

## TD 4 Architecture logicielle Branchements

Dans tout le TD, on considère le jeu d'instructions DIDE.

### 4.1 Format des instructions

- Quelle est l'amplitude des branchements ?

### 4.2 Conditionnelles

- Ecrire la séquence d'instructions qui place dans un registre le maximum des valeurs signées contenues dans deux registres.
- Ecrire une séquence d'instructions qui effectue la comparaison *non signée* de deux entiers en complément à 2 sur 64 bits. Les opérandes sont contenus dans (R3,R2) et (R5,R4) (mot de poids fort respectivement dans R3 et R5). Le résultat est retourné dans R6, avec  $R6 := 1$  si  $R3 : R2 \leq R5 : R4$ , sinon 0.

### 4.3 Boucles While

- Ecrire le code correspondant au fragment de programme suivant :

```
tant que a > b faire
    a := a - b;
ftq
```

où a et b sont des entiers implantés consécutivement à partir de l'adresse 0x00001000. Ici et dans la suite, les registres ne seront rangés en mémoire que lorsque c'est nécessaire.

### 4.4 Boucles Repeat

- Soient les deux boucles :

B1	B2
Pour i = 0, 999	Pour i = 1, 999
s = s + X(i) + Y(i)	Y(i-1) = X(i) + X(i-1)

X et Y sont des vecteurs d'entiers (32 bits), implantés respectivement à partir des adresses 0x10000000 et 0x20000000. s est un entier placé à l'adresse 0x00000100.

Ecrire le code des deux boucles. On écrira d'abord le corps de chaque boucle, puis le test de sortie, puis les initialisations.

- Ecrire le programme qui recherche la première occurrence de la valeur 0x12345678 dans un tableau de 10 entiers. Le résultat est l'indice (à partir de 0) de l'élément de tableau correspondant, et -1 si pas trouvé ; il est retourné dans le registre R1. Le tableau est implanté à partir de l'adresse 0x00001000.