

Rattrapage

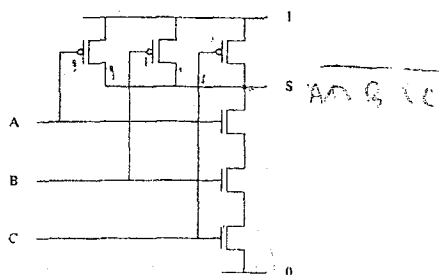
Circuits et architecture des ordinateurs

Master d'informatique -

Juin 2007, durée 2h.

L'examen se compose de cinq exercices indépendants. Les documents sont interdits à l'exception d'une feuille de mémento.

► **Exercice 1** On considère le circuit suivant



réalisé avec des transistors CMOS.

a) Remplir le tableau suivant qui donne la valeur de la sortie S en fonction des trois entrées A , B et C .

A	B	C	S
0	0	0	?
⋮	⋮	⋮	⋮

b) Donner la fonction réalisée par ce circuit.

► **Exercice 2** On désire utiliser l'*op-code* 1101 pour ajouter une nouvelle instruction au micro-processeur LC-3. Donner l'instruction la plus utile parmi les instructions possibles suivantes :

- a) **MOVE** R_i, R_j : R_j est copié dans le registre R_i .
- b) **OR** R_i, R_j, R_k : le *ou* logique bit-à-bit de R_j et R_k est mis dans R_i .
- c) **SHFL** $R_i, R_j, \#2$: le décalé de deux positions vers la gauche de R_j est mis dans R_i .
- d) **MUL** R_i, R_j, R_k : le produit en complément à deux de R_j et R_k est mis dans R_i .

Justifier votre réponse.

► **Exercice 3** Donner les accès à la mémoire nécessaires pour l'exécution complète de chacune des instructions suivantes du micro-processeur LC-3.

Instruction	Accès mémoire
LD R0, #12	Oui
ADD R0, R1, R3	Non
ADD R0, R0, #-1	Non
JMP loop	Non
LR R0, R5, #1	Oui
STI R0, #123	Oui

► **Exercice 4** Le fragment de programme suivant est sensé décaler le contenu du registre R3 de 4 positions vers la gauche mais il contient une erreur. Corriger le programme pour qu'il donne le bon résultat.

```

        .ORIG 0x3000
        AND R2, R2, #0
        ADD R2, R2, #4
loop:    BRz done
        ADD R2, R2, #-1
        ADD R3, R3, R3
        BR loop
done:    HALT
        .end

```

► **Exercice 5** Le but de cet exercice est d'écrire des programmes en langage assembleur du micro-processeur LC-3 étudié en cours.

- Écrire une routine qui compare deux valeurs entières. Cette routine prend deux valeurs x et y dans les registres R0 et R1 et retourne -1, 0 ou 1 dans R0 suivant que $x < y$, $x = y$ ou $x > y$.
- Modifier si nécessaire le programme pour qu'il fonctionne encore avec des valeurs signées.
- Écrire une routine calcule le maximum et le minimum de deux valeurs entières. Cette routine prend deux valeurs x et y dans les registres R0 et R1. Elle retourne la plus petite valeur dans R0 et la plus grande dans R1. Cette routine peut utiliser d'autres registres.
- Modifier si nécessaire le programme pour qu'il fonctionne encore avec des valeurs signées.
- Modifier le programme pour qu'il ne modifie aucun autre registre que R0, R1 et R7.