

Rattrapage

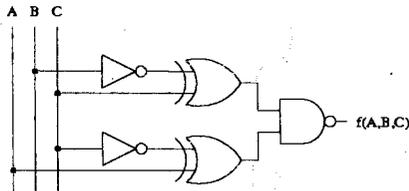
Circuits et architecture des ordinateurs

— Master d'informatique —

Juin 2010, durée 2h.

L'examen se compose de cinq exercices. L'exercice 2 utilise la fonction étudiée dans l'exercice 1. Les autres exercices sont indépendants. Les documents sont interdits à l'exception d'une feuille de mémento.

► Exercice 1



- Donner la table de vérité de la fonction f réalisée par le circuit ci-dessus.
- Donner un circuit équivalent en utilisant uniquement des portes NOT, AND et OR.
- Donner un circuit équivalent en utilisant uniquement des portes NAND.

► Exercice 2 Donner un programme en langage assembleur du processeur LC-3 qui calcule la fonction f de l'exercice précédent. Le programme doit se présenter comme une routine qui prend les valeurs A , B et C dans les registres $R0$, $R1$ et $R2$. Le résultat est retourné dans le registre $R0$. La valeurs booléennes 0 et 1 doivent être codées par les entiers 0 et 1.

► Exercice 3 On considère le programme suivant écrit en assembleur LC-3.

```

.ORIG x3000
mystere: LD R0,n
         LD R6,sp
         JSR myst
         TRAP x25      ; HALT
n:       .FILL 5
sp:      .FILL 0x4000

myst:   AND R0,R0,R0
    
```

1

```

BRz fini
ADD R6,R6,-1
STR R7,R6,0
ADD R0,R0,-1
JSR myst
LDR R7,R6,0
ADD R6,R6,1
ADD R0,R0,R0
fini:   ADD R0,R0,1
        RET
        .END
    
```

- Quel est le contenu en binaire du registre $R0$ lorsque le programme atteint l'instruction TRAP ?
- Quel serait le contenu du registre $R0$ lorsque le programme atteint l'instruction TRAP si la valeur chargée dans le registre $R0$ à la première instruction était respectivement 0, 1, 2, 3 et 20.
- Que calcule la routine myst quand la valeur contenue au départ dans $R0$ est positive ?

► Exercice 4 On considère la fonction donnée par le tableau ci-dessous. Cette fonction prend quatre entrées binaires $E3, E2, E1, E0$ et calcule quatre sorties binaires $S3, S2, S1, S0$.

| $E3$ | $E2$ | $E1$ | $E0$ | $S3$ | $S2$ | $S1$ | $S0$ |
|------|------|------|------|------|------|------|------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

- Donner un circuit qui calcule $S3, S2, S1, S0$ en fonction de $E3, E2, E1, E0$.
 - Donner un circuit qui calcule la fonction inverse en donnant $E3, E2, E1, E0$ en fonction de $S3, S2, S1, S0$.
- Exercice 5 Soit une machine dotée d'un bus d'adresses de 10 bits et d'un bus de données de 8 bits. Cette machine dispose également d'une mémoire cache

2

à accès direct à 8 entrées. Chaque ligne du cache contient un seul mot mémoire. On rappelle que pour un cache à accès direct, une donnée située à une adresse est toujours placée dans la même ligne de cache. On stockera une partie de l'adresse ainsi que la donnée associée à cette adresse.

- a) Quelle taille mémoire le processeur peut-il adresser ?
- b) Quelle est la taille du cache ?
- c) Décrire les informations stockées dans chaque ligne du cache.
- d) On accède successivement aux données $D1, D2, \dots, D12$ dont les adresses, exprimées en binaire, sont :

| Adresse | Donnée | Adresse | Donnée |
|------------|--------|------------|--------|
| 000000000 | D1 | 000000000 | D7 |
| 0000010100 | D2 | 0000010100 | D8 |
| 0000110110 | D3 | 0000110110 | D9 |
| 000110000 | D4 | 000110000 | D10 |
| 0000001100 | D5 | 0000001100 | D11 |
| 000100100 | D6 | 000100100 | D12 |

Déterminer l'état du cache (adresses et données stockées dans le cache) lorsque l'on aura accédé à toutes ces adresses dans cet ordre.