

# 2E200 : Electronique Numérique, Combinatoire et Séquentielle

Bertrand Granado

Licence E<sup>2</sup>A

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

## 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 ?



# Plan du Cours

- Introduction
- Algèbre de Boole et Logique Combinatoire
- Fonctions Combinatoires Complexes
- **Eléments séquentiels de base : Les Bascules**

# Les éléments séquentiels de base

- Éléments de base

# Les éléments séquentiels de base

- Éléments de base
- Régulation du flux des données

# Les éléments séquentiels de base

- Éléments de base
- Régulation du flux des données
- **Fonction Mémorisation**

# Les éléments séquentiels de base

- Éléments de base
- Régulation du flux des données
- Fonction Mémorisation
- Éléments Asynchrones

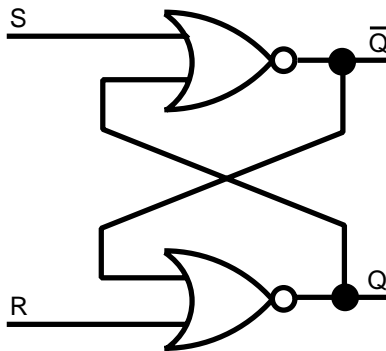
# Les éléments séquentiels de base

- Éléments de base
- Régulation du flux des données
- Fonction Mémorisation
- Éléments Asynchrones
- Éléments Synchrones

# Éléments Asynchrones

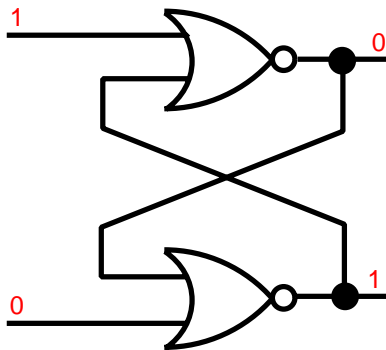
## Les Bascules Asynchrones

# Le bascule RS

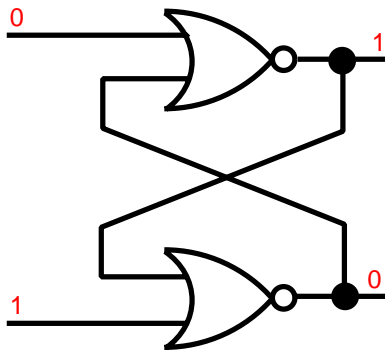




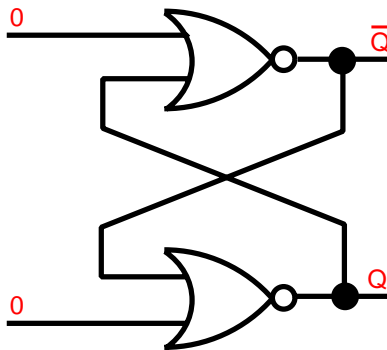
# Le bascule RS



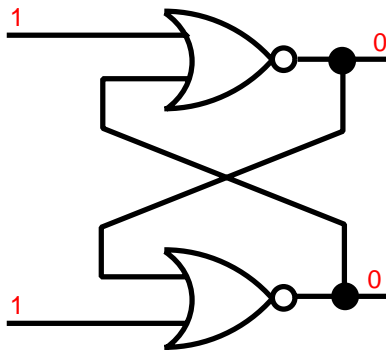
# Le bascule RS



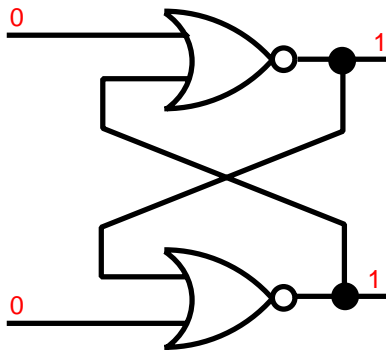
# Le bascule RS



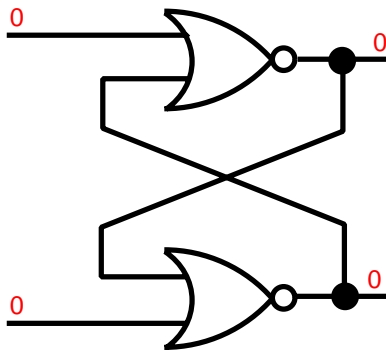
# Le bascule RS



# Le bascule RS



# Le bascule RS



## Le bascule RS

- Table de Vérité

# Le bascule RS

- Table de Vérité

$R$	$S$	$Q$	$\overline{Q}$
0	0	$Q$	$\overline{Q}$
0	1	1	0
1	0	0	1
1	1	Etat	Interdit



# Le bascule RS

- Table de Vérité

$R$	$S$	$Q$	$\overline{Q}$
0	0	$Q$	$\overline{Q}$
0	1	1	0
1	0	0	1
1	1	Etat	Interdit

- Elément Asynchrone

# Le bascule RS

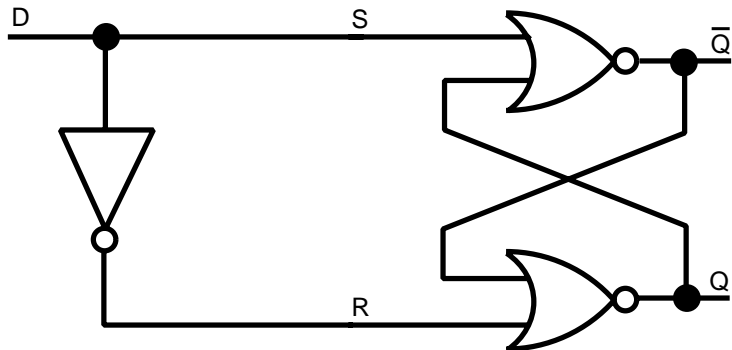
- Table de Vérité

$R$	$S$	$Q$	$\overline{Q}$
0	0	$Q$	$\overline{Q}$
0	1	1	0
1	0	0	1
1	1	Etat	Interdit

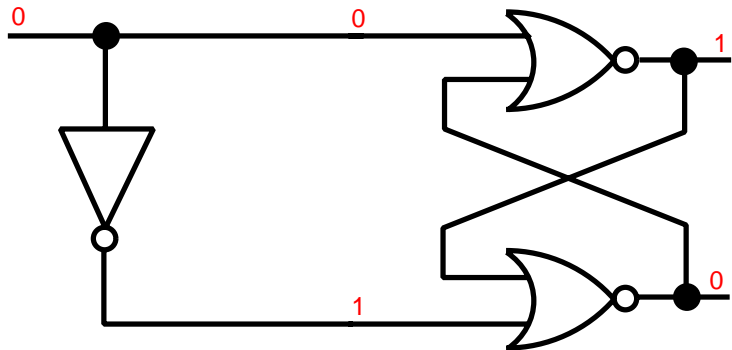
- Elément Asynchrone

- Base de toutes les bascules

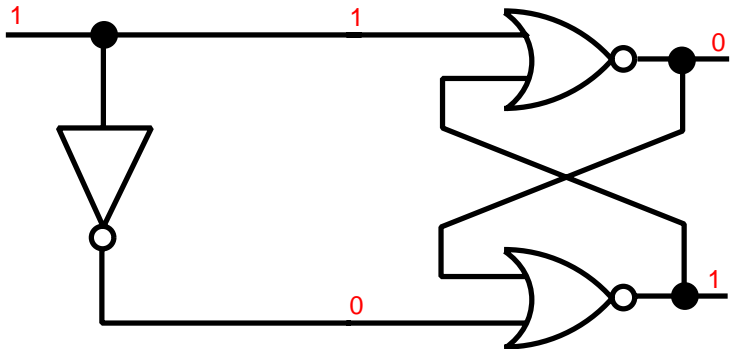
# La bascule D



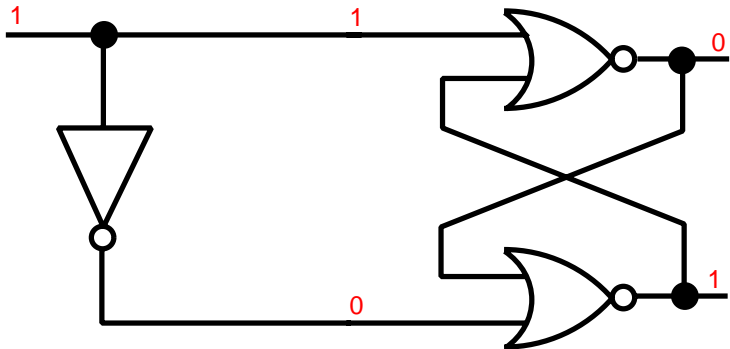
# La bascule D



# La bascule D

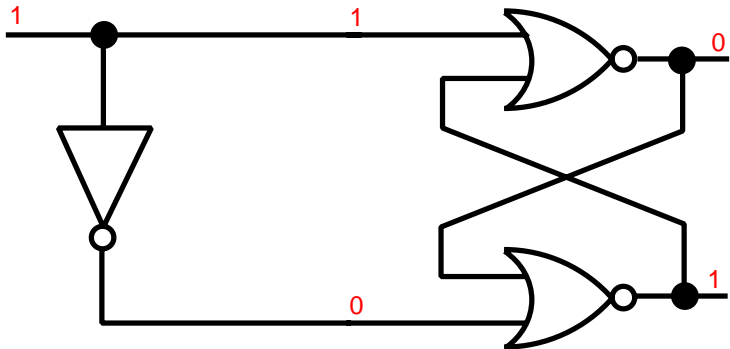


# La bascule D



- Plus d'état logiquement incohérent  $R=S=1$

# La bascule D

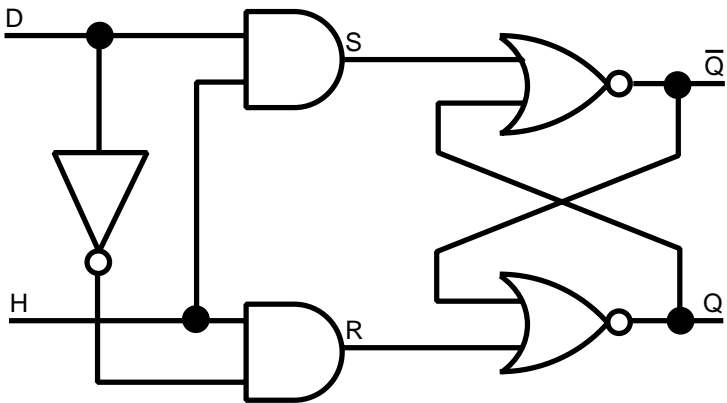


- Plus d'état logiquement incohérent  $R=S=1$
- Pas forcément intéressant, circuit qui réalise la fonction identité

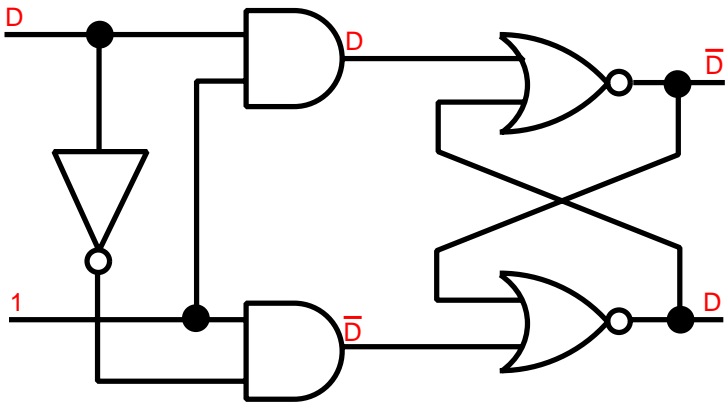
## Bascules Synchrones



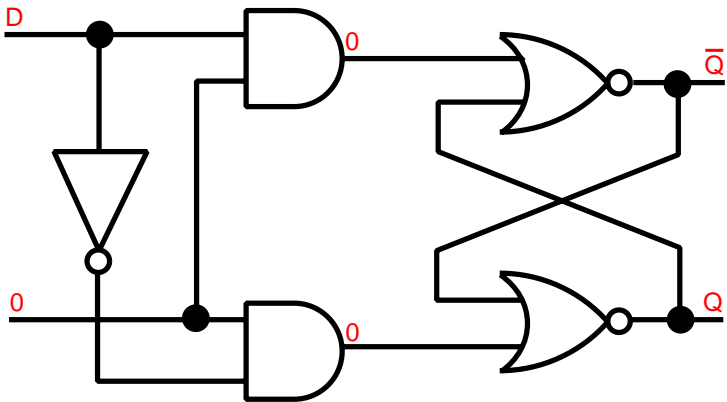
## La bascule D active sur niveau



# La bascule D active sur niveau

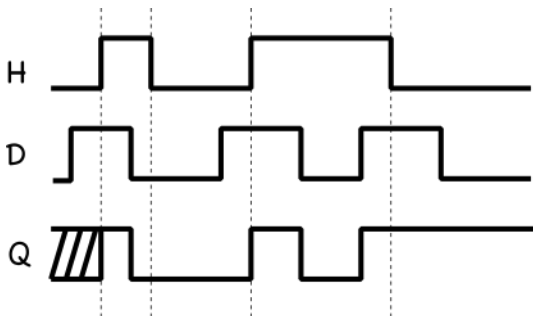


# La bascule D active sur niveau



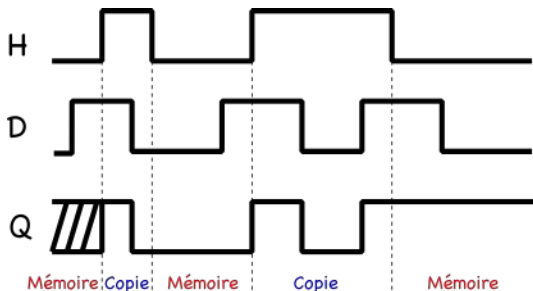
# La bascule D active sur niveau

- Evolution temporelle de Q en fonction du signal de commande H et de l'entrée D de la bascule



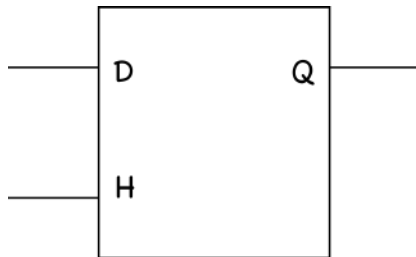
# La bascule D active sur niveau

- Evolution temporelle de Q en fonction du signal de commande H et de l'entrée D de la bascule



# La bascule D active sur niveau

- Cette bascule est aussi appelée Verrou ou Latch (Anglais)
- Symbole de la bascule D active sur niveau



# La bascule D active sur niveau

- Table de Vérité

$D$	$H$	$Q_{n+1}$	$\overline{Q_{n+1}}$
0	0	$Q_n$	$\overline{Q_n}$
0	1	0	1
1	0	$Q_n$	$\overline{Q_n}$
1	1	1	0

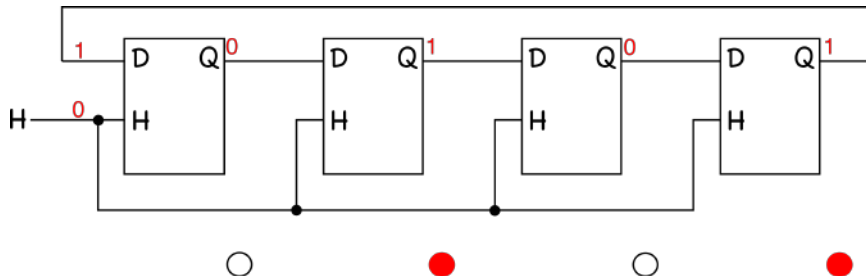
# La bascule D active sur niveau

- Table de Vérité

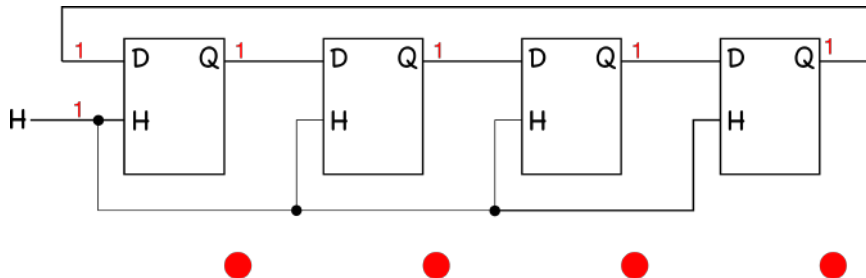
$D$	$H$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	0	$Q_n$	$\overline{Q_n}$
0	1	0	1
1	1	1	0



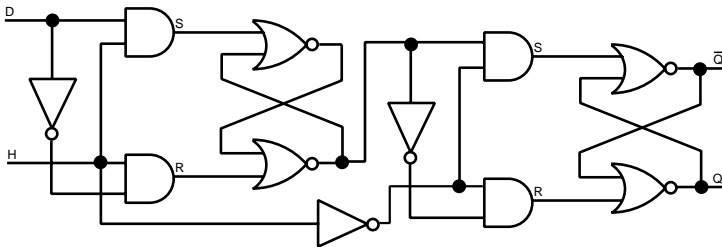
# La bascule D active sur niveau : chenillar



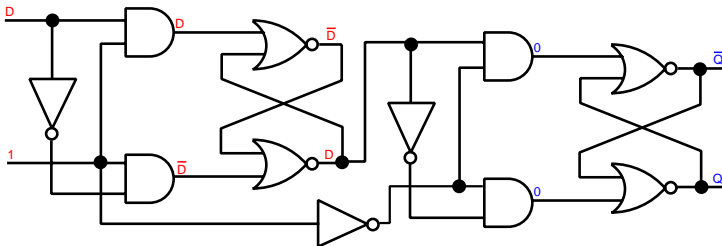
# La bascule D active sur niveau : chenillar



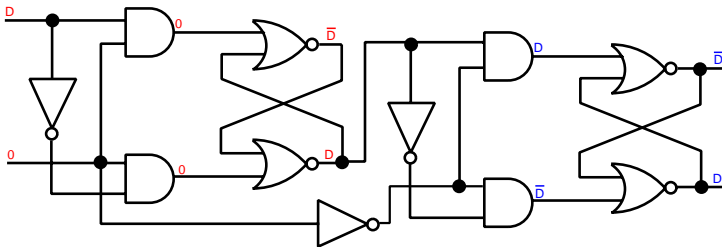
## La bascule D active sur front



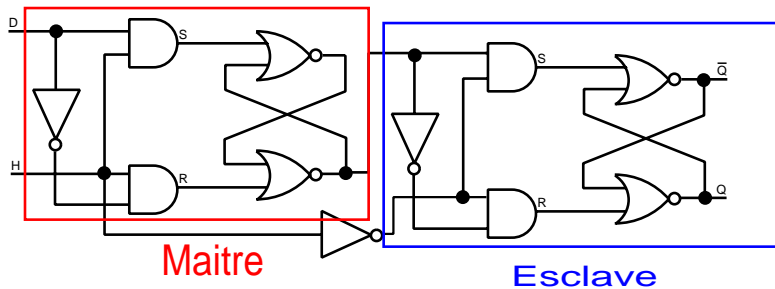
# La bascule D active sur front



# La bascule D active sur front



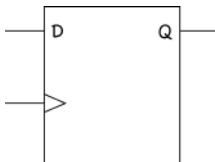
# La bascule D active sur front



## Architecture Maître-Esclave

# La bascule D active sur front

- Symbole de la bascule D active sur front
- L'horloge et le front sont symbolisés par un triangle





# La bascule D active sur front

- Table de Vérité

$D$	$H$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	0	$Q_n$	$\overline{Q_n}$
X	1	$Q_n$	$\overline{Q_n}$
0	$\uparrow$	0	1
1	$\uparrow$	1	0

- Front Montant

# La bascule D active sur front

- Table de Vérité

$D$	$H$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	0	$Q_n$	$\overline{Q_n}$
X	1	$Q_n$	$\overline{Q_n}$
0	$\uparrow$	0	1
1	$\uparrow$	1	0

- Front Montant

- Front descendant

$D$	$H$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	0	$Q_n$	$\overline{Q_n}$
X	1	$Q_n$	$\overline{Q_n}$
0	$\downarrow$	0	1
1	$\downarrow$	1	0

# La bascule D active sur front

- Table de Vérité

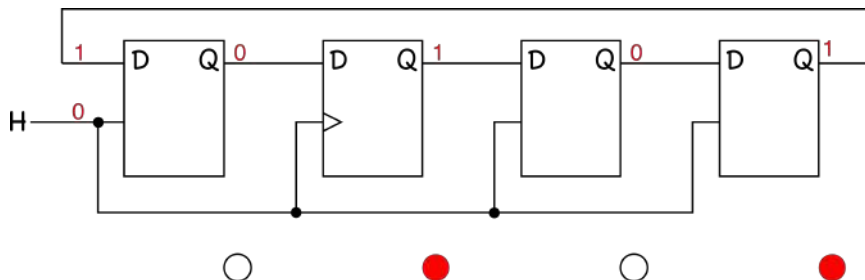
$D$	$H$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	0	$Q_n$	$\overline{Q_n}$
X	1	$Q_n$	$\overline{Q_n}$
0	$\uparrow$	0	1
1	$\uparrow$	1	0

- Front Montant

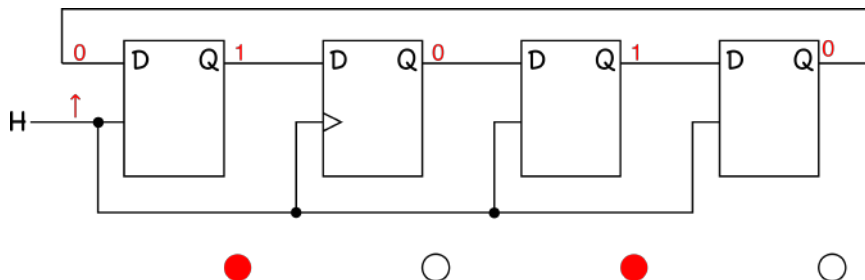
- Front descendant

$D$	$H$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	0	$Q_n$	$\overline{Q_n}$
X	1	$Q_n$	$\overline{Q_n}$
0	$\downarrow$	0	1
1	$\downarrow$	1	0

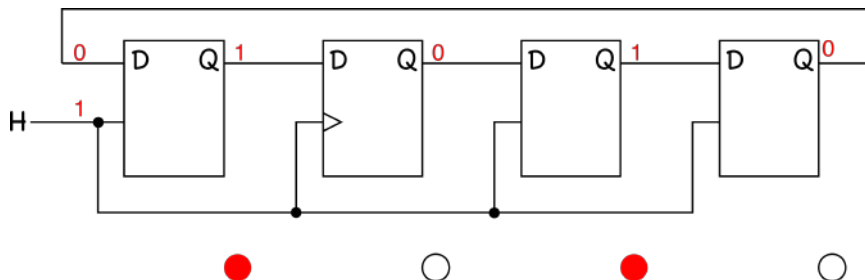
# La bascule D active sur front : chenillar



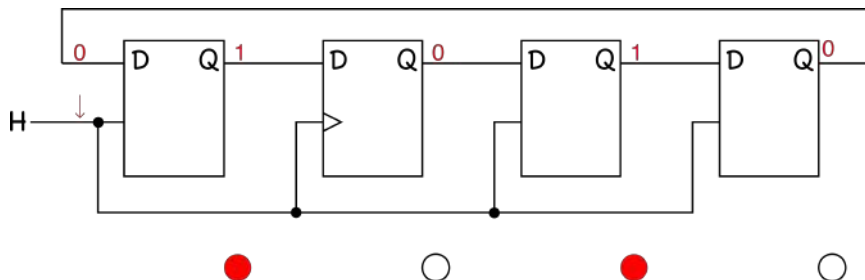
# La bascule D active sur front : chenillar



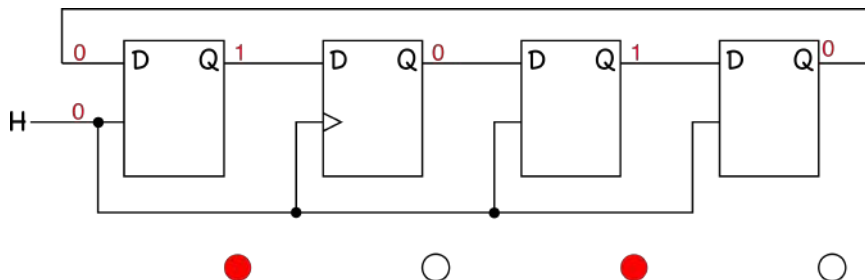
# La bascule D active sur front : chenillar



# La bascule D active sur front : chenillar

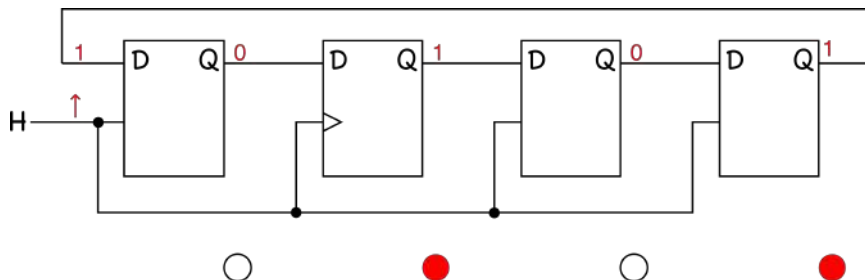


# La bascule D active sur front : chenillar

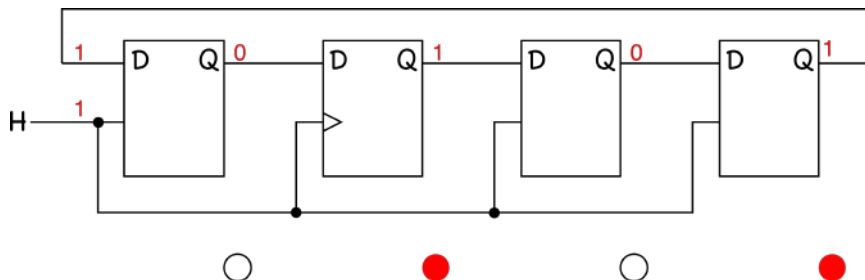




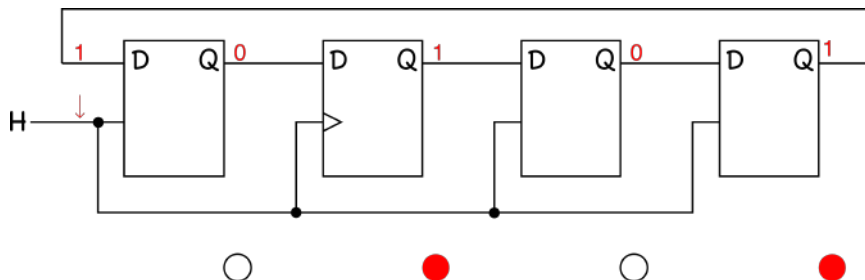
# La bascule D active sur front : chenillar



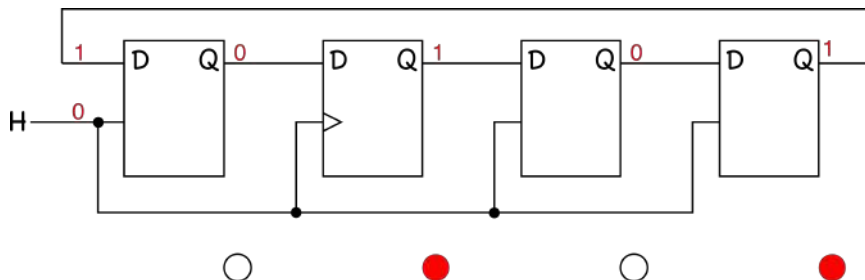
# La bascule D active sur front : chenillar



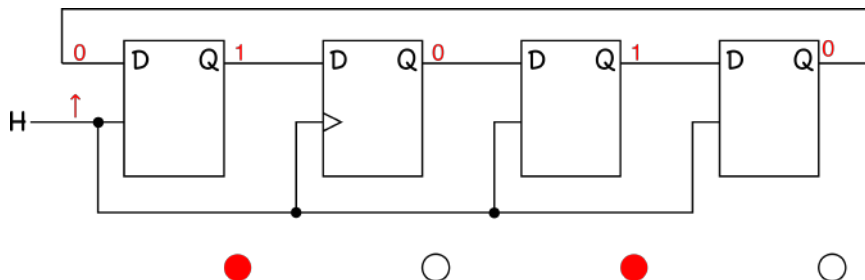
# La bascule D active sur front : chenillar



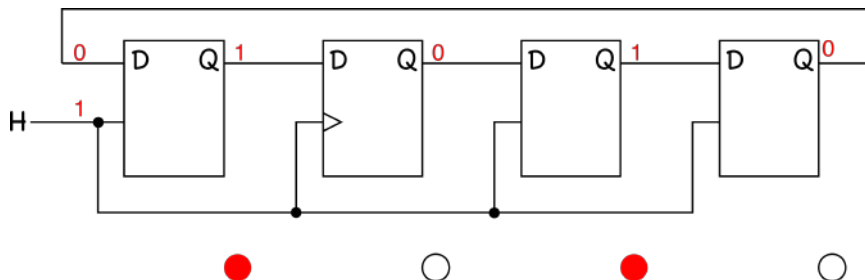
# La bascule D active sur front : chenillar



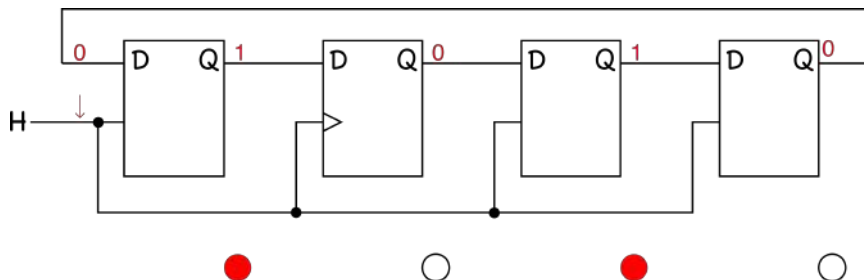
# La bascule D active sur front : chenillar



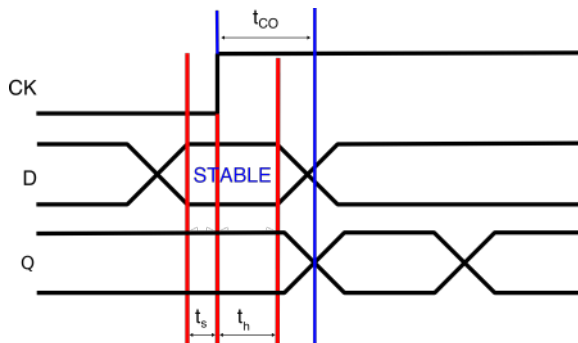
# La bascule D active sur front : chenillar



# La bascule D active sur front : chenillar

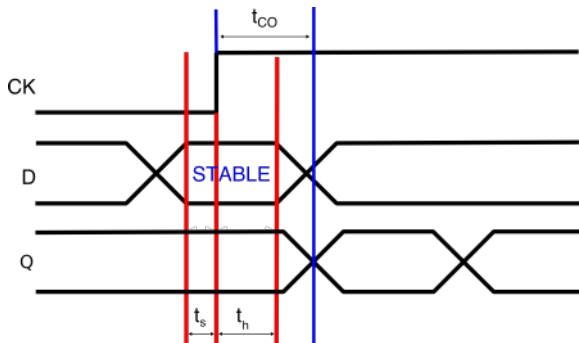


# La bascule D active sur front : Considérations Temporelles



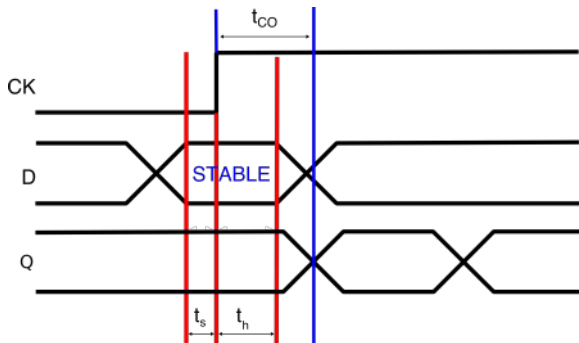


# La bascule D active sur front : Considérations Temporelles



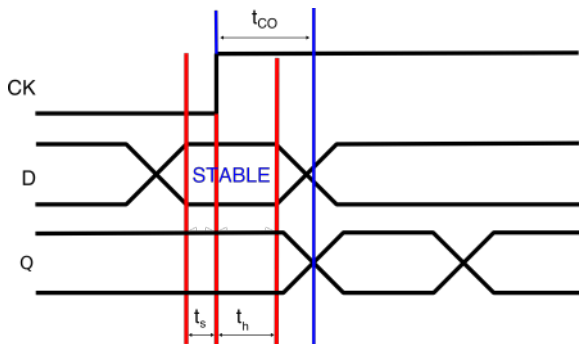
- $t_s$  est le temps de prépositionnement (*setup en anglais*)

# La bascule D active sur front : Considérations Temporelles



- $t_s$  est le temps de prépositionnement (*setup en anglais*)
- $t_h$  est le temps de maintien (*hold en anglais*)

# La bascule D active sur front : Considérations Temporelles



- $t_s$  est le temps de prépositionnement (*setup en anglais*)
- $t_h$  est le temps de maintien (*hold en anglais*)
- $t_{CO}$  est le temps de propagation de la bascule D (*CO : Clock to Output*)

# Vhdl : Bascule D active sur front

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

entity bascule is
  port ( d, clk : in std_logic;
        q : out std_logic );
end entity bascule;

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

# Entrées Asynchrones

- Nécessité de forcer les sorties



# Entrées Asynchrones

- Nécessité de forcer les sorties
- Ajout d'entrées asynchrones
- RAU : Remise à Un ou entrée *Set* en anglais

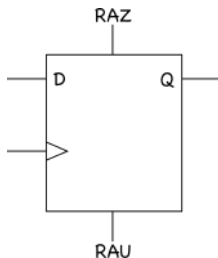
# Entrées Asynchrones

- Nécessité de forcer les sorties
- Ajout d'entrées asynchrones
- RAU : Remise à Un ou entrée *Set* en anglais
- RAZ : Remise à Zéro ou entrée *Reset* en anglais



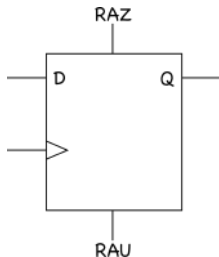
# Entrées Asynchrones

- Entrées Asynchrones active à 1



# Entrées Asynchrones

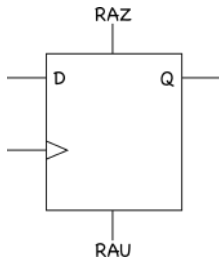
- Entrées Asynchrones active à 1



$D$	$H$	$RAU$	$RAZ$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	X	1	0	1	0
X	X	0	1	0	1
X	0	0	0	$Q_n$	$\overline{Q_n}$
X	1	0	0	$Q_n$	$\overline{Q_n}$
0	$\uparrow$	0	0	0	1
1	$\uparrow$	0	0	1	0

# Entrées Asynchrones

- Entrées Asynchrones active à 1



$D$	$H$	$RAU$	$RAZ$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	X	1	0	1	0
X	X	0	1	0	1
X	0	0	0	$Q_n$	$\overline{Q_n}$
X	1	0	0	$Q_n$	$\overline{Q_n}$
0	$\uparrow$	0	0	0	1
1	$\uparrow$	0	0	1	0

- Combinaison  $RAU=RAZ=1$  interdite

# Vhdl : Bascule D active sur front avec entrées asynchrones

- Entrées RAU et RAZ actives à 1

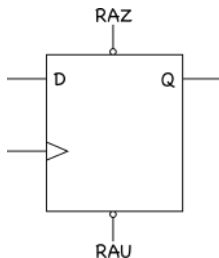
```
library ieee;
use ieee.std_logic_1164.all;

entity bascule is
  port ( d, clk, rau, raz : in std_logic;
        q : out std_logic );
end entity bascule;

architecture comport of bascule is
begin
  stockage : process(d,clk) is
  begin
    if (rau = '1' ) then
      q <= '1';
    elsif (raz = '1') then
      q <= '0';
    elsif clk='1' and clk'event then
      q <= d;
    end if;
  end process stockage;
end architecture comport;
```

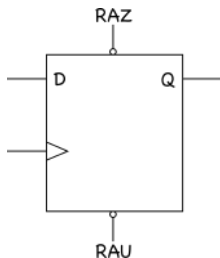
# Entrées Asynchrones

- Entrées Asynchrones active à 0



# Entrées Asynchrones

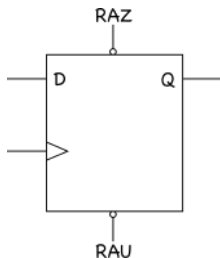
- Entrées Asynchrones active à 0



$D$	$H$	$RAU$	$RAZ$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	X	0	1	1	0
X	X	1	0	0	1
X	0	1	1	$Q_n$	$\overline{Q_n}$
X	1	1	1	$Q_n$	$\overline{Q_n}$
0	$\uparrow$	1	1	0	1
1	$\uparrow$	1	1	1	0

# Entrées Asynchrones

- Entrées Asynchrones active à 0



$D$	$H$	$RAU$	$RAZ$	$Q_{n+1}$	$\overline{Q_{n+1}}$
X	X	0	1	1	0
X	X	1	0	0	1
X	0	1	1	$Q_n$	$\overline{Q_n}$
X	1	1	1	$Q_n$	$\overline{Q_n}$
0	$\uparrow$	1	1	0	1
1	$\uparrow$	1	1	1	0

- Combinaison  $RAU=RAZ=0$  interdite

# Vhdl : Bascule D active sur front avec entrées asynchrones

- Entrées RAU et RAZ actives à 0

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

entity bascule is
    port ( d, clk, rau, raz : in std_logic;
          q : out std_logic );
end entity bascule;

architecture comport of bascule is
begin
    stockage : process(d,clk) is
    begin
        if (rau = '0' ) then
            q <= '1';
        elsif (raz = '0') then
            q <= '0';
        elsif clk='1' and clk'event then
            q <= d;
        end if;
    end process stockage;
end architecture comport;
```



# Autres Bascules ?

Il existe d'autres types de bascules

- Les bascules JK (Joker and King) : tombé en désuétude
- Les bascules T (Toggle) : c'est une bascule qui change d'état après chaque front d'horloge. Réalisable avec une bascule D où l'on réinjecte la sortie  $\overline{Q}$  sur l'entrée D

## 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