Séquence 3 : Exercices

**Seq3.Exercice 1 : Circuit combinatoire**

Trois cellules photoélectriques sont éclairées par trois rayons lumineux clignotants. Ces rayons doivent clignoter de manière séquentielle pour que les trois rayons ne soient jamais allumés ou éteints en même temps. Chaque cellule sert à contrôler un de ces rayons et constitue un circuit qui produit une tension de sortie BASSE quand elle est dans le noir, et une tension de sortie HAUTE quand elle est éclairée.

Créez un circuit logique dont les entrées sont les tensions de sortie de ces trois cellules et qui doit produire un signal de niveau HAUT quand les trois rayons sont allumés ou éteints en même temps.

**Seq3.Exercice 2 : Circuit combinatoire**

Quatre grandes cuves dans une usine de fabrication de produits chimiques contiennent différents liquides chauffés. Des capteurs limnimétriques (de niveau) servent à déceler le dépassement d’un niveau préétabli dans les cuves A et B. Des capteurs thermométriques surveillent la température des cuves C et D pour qu’elle ne descende pas sous une température de consigne. Supposez que les capteurs limnimétriques sont à 0 quand le niveau est correct et à 1 quand il est trop haut. En outre, supposez que les capteurs thermométriques sont à 0 quand la température est acceptable et à 1 quand elle est trop basse.

Concevez un circuit logique qui sonne l’alerte quand se produisent en même temps un niveau trop haut dans A ou B et une température trop basse dans la cuve C ou la cuve D.

**Seq3.Exercice 3 : Comparateur**

On rencontre très souvent la nécessité de comparer deux entiers (A = B, A > B ou A < B).

1. Ecrire la table de vérité correspondant à ces trois fonctions de comparaison de 2 bits. La fonction S1 doit être égale à 1 si et seulement si A > B, la fonction I1 si et seulement si A < B et la fonction E1 si et seulement si A = B. Ce qui nous donne :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | S1 (A > B) | I1 (A < B) | E1 (A = B) |
| 0 | 0 |  |  |  |
| 0 | 1 |  |  |  |
| 1 | 0 |  |  |  |
| 1 | 1 |  |  |  |

2. En déduire les expressions logiques de S1, I1 et E1.

3. Donner le diagramme d'un bloc logique comparant deux bits A et B, en utilisant des inverseurs, des portes ET et OU.

On désire comparer deux nombres binaires de deux bits chacun, A (A1 A0) et B (B1 B0), le poids des bits étant 20 pour A0 et B0 et 21 pour A1 et B1.

Le comparateur aura 3 sorties : S2 doit être égale à 1 si et seulement si A > B, la fonction I2 si et seulement si A < B et la fonction E2 si et seulement si A = B.

1. Donner la table de vérité liant S2, I2, E2 à A1 A0 B1 B0.
2. Simplifier ces fonctions à l’aide de tableaux de Karnaugh.
3. Dessiner le logigramme en utilisant des portes NAND
4. Les formes d’onde de la figure Exo3.1 sont appliquées au comparateur telles qu’illustrées. Déterminez la forme d’onde de la sortie égalité (A=B).

|  |  |
| --- | --- |
| ***Figure Exo3.1*** |  |

**Seq3.Exercice 4 : Additionneur binaire**

# Un demi-additionneur permet de réaliser l’addition de deux bits (variables ai, bi). Le résultat de l’addition est la somme Si, et une retenue Ri.

1. Donner la table de vérité du demi-additionneur : deux entrées ai, bi, deux sorties Si et Ri.
2. En déduire les équations booléennes de Si et Ri en fonction de ai et bi.
3. Faire le schéma du demi-additionneur avec des portes logiques.

Pour additionner deux nombres à plusieurs bits, il faut tenir compte de l’éventuelle retenue de l’étage précédent (dans l’opération, les retenues se propagent d’un rang au rang supérieur). Un demi-additionneur ne peut donc réaliser cette opération. Il faut un circuit à 3 entrées, c’est ce que l’on appelle un additionneur complet.

Les variables Si et Ri correspondent au résultat (somme et retenue) de l’addition de ai, bi et Ri-1

Donner la table de vérité de l’additionneur complet : 3 entrées ai, bi, Ri-1, deux sorties Si et Ri.

1. En utilisant un tableau de Karnaugh pour chacune des deux sorties, en déduire les équations logiques de Si et de Ri.
2. Faire le schéma de l’additionneur complet avec des portes logiques.

**Seq3.Exercice 5 : Schéma interne d’un multiplexeur**

Un multiplexeur est un circuit logique à N entrées de données (Ei) et une sortie S qui transmet les informations provenant de l’entrée sélectionnée. Pour sélectionner une entrée, le multiplexeur reçoit un code de sélection qui provient de n entrées complémentaires appelées entrées d’adresse (Aj). Les nombres N et n sont liés par la relation N=2n.

Multiplexeur élémentaire à 2 entrées de données (donc une entrée d’adresse) :

1. Donner la table de vérité du multiplexeur
2. En déduire l’équation logique de la sortie S
3. Donner le schéma du multiplexeur

Multiplexeur à 4 entrées de données (donc 2 entrées d’adresse) :

1. Donner la table de vérité du multiplexeur
2. En déduire l’équation logique de la sortie S
3. Donner le schéma du multiplexeur

**Seq3.Exercice 6 : Application du multiplexeur – sélection de données**

1. Pour le multiplexeur de la figure Exo6.1, déterminez la sortie en réponse aux états d’entrée suivants : D0=0, D1=1, D2=1, D3=0, S0=1, S1=0.

On rappelle que S0 et S1 sont les entrées de sélection, et D0 à D3 les entrées de données.

La table de vérité du MUX est la suivante :

|  |  |
| --- | --- |
| Entrées de sélection  S1 S0 | Y |
| 0 0 | D0 |
| 0 1 | D1 |
| 1 0 | D2 |
| 1 1 | D3 |

|  |  |
| --- | --- |
| ***Figure Exo6.1*** (p334) |  |

1. Si les entrées de la sélection des données du multiplexeur de la figure Exo 6.1 suivent la séquence illustrée par les formes d’onde de la figure Exo6.2, déterminez la forme d’onde de sortie en réponse aux entrées de données spécifiées à la question 1.

|  |  |
| --- | --- |
| ***Figure Exo6.2*** |  |

1. On observe les formes d’onde de la figure Exo6.3 aux entrées d’un multiplexeur 74HC151 à 8 entrées. Dessinez la forme d’onde de la sortie Y.

On rappelle que S0, S1, S2 sont les entrées de sélection.

Il faut un niveau BAS sur l’entrée pour permettre aux données d’entrée sélectionnées de traverser jusqu’à la sortie. Si cette entrée est au niveau HAUT, alors Y=0.



|  |  |
| --- | --- |
| ***Figure Exo6.3*** |  |

**Seq3.Exercice 7 : Démultiplexeur4**

Le démultiplexeur à une entrée et quatre sorties représenté à la figure Exo7.1 est utilisé de la façon suivante : la forme d’onde série de l’entrée de données et les entrées de sélection des données (S0 et S1) sont illustrées à la figure Exo7.2.

|  |  |
| --- | --- |
| ***Figure Exo7.1*** |  |
|  |  |

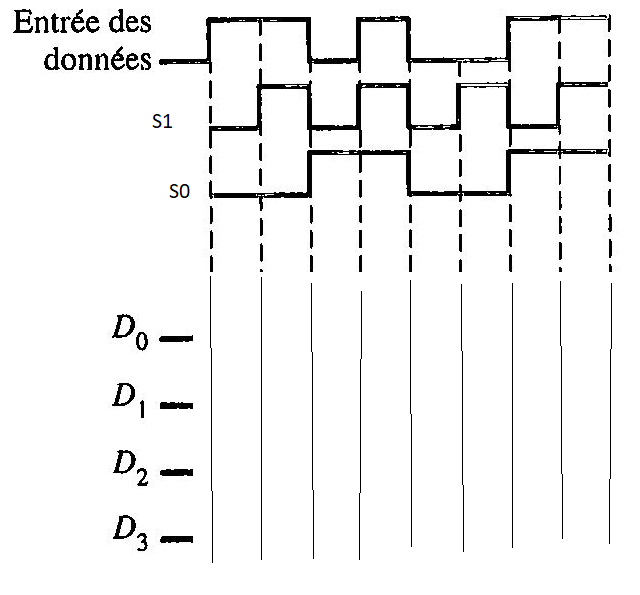
1. Complétez la table de vérité du démultiplexeur. E est l’entrée de données.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S1 | S0 | D0 | D1 | D2 | D3 |
| 0 | 0 | E | 0 | 0 | 0 |
| 0 | 1 |  |  |  |  |
| 1 | 0 |  |  |  |  |
| 1 | 1 |  |  |  |  |

1. Déterminez les formes d’onde des sorties de D0 à D3 du démultiplexeur.

|  |  |
| --- | --- |
| ***Figure Exo7.2*** |  |

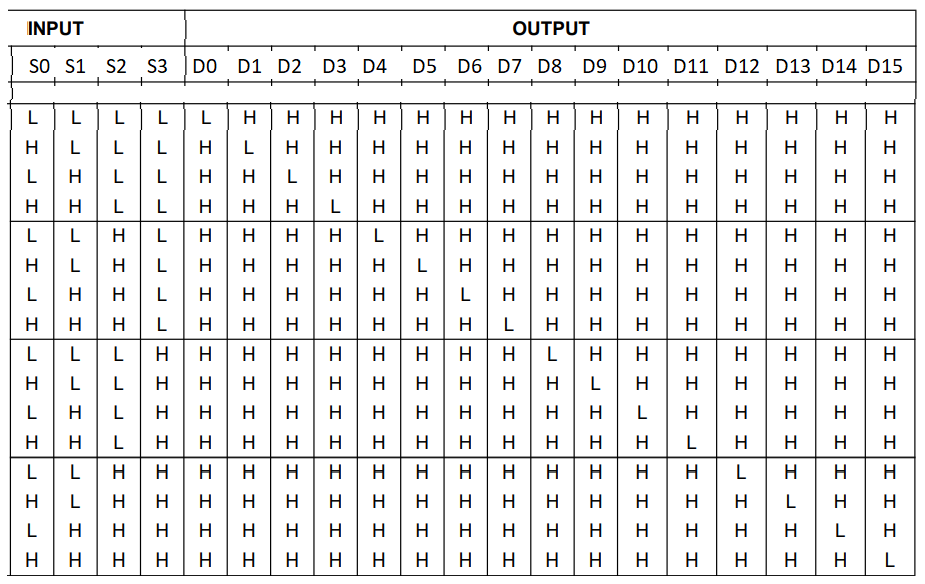
1. Développez le chonogramme du démultiplexeur si les formes d’onde S0 et S1 sont inversées.

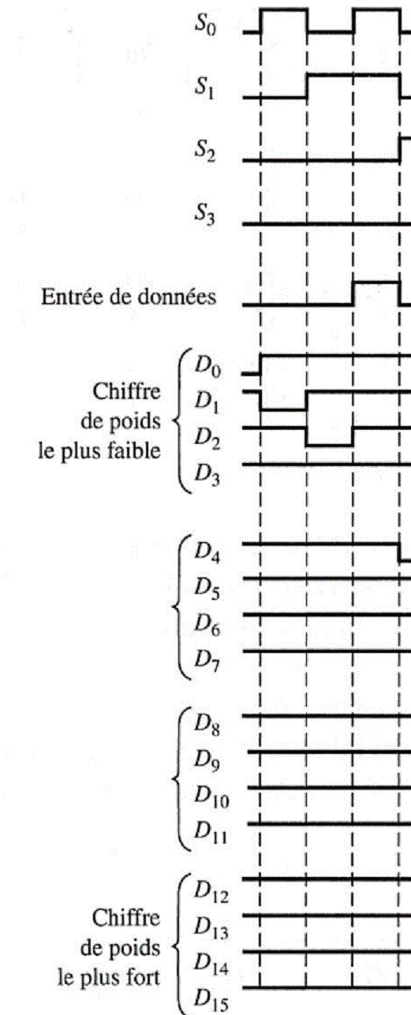


**Seq3.Exercice 8 : Démultiplexeur16**

Complétez le chronogramme de la figure Exo8.1 (terminez de représenter l’entrée E et les sorties D0 à D15) d’un CI 74HC154 utilisé dans une application de démultiplexage, à partir des conditions d’entrée suivantes. Les entrées de sélection S0 à S3 suivent une séquence binaire ascendante débutant à 0000 et l’entrée de données est un flux série formé de données DCB représentant le nombre décimal 2468. Le chiffre de poids le plus faible (8 soit 1000 en DCB) est le premier de la séquence, avec son bit de poids le plus faible placé en premier (d’où le début du tracé de E : 0001).

La table de vérité du démultiplexeur est la suivante :





***Figure Exo8.1 : chronogramme du composant 74154 utilisé comme démultiplexeur, à compléter.***

**Seq3.Exercice 9 : Génération d’une fonction logique quelconque avec un multiplexeur**

1. Réaliser la fonction logique suivante en utilisant uniquement un MUX 3 entrées d’adresse (8 entrées de données).

2. Même question en utilisant un MUX 2 entrées d’adresses (4 entrées de données) et des inverseurs si besoin.

**Seq3.Exercice 10 : Circuit de changement de code**

Un décodeur reçoit en entrée des données codées en **binaire** et fournit en sortie ces données en code **Gray (ou binaire réfléchi)**. Pour chacun des bits de sortie g0, g1, g2, g3 il s'agit donc d'établir un circuit de décodage ayant chacun en entrée les quatre bits b0, b1, b2, b3.

1. Donner les quatre tableaux de Karnaugh correspondants à g0, g1, g2, g3.
2. Donner les expressions des fonctions de sortie g0, g1, g2, g3.
3. Vérifier que le transcodeur nécessite seulement 3 portes XOR (OU EXCLUSIF) et présenter le schéma sous une forme qui serait facilement extensible à un nombre quelconque de bits.
4. Refaire les 3 premières questions en concevant cette fois un décodeur qui reçoit des données codées en Gray à l’entrée (g0, g1, g2, g3), et qui donne en sorties le mot binaire correspondant (b0, b1, b2, b3).

**Seq3.Exercice 11 : Générateur du code complément à 2 d’un nombre de 4 bits**

Un nombre binaire (N4N3N2N1) peut être converti en code complément à 2 (C4C3C2C1) par deux procédures différentes :

* complémenter chaque bit du nombre initial et ajouter 1 au résultat obtenu, en ignorant la retenue éventuelle
* recopier les bits du nombre initial à partir du bit de poids faible (LSB) vers le bit de poids fort (MSB), jusqu’au premier 1 inclus, et au delà complémenter chaque bit.

1. Vérifier l’équivalence des deux procédures sur deux exemples (deux nombres binaires à 4 bits que vous choisissez).
2. Pour chacune des deux procédures, proposer une expression de C4, de C3, de C2 et de C1, en fonction des bits N4, N3, N2 et N1.
3. Pour chacune des deux procédures, proposer un circuit, facilement extensible à un nombre quelconque de bits (prévoir l’ajout d’un cinquième bit en entrée, et d’une cinquième sortie), permettant d’obtenir le complément à deux d’un nombre binaire.

**Seq3.Exercice 12 : Décodeur binaire**

1. Combien de sorties comporte un décodeur binaire qui a n entrées ?
2. Donner la table de vérité d’un décodeur binaire 4 entrées, actif au niveau bas (une sortie est active lorsqu’elle est à 0).

**Seq3.Exercice 13 : Encodeur de priorité**

1. Remplir la table de vérité de l’encodeur de priorité suivant, pour lequel on obtient en sortie le code binaire de l’entrée prioritaire (code binaire donné sur S1S0 avec S1 poids fort). Une entrée Ei est active si la variable binaire qui lui correspond est mise à 1. Si plusieurs entrées sont activées en même temps, la priorité est donnée à l’entrée dont le numéro est le plus important, c’est donc le code binaire de cette entrée qui est fourni sur les sorties.

Afin de réduire la taille de la table de vérité, tout en considérant tous les cas, il est conseillé d’utiliser le signe X (indéterminé) pouvant signifier ici « quelque soit », dans les entrées.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Entrées*** | | | | ***Sorties*** | |
| E0 | E1 | E2 | E3 | S1 | S0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

2. En déduire les expressions simplifiées au maximum des deux sorties S1, S0.

3. Faire le schéma de l’encodeur.