20/12/2021

Exercice sur les Automates à Etats Finis (FSM)

Sujet

Dans les transmissions téléphoniques à grande distance, les informations transitent sous forme numérique en série (un bit à la fois) et au rythme d'une horloge. De manière à éviter de longues suites de '1' (ce qui correspond à une ligne au repos en transmission asynchrone), et par conséquent des composantes continues que la ligne téléphonique risque de filtrer, le code binaire est transformé en 3 niveaux de tension sur la ligne (tel que sur un câble coaxial par exemple) :

- un niveau logique '0' correspond toujours à une tension nulle
- les niveaux logiques '1' sont représentés par des impulsions, qui durent une période de l'horloge de transmission, alternativement positives et négatives, d'où le nom du code.

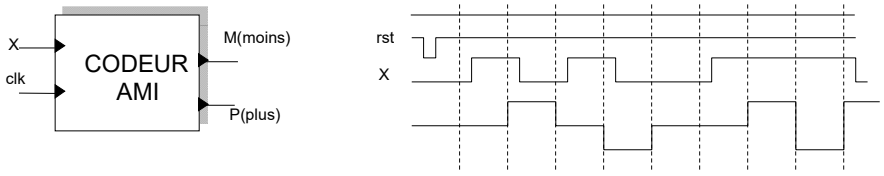


Figure 1 : Symbole et chronogramme du codeur AMI (Alternate Mark Inversion)

Comme le montre la figure 1, la création d'impulsions alternées s'obtient à l'aide de deux signaux de sortie : *plus(P)* et *moins(M)* respectant la convention suivante :

si X= 0 P=M=0
si X=1 P=1,M=0 ou P=0,M=1 en alternance.

La figure ci-dessous décrit le fonctionnement séquentiel à partir d'un diagramme de quatre états sur le principe de la machine de MOORE. L'entrée x n'étant pas considérée comme synchronisée avec l'horloge *clk* (même si bien entendu la durée d'un bit correspond à la période de *clk*), il est préférable d'établir le comportement du circuit à partir d'un modèle de machine de Moore pour obtenir des signaux corrects en sortie. Deux états doivent être distingués : un nombre de 1 paire (sortie plus à 1), un nombre de 1 impaire (sortie plus à 0). Une entrée à 1 conduit l'automate d'un état à l'autre en alternance. Lors de l'arrivée d'un ou plusieurs zéros, il faut pouvoir conserver en mémoire si le nombre de 1 courant était pair ou impair pour une obtenir une génération correcte des sorties lors d'une réapparition d'un 1. On considère le circuit dans l'état initial *zp* à l'initialisation, mais il s'agit d'une simple convention sur les sorties à appliquer au départ.

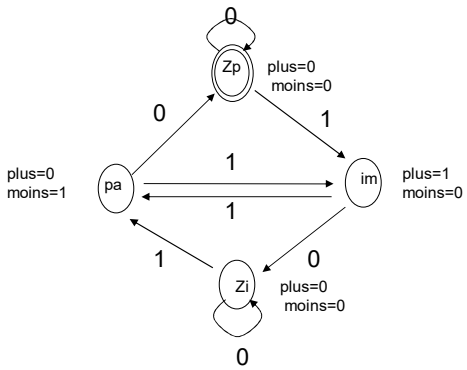


Figure 1: Diagramme des états du circuit

Questions:

A. Dressez la table des états et proposez un codage binaire adjacent des états à partir des règles heuristiques.

B. Effectuez la synthèse de ce circuit en bascules JK à partir de ce codage en recherchant

- La table des transitions
- Les équations d'entrée des FF et des sorties
- Le bloc diagramme.

C. Proposez un codage des états par les sorties (ou codage de Moore) de manière à optimiser le temps de décodage des sorties par rapport à l'horloge.

D. Donnez la table des transitions pour donner suite à ce codage de Moore.

E. Donnez le bloc diagramme d'une solution obtenue par codage one hot.

F. Pour ce circuit de la question E, calculez les délais internes au circuit (table des délais) en considérant:

- 3.5 (TcoFF)
- 2.4 ns (ET/OU à deux entrées)
- 1 ns (INV)

G. Déterminez les périodes minimales des chemins à respecter en considérant :

- un input délai de 3.5 ns
- un output délai de 5 ns
- un temps de setup (T_{su}) de 0.1 ns

Correction

A. La table des états est la suivante:

y_1y_0	x=0	x=1	P	M
Zp	Zp	Im	0	0
Pa	Zp	Im	0	1
im	Zi	Pa	1	0
Zi	Zi	Pa	0	0

D'après la règle heuristique Ia, les états *zp* et *pa* doivent être codés avec des combinaisons adjacentes car *zp* et *im* sont des états suivants identiques à ces deux états. Même chose avec *im* et *zi* car *zi* et *pa* sont des états suivants identiques à ces deux états. La règle dit également que, si possible, les états suivants de ces lignes, *zi* et *im* ainsi que *zi* et *pa* devront être codés avec des codes adjacents. Il est évident à ce stade qu'il n'existe plus de degrés de liberté. On peut alors adopter le codage :

zp (**00)**
pa (**01)**
im (**10**)
zi **(11**)

ce qui permet de satisfaire toutes les contraintes d'adjacences.

B. Tables de transition et d'excitation pour le codage binaire:

			x=0		x=1			
y_1y_0 \ x	0	1	J_1 K_1	J_0 K_0	J_1 K_1	J_0 K_0	P	M
00	00	10	0 X	0 X	1 X	0 X	0	0
01	00	10	0 X	X 1	1 X	X 1	0	1
10	11	01	X 0	1 X	X1	1 X	1	0
11	11	01	X 0	X 0	X1	X 0	0	0

A partir de la table des excitations et simplifications éventuelles par des tables de Karnaugh, les équations d'entrées des bascules s'écrivent :

$J_1 = x$
$K_1 = x$
$J_0 = y_1$
$K_0 = \overline{y_1}$

et celles des sorties :

$P = y_1.\overline{y_0}$
$M = \overline{y_1}.y_0$

Le bloc diagramme se détermine très facilement:

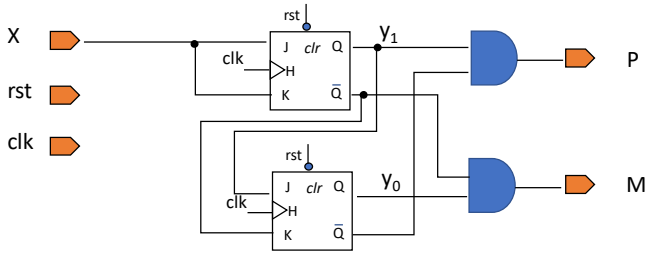


Figure 2: Bloc diagramme du circuit (codage binaire)

C. Dans le codage par les sorties, on cherche à coder les états à l'aide des combinaisons des sorties. On peut observer sur la table des états que les combinaisons des sorties sont différentes sauf pour *Zp* et *Zi*:

Etat	P M
Zp	0 0
Zi	0 0
pa	0 1
im	1 0

Il faut donc un bit supplémentaire pour dissocier le code à affecter à ces deux états. On choisit ici de mettre ce bit supplémentaire en tête du code (par exemple) et on peut adopter 110 pour *im* et 001 pour *pa* (ce qui permet de respecter en partie les adjacences définies par la règle heuristique Ia). On note que du fait de ce codage, les sorties ont automatiquement les équations :

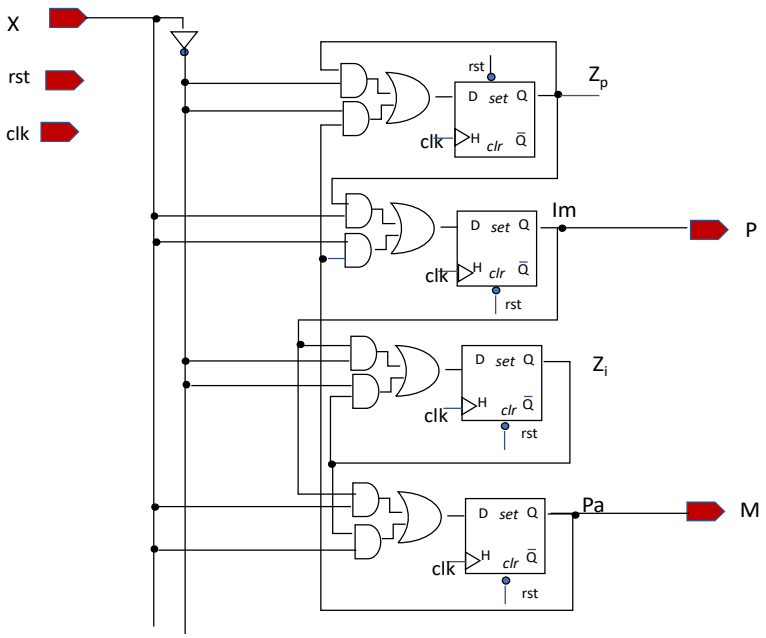
$P = Y_1$
$M = Y_0$

D. La table des transitions après codage par les sorties s'écrit :

$y_2y_1y_0$ \ x	0	1	P M
000	000	100	0 0
001	000	100	0 1
110	100	001	1 0
100	100	001	0 0

On observe que le codage des états par les sorties réclame ici davantage de logique séquentielle (3 FF ou lieu de 2 pour le codage binaire) mais une amélioration des performances est possible par rapport au chemin menant vers les sorties, car il n'y a plus de logique combinatoire entre les FF et les sorties.

E. Avec un codage une bascule par état, la synthèse est directe car on peut déterminer le schéma d'implémentation directement à partir du diagramme des états. Toutes les flèches arrivant sur un état sont implémentées par un OU. Un ET permet d'implémenter les conditions de chaque flèche (état et entrée). Ne pas oublier de mettre le rst sur la FF de l'état initial.



F. Le délai clk→clk est de 3.5(TcoFF)+2.4(Et2)+2.4(Ou2)=8.3 ns.

Le délai X→clk est de 1.0(Inv)+2.4(Et2)+2.4(Ou2)=5.8 ns.

Le délai clk→Z est de 3.5 ns (Tco FF) et on peut observer que ce délai (temps de décodage des sorties) est minimal comme dans le codage de Moore. Mais la solution one hot est plus couteuse (4 FF au lieu de 3 précédemment).

On peut noter qu'il n'y a pas nécessité de calculer le délai X→Z car c'est une machine de Moore.

G. Si on considère un input delai de 3.5 ns, un T_{su} de 0.1 ns et un output délai de 5 ns, on obtient pour les trois types de chemins :

- T_{min} (clk→clk)= délai clk→clk(8.3)+T_{su}(0.1)= 8.4 ns
- T_{min} (X→clk)= délai X→clk(5.8)+input délai(3.5)+T_{su}(0.1)=9.4 ns
- T_{min}(clk→Z)=délai clk→Z(3.5)+output délai(5)=8.5 ns

Le chemin le plus long est X→clk. Ce délai permet de calculer la fréquence maximale du circuit qui est de $1/9.4*10^{-9}$, soit environ 106 Mhz.