

CEG 3555: Systèmes Numériques II  
(Automne 2023)  
Prof. Rami Abielmona  
Devoir #3: *Concepts de Circuits Séquentiels*

16 Octobre, 2023

## Objectif and Date Due

L'objectif de ce devoir est de démontrer une compréhension de concepts de circuit séquentiels. La date due de ce devoir est **Lundi 06 Novembre, 2023 à 20:00**, et doit être soumis sur Brightspace.

## Question I

Cette question se concerne avec la réduction de table d'état en utilisant la *méthode de division d'équivalence à classe*. La partie c est question 6.18 de votre manuel.

### Partie a

Minimiser la table d'état 1, et démontrez vos cloisons et votre table finale réduite.

Présent état $y_2 y_1 y_0$	Prochain état		Rendement	
	$x = 0$ $Y_2 Y_1 Y_0$	$x = 1$ $Y_2 Y_1 Y_0$	$x = 0$	$x = 1$
A	B	C	0	0
B	C	D	1	0
C	B	E	0	0
D	F	E	0	0
E	G	A	0	0
F	F	H	1	0
G	A	D	1	0
H	A	F	1	1

Table 1: Table d'état de question Ia

## Partie b

Minimiser la table d'état 2, et démontrez vos cloisons et votre table finale réduite.

<i>Présent état</i> $y_2y_1y_0$	<i>Prochain état</i>		<i>Rendement</i>
	$x = 0$	$x = 1$	
	$Y_2Y_1Y_0$	$Y_2Y_1Y_0$	
A	B	C	1
B	D	E	0
C	A	F	1
D	E	C	0
E	G	H	1
F	B	H	1
G	D	F	0
H	F	E	1

Table 2: Table d'état de question Ib

## Partie c

Dans la figure 1, supposez que les sorties non-spécifiées dans les états B et G sont 1 et 0, respectivement. Réduisez la table d'état pour cette machine de Mealy, et démontrez vos cloisons et votre table finale réduite.

Present state	Next state		Output $z$	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	B	C	0	0
B	D	—	0	—
C	F	E	0	1
D	B	G	0	0
E	F	C	0	1
F	E	D	0	1
G	F	—	0	—

Figure 1: Table d'état de question Ic

## Question II

Cette question se concerne avec la conception en utilisant la méthode de FSM. Les parties a et b sont questions 6.37 et 6.38 de votre manuel.

## Partie a

Le FSM de l'arbitre défini en section 6.8, et reproduit en figure 2, peut causer le troisième dispositif à ne jamais obtenir le bus si les dispositifs 1 et 2 demandent des *request* sans interruption. Modifiez le FSM pour s'assurer que le dispositif 3 obtiendra le bus, tels que s'il demande un *request*, les dispositifs 1 et 2 vont être entretenus seulement une fois avant qu'on accorde le dispositif 3 sa demande.

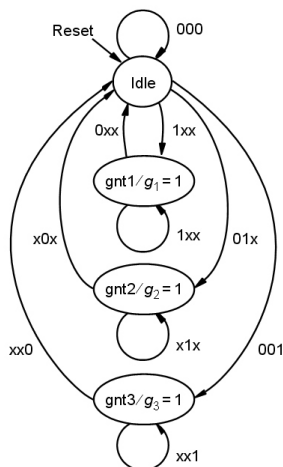


Figure 2: FSM de l'arbitre décrit dans le manuel

## Partie b

Écrivez le code en VHDL du FSM que vous venez de concevoir en partie a de ce problème.

## Partie c

Vous êtes fournis avec ce problème suivant:

Un circuit séquentiel contient une entrée,  $w$ , et une sortie,  $z$ . Sa fonction est d'analyser chaque séquence d'entrée et de produire un 1 si la séquence contient seulement un groupe de 0s. Par exemple, si notre séquence d'entrée est 11000011, alors  $z = 1$  et si la séquence est 100101, alors  $z = 0$ .

Après extraire le diagramme d'état ou table d'état, ou les deux, des spécifications précédentes et avoir choisi une machine de Moore ou de Mealy, votre tâche est de concevoir et démontrer la table de transition, les équations de conception et le circuit final qui réalisent la fonction, en utilisant des **bascules de type D seulement**.

### Question III

Cette question se concerne avec les tables de transition d'états.

#### Partie a

La table d'état démontrée dans la table 3 est pour un circuit synchrone séquentiel. Choisissez un encodage et attribution spécifique pour les états, et déterminez les équations d'excitation et de conception pour le circuit en utilisant des bascules de **type D et S-R**.

<i>Présent</i> état $y_2y_1y_0$	<i>Prochain état</i>		<i>Rendement</i>	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
	$Y_2Y_1Y_0$	$Y_2Y_1Y_0$		
A	B	A	0	0
B	A	C	1	0
C	D	A	0	0
D	D	E	1	0
E	C	D	1	1

Table 3: Table d'état de question IIIa

#### Partie b

Pour atteindre une table de transition, on doit choisir une attribution d'état plus ou moins efficace, qui produit des expressions simples pour les variables de prochain-état et les sorties de système. Il existe des règles pour essayer de nous aider dans nos choix:

**Règle I** Deux ou plus présents états qui ont le même prochain-état pour une combinaison spécifique d'entrée devraient être rendus adjacent;

**Règle II** Pour n'importe quel présent état et deux combinaisons adjacentes d'entrée, les deux prochains états devraient être rendus adjacent;

**Règle III** Deux ou plus présents états qui produisent le même symbole de sortie, c.-à.-d. 0 ou 1, pour une combinaison spécifique d'entrée devraient être rendus adjacent. Cette règle doit seulement être appliquée à un des symboles de sortie.

Répétez la section a de ce problème, et essayez de trouver une attribution qui satisfait autant que possible les contiguités d'état recommandées par les règles données ci-dessus. Comparez vos expressions à ceux dans la partie a. Utilisez la même méthode discutée en classe et démontrez votre **carte d'attribution d'état finale**.