



HAL
open science

Les opérations logiques dans les calculatrices digitales

Roland Rouxel

► **To cite this version:**

Roland Rouxel. Les opérations logiques dans les calculatrices digitales. Autre [cs.OH]. Université Joseph-Fourier - Grenoble I, 1956. Français. NNT: . tel-00165366v2

HAL Id: tel-00165366

<https://theses.hal.science/tel-00165366v2>

Submitted on 5 May 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

T H E S E S
présentées à la FACULTE DES SCIENCES
de l'UNIVERSITE de GRENOBLE
pour obtenir
le grade d' INGENIEUR-DOCTEUR
par

Roland ROUXEL
Ingénieur I.E.G.

--:--:

PREMIERE THESE :

"Contribution à l'étude
des phénomènes transitoires dans les enroulements".

DEUXIEME THESE :

Propositions données par la Faculté :
"Les opérations logiques dans les calculatrices digitales".

soutenues le 7 JUILLET 1956,
devant la Commission d'Examen :

MM. NEEL L.	Président,
KUNTZMANN J.	Rapporteur,
FALLOT M	} Examineurs.
PILLET E.	

LES OPERATIONS LOGIQUES

DANS LES

CALCULATRICES DIGITALES

-:-:-

S O M M A I R E

--:--:--

INTRODUCTION,	page 1
I - STRUCTURE GENERALE D'UNE CALCULATRICE UNIVERSELLE,	page 2
II - LES OPERATIONS LOGIQUES,	page 4
A) Réalisation des aiguillages.	
B) Modification des instructions.	
III - QUELQUES EXEMPLES DE REALISATION,	page 8
1°) Machines à fonctionnement non séquentiel, .	page 8
A) Elliott 402, 1+1 adresse, mots banalisés.	
B) Deuce, 1+1 ou 2+1 adresses, mots non banalisés.	
C) Swac, 3+1 adresses, mots banalisés.	
2°) Machines à fonctionnement séquentiel,	page 18
A) Gamma, 1 adresse, mots non banalisés.	
B) M.T.C., 1 adresse, mots banalisés.	
C) Cab, 1 adresse, mots banalisés.	
D) Elecom 125, 2 adresses, mots banalisés.	
E) Seac, 3 ou 3+1 adresses, mots banalisés.	
IV - CONCLUSIONS,	page 32

--:--:--

I N T R O D U C T I O N

--:--:--:--

Tout problème mathématique peut être décomposé en une suite d'opérations élémentaires, arithmétiques ou logiques, et on appelle calculatrice digitale universelle une machine capable d'effectuer une telle suite d'opérations. Une instruction est formée d'une ou plusieurs opérations élémentaires et l'ensemble des instructions constitue le programme.

La machine sera dite automatique si elle est capable d'enchaîner les instructions : le programme peut être entièrement exécuté sans intervention humaine.

Nous rappellerons brièvement la structure générale d'une calculatrice automatique universelle, puis nous étudierons plus particulièrement l'ensemble des opérations logiques.

--:--:--:--:--:--

- I -

STRUCTURE GENERALE D ' UNE
CALCULATRICE UNIVERSELLE.

--:--:--

D'une manière très schématique, toute calculatrice universelle peut être décomposée en cinq parties :

- l'organe d'entrée,
- l'organe de sortie,
- l'organe de mémoire,
- l'organe de calcul,
- l'organe de commande.

Les organes d'entrée et de sortie assurent la communication de la machine avec l'extérieur.

Les informations, nombres ou instructions, sont enregistrées dans un ou plusieurs organes de mémoires, capables de les restituer dès que le calcul l'exige.

Chaque nombre ou instruction est identifié par son adresse, numéro d'ordre désignant la cellule de mémoire contenant ce nombre ou cette instruction.

Les organes de calcul sont chargés d'exécuter les opérations arithmétiques ou logiques caractérisées dans l'instruction par un symbole appelé lettre de fonction.

La coordination des différents organes est réalisée par un ou plusieurs circuits de commande qui interprètent les instructions

et règlent les transferts d'information entre ces organes.

L'exécution d'un programme implique que l'on fournisse à la machine l'adresse de la première instruction ; elle est ensuite capable de trouver l'adresse des instructions suivantes. La séquence se poursuit jusqu'à la dernière instruction qui sera un ordre d'arrêt de la machine.

--:--:--

- II -

OPERATIONS LOGIQUES

--:--:--

Elles concernent :

- la réalisation des aiguillages,
- la modification des instructions.

A) REALISATION DES AIGUILLAGES. -

Le caractère automatique d'une calculatrice exige qu'elle soit capable de choisir entre plusieurs éventualités prévues à l'avance. Ce choix sera en général basé sur une opération de test. Cette opération peut revêtir plusieurs formes suivant le mode de fonctionnement de la machine. On distingue :

a) Les machines à fonctionnement non séquentiel.

Chaque instruction contient explicitement l'adresse de l'instruction suivante.

L'opération de test sera de la forme :

- si le test est vérifié, passer à l'instruction d'adresse n ;
- si le test n'est pas vérifié, passer à l'instruction d'adresse n'.

b) Les machines à fonctionnement séquentiel.

L'adresse de l'instruction suivante n'est pas explicitée : les adresses des instructions successives sont rangées dans l'ordre naturel ; un compteur ordinal enregistre à chaque instant l'adresse de l'instruction suivante.

Une rupture de séquence peut être :

- conditionnelle :

- si le test est vérifié, sauter à l'instruction d'adresse n
- si le test n'est pas vérifié, continuer la séquence normale des instructions.

- inconditionnelle :

sauter dans tous les cas à l'instruction d'adresse n.

La rupture de séquence inconditionnelle peut être réalisée soit par une instruction spéciale, soit par un test toujours vérifié.

Un test ne permet de choisir qu'entre deux alternatives. Un aiguillage comportant un plus grand nombre de voies peut être résolu :

- soit par décomposition en une succession d'aiguillages simples,
- soit par un ordre de transfert combiné avec des modifications d'adresses de certaines instructions. Nous en donnerons un exemple au sujet de la calculatrice "CAB".

PRINCIPAUX TESTS UTILISES. -

Les tests utilisés peuvent porter :

- sur le signe d'un nombre, en général contenu dans l'accumulateur.
Distinguer si ce nombre est : ≥ 0 ou < 0
 ≤ 0 ou > 0
 $\neq 0$ ou $= 0$

On peut comparer deux nombres (ou leurs valeurs absolues) en testant le signe de leur différence.

- sur la comparaison de deux nombres (ou de leurs valeurs absolues) : il s'agit d'instructions spéciales rencontrées dans les machines à plusieurs adresses.

- sur des opérations de contrôle, telles que le dépassement de capacité.

Le résultat d'un test est, en général, utilisable immédiatement.

On pourra réaliser un aiguillage à distance en conservant en mémoire le résultat du calcul sur lequel doit porter le test.

B) MODIFICATION DES INSTRUCTIONS. -

Une instruction est constituée de la manière suivante :

- une lettre de fonction, spécifiant l'opération arithmétique ou logique à effectuer ;
- une ou plusieurs adresses :
 - l'adresse du nombre sur lequel porte l'opération (éventuellement adresses de plusieurs nombres),
 - l'adresse où doit être placé le résultat (éventuellement),
 - l'adresse de la prochaine instruction (éventuellement).

Les modifications d'instructions peuvent porter :

- sur l'ensemble de l'instruction,
- sur une partie de l'instruction.

Les modifications d'adresses sont les plus importantes.

L'enchaînement des instructions constituant un programme est en effet rarement linéaire (absence de boucles), et certaines instructions seront exécutées un grand nombre de fois : la même opération porte sur plusieurs nombres et il est nécessaire de pouvoir modifier la partie adresse de l'instruction. On réalise ainsi d'importantes économies de positions de mémoire.

Il peut être également nécessaire de modifier, suivant les résultats d'un calcul, l'opération elle-même, ou la totalité de l'instruction.

Réalisation des modifications d'instructions.

D'une manière générale, une instruction sera modifiée, en totalité ou en partie, par substitution, ou addition d'une quantité convenable.

Deux grandes catégories de machines sont à distinguer :

- Les machines à mots non banalisés. -

Bien que représentés à l'intérieur de la machine par des grandeurs de même nature, les nombres et les instructions sont nettement séparés. En particulier, les instructions ne pénètrent jamais dans l'organe de calcul, et des instructions spéciales seront nécessaires pour réaliser des modifications d'instructions.

- Les machines à mots banalisés. -

Rien ne distingue à l'intérieur de la machine, un nombre d'une instruction. Les instructions peuvent être traitées dans l'organe de calcul et aucune instruction spéciale ne sera nécessaire pour modifier une instruction.

Ces modifications peuvent être automatiques dans les machines à mots banalisés. Cette possibilité doit cependant avoir été prévue par le constructeur et est liée à la forme de l'instruction.

Nous citerons quelques exemples de réalisation des différentes opérations logiques. La forme des instructions est une des caractéristiques essentielles d'une calculatrice et peut servir de critère pour une classification.

Nous étudierons successivement quelques machines :

- à fonctionnement non séquentiel,
- à fonctionnement séquentiel.

--:--:--:--

- III -

QUELQUES EXEMPLES DE REALISATION

-:-:-:-

1°) MACHINES A FONCTIONNEMENT NON SEQUENTIEL. -

A) ELLIOTT 402	<u>1 + 1 adresse</u> <u>mots banalisés.</u>
----------------	--

Forme de l'instruction.

Les instructions et les nombres sont traités de la même manière : la machine est à mots banalisés.

La longueur du mot est de 32 digits binaires et l'instruction est composée de la manière suivante :

0 à 9	10 à 16	17 à 26	27 à 31
-------	---------	---------	---------

adresse A_2 de	lettre de	Adresse A_1
l'instruction suivante	fonction	Contrôle

C'est une instruction du type 1 + 1 adresse.

Une description sommaire des organes de mémoires et de calcul permettra de préciser la signification des différents digits.

- L'organe de mémoire se compose d'une part d'une mémoire magnétique à tambour, de temps d'accès moyen, dont la surface est divisée en 23 pistes de 128 mots chacune, d'autre part d'une mémoire à accès rapide, constituée de 15 lignes à retard de chacune 1 mot.

Parmi les 23 pistes du tambour, 8 seulement sont reliées électriquement à la machine :

7 pistes numérotées de 0 à 6,

1 quelconque des 16 pistes restantes désignée par le N° 7.

Adresse : Chaque mot est identifié par son adresse qui comprend :

- le numéro de la piste (3 digits),
- l'adresse dans la piste (7 digits).

Une instruction spéciale sur laquelle nous reviendrons permet d'identifier la piste N° 7, 7-0, 7-1, 7-2,, 7-15, choisie.

Les adresses 1 à 15 de la piste 0 correspondent aux 15 mots des mémoires rapides

6 L'organe de calcul comprend : un accumulateur et deux registres désignés par "M-registre" et "registre-multiplieur".

Les nombres s'additionnent dans l'accumulateur.

Le "M-registre" contient le multiplicande et sert de mémoire intermédiaire pour contenir les nombres pendant l'exécution des calculs

Le "registre-multiplieur" contient le multiplicateur ; il n'est pas directement adressable et le programmeur n'a pas à s'en soucier.

Lettre de fonction.

L'opération à exécuter est indiquée par les 7 digits 10 à 16 dont l'ensemble constitue la "lettre de fonction". Elle se subdivise en trois groupes de 2, 3 et 2 digits.

Les 2 premiers digits désignent la "source" ;

les 3 digits suivants : l'opération à exécuter ;

les 2 derniers digits : la "destination".

Désignons par (A), (S) et (D) le contenu de l'accumulateur, de la source et de la destination avant exécution de l'ordre, et par (A)', (S)' et (D)' le contenu des mêmes organes immédiatement après l'exécution de l'ordre. Un ordre de source S, destination D et opération F aura pour effet :

$$(D)' = (A)$$

$$(A)' = \phi [(A) (S)] \quad \text{où } \phi \text{ est l'opération}$$

indiquée par F,

$$(S)' = (S) \text{ à moins que la source et la destination}$$

ne soient identiques, auquel cas :

$$(S)' = (D)' = (A)$$

En d'autres termes, avant l'exécution de l'ordre, le contenu de l'accumulateur va à la "destination", et le résultat de l'opération commandée va dans l'accumulateur.

Il y a 4 "sources" et 4 "destinations" possibles:
source et destination "zéro" (0) : le contenu de l'accumulateur est perdu ;
source et destination "M registre" (1) : produit et division ;
source et destination "mémoire" : l'une au moins, de la source ou de la destination, doit comporter cette indication pour que l'adresse A_1 du nombre sur lequel doit porter l'opération soit considérée comme une adresse ;
source "entrée" et destination "sortie".

L'opération à effectuer est indiquée par 3 digits et l'on a 8 possibilités différentes. Ces opérations présentent toutes un caractère arithmétique et nous étudierons maintenant la signification des digits de contrôle qui permettent d'effectuer les diverses opérations logiques.

Digits de contrôle.

Ils se divisent en deux groupes de 2 et 3 digits ; les digits C et les digits B.

Les digits C permettent la réalisation des aiguillages et les digits B les modifications automatiques d'adresses.

- Les digits C peuvent représenter les nombres 0,1,2,3 qui ont la signification suivante :

$C = 0$: l'instruction s'exécute normalement.

$C = 1$: Saut conditionnel.

Si le contenu de l'accumulateur est positif, l'adresse de l'instruction suivante est celle indiquée par A_2 (instruction normale).

Si le contenu de l'accumulateur est négatif : l'adresse de l'instruction suivante est en A_1 , si l'opération commandée n'est pas une multiplication ou un ordre, de décalage.

Le test est effectué sur le résultat de l'instruction précédente.

C = 2 : Le contenu de la mémoire d'accès rapide N° 1 est augmenté de 1 lorsque l'instruction est exécutée. D'autre part, si le contenu de cette mémoire est nul au moment de l'exécution de l'instruction précédente, la prochaine instruction est prise en A_1 au lieu de A_2 (adresse normale de la prochaine instruction). On peut ainsi réaliser un saut conditionnel sur zéro en plaçant le nombre à tester dans cette position de mémoire.

C = 3 : Les 7 digits de la lettre de fonction sont alors obligatoirement nuls. A_1 n'a pas la signification normale d'une adresse : les 4 derniers¹ digits de A_1 désignent l'une des 16 pistes N° 7 que l'on désire connecter à la machine. Celle-ci reste connectée à la machine jusqu'à la prochaine instruction où C = 3.

Les digits B, au nombre de 3, permettent la modification automatique de l'adresse A_1 contenant le nombre sur lequel porte l'instruction. Le nombre n représenté par les digits B, $0 < n \leq 7$, indique la mémoire d'accès rapide dont le contenu doit être ajouté automatiquement à l'adresse de l'instruction avant exécution.

C'est l'utilisation normale des mémoires 1 à 7. On peut augmenter de 1 à chaque instruction le contenu de la mémoire d'accès rapide N° 1 grâce aux digits C (C = 2).

La combinaison des digits C et B permet ainsi d'augmenter de 1 l'adresse A_1 d'une instruction à chaque cycle d'exécution.

Le programme n'est pas modifié.

B) DEUCE	1 + 1 ou 2 + 1 adresses mots non banalisés.
----------	--

Forme de l'instruction.

Cette machine est à mots non banalisés et il existe des positions de mémoires spécialement réservées aux instructions. Nous verrons l'intérêt de ce mode de fonctionnement.

La longueur des nombres et des instructions est de 32 digits binaires et l'instruction est composée de la manière suivante :

1	2,3,4	5 à 9	10 à 14	15, 16	17 à 21	22 à 25	26 à 30	31	32
N	S	D	C	W	T	G			

Les digits 1, 22 à 25 et 31 sont inutilisés.

La position de chaque mot est définie :

- par une adresse-lieu,
- par une adresse-temps.

La description des organes de mémoires permettra de préciser ceci ainsi que la signification des différents digits. Indiquons dès maintenant que les digits N, S et D sont relatifs aux adresses-lieux de :

- la prochaine instruction (N),
- la source (S),
- la destination (D),

et que les adresses-temps se déduisent des digits W et T :

- W est relatif à l'adresse-temps de la source,
- T est relatif à l'adresse-temps : de l'instruction suivante,
de la destination.

C et G sont deux nombres caractéristiques dont nous préciserons ultérieurement la signification.

Une instruction a pour effet de commander le transfert d'un nombre de la "source" vers la "destination", et d'indiquer les adresses-temps et lieux de l'instruction suivante. Il existe des sources (ou des destinations) fonctionnelles qui jouent le rôle de lettre de fonction : l'instruction peut alors être considérée de la forme 1 + 1 adresse. C'est le cas général.

Il existe également des sources et des destinations qui sont de simples mémoires de travail : si la source et la destination désignent une mémoire de travail, il s'agit d'un simple transfert, et l'instruction peut être considérée du type 2 + 1 adresses.

Précisons maintenant les notions d'adresses-lieux et d'adresses-temps.

L'organe de mémoire comprend d'une part une mémoire à tambour magnétique de 256 pistes de 32 nombres. Cette mémoire n'est pas directement adressable, et une instruction spéciale permet le transfert-bloc d'une piste entière vers une mémoire de travail spécialisée

appelée D L 11 (ligne à retard de 32 nombres). Les nombres peuvent ensuite être répartis par le jeu des différentes instructions entre les autres mémoires de travail ou des organes fonctionnels.

Les mémoires de travail (lignes à mercure), d'accès moyen ou rapide, sont au nombre de 21 et se répartissent comme suit :

- 4 mémoires temporaires de chacune 1 nombre : { T S 13
.
.
T S 16

- 3 mémoires doubles de chacune 2 nombres : { D S 19
D S 20
D S 21

- 2 mémoires quadruples de chacune 4 nombres : { Q S 17
Q S 18

- 12 lignes à retard :

8 lignes pouvant contenir
chacune 32 instructions : { D L 1
.
.
.
D L 8

4 lignes de 32 nombres : { D L 9
.
.
.
D L 12

Les adresses-lieux se rapportent :
soit à l'une des 21 mémoires de travail précédentes,
soit à l'un des 11 organes spéciaux ("fonctionnels") qui permettent d'effectuer les opérations arithmétiques ou logiques.

Remarquons que l'adresse-lieu, N, de l' "instruction suivante" se rapporte nécessairement à l'une des mémoires D L

.... D L₈ (au total 32 x 8 soit 256 instructions).

Il est encore nécessaire de définir la position du mot (nombre ou instruction) dans la mémoire de travail : c'est le rôle de l'adresse-temps.

Dans une machine série, le temps est découpé en intervalles égaux dont chacun correspond à la durée d'une impulsion binaire.

Un mot est formé de 32 digits binaires dont l'ensemble définit le cycle-mineur (durée de passage d'un mot en un point quelconque de la machine). On pourra caractériser la position d'un mot dans une ligne à retard D L par un nombre m de cycles-mineurs, numéroté de 0 à 31 (32 mots par ligne à retard). Ainsi :

D L 1₅ signifie le mot qui, dans la ligne D L 1 (adresse-lieu) est disponible au cycle-mineur 5 (adresse-temps).

Les mots des mémoires simples sont disponibles à chaque cycle mineur.

Les adresses-temps de la prochaine instruction, de la source et de la destination sont alors définies de la manière suivante :

m est le numéro du cycle mineur où l'instruction I_m est entrée dans la mémoire d'ordre ;

l'adresse-temps de l'instruction suivante est alors :

$$m + T + 2$$

soit T + 2 cycles mineurs après l'instruction I_m ;

les adresses-temps de la source et de la destination sont les mêmes

$$m + W + 2$$

soit W + 2 cycles mineurs après l'instruction I_m.

Chaque instruction est en effet exécutée en deux temps :

- l'instruction est "décodée" (établissement des circuits nécessaires au transfert),
- l'instruction est exécutée.

La spécialisation des mémoires (séparation des nombres et des instructions) permet d'indiquer la deuxième phase d'une instruction (exécution) et la première phase de l'instruction suivante (décodage).

Ceci est toujours possible dans les machines à mots non banalisés. Notons cependant qu'un organe de mémoire double (CAB 3 000) donne la même possibilité pour les machines à mots banalisés, d'où un gain de vitesse appréciable.

Nous pouvons maintenant préciser la signification du nombre caractéristique C et de l'index G. Nous étudierons ensuite les diverses opérations logiques.

Le nombre caractéristique C peut prendre les valeurs 0, 2 et 1 et spécifie la durée du transfert :

C = 0 : le transfert dure 1 cycle mineur,

C = 2 : le transfert dure 2 cycles mineurs,

C = 1 : le transfert commence au cycle mineur $m + W + 2$, et dure jusqu'au cycle mineur $m + T + 2$.

Si $W > T$, la machine ajoute automatiquement un cycle majeur (32 cycles mineurs) à T.

L'index G peut prendre les valeurs 0 ou 1 et spécifie le mode de fonctionnement de la machine :

G = 0 : les instructions sont normalement exécutées,

G = 1 : on a un fonctionnement pas à pas.

Opérations logiques.

Saut conditionnel :

Les destinations 27 et 28 permettent de tester le contenu d'une des mémoires simples T S désignée par la source.

D = 27 $(T S) \geq 0$: l'adresse de l'instruction suivante est celle indiquée.

$(T S) < 0$: l'adresse-lieu de l'instruction suivante est celle indiquée, l'adresse-temps est décalée de 1 cycle mineur : $m + T + 3$.

D = 28 : le test est alors = 0
ou $\neq 0$.

Modification de l'adresse d'une instruction :

Cette modification peut porter soit sur l'adresse-lieu, soit sur l'adresse-temps.

a) adresse-lieu : L'adresse-lieu peut être modifiée par substitution (destination 0). Le mot transféré devient l'instruction suivante ; le transfert s'effectue dans la mémoire d'ordre.

b) adresse-temps : L'adresse-temps peut être modifiée par addition (source 28). On ajoute 1 à l'adresse-temps de l'instruction suivante : $(W \rightarrow W + 1)$.

Ces modifications d'adresses ne sont pas automatiques et nécessitent des instructions spéciales. Notons la différence avec l'adresse flottante telle que nous l'avons décrite à propos de l'ELLIOTT 402.

c)

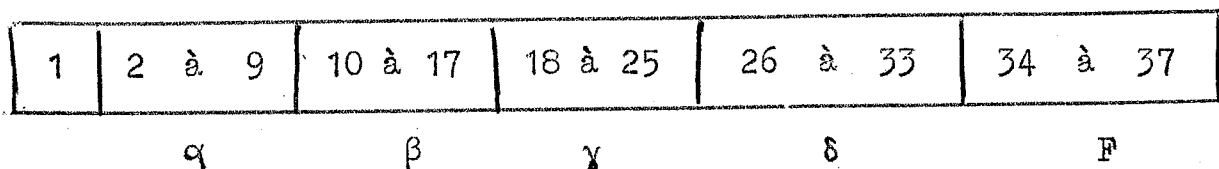
SWAC

 | 3 + 1 adresses
 |
 | mots banalisés.

Forme de l'instruction.

Les nombres et les instructions sont traités de la même manière : la machine est à mots banalisés.

La longueur du mot est de 37 digits binaires. Le premier digit est un digit de signe inutilisé dans les mots-instructions. Les 36 digits restants sont divisés en 4 groupes de 8 digits et un groupe de 4 digits spécifiant l'opération à effectuer.



Les digits α , β et γ indiquent les adresses de 3 nombres dans la mémoire d'accès rapide (256 mots), δ indique en général l'adresse de l'instruction suivante, mais peut également avoir une

signification particulière (dépendant de la lettre de fonction).

La mémoire générale (tambour magnétique) n'est pas directement adressable et des instructions spéciales permettent le transfert de blocs de 8, 16, 32 ou 64 mots de cette mémoire à la mémoire rapide, ou vice-versa.

Opérations logiques.

Saut conditionnel :

Les tests suivants sont utilisés :

Instruction

I_m : $(\alpha) - (\beta)$ le résultat est mis en γ
si $(\gamma) > 0$, la prochaine adresse est en δ ;
si $(\gamma) < 0$, la prochaine adresse est en I_{m+1}

Instruction

I_m : $|(\alpha)| - |(\beta)|$ le résultat est mis en γ
si $(\gamma) > 0$, la prochaine adresse est en δ ;
si $(\gamma) < 0$, " " " est en I_{m+1}

Dépassement de capacité :

Les dépassements de capacité consécutifs à une addition ou soustraction conduisent à un saut à l'adresse δ .

Dans le cas contraire, la prochaine instruction est prise en I_{m+1} .

Il existe également des instructions d'addition ou de soustraction, telles que la prochaine instruction soit toujours en δ .

Cette machine peut être considérée, soit comme à 3 + 1 adresses ou 3 adresses (avec rupture de séquence inconditionnelle).

Modification d'adresse :

Celle-ci n'est pas automatique : les instructions et les nombres sont traités de la même manière, et on peut modifier soit la totalité d'une instruction, soit simplement la partie adresse de l'instruction en utilisant l'instruction d'"intersection".

2°) MACHINES A FONCTIONNEMENT SEQUENTIEL. -

A) GAMMA	1 adresse
	mots non banalisés.

Forme de l'instruction.

Cette machine est du type à "connexions" : les ordres sont donnés à la calculatrice par un tableau de connexions amovibles qui permet d'afficher, pour chaque problème, l'ensemble des opérations arithmétiques ou logiques élémentaires.

Une machine à cartes perforées, connectées à la calculatrice, permet l'introduction des données numériques, et l'extraction des résultats.

Chaque opération élémentaire est définie par 4 "instructions" élémentaires, dont l'ensemble forme la ligne d'instruction (équivalente au mot-instruction). Il existe 64 lignes d'instructions.

Chaque ligne d'instructions spécifie :

- le type d'opération (TO), arithmétique ou logique (lettre de fonction)
- l'adresse (AD) du nombre sur lequel doit porter cette opération,
- l'ordre "début" d'adresse (OD),
- l'ordre "fin" d'adresse (OF).

Ces deux derniers ordres complètent l'adresse en limitant aux positions voulues, celles des 12 positions décimales de la mémoire d'adresse AD. Chaque mémoire est, en effet, de longueur variable et peut être adaptée à la longueur du mot à enregistrer (au maximum 12 chiffres décimaux).

Les instructions sont donc du type à 1 adresse.

L'évolution des différentes phases d'une opération est sous le contrôle permanent de deux comparateurs :

- le comparateur de cadrage, chargé des décalages (nécessités en particulier par les positions variables que peut occuper un mot dans une mémoire).
- le comparateur général, chargé des transferts.

Ce comparateur est doté de mémoire et conserve le souvenir des comparaisons faites : le résultat d'un test n'est pas obligatoirement utilisé immédiatement, et il peut être également utilisé autant de fois qu'on le désire.

Le fonctionnement est du type séquentiel : la calculatrice prend connaissance des différentes instructions, ligne par ligne, du tableau de connexions. Il existe 64 lignes d'instructions et une mémoire spéciale (NL) (compteur ordinal) dont le contenu progresse, unité par unité, de 0 à 63, sélectionne les différentes lignes d'instructions.

Différentes "variantes" permettent la réalisation des ruptures de séquence.

Opérations logiques.

- Ruptures de séquence :

La progression spéciale de la mémoire NL peut être provoquée à partir des éventualités suivantes :

- à partir d'éventualités observées sur la machine connectée ("variantes à relais" VR).
- à partir d'éventualités observées sur le calculateur ("variantes comparaison" VC). Cette opération doit toujours être précédé de l'opération comparaison dont le résultat est conservé par le comparateur jusqu'à la prochaine comparaison. On compare le contenu de l'accumulateur (mémoire M_1) au contenu d'une mémoire (plus grand, plus petit ou égal).
- à partir d'une nécessité systématique (VS) (rupture de séquence inconditionnelle).

Ces ruptures de séquence sont commandées par la lettre de fonction "0".

L'adresse (AD) et l'ordre de fin d'adresse (OF) permettent de les distinguer entre elles.

L'adresse de l'instruction suivante est telle que :

$$(NL) = 4(OD) + \frac{(OF)}{4}$$

Le numéro (NL) de la prochaine ligne d'instruction étant fixé, un tableau de connexions spécial permettra de lever l'indétermination de (OD) et (OF), compte tenu de l'adresse (AD) et de l'ordre (OF) relatifs à la variante spécifiée.

Une rupture de séquence procède par suppression de lignes d'instructions : les diverses voies possibles sont nécessairement disposées à la suite des unes des autres.

Les tests utilisés peuvent concerner :

soit l'observation d'un certain fait,

soit l'observation de plusieurs faits :

a) si a ou b , ou les deux, sont observés, sauter à l'instruction x, sinon continuer la séquence normale des instructions;

b) si a et b sont observés, sauter à l'instruction x, sinon continuer la séquence normale des instructions.

- Modifications des instructions :

Il est possible de modifier soit l'adresse d'une instruction, soit la lettre de fonction.

Ces modifications se font par substitution en sélectionnant, sur le tableau de connexions, les différentes adresses ou opérations choisies. Toutes les connexions correspondant aux diverses éventualités doivent évidemment être faites.

Les modifications d'adresse peuvent s'effectuer par la sélection :

soit de l'adresse (AD)

soit des deux instructions (OD) et (OF)

soit des 3 instructions (AF), (OD) et (OF).

Les modifications de lettre de fonction peuvent être obtenues en agissant :

soit sur la lettre de fonction proprement dite (TO),
soit sur l'adresse (AD) ou l'ordre (OF) pour les variantes.

B) M . T . C .	<u>1 adresse</u>
	<u>Mots banalisés.</u>

Forme de l'instruction.

La longueur du mot est de 16 digits binaires, et les instructions sont du type à 1 adresse :

5 digits sont réservés à la lettre de fonction,
11 digits à l'adresse du nombre sur lequel porte l'opération.

Dans certaines instructions spéciales (ordre de décalage), certains digits, parmi les 11 normalement réservés à l'adresse, ont une signification particulière et spécifient des variantes de l'opération commandée.

Ces instructions ne présentent pas un caractère logique, et nous ne les étudierons pas.

Les 11 digits réservés à l'adresse permettent d'indiquer l'une des 2 048 positions de mémoires. La mémoire, à tambour magnétique est divisée en 15 groupes de 2 048 mots. Il existe aussi 2 groupes de 2 048 mots à accès rapide (tores magnétiques).

Une instruction spéciale "sof a" indique que toutes les instructions qui suivent se rapportent au groupe "a)", jusqu'à la prochaine instruction du même type.

Opérations logiques.

La machine est à fonctionnement séquentiel : on distinguera les ordres de ruptures de séquence conditionnels et inconditionnels.

- Saut conditionnel :

Les ordres de saut conditionnel sont les suivants :

- 1) "tn x" : si le signe du contenu de l'accumulateur est négatif, sauter à l'instruction d'adresse x dans le même groupe.
- 2) "tn0 x" : si le signe du contenu de l'accumulateur est négatif, sauter à l'instruction d'adresse x dans le groupe "a", indiqué par la précédente instruction "sof a".
- 3) "id x" : on compare le contenu de l'accumulateur au mot d'adresse x.

S'il y a identité, on saute à l'instruction C+2 (C est l'adresse de l'instruction "id x").

Dans le cas contraire, on exécute normalement l'instruction suivante (C+1), et l'accumulateur contient des 1 dans chaque position où il n'y a pas correspondance avec le mot d'adresse x.

- 4) "t0 x" : cette instruction concerne les dépassements de capacité. Normalement, un dépassement de capacité arrête la machine. L'instruction "t0 x" placée après les ordres d'addition ou de soustraction supprime cet arrêt, et, en cas de dépassement de capacité, ordonne de sauter à l'instruction d'adresse x. Cette instruction est sans effet s'il n'y a pas dépassement de capacité.

- Saut inconditionnel :

Les instructions de saut inconditionnel sont les suivantes :

- 1) "tr x" : sauter à l'instruction d'adresse x du même groupe.
- 2) "tr 0 x" : sauter à l'instruction d'adresse x du groupe "a" indiqué par la précédente instruction "sof a".

- Modifications des instructions :

La machine est à mots banalisés, et les instructions peuvent être modifiées totalement ou en partie, par addition ou substitution. Deux instructions spéciales permettent de remplacer en bloc l'adresse d'une instruction :

- 1) " ra x" : on remplace l'adresse de l'instruction placée en x (digits 5 - 15) par les digits 5 - 15 de l'accumulateur.
- 2) "rf x" : on remplace l'adresse de l'instruction placée en x par les digits 5 - 15 d'une mémoire d'accès rapide "LR 2". Cette mémoire contient normalement une adresse y+1 où y est l'adresse de l'instruction qui a effectué le dernier transfert. Cette instruction facilite le retour au programme principal après l'exécution d'un sous-programme.

c)

CAB

 | 1 adresse.
| mots banalisés.

Il existe plusieurs séries de cette calculatrice.

"CAB 2 000"

Forme de l'instruction.

La longueur du mot peut être suivant les modèles de 22, 32 ou 40 digits. Une grande longueur de mot permet des calculs plus précis mais, dans les machines série, nécessite un temps de calcul plus élevé. Nous étudierons plus particulièrement la calculatrice " CAB 2 022".

L'instruction est du type 1 adresse. La lettre de fonction est définie par les digits 16 à 20 ; l'adresse, par les digits 2 à 15.

1	2 à 15	16 à 20	21	22
B	adresse	lettre de fonction	P	θ

Le premier digit B concerne la modification automatique des adresses.

L'indice "paramètre" P permet l'introduction directe des constantes : l'adresse N est alors considérée, comme un nombre, ce qui permet une économie des positions de mémoire.

Le dernier digit θ ou "lettre de code" est un indice particulier de modification d'adresse.

L'organe de calcul comprend deux totalisateurs A et B accessibles séparément (ou simultanément) par la lettre de fonction. Le registre intervient comme mémoire temporaire dans la multiplication et contient le multiplieur.

Opérations logiques.

Nous distinguerons les ruptures de séquences conditionnelles et inconditionnelles et les modifications d'adresses.

- Saut conditionnel :

On teste le contenu de l'un des totalisateurs A ou B. Les instructions suivantes sont utilisées pour réaliser des ruptures de séquence conditionnelles:

ZN	si	(A) = 0,	sauter à l'instruction d'adresse N.
EN	si	(A) ≥ 0,	" " "
GN	si	(A) < 0,	" " "
QN	si	(B) ≥ 0,	" " "
YN	si	(B) < 0,	" " "

- Saut inconditionnel :

L'instruction ZPN signifie : sauter dans tous les cas à l'adresse N.

Deux lettres de fonctions J et W permettent de réaliser des

ruptures de séquence inconditionnelles et l'introduction automatique de sous-programmes. On les appelle "opérations symbolisées". J et W sont munies d'une deuxième lettre spécifiant l'opération à effectuer (numéro du sous-programme choisi).

Les lettres J et W ont pour effet :

- d'effectuer une rupture de séquence inconditionnelle vers un programme d'aiguillage prévu par le constructeur, puis vers le sous-programme choisi.

- de recopier dans le registre B l'adresse de retour au programme principal ($n+1$ si n est l'adresse de l'instruction J ou W).

- Modifications d'instructions :

La machine est à mots banalisés et toutes les opérations sont possibles sur les instructions. Il est cependant prévu des modifications automatiques d'adresse.

- Indice B de modification d'adresse :

Toutes les instructions se référant à une adresse N peuvent être suivies de la lettre B. Dans ce cas, l'adresse N est modifiée en ajoutant, avant exécution de l'ordre, le contenu du registre B à l'adresse N.

Cet indice sert en particulier au comptage. Combiné avec un ordre de rupture de séquence, il est possible de décrire une boucle un nombre de fois fixé à l'avance. On peut également utiliser cet indice pour la réalisation d'aiguillages à plusieurs voies. Le choix entre les différentes branches est déterminé par le contenu du totalisateur B ; un ordre de saut inconditionnel vers les différentes branches est inscrit à chaque adresse $N + (B)$.

- Lettre code θ :

Cet indice permet d'écrire tous les programmes ou sous-programmes en adresse relative : l'adresse de la première instruction est 0. A l'entrée du programme, la machine rétablira le numérotage réel en ajoutant l'adresse relative à l'adresse initiale.

"CAB 3 000".

La calculatrice "CAB 3 000" se différencie de "CAB 2 000" par une plus grande spécialisation des organes et des lettres de fonction, d'où un gain de vitesse appréciable est une souplesse de fonctionnement accrue.

La décomposition de la mémoire d'accès rapide, seule adressable par mots, en deux groupes de 512 mots, permet un travail simultané sur chacun d'eux : la phase d'exécution d'un ordre est imbriqué avec la phase de codage de l'ordre suivant. En cas d'incompatibilité d'adresse "Ordre" et "Nombre", c'est-à-dire lorsque celles-ci se réfèrent au même groupe de mémoires rapides, un dispositif spécial place un cycle mineur d'attente entre ces deux phases.

La mémoire générale à tambour magnétique est considérée comme mémoire externe et n'est accessible que par groupes de mots.

L'organe de calcul comprend 4 totalisateurs dont l'un est spécialisé dans les modifications d'adresses. Chacun d'eux est accessible directement.

Des indices de spécialisation de lettres de fonction reviennent en fait à doubler le nombre d'opérations possibles. Citons par exemple :

la multiplication additive ou soustractive,

l'addition ou soustraction, avec ou sans effacement, etc..

Du point de vue des opérations logiques, les " CAB 3 000 " sont identiques aux " CAB 2 000". Il faut cependant ajouter aux tests classiques (plus grand; plus petit ou égal à 0), le contrôle du dépassement de capacité qui conduit également à une rupture de séquence conditionnelle.

D)

ELECOM 125

 | 2 adresses.
mots banalisés.

Forme de l'instruction.

La longueur du mot est de 44 digits binaires groupés par ensembles de 4 digits (système décimal codé). Chaque instruction comprend :

1 signe	:	4 digits binaires,
1 lettre de fonction	:	2 x 4 digits binaires,
1 adresse a	:	4 x 4 " "
1 adresse b	:	4 x 4 " "

Les digits de signe permettent de spécifier des variantes des lettres de fonction.

Les adresses a et b sont :

- soit celles des opérateurs, le résultat reste alors dans l'accumulateur,
- soit l'adresse d'un des opérateurs, et l'adresse où doit être mis le résultat. Au début de l'opération, l'accumulateur contient l'autre opérateur.
- soit l'adresse a ou b de la prochaine instruction après les opérations de test.

L'organe de calcul comprend trois totalisateurs :

- l'accumulateur A,
- une mémoire temporaire L contenant le multiplicande ou le diviseur,
- un registre X contenant le multiplieur ou le quotient.

Opérations logiques.

- Saut conditionnel :

Les possibilités de rupture de séquence conditionnelles sont nombreuses. Les instructions suivantes permettent de les réaliser :

(18) : on compare le contenu des adresses a et b :

(a) < (b),	le contenu de l'accumulateur est	(A) = 0,4
(a) = (b)	"	" (A) = 0
(a) > (b)	"	" (A) = -0,4

a et b sont traités comme deux nombres de 44 digits binaires ; les digits de signe sont traités comme de simples digits supplémentaires et non comme des signes algébriques. Cette instruction est utile pour comparer deux mots alphanumériques entre eux ou à un nombre. Par exemple, un signe négatif (0100) sera considéré comme supérieur à un espace (0010) ou au signe caractéristique d'un mot alphanumérique (0011), mais il sera considéré comme inférieur au signe 0101 précédant les instructions normalement exécutées.

Cette instruction est suivie de l'instruction de saut :

(70) : on teste le contenu de l'accumulateur A :

- si $|A| = 0$: la prochaine instruction est à l'adresse suivante (fonctionnement séquentiel normal) ;
- si $|A| \neq 0$: la prochaine instruction sera prise en a ou b suivant le "signe".
Si le signe est :
 - 0001)
 - 0010) sauter à l'adresse a ;
 - 0011)
 - 0100)
 - 0101) sauter à l'adresse b.
 - 0110)

(50) : le test porte sur l'adresse a-1 considérée comme un nombre :

si $a-1 = 0$: la prochaine instruction est à l'adresse
suivante (fonctionnement séquentiel normal) ;

si $a-1 > 0$: la prochaine instruction est à l'adresse b.

Dans les deux cas, l'instruction 50 est remise en mémoire à son adresse initiale, l'adresse a étant remplacée par a-1.

Cette instruction est utile pour le comptage et permet par exemple, de décrire une boucle un nombre de fois fixé à l'avance.

Une rupture de séquence peut être également due à un dépassement de capacité consécutif aux instructions d'addition ou de soustraction. La prochaine instruction n'est pas en C+1, mais dans une position de mémoire fixe. Cette instruction est exécutée, et on passe ensuite à l'adresse C+2 ou C+3, suivant les cas, à moins que l'instruction d'adresse fixe ne soit elle-même une instruction de saut.

On reviendra à l'adresse C+2 si le dépassement de capacité apparaît lorsque (a) est ajouté, ou soustrait, au contenu de l'accumulateur (A).

On reviendra à l'adresse C+3 si le dépassement apparaît lorsque (b) est ajouté, ou soustrait, au contenu de l'accumulateur (A) + a.

- Saut inconditionnel :

(20) : mettre (A) en (b), l'instruction suivante est prise en a.

(25) : " (X) " " " " " " "

(26) : on effectue le produit logique du mot d'adresse C+1 et des 4 derniers digits décimaux du mot d'adresse b (si le mot d'adresse b est une instruction, ces quatre digits décimaux constituent l'adresse a de cette instruction). L'instruction suivante est prise en a.

(4) ; c'est un ordre de normalisation : le nombre de décalage est enregistré dans X et à l'adresse b. L'instruction suivante est prise en a.

Il n'existe pas de modification automatique d'adresse.
 Ces modifications sont réalisées par opérations sur les instructions
 considérées comme nombres (voir instruction 26).

E)

SEAC

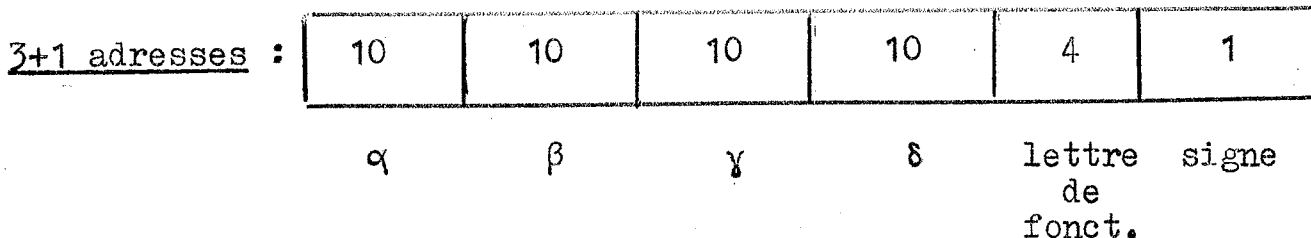
 3 ou 3+1 adresses.

 mots banalisés.

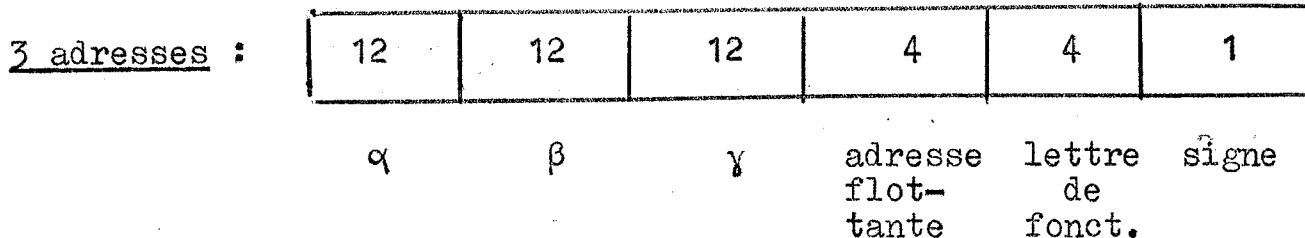
 série.

Forme de l'instruction.

La calculatrice SEAC peut être programmée en fonctionnement
 séquentiel ou non séquentiel. La longueur du mot est de 45 digits
 binaires et une instruction est constituée de l'une ou l'autre
 manière suivante :



α , β , γ sont les adresses de nombres,
 δ est l'adresse de l'instruction suivante,
 4 digits sont réservés à la lettre de fonction,
 et 1 digit au signe (sans signification dans les mots-instructions).



α et β indiquent, comme précédemment, les adresses de nombres (opéra-
 teurs),
 γ peut être : soit l'adresse d'un nombre (résultat),
 soit l'adresse de l'instruction suivante (saut condi-
 tionnel).

4 digits spécifient la lettre de fonction,
1 digit est réservé au signe (sans signification dans les mots-
instructions),

Les 4 digits restants concernant :

- la modification automatique des adresses,
- l'introduction automatique des sous-programmes.

Opérations logiques.

1°) Comparaison des valeurs absolues ou algébriques des
mots d'adresses α et β ;

si $(\alpha) < (\beta)$: sauter à l'instruction d'adresse γ .

Dans le cas contraire :

- fonctionnement non séquentiel (3+1 adresses), sauter à l'adresse δ ,
- fonctionnement séquentiel (3 adresses) : suivre l'ordre normal des instructions.

2°) Modification automatique d'adresse :

(uniquement dans le fonctionnement séquentiel).

3 des quatre digits d'adresse flottante indiquent si les adresses α , β ou γ doivent être considérées comme absolues ou relatives. Chaque instruction peut donc comporter des adresses absolues ou relatives (dans ce dernier cas, on ajoute à l'adresse indiquée, l'adresse de l'instruction en cours d'exécution).

Le dernier digit, d , permet l'introduction automatique des sous-programmes.

Une des particularités intéressantes de cette machine réside dans le fait qu'elle possède deux organes de commande indépendants. L'absence ou la présence du digit d passe le contrôle à l'un ou l'autre de ces organes : il est ainsi possible d'introduire automatiquement un sous-programme dans le programme principal.

- IV -

C O N C L U S I O N S

-:-:-:-

Il ressort de l'étude précédente que, d'une part la forme des instructions, d'autre part, la manière dont sont traitées les instructions (machines à mots banalisés ou non banalisés), peuvent être considérées comme deux caractéristiques essentielles d'une calculatrice.

Nous essaierons de préciser l'influence de ces deux facteurs sur les possibilités d'opérations logiques, et de dégager, parmi les diverses tendances, la ligne directrice de l'évolution actuelle de la structure logique des calculatrices.

Une remarque s'impose en ce qui concerne la banalisation des cellules de mémoires : la très grande majorité des machines actuelles sont à mots banalisés. La non-banalisation se rencontre :

- soit sur des adaptations modernes de matériel existant en grande série (machines GAMMA de BULL),
- soit sur des machines d'un fonctionnement très particulier (DEUCE).

Le principal avantage de cette conception réside dans un accroissement de vitesse : la phase "exécution" d'une instruction peut être imbriquée avec la phase "décodage" de l'instruction suivante, les nombres et les instructions étant toujours en des positions de mémoires distinctes. Ce mode de fonctionnement n'est cependant pas incompatible avec la banalisation des cellules de mémoires (nous en avons donné un exemple : CAB 2200 et 3000).

D'après les exemples cités, l'avantage des mémoires banalisées semble évident pour la modification (automatique ou non) des instructions et la souplesse de fonctionnement.

Examinons maintenant l'influence de la constitution du mot.

La composition d'une instruction est étroitement liée à la longueur du mot et nous devons comparer les différents types d'instructions pour une longueur de mot donnée.

D'une manière générale, une grande longueur de mot paraît préférable : la précision des calculs est améliorée et l'on peut disposer de digits spécialisés dans l'instruction. (Il est d'ailleurs possible également, de disposer de deux instructions par mot). Cette longueur influe cependant :

- sur le volume, et par suite sur le prix de la machine,
- sur le temps de calcul (machine série) ou la complexité des circuits (machine parallèle).

Les instructions les plus fréquemment rencontrées sont du type à 1 adresse. L'indication de l'adresse de la prochaine instruction (fonctionnement non séquentiel) est en effet le plus souvent triviale et il semble préférable d'utiliser ces digits pour spécifier soit des variantes, soit l'exécution automatique de certaines opérations (CAB).

Il y a évidemment contradiction entre l'automatisme de ces opérations et la simplicité de la conception de la machine.

Une plus grande souplesse de fonctionnement conduit nécessairement à une spécialisation plus grande des organes. Nous avons vu, par exemple, que les calculatrices du type "CAB 3 000" possèdent un registre spécialisé dans les modifications d'adresse, "SEAC" possède deux organes de contrôle qui permettent l'introduction automatique des sous-programmes, et ces deux calculatrices nous semblent assez voisines du point de vue de la conception logique. Il est intéressant de remarquer à ce sujet que "CAB 3 000" est du type 1 adresse et "SEAC" du type 3 adresses. Cette distinction n'est pas, à notre avis, essentielle du point de vue logique : une instruction à plusieurs adresses peut toujours être décomposée en plusieurs instructions à 1 adresse.

Cette différence tient à la longueur des mots utilisés :

("CAB 3 000" : 24 digits binaires, "SEAC" : 45 digits binaires).

Bien que datant de plusieurs années, et qu'étant l'une des premières calculatrices réalisées, la conception logique de " SEAC " semble avoir fait école et nous paraît toujours susceptible de résumer la structure logique des calculatrices.

-:-:-