

Logique séquentielle

I Compteur aléatoire

Réaliser avec des bascules D un compteur qui réalise la séquence suivante :

$0 \rightarrow 4 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 0$

Pour pouvoir réaliser un cycle de 6 transitions en comptant jusqu'à 5 au maximum, il va falloir 3 bascules (2^3 états maximum et codage jusqu'à 7)

Il faut ensuite évaluer les séquences $S_n = \{Q_2 Q_1 Q_0\}$ et les valeurs de D_0 , D_1 et D_0 qui permettent de passer de S_{n-1} à S_n en se souvenant de la table de vérité (très simple) d'une bascule D : $Q_{n+1} = D_n$

Etats	S_{n-1}			S_n			D_2	D_1	D_0
	Q_2	Q_1	Q_0	Q'_2	Q'_1	Q'_0			
0	0	0	0	1	0	0	1	0	0
4	1	0	0	0	0	1	0	0	1
1	0	0	1	0	1	1	0	1	1
3	0	1	1	1	0	1	1	0	1
5	1	0	1	0	1	0	0	1	0
2	0	1	0	0	0	0	0	0	0

On va ensuite déterminer les expressions de D_0 , D_1 et D_2 en fonction de Q_0 , Q_1 et Q_2 . à l'aide de tableaux de Karnaugh

pour D_2 :

$Q_1 Q_0$	00	01	11	10
Q_2				
0	1	0	1	0
1	0	0	X	X

$$D_2 = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0$$

pour D_1 :

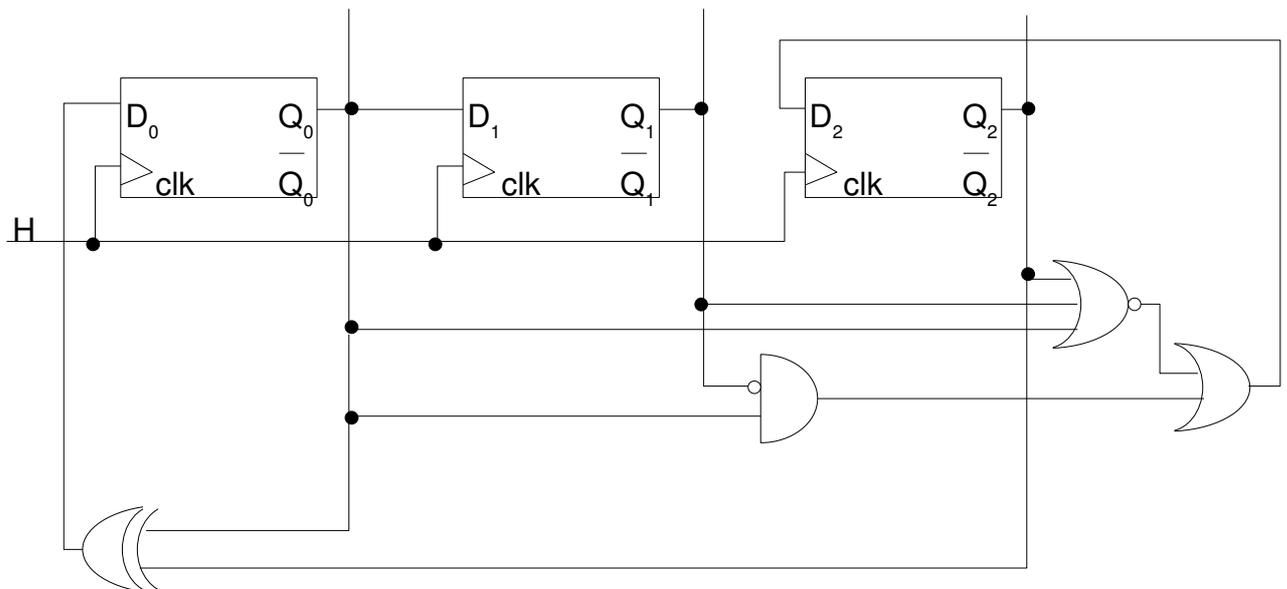
$Q_1 Q_0$	00	01	11	10
Q_2				
0	0	1	0	0
1	0	1	X	X

$$D_1 = \bar{Q}_1 Q_0$$

pour D_0 :

$Q_1 Q_0$	00	01	11	10
Q_2				
0	0	1	1	0
1	1	0	X	X

$$D_0 = \bar{Q}_0 Q_2 + Q_0 \bar{Q}_2 = Q_0 \oplus Q_2$$



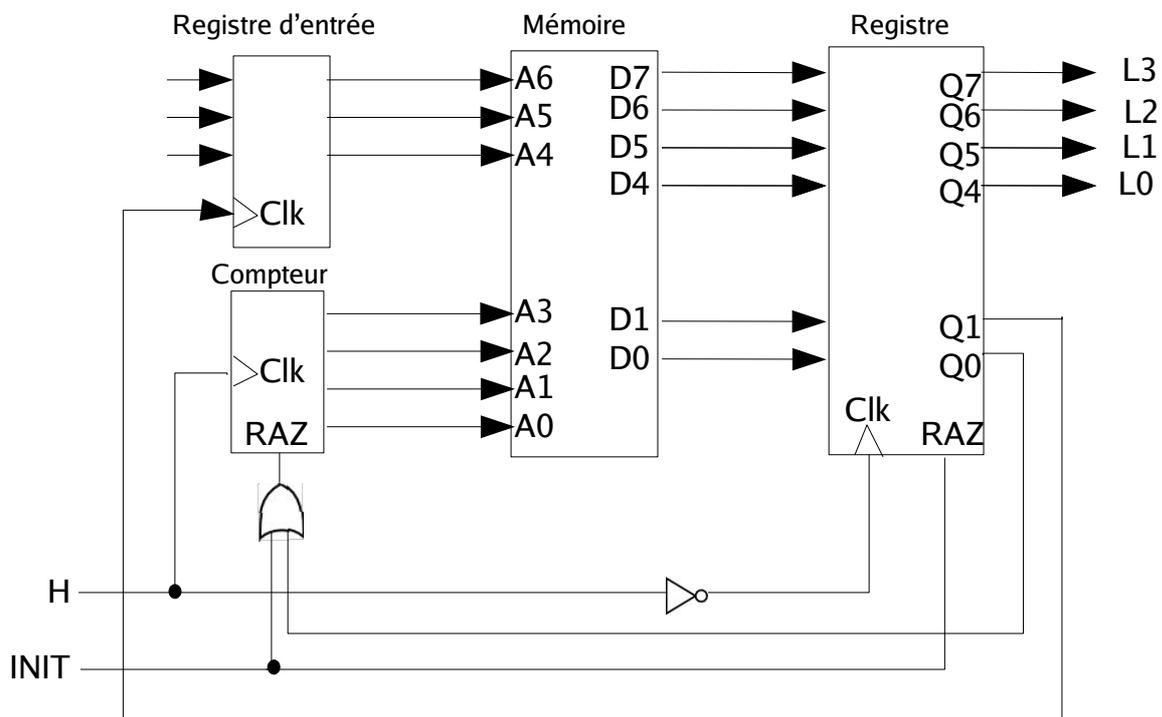
II Réalisation d'une animation lumineuse (chenillard)

On veut réaliser un système programmable de commande d'un jeu de lumières constitué de 4 lampes L0, L1, L2 et L3

Une séquence constitue une animation et peut comporter jusqu'à 16 pas. Chaque pas de séquence est stockée dans une case mémoire. On peut avoir jusqu'à 8 séquences.

La fréquence de l'horloge peut varier de 1 Hz à 25 Hz

Le circuit suivant est utilisé pour la réalisation du chenillard.



Chaque animation commence par le chargement du registre d'entrée (choix de la séquence) et se termine par la remise à zéro du compteur

1. Analysez les éléments du circuit pour en décrire les fonctions

- rôle des bits A0 à A6
- rôle des sorties Q0 à Q7
- format de stockage de l'animation

Les bits A6 à A4 permettent d'adresser la séquence et donc la zone où le programme de l'animation sera stockée. Les éléments de la séquence d'animation seront adressés par les bits A0 à A3 :

- adresse de début dans la zone de stockage : 000
- adresse de fin : 111

l'adresse du début de la zone de stockage est donc *code d'animation*16* et l'adresse de fin est donnée par *code d'animation*16 + 15*

Les sorties Q4 à Q7 du registre servent à piloter les lampes
 La sortie Q1 sert à valider le chargement du registre d'entrée
 La sortie Q0 remet à zéro le compteur

Dans chaque mot de la mémoire permettant de décrire la séquence d'animation, on trouve 3 parties pour

- l'état des lampes
- la commande de l'horloge du registre d'entrée
- la commande de remise à zéro du compteur

Chaque séquence d'animation va comporter 3 parties :

- le chargement du code d'entrée
- la génération des sorties
- la remise à zéro du compteur en fin de séquence

Format du mot mémoire

<i>Sorties</i>						<i>Horloge registre d'entrée</i>	<i>Remise à 0 du compteur</i>
Q7	Q6	Q5	Q4			Q1	Q0
L0	L1	L2	L3			CHG_E	RAZ

Chargement du code d'entrée :

- sur un front montant sur CHG_E, généré à l'aide de la sortie D1
- 1er mot : toutes les sorties à 0, CHG_E (D1) à 0, RAZ (D0) à 0
- 2ème mot : toutes les sorties à 0, CHG_E (D1) à 1, RAZ (D0) à 0 pour générer le front montant sur CHG_E

Pendant la génération des sorties, CHG_E et RAZ seront maintenus inactifs, c'est-à-dire à 0

2. Déterminez la zone de stockage de la séquence 0 dont le code d'animation vaut 0 et donner son contenu permettant de réaliser sur les sorties un registre à décalage qui allume successivement les lampes de gauche à droite, puis de droite à gauche

Contenu de la mémoire pour le décalage

<i>adresse</i>	<i>D7</i>	<i>D6</i>	<i>D5</i>	<i>D4</i>	<i>D3</i>	<i>D2</i>	<i>D1</i>	<i>D0</i>	<i>opération</i>
00H	0	0	0	0	0	0	0	0	chargement du code d'entrée
01H	0	0	0	0	0	0	1	0	
02H	1	0	0	0	0	0	0	0	L0 allumée
03H	0	1	0	0	0	0	0	0	L1 allumée
04H	0	0	1	0	0	0	0	0	L2 allumée
05H	0	0	0	1	0	0	0	0	L3 allumée
06H	0	0	1	0	0	0	0	0	L2 allumée
07H	0	1	0	0	0	0	0	0	L1 allumée
08H	1	0	0	0	0	0	0	0	L0 allumée
09H	0	0	0	0	0	0	0	1	remise à 0 du compteur

Même question pour la séquence 1 qui allume alternativement les lampes paires puis les lampes impaires

Contenu de la mémoire pour le clignotement

<i>adresse</i>	<i>D7</i>	<i>D6</i>	<i>D5</i>	<i>D4</i>	<i>D3</i>	<i>D2</i>	<i>D1</i>	<i>D0</i>	<i>opération</i>
10H	0	0	0	0	0	0	0	0	chargement du code d'entrée
11H	0	0	0	0	0	0	1	0	
12H	1	0	1	0	0	0	0	0	L0 et L2 allumées
13H	0	1	0	1	0	0	0	0	L1 et L3 allumées
14H	1	0	1	0	0	0	0	0	L0 et L2 allumées
15H	0	1	0	1	0	0	0	0	L1 et L3 allumées
16H	1	0	1	0	0	0	0	0	L0 et L2 allumées
17H	0	1	0	1	0	0	0	0	L1 et L3 allumées
18H	0	0	0	0	0	0	0	1	remise à 0 du compteur

III Réalisation d'un circuit "Barillet" (Barrel shifter)

On souhaite synthétiser un circuit sur 4 bits nommé "Barrel Shifter". Ce circuit possède 4 entrées de données (I[0:3]) et 4 sorties de données (O[0:3]). La fonction de ce circuit est de mettre les 4 entrées en communication avec les 4 sorties, et ce, en respectant un décalage, lui-même commandé par 2 entrées de commande S[0:1].

Les 4 modes de décalage souhaités sont :

$S_1 S_0$	O_3	O_2	O_1	O_0
0 0	I_3	I_2	I_1	I_0
0 1	I_0	I_3	I_2	I_1
1 0	I_1	I_0	I_3	I_2
1 1	I_2	I_1	I_0	I_3

Faire la synthèse de ce circuit en n'utilisant que des multiplexeurs 2 vers 1

Un multiplexeur 2 vers 1 ne possède qu'une seule ligne de sélection. Il ne peut aiguiller que 2 entrées vers une seule ligne de sortie. Pour réaliser le circuit permettant d'envoyer 4 entrées vers 4 sorties, il faudra donc cascader 2 étages de multiplexeurs 2 vers 1

Dans un premier temps, on "oublie" l'aiguillage de S_0 en combinant ses sorties :

S_1	O_3	O_2	O_1	O_0
0	I_3+I_0	I_2+I_3	I_1+I_2	I_0+I_1
1	I_1+I_2	I_0+I_1	I_3+I_0	I_2+I_3

En posant $M_{ij} = I_i + I_j$, on peut écrire la table de vérité suivante, selon S_0 :

S_0	M_{30}	M_{23}	M_{12}	M_{01}
0	I_3	I_2	I_1	I_0
1	I_0	I_3	I_2	I_1

et

S_1	O_3	O_2	O_1	O_0
0	M_{30}	M_{23}	M_{12}	M_{01}
1	M_{12}	M_{01}	M_{30}	M_{30}

un premier étage de multiplexeurs pilotés par S_0 va générer les M_{ij} à partir des entrées I_i puis un second étage va générer les sorties O_j .

