CHIMIA Vol. 25 Fasc. 5-1970 Pag. 141–180

Anwendung der Computer in der Chemie

Diesem heute hochaktuellen Thema war der wissenschaftliche Teil der Wintertagung des Schweizerischen Chemiker-Verbandes vom 6. Februar 1971 in Lausanne gewidmet. Wir bringen nachfolgend die vier bei diesem Anlaß gehaltenen Vorträge (NICOUD, GÄUMANN, STAHL, REGENASS) sowie einen thematisch zugehörigen Aufsatz (FRAADE), dessen Inhalt bereits am 3. Juni 1970 vom Autor in Lausanne vorgetragen wurde.

Organisation d'un ordinateur*

Par J.-D. NICOUD**
Ecole Polytechnique Fédérale, Lausanne

Summary

This article conveys some basic knowledges about small computers. It describes briefly how all information can be coded in binary, of what the central unit of a computer is made of, how a programme can be executed and what the various programming languages are.

1. Introduction

L'ordinateur s'est imposé pour le calcul scientifique et les tâches de gestions; il pénètre maintenant de plus en plus dans les laboratoires pour décharger les expérimentateurs de leur travail de routine et permettre de nouveaux types d'expériences.

Un très petit ordinateur (on parle du reste de mini-ordinateur) dont le prix est bien inférieur au prix de l'appareillage utilisé pour l'expérience elle-même, peut contrôler le déroulement de cette expérience, enregistrer les résultats principaux et effectuer au fur et à mesure tous les calculs nécessaires. Les appareils récents voient ainsi leurs possibilités et leur rentabilité multipliées.

L'utilisation judicieuse d'un mini-ordinateur nécessite naturellement quelques connaissances sur sa structure et sa programmation; cet article se propose de donner au lecteur une idée générale et lui faciliter l'approche de la documentation spécialisée.

2. Formes de l'information

Toute expérience conduit à recueillir un certain nombre d'informations sur le phénomène étudié. Ces informations peuvent être classées selon deux groupes principaux:

Lorsque les grandeurs sont discrètes ou quantifiées, comme par exemple une somme d'argent, un numéro atomique ou la présence ou non d'une réaction, un nombre leur correspond de façon naturelle et il s'agit d'une information *digitale*.

Au contraire, si les valeurs mesurées peuvent admettre des variations continues comme par exemple un volume de liquide, une tension électrique ou un signal sonore, on parle d'information *analogique*.

En choisissant une certaine unité et en procédant par approximation, on peut toujours faire correspondre à une grandeur analogique une grandeur digitale.

A ces deux formes de l'information correspondent deux types de calculatrices:

Les ordinateurs, ou calculatrices digitales, ne peuvent travailler que sur des nombres ou des informations codées. Les calculatrices analogiques au contraire traitent des signaux à variation continue.

Le champ d'application des calculatrices purement analogiques est restreint. La représentation digitale de l'information présente en effet des avantages importants: la précision n'est pas limitée, les résultats peuvent être imprimés directement et des calculs complexes peuvent être effectués sur les grandeurs mesurées. De plus, grâce aux circuits logiques électroniques, la rapidité de traitement de l'information est considérable.

Les circuits logiques imposent toutefois une contrainte absolue: toute information doit être représentée en *binaire*, sous forme d'un suite de «0» et de «1», ces deux états correspondant pour chaque circuit élémentaire à deux niveaux de courant ou de tension bien distincts.

3. Représentation des nombres

A chaque nombre réel on peut faire correspondre son équivalent dans le système de numération binaire. Le tableau I donne les équivalents des 20 premiers nombres entiers et quelques exemples. La traduction de décimal en binaire et de binaire en décimal peut être faite en appliquant quelques règles de calcul simples; en général,

* Présenté à l'assemblée d'hiver de l'Association Suisse des Chimistes le 6 février 1971 à Lausanne.

** Adresse: Chaire de Systèmes logiques de l'EPF-L, chemin de Bellerive 16, 1007 Lausanne.

Tableau I. Equivalence de quelques nombres dans les systèmes décimal, binaire, octal et décimal codé binaire (BCD)

Décimal	Binaire	Octal	BCD
0	0	0	0000
1	1	1	0001
2	10	2	0010
3	11	3	0011
4	100	4	0100
5	101	5	0101
6	110	6	0110
7	111	7	0111
8	1000	10	1000
9	1001	11	1001
10	1010	12	0001 0000
11	1011	13	0001 0001
12	1100	14	0001 0010
13	1101	15	0001 0011
14	1110	16	0001 0100
15	1111	17	0001 0101
16	10000	20	0001 0110
17	10001	21	0001 0111
18	10010	22	0001 1000
19	10011	23	0001 1001
20	10100	24	0010 0000
109	1101101	155	0001 0000 1001
4,5	100,1	4,4	0100,0101

cette conversion n'est pas un souci pour l'utilisateur; l'ordinateur lui-même s'en charge.

L'écriture des nombres binaires est rébarbative et des erreurs peuvent apparaître facilement si l'opérateur doit manipuler ces longues chaînes de «0» et de «1». Pour cette raison, on remplace généralement les nombres binaires par leurs équivalents dans le système octal de base 8. La règle de conversion est très simple: de part d'autre de la virgule, les groupes de trois bits (chiffres binaires) sont remplacés par les chiffres de 0 à 7 équivalents (tableau I).

C'est ainsi par exemple que les numéros d'ordre des instructions, comptées en binaire par la machine, sont imprimées en octal lors du «listing» du programme.

Un autre procédé est souvent utilisé pour coder en binaire les nombres décimaux. Il consiste à conserver la structure générale du nombre décimal (unités, dizaines, etc.) et à remplacer chaque chiffre par un groupe de 4 bits au moins.

Par exemple dans le code BCD (Binary Coded Decimal), chaque chiffre est remplacé par son équivalent binaire (tableau I). La traduction est plus facile qu'en binaire pur, mais la représentation en BCD n'est pas aussi «économique», car elle nécessite plus de bits pour représenter le même nombre. De plus, les opérations arithmétiques sont délicates à exécuter en BCD.

Ce code n'est donc généralement pas utilisé à l'intérieur même de l'ordinateur, lors de l'exécution des instructions. Par contre, la plupart des équipements périphériques (imprimante, volt-mètre digital, etc.) travaillent avec le code BCD ou un code similaire.

4. Significations possibles d'un mot binaire

Un mot binaire peut représenter n'importe quelle information, pour autant qu'un code convenable ait été défini. Aux lettres et signes d'une machine à écrire correspondent divers codes normalisés à 5, 6, 7 ou 8 bits (tableau II). Un groupe de bits peut également être associé à un ensemble d'opérations fort complexe.

Par exemple, le même mot binaire de 12 bits 001 101 010 011 (en octal 1523) peut représenter suivant les cas:

a) un nombre écrit en binaire correspondant au nombre décimal 851,
b) un nombre codé en BCD 0011 0101 0011 équivalent par conséquent au nombre décimal 353,
c) le groupement des deux lettres MS dans le code ASCII du tableau II,
d) dans un certain type d'ordinateur l'instruction «ajouter à l'accumulateur le nombre dont l'adresse se trouve dans la position mémoire d'adresse 523».

On voit donc que cette notion de codage de l'information est extrêmement importante pour bien comprendre le fonctionnement d'un ordinateur.

Tableau II. Code ASCII à 6 bits pour la représentation des lettres, des chiffres et des signes (extrait)

Caractère	Binaire	Octal	Caractère	Binaire	Octal
A	000 001	01	0	110 000	60
B	000 010	02	1	110 001	61
C	000 011	03	2	110 010	62
D	000 100	04	3	110 011	63
E	000 101	05	4	110 100	64
F	000 110	06	5	110 101	65
G	000 111	07	6	110 110	66
H	001 000	10	7	110 111	67
I	001 001	11	8	111 000	70
J	001 010	12	9	111 001	71
K	001 011	13			
L	001 100	14	Espace	100 000	40
M	001 101	15	!	100 001	41
N	001 110	16	"	100 010	42
	...			...	
Z	011 010	32	/	101 111	57

5. Structure d'un ordinateur

Un ordinateur permet de manipuler des mots binaires d'une certaine longueur (12 bits ou 16 bits dans les mini-ordinateurs). Il peut effectuer des opérations très simples sur ces mots et en stocker un très grand nombre dans sa mémoire. La structure de l'unité centrale d'un petit ordinateur peut être décomposée en trois blocs principaux (figure 1).

La *mémoire principale* contient quelques milliers de mots. Ce n'est dans le fond rien d'autre qu'un ensemble de cases dans lesquelles on peut mettre et aller rechercher un mot. Chaque case porte un numéro d'ordre binaire

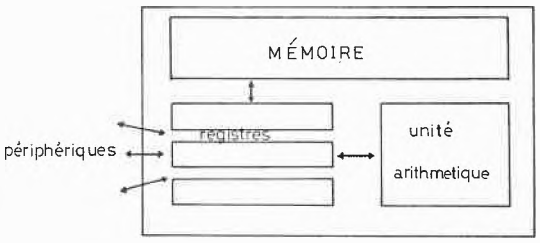


Fig. 1. Schéma-bloc de l'unité centrale d'un ordinateur

(ou octal) appelé *adresse* et les différents mots binaires stockés dans la mémoire peuvent être retrouvés grâce à leur adresse.

L'*unité arithmétique* permet d'effectuer quelques opérations simples sur des mots binaires: addition, soustraction, décalage du mot à droite ou à gauche, masquage de certains bits. Un décalage d'une position à gauche correspond à une multiplication par deux, à droite à une division par deux. Les opérations plus complexes sont remplacées par des suites d'opérations simples.

Les registres peuvent comme la mémoire contenir des mots binaires. On y place des mots «en transit» entre la mémoire, l'unité arithmétique et les périphériques ainsi que des mots «en observation» qui vont agir sur le comportement de la machine.

6. Principe de fonctionnement

Dans un ordinateur, toutes les informations nécessaires à la résolution d'un problème ou à l'accomplissement d'une tâche sont préalablement stockées dans la mémoire. Celle-ci contient alors deux types de mots: les *instructions* et les *données*.

Les instructions comportent deux parties; les trois ou quatre premiers bits correspondent au *code opératoire*, qui précise la nature de l'opération à effectuer, et les bits suivants représentent l'adresse d'une position de mémoire. Par exemple, dans un ordinateur utilisant des mots de 12 bits, l'instruction 001011011100 (en octal 1334) signifie «additionner» (sens des trois premiers bits) le nombre situé dans la position de mémoire 334 (adresse donnée par les neuf derniers bits).

Les données sont des nombres binaires, ou des textes codés (cf. § 4).

Les instructions sont groupées dans la mémoire dans l'ordre où elles doivent être effectuées et les données sont placées dans une autre zone de la mémoire.

Lorsque l'ordinateur reçoit l'impulsion «départ», il transfère l'instruction qui est dans la première position de mémoire dans l'un des registres de la machine, le *registre d'instruction*. Cette instruction indique à l'ordinateur ce qu'il doit faire (additionner, comparer, sauter une portion de programme, imprimer un résultat, etc.) et quelle donnée (représentée par l'adresse de la position de mémoire où il se trouve) doit intervenir.

L'instruction est exécutée en allant chercher dans la mémoire la donnée correspondante et en effectuant l'opération indiquée. Le résultat est laissé dans un registre appelé *accumulateur*.

Une nouvelle instruction est alors recherchée dans la mémoire, dans la position qui suit celle de l'instruction qui vient d'être effectuée, et ainsi de suite.

Les deux phases «recherche de l'instruction suivante» et «exécution de cette instruction» se suivent jusqu'à la fin du programme, avec toutefois toutes les possibilités, grâce à certaines instructions spéciales, de rompre la séquence normale, de parcourir plusieurs fois la même suite d'instructions (sous-programme) ou de modifier par le programme lui-même l'effet d'une instruction.

7. Langages de programmation

Lors de l'exécution d'un programme, la mémoire ne peut pas contenir autre chose que les instructions binaires qui viennent d'être évoquées. Ce *langage machine* (tableau III) change pour chaque type d'ordinateur et est très malcommode à utiliser, car le programmeur doit savoir où se trouve exactement chacune des informations qu'il doit mettre en mémoire.

Tableau III. Exemples de rédaction d'un programme calculant un résultat égal à la différence de deux températures dans différents langages

a)	001 011 011 100 010 000 110 010 001 001 010 000 100 111 101 010 Programme machine binaire
b)	1334 2062 1120 4752 Programme machine octal
c)	ADD TEMP 1 COMPL ADD TEMP 2 STO RESULT Programme en langage symbolique
d)	RESULT = − TEMP 1 + TEMP 2 Programme en langage évolué (FORTRAN)

Les *langages symboliques* ou langages d'assemblage évitent cet inconvénient en substituant aux numéros des positions de mémoire des noms, plus faciles à mémoriser pour le programmeur. Des abbréviations mnémotechniques remplacent également les codes opératoires binaires.

Ainsi, au lieu d'écrire 001011011100 (octal 1334) correspondant à l'instruction d'addition d'une certaine température dont la valeur a été placée dans la position de mémoire 334, le programmeur notera en langage symbolique ADD TEMP 1, ce qui est beaucoup plus proche de son mode de pensée.

Un programme spécial d'ordinateur, appelé *assembleur*, doit alors effectuer la traduction de l'instruction symbolique écrite par le programmeur en une instruction machine exécutable.

Pour faciliter encore plus la tâche du programmeur amené à résoudre toujours le même genre de problème, des *langages évolués* ont été définis. Ils sont indépendants de l'ordinateur utilisé et sont particulièrement bien adaptés à certaines tâches. Le FORTRAN simplifie considérablement l'écriture des expressions algébriques (tableau III), le COBOL est prévu pour les tâches de gestion, le BASIC pour le dialogue direct à partir d'un télétype.

Pour chaque langage, un programme de traduction adéquat appelé *compilateur* doit être livré avec l'ordinateur pour analyser les instructions évoluées et les convertir en des suites d'instructions machine. Ces programmes sont très complexes et dans les mini-ordinateurs, la petitesse de la mémoire limite leur utilisation.

L'ensemble des programmes de traduction, de contrôle et de résolution de problèmes courants forme ce que l'on appelle le «software» de la machine, par opposition au «hardware» qui représente la machine elle-même et ses périphériques. La qualité et la souplesse tant du software que du hardware participent à l'efficacité d'une installation.

8. Conclusion

Les ordinateurs sont formés de circuits électroniques simples, mais très nombreux. Le développement des circuits intégrés entraîne donc des diminutions importantes du prix et des dimensions des ordinateurs.

Mentionnons par exemple que l'on peut actuellement en une seule suite d'opérations photographiques et chimiques faire apparaître sur une pastille de silicium de quelques mm^2 un circuit comportant 10000 transistors et correspondant par exemple à toute l'unité arithmétique d'un ordinateur. Cette miniaturisation est de plus très favorable à une augmentation générale des performances. Les appareils périphériques tels que les lecteurs de bande, mémoires magnétiques à cassette, imprimantes, convertisseurs analogiques-digitaux bénéficient également de cette évolution.

L'ordinateur est un outil puissant; il est heureux que les progrès de la technologie mettent cet outil à la disposition de chacun.

Bibliographie

B. S. WALKER, *Einführung in die Computer-Technik*, Safari-Verlag, Berlin 1969.

Burroughs Corporation, *Digital Computer Principles*, McGraw-Hill, New-York 1969.

M. PONTE et P. BRAILLARD, *L'informatique*, Collection Le Rayon de la Science, Seuil, Paris 1969.