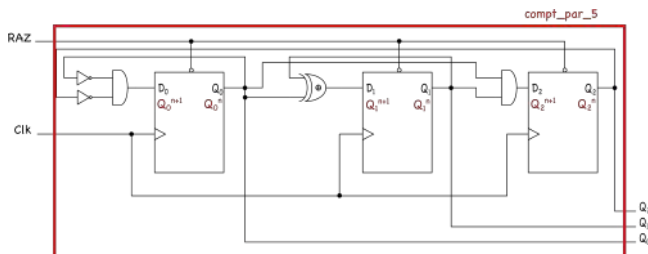
Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 5, schéma



Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 5, schéma



Les compteurs Synchrones

- Synthèse d'un compteur synchrone par 5, description VHDL. Utilisation de bascules avec entrées asynchrones actives à 0.

```
library ieee;
use ieee.std_logic_1164.all;

entity compt_par_5 is
port (clk, raz : in std_logic;
      q : out std_logic_vector(2 downto 0));
end entity compt_par_5;

architecture struct of compt_par_5 is
signal rau : std_logic;
signal dint, qint : std_logic_vector(2 downto 0);
begin
    rau <= '1';
    generebasc : for i in 0 to 2 generate
        basci : entity work.basculer_raz_rau(comport)
            port map(clk=>clk,raz => raz, rau => rau, d => dint(i), q => qint(i));
        end generate generebasc;

    dint(0) <= not(qint(2)) and not(qint(0));
    dint(1) <= qint(1) xor qint(0);
    dint(2) <= qint(1) and qint(0);
    q <= qint;
end architecture struct;
```

Les compteurs Synchrones

- Il est possible d'ajouter des entrées de contrôle à un compteur
- Exemple : synthèse d'un compteur/décompteur par 4. Il est ici nécessaire d'ajouter une entrée qui indique si le compteur compte ou le compteur décompte
- Ajout d'une entrée UD , si $UD = 0$ alors le compteur compte sinon le compteur décompte.

Les compteurs Synchrones

- Exemple : synthèse d'un compteur/décompteur par 4 avec entrée UD
 - Nombre de bascules $\lceil \log_2(4) \rceil = 2$. Une bascule D_0 et une bascule D_1 .
 - Table des transitions :

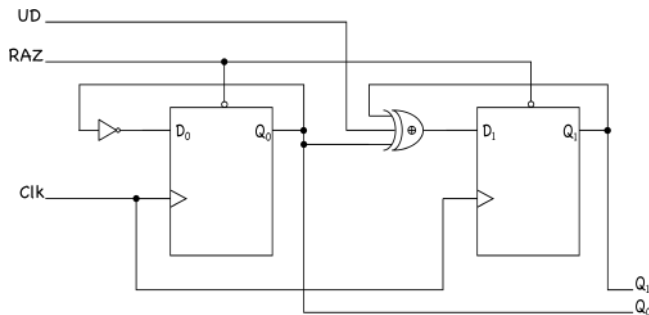
UD	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

Instant n Instant $n+1$

- Equations, il est possible d'utiliser les tableaux de Karnaugh pour obtenir ces équations :
 - ★ $Q_0^{n+1} = \overline{Q_0^n}$
 - ★ $Q_1^{n+1} = UD \oplus Q_1^n \oplus Q_0^n$
- Q_0^{n+1} est l'entrée D de la bascule D_0 , Q_0^n en est sa sortie Q
- Q_1^{n+1} est l'entrée D de la bascule D_1 , Q_1^n en est sa sortie Q

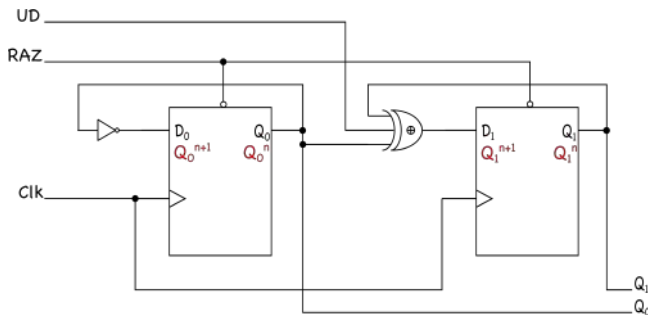
Les compteurs Synchrones

- Synthèse d'un compteur/décompteur synchrone par 4, schéma



Les compteurs Synchrones

- Synthèse d'un compteur/décompteur synchrone par 4, schéma



Les compteurs Synchrones

- Synthèse d'un compteur/décompteur synchrone par 4, description VHDL Structurelle.
Utilisation de bascules avec entrées asynchrones actives à 0.

```
library ieee;
use ieee.std_logic_1164.all;

entity comptdecompt_par_4 is
port (clk, raz, UD : in std_logic;
      q : out std_logic_vector(1 downto 0));
end entity comptdecompt_par_4;

architecture struct of comptdecompt_par_4 is
signal rau : std_logic;
signal dint,qint : std_logic_vector(1 downto 0);
begin
    rau <= '1';
    generebasc : for i in 0 to 1 generate
        basci : entity work.basculer_raz_rau(comport)
            port map(clk=>clk,raz => raz, rau => rau, d => dint(i), q => qint(i));
    end generate generebasc;

    dint(0) <= not(qint(0));
    dint(1) <= UD xor qint(1) xor qint(0);
    q <= qint;
end architecture struct;
```

Les compteurs Synchrones

- Synthèse d'un compteur/décompteur synchrone par 4, description VHDL Comportementale.
- Utilisation du package `ieee.std_logic_unsigned.all`

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity comptdecompt_par_4 is
port( clk, raz, UD : in std_logic;
      q : out std_logic_vector(1 downto 0));
end entity comptdecompt_par_4;

architecture comport of comptdecompt_par_4 is
signal qint : std_logic_vector(1 downto 0);
begin
  UpDown : process(clk, raz, UD) is
  begin
    if (raz = '0') then
      qint <= "00";
    elsif (clk'event and clk = '1') then
      if (UD = '0') then
        qint <= qint + '1';
      else
        qint <= qint - '1';
      end if;
    else
      qint <= qint;
    end if;
  end process UpDown;

  q <= qint;
end architecture comport;
```

Plan

- 1 Introduction : L'électronique numérique à l'aube de 2020 / Méthodes et outils de Conception des systèmes numériques
- 2 Algèbre de Boole
- 3 Codage
- 4 Les composants combinatoire simples
- 5 Les composants combinatoires complexes
- 6 Les composants séquentiels
- 7 Interface avec l'environnement continu : Conversion Analogique vers Numérique et Numérique vers Analogique