

Rapport labo 7 SYL

Explications des circuits :

decodeur_sorties :

- Ce circuit prend en entrée un état de la machine et définit l'état des sorties.
- Les états sont définis sur 7 bits (1 parmi M donc un bit à 1 par état) :
 - o Wait : 0000001
 - o Scan : 0000010
 - o Throw : 0000100
 - o Move red : 0001000
 - o Move blue : 0010000
 - o Drop : 0100000
 - o Move init : 1000000
- Les variables des sorties sont aussi sur 7 bits :
 - o Bit0 : scan_o
 - o Bit1 : throw_o
 - o Bit2 : move_o
 - o Bit3 : dest_red
 - o Bit4 : dest_blue
 - o Bit5 : dest_init
 - o Bit6 : drop
- Équations des sorties en fonction des états présents, page 4

decodeur_etat :

- Prend en entrée l'état présent, les entrées de la machine et définit l'état futur en fonction de cela.
- Entrées de la MSS :
 - o Ready_i : est à 1 quand le bras est prêt à effectuer un scan
 - o Color_i : sur 2 bits, indique la couleur du bloc ou une erreur
 - o Timer : est à 1 quand le timer est terminé
 - o Reset : est à 1 lorsque l'on veut faire un reset
- Q[6:0] : les bits de l'état présent en entrée du circuit
- State*_bit*[6:0] : les bits de l'état futur :
 - o Dans notre cas il peut y avoir qu'un bit à 1 à la fois (1 parmi M)
 - o 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 sorties en fonction de l'état de la machine et détermine son prochain état en fonction de l'état actuel et des entrées.
- La machine fait avancer les états avec une horloge et des bascules D
 - o Chaque bascule gère un bit de l'état
 - o 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
 - o 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 entrée 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
 - o 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 entrée ready_i, color_i, reset_i et clk_i déjà expliqué précédemment
- Contient les circuits [timer](#) et [MSS](#)
 - o MSS et timer sont donc utilisé afin de gérer les sorties pour faire fonctionner le bras
- La sortie 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

Etat-prés	Etat-future P(ready, color, timer)														
	00	1	11	00	10	01	1	0	S6	S5	S4	S3	S2	S1	S0
wait E0	(E0)	E1							0	0	0	0	0	0	0
Scan E1		(E1)	E2	E4	E3				0	0	0	0	0	0	1
Throw E2						E0			0	0	0	0	0	1	0
Moved red E3							E5	(E3)	0	0	0	1	1	0	0
Move blue E4							E5	(E4)	0	0	1	0	1	0	0
Drop E5						E6			1	0	0	0	0	0	0
Move init E6							E6	(E6)	0	1	0	0	1	0	0

Équations des états futures

$$\begin{aligned}
 E_0^+ &= Q_0 \cdot \text{ready}_i + Q_2 + Q_6 \cdot \text{timer} \\
 E_1^+ &= Q_0 \cdot \text{ready}_i + Q_1 \cdot \text{color}_{i1} \cdot \text{color}_{i2} \\
 E_2^+ &= Q_1 \cdot \text{color}_{i1} \cdot \text{color}_{i0} \\
 E_3^+ &= Q_1 \cdot \text{color}_{i1} \cdot \text{color}_{i0} + Q_3 \cdot \text{timer} \\
 E_4^+ &= Q_1 \cdot \text{color}_{i1} \cdot \text{color}_{i0} + Q_4 \cdot \text{timer} \\
 E_5^+ &= Q_3 \cdot \text{timer} + Q_4 \cdot \text{timer} \\
 E_6^+ &= Q_5 + Q_6 \cdot \text{timer}
 \end{aligned}$$

	Q6	Q5	Q4	Q3	Q2	Q1	Q0
E0 =	0	0	0	0	0	0	1
E1 =	0	0	0	0	0	1	0
E2 =	0	0	0	0	1	0	0
E3 =	0	0	0	1	0	0	0
E4 =	0	0	1	0	0	0	0
E5 =	0	1	0	0	0	0	0
E6 =	1	0	0	0	0	0	0

E*+ : états futurs

E* : états présents

Q[6:0] : bit[6:0] de l'état présent

Nom de l'état	Abréviation correspondante
Wait	E0
Scan	E1
Throw	E2
Move Red	E3
Move Blue	E4
Drop	E5
Move init	E6

Équations des sorties

Handwritten equations for outputs S0 through S6:

- $S_0 = Q_7$
- $S_1 = Q_2$
- $S_2 = Q_3 + Q_4 + Q_6$
- $S_3 = Q_3$
- $S_4 = Q_4$
- $S_5 = Q_6$
- $S_6 = Q_5$

Sorties:

S[0]: scan_o

S[1]: throw_o

S[2]: move_o

S[3]: dest_red

S[4]: dest_blue

S[5]: dest_init

S[6]: drop

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.