

IF40. Médian automne 2013.

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. Toute réponse non expliquée est comptée comme fausse.

Dans cet exercice, toutes les questions portent sur le fonctionnement du DSP contrôleur TMS320LF24LF07

- 1.1) 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=50FFF0F0(h).
- 1.2) **Recopier et compléter** le tableau ci-dessous (cases avec ??) après l'exécution de l'instruction **LACC *0-,AR1** :

Avant instruction		Après instruction	
ARP	2	ARP	??
AR1	376h	AR1	??
AR2	377h	AR2	??
AR0	2h	AR0	??
Data Memory 376h	0200h	Data Memory 376h	??
Data Memory 377h	0201h	Data Memory 377h	??
Data Memory 378h	0202h	Data Memory 378h	??
ACC	0000FF00h	ACC	??

Exercice 2 : valeur minimale d'une campagne de mesures

On utilise le DSP pour calculer la valeur minimale de 5000 mesures préenregistrées en mémoire dans le tableau MESURE (valeurs comprises entre 0 et 1023). La valeur minimale sera stockée dans la variable MIN et ensuite cette valeur sera divisée par 5. Le quotient de la division sera stocké dans la variable RESULTAT.

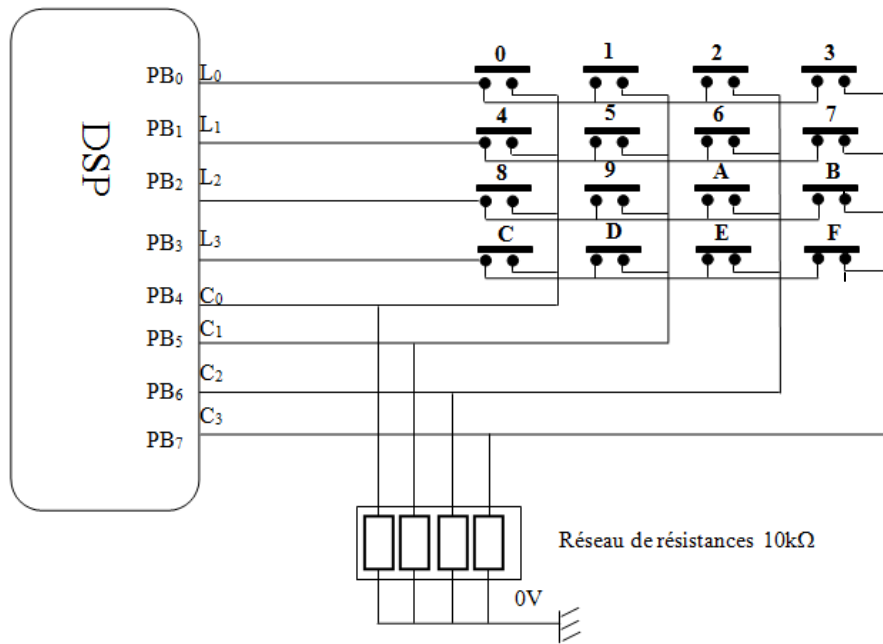
- 2.1) **Ecrire** l'organigramme du sous-programme VALEURMIN qui calcule la valeur minimale et stocke le résultat dans la variable MIN et qui réalise ensuite la division de MIN par 5 et stocke ce résultat dans la variable RESULTAT.
- 2.2) **Ecrire** en assembleur le sous-programme VALEURMIN.

Exercice 3 : scrutation d'un clavier et affichage du numéro de la touche appuyée

Dans cette étude, nous allons réaliser un sous-programme de scrutation de clavier hexadécimal et ensuite un sous-programme affichant la dernière touche appuyée sur un afficheur 7 segments. Le schéma de principe est donné page suivante.

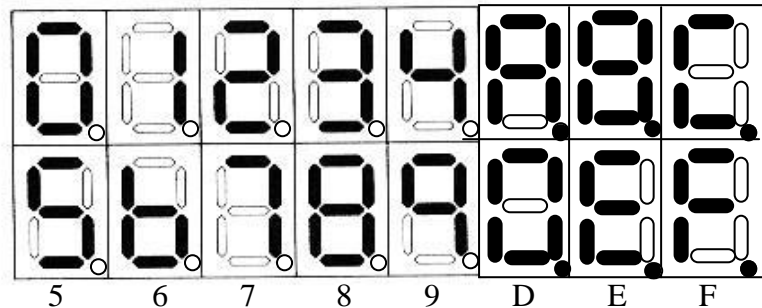
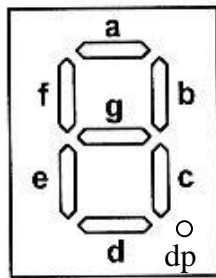
Fonctionnement de la scrutation du clavier : le clavier est relié au port B du DSP. Les broches IOPB0 à IOPB3 sont configurées en sortie et les broches IOPB4 à IOPB7 sont configurées en entrée. Pour scruter le clavier, on envoie un niveau logique un sur la ligne L0 (IOPB0=1) et ensuite on lit le niveau logique sur les colonnes (IOPB4 à IOPB7). Si aucune touche de cette ligne n'est appuyée toutes les colonnes sont à 0, on passe à la ligne suivante. Lorsqu'un niveau logique un apparaît sur une colonne, on peut calculer le numéro de la touche appuyée à partir du numéro de ligne et du numéro de colonne. Ce numéro est stocké dans la variable TOUCHE. Si aucune touche n'est appuyée, on sort du sous-programme.

Schéma du clavier :



Afficheur 7 segments : (segment allumé IOPCx=1). Les broches du port C sont configurées en sortie.

- PC0 → a
- PC1 → b
- PC2 → c
- PC3 → d
- PC4 → e
- PC5 → f
- PC6 → g
- PC7 → dp



Le segment dp est allumé lorsque le symbole à afficher est une lettre.

- 3.1) **Ecrire** en assembleur le sous-programme **INITPORT** qui configure les ports B et C. A la fin de la configuration, toutes les sorties doivent être au niveau logique 0.
 - 3.2) **Proposer** un organigramme du sous-programme **CLAVIER** détectant l'appui sur une touche et stockant dans la variable **TOUCHE** la valeur de la touche appuyée (valeur de 0 à F en hexadécimal : touche 0 appuyée valeur 0 en hexadécimal, touche A appuyée valeur A en hexadécimal etc...).
 - 3.3) **Ecrire** en assembleur le sous-programme **CLAVIER**.
 - 3.4) **Donner** pour chaque touche à afficher (0 à F), l'état des sorties PC7 à PC0. **Donner** les résultats sous forme de tableau avec un code hexadécimal où PC7 représente le poids fort et PC0 le poids faible.
- | Touche | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 | Code hexadécimal |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|------------------|
| | | | | | | | | | |
- 3.5) **Ecrire** en assembleur le sous-programme **TABLE** qui stocke en mémoire les valeurs hexadécimales obtenues à la question précédente.
 - 3.6) **Ecrire** en assembleur le sous-programme **AFFICHEUR** affichant la valeur **TOUCHE** à partir du port C en code 7 segments.
 - 3.7) **Ecrire** en assembleur le programme principal qui initialise les ports et appelle en boucle les sous -programmes **CLAVIER** et **AFFICHEUR**. Vous déclarerez la variable **TOUCHE** et les autres variables nécessaires pour le bon fonctionnement du programme.