# Systèmes et fonctions électroniques – Projet : Etude théorique

## 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. Ainsi, 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 JK.

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 JK. Il existe quatre état possible : 0 -> 3. On utilise alors un compteur synchrone modulo 4 et donc 2 bascules JK.

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 |

Ainsi :

On obtient alors le circuit suivant :

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

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.

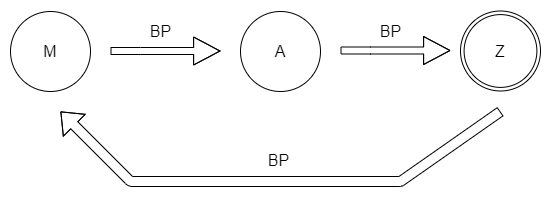
Postulats :

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 1 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 2 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 1 Schéma logique du problème 2 (en excluant la mise à zéro)