

Nom:  
Prénom:  
Groupe:

## Contrôle Court n°2

*Calculatrice et documents interdits - Durée 1h30 - Répondre sur la feuille*

### **1. QUESTIONS DE COURS SUR LA STRUCTURE DES ORDINATEURS**

1.1. Qu'est-ce qu'une variable ? Répondre en donnant ses trois caractéristiques.

Pour stocker et accéder à une donnée.

Une adresse (associée au symbole/nom)

Une taille (associée au type)

Une interprétation (quand on veut l'afficher par exemple)

1.2. Dans une architecture Von Neumann, où sont les données, où sont les programmes ?

Les données sont associées à des variables, en mémoire.

Les programmes sont composés d'instructions, en mémoire.

Dans une architecture Von Neumann, il s'agit de la même mémoire.

1.3. Qu'est-ce qu'une instruction ? Répondre en expliquant comment elle s'écrit dans la machine.

Un code de commande en binaire, sur un certain nombre de bits (octets).

Code Opération (type d'opération),

Eventuellement suivit d'un ou plusieurs Code Opérande (paramètres).

1.4. Comment le CPU sait-il où est la prochaine instruction à exécuter ? Que fait-il pour l'examiner ?

Il note son adresse dans le registre pointeur d'instruction (mis à jour par compteur ordinal)

Pour l'examiner, il la charge à partir de la mémoire via le bus de données vers un registre.

Une fois dans ce registre, il peut la décoder.

1.5. Qu'est-ce qu'un mode d'adressage ? Quel registre particulier est utilisé pour l'adressage direct ?

Méthode utilisée pour interpréter l'opérande (absente, valeur, adresse...)

En adressage direct, l'opérande est une adresse.

Chargée par le bus d'adresse vers un Registre Tampon pour les Adresses ;

Puis envoyée sur le bus d'adresse pour désigner une donnée en mémoire.

1.6. Qu'est-ce que l'adressage indirect ? A quoi sert-il usuellement ?

L'opérande n'est pas donnée (implicite),

mais elle est dans un registre (spécifié par code opération).

Cette valeur est une adresse (comme en direct et non comme implicite ou immédiat)

Utilisé lors d'utilisation (parcours) de tableaux (ou de chaînes).

1.7. Qu'est-ce que la pile ? Quelles opérations fait-on avec ? Quel registre permet sa gestion ?

Espace de stockage en mémoire comme un tableau à gestion d'accès automatique.

On peut empiler dessus ou dépiler des valeurs (LIFO).

Cette gestion est réalisée grâce au pointeur de haut de pile SP.

## **2. EXERCICES SUR LA STRUCTURE DES ORDINATEURS**

2.1. Représentez les éléments principaux d'un processeur (Von Neumann) a 1 accumulateur relié à une mémoire (64ko en mots de 8bits). Son jeu d'instruction permet l'adressage direct, indirect indexé et l'empilement. Précisez la taille des bus et des registres.

Bus de données 8 bits, bus d'adresse 16 bits

ALU, ACC, RTUAL, (Flags)

Décodeur, RI, IP

RTA, Base indirect, SP

2.2. Détaillez les étapes de l'exécution de l'instruction: "AND AX, [BP+6]"  
(masque l'accumulateur 16 bits avec la valeur 6 octets au dessous de BP dans la pile).

Etat : @instruction stockée dans IP, IP sur le bus d'@, bus data sur RI.

- Valide le chargement du code instruction dans RI,  
Valide augmentation IP pour pointer l'opérande.
  - aiguille IP sur bus d'adresse
  - aiguille BP sur une UAL
  - et aiguille l'opérande (index) via le bus données vers l'RT de cette UAL
- Valide le chargement de l'opérande.
  - l'UAL est réglée en addition pour faire BP+index,
  - aiguille la sortie de l'UAL sur bus d'adresse
  - et aiguille valeur pointée via bus de données sur RTUAL
- Valide le chargement de la valeur dans RTUAL.
  - l'UAL est réglée en AND,
  - ACC sur UAL
- Valide le calcul,  
Valide augmentation IP pour pointer l'instruction suivante si ça n'a pas encore été fait
  - aiguille IP sur bus d'adresse
  - aiguille IP sur bus d'adresse

2.3. L'instruction AND AX, 06 a pour code 234606. Elle est implantée à l'adresse 0100.

AX vaut B000. BP vaut FF00 et la valeur pointée vaut FFFF.

Donnez l'adresse de la valeur pointée ainsi que le contenu des registres après exécution de l'instruction.

RI=2346      IP=0103

AX=FFFF      RTAX=FFFF

BP=FF00      (RTA+/-=FF06)

Mémoire inchangée

### **3.COMPILEATION**

3.1. Quelles sont les principales étapes de la compilation ? Quels sont leurs rôles ?

Compilation : sert à transformer du texte en code machine

Edition de liens : rassemble les morceaux

3.2. Que fait l'instruction machine d'appel à une fonction ? Précisez comment se passe le retour.

Empile l'adresse de retour, charge ip avec l'opérande

Au retour on dépile l'adresse de retour dans ip

3.3. Quelles sont les règles de compilation du langage C pour les variables locales, les paramètres et les valeurs retournées ?

Variables interne sur pile dans l'ordre (1<sup>er</sup> déclaré => 1<sup>er</sup> sur la pile)

Paramètres sur pile à partir de la droite (@ret dernière sur la pile)

Passage par registre UN registre (l'accumulateur) donc UNE seule valeur de retour

3.4. Donnez dans l'ordre ce que l'on empile lors d'un appel à une fonction et précisez pour chaque objet empilé pourquoi on utilise la pile.

D'abord les paramètres ;

on peut en mettre autant que l'on veut

et ce n'est une copie des valeurs (on ne donne pas accès aux originaux).

Ensuite l'@retour ;

pour permettre retours lors d'appels imbriqués de fonctions.

Ensuite une copie de BP ;

une sauvegarde temporaire pour redéfinir "la base" de la pile de la fonction.

Ensuite les variables locales ;

on peut en mettre autant que l'on veut

et elles sont locales.