

Architecture d'un ordinateur: introduction au modèle de Von Neumann (TD):

Le M99 est doté de 100 cases mémoire (la grille en haut), et d'un processeur (en bas). La mémoire est composée de 100 mots mémoire de 3 chiffres (valeur de 000 à 999). Chaque mot mémoire peut être désigné par une adresse codée sur deux chiffres. Cette mémoire va contenir données et instructions.

Unité Arithmétique et Logique (ALU): Le processeur dispose de trois registres directement utilisables: A et B sont utilisés pour les opérandes des opérations tandis que R est pour le résultat. Ces registres sont de 3 chiffres, mais contrairement à la mémoire, ils ont un signe. Leurs valeurs sont donc comprises entre -999 et 999. 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. Lorsqu'on utilise le M99, on peut noter le numéro de l'instruction à exécuter dans la case prévue à cette effet, mais en pratique, il est plus simple de le matérialiser avec un pion situé sur une des cases de la grille mémoire, ou même de suivre avec son doigt.

Unité de commande: Elle pilote l'ordinateur. Son cycle de fonctionnement comporte 3 étapes:

1. (fetch) charger l'instruction depuis la case mémoire pointée par PC dans la case instr (sous le PC).
2. (decode) identifier l'opération à réaliser à partir des 3 chiffres la codant et grâce au pense-bête de droite.
3. (execute) exécuter cette instruction, puis incrémenter ensuite le PC.

Premier jeu d'instructions:

<i>Code</i>	<i>Mnémonique</i>	<i>Instruction à réaliser</i>
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 Rx dans Ry (R0: R; R1: A; R2: B)
4 0 0	ADD	Ajoute les valeurs des registres A et B, produit le résultat dans R
4 0 1	SUB	Soustrait la valeur du registre B à celle du registre A, produit le résultat dans R
4 x y	etc	(d'autres opérations arithmétiques et logiques peuvent être encodées ainsi)
5 x y	JMP x y	Branche (saute) en xy (PC reçoit la valeur xy)
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++) si la valeur du registre R est égale à xy
8 x y	JNE x y	Saute une case (PC++) si la valeur du registre R est différent de xy

PC est incrémenté deux fois si la condition de JEQ ou JNE est vraie: par l'unité de commande et par l'opération. On peut utiliser le mnemotechnique DAT x y z pour stocker une valeur numérique arbitraire xyz en mémoire. Les dépassements de capacité (overflow) sont gérés de manière cyclique: $999 + 1 = 0$; $0 - 10 = 990$.

Boot et arrêt: La machine démarre avec la valeur nulle comme pointeur d'instruction (PC=0) et elle s'arrête si le pointeur d'instruction vaut 99. On peut donc utiliser le mnémonique HLT comme synonyme de JMP 99. En cas de faute (comme par exemple une division par zéro), le programme est interrompu (PC := 99).

Entrées/sorties: Les entrées/sorties sont mappées en mémoire: Écrire le mot mémoire 99 écrit sur le terminal, tandis que les valeurs saisies sur le terminal seront lues dans le mot mémoire 99.

Questions:

1. Que fait le programme chargé à l'adresse 0?
2. Que fait le programme chargé à l'adresse 13?
3. Écrire à partir de l'adresse 20 un programme affichant le maximum de deux entrées clavier.
4. Écrire un programme réalisant la somme de deux entrées clavier.

Source: Cette activité de découverte a été inventée par le groupe InfoSansOrdi: <https://github.com/InfoSansOrdi/M999>