

Institut National
de formation
en Informatique

Janvier 2002.

EMD N° 1 de Structure Machines, 2^{ème} année.

Laminovich

Durée: 2 heures.

Documents interdits

ATTENTION!!!

Rédigez les parties I et II sur des copies séparées.
Les exercices appartenant à des parties différentes et rédigés
sur la même copie ne seront pas corrigés.

Partie I:**Exercice I: (5 points)**

On désire réaliser une FIFO qui pourra être lue par deux processeurs différents P0 ou P1. Les mots de la FIFO sont composés de deux champs:

Info	NP: N° du processeur qui doit recevoir l'info
------	-----------------------------------------------

Lorsqu'un mot est écrit dans la FIFO, les deux champs sont écrits en même temps.

La lecture dans la FIFO est réalisée par un circuit CL. Lorsqu'un processeur demande une lecture, CL vérifie d'abord le numéro NP du mot qui se trouve dans l'étage de sortie de la FIFO et le numéro du processeur. La lecture ne peut se faire que si le numéro qui se trouve dans le champ NP du mot est le même que le numéro du processeur.

- Si les deux numéros sont les mêmes, alors le mot est lu et le champ information est envoyé vers le bus du processeur Pi qui a demandé la lecture.
- Si les deux numéros sont différents, alors la lecture ne se fait pas.

Attention: les deux processeurs peuvent demander des lectures en même temps.

Questions:

1. Représenter sur un schéma les entrées et les sorties du circuit CL, ainsi que les connexion avec les différents processeurs, avec les bus de données et avec la FIFO (ne pas détailler l'intérieur de la FIFO ni le contenu du circuit CL).
2. Représenter la table de vérité du circuit CL.
3. Réaliser le circuit CL en utilisant une ROM de 4 entrées et un minimum de circuits.

Exercice II: (5 points)

A- Soit 4 processeurs P0...P3. Chaque processeur possède une FIFO. Un cinquième processeur PM envoie des informations vers les 4 processeurs en les écrivant dans leur FIFO. Chaque mot envoyé par PM est composé de deux champs: un champ "information" et un champ "numéro de processeur".

Un circuit CE s'occupe de l'orientation de l'information vers les FIFO. Il lit le champ numéro de processeur du mot à écrire (c'est ce champ qui permet de choisir la FIFO dans laquelle on doit écrire). CE vérifie alors que la FIFO dont le numéro est choisi n'est pas pleine. Dans ce

cas, il envoie le champ "information" du mot en entrée de la FIFO concernée, et il déclenche une commande d'écriture dans cette FIFO.

Si la FIFO est pleine, le circuit CE envoie toute l'information contenue dans le mot (les deux champs) vers une mémoire associative qui sert à sauvegarder les mots pour ne pas qu'ils soient perdus. Il déclenche alors un algorithme d'écriture de toute l'information dans la mémoire associative.

- On suppose que la mémoire associative est suffisamment grande pour ne jamais être pleine;
- On suppose également qu'au début du traitement, tous les mots de la mémoire associative sont à "00000000.....".

Questions:

1. Représenter sur un schéma les entrées et les sorties du CE en détaillant tous les signaux qui le connectent avec le processeur PM, avec les FIFO des processeurs Pi, avec les différents bus, et avec la mémoire associative.
2. Réaliser CE avec un minimum de circuits.

ATTENTION!!!

Soignez la présentation,
un exercice illisible ne sera pas corrigé

ATTENTION!!!

Soignez la présentation,
un exercice illisible ne sera pas corrigé

Partie II:**Suite de l'exercice II:****Question: (2 points)**

Donner toutes les instructions du programme qui permet d'écrire les deux champs d'un mot dans la mémoire associative ~~lorsque l'une des FIFO est pleine.~~

B- Lorsque le processeur PM a terminé l'envoi de tous les mots, il déclenche un algorithme qui permet d'envoyer tous les mots stockés dans la mémoire associative vers les processeurs P_i , l'un après l'autre: on lit d'abord tous les mots de P_0 , puis ceux de P_1 , et ainsi de suite.

Question : (3 points)

Si on suppose que les FIFO des processeurs P_i ne sont jamais pleines, écrire le programme qui lit tous les mots occupés de la mémoire associative dans l'ordre et les envoie successivement aux différents processeurs.

Exercice III: (5 points)

On désire réaliser un circuit qui permet d'éviter le blocage des roues d'une voiture pendant le freinage. Pour cela, on a besoin de la vitesse de chacune des deux roues avant. Supposons que cette vitesse soit codée en binaire sur deux bits:

- $VG1, VG0$ pour la vitesse de la roue gauche.
- $VD1, VD0$ pour la vitesse de la roue droite.

Chacune des roues peut donc prendre 4 vitesses différentes.

Le circuit à réaliser génère deux commandes de freinage FG pour la roue gauche et FD pour la roue droite selon le mode de fonctionnement suivant:

- Si la vitesse des deux roues est la même, les deux commandes FG et FD sont à 0.
- Si la vitesse de la roue gauche est supérieure à celle de la roue droite, on freine la roue gauche seulement ($FG=1, FD=0$)
- Si la vitesse de la roue droite est supérieure à celle de la roue gauche, on freine la roue droite seulement ($FG=0, FD=1$).

Question:

Donner la table de vérité et le schéma de ce circuit en le réalisant à l'aide d'une ROM seulement.

ATTENTION!!!

Rédigez les parties I et II sur des copies séparées.

Bon courage et bonne année