



**Module** : Architecture des Ordinateurs  
**Filière** : Licence Informatique

**Responsable:** Mokrani Hocine  
**Semestre** : 03

## Série 1 : Chapitre Rappel + chapitre Mémoires (solutions)

### Question de cours

1. Qu'est ce qui caractérise la machine de Von Neumann par rapport aux autres machines ?

**Réponse** : elle utilise une unique structure pour stocker à la fois le programme et les données.

2. Quels sont les éléments de base de la machine de Von Neumann ?

**Réponse** : Une unité centrale de traitement, une mémoire centrale, des interfaces E/S et des bus de communication.

3. Faites la correspondance entre les unités fonctionnelle d'un ordinateur et celle du système nerveux humain.

**Réponse** : (1) Processeur - (C) Cerveaux ; (2) Mémoire - (A) Mémoire naturelle ; (3) Unités Centrales - (C) Cerveaux ; (4) Périphériques - (B) Organe de sensation ; (4) Périphériques - (D) La bouche, les oreilles, les yeux.

4. Dans un ordinateur, quelle est la différence entre un registre et une mémoire ? (Comparez la nature, la taille, la vitesse et le moyen d'accès aux informations)

	Registre	Mémoire
<b>Nature</b>	SRAM	DRAM
<b>Taille</b>	Petite	Grande
<b>Vitesse</b>	Grande	Petite
<b>Accès</b>	Direct	Bus

5. Quel est le rôle de l'UAL ?

**Réponse** : Le rôle d'une UAL est d'exécuter les instructions de calcul arithmétique et logique.

6. Quel est le rôle du séquenceur dans une unité de commande ?

**Réponse** : Le rôle d'un séquenceur est d'interpréter, de déclencher et coordonner les différents composants intervenant dans l'exécution d'instruction.

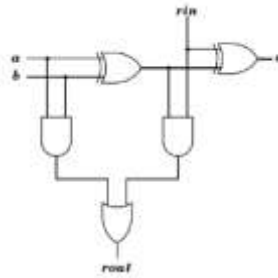
7. Quels sont les éléments qui constituent une unité de commande et de contrôle dans la CPU ?

**Réponse** : UCC se compose :

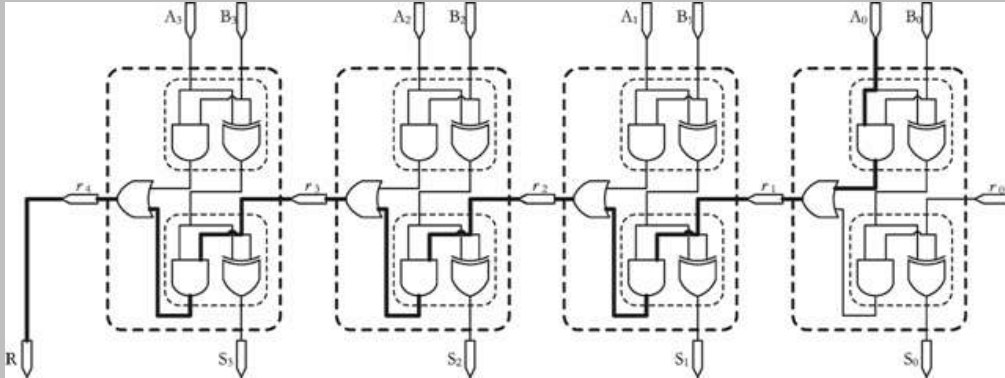
- d'un compteur ordinal (PC pour Program compter) : qui contient l'adresse de l'instruction à exécuter.
- d'un registre d'instruction (RI) : qui contient l'instruction à exécuter.
- d'un séquenceur : qui décode le code d'opération de l'instruction en cours d'exécution. Selon le code de l'opération (de l'instruction), le séquenceur active un ensemble de composant selon un ordre prédéfini. Par exemple, les composants intervenant dans l'accès à la mémoire ne sont pas les mêmes composants utiliser pour la réalisation de calcul.
- et d'une horloge : le rôle de l'horloge est de cadencer les différents composants. Elle permet en réalité de décomposer le temps en cycle. Ce qui permet au séquenceur (et à l'ensemble de composant) d'identifier le début et la fin d'une étape dans la séquence d'exécution d'une opération.

### Exercice 1

1. Réalisez un additionneur 4 bits à partir de l'additionneur 1 bit suivant :



**Réponse :**

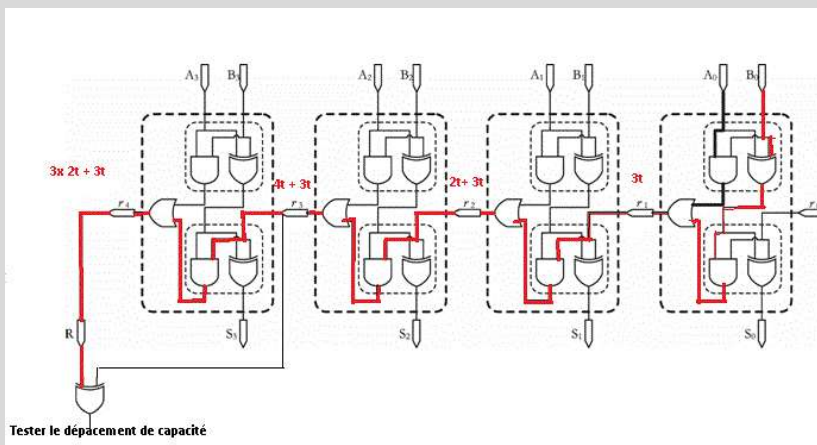


2. Dans le cas de l'additionneur 4 bits, étendre le dispositif pour tester la validité du résultat dans le cas d'entiers codés à complément à 2, i.e, tester s'il y a dépassement de capacité.

**Réponse :** Un débordement ne peut pas se produire lors de l'addition de nombres de signes opposés. Une autre façon à détecter le débordement: Si la retenue d'entrée à l'additionneur des bits de signe (C3) est différente de la retenue sortante (C4), alors il y a débordement. Ce résultat peut être testé en utilisant la porte Xor pour comparer les retenues (C3) et (C4).

3. En supposant que le temps de franchissement d'une porte est de  $t$ , au bout de combien de temps le résultat de l'opération est disponible pour un additionneur  $n$  bits ?

**Réponse :**

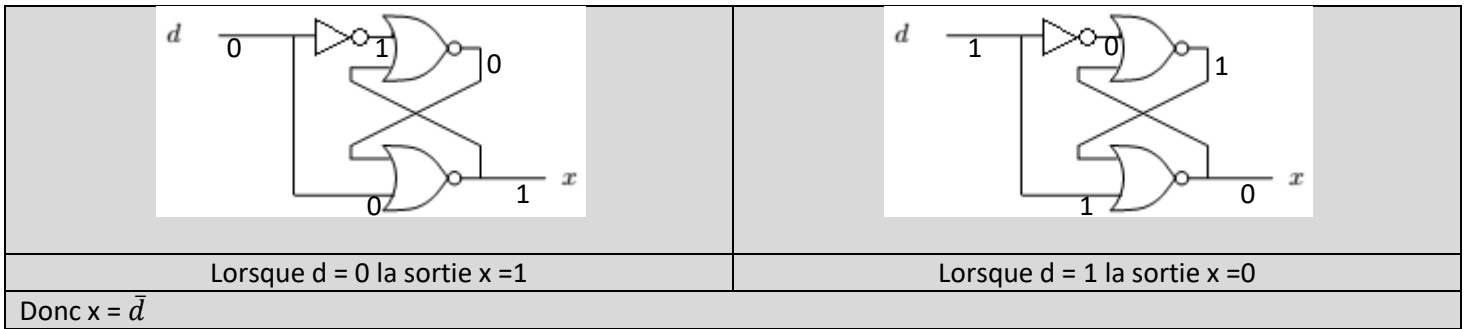


Il faut chercher le chemin le plus long pour calculer le temps qu'il faut pour additionner  $n$  bits. La figure montre ce chemin pour un additionneur 4 bits. Selon la figure, il faut  $10t$  pour additionner 4 bits. En généralisant la méthode de calcul, on trouve qu'il faut  $2*(n+1)*t$  pour additionner  $n$  bits.

## Exercice 2

a. Analyser le comportement du circuit.

a) Méthode schématique :



b) Méthode algébrique :

$$\begin{aligned}
 x_{t+1} &= \overline{d + (\bar{d} + x_t)} \\
 x_{t+1} &= \bar{d} \cdot (\bar{d} + x_t) \\
 x_{t+1} &= \bar{d} \cdot \bar{d} + \bar{d} \cdot x_t \\
 x_{t+1} &= \bar{d} (1 + x_t) \\
 x_{t+1} &= \bar{d}
 \end{aligned}$$

2. Que se passe-t-il si on remplace les portes non-ou par des portes non-et?

**Réponse :** Si on remplace les portes NOR par des portes NAND on trouvera le même résultat.

**Exercice 3**

<p>1. Exprimer les valeurs de sortie du cycle suivant <math>y_2', y_1', y_0'</math> en fonction des valeurs du cycle présent <math>y_2, y_1, y_0</math> et de l'entrée <math>x</math>.</p>	<p><b>Réponse :</b></p> $  \begin{aligned}  y_2' &= x \\  y_1' &= y_2 \\  y_0' &= \overline{(y_1 \cdot \bar{x}) \cdot (y_1 \cdot \bar{x})}  \end{aligned}  $																																																																																																																							
<p>2. Donner la table d'états correspondant à ce circuit.</p>	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th><math>x</math></th> <th><math>y_2</math></th> <th><math>y_1</math></th> <th><math>y_0</math></th> <th><math>y_2'</math></th> <th><math>y_1'</math></th> <th><math>y_0'</math></th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	$x$	$y_2$	$y_1$	$y_0$	$y_2'$	$y_1'$	$y_0'$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	1	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0	0	0	1	0	0	1	0	0	1	1	0	0	1	0	1	0	1	0	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1
$x$	$y_2$	$y_1$	$y_0$	$y_2'$	$y_1'$	$y_0'$																																																																																																																		
0	0	0	0	0	0	0																																																																																																																		
0	0	0	1	0	0	0																																																																																																																		
0	0	1	0	0	0	1																																																																																																																		
0	0	1	1	0	0	1																																																																																																																		
0	1	0	0	0	1	0																																																																																																																		
0	1	0	1	0	1	0																																																																																																																		
0	1	1	0	0	1	1																																																																																																																		
0	1	1	1	0	1	1																																																																																																																		
1	0	0	0	1	0	0																																																																																																																		
1	0	0	1	1	0	0																																																																																																																		
1	0	1	0	1	0	1																																																																																																																		
1	0	1	1	1	0	1																																																																																																																		
1	1	0	0	1	1	0																																																																																																																		
1	1	0	1	1	1	0																																																																																																																		
1	1	1	0	1	1	1																																																																																																																		
1	1	1	1	1	1	1																																																																																																																		
<p>3. Expliquer en une phrase ce que fait le circuit.</p>	<p><b>Réponse :</b> Ce circuit est un registre à décalage.</p>																																																																																																																							
<p>4. Comment pourrait-on simplifier ce circuit ?</p>	<p><b>Réponse :</b> <math>y_0' = \overline{(y_1 \cdot \bar{x}) \cdot (y_1 \cdot \bar{x})} = y_1</math></p>																																																																																																																							

**Exercice 4**

1. Quelles sont les principales différences entre la DRAM et la SRAM ?

**Réponse :**

- La nature, la SRAM est constitué que de transistor (6 transistors pour 1 bit). Par contre, une DRAM est constitué d'un transistor et un condensateur pour 1 bit, ce qui rend les SRAM plus chers que les DRAM.
- La vitesse, la SRAM étant bien plus rapide que la DRAM, car la DRAM à besoin d'un temps de rafraichissement des données vu sa nature physique (il faut recharger le condensateur chaque période du temps).
- La capacité (les SRAM étant utilisée pour stocker de petites quantités de données, à l'inverse de la DRAM, qui est utilisée pour des volumes bien plus importants).

2. Où utilise-t-on de la DRAM ? De la SRAM ?

**Réponse :** La SRAM ce trouve dans l'unité centrale (Processeur), sous forme de registre, alors que la DRAM compose la mémoire centrale de l'ordinateur.

3. Classez les mémoires suivantes par taille et par rapidité : RAM, registres, disques durs, cache L1, cache L2, cd-rom.

**Réponse :**

Par rapidité décroissante : 1- Registres, 2- Cache L1, 3- Cache L2, 4- Ram, 5- Disques durs, 6- CD-Rom.

Par taille décroissante : 1- Disques Durs, 2-Ram, 3- CD-Rom, 4- Cache L2, 5- Cache L1, 6- Registres.

4. On considère une mémoire centrale de 2 Mo où chaque octet est adressable séparément ;

- a. Calculer l'adresse, en octal, du sixième élément d'un tableau dont l'adresse du premier élément est  $(76)_8$  et dont tous les éléments sont composés de 16 bits.

**Réponse :**

$\text{Adrs\_N\_Elem} = \text{Adrs\_1\_Elem} + (\text{N}-1 \times \text{Taille (Elément)})$

$$(76)_8 + (5 \times 2)_8 = (76)_8 + (12)_8 = (110)_8$$

- b. Calculer, en décimal, le nombre d'octets précédant l'adresse  $(76)_8$ ;

**Réponse :** vu que les adresses mémoire commencent par 0, donc le nombre d'octets précédents et de  $(76)_8 = (62)_{10}$

- c. Calculer la taille de cette mémoire en l'exprimant en mots de 16 bits puis en mots de 32 bits.

**Réponse :**

La taille de la mémoire ne change pas, ce n'est que l'expression qui change.

Taille mémoire =  $2 \times 2^{20}$  octets et 16 bits = 2 octets => Taille mémoire =  $2 \times 2^{20} / 2 = 2^{20}$  mots de 16 bits.

Taille mémoire =  $2 \times 2^{20}$  octets et 32 bits = 4 octets => Taille mémoire =  $2 \times 2^{20} / 4 = 2^{19}$  mots de 32 bits.

**Exercice 5**

Si le registre d'adresse d'une mémoire comporte 32 bits, calculer:

- a. Le nombre de mots adressables si 1 mot = 1 byte;

**Réponse :** Le nombre de mots adressables (si 1 mot = 1 byte) =  $2^{32} = 4\,294\,967\,296$  mots

- b. La plus haute adresse possible pour ces mots de 1 byte;

**Réponse :** La plus haute adresse possible =  $(4294967296 - 1)_{10} = (4\,294\,967\,295)_{10} = 0xFFFFFFFF$

- c. Le nombre de mots adressables si 1 mot = 32 bits;

**Réponse :** Ça ne changera pas, le nombre de Mots est indépendant de la taille de ce dernier, elle ne dépend que de la taille du registre d'adresse et la taille de la mémoire.

- d. La plus haute adresse possible pour ces mots de 32 bits.

**Réponse :** Ça ne changera pas, La plus haute adresse possible =  $(4\ 294\ 967\ 296 - 1)_{10} = (4\ 294\ 967\ 295)_{10} = 0xFFFFFFFF$

## Exercice 6

Soit une mémoire centrale de 1 Mmots de 32 bits réalisée avec des puces de stockage. Cette mémoire peut être organisée suivant plusieurs principes. Considérez les trois principes suivants :

1. Un bit par puce : un mot est constitué de  $32 \times 1$  bit provenant chacun d'une puce différente, donc 32 puces sont nécessaires pour réaliser un mot;
2. 16 bits par puce : un mot est constituée de  $2 \times 16$  bits; deux puces sont donc nécessaires pour former un mot de 32 bits;
3. 32 bits par puce : un mot est constitué de  $1 \times 32$  bits.

Calculer :

- a. Le nombre de bits nécessaires pour adresser toute la mémoire dans chacun des cas;
- b. Le nombre de puces nécessaires pour construire la mémoire dans chacun des cas ;
- c. Le nombre de pattes (de chaque puce) utilisées pour l'adressage et pour les données dans chacun des cas.

<b>Mémoire de 1 Mmots de 32 bits ==&gt; 1 cellule à 32 bits.</b>
a. Le nombre de bits nécessaires pour adresser toute la mémoire dans chacun des cas;
a.1. Pour adresser 1 Mmots, il faut $\log_2(1\ Mmots) = \log_2(2^{20}) = 20\ bits$
a.2. La même chose (la taille de la mémoire n'a pas changé, c'est la construction d'une cellule qui change).
a.3. La même chose (la taille de la mémoire n'a pas changé, c'est la construction d'une cellule qui change).
b. Le nombre de puces nécessaires pour construire la mémoire dans chacun des cas ;
b.1. -On a $2^{20}$ cellules mémoire. - Chaque cellule est réalisée par 32 puces. - Donc : Nombre_puce = $2^{20} \times 2^5 = 2^{25}$ puces.
b.2. -On a $2^{20}$ cellules mémoire. - Chaque cellule est réalisée par 2 puces de 16 bits chaque une. - Donc : Nombre_puce = $2^{20} \times 2 = 2^{21}$ puces.
b.3. -On a $2^{20}$ cellules mémoire. - Chaque cellule est réalisée par 1 puce de 32 bits. - Donc : Nombre_puce = $2^{20} \times 1 = 2^{20}$ puces.
c. Le nombre de pattes (de chaque puce) utilisées pour l'adressage et pour les données dans chacun des cas.
c.1. Une puce à 4 pattes : une patte pour l'activation, une patte pour le mode d'accès lecture\écriture, deux pattes pour les données, une pour l'entrée l'autre pour la sortie).
c.2. Une puce à 34 pattes : une patte pour l'activation, une patte pour le mode d'accès lecture\écriture, 32 pattes pour les données (16 pattes pour l'entrée et 16 pattes pour la sortie).
c.2. Une puce à 66 pattes : une patte pour l'activation, une patte pour le mode d'accès lecture\écriture, 64 pattes pour les données (32 pattes pour l'entrée et 32 pattes pour la sortie).

## Exercice 7

Nous avons à notre disposition 4 puces de mémoire de largeur 8 bits et hauteur  $2^{13} = 8\ 192$ . Proposer une méthode pour connecter les 4 puces dans le but d'obtenir une mémoire ayant une largeur de 16 bits et une hauteur de  $2^{14} = 16\ 384$ .

