Maîtrise Informatique 1998-1999 — Bordeaux I

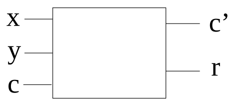
Modèles et Techniques de Programmation

Projet - Common Lisp

On considérera le programme `circuits.lisp` vu en TD (accessible dans `~casteran/MTP/TD6`).

1. Additionneurs:

   (a) Réaliser le circuit de l'additionneur binaire $x + y = r$ avec retenue c' ("*carry*") obtenu à partir du schéma suivant ($c' = xy$ **ou** xc **ou** yc, $r = x$ **xor** $(y$ **xor** $c)$):

   (b) A partir de cet additionneur binaire, écrire la fonction qui construit un additionneur pour nombres binaires entre 0 et $2^n - 1$. Considerant que celui-ci effectue $(x_{n-1}, ...x_0) + (y_{n-1}, ..., y_0) + c = (z_{n-1}, ..., z_0) + 2^n c'$, on fera en sorte que le vecteur des entrées du circuit soient: $(x_{n-1}, ...x_0, y_{n-1}, ..., y_0, c)$ et le vecteur de sortie $c', z_{n-1}, ..., z_0$.
   Par exemple, si $n = 3$, l'addition $3 + 5$ se définit par

   $$x_2 = \mathtt{nil}, x_1 = \mathtt{t}, x_0 = \mathtt{t}, y_2 = \mathtt{t}, y_1 = \mathtt{nil}, y_0 = \mathtt{t}, c = \mathtt{nil}$$

   et

   $$c' = \mathtt{t}, z_2 = z_1 = z_0 = \mathtt{nil}$$

   (c) Simuler et tester.

2. Les classes correspondant aux portes ET et OU binaires existent déjà. Implémenter les classes qui correspondent aux mêmes portes mais avec un nombre arbitraire d'entrées.

3. En l'état actuel, le programme `circuits.lisp` ne fait aucun test de cohérence : en effet, rien par exemple n'interdit de connecter des adresses absurdes entre elles (e.g. connecter la 36-ème sortie d'un composant qui n'en a que deux ou de laisser une entrée d'un composant non connectée). Par conséquent, obtenir des messages d'erreur est possible lors de la simulation et ceux-ci seront de plus difficilement interprétables.

   Modifier `circuits.lisp` afin que le maximum d'erreurs potentielles soient détéctées à la construction d'un circuit. On pourra par exemple utiliser la macro `assert`, modifier la fonction `make-complex-circuit`, etc...

4. *(facultatif: la note 20 peut s'obtenir sans cette question).* Modifier `circuits.lisp` afin qu'il soit possible de faire un test de stabilité d'un circuit C, i.e. de détecter lors d'une simulation de C le cas où aucun changement dans les états des portes de C n'est intervenu entre le temps t et le temps $t + 1$.

Règles:

- Le devoir doit être réalisé par deux (pas nécessairement du même groupe de TD).
- Il comprendra:
  1. Un texte clair et concis expliquant vos implémentations tout en justifiant vos choix.
  2. Une impression de vos sources (format `a2ps -2`, pas de recto/verso).
  3. Le nom d'un répertoire où elles sont consultables.

  Remarque: il n'est pas nécessaire de rendre le projet sous la forme d'un mémoire. La note sera essentiellement basée sur le contenu rédactionnel et la qualité de la programmation.
- Le devoir doit être remis le vendredi 28 Novembre 1998 à l'occasion du cours.