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 de conversions en base binaire, décimale ou hexadécimal, de calcul de base en suivant le modèle Von Neumann.

1 Conversions

Question 1: (*5 minutes*) **Conversion** $Base_{10}$ **-** $Base_2$

1. Convertir le nombre $10_{(10)}$ en base 2.
2. Convertir le nombre $45_{(10)}$ en base 2.
3. Convertir le nombre $173_{(10)}$ en base 2.

Conseil

TODO : Conseils pour passer de la base décimale à la base binaire
Useful link : http ://www.circuits-logiques.polymtl.ca/help/Chapitre05.pdf

Question 2: (*5 minutes*) **Conversion** $Base_{10}$ **-** $Base_3$**,** $Base_8$**,** $Base_{16}$

1. Convertir le nombre $40_{(10)}$ en base 8.
2. Convertir le nombre $52_{(10)}$ en base 3.
3. Convertir le nombre $254_{(10)}$ en base 16.

Conseil

TODO : Conseils pour passer de la base décimale aux bases 3, 8 et 16.

Solution

Présenter les étapes détaillées permettant d'aboutir à la solution.

Question 3: (*10 minutes*) **Conversion** $Base_3$ **-** $Base_{16}$ **en** $Base_8$

1. Convertir le nombre $10110_{(2)}$ en base 10.
2. Convertir le nombre $4321_{(5)}$ en base 10.
3. Convertir le nombre $ABC_{(16)}$ en base 10.

Conseil

TODO : Conseils pour passer des bases 3 et 16 à la base 10.

Solution

Présenter les étapes détaillées permettant d'aboutir à la solution.

2 Arithmétique binaire

Question 4: (*15 minutes*) **Addition de nombres binaires**

1. Additionner $\mathbf{01010101}_{(2)}$ et $\mathbf{10101010}_{(2)}$
2. Additionner $\mathbf{01011111}_{(2)}$ et $\mathbf{10000001}_{(2)}$
3. Additionner $\mathbf{01110100}_{(2)}$ et $\mathbf{00011010}_{(2)}$

Conseil

Table d'addition binaire :

a	b	s=a+b	r
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TODO : Rajouter un exemple

Question 5: (*15 minutes*) **Soustraction de nombres binaires**
Effectuer les opérations suivantes :

Conseil

Table de soustraction binaire :

a	b	s=a-b	r
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Exemple :

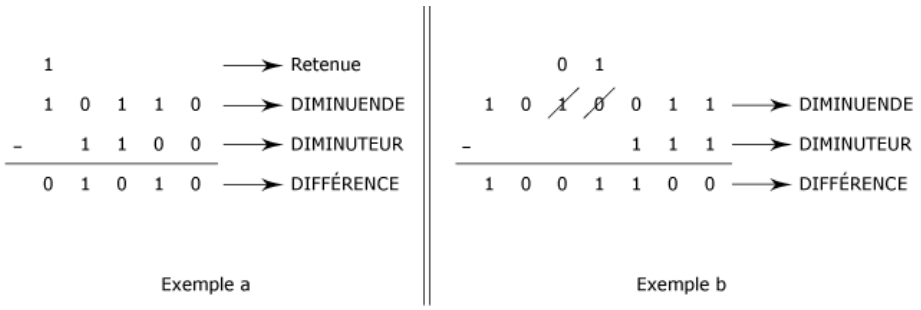

FIGURE 1 – Exemple de soustraction de nombres binaires

3 Conversion et arithmétique

Question 6: (*20 minutes*) **Conversion et addition :**
Effectuer les opérations suivantes :

1. $111101_{(2)} + 110_{(2)} = \ldots_{(10)}$
2. $111111_{(2)} + 000001_{(2)} = \ldots_{(10)}$
3. $127_{(10)} + \mathrm{ABC}_{(16)} = \ldots_{(8)}$

Question 7: (🕒 *20 minutes*) **Conversion et soustraction :**
Effectuer les opérations suivantes :

1. $101010_{(2)}$ - $010101_{(2)}$ = ...$_{(10)}$
2. $64_{(10)}$ - $001000_{(2)}$ = ...$_{(10)}$
3. $FFF_{(10)}$ - $127_{(10)}$ = ...$_{(2)}$

4 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 **00100000**.

Dans la mémoire, les instructions sont les suivantes :

Adresse	Valeur
00011111	00100100
00100000	10110110
00100001	11101101

Les registres sont les suivantes :

Registre	Valeur
00	11100011
01	01101100
10	00100101
11	00000000

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

Numéro	Valeur
00	MOV
01	XOR
10	ADD
11	SUB

Question 8: (🕓 *5 minutes*) **Fetch**
À la fin de l'opération **FETCH**, quelles sont les valeurs du **Process Counter** et de l'**Instruction Register** ?

Question 9: (🕓 *5 minutes*) **Decode**

Question 10: (🕓 *5 minutes*) **Execute**