

*Institut Supérieur des Etudes Technologiques en Communications de Tunis*

|  |  |
| --- | --- |
| **Enseignant :** | Nadia KHOUJA |
| **Niveau :** | ***Master*** |
| **Date :** | ***Mai 2016*** |
| **Nb pages :** | ***2*** |
| **Documents :** | ***Non autorisés*** |
|  |  |

**Groupe : Smart’Com**

# Examen

Digital Signal Processor

**Durée 1h 30**

**Exercice 1 :**

Analyser le programme assembleur suivant :

.section L1\_code;

.global \_\_program1;

.align 8;

\_\_program1:

L0 = 0;

I0 = R1;

P1 = R0;

P0 = R2;

P2 = 16(Z);

R3 = -1;

R0.L = W[I0++];

LSETUP(OUT\_LOOP\_START,OUT\_LOOP\_END) LC0 = P1 >> 1;

OUT\_LOOP\_START:

LSETUP(IN\_LOOP\_START,IN\_LOOP\_END) LC1 = P2;

IN\_LOOP\_START:

CC = BITTST(R0,0);

R2 = 1(Z);

IF !CC R2 = R3;

R0 = R0 >> 1 || B[P0++] = R2;

OUT\_LOOP\_END:

R0.L = W[I0++];

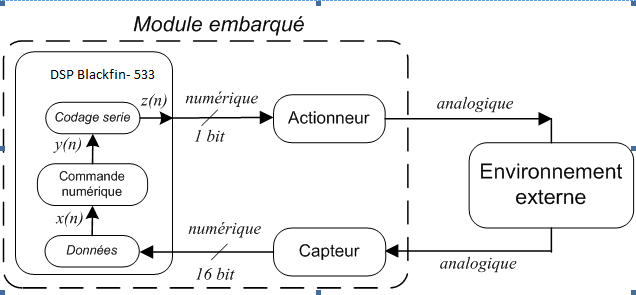
RTS;

NOP;

\_\_program1.end:

**Exercice 2 :**

On se propose d’écrire un programme pour un processeur DSP de type blackfin-533 afin d’implémenter une commande de contrôle numérique d’un actionneur dans un module embarqué. La figure 1 illustre l’architecture globale du module embarqué incluant l’étage numérique, le capteur et l’actionneur.



Le capteur fournit des données numériques codées sur 16 bits à l’étage numérique. Après traitement de ces données au sein du processeur DSP, une commande sera générée et envoyée à l’actionneur qui se chargera de la convertir en analogique.

L’objectif de l’étage numérique consiste à effectuer :

* Le calcul de la valeur de la commande à partir des données collectées du capteur,
* Le codage de la valeur de la commande en un format série,
* La transmission série vers l’actionneur.

Le calcul de la valeur de la commande consiste à en déduire la dérivée du signal d’entrée issue du capteur selon l’équation suivante :

*y(n)=x(n)-x(n-1)*

Le code série *z(n)* est obtenu en codant le signe de *y(n)* sur huit bits selon le procédé suivant :

* Si la dérivée *y(n)* est positive, *z(n)* prend ‘0x00’ comme valeur
* Si la dérivée *y(n)* est négative, *z(n)* prend ‘0xFF’ comme valeur

La transmission vers l’actionneur consiste à remplir un espace mémoire de 8 bits avec les valeurs de *z(n)*. Un mécanisme d’accès mémoire se chargera d’envoyer à l’interface série du processeur connectée avec l’actionneur.

Les données issues du capteur sont numériques et codées sur 16 bits. On assume que ces données sont stockées dans un espace mémoire dédié. On s’intéresse dans cette partie à déclarer les variables qui vont être utilisées par le programme.

**1.** Déclarer un tableau « data » ayant les données à l’entrée *x(n)* codées sur 16 bits suivantes:

Entrée 0x1234

0x5678

0x9011

0xF213

0x1234

0x5678

**2.** Initialiser une variable « Sortie » de type tableaux ayant six données codées sur 8 bits.

**3.** Ecrire l’instruction assembleur qui permettra de télécharger une première valeur du Tableau « Entrée » dans un registre r2. Cette opération permet de charger les 16 bits LSB de l’espace mémoire et complète les 16 bits restants du registre par le bit du signe.

**4.** Ecrire l’instruction assembleur qui permettra de télécharger une deuxième valeur du Tableau « Entrée » sans mise à jour d’adressage dans un registre r3. Cette opération permet de charger les 16 bits LSB de l’espace mémoire et complète les 16 bits restants du registre par le bit du signe.

**5.** Ecrire l’instruction assembleur qui permettra de réaliser l’opération de soustraction entre deux éléments successifs du tableau « Entrée » dans le registre r0.

**6.** Écrire les instructions assembleur qui permettront de réaliser l’opération de codage et de stoker les échantillons *z(n)* dans le tableau « Sortie » Cette opération permet de stocker les 8 bits LSB du registre dans l’espace mémoire.

**7.** Réitérer le traitement précédent pour les six échantillons à l’entrée.

**8.** Réécrire tout le code en un seul programme en réarrangeant toutes les instructions des différentes parties.