

IF40. Médian automne 2011.

Documentation non autorisée. Calculatrice non autorisée.

La documentation du DSP contrôleur TMS320LF2407 nécessaire pour la résolution des exercices est donnée en annexe.

Exercice 1 : instructions assembleur.

Pour chaque question de l'exercice 1, vous devez obligatoirement indiquer votre démarche. Dans cet exercice, toutes les questions portent sur le fonctionnement du DSP contrôleur TMS320LF2407

- 1.1) Après l'exécution de l'instruction suivante **OR #010FFh**, indiquer le contenu de l'accumulateur sachant que le contenu de celui-ci avant l'instruction est ACC=50FF0000(h).
- 1.2) Après l'exécution de l'instruction suivante **AND #010FFh,8**, indiquer le contenu de l'accumulateur sachant que le contenu de celui-ci avant l'instruction est ACC=50FF0000(h).
- 1.3) **Compléter** le tableau ci-dessous (cases avec ??) après l'exécution de l'instruction **LACL *,AR2** :

Avant instruction		Après instruction	
ARP	0	ARP	??
AR2	376h	AR2	??
AR0	377h	AR0	??
Data Memory 376h	0200h	Data Memory 376h	??
Data Memory 377h	0201h	Data Memory 377h	??
Data Memory 378h	0302h	Data Memory 378h	??
ACC	FFFFFF00h	ACC	??

- 1.4) **Compléter** le table ci-dessous (cases avec ??) après l'exécution de l'instruction **MPYA *,AR4** sachant que (PM=0)

Avant instruction		Après instruction	
ARP	3	ARP	??
AR3	30Dh	AR3	??
Data Memory 30Dh	7h	Data Memory 30Dh	??
TREG	8h	TREG	??
PREG	36h	PREG	??
ACC	54h	ACC	??

Exercice 2 : Gestion d'afficheurs sept segments.

Cahier des charges : supposons que 4 digits soient mémorisés dans le tableau ENTREE aux adresses 8000h, 8001h, 8002h, 8003h de la mémoire. Chaque contenu à ces adresses (format 16 bits) se présente sous la forme 0000 0000 0000 $a_3a_2a_1a_0$.

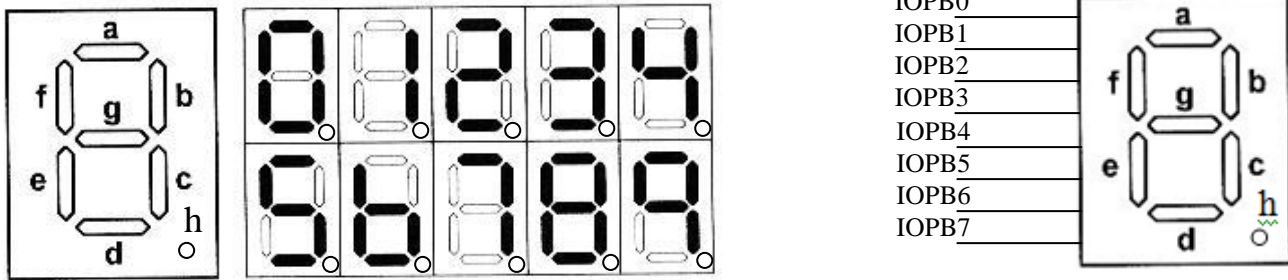
Le chiffre à afficher correspond à la conversion en décimal du nombre binaire constitué par $a_3a_2a_1a_0$ où a_3 représente le bit de poids fort et a_0 le bit de poids faible.

Nous voulons connecter 4 afficheurs au DSP qui devra afficher le contenu des mémoires de données qui viennent d'être définies. Cependant, comme il est prohibitif d'immobiliser 8 fois 4, donc 32 lignes d'entrées sorties, la solution de multiplexage des données sera choisie. Ainsi 8 lignes de données délivreront le digit à afficher et 4 autres lignes l'adresse de l'afficheur.

Pour cela, les afficheurs disposent d'un registre de mémorisation.

2.1) Affichage d'un digit.

L'afficheur 7 segments auxquels on ajoute un point va afficher un nombre compris entre 0 et 9 comme indiqué ci-dessous :



Les segments sont reliés au port d'entrée sortie B (à configurer en sortie) du DSP selon les connexions suivantes :

IOPB0 → a IOPB1 → b IOPB2 → c IOPB3 → d
 IOPB4 → e IOPB5 → f IOPB6 → g IOPB7 → h

Un niveau logique 1 sur une sortie IOPBx allume le segment relié à celle-ci.

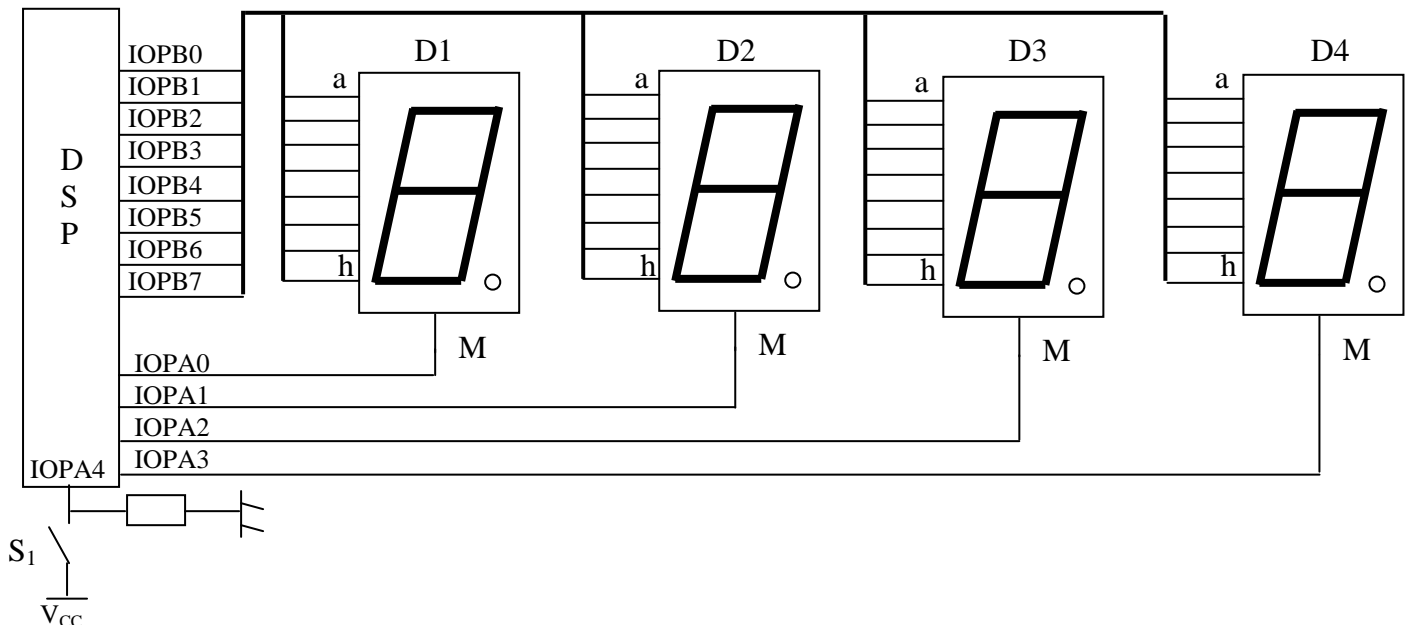
- 2.1.1) **Ecrire** en assembleur le sous-programme INITPB permettant d'initialiser le port B. A la fin de l'initialisation, tous les segments devront être éteints.
- 2.1.2) **Donner** les mots hexadécimaux du port B pour l'affichage de chaque chiffre. **Présenter** votre démarche sous forme d'un tableau.

Les dix mots hexadécimaux obtenus à la question précédente seront stockés dans la mémoire donnée. La zone de mémoire utilisée sera déclarée par la variable TABLE.

- 2.1.3) **Ecrire** en assembleur le sous-programme REMPLISSAGE qui va stocker les mots hexadécimaux dans la zone mémoire.

2.2) Multiplexage de l'affichage.

Schéma :



L'affichage des quatre chiffres se fait à chaque front montant sur l'entrée IOPA4 (provoqué par un appui sur S₁).

Lorsque le front montant est détecté (par le sous programme FRONT qui met la variable FM à 1), D1 doit afficher le chiffre stocké à l'adresse 8000h, D2 le chiffre de l'adresse 8001h, D3 le chiffre de l'adresse 8002h et D4 le chiffre de l'adresse 8003h.

On ne veut afficher que des chiffres du système décimal. Si un des quatre chiffres à afficher est supérieur à 9, on n'affiche aucun chiffre, seuls les quatre points seront allumés.

Si l'entrée M d'un afficheur se trouve à l'état logique 0, l'afficheur affiche le chiffre correspondant aux valeurs présentes sur les sorties du port B (IOPB0 à IOPB7).

Si l'entrée M d'un afficheur se trouve à l'état logique 1, l'afficheur mémorise le dernier digit et l'affiche en permanence.

Les broches IOPA0, IOPA1, IOPA2 et IOPA3 doivent être configurées en sortie.

- 2.2.1) **Ecrire** en assembleur le sous-programme INITPA permettant d'initialiser les broches du port A utilisées.
- 2.2.2) **Ecrire** l'organigramme du sous-programme FRONT qui détecte un front montant sur l'entrée IOPA4 et met la variable FM à 1. Si le front montant n'est pas détecté, la variable FM reste à 0.
- 2.2.3) **Ecrire** en assembleur le sous-programme FRONT.
- 2.2.4) **Ecrire** l'organigramme du sous-programme TEST qui teste si les 4 chiffres à afficher sont compris entre 0 et 9. Si c'est le cas, la variable VALID sera mise à 1 sinon elle reste à 0.
- 2.2.5) **Ecrire** en assembleur le sous-programme TEST.
- 2.2.6) **Ecrire** l'organigramme du sous-programme AFFICHAGE qui affiche les 4 chiffres si les variables FM et VALID sont à 1. Si seule la variable VALID est à 0, alors il n'y aura aucun chiffre affiché, seuls les quatre points seront allumés.
- 2.2.7) **Ecrire** en assembleur le sous-programme AFFICHAGE.
- 2.2.8) **Ecrire** en assembleur le programme principal qui initialise les ports puis appelle en boucle les sous-programmes FRONT, TEST et AFFICHAGE. Vous n'oublierez pas de déclarer les différentes variables utilisées.