SYL_Labo4

Analyse des entrées

Le circuit original comprend les entrées `ready_i`, `color_i`, `clk_i` et `reset_i`.

- `ready_i` : Arrive seulement lorsque le bras est en position initiale pour dire qu'il peut procéder au scan d'une boîte.
- `color_i` : Arrive une fois qu'un scan a été effectué et défini la couleur de la boîte.
  - 00 : Couleur indéterminé
  - 01 : Rouge
  - 10 : Bleu
  - 11 : Erreur
- `clk_i` : Horloge
- `reset_i` : Reset asynchrone de tout le système.

Ces entrées à elles seules ne nous permettent pas de faire tous les changements d'états de la façon demandée. Pour ça il nous faudra un compteur qui nous donnera une valeur 'compteur_done' qui servira à nous dire si le bras à fini de se déplacer car celui-ci prend 3 coups d'horloge pour effectuer son déplacement.
Ce compteur renvoi `1` quand il arrive à `1`.

Analyse des sorties

Les sorties correspondent simplement à la prochaine chose à faire soit au prochain état de la machine.

- `scan_o` : 1 quand la prochaine chose à faire c'est scanner une pièce
- `throw_o` : 1 quand le scan à rendu la valeur 00 donc une couleur indéterminé
- `move_o` : 1 quand le bras doit se déplacer, sera à 1 en même temps que la sortie définissant où il doit aller
- `dest_red_o` : 1 en même temps que `move_o` pour se déplacer à la zone rouge
- `dest_blue_o` : 1 en même temps que `move_o` pour se déplacer à la zone bleu
- `dest_init_o` : 1 en même temps que `move_o` pour se déplacer à la zone initiale
- `drop_o` : 1 quand le bras arrive en zone bleu/rouge et va devoir lacher la pièce tenue

Le bras étant une machine d'état séquenciel de MOORE, les sorties dépendent uniquement de l'état actuel et non pas des entrées.

Graphe des états

Le graphe d'état correspondant à notre machine est la suivante:

00 : couleur inconnue
01 : couleur rouge
10 : couleur bleu
11 : erreur de scan

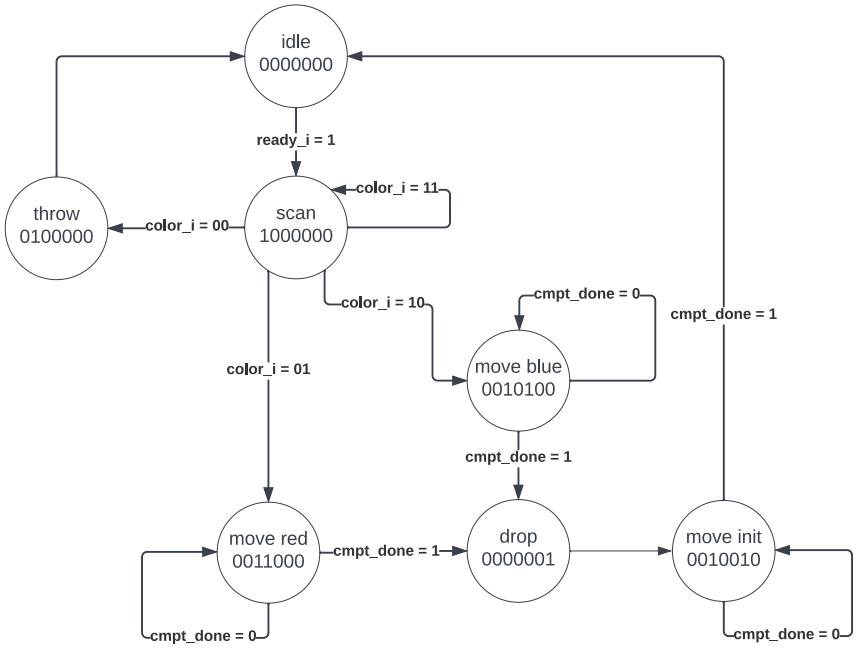

Table des états

Voici la table des états correspondant au graphe ci-dessus:

		Etat futur (ready,color1,color2,cmpt_done)										
	Etat présent	0000	xxx1	0010	0100	0110	1000	1010	1100	1110	Sortie	
Idle	0000001	0000001	-	0000001	0000001	0000001	0000010	0000010	0000010	0000010	0000000	Idle
Scan	0000010	0000100	-	0010000	0001000	0000010	0000100	0010000	0001000	0000010	1000000	Scan
Throw	0000100	0000001	-	0000001	0000001	0000001	0000001	0000001	0000001	0000001	0100000	Throw
Move Blue	0001000	0001000	0100000	0001000	0001000	0001000	0001000	0001000	0001000	0001000	0010100	Move Blue
Move Red	0010000	0010000	0100000	0010000	0010000	0010000	0010000	0010000	0010000	0010000	0011000	Move Red
Drop	0100000	1000000	-	1000000	1000000	1000000	1000000	1000000	1000000	1000000	0000001	Drop
Move Init	1000000	1000000	0000001	1000000	1000000	1000000	1000000	1000000	1000000	1000000	0010010	Move Init

Création des équations

Pour la création des équation nous nous sommes basé sur la table des états. La machine étant MOORE, les sorties sont facile à définir selon l'état actuel surtout puisque nous avons utilisé la notation 1 parmi M qui nous permet de regarder qu'un seul bit de l'état pour définir chaque sortie.

Sorties

Il n'y a pas de raison de trouver des équations pour les sorties car elles ne dépendent que de l'état présent qui lui n'a qu'un bit à 1. Ceci veut dire que nous pouvons simplement controler si un certain bit est à 1, si c'est le cas la sortie vaut une constante prédéfinie.

Etat présent	**Sortie**
0000001	0000000
0000010	1000000
0000100	0100000
0001000	0010100
0010000	0011000
0100000	0000001
1000000	0010010

Etats Futurs

$$idle^+ = idle * \overline{ready} + throw + moveInit * done$$

$$scan^+ = idle * ready + scan * color1 * color2$$

$$throw^+ = scan * \overline{color1} * \overline{color2}$$

$$moveBlue^+ = scan * color1 * \overline{color2} + moveBlue * \overline{done}$$

$$moveRed^+ = scan * \overline{color1} * color2 + moveRed * \overline{done}$$

$$drop^+ = moveBlue * done + moveRed * done$$

$$moveInit^+ = drop + moveInit * \overline{done}$$