

TD 1 Architecture des ordinateurs

Table des matières

1) Question	1
2) Question	1
3) Question	3
4) Question	3
5) Question	4
6) Question	4
7) Question	5
8) Question	5
9) Question	5
10) Question	5
11) Question	5
12) Question	5
13) Question	6
14) Question	6
15) Question	6
16) Question	6
17) Question	6
18) Question	7
19) Question	7
20) Question	7

1) [Question](#)

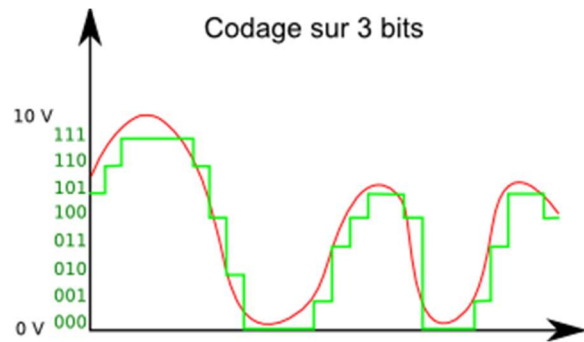
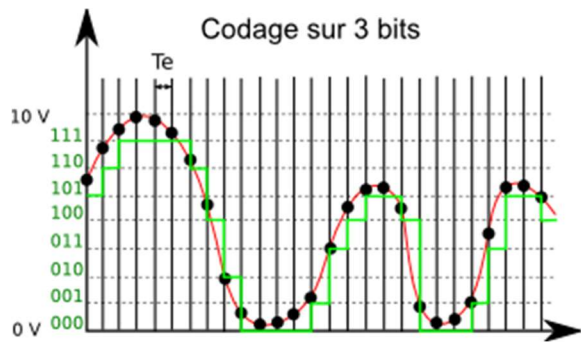
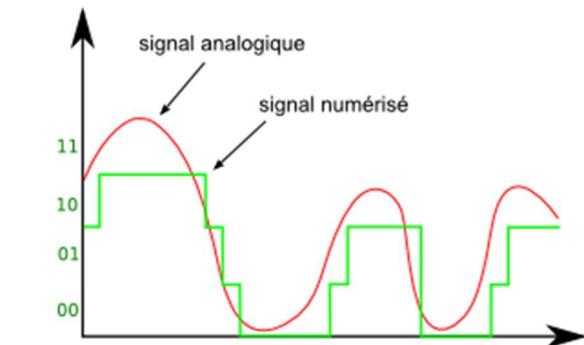
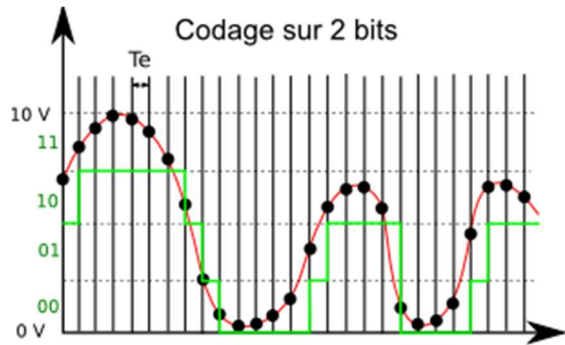
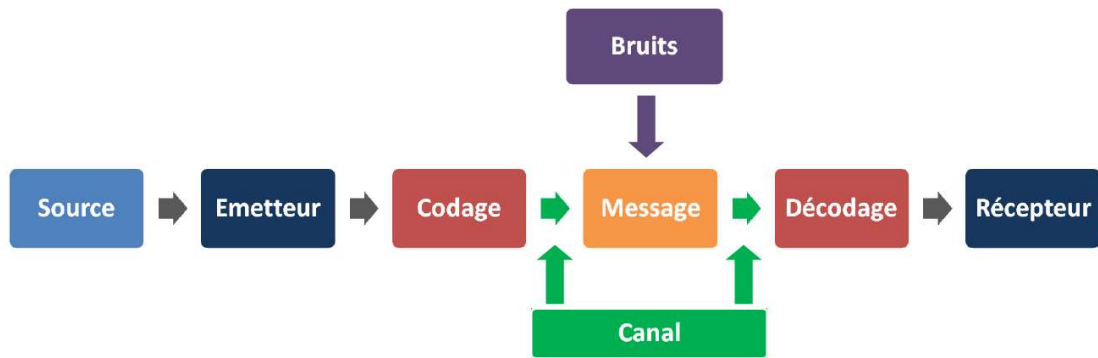
Pourquoi a-t-on un circuit d'horloge dans une machine informatique ?

A quoi sert une carte mère, donnez les deux caractéristiques principales

Qu'est ce que le BIOS d'une carte mère ?

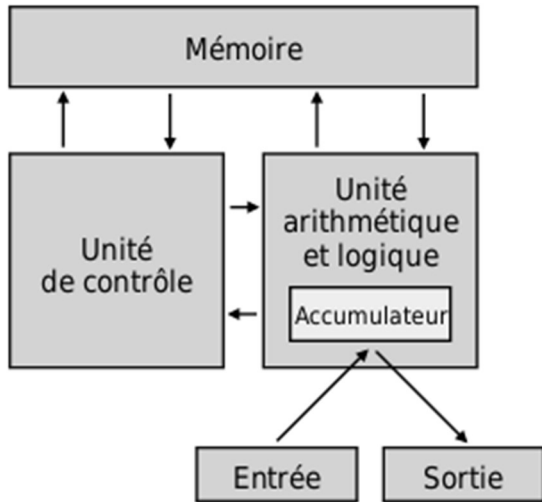
2) [Question](#)

Expliquez cette suite de schéma



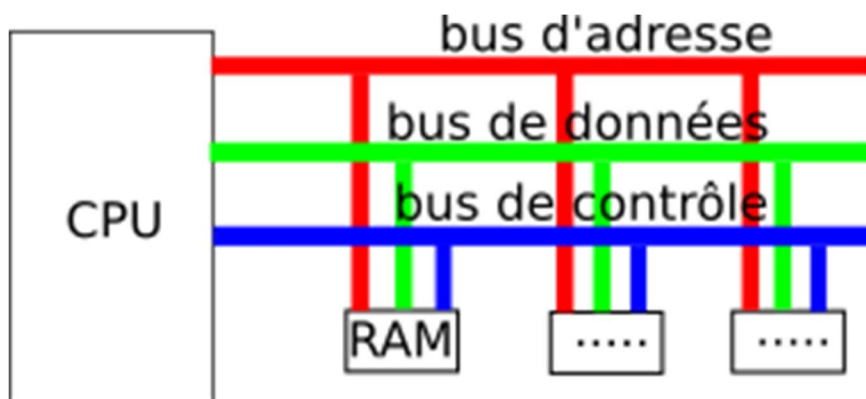
3) [Question](#)

Expliquez ce schéma, quelle est son origine ?



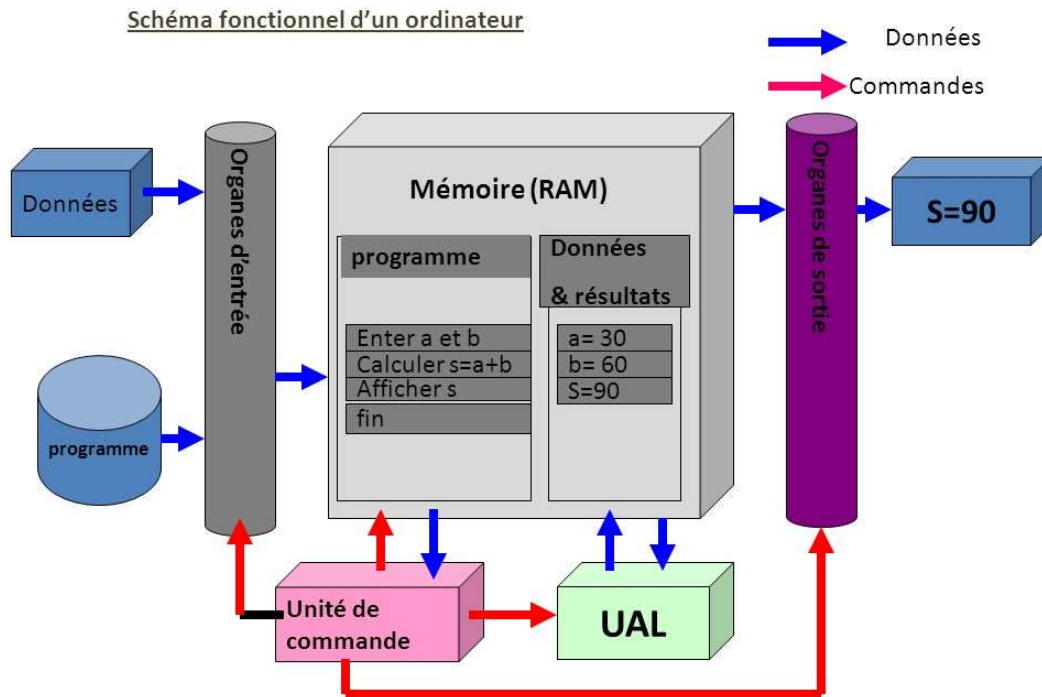
4) [Question](#)

Expliquez ce que sont les bus dans ce schéma, rappelez leur rôle



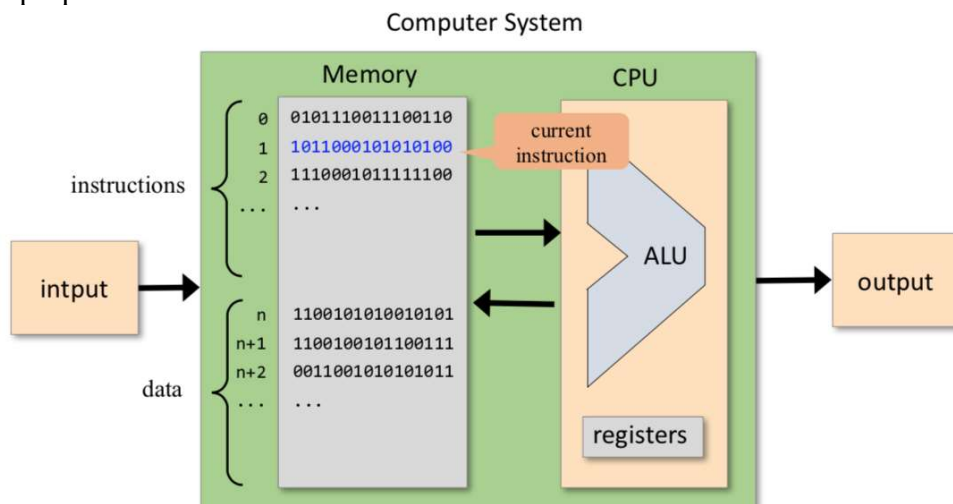
5) Question

Expliquez les flèches dans ce schéma



6) Question

Expliquez ce schéma ci-dessous



Handling instructions:

- 1011 means "addition" (operation)
- 000101010100 means "operate on memory address 340" (addressing)
- Next we have to execute the instruction at address 2 (control)

7) [Question](#)

Transformez les nombres en passant d'une base à une autre
Transformez :

1. 10_{10} en binaire sur 8 bits
2. 0110_2 en décimal
3. 129_{10} en binaire
4. 10110010_2 en décimal
5. 524_{10} en binaire
6. 110100110010_2 en décimal

8) [Question](#)

Transformez les nombres en passant d'une base à une autre
Transformez :

1. 123_{10} en hexadécimal
2. $111\ 0000\ 0110_2$ en hexadécimal
3. 129_{16} en binaire
4. $AF0_{16}$ en décimal
5. 524_{16} en binaire
6. $110100110010010011100110011100110111_2$ en hexadécimal
7. $192.168.0.252_{10}$ en hexadécimal

9) [Question](#)

Quelles sont les différences entre un kibioctet et un kilooctet ?
Quelles sont les différences entre un mébioctet et un mégaoctet ?

10) [Question](#)

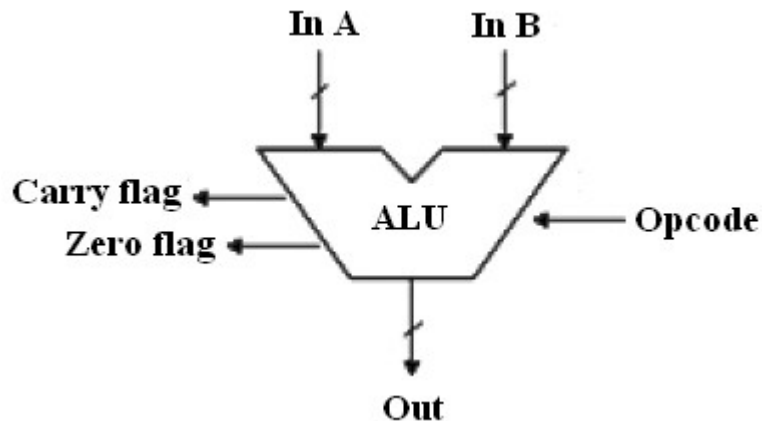
Quelles sont les différences entre un kibioctet et un kilooctet ?
Quelles sont les différences entre un mébioctet et un mégaoctet ?

11) [Question](#)

Avec 10 chiffres binaires, quel est le nombre maxi pouvant être codé en base 10 ? Combien de nombre peuvent être codés ?

12) [Question](#)

Pour une addition en binaire , effectuez une addition simple sur 1 bit avec en entrée 2 entrées sur 1 bit, avec toutes les valeurs possibles. Dans une ALU, il y a deux indicateurs un bit de retenue, et un bit indicateur de 0.



Faites une table avec en entrée les 2 bits, en sortie sur 1 bit le résultat et les valeurs, pour chaque cas, du bit de « carry flag » et du bit de « zero flag »

13) [Question](#)

Pour un codage de nombre négatif, on utilise la représentation « signe et valeur absolue »
 Quel est l'inconvénient d'opérations avec une représentation signe et valeur absolue ?
 Vous rappellerez le principe de cette représentation.
 Par exemple $10010011_2 = -19_{10}$

14) [Question](#)

Un nombre négatif est le complément à 2 d'un nombre positif. Un complément à 2 est le complément à 1 d'un nombre puis l'ajout d'un 1 sur le bit de poids faible. Le complément à 1 d'un nombre binaire est l'inversion de tous les bits d'un nombre binaire.
 Pour des entiers dans un système informatique, des « INT » sur 16 bits, utilisés dans un langage C, les nombres négatifs sont codés en complément à 2.
 Trouvez -52_{10} en binaire (codage sur 16 bits) en utilisant le complément à 2.

15) [Question](#)

En représentation en complément à 2, pourquoi y'a-t-il un nombre différent de représentation de nombres négatifs et positifs. Pour vous aider, prenez un exemple sur 8 bits.

16) [Question](#)

En utilisant l'arithmétique en complément à 2, effectuez l'addition de 118 et de -36

17) [Question](#)

Lorsqu'ils sont stockés en mémoire dans un ordinateur, les nombres binaires ont une taille limitée : souvent 1, 2 ou 4 octets.
 Parfois, le résultat d'une opération arithmétique, par exemple d'une addition, ne peut pas tenir dans la taille imposée. On dit alors qu'il y a **débordement (overflow)**. Comment le détecter ?
 Cela dépend de la représentation des nombres.

30/09/2022

Effectuez avec **des nombres non signés** l'opération $156 + 168$ après les avoir transcodés en binaire.

Effectuez le même chose pour $27 - 100$

Comment allez vous voir s'il y a un débordement ?

18) [Question](#)

Pour continuer à étudier l'**overflow**.

Effectuez avec des **nombres en complément à 2** l'opération $-60 + (-61)$ après les avoir transcodés en binaire.

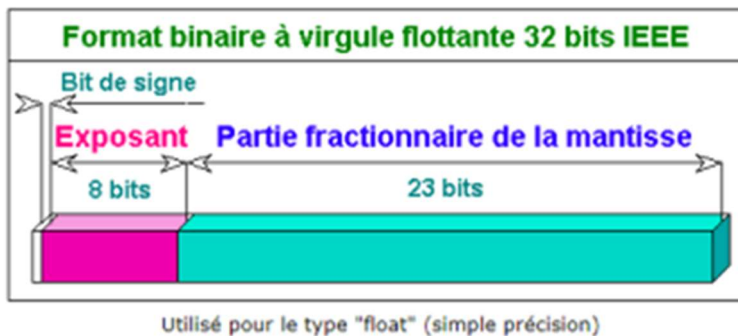
Effectuez le même chose pour $77 + 68$

19) [Question](#)

Codez en notation virgule fixe de 78,187510 sur 12 bits, 8 bits partie entière, 4 bits partie fractionnaire

20) [Question](#)

Codez sur 32 bits IEEE 754 en virgule flottante -1039,0. Vous vous servez de ce format.



Vous le mettrez aussi en hexadécimal.