

# Exercices d'application

## Exercice 1 : QCM d'application

1- Quelle affirmation est exacte :

- ☐ La mémoire RAM ne fonctionne qu'en mode lecture
- ☐ La mémoire RAM permet de stocker des données et des programmes.
- ☐ Une mémoire ROM ne peut plus être utilisée.
- ☐ La mémoire RAM permet de stocker définitivement des données

2- Une conséquence de la loi de Moore est que mon ordinateur :

- ☐ télécharge plus vite les fichiers sur internet
- ☐ Comprime plus vite les fichiers
- ☐ Lit plus vite mes films

3- La mémoire qui permet les opérations de lecture et d'écriture simultanées est :

- ☐ ROM
- ☐ RAM
- ☐ EPROM
- ☐ EEPROM

4- Qui n'est pas considéré comme un périphérique de l'ordinateur ?

- ☐ Disque
- ☐ Clavier
- ☐ Moniteur
- ☐ CPU

5- Le composant informatique le plus rapide est :

- ☐ RAM
- ☐ cache
- ☐ registre
- ☐ disque dur

6- Qu'utilise l'ALU pour stocker les résultats intermédiaires ?

- ☐ L'accumulateur
- ☐ Les registres
- ☐ Le tas
- ☐ La pile

7- Le processeur est composé des pièces suivantes :

- ☐ CU (unité de contrôle) et mémoire principale
- ☐ CU et ALU
- ☐ Mémoire principale et stockage
- ☐ Système d'exploitation et application

8- Le composant du CPU chargé de comparer le contenu de deux données est :

- ☐ ALU
- ☐ CU
- ☐ Mémoire
- ☐ Registre

9- Lequel de ces périphériques n'est pas un périphérique d'entrée ?

- ☐ le clavier standard
- ☐ la souris standard
- ☐ l'écran standard
- ☐ le scanner

10- Quel est le rôle du processeur dans un ordinateur ?

- ☐ Stocker de manière temporaire les données
- ☐ Exécuter les instructions et les calculs qui lui sont donnés
- ☐ Stocker de manière définitive les données
- ☐ Relier les périphériques à l'ordinateur



## Exercice 2 : Principe de fonctionnement d'un UAL

L'Unité Arithmétique et Logique (ou UAL) est la partie du processeur qui réalise les calculs et les opérations logiques élémentaires. Selon le processeur, cette unité peut réaliser des opérations plus ou moins complexes, sur des nombres plus ou moins longs. Nous allons considérer un UAL simpliste pouvant travailler sur des nombres de 4 bits. Dans la suite, nous noterons  $\bar{a}$  l'opposé de  $a$  bit à bit. L'architecture globale est la suivante :

x INV : renvoie  $\bar{a}$  si  $op_3 = 1$  et  $a$  sinon.

x ADD : renvoie la somme des deux nombres  $e_1 + e_2$ .

x NAND : renvoie le résultat de l'opération logique NON( $e_1$  ET  $e_2$ )

x MUX : renvoie  $e_3$  si  $op_1=0$  et  $e_4$  sinon.

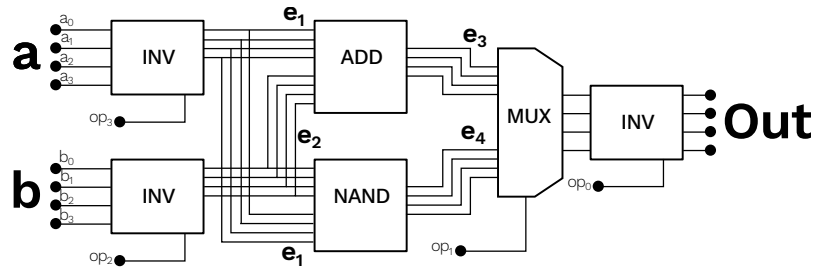


Figure 1: Schéma interne de l'UAL

Les bits  $op_3$  à  $op_0$  correspondent au code opération (opcode). Ce code sert à déterminer le type d'opération effectuée par l'unité. Par exemple, avec le code 0000, l'unité réalise  $a + b$ , alors qu'avec 0010, elle réalise  $a \text{ nand } b$ .

La plupart des UAL ont des bits supplémentaires en sortie permettant de savoir s'il reste une retenue à la fin de la somme (overflow), si le résultat est négatif (les entiers sont représentés en complément à 2), égal à zéro. . .

### Exemples d'utilisation

Le tableau ci-contre présente quelques exemples d'utilisation. On remarque qu'il est possible de faire une soustraction. En effet, par définition du complément à 2, on a :

$$-a = \text{non}(a) + 1, \text{ donc } \text{non}(a) = -a - 1$$

Pour la 4<sup>e</sup> ligne, l'opération effectuée correspond à  $\text{non}(\text{non}(a) + b)$ . On a donc :

$$\begin{aligned} \text{non}(\text{non}(a) + b) &= \text{non}(-a - 1 + b) \\ \text{non}(\text{non}(a) + b) &= -(-a + b - 1) - 1 = a - b \end{aligned}$$

| Cas | opcode | a    | b    | Out  | Commentaire         |
|-----|--------|------|------|------|---------------------|
| 1   | 0000   | 0011 | 0101 | 1000 | $a+b$               |
| 2   | 0000   | 0001 | 1111 | 0000 | overflow            |
| 3   | 0101   | 0011 | 0101 | 0010 | $b-a$               |
| 4   | 1001   | 0011 | 0101 | 1110 | $a-b$               |
| 5   | 0101   | 0011 | 0000 | 1101 | $-a$                |
| 6   | 0001   | 1100 | 0000 | 0011 | $\text{non } a$     |
| 7   | 0010   | 0011 | 0101 | 1110 | $a \text{ nand } b$ |
| 8   | 0011   | 0011 | 0101 | 0001 | $a \text{ et } b$   |
| 9   | 1110   | 0011 | 0101 | 0111 | $a \text{ ou } b$   |

Le fichier de simulation de cette UAL (logiciel Logisim) est présent sur Moodle dans les fichiers associés à l'exercice

1. A partir d'une simulation sur logisim, **compléter** le tableau pages 3/4 et 4/4 afin de décrire les résultats trouvés sur la sortie OUT de la tables des exemples d'utilisation ci-dessus.
2. **Déterminer** le résultat des opérations suivantes :

| opcode | a    | b    | Out |
|--------|------|------|-----|
| 0000   | 1011 | 0110 |     |
| 0101   | 1011 | 0110 |     |
| 1001   | 1011 | 0110 |     |
| 0010   | 1011 | 0110 |     |
| 0011   | 1011 | 0110 |     |
| 1110   | 1011 | 0110 |     |
| 0101   | 1011 | 0000 |     |



Détail de l'exemple d'utilisation

| Cas | Etat de l'UAL | Commentaires |
|-----|---------------|--------------|
| 1   |               |              |
| 2   |               |              |
| 3   |               |              |
| 4   |               |              |
| 5   |               |              |

|   |  |  |
|---|--|--|
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |



Détail de la question 2

| Cas | Etat de l'UAL | Commentaires |
|-----|---------------|--------------|
| 1   |               |              |
| 2   |               |              |
| 3   |               |              |
| 4   |               |              |
| 5   |               |              |

|   |  |  |
|---|--|--|
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |

