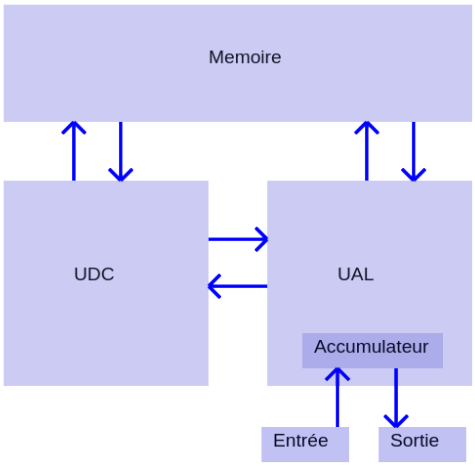
Système sur Puce (System on a chip - Soc)

I – Rappel de 1er Architecture matérielle

Schéma d'une machine de Von Neumann :



Cours 01 - Sur votre feuille, fournir les explications liées aux termes :

Mémoire

CPU = (UDC+UAL)

Registres dont Accumulateur

Entrée / Sortie

Bus de données – d'adresses – de contrôle.

On travaille avec M10 et ce jeu d'instructions très réduit :

- `ADD @2 @3` : additionne les contenus des zones-mémoires et place le résultat dans l'accumulateur.
- `SUB @2 @3` : Calcule contenu 2 - contenu 3 et place le résultat dans l'accumulateur.
- `STR @5` (store from register) : copie le contenu du registre-accumulateur en zone-mémoire 5.
- `LDR @5` (load to register): copie le contenu de la zone-mémoire 5 dans l'accumulateur.

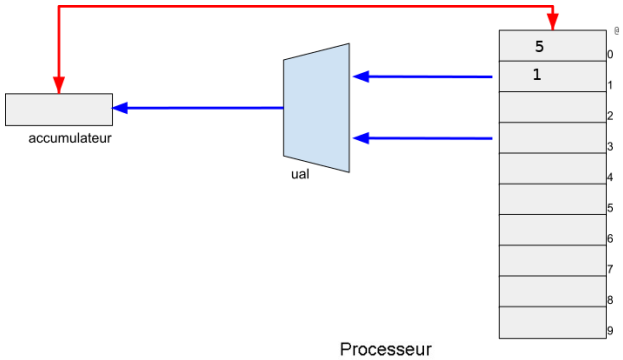
01° Exécuter le programme suivant en utilisant la mémoire initiale représentée ci-dessous. Que contiennent les mémoires à la fin ?

SUB @0 @1

STR @1

ADD @0 @1

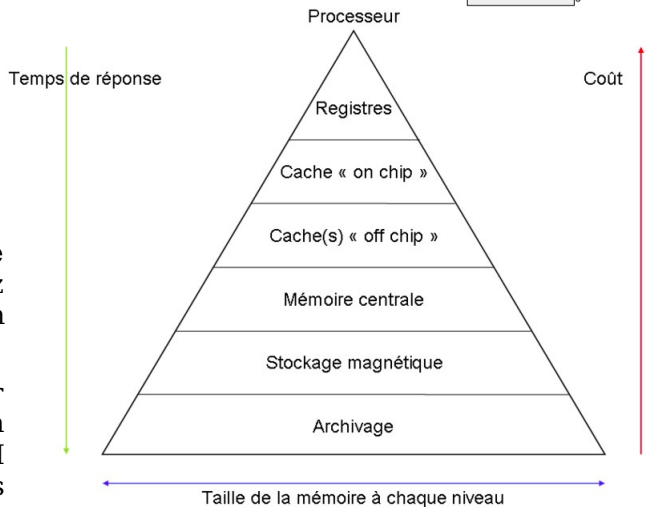
STR @2



Sur votre feuille répondre aux question ci-dessous :

Cours 02 - Expliquer la représentation ci-contre.

Cours 03 - Quelle est la différence entre une mémoire volatile et une mémoire non volatile ?



02° Que doit faire votre ordinateur si vous avez besoin de 4 Go de mémoire vive (RAM) mais que vous en avez uniquement 3 ? Quel est la conséquence pour l'exécution des programmes ?

03° Si on part du principe que le système doit pouvoir transporter en une seule opération une adresse via son bus d'adresses, combien d'adresses-mémoires RAM différentes peut-on avoir dans un ordinateur dont le bus d'adresse est un bus 16 bits ? Si on considère que chaque case mémoire correspond à un octet, quelle est la mémoire vive maximale disponible sur ce système s'il ne disposant pas d'autres manières d'adresser sa mémoire ?

04° Faire de même pour un ordinateur muni d'un processeur 32 bits, et de bus d'adresses de 32 bits.

05° Votre disque dur ou votre carte SD correspondent-t-ils à la mémoire vive ?

06° Combien d'opérations pour stocker ou lire un grand entier stocké sur 4 octets avec un bus de données de 8 bits connaissant l'adresse mémoire du premier octet ?

07° Combien d'opérations pour stocker ou lire un grand entier stocké sur 4 octets avec un bus de données de 32 bits connaissant l'adresse mémoire du premier octet ?

08° La même lecture d'un entier sur 4 octets va-t-elle être plus rapide avec un bus de données 64 bits ?

09° Dans quel cas le bus de données 32 bits va-t-il alors être plus efficace ?

II – Microprocesseur

Cours 04 - Que contient la puce d'un microprocesseur ?

Cours 05 – Quels sont les avantages d'un microprocesseur par rapport à un 'gros' processeur ?

Cours 06 – Qu'est-ce que la loi empirique de Moore ?

Cours 07 – Comment se nomme le composant qui permet au microprocesseur de communiquer avec le reste du système ?

III – M999

Chaque case mémoire peut contenir une valeur dans l'intervalle [0;999].

Notre processeur possède un registre accumulateur R (comme résultat), et deux registres généraux A et B qui peuvent servir, notamment, à stocker les opérandes.

Opérations de copie

- Une instruction **LDA** (load to A) de valeur $(\mathbf{0+adr})_2$: copie le contenu de l'adresse donnée dans A.
  Lorsqu'on exécute 045, la machine sait qu'elle doit placer dans A une copie du contenu de @45.
  Particularité de 099 : l'instruction **099** récupère l'entrée clavier dans A.

- Une instruction **LDB** (load to B) de valeur $(\mathbf{1+adr})_2$: copie le contenu de l'adresse donnée dans B.
  Lorsqu'on exécute 145, la machine copie le contenu de @45 dans le registre B.
  Particularité de 199 : l'instruction **199** récupère l'entrée clavier dans B.

- Une instruction **STR** (store from R) de valeur $(\mathbf{2+adr})_2$: copie le contenu R à l'adresse fournie.
  L'instruction 245 veut dire de copier le contenu de R vers @45.
  Particularité de **299** : l'instruction 299 veut dire d'afficher le régistre R sur l'écran.

- Une instruction **MOV** (move from rs to rd) de valeur $(\mathbf{4 + code\ rs + code\ rd})_2$: copie le contenu d'un registre source vers un registre destination. On considèrera au moins les codes suivants pour les registres :
  - Code 0 pour le registre A.
  - Code 1 pour le registre B.
  - Code 2 pour le registre R.

  L'instruction 402 veut dire de copier le contenu de A dans le registre R.

Opérations arithmétiques

- Une instruction **ADD** de valeur $\mathbf{300}_2$: additionne les contenus lues via A et B et place le résultat dans R.
- Une instruction **SUB** de valeur $\mathbf{301}_2$: soustrait le contenu de B au contenu de A et place le résultat "A - B" dans R.
- Une instruction **NOP** de valeur $\mathbf{399}_2$: on ne fait rien.

Cours 08 - Comment l'UDC parvient-elle à savoir ce qu'elle doit faire au démarrage ?

Cours 09 - Comment l'unité de commande (UDC) parvient-elle à savoir ce qu'elle doit faire ensuite ?

10° Utiliser le premier programme fourni pour voir ce qu'il fait.

Contenu de @00 : 399
Contenu de @01 : 006
Contenu de @02 : 402
Contenu de @03 : 299
Contenu de @04 : -
Contenu de @05 : -
Contenu de @06 : 123

Cours 10 - Comment l'unité de commande (UDC) parvient-elle à savoir qu'elle doit cesser de fonctionner ?

Cours 11 - Du coup, on fait comment pour dire à la machine de stopper alors que PC pointe encore sur une autre adresse que @99 ?

Opérations de branchement

- Une instruction **JPM** (Jump to ou Go to) de valeur **5+adr**$_2$: on place l'adresse fournie dans le registre PC, modifant du même coup la prochaine instruction à réaliser.
  De cette façon 599 veut dire de placer @99 dans PC et ainsi la prochaine instruction réaliser pointera vers @99 et signifie donc l'arrêt de la machine. On pourrait ainsi créer une instruction assembleur HLT qui ne serait que l'instruction JPM @99, autrement dit 599 en mémoire.

- Une instruction **JPP** (jump if positive) de valeur **6+adr**$_2$: on modifie la valeur PC par l'adresse fournie SI le registre R contient une valeur strictement positive. Sinon, on ne modifie pas PC.

11° Utiliser le programme fourni pour voir ce qu'il fait.

12° Même question mais on inverse les contenus mémoire de @11 et @12.

13° On revient au programme de la question 11 mais l'adresse @1 contient 514.

Cours 12 – Qu'est-ce qui différencie globalement un microprocesseur à jeu d'instructions réduit et un microprocesseur à jeu d'instruction étendu ?

Cours 13 – Un programme créé avec un langage interprété peut-il tourner sur n'importe quelle machine quelque soit son microprocesseur ?

Cours 14 – Un programme créé avec un langage compilé peut-il tourner sur n'importe quelle machine quelque soit son microprocesseur ?

IV – Microcontrôleur

Cours 15 – Qu'est-ce qu'un microcontrôleur ?

V – Système sur puce - Soc

La puce contient le microprocesseur (ou le microcontrôleur) ainsi que tout ce qui permet au système informatique de fonctionner !

- Le processeur
- La mémoire vive
- Les processeurs esclaves (le GPU, la carte son, la carte de chiffrement...)
- Beaucoup de capteurs, d'actionneurs ou de dispositifs de communication :
  - cartes réseaux et antennes Wifi, Bluetooth, radio
  - cartes et antenne GPS
  - capteurs de type accéléromètre, magnétomètre...
- Le gestionnaire d'énergie

Cours 16 – Quels sont les deux avantages principaux d'un Soc ?