

# UE S1.2 - Ingénierie : Électronique Logique

Version du 6 novembre 2024

Jérôme Mathé

Département de Physique et Laboratoire LAMBE - Univ Évry val d'Essonne

Semestre 1



- 1 Informations générales
- 2 Compter
- 3 Codage d'informations
- 4 Logique combinatoire
- 5 Logique séquentielle



# Informations générales

- Calendrier
- Bibliographie
- Attendus du cours



# Calendrier

12h - 6 séances

|   |     |                                       |                 |
|---|-----|---------------------------------------|-----------------|
| 1 | CM1 | Numération binaire et Portes logiques | 06 Nov          |
| 2 | SP1 | TP Logique Combinatoire               | 14 Nov          |
| 3 | CM2 | Logique séquentielle                  | 20 Nov          |
| 4 | SP2 | Fonctions logiques                    | 27 Nov et 4 Déc |

## Évaluation

Evaluation de type III (2 TP)

Evaluation de type II (en début de séance le 20/11)

Evaluation de type I (le 07 Jan matin)



# Bibliographie

Sources d'inspiration de ce cours et informations supplémentaires.

**Cours** Electronique numérique, Etienne Messerli et Yves Meyer - Hes.so  
<http://reds.heig-vd.ch/share/cours/Manuel/>

**Cours** Processeurs et Architectures numériques. Tarik Graba, Telecom ParisTech  
<http://perso.telecom-paristech.fr/%7Egraba/>

**Cours** Tout est numérique, Henri Benisty,  
<http://paristech.institutoptique.fr/site.php?id=89>



# Attendus de ce cours

Vous serez capables de :

- Comprendre les fondements de la numération binaire.
- Comprendre les portes logiques. (table de vérités, combinaisons..)
- Comprendre les fonction avancée de style bascule, compteur etc.
- Concevoir et mettre en œuvre un circuit électronique logique à base de portes et de bascules.
- Vous serez par ailleurs capables de lire une documentation de composant électronique.



# Compter et recompter



# Énumération

Lorsqu'on compte, on passe d'abord par tous les chiffres dans l'ordre.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Arrivé au dernier (9), on n'a plus de symbole pour représenter le nombre suivant. On incrémente un deuxième signe puis on recommence avec le premier.

10, 11, 12, 13, 14, 15, 16, 17, 18, 19

et ainsi de suite : 20, 21, 22, ...





# Additions

Dans le système décimal on compte par paquet de 10. Pour sommer deux quantités, on somme par chiffre.

$$\begin{array}{r}
 1532 \\
 + 4251 \\
 \hline
 = 5783
 \end{array}$$

Le système d'incrémenter du chiffre suivant à chaque paquet de 10 est directement appliqué pour poser une addition.

$$\begin{array}{r}
 \overset{1}{1} \overset{1}{5} \overset{1}{8} 7 \\
 + 4 5 6 5 \\
 \hline
 = 6 1 5 2
 \end{array}$$



# Pourquoi compter en système binaire

Circuits numériques, tension et courant à deux valeurs différentes : soit il y en a soit il n'y en a pas.

S'il y a une tension, la valeur exacte n'a pas d'importance tant que l'on peut distinguer sans ambiguïté la valeur dans laquelle se trouve le circuit.

Ces deux états sont appelés : ON/OFF, T/F, vrai/faux, Oui/Non, haut/bas, 0/1

## Avantages

Circuit plus fiable car moins soumis au bruit et aux interférences

## Inconvénients

- Perte d'information
- Nécessite de nouveaux composants électroniques.



# Écritures Mathématiques

Comme notre langage nous permet de donner un nom à chaque décimale (tout du moins les 4 premières), le nombre 117 se prononce cent-dix-sept et non pas un-un-sept. Ceci nous permet de voir facilement que :

$$117 = 1 \times 100 + 1 \times 10 + 7 \times 1$$

ou encore

$$117 = 1 \times 10^2 + 1 \times 10^1 + 7 \times 10^0$$

Chaque décimale est associée à une puissance de dix. La puissance représente le rang (0 le plus à droite) et 10 est la taille de notre base (le nombre de symboles ou chiffres).



# Écritures Mathématiques

Peut-on déduire l'écriture binaire de 117 ?

$$117 = a \times 2^6 + b \times 2^5 + c \times 2^4 + d \times 2^3 + e \times 2^2 + f \times 2^1 + g \times 2^0$$

avec  $a, b, c, d, e, f, g$  ne peuvent prendre que la valeur 0 ou 1 (base 2).

ou encore

$$117 = a \times 64 + b \times 32 + c \times 16 + d \times 8 + e \times 4 + f \times 2 + g \times 1$$

On trouve **une** fois 64 dans 117. Donc  $a = 1$  et il reste  $117 - 64 = 53$ .

On trouve **une** fois 32 dans 53. Donc  $b = 1$  et il reste  $53 - 32 = 21$ .

On trouve **une** fois 16 dans 21. Donc  $c = 1$  et il reste  $21 - 16 = 5$ .

On trouve **zéro** fois 8 dans 5. Donc  $d = 0$ .

On trouve **une** fois 4 dans 5. Donc  $e = 1$  et il reste  $5 - 4 = 1$ .

On trouve **zéro** fois 2 dans 1. Donc  $f = 0$ .

On trouve **une** fois 1 dans 1. Donc  $g = 1$  et il reste 0.

$$117 = 1110101$$



# Énumération en binaire cette fois

Pour ne pas confondre les notations binaires avec un nombre décimal ne comportant que des 1 et des 0, on écrira plutôt :

$$d117 = b1110101$$

Les décimales en base 10 s'appellent de bits en base 2 (bit est la contraction de *binary digit*)

| Décimal | Binaire |                                    |
|---------|---------|------------------------------------|
| 0       | b0000   | Facile                             |
| 1       | b0001   | Simple                             |
| 2       | b0010   | On incrémente le deuxième bit.     |
| 3       | b0011   | On incrémente                      |
| 4       | b0100   | Les deux premiers bits sont pleins |
| 5       | b0101   |                                    |
| 6       | b0110   |                                    |
| 7       | b0111   |                                    |
| 8       | b1000   |                                    |



# Addition en binaire cette fois

Additionnons deux bits  $A$  et  $B$  :

$$A + B = ?$$

| A | B | A+B |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 1   |
| 1 | 0 | 1   |
| 1 | 1 | 10  |

Le système de retenue s'applique aussi en binaire.  $1 + 1 = 0$  avec retenue de 1.  
Exemple :

|      |   |    |   |   |   |   |
|------|---|----|---|---|---|---|
|      |   | 16 | 8 | 4 | 2 | 1 |
| d11  |   |    | 1 | 0 | 1 | 1 |
| +d07 | + |    | 0 | 1 | 1 | 1 |
| =d18 | = |    | 1 | 0 | 0 | 1 |
|      |   |    | 0 | 1 | 0 |   |



# Codage d'informations et logique combinatoire

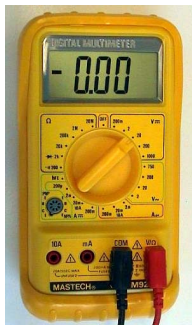


# Codage électrique

Support de codage de l'information : l'électricité

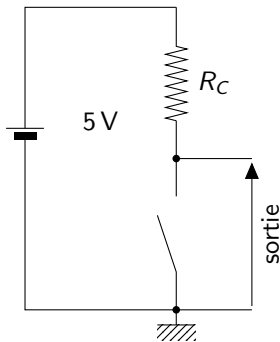
Code binaire moins sensible aux perturbations.

2 états du circuit différents = 2 valeur de tension différente  
Par exemple 0 V et 5 V.





# Codage électrique



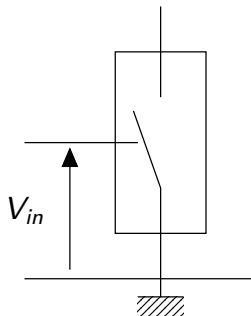
Si l'interrupteur est :

- fermé 0 V en sortie → bit '0'
- ouvert 5 V en sortie → bit '1'

Conversion d'une tension en information binaire  
Nécessité d'un interrupteur **mais que l'on commande électriquement.**



# Codage électrique

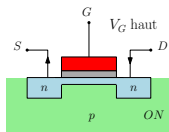
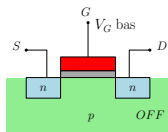
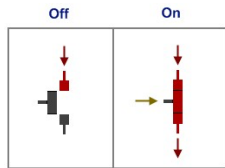


Interrupteur à commande électrique = **Transistor**

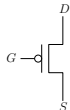
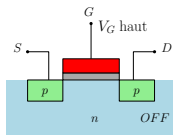
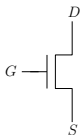
ouvert Si  $V_{in} < V_{ref}$

fermé Si  $V_{in} > V_{ref}$

MOSFET (Metal Oxyde Semiconductor Field Effect Transistor)



NFET



PFET



# Manipuler les booléens

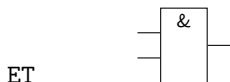
Porte    Symbole européen    Gate    Symbole US



NOT



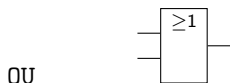
| A | $\bar{A}$ |
|---|-----------|
| 0 | 1         |
| 1 | 0         |



AND



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



OR



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



# Manipuler les booléens

Porte      Symbole européen      Gate      Symbole US

OU excl

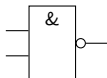


XOR



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

NON-ET



NAND



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



# Autres portes

## Question

Quelle est à votre avis la représentation d'une porte NOR ? Quelle est sa table de vérité ?

| $A$ | $B$ | $A \text{ NOR } B$ |
|-----|-----|--------------------|
| 0   | 0   |                    |
| 0   | 1   |                    |
| 1   | 0   |                    |
| 1   | 1   |                    |

## Question

Une porte XNOR ?

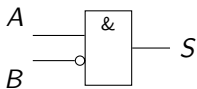
| $A$ | $B$ | $A \text{ XNOR } B$ |
|-----|-----|---------------------|
| 0   | 0   |                     |
| 0   | 1   |                     |
| 1   | 0   |                     |
| 1   | 1   |                     |



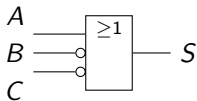
# Combinaison de portes

## Question

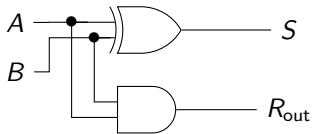
Construire les tables de vérités des opérations suivantes.



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



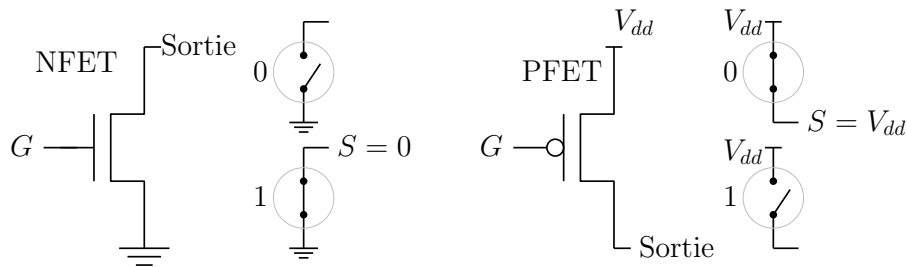
| A | B | C | S |
|---|---|---|---|
| 0 | 0 | 0 |   |
| 0 | 0 | 1 |   |
| 0 | 1 | 0 |   |
| 0 | 1 | 1 |   |
| 1 | 0 | 0 |   |
| 1 | 0 | 1 |   |
| 1 | 1 | 0 |   |
| 1 | 1 | 1 |   |



| A | B | S | $R_{out}$ |
|---|---|---|-----------|
| 0 | 0 |   |           |
| 0 | 1 |   |           |
| 1 | 0 |   |           |
| 1 | 1 |   |           |



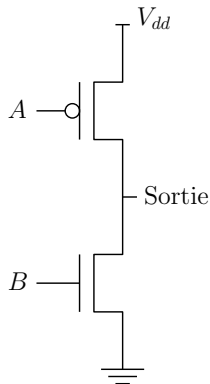
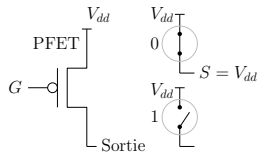
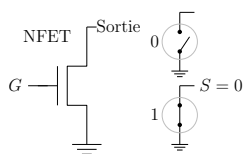
# Portes et CMOS



Transistors à réactions complémentaires : Complementary MOSFET = CMOS



## Portes et CMOS

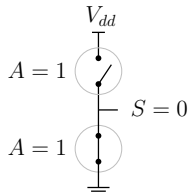
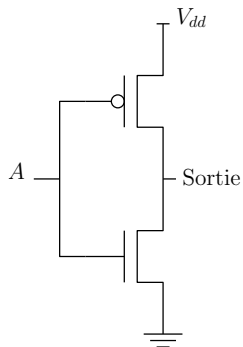
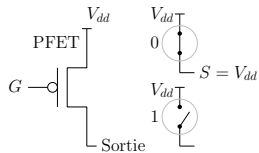
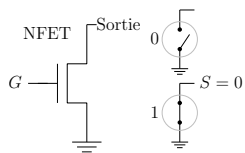


| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 0 | ? |
| 0 | 1 | ? |





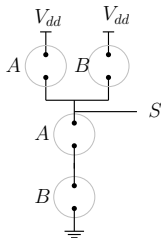
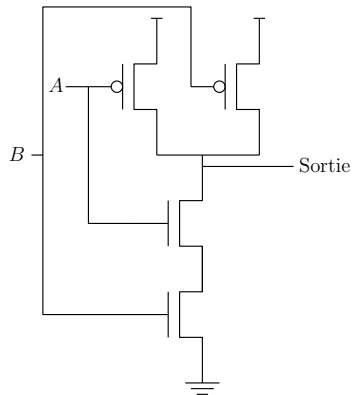
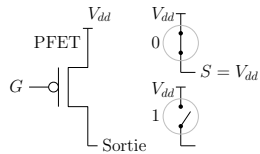
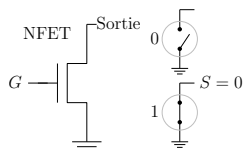
# Porte NON



| A | S |
|---|---|
| 0 | 1 |
| 1 | 0 |



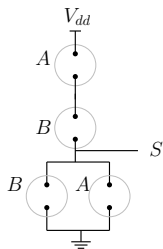
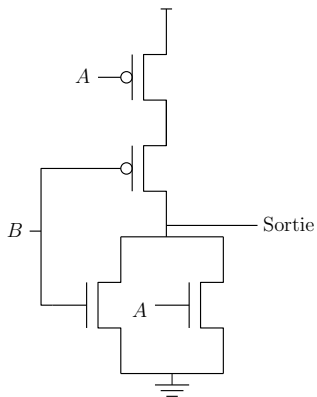
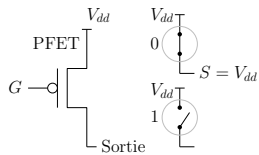
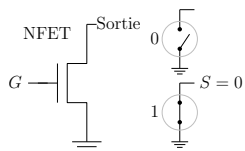
# Porte NAND



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



# Porte NOR



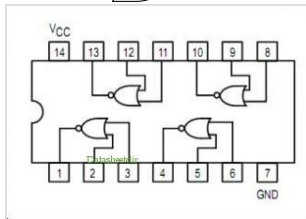
| A | B | S |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



# Portes logiques

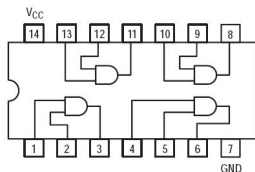
IC 7402

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



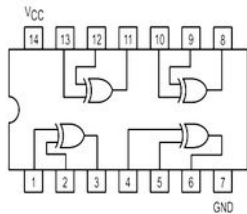
IC 7408

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



IC 7486

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



# Construction de circuits logiques

Retour sur les tables de vérité. Comment faire une addition complète ?

| $R_{in}$ | $A$ | $B$ | $R_{out}$ | $S$ | décimal |
|----------|-----|-----|-----------|-----|---------|
| 0        | 0   | 0   | 0         | 0   | 0       |
| 0        | 0   | 1   | 0         | 1   | 1       |
| 0        | 1   | 0   | 0         | 1   | 1       |
| 0        | 1   | 1   | 1         | 0   | 2       |
| 1        | 0   | 0   | 0         | 1   | 1       |
| 1        | 0   | 1   | 1         | 0   | 2       |
| 1        | 1   | 0   | 1         | 0   | 2       |
| 1        | 1   | 1   | 1         | 1   | 3       |

Table de Karnaugh de  $R_{out}$

| $AB \rightarrow$    | 00 | 01 | 11 | 10 |
|---------------------|----|----|----|----|
| $R_{in} \downarrow$ |    |    |    |    |
| 0                   | 0  | 0  | 1  | 0  |
| 1                   | 0  | 1  | 1  | 1  |

$$R_{out} = (R_{in} \cdot A) + (R_{in} \cdot B) + (A \cdot B)$$

$$= R_{in} \cdot (A \oplus B) + (A \cdot B)$$

Table de Karnaugh de  $S$

| $AB \rightarrow$    | 00 | 01 | 11 | 10 |
|---------------------|----|----|----|----|
| $R_{in} \downarrow$ |    |    |    |    |
| 0                   | 0  | 1  | 0  | 1  |
| 1                   | 1  | 0  | 1  | 0  |

$$S = (A \oplus B) \oplus R_{in} = A \oplus (B \oplus R_{in})$$

# Addition

$$S = (A \oplus B) \oplus R_{in}$$

$$R_{out} = R_{in} \cdot (A \oplus B) + (A \cdot B)$$

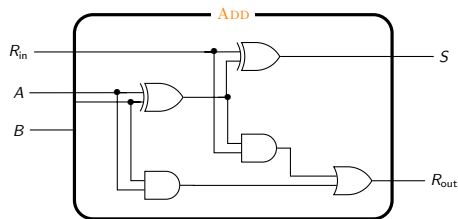
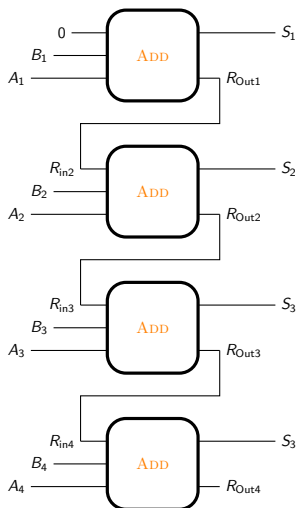


FIGURE – Schéma d'un additionneur 1 bit

| $R_{in}$ | $A$ | $B$ | $R_{out}$ | $S$ | Décimal |
|----------|-----|-----|-----------|-----|---------|
| 0        | 0   | 0   | 0         | 0   | 0       |
| 0        | 0   | 1   | 0         | 1   | 1       |
| 0        | 1   | 0   | 0         | 1   | 1       |
| 0        | 1   | 1   | 1         | 0   | 2       |
| 1        | 0   | 0   | 0         | 1   | 1       |
| 1        | 0   | 1   | 1         | 0   | 2       |
| 1        | 1   | 0   | 1         | 0   | 2       |
| 1        | 1   | 1   | 1         | 1   | 3       |



# Addition



Principe d'un additionneur 4 bits.

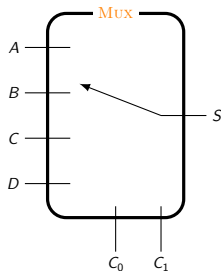
$$S = A + B$$

MSB :  $A_4$

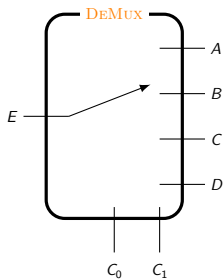
LSB :  $A_0$



# Multiplexage. Demultiplexage



| $C_0$ | $C_1$ | $S$ |
|-------|-------|-----|
| 0     | 0     | A   |
| 0     | 1     | B   |
| 1     | 0     | C   |
| 1     | 1     | D   |



| $C_0$ | $C_1$ | A | B | C | D |
|-------|-------|---|---|---|---|
| 0     | 0     | E | 0 | 0 | 0 |
| 0     | 1     | 0 | E | 0 | 0 |
| 1     | 0     | 0 | 0 | E | 0 |
| 1     | 1     | 0 | 0 | 0 | E |

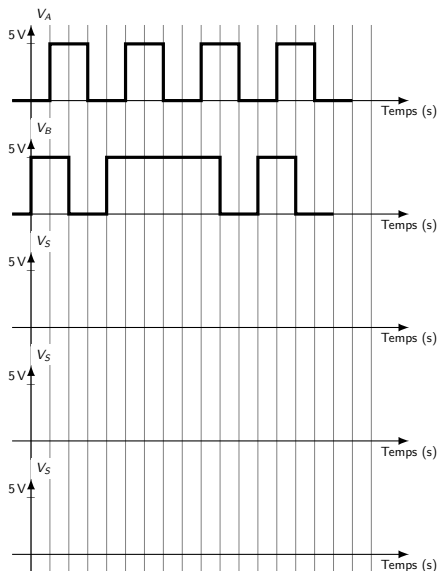
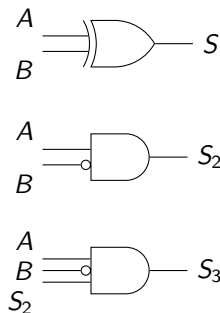




# Logique séquentielle



# Chronogramme

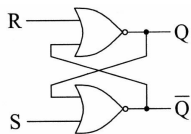


# Les fonctions d'électronique logique séquentielle

- Mise en mémoire
- Etat futur dépend de l'état présent (et des entrées)
- Cadencer des opérations logiques sur une horloge



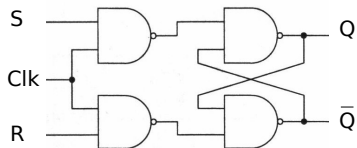
# Verrou



| $R$ | $S$ | $Q$ | $\bar{Q}$ |
|-----|-----|-----|-----------|
| 0   | 0   | $Q$ | $\bar{Q}$ |
| 0   | 1   | 1   | 0         |
| 1   | 0   | 0   | 1         |
| 1   | 1   | ?   | ?         |



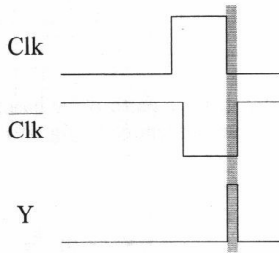
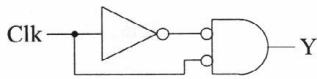
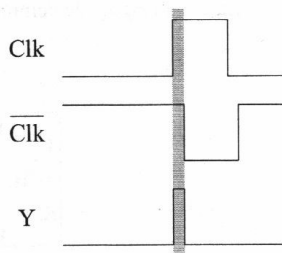
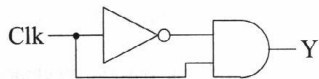
# Bascule RS latch (avec validation)



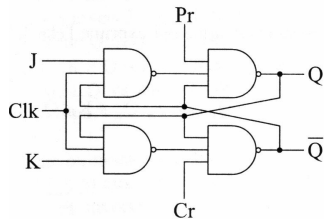
| $R_n$ | $S_n$ | $Q_{n+1}$ | $\bar{Q}_{n+1}$ |
|-------|-------|-----------|-----------------|
| 0     | 0     | $Q_n$     | $\bar{Q}_n$     |
| 0     | 1     | 1         | 0               |
| 1     | 0     | 0         | 1               |
| 1     | 1     | ?         | ?               |



# Detection de front d'horloge



# Bascule JK

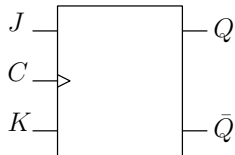


| $J_n$ | $K_n$ | $Q_n$ | $\bar{Q}_n$ | $S$ | $R$ | $\bar{Q}_{n+1}$ |
|-------|-------|-------|-------------|-----|-----|-----------------|
| 0     | 0     | 0     | 1           | 0   | 0   | 0               |
| 0     | 0     | 1     | 0           | 0   | 0   | 1               |
| 0     | 1     | 0     | 1           | 0   | 0   | 0               |
| 0     | 1     | 1     | 0           | 0   | 1   | 0               |
| 1     | 0     | 0     | 1           | 1   | 0   | 1               |
| 1     | 0     | 1     | 0           | 0   | 0   | 1               |
| 1     | 1     | 0     | 1           | 1   | 0   | 1               |
| 1     | 1     | 1     | 0           | 0   | 1   | 0               |

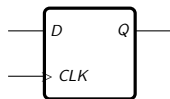
| $J_n$ | $K_n$ | $Q_{n+1}$   |
|-------|-------|-------------|
| 0     | 0     | $Q_n$       |
| 0     | 1     | 0           |
| 1     | 0     | 1           |
| 1     | 1     | $\bar{Q}_n$ |



# Bascule JK et D



| $J_n$ | $K_n$ | $Q_{n+1}$   |
|-------|-------|-------------|
| 0     | 0     | $Q_n$       |
| 0     | 1     | 0           |
| 1     | 0     | 1           |
| 1     | 1     | $\bar{Q}_n$ |



| $D$ | $CLK$ | $Q_{n+1}$ |
|-----|-------|-----------|
| 0   |       | 0         |
| 1   |       | 1         |
| X   | 0     | $Q_n$     |





# Fonction de base : bascule

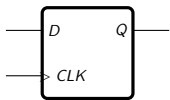


FIGURE – Symbole d'une bascule D

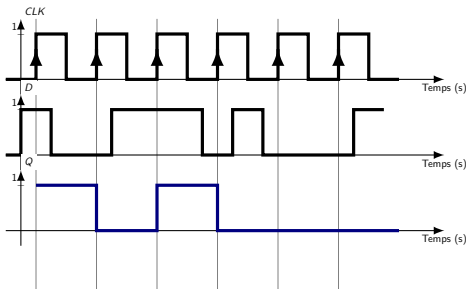
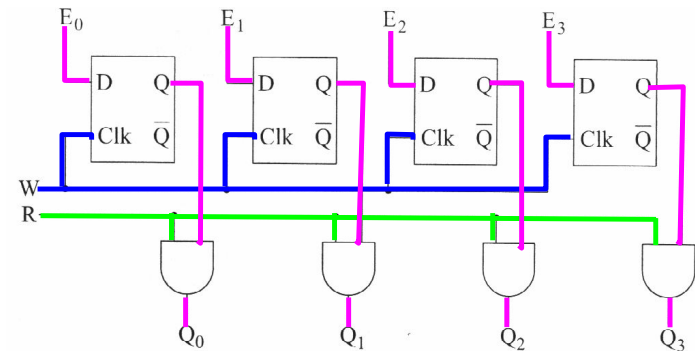


FIGURE – Chronogrammes typiques d'une bascule D

Évolution synchronisée par une horloge

$Q$  recopie  $D$  sur chaque front montant de  $CLK$ .

# Le registre de mémorisation



# Le registre à décalage

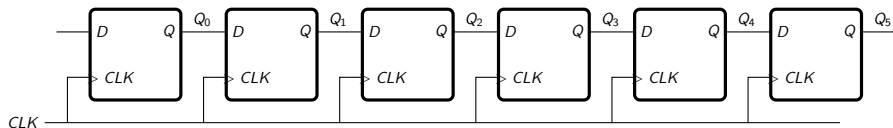
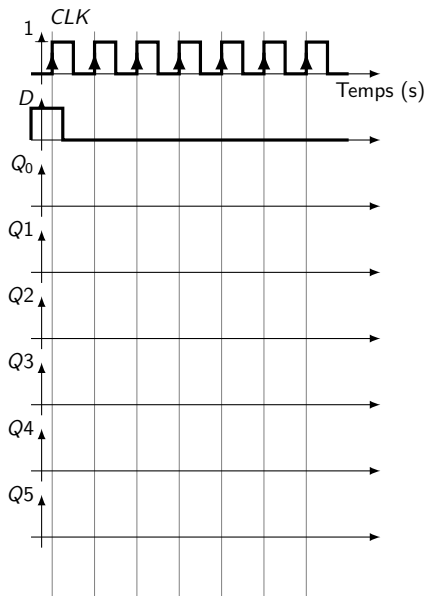


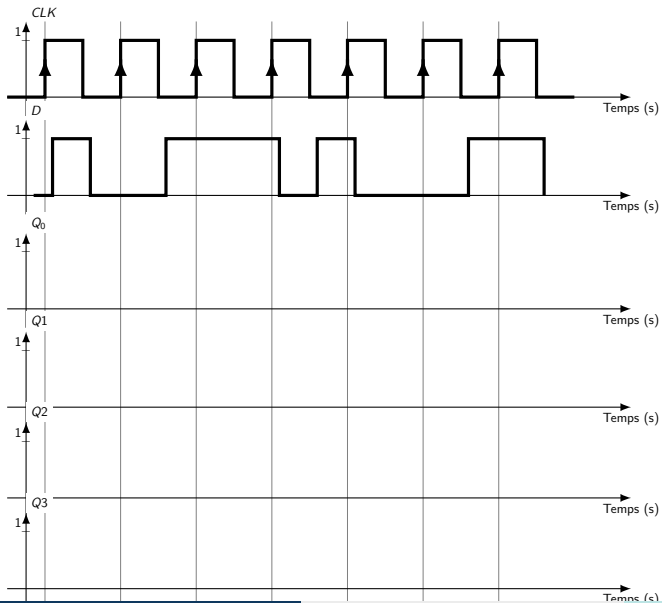
FIGURE – Registre à décalage 6 bits



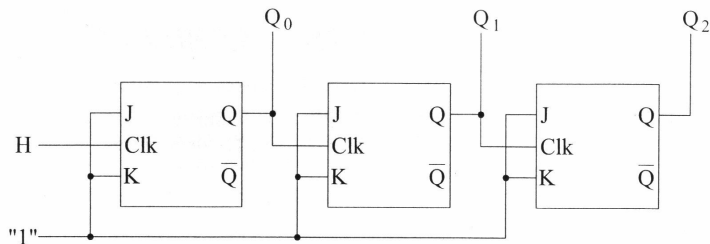
# Le registre à décalage



# Le registre à décalage



# Le compteur



# Séance Pratique 2

## Compréhension du travail demandé

- 1 Comment fonctionne l'afficheur 7 segments ?
- 2 De combien de signaux (tensions électriques) a-t-on besoin ?
- 3 Comment devraient être les chronogrammes d'évolution de ces signaux ?

## Conception du circuit

- 1 Comment fonctionne le composant logique ?
- 2 Combien de signaux peut-il fournir ?
- 3 Combien de signaux d'entrée sont-ils nécessaires ?
- 4 Comment doivent-être le(s) chronogramme(s) du ou des signaux d'entrée ?

## Réalisation et test

- 1 Quel circuit permet de tester de l'afficheur seul ?
- 2 Quel est câblage du composant logique ?
- 3 Le circuit fonctionne-t-il comme attendu ?

# Séance Pratique 2

## Compréhension du travail demandé

- 1 Comment fonctionne l'afficheur 7 segments ?
- 2 De combien de signaux (tensions électriques) a-t-on besoin ?
- 3 Comment devraient être les chronogrammes d'évolution de ces signaux ?

## Conception du circuit

- 1 Comment fonctionne le composant logique ?
- 2 Combien de signaux peut-il fournir ?
- 3 Combien de signaux d'entrée sont-ils nécessaires ?
- 4 Comment doivent-être le(s) chronogramme(s) du ou des signaux d'entrée ?

## Réalisation et test

- 1 Quel circuit permet de tester de l'afficheur seul ?
- 2 Quel est câblage du composant logique ?
- 3 Le circuit fonctionne-t-il comme attendu ?