Rapport labo 7 SYL

## Explications des circuits :

### decodeur\_sorties :

* Ce circuit prend en entré un état de la machine et défini l’état des sorties.
* Les états sont définis sur 7 bits (1 parmi M donc un bit à 1 par état) :
  + Wait : 0000001
  + Scan : 0000010
  + Throw : 0000100
  + Move red : 0001000
  + Move blue : 0010000
  + Drop : 0100000
  + Move init : 1000000
* La variables des sorties est aussi sur 7 bits :
  + Bit0 : scan\_o
  + Bit1 : throw\_o
  + Bit2 : move \_o
  + Bit3 : dest\_red
  + Bit4 : dest\_blue
  + Bit5 : dest\_init
  + Bit6 : drop
* Équations des sorties en fonction des états présents, page 4

### decodeur\_etat :

* Prend en entré l’état présent, les entrés de la machine et défini l’état future en fonctions de cela.
* Entrés de la MSS :
  + Ready\_i : est à 1 quand le bras est prêt à effectuer un scan
  + Color\_i : sur 2 bits, indique la couleur du bloc ou une erreur
  + Timer : est à un quand le timer est terminé
  + Reset : est à 1 lorsque l’on veut faire un reset
* Q[6:0] : les bits de l’état présent en entré du circuit
* State\*\_bit\*[6:0] : les bits de l’état future :
  + Dans notre cas il peut y avoir qu’un bit à 1 à la fois (1 parmi M)
  + Bit0 à 1 -> état 0, bit 1 à 1 -> état 1 etc…
* Équations pour chacun des bits d’états futur, page 3

### MSS :

* Contiens les circuits decodeur\_sorties et decodeur\_etat.
* Détermine l’état des sortis en fonction de l’état de la machine et détermine son prochain état en fonction de l’état actuel et des entrés.
* La machine fait avancer les états avec une horloge et des bascules D
  + Chaque bascule gère un bit de l’état
  + Les reset des bascules des bit 6 à 1 et le set de la bascule du bit0 sont relié au reset asynchrone de la machine
  + Cela pour que lorsque l’on fait un reset de la machine, elle se remette dans son état « Wait » (E0 : 0000001) où les bits 6 à 1 sont donc remis à 0 et le bit0 à 1
* Tables des états en page 3

### Timer

* Prend en entré un reset, un enable et une horloge
* Lorsque le enable est à 1, le timer décompte et est donc actif
* Si le enable est à 0, le timer charge la valeur 2 et est désactivé
  + Le timer décompte sur 3 temps donc de 2 à 0
* La sortie end\_timer\_o est à 1 lorsque le décompte arrive à 0

### auto\_arm\_top

* Prend en entré ready\_i, color\_i, reset\_i et clk\_i déjà expliqué précédemment
* Contient les circuits timer et MSS
  + MSS et timer sont donc utilisé afin de gérer les sorties pour faire fonctionner le bras
* La sorite move\_o relie l’enable du timer afin qu’il soit actif et décompte uniquement lorsque la machine est aux états Move red, Move blue ou Move init, pour qu’il décompte bien de 2 à 0, et n’aille pas déjà commencer quand la machine arrive à un de ces états. C’est donc aussi pour cela que le reste du temps, quand l’enable est à 0, la valeur 2 est chargé.

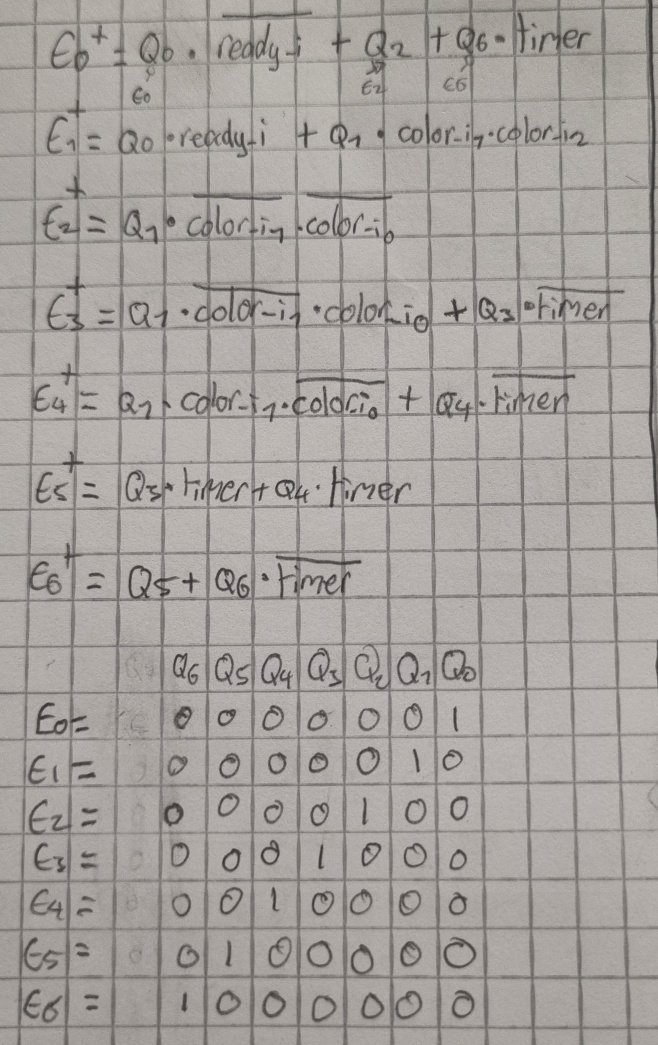
## Équations et tables de états :

### Tables des états

Une image contenant texte, écriture manuscrite, papier, document

Description générée automatiquement

### Équations des états futures



E\*+ : états futurs

E\* : états présents

Q[6 :0] : bit[6 :0] de l’état présent

E\*+ : états futurs

E\* : états présents

Q[6 :0] : bit[6 :0] de l’état présent

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

### Équations des sorties

Une image contenant texte, Police, blanc, typographie

Description générée automatiquement

Q\* : Correspondent au bit à 1 des différents états

Par exemple, la sortie S0 est à 1 quand la machine est dans l’état 1 (Scan, E1), et donc lorsque le bit 1 de la variable 7bits des états est à 1.

Ceci peut être fait car notre cas le permet grâce au 1 parmi M, et permet de simplifier.

## Remerciment

Nous tenons à remercier M.Auberson ainsi que Bastien pour tout le temps hors horaires qu’ils nous ont consacré.