

L'objectif de ce TD est de vous faire prendre conscience qu'il n'y a aucune intelligence dans un ordinateur, quel qu'il soit. Ce dernier ne fait que suivre les séquences d'instructions qui ont été données par l'humain qui l'a programmé.

Présentation

La machine M99 est un ordinateur en papier doté de 100 cases mémoire et d'un processeur. On vous offre cet ordinateur à chacun pour ce TD.

Mémoire

La mémoire est composée de 100 mots mémoires de 3 chiffres décimaux (valeur de 000 à 999) et d'un signe. Ces 100 mots mémoires sont adressables par des nombres sur 2 chiffres (de 0 à 99).

Processeur

Le processeur dispose de deux *registres généraux* nommés A et B et d'un registre accumulateur/résultat nommé R. Comme les mots mémoires, ces registres peuvent contenir des nombres signés sur 3 chiffres. Le processeur dispose aussi d'un quatrième registre nommé PC (*Program Counter*). C'est le pointeur d'instruction, contenant l'adresse mémoire de la prochaine instruction à exécuter.

L'unité arithmétique et logique *UAL* est en charge d'effectuer les calculs. Les opérandes et résultats sont dans les registres :

- A et B pour les opérandes,
- R pour le résultat.

L'unité de commande pilote l'ordinateur. Son cycle de fonctionnement, nommé *fetch / decode / execute*, comporte 3 étapes :

1. **charger l'instruction** depuis la case mémoire pointée par PC vers la zone dédiée dans l'UAL (la case sous le registre PC) et **incrémenter ensuite le PC** ;
2. **décoder l'instruction** : à partir des 3 chiffres codant l'instruction, identifier quelle est l'opération à réaliser (vous avez un pense-bête à droite de l'UAL) ;
3. **exécuter l'instruction**.

La machine démarre avec la valeur nulle comme pointeur d'instruction (PC=0). Elle s'arrête si le pointeur d'instruction vaut 99. On peut donc utiliser le mnémonique `HLT` comme synonyme de `JMP 99`.

Les entrées/sorties sont *mappées* en mémoire :

- écrire une valeur dans le mot mémoire à l'adresse 99 écrit cette valeur sur le terminal,
- les valeurs saisies sur le terminal seront lues dans le mot mémoire 99.

Dans ce TD c'est vous qui jouerez le rôle du processeur (unité de commande et UAL).

Jeu d'instruction

code	mnémonique	instruction à réaliser
0 x y	STR xy	copie le contenu du registre R dans le mot mémoire d'adresse xy
1 x y	LDA xy	copie le mot mémoire d'adresse xy dans le registre A
2 x y	LDB xy	copie le mot mémoire d'adresse xy dans le registre B
3 x y	MOV x y	copie registre x dans y (0 → R; 1 → A; 2 → B)
4 - -	opérations arithmétiques et logiques	
4 0 0	ADD	ajoute les valeurs des registres A et B, produit le résultat dans R (R := A + B)
4 0 1	SUB	soustrait la valeur du registre B à celle du registre A, produit le résultat dans R (R := A - B)
5 x y	JMP x y	branche en xy (PC reçoit la valeur xy). JMP 99 provoque l'arrêt de la machine
6 x y	JPP x y	branche en xy si la valeur du registre R est positive
7 x y	JEQ x y	saute une case (PC += 2) si la valeur du registre R est égale à xy
8 x y	JNE x y	saute une case (PC += 2) si la valeur du registre R est différente de xy

Exercices

Exercice 1 : Prise en main

- Q1.** Décodez (directement sur les cases mémoire de votre ordinateur) le programme chargé de l'adresse 0 à l'adresse 11.
- Q2.** En appliquant le cycle de commande jusqu'à un arrêt, expliquez ce que fait ce programme.
- Q3.** Décodez le programme débutant à l'adresse 13 puis essayez d'expliquer ce qu'il fait.
- Q4.** Écrivez un programme (en utilisant les mnémoniques) affichant le maximum de deux nombres saisis au clavier.
- Q5.** Codez votre programme. À quelle adresse mémoire pouvez-vous le stocker ?

Exercice 2 : Assembleur et compilateur

- Q1.** Que fait le programme débutant à l'adresse 40 (pour les entrées 5 et 2) ?
- Q2.** Peut-on le raccourcir ?
- Q3.** Corrigez ce programme quand la seconde entrée vaut 0.