

Systemes Digitaux

Cours 7

Conception de circuits séquentiels

- Au dernier cours, on a appris à analyser des circuits séquentiels:
 - À partir d'un circuit, on a voulu connaître son tableau d'état et son diagramme d'état
 - Avec cette information, il est possible de mieux comprendre le comportement du circuit
- Aujourd'hui, on va faire l'étape inverse:
 - À partir d'un comportement souhaité, on concevra le circuit pour le faire...

Conception de circuits séquentiels

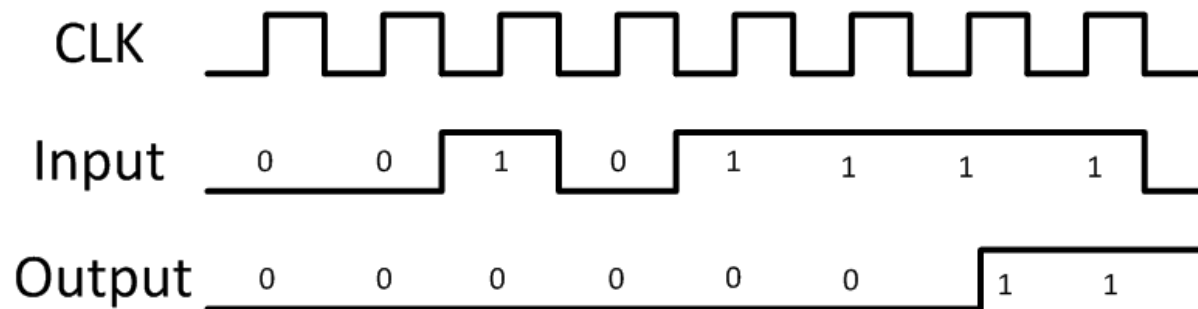
- L'approche comporte les étapes suivantes:
 - 1) Trouver le diagramme d'états à partir de la description en langage humain
 - 2) Réduire le nombre d'états (Pas aujourd'hui)
 - 3) Assigner des valeurs binaires aux états
 - 4) Dessiner un tableau d'états
 - 5) Dériver et simplifier l'équation logique en entrée des flip flops
 - 6) Dessiner le diagramme logique

Types de machines

- Il existe 2 façons de générer la sortie:
 - Machine de Moore: sortie définie dans les bulles
 - Machine de Mealy: sortie définie sur les flèches
- Machine de Moore:
 - Sortie ne dépend QUE de l'état
 - Sortie inchangé même si l'entrée change
- Machine de Mealy:
 - Sortie dépend de l'état ET de l'entrée
 - Requier moins d'états

Exemple

- Dessinez le circuit logique pour un détecteur de squence:
 - L'entrée reçoit un bit à chaque coup d'horloge
 - La sortie devient '1' quand 3 '1' consécutifs arrivent
 - Sinon, la sortie est '0'
 - Clarification: s'il y avait 4 '1' de suite, la sortie serait '1' pendant 2 cycles consécutifs...



Exemple

- On commence par traduire tout ça en un diagramme d'états...
 - On dessine un diagramme avec des cercles et des flèches qui correspond à la description
 - On donne des noms pertinents à chaque état pour assurer la compréhension
- Quels et combien d'états a-t-on besoin?
 - Un exemple d'états c'est: "État 1: j'ai reçu un 1"
 - Un autre exemple: "État 2: j'ai reçu deux 1"

Exemple

- Avec de l'expérience, on peut trouver les états rapidement...
 - Au début, c'est normal d'avoir des problèmes!
- Les états suivants devraient fonctionner

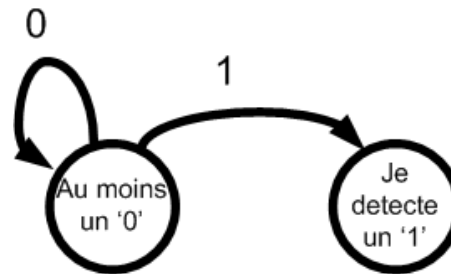


- Et si on se trompe, on n'a qu'à recommencer...

Dessignons les flèches qui connectent les bulles..

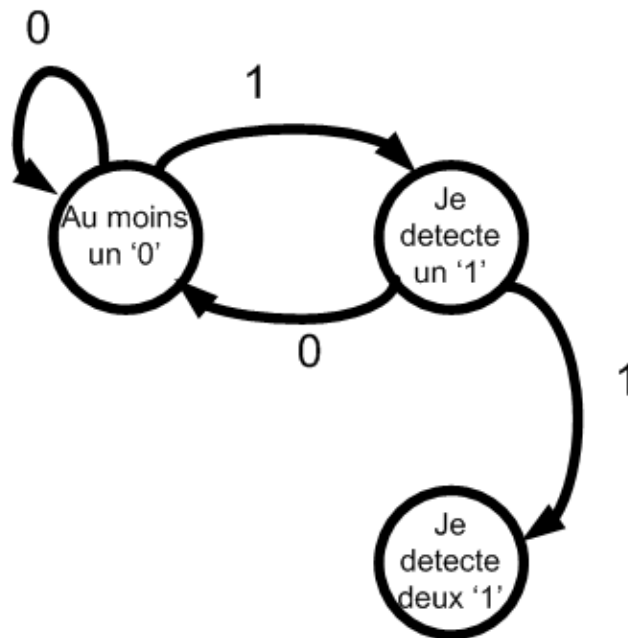
Exemple

- Pour analyser le système, on commence du début...
 - Quand j'allume le système, il n'y a rien dans le système encore... Il y a des 0 partout.
- Je commence donc à l'état "Au moins un '0' "
 - Si je reçois '0', je reste là
 - Si je reçois '1', je passe à l'état "Je détecte un '1' "



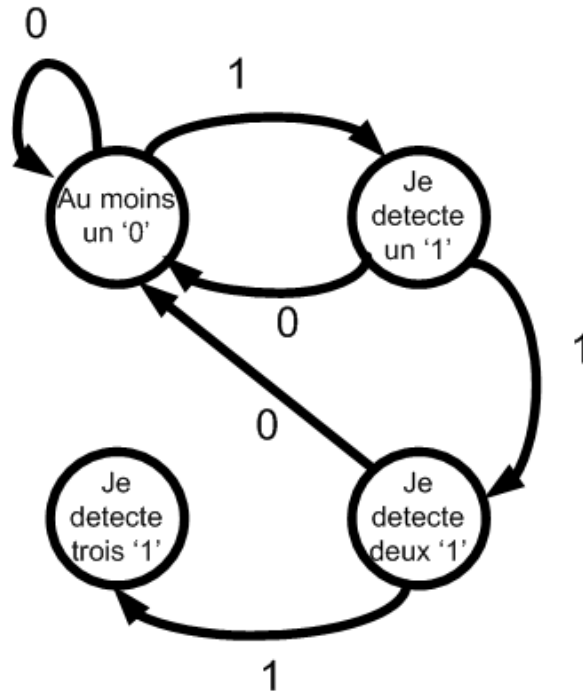
Exemple

- À ce point, j'ai déjà détecté un '1'
- Un autre bit arrive...
 - Si je détecte '0', je recommence du debut
 - Si je détecte '1', je passe à "Je détecte deux '1' "



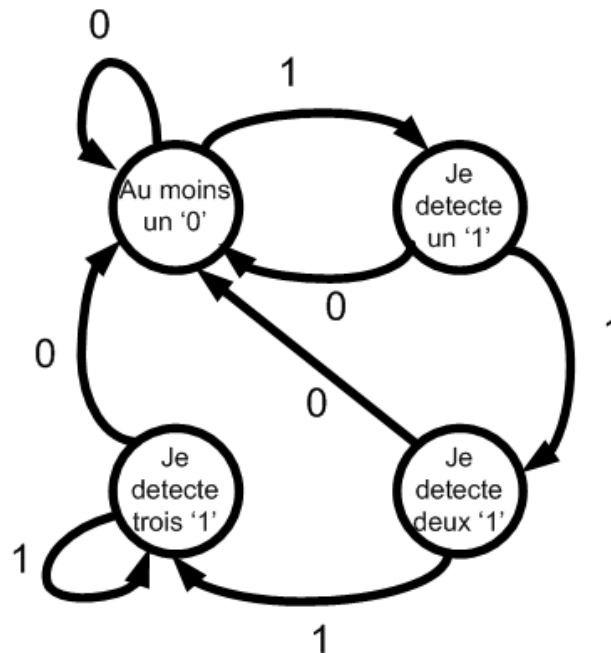
Exemple

- Maintenant, j'ai détecté deux '1'... Un autre bit arrive
 - Si je détecte '0', je recommence du début
 - Si je détecte '1', je passe à "Je détecte trois '1' "



Exemple

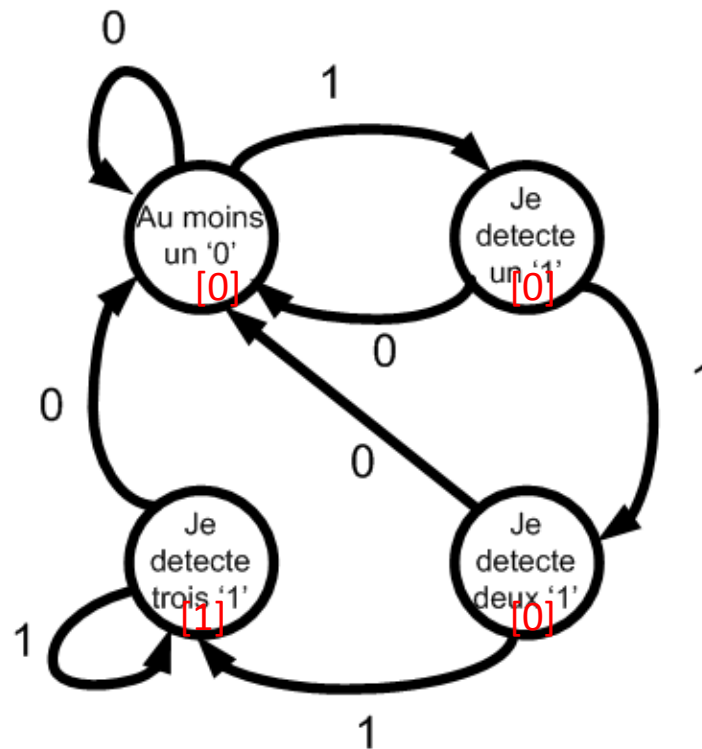
- Maintenant, j'ai détecté trois '1'... un autre bit arrive
 - Si je détecte '0', je recommence du début
 - Si je détecte '1', je reste là...



Il reste à déterminer
la sortie

Exemple

- En utilisant la machine de Moore, la sortie est à 1 quand je détecte 3 '1'...
 - La sortie est 0 dans tous les autres états

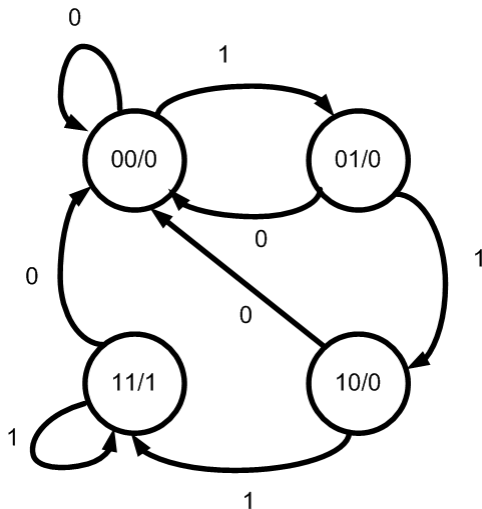


Exemple

- La prochaine étape est de donner des valeurs numériques aux 4 états
- Il y a 3 façons populaires d'assigner les états:
 - En binaire: ce qu'il y a de plus "normal"
 - En code grey: 000, 001, 011, 010, 110, ...
 - En one-hot: 1 flip flop par état
- On va y aller avec la méthode binaire pour commencer...

Exemple

- Voici comment j'ai assigné les états :
 - "Au moins un '0' ": 00
 - "J'ai détecté un '1' ": 01
 - "J'ai détecté deux '1' ": 10
 - "J'ai détecté trois '1' ": 11
- C'est un choix à faire



On obtient le diagramme d'état

Exemple

- On peut maintenant créer le tableau d'état:
 - Les entrées sont les états présents et X
 - Les sorties sont les prochains états et Y

$Q_1(t)$	$Q_0(t)$	X	$Q_1(t+1)$	$Q_0(t+1)$	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

3 sorties, 3 tables de Karnaugh

Exemple

- On fait les tables de Karnaugh

$Q_1(t)$	$Q_0(t)$	X	$Q_1(t+1)$	$Q_0(t+1)$	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

$Q_1(t+1)$

X \ Q_1Q_0	00	01	11	10
0				
1		1	1	1

$$XQ_0 + XQ_1$$

$Q_0(t+1)$

X \ Q_1Q_0	00	01	11	10
0				
1	1		1	1

$$XQ_1 + X\overline{Q_0}$$

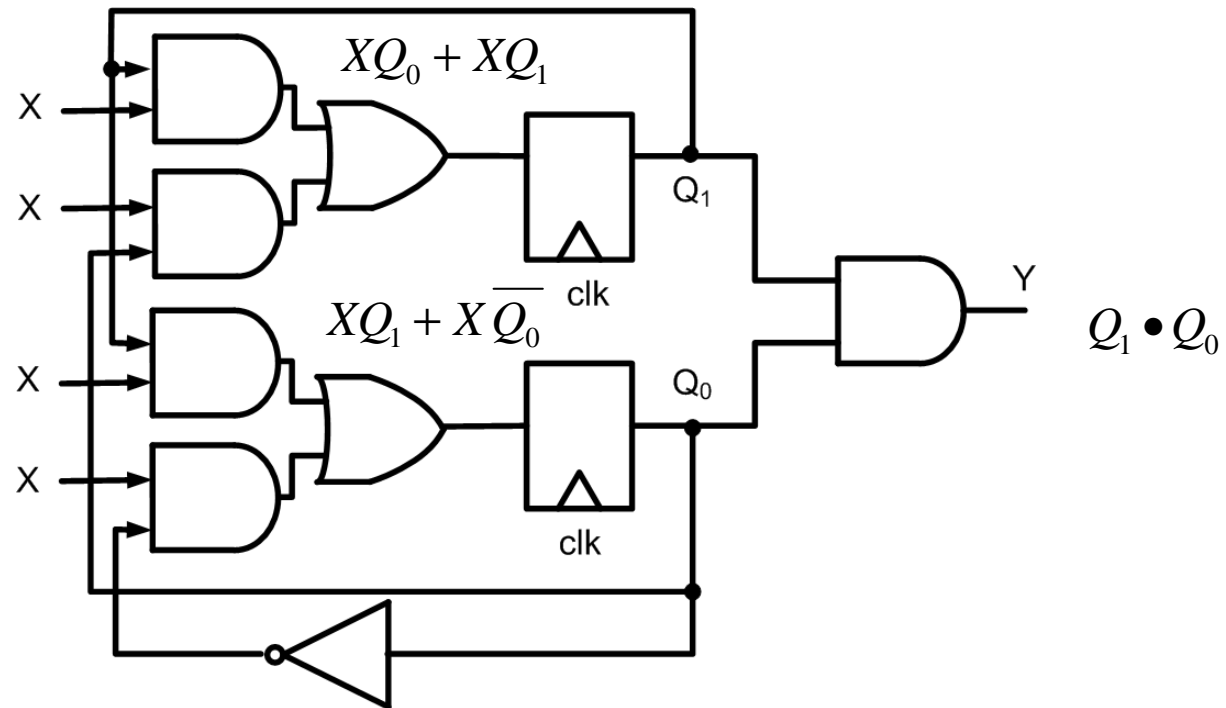
Y

X \ Q_1Q_0	00	01	11	10
0			1	
1			1	

$$Q_1 \cdot Q_0$$

Exemple

- Avec ces équations, on peut dessiner le circuit



Vérification

- Il est important de pouvoir vérifier que notre design est bon
- Pour ce faire, il existe plusieurs méthodes:
 - Analyser le circuit résultant pour voir si ça donne le même diagramme d'état
 - Mettre des valeurs typiques aux entrées et faire propager les signaux
 - Simuler le circuit si on a accès à un ordinateur...

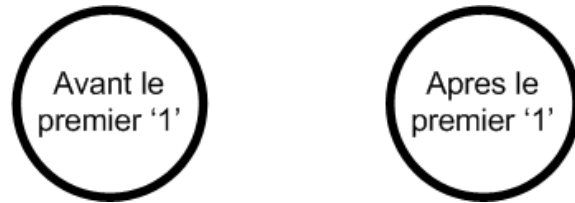
Exemple (seul)

- Concevez un circuit qui génère le complément à 2 d'un nombre en entrée
 - Les bits arrivent un par un
 - Ça génère une sortie à mesure qu'il reçoit les bits
 - Il reçoit le bit le MOINS significatif en premier
- Truc intéressant:
 - Pour faire un complément à 2, la sortie est égale à l'entrée JUSQU'À ce qu'on rencontre le premier 1
 - APRÈS CE '1', c'est l'inverse de l'entrée ('1' pas inversé)

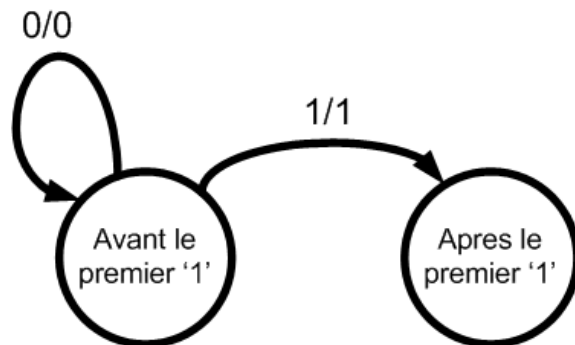
10100  01100

Exemple (seul)

- On fait le diagramme d'états:
 - On essaie d'utiliser 2 états

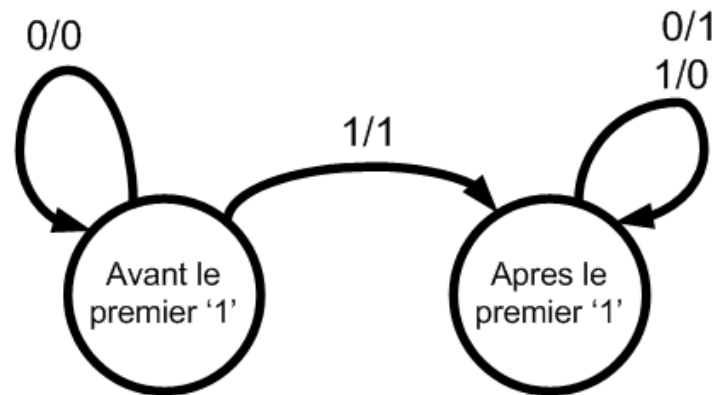


- Dans le premier état:
 - Si l'entrée est 0, la sortie est 0 et il reste là
 - Si l'entrée est 1, la sortie est 1 mais il passe au prochain état



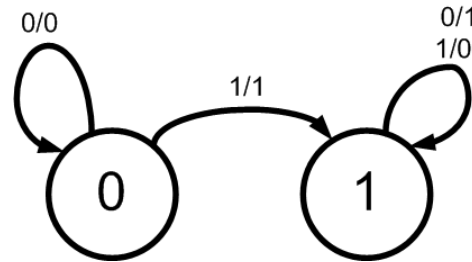
Exemple (seul)

- Dans le deuxième état:
 - Si l'entrée est 0, la sortie est 1
 - Si l'entrée est 1, la sortie est 0
 - Il restera toujours dans cet état (sauf quand la personne pèse sur "reset")



Exemple (seul)

- Avec 2 états, l'un va s'appeler '0' et l'autre va s'appeler '1':



- On fait le tableau d'états:

Q (t)	X	Q (t+1)	Y
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

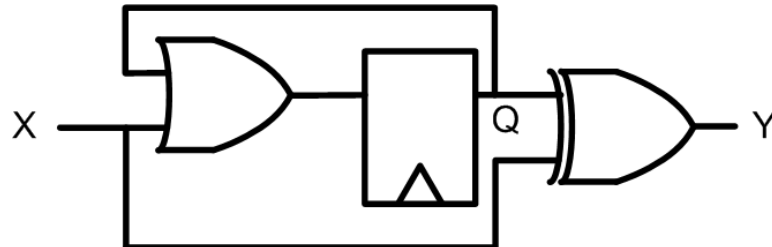
Exemple (seul)

- Par le tableau d'états, on voit que:
 - $Q(t+1)$ est déterminé par un OU
 - Y est déterminé par un OU-exclusif

Q(t)	X	Q(t+1)	Y
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

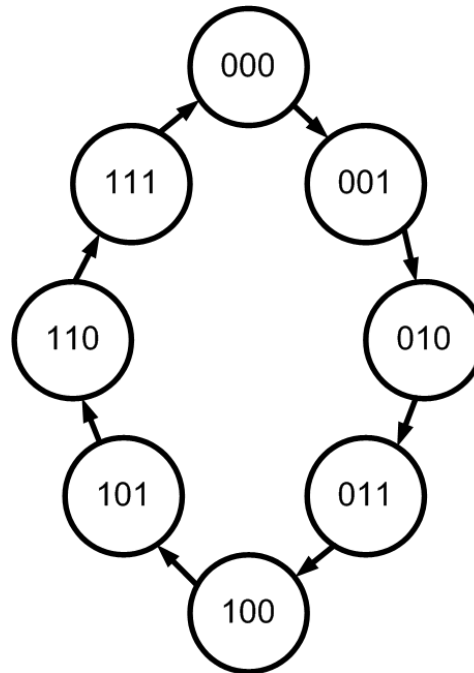
$$Q(t+1) = Q + X$$

$$Y = Q \oplus X$$



Exemple (seul)

- On commence par le diagramme d'états
 - Ça passe par tous les états allant de 0 à 7...



Exemple (seul)

- On fait le tableau d'états...

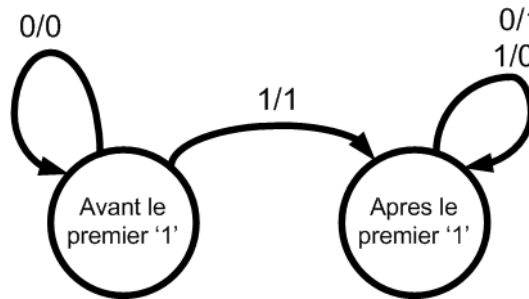
$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Implantation “one-hot”

- On est capable d’implanter le circuit avec un encodage “one-hot”
 - Ça ne demande pas de connaissances supplémentaires
- À la place d’avoir N flip flops pour 2^N états, on a N flip flops pour N états
 - Dans plusieurs cas, ça simplifie la logique combinatoire
 - Dans l’exemple précédent (avec un XOR et un OU), j’ai l’impression qu’il n’y aura pas d’avantage

Implantation "one-hot"

- On reprend les étapes à partir du diagramme d'états
 - Cette fois, on a 1 flip flop par état: on a donc 2 flip flops



- Le tableau des transitions sera différent:
 - Quand un état est à 1, les autres doivent être 0
 - Il y a donc un seul qui peut être actif à la fois

En one-hot, les états sont "01" et "10"...

Implantation "one-hot"

- On fait le tableau d'états

$Q_1(t)$	$Q_0(t)$	X	$Q_1(t+1)$	$Q_0(t+1)$	Y
0	0	0	X	X	X
0	0	1	X	X	X
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	1
1	0	1	1	0	0
1	1	0	X	X	X
1	1	1	X	X	X

Implantation "one-hot"

- On fait les tables de Karnaugh

X \ Q ₁ Q ₀	00	01	11	10
0	X		X	1
1	X	1	X	1

$$X + Q_1$$

X \ Q ₁ Q ₀	00	01	11	10
0	X	1	X	
1	X		X	

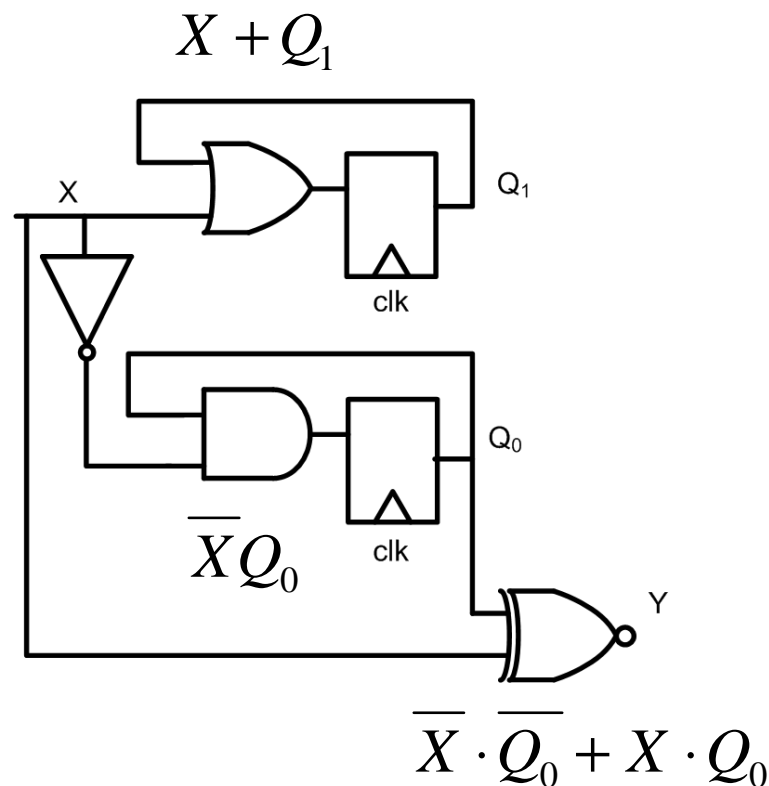
$$\overline{X}Q_0$$

X \ Q ₁ Q ₀	00	01	11	10
0	X		X	1
1	X	1	X	

$$\overline{X} \cdot \overline{Q_0} + X \cdot Q_0$$

Implantation "one-hot"

- On dessine le circuit logique:
 - Dans ce cas-ci, il n'y a aucun bénéfice à utiliser l'encodage one-hot!



Machine distributrice

- Machine distributrice de friandises
- Accepte 0.25\$ en entrée seulement
 - Entrée = 1 s'il y a 0.25\$ et Entrée = 0 sinon
- Donne un bonbon quand le total arrive à 1\$
 - Sortie = 1 quand le total est 1\$, Sortie = 0 sinon
- Machine de Moore

Contrôle de la qualité

- Mise en caisse automatique de bouteilles
 - Il y a 4 bouteilles dans une caisse.
- Caméra qui détecte si une bouteille est brisée
 - Entrée = 1 si brisée, entrée = 0 sinon
- On avertit à la 4^e bouteille s'il y a un problème
 - Sortie = 1 au 4^e cycle si au moins 1 bouteille brisée
 - Sortie = 0 le reste du temps
- Machine de Mealy

Détecteur de front montant

- Concevez le circuit pour un détecteur de front-montant:
 - La sortie est 1 seulement si l'entrée précédente est 0 et l'entrée présente est 1
 - Utilisez la machine de moore (sortie déterminée par l'état seulement)

UART

- Universal Asynchronous Receiver Transmitter
- Protocole de communication série:
 - Envoie séquentiellement les données
 - Envoie à une fréquence pré-définie
- Pour communiquer, les 2 bords doivent être en accord sur:
 - La vitesse
 - Le format

UART

- Il existe plusieurs vitesses communes:
 - 300bps, 1200bps, 2400bps... 115200bps
 - La vitesse de transmission est synchronisée par l'horloge
- Le format commun:
 - 8N1
 - 8 bits de données
 - Pas de bit de parité
 - Stop bit de 1

UART

- Quand aucune donnée n'est envoyée, le transmetteur envoie '1'
 - Pour commencer la transmission, on envoie un Start bit ('0')
 - Par la suite, on envoie le bit le moins significatif (LSB)
 - On envoie jusqu'au bit le plus significatif (MSB)
 - On envoie le Stop bit ('1')

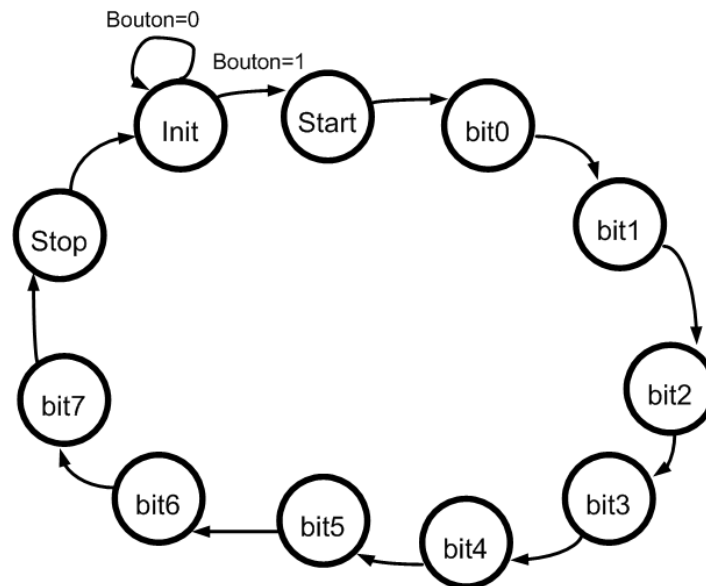
UART

- Au laboratoire, il sera question d'envoyer un signal du FPGA vers l'ordinateur par UART:
 - Vitesse de 57600 bps
 - 8N1
 - Déclencher l'envoi par un bouton poussoir
- L'information est envoyée sous forme ASCII
 - American Standard Code for Information Interchange
 - Code de 8 bits pour représenter les caractères

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
32	20	40	[space]	64	40	100	@
33	21	41	!	65	41	101	A
34	22	42	"	66	42	102	B
35	23	43	#	67	43	103	C
36	24	44	\$	68	44	104	D
37	25	45	%	69	45	105	E
38	26	46	&	70	46	106	F
39	27	47	'	71	47	107	G
40	28	50	(72	48	110	H
41	29	51)	73	49	111	I
42	2A	52	*	74	4A	112	J
43	2B	53	+	75	4B	113	K
44	2C	54	,	76	4C	114	L
45	2D	55	-	77	4D	115	M
46	2E	56	.	78	4E	116	N
47	2F	57	/	79	4F	117	O
48	30	60	0	80	50	120	P
49	31	61	1	81	51	121	Q
50	32	62	2	82	52	122	R
51	33	63	3	83	53	123	S

UART

- La transmission UART se fait de la manière suivante:
 - Init a une sortie de 1
 - Start a une sortie de 0 (Start bit)
 - Stop a une sortie de 1 (Stop bit)



UART

- Par exemple, pour envoyer un point d'exclamation:
 - On cherche son code ASCII avec Google (0x21)
 - On implante la machine à états

