

# Algorithmes et Pensée Computationnelle

## Architecture des ordinateurs

Le but de cette séance est de comprendre le fonctionnement d'un ordinateur. La série d'exercices sera axée autour de conversions en base binaire, décimale ou hexadécimale, de calcul de base en suivant le modèle Von Neumann.

## 1 Conversions

### Question 1: (🕒 10 minutes) Conversion

1. Convertir le nombre  $FFFFFF_{(16)}$  en base 10.
2. Convertir le nombre  $4321_{(5)}$  en base 10.
3. Convertir le nombre  $ABC_{(16)}$  en base 2.
4. Convertir le nombre  $254_{(10)}$  en base 15.
5. Convertir le nombre  $11101_{(2)}$  en base 10.

#### 💡 Conseil

N'oubliez pas qu'en Hexadécimal, A vaut 10, B vaut 11, C vaut 12, D vaut 13, E vaut 14 et F vaut 15.

#### >\_ Solution

1.  $FFFFFF_{(16)} = 16777215_{(10)}$
2.  $4321_{(5)} = 586_{(10)}$
3.  $ABC_{(16)} = 101010111100_{(2)}$
4.  $254_{(10)} = 11E_{(15)}$
5.  $11101_{(2)} = 29_{(10)}$

## 2 Conversion et arithmétique

### Question 2: (🕒 5 minutes) Conversion, addition et soustraction :

Effectuer les opérations suivantes :

1.  $10110101_{(2)} + 00101010_{(2)} = \dots_{(10)}$
2.  $70_{(10)} - 10101010_{(2)} = \dots_{(10)}$

#### 💡 Conseil

Convertissez dans une base commune avant d'effectuer les opérations.

#### >\_ Solution

1.  $10110101_{(2)} + 00101010_{(2)} = 202_{(10)}$
3.  $70_{(10)} - 10101010_{(2)} = 240_{(10)}$

### 3 Modèle de Von Neuman

Dans cette section, nous allons simuler une opération d'addition dans le **modèle de Van Neumann**, il va vous être demandé à chaque étape (FDES) de donner la valeur des registres.

#### État d'origine :

A l'origine, notre **Process Counter (PC)** vaut **00100001**.

Dans la mémoire, les instructions sont les suivantes :

Adresse	Valeur
00100001	00110100
00101100	10100110
01110001	11111101

Les registres sont les suivantes :

Registre	Valeur
00	01111111
01	00100000
10	00101101
11	00001100

Les opérations disponibles pour l'unité de contrôle sont les suivantes :

Numéro	Valeur
00	ADD
01	XOR
10	MOV
11	SUB

#### Question 3: (🕒 5 minutes) Fetch

À la fin de l'opération **FETCH**, quelles sont les valeurs du **Process Counter** et de l'**Instruction Register** ?

#### 💡 Conseil

Pour rappel, l'unité de contrôle (Control Unit) commande et contrôle le fonctionnement du système. Elle est chargée du séquençage des opérations. Après chaque opération **FETCH**, la valeur du Program Counter est incrémentée (valeur initiale + 1).

#### >\_ Solution

$PC = 00100001_{(2)} + 1 = 00100010_{(2)}$   
 $IR = 00110100_{(2)}$

#### Question 4: (🕒 5 minutes) Decode

1. Quelle est la valeur de l'opération à exécuter ?
2. Quelle est l'adresse du registre dans lequel le résultat doit être enregistré ?
3. Quelle est la valeur du premier nombre de l'opération ?
4. Quelle est la valeur du deuxième nombre de l'opération ?

#### Conseil

Pensez à décomposer la valeur de l'Instruction Register pour obtenir toutes les informations demandées.

Utilisez la même convention que celle présentée dans les diapositives du cours (Architecture des ordinateurs (Semaine 2) - Diapositive 15)

Les données issues de la décomposition de l'**Instruction Register** ne sont pas des valeurs brutes, mais des références. Trouvez les tables concordantes pour y récupérer les valeurs.

#### >\_ Solution

00 : **ADD** (valeur de l'opération à exécuter)

11 : Adresse du registre dans lequel le résultat doit être enregistré

01 : 00100000<sub>(2)</sub> (premier nombre)

00 : 01111111<sub>(2)</sub> (deuxième nombre)

#### Question 5: (🕒 5 minutes) **Execute**

Quel est résultat de l'opération ?

#### Conseil

Toutes les informations permettant d'effectuer l'opération se trouvent dans les données de l'**Instruction Register**.

#### >\_ Solution

$$00100000_{(2)} + 01111111_{(2)} = 10011111_{(2)}$$

## 4 Systèmes d'Exploitation

#### Question 6: (🕒 5 minutes) Sous Linux et MacOS, laquelle de ces commandes modifie le **filesystem** ?

1. `ls -la`
2. `sudo rm -rf ~/nano`
3. `sudo kill -9 3531`
4. `more nano.txt`
5. Aucune réponse n'est correcte.

#### >\_ Solution

La commande `sudo rm -rf ~/nano` permet de supprimer le répertoire **nano** situé dans le dossier `/Users/<Utilisateur.courant>` en mode super-utilisateur (utilisateur ayant des droits étendus sur le système).

#### Conseil

**Attention !** Certaines commandes listées ci-dessus peuvent avoir des conséquences irréversibles. Pour avoir une description détaillée d'une commande, vous pouvez ajouter **man** devant chaque commande sous Linux/MacOS ou ajouter **-h**, **--help** ou **/?** après chaque commande sous Windows.