ISIMA 1 – Assembleur **Vendredi 22 juin 2012**

Examen final Durée : 1h30

TP, TD, cours polycopié et notes de cours autorisés

Exercice I (5 points)

Répondre aux questions suivantes (5 ou 6 lignes par réponse au maximum) :

1-Est-ce que l'instruction assembleur suivant est correcte ? Expliquez

MOV R0,#0x12345678

2- Les processeurs actuels ont une architecture moins optimisée pour la vitesse pure que les processeurs qui étaient fabriqués il y a quelques années, pourtant ils sont plus puissants... pourquoi ?

3- A quoi sert le JTAG ?

4- Qu'est ce que c'est le Barrel Shifter ? Donner un exemple d'utilisation de ce dernier.

5- Expliquez pourquoi le langage assembleur est indispensable pour le développement d'un système d'exploitation.

Exercice II (15 points)

1. Ecrire un programme en assembleur ARM qui additionne deux tableaux entiers positifs (Tab1 et Tab2) de dimension N (N=8). Chaque élément du tableau est un nombre entier positif codé sur 32 bits.
   Les résultats des opérations doivent être sauvegardés dans le troisième tableau (Tab3) :
   Tab3[i]=Tab1[i]+Tab2[i]
   Les valeurs initiales des éléments des tableaux Tab1 et Tab2 doivent être initialisées (à initialiser en utilisant les directives de l'assembleur) :
   Tab1 = (0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8)
   Tab2 = (Ox8, 0x7, 0x6, 0x5, 0x4, 0x3, 0x2, 0x1).
2. On veut gérer l'accès à une section critique en utilisant un sémaphore binaire 8 bits représenté par la variable SemB. Plusieurs processus ont un accès exclusif à la section critique. La section critique est libre si la variable SemB est égale à 0. Pour indiquer qu'elle est occupée la variable est initialisée à 0xF.
   a- Ecrire un programme en assembleur pour tester si la section critique est libre. A l'aide d'une seule instruction non interruptible, on récupère la valeur de SemB et lui donne la valeur 0xF pour bloquer l'accès.
   Si l'ancienne valeur de SemB est 0, appeler la procédure SecCrit puis débloquer l'accès. Sinon reboucler pour attendre que la section se libère.
   b- Peut-on réaliser cette fonction en utilisant le langage C de base (sans bibliothèque) ? Expliquez.
3. Ecrire un programme en langage assembleur permettant de calculer la somme d'une trame 'Trame' de données (checksum). La trame comporte N octets (N=12) et le résultat de l'addition (somme) est codé sur 16 bits et doit être stocké à la fin de la trame.

La trame doit être initialisée avec les valeurs suivantes (à initialiser en utilisant les directives de l'assembleur) :

Trame[]={0x1, 0x1,0x1, 0x1, 0x1, 0x1, 0x1, 0x1, Ox2, 0x2, 0x2, 0x2, 0x2, 0x2, 0x2, 0x2, 0x3, 0x3, 0x3, 0x3, 0x3, 0x3, 0x3, 0x3}.