Architecture, Système, Réseau 2017-2018
{alice.pellet___mary,fabrice.mouhartem}@ens-lyon.fr, florent.de-dinechin@insa-lyon.fr

ASR1, TD/TP : circuits séquentiels et automates

Oh non, encore un TP sous logisim. Mais parfois il faut réfléchir sur papier.

Exercice 1: Compteur : Construisez un compteur 4 bits en assemblant un additionneur 4 bits et un registre 8 bits.
S'il y a du rouge partout, avez-vous pensé à l'initialisation?
Comptez.

Exercice 2: Considérer un circuit séquentiel comme un automate :
Quels sont les états possibles de votre compteur?
Sur une feuille de papier, dessinez l'automate correspondant.
Donnez un encodage des états de cet automate tel que la construction mécanique d'un automate synchrone retombe sur votre compteur. (cet exercice tourne un peu en rond mais c'est exprès).

Exercice 3: Suite de Fibonacci :
Proposez un circuit séquentiel pour calculer les termes de la suite de Fibonacci : $u_0 = 0$, $u_1 = 1$ et $u_i = u_{i-1} + u_{i-2}$ pour $i \geq 2$. Le circuit sera remis à zéro à l'aide d'un bouton `reset` : quand `reset` = 1 sur un front montant de l'horloge, le circuit est réinitialise, et la sortie produite est $u_2 = 1$. On suppose ensuite que l'on remet `reset` à 0, et à chaque cycle d'horloge le circuit produit un nouveau terme de la suite. Vous utiliserez des registres et un additionneur 16 bits.

Exercice 4: Différents encodages des états pour un automate reconnaisseur :
Il s'agit ici de construire en Logisim un circuit séquentiel reconnaissant le motif `111` : la sortie doit être à 1 sur un cycle si, lors des trois cycles précédents, l'entrée était à 1 (en tout cas sur le front montant de l'horloge, en fin de cycle).
On utilisera **obligatoirement** la méthodologie suivante (plus ou moins vue en cours) :

1. Quelles sont les entrées et les sorties du circuit séquentiel à construire? Dessinez sa boîte noire.
2. Dessiner un automate fini séquentiel équivalent au circuit à construire.
3. Encodez les états sur le moins de bits possible. Déduisez-en la fonction de transition (donnez sa table de vérité). On impose que *l'état suivant est calculé à partir de l'état courant et de l'entrée, et la sortie est calculée à partir de l'état courant uniquement.*
4. Dessiner sur papier le circuit, puis l'implanter dans Logisim et tester.
5. Dans un autre composant (du même fichier logisim), construire un circuit qui utilise des flips-flops pour se rappeler de 3 valeurs successives de l'entrée.
6. Tester l'égalité fonctionnelle de ces deux circuits.
7. Si un registre coûte 7 portes [1], lequel des deux circuits coûte le moins cher?

Exercice 5: Un multiplieur séquentiel :
En C par exemple, on peut implanter de la façon suivante l'algorithme de multiplication de la petite école pour multiplier deux entiers naturels `a_init` et `b_init` sur 8 bits, et obtenir le résultat sur 16 bits.

```
uint16_t mul(uint8_t a_init, uint8_t b_init) {
  uint16_t ax = a_init;
  uint8_t   b = b_init;
  uint16_t  c = 0;

  while(b != 0) {
    if(b & 0x01) c += ax; // b & 0x01 est le bit de poids faible de b
    ax <<= 1;             // décalage d'un bit à gauche
    b  >>= 1;             // décalage d'un bit à droite
  }
  return c;
}
```

Le but est d'implanter cet algorithme sous la forme d'un circuit séquentiel dans Logisim. Votre circuit prendra en entrée les entiers naturels `a_init` et `b_init` sur 8 bits, un signal `run` et un signal d'horloge. Il produira en sortie la valeur « courante » de `c`, ainsi qu'un signal `finished`, initialement à 0, et qui passera à 1 dès que le calcul est terminé.

1. Allez voir la page wikipedia du flip-flop