

TP assembleur

I Présentation du M99 un ordinateur en papier

Le M99 est une machine dotée de 100 cases mémoire (la grille en haut de la feuille), et d'un processeur (en bas de la feuille).

La mémoire est composée de 100 mots mémoire de 3 chiffres (valeur de 000 à 999). Ces 100 mots mémoire sont adressables par des adresses codées sur 2 chiffres. Cette mémoire va contenir données et instructions.

Le processeur dispose de deux registres généraux nommés A et B, et d'un registre accumulateur/résultat nommé R. Ces registres sont de 3 chiffres, mais au contraire de la mémoire, ils ont un signe. Ils peuvent donc contenir des valeurs comprises entre -999 et 999 .

Le processeur dispose aussi d'un quatrième registre nommé PC (ProgramCounter). 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 cet 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 arithmétique et logique

L'unité arithmétique et logique – ALU en anglais – 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.

Unité de commande

L'unité de commande pilote l'ordinateur. Son cycle de fonctionnement comporte 3 étapes :

1. charger l'instruction depuis la case mémoire pointée par PC vers la zone dédiée dans l'ALU (la case sous le registre PC). Incrémenter ensuite le PC.
2. décoder l'instruction : à partir des 3 chiffres codant l'instruction, identifier quelle est l'opération à réaliser en utilisant le pense-bête à droite de l'ALU.
3. exécuter l'instruction.

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 Rx dans Ry (R0: R; R1: A; R2: B)
4 - -	opérations arithmétique et logiques	
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
...	etc	...
5 x y	JMP x y	branche 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 = PC +2) si la valeur du registre R est égale à xy
8 x y	JNE x y	saute une case (PC = PC +2) si la valeur du registre R est différent de xy

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.

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.

II Exercice 1

Objectifs :

- prise en main du M99
- comprendre l'encodage des opcode dans la mémoire
- comprendre le cycle fetch/decode/execute

Q1: Que fait le programme chargé à l'adresse 0 ?

Pour répondre, il faut appliquer le cycle fetch/decode/exec en partant de l'adresse 0. Au fur et à mesure, il faudra traduire les codes en mnémoniques.

Recopier puis compléter le tableau ci-dessous pour suivre le programme :

PC : commande // explications	R	A	B
00: LDA 10 // Charge le contenu de la case 10 dans le registre A	42	42	
...			

Q2: Que fait le programme débutant à l'adresse 13?

Q3: Écrire un programme affichant le maximum de deux entrées clavier.

Écrire ce programme à partir de l'adresse mémoire 20. Les deux valeurs entrées seront mises respectivement dans le registre A et dans le registre B. En effet, comme elles sont utilisées immédiatement il n'est pas nécessaire de les écrire en mémoire.

III Exercice 2

Objectif : Voir l'intérêt d'un compilateur, et d'un langage de haut niveau

Q1: Que fait le programme débutant à l'adresse 40 ?

Faire fonctionner le programme avec pour entrées 5 et 2.

On remarquera que l'adresse 61 contiendra toujours le nombre 1

Q2: Que se passe-t-il si les entrées sont 5 et 0 ?

Faire fonctionner le programme avec les entrées 5 et 0. Qu'obtient-on ?

Source :

<https://github.com/InfoSansOrdi/M999>