

**Zeitschrift:** Bulletin technique de la Suisse romande  
**Band:** 89 (1963)  
**Heft:** 7

**Artikel:** Organisation générale des calculateurs arithmétiques  
**Autor:** Romano, M.A.  
**DOI:** <https://doi.org/10.5169/seals-66316>

#### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

#### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

#### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

**Download PDF:** 18.02.2026

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

## Conclusion

Les applications des mémoires électroniques à accès rapide se sont développées d'une manière considérable au cours de ces dernières années. Pour résoudre un problème donné l'ingénieur dispose maintenant de toute une gamme de solutions possibles : mémoires à tores, à films minces, à lignes à retard, etc. En fait le sujet est devenu tellement vaste que nous n'avons pu que l'effleurer au cours de cet article, en laissant de côté certains types de mémoires qui présentent un intérêt certain, comme les mémoires à lignes à retard ou les mémoires cryogéniques.

La tendance à l'accroissement de vitesse et de capacité s'affirmera sans doute au cours des prochaines années et il est probable que des solutions nouvelles permettront d'abaisser le prix des mémoires et d'apporter une plus grande souplesse dans leur utilisation. Dans cet ordre d'idée la réalisation de mémoires associatives dans lesquelles la recherche du contenu de la mémoire se ferait d'après ce contenu et non plus d'après des adresses codées fixes, apporterait un progrès certain.

## BIBLIOGRAPHIE

- J. A. RAJCHMAN : *A myriabit magnetic core matrix memory*. Proc. IRE, Vol. 41, p. 1407-1421, octobre 1953.
- J. A. RAJCHMAN and A. W. LO : *The Transfluxor*. Proc. IRE, p. 321-332, mars 1956.
- W. W. LAWRENCE JR. : *Recent developments in very high speed storage Technique*. Proc. EJCC, p. 101-103, décembre 1956.
- J. A. RAJCHMAN : *Ferrite apertured plate for random access memory*. Proc. IRE, Vol. 45, p. 325-334, mars 1957.
- R. L. GAVIN : *An analysis of the operation of a persistent supercurrent memory cell*. IBM J. Res. and Develop., Vol. 1, p. 304-308, octobre 1957.
- A. H. BOBECK : *A new storage element suitable for large sized memory arrays — The twistor*. Bell Syst. Tech. J., p. 1319-1340, novembre 1957.
- A. V. POHM and I. N. MITCHELL : *Magnetic film memories — A survey*. IRE Transactions on Electronic Computers, Vol. EC9, p. 308-314, septembre 1960.
- G. CONSTANTINE JR. : *New developments in load-sharing matrix switches*. IBM Journal, Vol. 4, p. 418-422, octobre 1960.
- J. A. RAJCHMAN : *Computer memories — A survey of the state of art*. Proc. IRE, p. 104-127, janvier 1961.

# ORGANISATION GÉNÉRALE DES CALCULATEURS ARITHMÉTIQUES<sup>1</sup>

par M. A. ROMANO<sup>2</sup>

## Introduction

Dans la *Critique de la raison pure*, Kant introduit une distinction entre jugements analytiques et jugements synthétiques, les premiers étant ceux dont la conclusion est implicitement contenue dans la définition des prémisses, les seconds ceux qui apportent un élément nouveau indémontrable. Il donne comme exemple l'égalité  $7 + 5 = 12$ , dont il a été facile de montrer par la suite qu'elle pouvait se démontrer par récurrence, c'est-à-dire par une forme évoluée de comptage, donc qu'elle était en fait un jugement analytique. Cet exemple est instructif, car, ainsi que nous le verrons, les calculateurs arithmétiques actuels sont des machines qui ne produisent que des jugements analytiques, c'est-à-dire en fait qui donnent des résultats contenus implicitement dans les données d'entrée, et nous allons suivre au cours du temps l'évolution des méthodes utilisées pour calculer la somme  $7 + 5 = 12$ .

## Arithmétique et comptage

La représentation la plus simple des nombres est par une série de bâtonnets, ainsi, l'égalité  $7 + 5 = 12$  peut aisément se vérifier dans ce système de numération, où elle devient  $||||| + |||| = ||||| + ||||$ .

Mais il est manifeste que ce système a des possibilités vite limitées lorsqu'on passe à des nombres tant soit peu importants ; d'où la représentation des nombres dans un système décimal (à base 10) sous la forme

$$N = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 \cdot 10 + a_0$$

où les  $a_i$  sont des entrées de 0 à 9 ; ainsi, dans ce système bien connu,

$$12 = 1 \cdot 10 + 2.$$

Mais un système de représentation ne vaut que par les modes opératoires qu'il permet pour effectuer les opérations arithmétiques. Le procédé le plus simple est le comptage, montré dans les exemples ci-dessus ; il a, ainsi qu'en le verra, été utilisé dans les calculateurs automatiques ; mais il est long et fastidieux, d'où l'introduction de méthodes plus rapides telles que la table de Pythagore pour l'addition ou la multiplication. Certains calculateurs (IBM 1620) utilisent ce dernier procédé pour effectuer ces opérations.

## Premiers essais de mécanisation du calcul

Parallèlement aux efforts pour simplifier le calcul (tels que abaque, tables de logarithmes, etc.), des essais de mécanisation furent entrepris dès le XVII<sup>e</sup> siècle. Pascal inventa la machine à additionner, qui calculait pour l'addition les procédés du calcul manuel ; elle était composée (fig. 1) d'une série de roues tournant par dixième de tour ; chaque roue représentait un chiffre du nombre, et sa position indiquait si ce chiffre était 0, 1, ..., 9. De plus, un système de report faisait que lorsqu'une roue passait de 9 à 0, elle faisait avancer la roue suivante d'un cran (notons que le report peut se propager de plusieurs positions vers la gauche — exemple :  $999 + 1$  — ce phénomène sera très important dans les calculateurs électroniques). Chaque roue contenait à l'intérieur une deuxième roue qui pouvait lui être soit solidaire, soit séparée ; pour effectuer  $7 + 5$ , on commençait par afficher 7 sur la

<sup>1</sup> Conférence prononcée à Genève le 4 décembre 1961, devant la section genevoise de l'Association suisse pour l'automatique (ASSPA).

<sup>2</sup> Ancien élève de l'Ecole polytechnique et ingénieur ESE, au laboratoire IBM France.

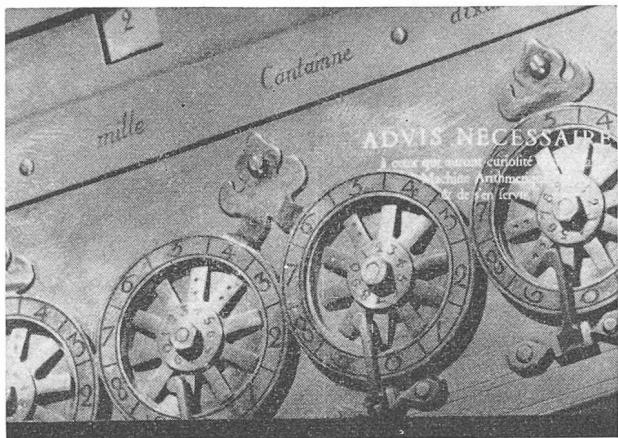


Fig. 1. — Machine à additionner de Pascal.

roue intérieure en entraînant la roue extérieure ; puis on ramenait la roue intérieure à 0 toute seule et on recommençait avec 5. On pouvait de manière analogue effectuer la soustraction ; lorsque le résultat était négatif, il était, de par la nature même du principe utilisé, représenté par son complément à la puissance de 10 correspondante.

D'autres essais du même type furent entrepris ; ainsi Leibniz avec le multiplicateur qui transposait de la même manière les règles de la multiplication décimale.

#### Apparition de la notion de programme

L'opération  $7 + 5 = 12$  ne représente qu'une action unique à effectuer ; dans des calculs plus évolués, on verra apparaître des séquences d'opérations de ce type, à effectuer dans un ordre bien déterminé ; une telle séquence constitue un programme, et la première machine utilisant cette notion pour des calculs est la machine de Babbage ; cependant l'idée de programme se trouvait contenue auparavant dans le célèbre métier à tisser de Jacquard ; pour fabriquer un dessin déterminé, il fallait à chaque passage de la tête soulever certains des crochets tenant les fils du métier ; à cette fin l'indication de ceux des crochets à soulever était contenue dans un jeu de cartes perforées, une carte pour chaque passage ; ces cartes, ficelées ensemble de façon à former une chaîne continue, tournent sur un axe à section carrée, et actionnent ou non, en fonction de leurs perforations, les crochets par l'intermédiaire d'un ingénieux système mécanique.

Babbage utilisa ce principe pour concevoir sa machine analytique (qu'il ne put réaliser), où les nombres étaient représentés par des roues, où se trouvait un organe qui devait préfigurer l'organe arithmétique des calculateurs modernes, et où la commande de la séquence des opérations à effectuer se faisait par deux mécanismes de Jacquard, l'un qui commandait l'unité arithmétique pour faire une addition, soustraction, etc., l'autre qui commandait les transferts de la « mémoire » et vers elle. Au cas où un résultat partiel devenait négatif, un dispositif était prévu pour faire sauter les chaînes d'un certain nombre de positions, de façon à pouvoir exécuter une séquence différente d'opérations.

Des machines plus modernes continuèrent selon cette voie, mais en devenant de plus en plus grandes et com-

plexes. Dans le Mark I, développé en 1944 par le professeur Aiken de Harvard, avec le concours de l'IBM, les données étaient sur des registres composés de roues, et le programme sur une bande de ruban perforé. Cette machine était donc toujours du type à « programme extérieur ».

#### Calculateurs à tableau

Une autre catégorie de machines est celle où la séquence des opérations du programme est indiquée par un tableau de connexions. Dans une machine de ce type, des registres contiennent les nombres à manipuler ; il y a deux « canaux » de transfert de nombres : l'un qui communique avec les entrées de tous les registres, l'autre avec leurs sorties. Le contrôle se fait par un tableau, avec une « horloge » qui marche de la façon suivante : un compteur en anneau avance régulièrement d'une position sous l'effet d'un multivibrateur par exemple (en fait les choses sont moins simples, c'est un deuxième anneau plus rapide qui avance sous l'effet d'un multivibrateur et à chaque tour fait avancer le premier anneau ; c'est le deuxième anneau qui sert d'« horloge » à la machine en découplant le temps de chaque opération, mais sa compréhension n'est pas nécessaire pour ce qui suit) ; cet anneau appelé anneau de programme envoie à chaque pas une impulsion qui se retrouve au tableau ; par câblage on peut faire commander par cette impulsion une addition, un transfert, etc., selon la séquence des opérations désirées ; le tableau de connexions ci-joint (fig. 2), qui est celui d'une machine imaginaire à tableau, représente le câblage qu'il y aurait à faire pour effectuer l'opération  $7 + 5 = 12$ .

La séquence des opérations est la suivante :

- au premier pas de programme, on commande une remise à zéro ;
- au deuxième pas de programme, on commande la lecture de la première carte dans le registre A (une commande lire et une commande registre A) ;
- au troisième pas, on commande la lecture de la carte suivante dans le registre B (une commande lire et une commande registre B).

Etc.

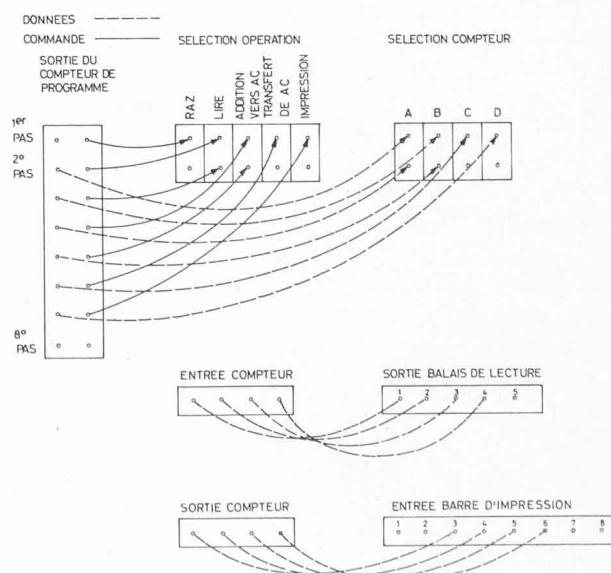


Fig. 2. — Tableau de connexions.

Notons que tous ces « pas » de programme sont de même longueur, fixée par la rotation d'éléments mécaniques (100 ms dans un cas typique).

## Calculateurs à programme enregistré

L'étape suivante a consisté à reconnaître qu'un calculateur automatique devait d'une part emmagasiner des nombres à traiter, d'autre part la séquence des opérations nécessaires à cette fin, et que ces deux nécessités n'étaient pas si différentes qu'elles pouvaient le paraître au départ. En effet, les ordres pourraient être représentés de façon codée, par exemple 1 pour l'addition, 2 pour la soustraction, 3 pour la multiplication, 4 pour la division. Dans l'exemple traité, pour indiquer qu'il faut ajouter le contenu du registre 15 de données à celui du registre 17 et placer le résultat dans le registre 18, par exemple, on pourrait représenter cela par 1, 15, 17, 18.

Les ordres pouvant donc être représentés comme des nombres, il devenait naturel d'emmagasiner la séquence des ordres nécessaires dans la même « mémoire » que les données ; cela fait que l'on peut non seulement les modifier à volonté manuellement comme un tableau de connexions, mais surtout agencer le programme de telle sorte qu'il se modifie lui-même en fonction de certains résultats partiels ; par exemple si dans l'exemple précédent, on veut qu'en fonction du signe d'un résultat antérieur, le résultat de l'addition aille dans le registre 23 au lieu du registre 18, il suffit d'ajouter le chiffre 5 au « mot » constituant l'ordre. Ce bond en avant prodigieux de conception a été rendu possible pratiquement par l'apparition de mémoires à prix réduit, et est donc lié à la généralisation de l'électronique dans la technique des calculateurs, que nous allons voir, avec les conséquences (numérotation binaire) qu'elle implique, avant de décrire la structure d'un calculateur à programme enregistré.

## Avènement de l'électronique et arithmétique binaire

Pendant et depuis la dernière guerre, sous la pression des besoins en calcul scientifique, des machines de plus en plus importantes virent le jour. Ces machines furent d'abord électromécaniques (à relais et compteurs à roues), puis électroniques (à tubes puis à transistors). Les circuits utilisés dans ces machines sont caractérisés par le fait qu'ils sont bistables, d'où la représentation des nombres par des éléments binaires ; ceci peut se faire :

1) Soit par une représentation binaire pure, en remplaçant la base 10 par une base 2 et en écrivant :

$$N = a_n 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_1 \cdot 2 + a_0$$

où les  $a_i$  ne peuvent prendre ici que les valeurs 0 ou 1.

Ainsi les nombres 7, 5 et 12 sont représentés par :

7	411
5	101
12	1100

Il est nécessaire, en plus de la représentation, de donner des modes opératoires permettant d'effectuer les opérations arithmétiques, afin d'éviter les longueurs inhérentes au procédé par comptage. Ainsi pour l'addition de 2 chiffres, on a une table extrêmement simple,

qui donne la somme partielle et le report à la position suivante :

	0	1
0	0	1
1	1	0

## Somme

	0	1
0	0	0
1	0	1

## Report

Par ce procédé on peut donc effectuer en plusieurs temps la somme  $7 + 5 = 12$ . Il faut d'abord convertir les nombres 7 et 5 en équivalents binaires (cf. ci-dessus), puis faire :

$$7 + 5 = 111 + 101 \rightarrow 010 \quad \text{et} \quad \begin{array}{r} 101 \\ \text{report 1} \end{array}$$

sommes partielles 1

en additionnant de même les sommes partielles et le report décalé vers la gauche d'une position :

- somme

partielle 1 + report 1 décalé  $\rightarrow$  1000 et 001  
 somme partielle 2 report 2

Enfin, en recommençant une fois de plus, on trouve : somme partielle 2 + report 2 décalé = 1100.

C'est bien le nombre 12.

On voit sur cet exemple que les opérations effectuées de façon aussi élémentaires sont longues du fait de la propagation des retenues ; d'où l'emploi de circuits plus évolués que ceux qui forment les opérations précédentes, et dont la compréhension nécessiterait des connaissances d'Algèbre de Boole ; qu'il nous suffise de dire que le procédé pour l'addition précédente consisterait à emmagasiner la somme partielle 1 et le report 1 dans deux registres et à les combiner ensuite selon certaines règles qui évitent dans une certaine mesure cette propagation très lente (et qui le serait encore beaucoup plus pour des nombres plus grands).

2) *L'autre méthode possible de représentation consiste à garder la base décimale, mais à représenter chaque chiffre par une combinaison d'éléments binaires, au nombre de 4 ou 5. Ainsi on pourrait représenter par exemple avec 4 éléments binaires les chiffres décimaux de 0 à 9 par leur représentation binaire normale, mais interdire les combinaisons correspondant aux nombres de 10 à 15 et agencer les reports en conséquence ; dans ce type de représentation, les nombres 7 et 5, qui n'ont qu'un chiffre seront représentés de la même manière qu'en binaire, mais le nombre 12 sera représenté par :*

0004 0010 au lieu de 0000 1100  
1 2

d'où une interprétation humaine plus facile.

Cette représentation décimale codée binaire aboutit à des circuits plus compliqués pour effectuer les opérations arithmétiques, mais a l'avantage d'éviter la traduction décimale-binaire et binaire-décimale à l'entrée et à la sortie ; elle sera donc utilisée dans les applications de type commercial où il y a beaucoup d'opérations d'entrée-sortie, et peu de calculs, alors que dans les applications scientifiques où la proportion est inversée, on utilise des machines du type binaire.

## Structure d'un calculateur à programme enregistré classique

On prendra comme exemple la calculatrice IBM 701.

Il est d'usage de représenter cette structure par la décomposition classique en cinq parties (fig. 3).

- L'entrée, qui permet l'introduction du programme et des données dans la mémoire ; ce peut être par exemple un lecteur de cartes perforées, où les perforations créent des contacts électriques entre balais et plaque conductrice ; ces signaux, convenablement interprétés, sont envoyés à l'organe central.
- La sortie, qui effectue les opérations inverses ; ce peut être par exemple une imprimante, qui convertit les signaux électriques venant de l'organe central en sélection de position d'une roue d'impression contenant l'alphabet.
- La mémoire, composée de registres, où, comme son nom l'indique, sont stockés les données et le programme.
- La partie arithmétique, qui effectue les opérations arithmétiques sur les données emmagasinées dans la mémoire ; ces opérations peuvent s'effectuer en série ou en parallèle (par mot).
- L'unité de contrôle, qui coordonne l'activité des quatre unités précédentes.

Voyons comment un calculateur imaginaire ayant cette structure effectuerait l'addition  $7 + 5 = 12$  ; supposons que des données 7, et 5, soient perforées dans deux cartes, et qu'on veuille imprimer le résultat de l'addition ; supposons encore que le programme nécessaire ait déjà été introduit dans la machine. La séquence des ordres de la machine sera la suivante :

### 1. Lire carte dans mémoire A

Cet ordre déclenche la lecture de la première carte contenue dans le lecteur (qui contient, rappelons-le, le nombre 7) et l'introduction de son contenu (7) dans la mémoire dont le numéro est A.

### 2. Lire carte dans mémoire B

Cet ordre déclenche une nouvelle lecture de carte, mais son contenu (5) va dans la mémoire B.

données : \_\_\_\_\_

commande : \_\_\_\_\_

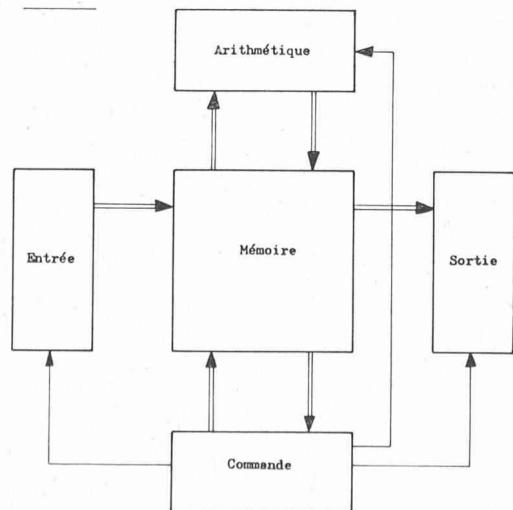


Fig. 3. — Organisation d'un calculateur classique.

(Remarquer qu'il s'ordonne autour de la mémoire : une opération à la fois.)

### 3. Remise à zéro de l'accumulateur et addition A

Cet ordre provoque l'inscription de l'information contenue dans la mémoire A, dans un registre spécial contenu dans l'organe arithmétique et appelé accumulateur ; ce registre stocke les résultats partiels des opérations arithmétiques.

### 4. Addition B

Cet ordre provoque l'addition de l'information contenue dans la mémoire B au contenu antérieur de l'accumulateur.

### 5. Transfert de l'accumulateur vers C

Cet ordre provoque le transfert du contenu de l'accumulateur (c'est-à-dire du nombre  $7 + 5 = 12$ ) dans la mémoire C.

### 6. Imprimer C

Cet ordre provoque l'impression du contenu de la mémoire C.

Il est facile de voir, dans cette structure séquentielle, que les opérations 3, 4 et 5 ne mettent pas en jeu les organes d'entrée et de sortie, donc ne mettent en jeu que des parties électroniques (mémoire, unité arithmétique et unité de programme) ; elles se déroulent donc très rapidement (à des vitesses allant de quelques microsecondes à quelques millisecondes selon les calculateurs). Par contre les opérations 1, 2 et 6 mettent en jeu des unités mécaniques d'entrée-sortie, et nécessitent plusieurs dizaines, voire plusieurs centaines de millisecondes selon les cas pour être effectuées. Pendant tout ce temps, les unités électroniques ne travaillent pas. Pour éviter cet inconvénient, les calculateurs modernes de grande puissance ont une structure plus complexe, avec des organes d'entrée et de sortie pourvus de registres (appelés mémoires-tampons) et ayant une certaine autonomie, de façon que les parties électroniques rapides puissent initier le départ des opérations mécaniques d'entrée-sortie, puis se remettre à calculer pendant que ces opérations se déroulent.

Notons avant d'en terminer avec les anciens calculateurs à programme enregistré une caractéristique qu'ils ont en commun avec les calculateurs plus modernes et qui découle de la banalisation de la mémoire ; c'est que, du fait que tant les instructions que les données sont dans la même mémoire (banalisation de la mémoire), chaque instruction s'effectue en deux temps :

— *cycle instruction*, pendant lequel l'instruction suivante à effectuer (connue grâce à un compteur de programme qui avance d'une unité à chaque instruction et rend compte de l'état d'avancement du programme) est prélevée dans la mémoire à l'adresse indiquée par le compteur de programme ; cette instruction remplit un mot dont la première moitié représente l'opération à effectuer, la deuxième partie l'adresse où la donnée à traiter se trouve,

— *cycle opération*, pendant lequel le mot opération est décodé, sa première partie positionne les circuits de commande de l'organe arithmétique, et la deuxième partie les circuits de sélection d'adresse de la mémoire ; le mot à traiter est alors introduit de la mémoire dans l'organe arithmétique (ou l'inverse selon l'opération à effectuer) et l'opération arithmétique, s'il y en a une, est effectuée.

Puis, le compteur de programme avance d'une unité, et ainsi de suite.

Mentionnons que cette séquence peut être plus compliquée, en particulier dans les machines utilisant la technique de la microprogrammation, où une instruction est en fait traduite par les circuits de contrôle en une séquence d'opérations à effectuer sous le contrôle d'une mémoire spécialisée.

## Organisation d'un ordinateur moderne : l'IBM 7090

A titre d'exemple de la structure d'un calculateur moderne et de la manière dont il effectuerait l'opération  $7 + 5 = 12$ , nous allons décrire rapidement le 7090. C'est un grand calculateur scientifique, dont l'organisation schématisée (fig. 4) est la suivante : il y a trois « parties » : la mémoire, la partie arithmétique et de programme, et le canal d'entrée-sortie, qui communiquent par un multiplexeur, qui est un organe purement passif permettant seulement d'assurer qu'à un instant donné deux et deux seulement des trois parties précédentes communiquent entre elles ; en fait le

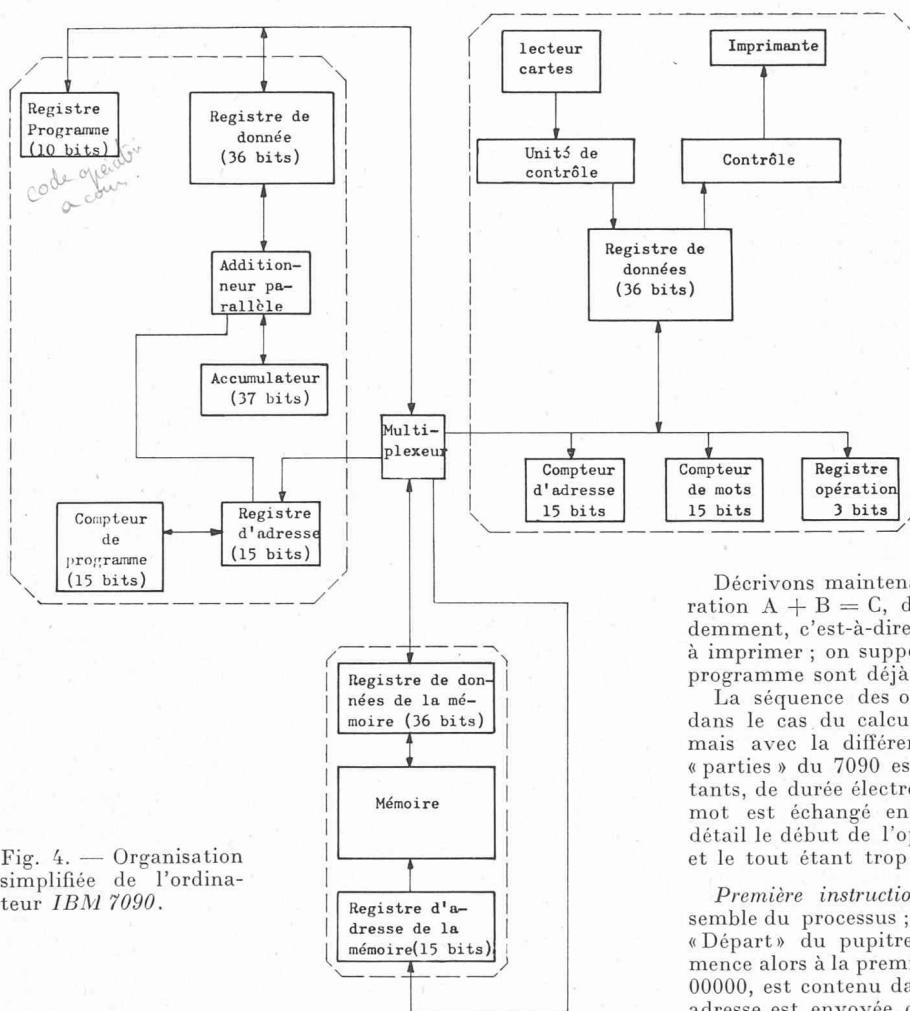


Fig. 4. — Organisation simplifiée de l'ordinateur IBM 7090.

7090 est « extensible » et peut avoir en plus de la mémoire et la partie arithmétique-programme, plusieurs (jusqu'à 6) canaux d'entrée-sortie ; chacune de ces « parties » a ici une autonomie beaucoup plus grande que dans le calculateur classique décrit précédemment et a en particulier un registre qui contient le mot qui vient d'entrer dans cette partie ou qui va en sortir, et qu'on appellera registre de données.

a) *La mémoire.* C'est une mémoire à ferrites, comportant 32 768 ( $= 2^{15}$ ) mots de 36 bits. La durée du cycle mémoire (temps nécessaire pour écrire un mot dans la mémoire ou de le lire) est de 2,18  $\mu$ s ; cette durée est égale au temps de mot du fait que le mot entre ou sort en parallèle de la mémoire. La mémoire a un registre d'un mot contenant le mot à écrire ou qui vient d'y être lu, et un registre de 15 bits contenant l'adresse de ce mot.

b) *La partie arithmétique et de programme, ou partie de traitement.* La machine a près de 200 « codes opérations » permettant d'effectuer des opérations arithmétiques, des opérations logiques (intersection, etc.), des tests, des sauts de programme, etc. L'arithmétique se fait en parallèle, avec des registres (tels que l'accumulateur et le multiplicateur-quotient) contenant les opérations et les résultats partiels ; les nombres contenus dans ces registres sont manipulés par un addeur sous le contrôle du programme. Cette partie contient donc aussi un registre programme contenant le code de l'opération en cours, un registre de donnée contenant le mot qui vient d'y entrer ou qui va en sortir, un compteur de programme qui compte les instructions et tient l'état de l'avancement du programme (il peut sauter lors de branchements), et un registre d'adresse qui contient l'adresse de mémoire relative à l'opération en cours. A titre d'exemple mentionnons que le temps nécessaire pour effectuer une opération d'addition est de 2 cycles mémoire, soit 4,4  $\mu$ s.

c) *Le canal d'entrée-sortie.* Ce canal peut lire ou écrire en une unité d'entrée ou de sortie ; il peut être relié à 10 dérouleuses de bandes magnétiques, un lecteur de cartes, une imprimante, etc. Il peut à un instant donné sélectionner une de ces unités et y lire ou y écrire, mais ne peut pas lire ou écrire sur deux unités simultanément. Il contient à cet effet un registre d'un mot contenant la donnée à lire ou écrire, un registre d'adresse contenant l'adresse dans la mémoire d'où vient ce mot (ou bien où il doit aller), un compteur de mots qui reçoit le total du nombre de mots à entrer ou à sortir lors d'une opération d'entrée ou de sortie et vérifie ainsi par comptage que l'opération est bien terminée, et un registre opération qui contient la description de l'opération à effectuer.

Décrivons maintenant la façon dont se déroulerait l'opération  $A + B = C$ , dans les mêmes conditions que précédemment, c'est-à-dire A et B étant perforés sur carte, et C à imprimer ; on supposera de même que les instructions du programme sont déjà dans la machine.

La séquence des opérations à effectuer est la même que dans le cas du calculateur classique décrit précédemment, mais avec la différence qu'à chaque étape, seule une des « parties » du 7090 est impliquée, sauf à de très brefs instants, de durée électronique et non plus mécanique, où un mot est échangé entre deux parties. Nous décrirons en détail le début de l'opération, la suite étant du même type et le tout étant trop long à décrire.

*Première instruction :* Il faut au départ démarer l'ensemble du processus ; à cette fin, on appuie sur le bouton « Départ » du pupitre du calculateur, le programme commence alors à la première instruction dont le numéro d'ordre 00000, est contenu dans le compteur de programme ; cette adresse est envoyée dans le registre adresse de la partie de traitement et dans celui de la partie mémoire. A l'adresse 00000 se trouve la première instruction à exécuter, qui va dans le registre de données de la mémoire, puis dans celui de la partie de traitement. La partie opération va dans le registre programme de la partie de traitement.

Cette première instruction a pour but de sélectionner le lecteur désiré sur le canal d'entrée-sortie ; la partie opération indique donc qu'une instruction de sélection de lecture est à effectuer, et le registre de donnée de la partie de traitement contient l'identification du canal et du lecteur choisi. En réponse, si le lecteur est prêt à fonctionner, le canal rend compte de la sélection.

Pendant ce temps le compteur de programme, ayant rempli son rôle, s'est positionné sur 00001 pour l'instruction suivante.

*Deuxième instruction :* Le compteur de programme étant à 00001, la partie traitement demande un mot de la mémoire, et le contenu de la mémoire 00001 va, de la même manière que précédemment, au registre de donnée de cette partie et à son registre programme.

L'instruction contient dans sa partie opération l'indication que les registres du canal doivent être remis à zéro et qu'un ordre doit y être envoyé. La partie de traitement a l'adresse de cet ordre dans son registre de donnée, cette adresse va donc aux registres d'adresse de la partie traitement et de la mémoire ; au même moment le registre programme est décodé, ce qui exécute l'instruction et envoie le mot à l'adresse indiquée vers le canal ; le canal reçoit donc la partie opération du mot, et la partie de comptage va dans le compteur de mot et l'adresse dans la partie adresse.

Etc., etc.

Ce fonctionnement est donc assez compliqué, mais il faut noter que :

— A chaque moment, tout l'ordinateur n'est pas mis en jeu ; on peut donc imbriquer les opérations consécutives d'un programme de telle sorte que pendant les

opérations lentes telles qu'entrée-sortie se déroulent en parallèle des calculs, le canal d'entrée-sortie et la partie traitement se partageant alternativement l'accès à la mémoire.

— Du point de vue de l'utilisateur, seule la séquence des instructions est à connaître, mais non le fonctionnement détaillé décrit précédemment ; ainsi le programme, pour effectuer  $A + B = C$  comprendrait 12 instructions dans le « langage » de la machine, c'est-à-dire que le programmeur aurait à écrire une séquence de 12 mots de 36 bits, chaque mot décrivant une des instructions (opération à effectuer et adresse mise en jeu).

Cet effort de programmation a d'ailleurs été simplifié par l'introduction de langages symboliques : le programmeur écrit une séquence très courte d'instructions globales, qui sont traduites par un programme spécial en instructions détaillées en « langage machine ». Dans le cas du 7090 par exemple, le programme d'interprétation FORTRAN a été mis au point ; de cette manière, le programmeur n'a plus qu'à écrire un nombre restreint d'instructions sous une forme se rapprochant du langage usuel ; ainsi dans l'exemple précédent, le programme serait écrit de la façon suivante :

```
LIRE 1, A/B  
C = A + B  
IMPRIMER 1, C  
1 MODÈLE (I 4) (cette instruction donne le  
format des nombres entrés et sortis).
```

#### **Evolution future des calculateurs électroniques**

En extrapolant l'évolution actuelle de ces machines, on peut déceler les grandes lignes de l'évolution future des calculateurs ; ce sont :

1. La tendance à l'augmentation de puissance des calculateurs continuera ; ceci est lié à l'accroissement des besoins des marchés traditionnels des calculateurs (calcul scientifique et traitement de la gestion) ainsi qu'à leur introduction dans de nouveaux domaines (tels que le contrôle des processus industriels). Cette augmentation de puissance se traduira par un accroissement tant de la capacité de mémoire que de la vitesse des circuits logiques et de calcul. Les calculateurs de grosses dimensions auront toujours sur les petits calculateurs l'avantage d'un meilleur rapport performance/prix, et ce facteur contribuera à cette évolution, la situation étant voisine de celle des centrales électriques.

2. Un autre trait commun avec l'évolution de l'électrotechnique est que ces calculateurs seront de plus en plus interconnectés, soit sur place, soit plus souvent à distance ; on assistera ainsi progressivement à la mise en place de véritables réseaux de calculateurs. Cette évolution apparaît encore un peu lente, mais elle commence néanmoins à se faire jour. De même les unités d'entrée-sortie d'un calculateur pourront parfois se trouver à grande distance de ce calculateur, comme il apparaît dans les systèmes de réservation de places d'avion.

3. Un autre phénomène est l'accroissement de la complexité de la structure des calculateurs. Cette structure est influencée par deux facteurs :

##### *a) Le coût relatif logique/mémoire*

Dans les machines à relais un élément de mémoire coûtait aussi cher ou même plus qu'un élément logique ; elles avaient donc une structure assez parallèle, avec

plusieurs opérations arithmétiques se déroulant simultanément. Par contre avec l'avènement de l'électronique et des technologies telles que les mémoires à tambour puis à ferrite, le coût d'un élément de mémoire devenait bien plus faible que celui d'un élément logique (il est très difficile de donner des ordres de grandeurs, ces chiffres variant beaucoup selon les technologies et les vitesses, mais on peut dire très grossièrement que le coût d'un élément de mémoire est de l'ordre du franc suisse, celui d'un élément logique une trentaine de fois plus grand). Cela a conduit naturellement à des machines de structure plus séquentielle, où l'organe de traitement ne fait qu'une chose à la fois et remet chaque fois en mémoire les résultats de ses opérations. L'expression la plus idéale de cette tendance, jamais réalisée pratiquement car impraticable, est la machine de Turing.

On peut estimer que dans les dix années à venir, ces coûts diminueront dans des proportions très appréciables (de plus de dix fois) ; si leur rapport reste constant, ce facteur ne contribuera pas à l'évolution des calculateurs, mais si des technologies nouvelles (telles que les cryotrons) rendaient les éléments logiques aussi bon marché que ceux de mémoire, cela contribuerait à rendre les calculateurs plus parallèles qu'ils ne le sont aujourd'hui.

##### *b) L'importance des opérations d'entrée-sortie*

Ce facteur tend à augmenter de plus en plus au fur et à mesure des progrès de l'automatisation. Au début, les calculateurs ne furent conçus que pour automatiser les opérations de calcul (d'où leur nom), mais de plus en plus on tend à vouloir automatiser tout l'ensemble d'un processus industriel ou de gestion ; dans ces conditions, le rôle principal du calculateur devient bien plus de contrôler la circulation de l'information (à la manière d'un standard téléphonique) que de calculer au sens propre du terme ; ce facteur contribuera à accentuer la complexité des calculateurs par la multiplication des organes d'entrée-sortie et la nécessité de les coordonner. Notons en passant que l'IBM a choisi le mot « ordinateur » pour ses calculateurs en partie à cause de ce fait.

4. Le coût des calculateurs baissant, le coût du travail humain nécessaire, soit pour les entretenir en état de marche, soit pour rédiger les programmes nécessaires, devra de plus en plus être réduit pour ne pas constituer le goulet d'étranglement de l'utilisation des calculateurs. On tendra donc (même si d'un point de vue purement technique on aboutit à un fonctionnement moins efficace de la sorte) à utiliser de plus en plus des langages symboliques (FORTRAN, ALGOL, COBOL, etc.) et des détections automatiques de pannes. Le calculateur évoluant vers quelque chose de plus en plus « intelligent » sera capable de détecter ses propres pannes, d'interpréter un code voisin du langage des mathématiques, voire même des graphismes manuels. Soulignons cependant qu'il s'agit toujours de machines déterministes, qui ne donnent ces résultats que parce qu'un cerveau humain leur aura préalablement donné des instructions leur permettant de calculer, de s'adapter aux circonstances, etc. Elles ne « pensent » donc pas au sens propre du terme, puisqu'elles ne sont capables d'émettre que des jugements analytiques par un ensemble d'algorithmes qu'on leur aura « soufflés ». Les « machines qui pensent » relèvent plus de la science-fiction que de la technique, et il est peu probable d'en voir se substituer complètement au cerveau humain dans un avenir prévisible.