



Travaux Dirigés

Electronique Numérique

SIN – S1

IUT de Cergy-Pontoise

Département GEII

TD 1

Fonctions logiques

Exercice 1 : Simplifications

Simplifier au maximum les expressions logiques suivantes :

1. $\bar{A} \cdot B + A \cdot B$
2. $(A + B) \cdot (A + \bar{B})$
3. $A + A \cdot B$
4. $A \cdot (A + B)$
5. $A + B \cdot \bar{C} + \bar{A} \cdot \overline{B \cdot \bar{C}} \cdot (A \cdot D + B)$
6. $\bar{A} \cdot \bar{B} + \overline{A + B + C + D}$
7. $(A \oplus B) \cdot B + A \cdot B$
8. $A + \bar{A} \cdot B + \bar{A} \cdot \bar{B}$

Exercice 2 :

Donner la fonction logique correspondant au circuit de la figure 1.

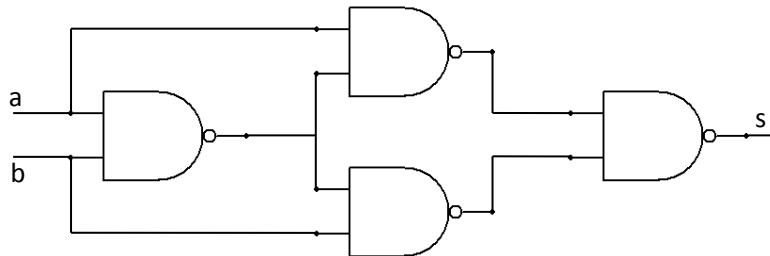


Figure 1

Exercice 3 : Circuits CMOS

- 1) Quelle est la fonction logique correspondant à la figure 2. (porte logique complexe en technologie CMOS) ?

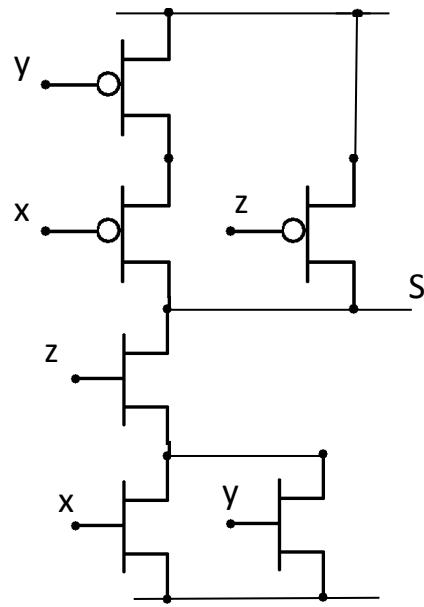


Figure 2 : Porte CMOS complexe

- 2) Donner le schéma avec porte complexe CMOS pour implanter la fonction $s = x \cdot y + z$

TD 2

Simplification des fonctions

Exercice 1 :

Ecrire l'expression $\overline{A \oplus B}$ avec les opérateurs ou, et, non.

Exercice 2 :

- 1) Convertir l'expression $W\bar{X}Y + \bar{X}Y\bar{Z}$ en une somme de produit (sdp) de forme standard.
- 2) Convertir l'expression $(A + \bar{B}).(B + C)$ en un produit de sommes (pds) standard.
- 3) Convertir l'expression $\bar{A}B\bar{C} + AB + AB\bar{C}$ en un pds standard.
- 4) Convertir l'expression $(A + B).\bar{A}.(\bar{A} + \bar{B} + C)$ en une sdp standard.

Exercice 3 :

- 1) Créer une table de vérité pour la sdp standard : $\bar{A}B\bar{C} + A\bar{B}C$
- 2) Créer une table de vérité du pds suivant : $(A + \bar{B} + C).(A + B + \bar{C}).(\bar{A} + \bar{B} + \bar{C})$

Exercice 4 :

Dresser le tableau de Karnaugh des sdp suivantes :

- 1) $BC + \bar{A}\bar{C}$
- 2) $\bar{B}\bar{C} + A\bar{B} + AB\bar{C} + A\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D + A\bar{B}CD$

Exercice 5 :

Utiliser un tableau de Karnaugh pour minimiser les expressions suivantes :

- 1) $A + B\bar{C} + CD$
- 2) $\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + ABCD + ABC\bar{D}$
- 3) $\bar{A}B(\bar{C}.\bar{D} + \bar{C}.D) + AB(\bar{C}.\bar{D} + \bar{C}.D) + A\bar{B}.\bar{C}D$
- 4) $(\bar{A}.\bar{B} + A\bar{B})(CD + C\bar{D})$
- 5) $\bar{A}\bar{B} + A\bar{B} + \bar{C}.\bar{D} + C\bar{D}$
- 6) $(A + \bar{B} + C + \bar{D}).(\bar{A} + B + \bar{C} + D).(\bar{A} + \bar{B} + \bar{C} + \bar{D})$
- 7) $(X + \bar{Y}).(W + \bar{Z}).(\bar{X} + \bar{Y} + \bar{Z}).(W + X + Y + Z)$

TD 3

Conception de circuits

Exercice 1 :

Développez un circuit logique muni de 4 variables d'entrée qui donnera une sortie égale à '1' si et seulement si lorsque uniquement 3 variables d'entrée sont égales à '1'.

Exercice 2 :

Soit un nombre binaire sur 4 bits ($A = A_3A_2A_1A_0$) codé en BCD. Concevoir un circuit logique qui donne une sortie haute quand le nombre binaire est supérieur à 0010 et inférieur à 1010.

Exercice 3 :

On cherche à concevoir un décodeur pour piloter un afficheur 7 segments. La logique de décodage requiert 4 entrées BCD et 7 sorties, soit une pour chaque segment de l'afficheur, comme l'illustre la figure 3.

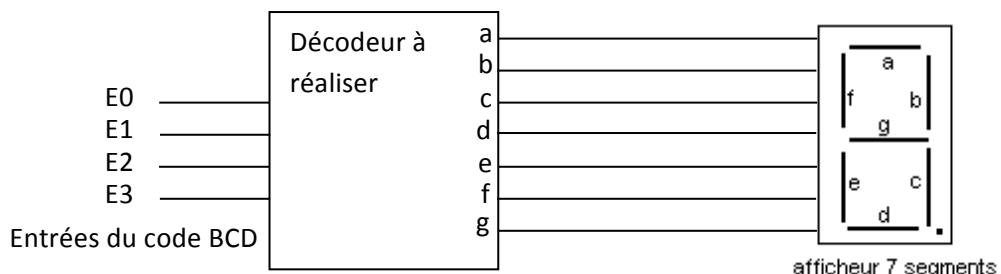


Figure 3

La table de vérité ci-après contient en réalité 7 tables de vérité ; nous pourrions la diviser et obtenir la table de chaque segment. Un '1' placé dans une colonne de sortie des segments de la table indique un segment en état de marché (allumé).

| Nombre décimal | E3 | E2 | E1 | E0 | A | b | c | d | e | f | g |
|----------------|----|----|----|----|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | X | X | X | X | X | X | X |
| 11 | 1 | 0 | 1 | 1 | X | X | X | X | X | X | X |
| 12 | 1 | 1 | 0 | 0 | X | X | X | X | X | X | X |
| 13 | 1 | 1 | 0 | 1 | X | X | X | X | X | X | X |
| 14 | 1 | 1 | 1 | 0 | X | X | X | X | X | X | X |
| 15 | 1 | 1 | 1 | 1 | X | X | X | X | X | X | X |

TD 4

Fonctions combinatoires

Exercice 1 :

- 1) Trouver les équations logiques régissant le fonctionnement du décodeur à 3 entrées (1 parmi 8) dont la table de vérité est résumée dans la figure 4.

| INPUTS | | | | | | OUTPUTS | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| <i>E</i> 1 | <i>E</i> 2 | <i>E</i> 3 | <i>A</i> 0 | <i>A</i> 1 | <i>A</i> 2 | <i>O</i> 0 | <i>O</i> 1 | <i>O</i> 2 | <i>O</i> 3 | <i>O</i> 4 | <i>O</i> 5 | <i>O</i> 6 | <i>O</i> 7 |
| H | X | X | X | X | X | H | H | H | H | H | H | H | H |
| X | H | X | X | X | X | H | H | H | H | H | H | H | H |
| X | X | L | X | X | X | H | H | H | H | H | H | H | H |
| L | L | H | L | L | L | L | H | H | H | H | H | H | H |
| L | L | H | H | L | L | H | L | H | H | H | H | H | H |
| L | L | H | L | H | L | H | H | L | H | H | H | H | H |
| L | L | H | H | H | L | H | H | H | L | H | H | H | H |
| L | L | H | L | L | H | H | H | H | H | L | H | H | H |
| L | L | H | H | L | H | H | H | H | H | H | L | H | H |
| L | L | H | L | H | H | H | H | H | H | H | H | L | H |
| L | L | H | H | H | H | H | H | H | H | H | H | H | L |

H= High voltage Level ; L= Low voltage Level ; X= Don't care

Figure 4 : table de vérité d'un décodeur

- 2) Donner la structure logique d'un tel décodeur

Exercice 2 :

A partir du décodeur conçu à l'exercice 1, concevoir un décodeur 1 parmi 32.

Exercice 3 :

Quelle est la structure logique du codeur dont la table de vérité est fournie dans le tableau ci-dessous :

| Chiffre décimal | A3 | A2 | A1 | A0 |
|-----------------|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |

Exercice 4 :

Dans le cas d'un encodeur prioritaire, si plusieurs lignes d'entrée sont actives simultanément, le résultat correspondant à une seule parmi celles-ci est affiché en sortie. La règle peut être, par exemple, de mettre en sortie le code correspondant au linge d'entrée d'indice le plus élevé. Par exemple, si A7 et A8 sont dans l'état 1, l'encodeur prioritaire donne en sortie le code correspondant à A8.

- 1) En tenant compte des informations précédentes, compléter la table de vérité suivante correspondant à un encodeur de priorité décimal :

| A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | Y3 | Y2 | Y1 | Y0 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | | | | | | | | | | 0 | 0 | 0 | 0 |

- 2) Donner les équations logiques exprimant les Y_i en fonction des variables d'entrée.

Exercice 5 :

Concevoir un transcodeur 4 bits traduisant une entrée codée en binaire vers une sortie exprimée à l'aide du code Gray (table de vérité ci-dessous).

| Code binaire | Code Gray |
|--------------|-----------|
| 0000 | 0000 |
| 0001 | 0001 |
| 0010 | 0011 |
| 0011 | 0010 |
| 0100 | 0110 |
| 0101 | 0111 |
| 0110 | 0101 |
| 0111 | 0100 |
| 1000 | 1100 |
| 1001 | 1101 |
| 1010 | 1111 |
| 1011 | 1110 |
| 1100 | 1010 |
| 1101 | 1011 |
| 1110 | 1001 |
| 1111 | 1000 |

TD 5

Les multiplexeurs

Exercice 1 : Les multiplexeurs

- 1) Trouver les équations logiques régissant le fonctionnement du multiplexeur à 4 entrées plus une entrée de validation active au niveau bas : \overline{val} . Combien faut il d'entrées de sélection ?
- 2) Même question que précédemment mais pour un multiplexeur à 8 entrées plus une entrée de validation \overline{val} supplémentaire.

Exercice 2 :

A l'aide de multiplexeurs 8 entrées définis dans l'exercice 1) , concevoir un multiplexeur à 16 entrées.

Exercice 3 :

- 1) Créer un circuit pour mettre en œuvre la fonction logique spécifiée au tableau suivant en utilisant un multiplexeur à 8 entrées.

| A2 | A1 | A0 | Y |
|----|----|----|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Exercice 4 : Les démultiplexeurs

Déterminer le circuit logique permettant de réaliser un démultiplexeur à une entrée et 4 sorties.

TD 6

Représentation des nombres

Exercice 1 :

Convertir les nombres binaires suivants en décimal, octal, hexadécimal :

0011001 100001 0101101 11010 11111111

Exercice 2 :

Convertir les nombres décimaux suivant en binaire, octal et hexadécimal.

14 25.225 32.625

Exercice 3 :

Déterminer l'équivalent binaire et décimal des nombres hexadécimaux suivants :

- 1) 10A4
- 2) F8E
- 3) 1742

Exercice 4 :

Déterminer la valeur décimale des nombres binaires suivants exprimés selon la notation signe grandeur.

- a) 10010101
- b) 01110111
- c) 11011101

Exercice 5 :

Déterminez la valeur décimale des nombres binaires négatifs suivants exprimés selon la notation en complément à un.

- a) 00010111
- b) 11101000
- c) 0101010
- d) 11111111

Exercice 6 :

Déterminer la valeur décimale des nombres binaires négatifs suivants exprimés selon la notation en complément à deux.

- a) 01010110
- b) 10101010
- c) 00001111
- d) 10000001

Exercice 7 :

Convertissez les nombres suivants en notation complément à 2 (sur 16 bits) :

- a) 145
- b) -153
- c) -49
- d) 225

Exercice 8 :

Effectuez les opérations suivantes avec des nombres entiers signés en complément à 2 :

- 1) 011 + 011
- 2) 100 + 011
- 3) 110 + 100
- 4) 111 – 010
- 5) 011 – 101
- 6) 100 – 111

TD 7

Circuits arithmétiques

Exercice 1 :

Ecrire les équations logiques du comparateur à 1 bit dont on donne la table de vérité dans le tableau suivant :

| AB | S1 | S2 | S3 |
|----|----|----|----|
| 00 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 |
| 10 | 1 | 0 | 0 |
| 11 | 0 | 1 | 0 |

Comparateur 1 bit

Exercice 2 :

Pour comparer des mots binaires de plusieurs bits, il faut disposer d'un comparateur cascadable. Pour cela, on ajoute 3 entrées E1, E2, E3, destinées à recueillir les informations de sortie S1, S2, S3, provenant de la comparaison des bits de plus faible poids.

- 1) Déterminer les équations logiques du comparateur à 1 bit cascadable en complétant la table de vérité donnée dans le tableau suivant :

| AB | E1 | E2 | E3 | S1 | S2 | S3 |
|-----|----|----|----|----|----|----|
| 01 | X | X | X | | | |
| 10 | X | X | X | | | |
| A=B | 0 | 0 | 1 | | | |
| A=B | X | 1 | X | | | |
| A=B | 1 | 0 | 0 | | | |
| A=B | 0 | 0 | 0 | | | |
| A=B | 1 | 0 | 1 | | | |

Comparateur cascadable

- 2) Décrire schématiquement l'architecture d'un comparateur 3 bits à partir de comparateur 1 bit.

Exercice 3 :

Montrer comment peut on transformer un comparateur cascadable 1 bit en un comparateurs 3 bits avec le minimum des portes logiques en plus.

Exercice 4 :

On souhaite afficher le résultat de la comparaison sur un afficheur 7 segments à cathode commune.

Etudier le circuit qui permet d'écrire S(supérieur) si $A > B$, E(égal) si $A = B$ et I(inférieur) si $A < B$.

Exercice 5 :

On souhaite transmettre des mots de 4 bits sur un canal entâché d'erreurs de transmission. Le codage de Hamming permet de corriger une erreur portant sur un seul bit.

Pour cela, au lieu d'émettre le mot binaire de 4 bits (m_1, m_2, m_3, m_4), on va émettre un mot de 7 bits ($k_1, k_2, m_1, k_3, m_2, m_3, m_4$), contenant 3 bits de contrôle k_1, k_2, k_3 , calculés de façon à satisfaire les tests de parité :

$$t_1 = \text{parité}(k_1, m_1, m_2, m_4) = 0$$

$$t_2 = \text{parité}(k_2, m_1, m_3, m_4) = 0$$

$$t_3 = \text{parité}(k_3, m_2, m_3, m_4) = 0$$

$t_1 = 0$ signifie que le mot binaire (k_1, m_1, m_2, m_4) contient un nombre pair de bits au niveau haut.

A la réception, on effectue les mêmes tests de parité sur le mot ($K_1, K_2, M_1, K_3, M_2, M_3, M_4$) reçu.

$$T_1 = \text{parité}(K_1, M_1, M_2, M_4) = 0$$

$$T_2 = \text{parité}(K_2, M_1, M_3, M_4) = 0$$

$$T_3 = \text{parité}(K_3, M_2, M_3, M_4) = 0$$

$(T_3 T_2 T_1) = 000$ indique qu'il n'y a pas d'erreur de transmission ; sinon le mot binaire ($T_3 T_2 T_1$) indique la position du bit erroné, qu'il est alors facile de corriger.

Exemple :

Mot à transmettre : 1101

Mot émis : 1010101

Mot reçu : 1000101

Résultat : $T_3 T_2 T_1 = 011$, il y a une erreur sur le bit N°3.

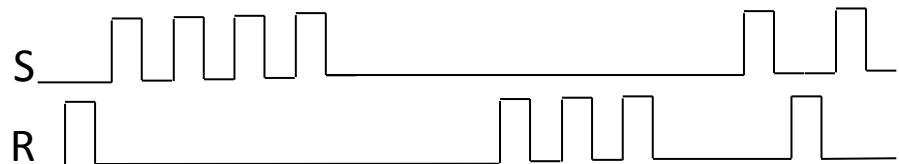
- 1) Représenter un schéma synoptique de la transmission de données, en faisant figurer l'émetteur, le récepteur ainsi que le correcteur et en précisant les entrées/sorties de chaque étage.
- 2) Proposer un schéma pour chacun des 3 modules précédents.

TD 8

Les bascules

Exercice 1 :

Déterminer la forme d'onde des sorties Q et \bar{Q} d'un bistable RS avec des niveaux valides à l'état HAUT.



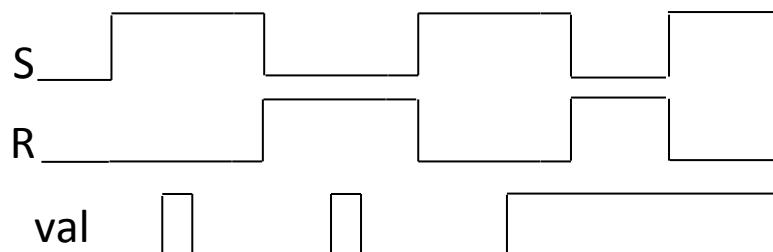
Exercice 2 :

On souhaite concevoir une bascule nommée « clocked RS NAND Latch » (verrou RS en NAND synchronisé sur horloge) possédant en plus des entrées R et S une entrée C servant d'autorisation à niveau HAUT. D'autre part, la fonction de sortie de la bascule est directement synthétisée à l'aide de portes NAND à 2 entrées.

Donner le schéma de cette bascule.

Exercice 3 :

Déterminer la forme d'onde de la sortie Q si les entrées illustrées ce-dessous sont appliquées à un bistable RS à entrée de validation et si son état initial est 0.

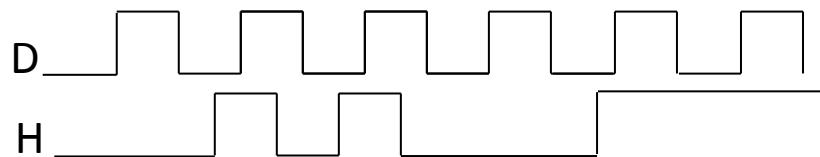


TD 9

Les bascules (2)

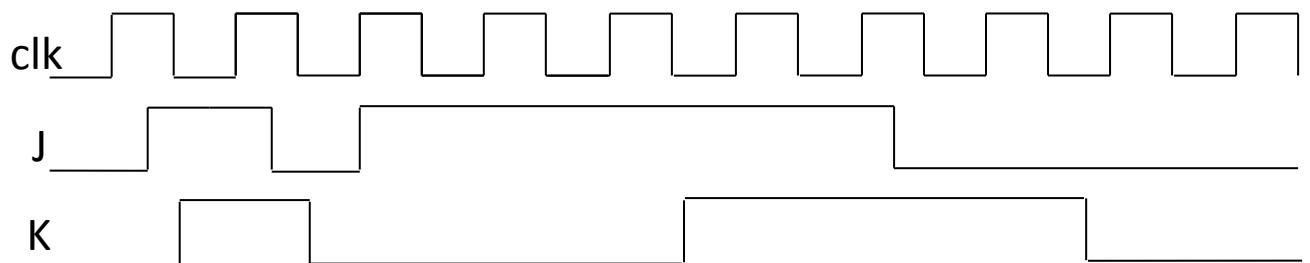
Exercice 1 :

Déterminer la forme d'onde des sorties Q si on applique les entrées illustrées sur la figure ci-dessous à une bascule D latch et si son état initial est 0.



Exercice 2 :

Déterminer la forme d'onde des sorties Q si on applique les entrées illustrées sur la figure ci-dessous à une bascule JK à déclenchement par front positif et si son état initial est 0.



Exercice 3 :

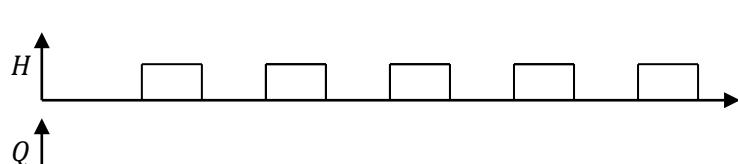
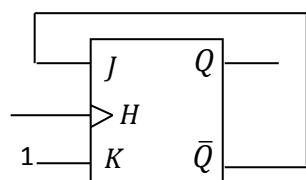
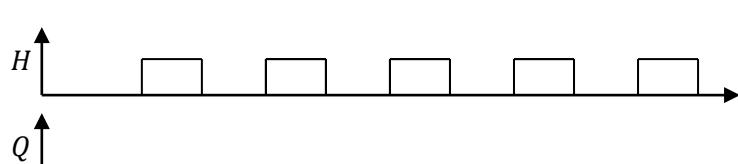
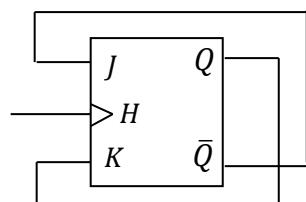
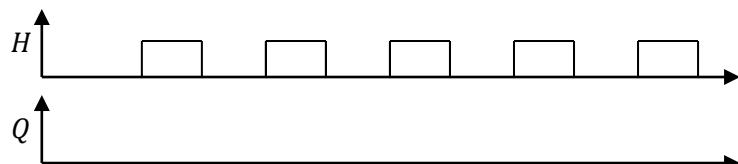
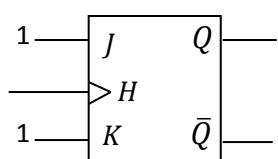
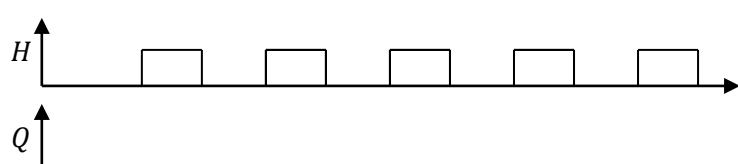
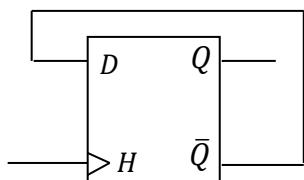
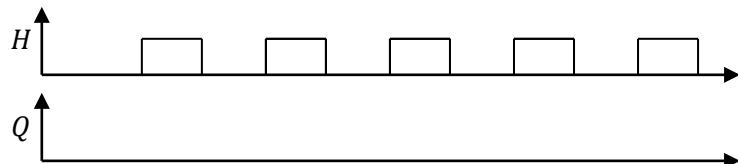
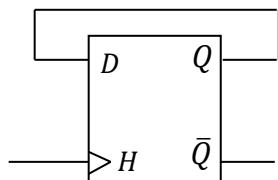
On désire concevoir une bascule JK en utilisant comme base une bascule D avec les entrées synchrones CE, CLK et l'entrée asynchrone CLR. Proposer plusieurs schémas possibles.

TD 10

Logique séquentielle

Exercice 1 :

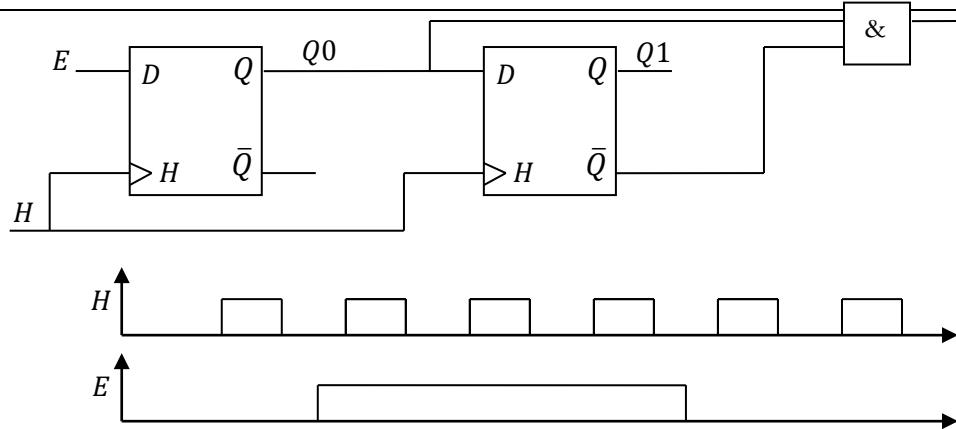
Compléter les chronogrammes pour chacun des schémas suivants :

**Exercice 2 :**

Analyser le fonctionnement du montage de la figure 5.

Exercice 3 :

Illustrer les formes d'ondes d'un compteur en anneau comportant 5 bascules. Présumez que la première bascule est à l'état HAUT.

**Figure 5****Exercice 4 :**

Donner la séquence d'évolution d'un générateur pseudo-aléatoire à 4 bascules D.

Exercice 5 à modifier:

Déterminer les états du registre bidirectionnel de 8 bits de la figure ci-dessous après chaque impulsion de signal d'horloge en réponse à la forme d'onde de commande DROITE/GAUCHE illustrée. Un niveau HAUT sur cette entrée valide un décalage vers la droite, alors qu'un niveau BAS valide un décalage vers la gauche.

Présumez qu'au départ le registre stocke la valeur binaire du nombre décimal 76. La position de droite représente le chiffre de poids le plus faible. Un niveau BAS est présent sur la ligne d'entrée des données.

Exercice 6 :

Concevoir un digicode à 4 chiffres glissants. Ce circuit gère numériquement une serrure en respectant le principe de fonctionnement suivant :

L'utilisateur dispose d'un clavier sur lequel il peut saisir un chiffre hexadécimal (0 à F). La serrure ne s'ouvre qu'à partir du moment où l'utilisateur a tapé, dans l'ordre, les 5 chiffres corrects du code.

En cas d'erreur de frappe, l'utilisateur peut directement recommencer une nouvelle saisie du code, à la suite des chiffres déjà saisis.

L'interface est donc minimale : l'entrée du circuit est le code en provenance du clavier (clavier[3 :0]) combiné avec l'information qu'un nouveau chiffre vient d'être saisi (CHIFFRE_OK) et la sortie est la commande d'ouverture de la serrure (OUVRIR). Un bouton ENREG (Enregistrement) dans une zone sécurisée (et inaccessible aux utilisateurs standards) permet la mémorisation d'un nouveau code.

TD 11

Les compteurs

Exercice 1 :

Réaliser un compteur binaire asynchrone modulo-8 utilisant des bascules D. On considérera que les bascules sont actives sur front montant.

Exercice 2 :

Réaliser un compteur binaire asynchrone modulo 5 utilisant :

- a) Des bascules D
- b) Des bascules JK

Exercice 3 :

Réaliser un compteur binaire synchrone modulo 5 utilisant des bascules JK.

Exercice 4 :

Développer un compteur synchrone réversible de 3 bits afin de produire une séquence de code Gray. Le compteur doit exécuter un comptage lorsque l'entrée de commande $COMPTAGE/DECOMPTAGE$ vaut 1 et un décomptage lorsque celle-ci vaut 0.

Exercice 5 :

On considère les compteurs suivants :

- Le compteur A est un compteur synchrone intégré modulo 5.
- Le compteur B est un compteur synchrone intégré modulo 2.
- Le compteur C est un compteur synchrone intégré modulo 3.

Chaque compteur dispose :

- D'une entrée de validation de comptage EN,
- D'une entrée de remise à zéro, \overline{CLR} ,
- D'une sortie MAX de détection du maximum du compteur.

Donner le schéma bloc du système réalisant un compteur BCD synchrone de 0 à 29.

Exercice 6 :

On cherche à réaliser un circuit permettant d'implanter une horloge numérique comptant les heures, les minutes et les secondes, et affichant ces valeurs sur des afficheurs 7-segments. Le circuit dispose d'une horloge de 50Hz disponible sur l'une de ces entrées.

Donner le schéma de principe d'un tel système.