Université Bordeaux 1, Sciences et technologies
UFR Informatique et Mathématiques

Licence d'informatique
Session de septembre 2002

Architecture et Microprocesseurs (AMP)
Le 18 septembre 2002

Durée : trois heures

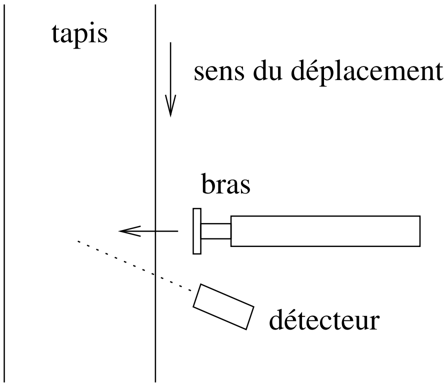
Tous documents autorisés sauf photocopies illégales de livres. Votre voisin n'est pas un document.

Rappel : Soyez bref et clair ! Personne n'est impressionné par la quantité !

Bon courage !

Exercice 1. (7p)

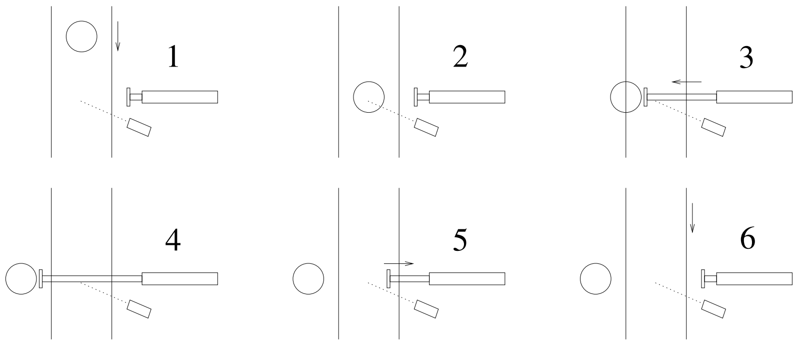
Dans une usine, on souhaite déplacer des objets qui arrivent sur un tapis roulant illustré par la figure suivante :



Lorsqu'un objet arrive devant le détecteur, on souhaite :

1. arrêter le tapis,
2. déclencher le déplacement latéral d'un bras pour pousser l'objet en dehors du tapis,
3. retirer le bras,
4. redémarrer le tapis.

Cette séquence est illustrée par la figure suivante :



L'objectif de l'exercice est de construire un circuit séquentiel capable d'effectuer cette suite d'événements. Nous possédons pour cela une entrée d qui telle que $d = 1$ si et seulement si le détecteur a détecté un objet, et trois sorties : t qui détermine si le tapis se déplace ($t = 1$) ou non ($t = 0$), $b1$ qui détermine si le bras se déplace à gauche ($b1 = 1$) ou non ($b1 = 0$) et $b2$ qui détermine si le bras se déplace à droite ($b2 = 1$) ou non ($b2 = 0$). Pour le déplacement du bras, nous savons qu'il faut deux cycles d'horloge pour un déplacement d'une position extrême à l'autre.

Afin de réaliser le déplacement du bras, il faut une sortie *invisible* (appelée i) dont le but est de déterminer si l'on se trouve dans le premier ou le deuxième cycle de déplacement du bras.

a. **(4p)** Remplir la table d'états suivante :

d	t	b1	b2	i	t' b1' b2' i'
0	0	0	0	0	
0	0	0	0	1	
0	0	0	1	0	
0	0	0	1	1	
0	0	1	0	0	
0	0	1	0	1	
0	0	1	1	0	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	0	1	
0	1	0	1	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	0	1	
0	1	1	1	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	0	1	
1	0	0	1	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	0	1	
1	0	1	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

b, **(3p)** Réaliser le circuit pour le signal $b1$.

Exercice 2. (13p)

Certains langages de programmation (Fortran, par exemple) utilisent souvent des boucles dont le nombre d'itérations est connu avant que l'exécution de la boucle ne commence. Pour ces langages, il est raisonnable pour l'architecture de proposer une façon d'exécuter ces boucles d'une manière rapide.

Une possibilité souvent utilisée est de proposer *un registre supplémentaire* contenant le nombre d'itérations restantes et une *instruction supplémentaire* (appelée LOOP) dont le fonctionnement est :

- tester si le contenu du registre est zéro,
- si oui, stocker l'adresse donnée en argument dans le compteur ordinal,
- décrémenter le registre

(a) [**4p**] Rajouter les circuit nécessaires pour la réalisation de l'instruction LOOP à l'architecture donnée en annexe.

(b) [**3p**] Réaliser le microprogramme de l'instruction LOOP.

(c) [**6p**] Donner le code généré par le compilateur pour l'architecture donnée en annexe pour les instructions suivantes :

```
{
  int x = 0;
  int i;
  for(i = 0; i < n; i++)
    x += 3;
}
```

La valeur de la variable `n` se trouve dans la mémoire centrale. Vous pouvez vous servir de l'ensemble des instructions présentées en cours, ainsi que de la nouvelle instruction LOOP.