ISEN CSI3 - CIR3 12 novembre 2019

Durée CSI3 : 2h, CIR3 : 1h

SANS DOCUMENT

AVEC CALCULATRICE

ELECTRONIQUE NUMERIQUE

1. Machine à états finis (CSI3 seulement)

Une machine à états finis est utilisée pour commander une vitre électrique de voiture.

- Le protocole d'utilisation est le suivant:
  - Un appui fugitif (inférieur ou égal à 1 sec) sur le bouton M fait monter inconditionnellement la vitre jusqu'à sa position finale haute,
  - Un appui fugitif (inférieur ou égal à 1 sec) sur le bouton D fait descendre inconditionnellement la vitre jusqu'à sa position finale basse,
  - Un appui prolongé (supérieur à 1 sec) sur le bouton M fait monter la vitre et ce tant que le bouton M reste appuyé,
  - Un appui prolongé (supérieur à 1 sec) sur le bouton D fait descendre la vitre et ce tant que le bouton D reste appuyé,
  - On supposera qu'un appui simultané sur M et D est mécaniquement impossible.

- La détection de position de la vitre est réalisée par deux capteurs PFB et PFH pour les positions finales basse et haute respectivement. La vitre ne doit bien entendu jamais dépasser les positions finales basse et haute.

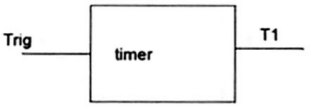
- L'appui sur un bouton ou l'actionnement d'un capteur de position génère un état logique 1.

- Le moteur est commandé selon le tableau de la figure 1

- Pour gérer les délais, vous disposez d'un bloc timer qui fournit une impulsion T1 à l'état 1 durant 1 seconde après déclenchement par l'entrée Trig (Figure 2).

MC1	MC0	état
0	0	arrêt
0	1	descente
1	0	montée
1	1	combinaison interdite

Figure 1



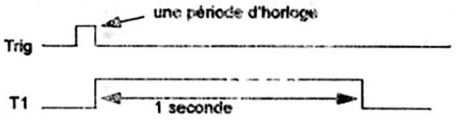


Figure 2

a/ Etablissez la liste des entrées et sorties de la machine

b/ Etablissez le graphe d'état de cette machine. On supposera que la machine démarre sur un état connu pour lequel la vitre est en position fermée. Par souci de clarté :

- vous donnerez un nom **pertinent** aux différents états.
- la valeur des sorties ne sera pas notée sur le graphe mais sur le tableau de la page 3.

Par convention, pour une variable X, on notera X quand X=1 et $\overline{X}$ quand X=0. Toute variable non spécifiée est considérée comme étant indifférente. Vous préciserez clairement, le cas échéant, la relation logique qui existe entre les variables. Par exemple, pour 3 variables X, Y et Z, la condition X=1 ET Y=0, Z indifférent sera notée $X\overline{Y}$.

c/ Combien de bascules seraient nécessaires au minimum pour la réalisation de cette machine ?

2. Microcontrôleur (CSI3 et CIR3)

1 – Sur le PIC18, en adressage direct, combien l'instruction (qui est codée sur 16 bits) contient-elle de bits d'adresse ? 8
Combien manque-t-il de bits pour connaître l'adresse complète ? 4
De ce fait, en combien de banques la mémoire est-elle découpée ? $2^4 = 16$

(Remarque : ce point a été vu en détail lors de la correction du quizz la semaine dernière.)

2 – On suppose que les registres FSR0H et FSR0L sont respectivement chargés avec les valeurs 0x03 et 0x40. (FSR0H et FSR0L constituent le pointeur n°0. Dans le programme, on utilisera ce pointeur grâce au mot-clé INDF0.)
On exécute le programme suivant :

```
...
MOVLW 0x50      0x50 -> W          ; MOVLW signifie MOVE literal to WREG
MOVWF INDF0     0x50 -> 0x340      ; MOVWF signifie MOVE WREG to f
INCF FSR0L      FSR0L = 0x41       ; INCF signifie INCREMENT f
MOVWF INDF0     0x50 -> 0x341
INCF INDF0      0x51 -> 0x341
INCF FSR0L      FSR0L = 0x42
INCF WREG       0x51 -> W
MOVWF INDF0     0x51 -> 0x342
```

(Remarque : par défaut, pour l'instruction INCF, le résultat est bien placé dans le registre dont on modifie le contenu, et non dans l'accumulateur.)

Ce programme permet-il de charger des valeurs dans la mémoire RAM ou la mémoire Flash ?
Après l'exécution du programme, quelle est la valeur (en hexadécimal) que contient l'accumulateur ? 0x51
Quelle est la valeur (en hexadécimal) que contient le registre à l'adresse 0x340 ? 0x50
Quelle est la valeur (en hexadécimal) que contient le registre à l'adresse 0x341 ? 0x51
Quelle est la valeur (en hexadécimal) que contient le registre à l'adresse 0x342 ? 0x51

3 – Que contient le compteur ordinal (ou Program Counter) ? adresse de la prochaine instru

4 - On exécute la partie de programme suivante (on a préalablement déclaré une variable d'un octet nommée *my_var*):

```
...
        MOVLW 0x03          0x03 -> W
        MOVWF my_var        0x03 -> my_var
my_label
        INCF my_var         0x04 -> my_var passage 1    0x05 -> my_var passage 2
        BTFSS my_var, 0     ; Bit Test File Skip if Set : teste le bit n°0 (le LSB) du registre my_var
                            ; et saute l'instruction suivante s'il vaut 1
        GOTO my_label       passage 1        x passage 2
        ...
```

0000 0
0001 1
0010 2
0011 3
0100 4
0101 5
0110 6
0111 7

-> 0x04

Que contient la variable my_var juste après le 1er passage par l'instruction my_label ?

Combien de fois le programme va-t-il passer par l'instruction my_label, et que vaut la variable my_var après chaque passage ?

au total 2 fois.
passage 1 : 0x04
passage 2 : 0x05

NOM:

PRENOM:

FEUILLE A RENDRE AVEC LA COPIE

Exercice 1

Nom de la sortie → Nom de l'état ↓														