

TD n° 3 : CONDITIONNELLES et BOUCLES

On utilise dans ce TD les jeux d'instructions du NIOS II et de l'ARM.

1. Conditionnelles

a) Ecrire la séquence d'instruction qui place dans un registre la valeur absolue du contenu de ce registre

- pour le NIOS II
- pour ARM sans utiliser les instructions conditionnelles, puis en utilisant les instructions conditionnelles.

Pour le NIOS II, on utilisera la pseudo-instruction MOVIA pour charger les adresses dans les registres. Pour l'ARM, on utilisera la pseudo instruction ADR Registre, Adresse.

b) En code ASCII, les caractères 0 à 9 sont représentés sur 8 bits par les configurations hexadécimales : 30 à 39. Si A et B sont chacun un chiffre ASCII, écrire en assembleur (NIOS II puis ARM) la séquence d'instructions qui place dans la variable C la représentation ASCII du plus grand des deux. L'adresse de A est A000 0000_H, celle de B est A000 0001_H et celle de C est A000 0002_H.

2. Boucles

Soient les trois boucles :

B1

```
For (i=0; i<1000; i++)  
  s = s + X[i] + Y[i];
```

B2

```
For (i=0; i<1000; i++)  
  if (X[i]>0  
      s = s + X[i] ;
```

B3

```
For (i=1; i<999; i++)  
  Y[i-1] = X[i] + X[i-1];
```

On suppose que les vecteurs X et Y sont des entiers (32 bits), implantés à partir des adresses 1000 0000_H et 2000 0000_H. La variable s est placée à l'adresse 100_H.

Ecrire le code des trois boucles pour NIOS II et ARM.

3. Boucles (exercice 2)

a) On suppose que le vecteur X contient 1000 entiers (32 bits), implantés à partir des adresses 1000 0000_H. Les variables min et max sont aux adresses 100_H.et 104_H.

Ecrire le programme NIOS II, puis le programme ARM qui place dans les variables min et max les valeurs minimale et maximale du vecteur X[1000] ;

4. Instructions pour évaluation des conditions et branchements

NIOS II (Voir poly)

ARM

CMP , TST	CMP Rs1, Rs2 TST Rs1, Rs2	Rs1-Rs2 → RCC Rs1 and Rs2 → RCC
Instructions arithmétiques avec suffixe S : ADDS, SUBS, etc	SUBS Rd, Rs1, Rs2	Rd ← Rs1 – Rs2 Positionne Rcc
Bcond (LT, LE, GT, GE, EQ, NE...)	Bcond, déplacement	Si cond, alors CP ← NCP +déplacement
BL	BL déplacement	R14 ← NCP CP ← NCP + déplacement
BLcond	BLcond déplacement	Si cond, alors { R14 ← NCP CP ← NCP +déplacement }