Semaine 11 : architecture des ordinateurs

1 [N2] Un programme à étudier

Considérez le programme suivant :

```
 0: charge        r4, 0
 1: cont_ppe  r0, r1, 6
 2: charge        r5, r0
 3: charge        r0, r1
 4: charge        r1, r5
 5: charge        r4, 1
 6: cont_ppe  r1, r2, 11
 7: charge        r5, r1
 8: charge        r1, r2
 9: charge        r2, r5
10: charge        r4, 1
11: cont_ppe  r2, r3, 16
12: charge        r5, r2
13: charge        r2, r3
14: charge        r3, r5
15: charge        r4, 1
16: cont_egal r4, 1, 0
```

L'instruction « `cont_ppe a, b, L` » signifie « continue si `a` est plus petit ou égal à `b` ». Elle compare les valeurs `a` et `b` et, si `a`≤`b`, l'exécution continue à la ligne `L`. Les opérandes `a` et `b` peuvent être soit des constantes, soit indiquer les registres à lire pour obtenir les valeurs à comparer.
L'instruction « `cont_egal a, b, L` », qui signifie « continue si `a` égal `b` », agit de manière similaire, mais si les deux valeurs sont *identiques*.

a) Supposez que, avant le début de l'exécution, les registres contiennent les valeurs suivantes :

r0	r1	r2	r3
123	473	17	365

Simulez l'exécution du programme jusqu'à sa fin. Quelle est la valeur de ces quatre registres à la fin de l'exécution ?

b) Quelle est la fonction de ce programme ? Justifiez votre réponse.

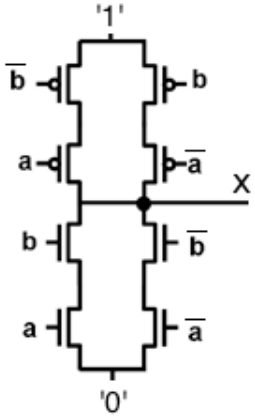
c) Que se passerait-il si on remplaçait toutes les instructions `cont_ppe` avec des instructions `cont_pp` qui réalisent la comparaison stricte `a`<`b` au lieu de `a`≤`b` ?

2 [N1] Somme de valeurs absolues

Ecrivez un programme assembleur pour calculer la somme des valeurs absolues de deux nombres. Utilisez les instructions vues au cours (p.ex. `somme r1, r2, r3` ou `soustrait r1, r2, r3`) et dans l'exercice précédent. Au départ, les registres `r0` et `r1` contiennent les deux nombres à traiter et la somme de leurs valeurs absolues doit se trouver dans le registre `r2` à la fin du programme.

3 [N1] Circuit

Considérez le circuit suivant où la notation $\overline{x}$ représente la négation logique du bit x (c.-à-d. $\overline{x} = 1 - x$). Quelle est la table de la vérité de ce circuit ? Quelle fonction logique réalise-t-il ?



Pour aller plus loin

4 Comparaison de valeurs horaires

Ecrivez un programme assembleur qui détermine laquelle de deux heures, H_1 et H_2, exprimées en heures et minutes entre 00h00 et 24h00, est la plus petite. Au départ, le registre `r0` (resp. `r2`) contient le nombre des heures de H_1 (resp. H_2) et le registre `r1` (resp. `r3`) contient les minutes de H_1 (resp. H_2). A la fin de l'exécution du programme, `r4` doit contenir la valeur 1 si H_1 est une heure strictement plus petite que H_2 et 0 sinon.

Exemple : si l'on doit comparer H_1 =8h10 et H_2 =21h45, on vous donne `r0` = 8, `r1` = 10, `r2` = 21 et `r3` = 45 et à la fin de l'exécution, `r4` devra contenir 1.

Rappel : l'instruction « `cont_pp a, b, L` » fait continuer l'exécution à la ligne `L` si `a` est un nombre strictement plus petit que `b`.