Nom et prénom :
Numéro d'étudiant :

Contrôle continu de TP - LIF6 Archi - Ve. 24/4/2015, durée 1h30

Seules les notes de TP/TD sont autorisées ; la communication (orale ou numérique) entre étudiant-e-s est interdite ; il en va de même pour la consultation d'internet : vos navigateurs doivent rester fermés pendant le TP [1]. Respectez les consignes données par votre chargé-e de TP en plus de celles données dans ce sujet. Le TP est à faire prioritairement seul, mais votre chargé de TP pourra autoriser les binômes si cela est nécessaire par manque de machines fonctionnelles. Cet énoncé ainsi que vos deux fichiers sont à rendre à votre chargé-e de TP.

1. Circuits dans LOGISIM

Récupérez le fichier `tpnote.circ` sur la page web du cours, ainsi que le fichier `msg.mem`. Le circuit LOGISIM donné dans ce fichier lit successivement les cases de la mémoire de façon séquentielle, et affiche les caractères qu'elles contiennent sur un écran (composant `TTY`) en interprétant leurs contenus comme du code ASCII. Le but est de modifier le circuit pour déchiffrer le message contenu dans `msg.mem`.

Question 1:

Ouvrir `tpnote.circ` dans LOGISIM, et charger `msg.mem` dans la mémoire RAM du circuit. Remettez le compteur d'adresse à zéro, et faites tourner la simulation.

(a) Quels caractères s'affichent à l'écran ?

(a) _____

(b) Quelle est la taille des cases de la mémoire ?

(b) _____

(c) Quelle est la taille des adresses ?

(c) _____

(d) Quelle est la taille de l'entrée du composant « écran » ?

(d) _____

(e) À quoi sert le splitter ?

_____

Question 2:

Ajoutez des composants de façon à stopper l'affichage sur l'écran dès que la case mémoire lue contient 0. Vous pourrez consulter avec avantage l'aide du composant écran (`TTY`). Vous ajouterez aussi un bouton permettant d'effacer le contenu de l'écran.

Question 3:

Pour décoder le message, si on lit x en mémoire, on affichera le code ASCII $x[6 \cdots 0]$ (7 bits de poids faibles de x) uniquement si le nombre de bits à 1 parmi les 9 bits de poids forts $x[15 \cdots 7]$ est supérieur strictement à 3. On pourra utiliser le composant `Bit Adder` et un comparateur (on fera attention à bien utiliser la comparaison en base 2 et pas en complément à 2).

Question 4:

Quel est le message obtenu une fois décodé ?

4. _____

2. Programmation en assembleur avec PennSim

Récupérez le fichier `tpnote.asm` sur la page web du cours. N'oubliez pas de télécharger le code du système d'exploitation du LC3, `lc3os.asm`, de l'assembler et de le charger dans la RAM de PennSim avant de tenter d'exécuter vos programmes !

1. Après le TP, vous pourrez consulter la section « méthodes anciennes » de la page http://fr.wikipedia.org/wiki/Methodes_d'execution pour en savoir plus sur les risques encourus. Mieux vaut poser des questions à votre chargé-e de TP…

Il s'agit de compléter la routine `dechiffre` pour qu'elle permette de déchiffrer le message qui se trouve rangé à partir de l'adresse `msg` sous la forme d'une chaîne de caractères se terminant par 0. La routine prend comme paramètres l'adresse du début du message dans `R0`, et la clé du chiffrement `k` dans `R1`. Pour décoder, la routine remplacera chacun des caractères `c` du message par `k^c` (ou-exclusif bit-à-bit entre `k` et `c`), sauf le caractère 0 final.

Question 1:

Assemblez et exécutez une première fois le programme : quel est le message affiché ?

1. _____

Question 2:

Si a et b sont deux variables booléennes, on rappelle que $a \oplus b$ désigne le ou-exclusif entre a et b. En utilisant les lois de Morgan, vérifiez que

$$a \oplus b = \overline{\overline{(\overline{a} \cdot b)} \cdot \overline{(a \cdot \overline{b})}}$$

Question 3:

On suppose que `R1` contient la clé et `R3` un caractère du message. Donnez un morceau de code en assembleur pour remplacer `R3` par `R1^R3`, en utilisant `R4` et `R5` comme variables intermédiaires du calcul.

Question 4:

En utilisant `R2` comme un pointeur pour parcourir le message, donnez un pseudo-code pour la routine `dechiffre`. Vous référencerez simplement le code de la question précédente par (*).

Question 5:

Complétez la routine `dechiffre` dans le fichier `tpnote.asm`, d'après le pseudo-code de la question précédente. Testez votre programme, en l'assemblant et en l'exécutant : quelle est la chaîne de caractères affichée ?

5. _____

Question 6:

Comment faire pour coder un message avec la clé fournie ? Complétez le programme (`tpnote.asm`) pour tester votre proposition.