PHY4103 - Systèmes et fonctions électroniques II – Projet Electronique Numérique : Etude théorique

Coordonnatrice : Nesma HOUMANI

Sommaire

[Problème 1 : Réalisation de la commande d’un ascenseur 1](#_Toc32274107)

[Problème 2 : Réalisation d’un chronomètre avec bouton de commande et affichage en décimal 4](#_Toc32274108)

[Problème 3 : Réalisation d’un feu de signalisation pour piétons 8](#_Toc32274109)

# Problème 1 : Réalisation de la commande d’un ascenseur

Contexte :

On souhaite réaliser la commande d’un ascenseur pouvant desservir quatre niveaux : 0, 1, 2 et 3. À tout moment, l’ascenseur se trouve dans l’un des trois états suivant : arrêt, montée, descente.

Postulats :

L’utilisateur interagit avec l’ascenseur en demandant un étage sur les quatre desservis. Nous utiliserons pour cela quatre boutons poussoirs, un pour chaque étage. En utilisant l’Arduino ou le FPGA, la demande de l’utilisateur sera codée sur deux bits :

L’ascenseur possède un attribut étage représentant l’étage actuel de l’ascenseur (codé également sur deux bits) et un attribut état indiquant si l’ascenseur se trouve à l’arrêt ou non (un bit) et s’il doit monter ou descendre (un bit) :

* : étage de l’ascenseur
* : pour : 0 = arrêt, 1 = non-arrêt, pour : 1 = monter, 0 = descendre

Il s’agit alors de comparer la demande de l’utilisateur avec l’étage actuel de l’ascenseur. Si la demande est plus petite, on passe à 1 et à 0 car l’ascenseur doit descendre. On répète le processus jusqu’à ce que l’ascenseur parviennent jusqu’à l’utilisateur. Le tout à l’aide de comparateurs et d’un compteur avec des bascules JK.

Comparateur :

Tableau 1 - Table de vérité du comparateur

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Niveau | Utilisateur | | Etage de l’ascenseur | | Arrêt/  Non-arrêt | Montée/  Descente |
|  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | X |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | X |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | X |
| 1 | 1 | 1 | 1 | 1 | 0 |
| 3 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | X |

On obtient alors :

Bascule JK :

Pour incrémenter ou décrémenter et on utilise un compteur avec des bascules JK. Il existe quatre états possibles : 0 -> 3. On utilise alors un compteur synchrone modulo 4 et donc 2 bascules JK.

Compteur synchrone :

Tableau 2 - Table de vérité du compteur synchrone

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INC |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | X | 0 | X |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | X | 1 |
| 0 | 1 | 0 | 0 | 1 | X | 1 | 1 | X |
| 0 | 1 | 1 | 1 | 0 | X | 0 | X | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | X | 1 | X |
| 1 | 0 | 1 | 1 | 0 | 1 | X | X | 1 |
| 1 | 1 | 0 | 1 | 1 | X | 0 | 1 | X |
| 1 | 1 | 1 | 1 | 1 | X | 0 | X | 0 |

Tables de Karnaugh :

J1 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| e\_0  a\_1 a\_0 | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | X | X |
| 1 | 0 | 1 | X | X |

K1 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| e\_0  a\_1 a\_0 | 00 | 01 | 11 | 10 |
| 0 | X | 0 | 0 | 1 |
| 1 | X | X | 0 | 0 |

J0 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| e\_0  a\_1 a\_0 | 00 | 01 | 11 | 10 |
| 0 | 0 | X | X | 1 |
| 1 | 1 | X | X | 1 |

K0 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| e\_0  a\_1 a\_0 | 00 | 01 | 11 | 10 |
| 0 | X | 1 | 1 | X |
| 1 | X | 1 | 0 | X |

On obtient alors le circuit suivant :

# Problème 2 : Réalisation d’un chronomètre avec bouton de commande et affichage en décimal

Figure 1 - Schéma logique du problème 1

Contexte :

On souhaite réaliser un chronomètre qui pourra mesurer des temps entre 0 et 15 secondes, par pas d’une seconde. Le chronomètre sera contrôlé grâce à un bouton poussoir, BP, qui fera varier successivement l’état du compteur entre marche, arrêt et mise à zéro. Le compteur n’augmentera pas au-delà des 15 secondes. La valeur du chronomètre sera affichée.

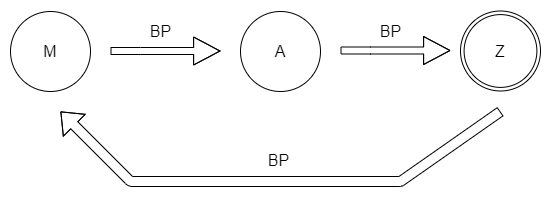
Postulats :

Le comptage des secondes se fera à l’aide de bascules JK. Puisqu’on a 16 valeurs (0 – 15), 4 bascules seront utilisées pour cette partie. La fréquence de l’horloge en entrée de ces bascules sera 1 Hz (période : 1 s) pour avoir un incrément toutes les secondes. Le mot binaire, T, associée à la valeur du chronomètre se décompose selon les bits suivants :

L’état du chrono, E, sera déterminé par deux bits :

* : pour : 0 = ‘activation de ’ (M/A), 1 = mise à zéro (Z), pour : 1 = démarrage (M), 0 = arrêt (A)

On bouclera de façon systématique entre ces valeurs lors de l’appui du bouton poussoir, en partant de Z.



Les entrées des bascules pour le comptage sont déterminées grâce à la table de transition (Tableau 2). L’astuce utilisée pour remettre à zéro le compteur est de faire un ET-logique entre T et /e1. De ce fait, si e1 vaut 1, T+1 sera égal à 0000.

Tableau 3 Récapitulatif des variables utilisées

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbole** | **Description** | **Nombre de bits** | **Valeurs possibles** |
| BP | Bouton Poussoir | 1 | {0,1} |
| T | Valeur du chronomètre | 4 | {0000 – 1111} |
| E | Etat du chronomètre | 2 | {00,01,10} |

Tableau 4 Table de transition de T

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0X | 0X | 0X | 1X |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0X | 0X | 1X | X1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0X | 0X | X0 | 1X |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0X | 1X | X1 | X1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0X | X0 | 0X | 1X |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0X | X0 | 1X | X1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0X | X0 | X0 | 1X |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1X | X1 | X1 | X1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | X0 | 0X | 0X | 1X |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | X0 | 0X | 1X | X1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | X0 | 0X | X0 | 1X |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | X0 | 1X | X1 | X1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | X0 | X0 | 0X | 1X |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | X0 | X0 | 1X | X1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | X0 | X0 | X0 | 1X |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | X0 | X0 | X0 | X1 |

A partir de cette table, on retrouve les équations suivantes pour les  :

On peut donc en déduire le circuit suivant :

A close up of text on a black background

Description automatically generated

Figure 2 Schéma logique du problème 2 (en excluant la mise à zéro)

# Problème 3 : Réalisation d’un feu de signalisation pour piétons

Contexte :

On souhaite réaliser un feu de croisement entre voitures et piétons. Il y aura deux variantes de ce feu :

* Le premier sera un système cyclique, indépendant des voitures, V, et des piétons, P. Le feu pour les voitures passera du vert(V\_V) à l’orange(V\_O) puis au rouge(V\_R). Le feu pour les piétons restera rouge(P\_R) pendant que celui des voitures est vert ou orange. Dès que ce dernier passe au rouge, le feu du piéton passe au vert(P\_V). Ensuite, quelques instants avant le passage au vert du feu des voitures, celui des piétons passe au rouge.
* Le système prend en compte l’arrivée d’un piéton, grâce à un bouton poussoir qui symbolisera une demande. Aussi longtemps qu’il n’y a pas de piétons, le feu des voitures reste vert. A l’arrivée du piéton, on fait une fois le cycle précédemment décrit, en espérant que le piéton ne mette pas trop de temps à traverser…

Postulats :

La base de temps du système est un mot binaire, C, de 4 bits (C0 étant le LSB et C3, le MSB). Un compteur modulo 16 est réalisé à base de bascules JK comme décrit dans la figure 3. La fréquence de l’horloge, H, sera de l’ordre du Hertz.

A picture containing sky

Description automatically generated

Figure 3 Compteur Modulo 16 avec des bascules JK

Puisque 4 bits ont été choisis pour le compteur, la période de changement des feux est de 16. Le feu des voitures maintiendra le vert et le rouge pendant 6 cycles chacun et l’orange, pendant 4 cycles. Donc, le vert du feu piétons durera 5 cycles (6 – 1 car on souhaite que le feu du piéton passe au rouge quelques instants avant que celui des voitures passe au vert). Le rouge durera 11 cycles.

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbole** | **Description** | **Nombre de bits** | **Valeurs possibles** |
| C | Compteur qui sert comme base de temps (LSB : C0, MSB : C3) | 4 | {0000 – 1111} |
| V\_V | Feu vert des voitures | 1 | {0,1} |
| V\_O | Feu orange des voitures | 1 | {0,1} |
| V\_R | Feu rouge des voitures | 1 | {0,1} |
| P\_V | Feu vert des piétons | 1 | {0,1} |
| P\_R | Feu rouge des piétons | 1 | {0,1} |
| BP | Bouton Poussoir pour demander de traverser | 1 | {0,1} |

Tableau 5 Table de vérité des sorties (des feux)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **C3** | **C2** | **C1** | **C0** | **V\_V** | **V\_O** | **V\_R** | **P\_V** | **P\_R** |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

On retrouve les expressions suivantes pour les sorties :

Cela nous donne le circuit suivant :

A close up of text on a white background

Description automatically generated

Figure 4 Schéma logique des feux de signalisation