

## Labo 8

### Utilisation du compilateur d'ASM

*Le but de ce laboratoire est de vous familiariser avec le compilateur CASM qui permet d'écrire du code ASM de haut niveau et de le compiler automatiquement vers du VHDL.*

Vous trouverez sur le site les fichiers du labo

Comme dans le labo 7, l'application choisie est la simulation de la trajectoire d'objets dans l'espace en fonction de diverses forces qui agissent dessus.

Pour simplifier les aspects algorithmiques, on suppose qu'à chaque nouvelle image, les opérations suivantes doivent être réalisées pour chaque objet:

- 1) Calcul des forces ( $F_x, F_y$ ) qui agissent sur l'objet (selon l'application)
- 2) Calcul des vitesses :  $V_x = V_x + F_x$ ;  $V_y = V_y + F_y$ ;
- 3) Calcul des positions :  $P_x = P_x + V_x$ ;  $P_y = P_y + V_y$ ;

De plus, si l'objet arrive sur un bord de l'écran, il doit être réfléchi et une partie de sa vitesse est absorbée :  $V_x = -V_x/2$  ou  $V_y = -V_y/2$ ;

Le fichier d'exemple « magnetBall.asm » illustre le cas d'une seule balle qui se déplace sous l'action d'une force en relation avec un autre objet. Le système fonctionne en virgule fixe avec 5 bits après la virgule.

- 1) Copiez les fichiers du labo dans votre répertoire
- 2) Compilez le fichier asm au moyen de la commande :  
`java -jar ~pift6223/FPGA/ASM2d0b.jar -Wall < le path de « magnetBall.asm »>`
- 3) Compilez le projet sous QuartusII et exécutez-le

Ensuite, modifiez le fichier pour réaliser le système suivant :

Deux balles supplémentaires sont ajoutées et elles exercent des forces du même type que la souris. Chaque balle est donc soumise à la somme de trois forces en X et trois forces en Y. Le but du jeu consiste à utiliser la souris pour que la bille rouge passe entre la bille verte et la bille bleue. Un point est marqué (et s'affiche sur le display) chaque fois que cela arrive.

**Bon travail !**