

**Zeitschrift:** Bulletin technique de la Suisse romande  
**Band:** 85 (1959)  
**Heft:** 22

**Artikel:** Réglage et calculatrices électroniques: description de la machine arithmétique électronique Zebra et de ses performances  
**Autor:** Jeanneret, C.  
**DOI:** <https://doi.org/10.5169/seals-64144>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. 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. [Siehe Rechtliche Hinweise.](#)

### **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. [Voir Informations légales.](#)

### **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. [See Legal notice.](#)

**Download PDF:** 17.11.2024

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

# BULLETIN TECHNIQUE DE LA SUISSE ROMANDE

paraissant tous les 15 jours

**ORGANE OFFICIEL**

de la Société suisse des ingénieurs et des architectes  
de la Société vaudoise des ingénieurs et des architectes (S.V.I.A.)  
de la Section genevoise de la S.I.A.  
de l'Association des anciens élèves de l'EPUL (Ecole polytechnique  
de l'Université de Lausanne)  
et des Groupes romands des anciens élèves de l'E.P.F. (Ecole  
polytechnique fédérale de Zurich)

**COMITÉ DE PATRONAGE**

Président: J. Calame, ing. à Genève  
Vice-président: E. d'Okolski, arch. à Lausanne  
Secrétaire: S. Rieben, ing. à Genève

**Membres:**

Fribourg: H. Gicot, ing. ; M. Waeber, arch.  
Genève: G. Bovet, ing. ; Cl. Grosgrin, arch. ; E. Martin, arch.  
Neuchâtel: J. Béguin, arch. ; R. Guye, ing.  
Valais: G. de Kalbermatten, ing. ; D. Burgener, arch.  
Vaud: A. Chevalley, ing. ; A. Gardel, ing.  
M. Renaud, ing. ; Ch. Thévenaz, arch.

**CONSEIL D'ADMINISTRATION**

de la Société anonyme du « Bulletin technique »  
Président: A. Stucky, ing.  
Membres: M. Bridel ; R. Neeser, ing. ; P. Waltenspühl, arch.  
Adresse: Ch. de Rosneck 6, Lausanne

**RÉDACTION**

D. Bonnard, ing.  
Rédaction et Editions de la S. A. du « Bulletin technique »  
Tirés à part, renseignements  
Adresse: Case Chauderon 475, Lausanne

**ABONNEMENTS**

1 an . . . . .	Suisse Fr. 26.—	Etranger . . . . .	Fr. 30.—
Sociétaires . . . . .	» » 22.—	» . . . . .	» 27.—
Prix du numéro . . . . .	» » 1.60		

Chèques postaux: « Bulletin technique de la Suisse romande »,  
N° II. 87 75, Lausanne.

Adresser toutes communications concernant abonnement, changements  
d'adresse, expédition, etc., à: Imprimerie La Concorde, Terreaux 29,  
Lausanne

**ANNONCES**

Tarif des annonces:	
1/1 page . . . . .	Fr. 275.—
1/2 » . . . . .	» 140.—
1/4 » . . . . .	» 70.—
1/8 » . . . . .	» 35.—

Adresse: Annonces Suisses S. A.  
Place Bel-Air 2. Tél. (021) 22 33 26. Lausanne et succursales

**SOMMAIRE**

Description de la machine arithmétique électronique ZEBRA et de ses performances, par C. Jeanneret, physicien diplômé.  
Conception et structure des ordinateurs modernes, par P.-A. Bobillier, ingénieur EPUL, licencié ès sciences.  
Bibliographie. — Divers. — Les Congrès.  
Documentation générale. — Documentation du bâtiment. — Informations diverses.

## RÉGLAGE ET CALCULATRICES ÉLECTRONIQUES \*

### DESCRIPTION DE LA MACHINE ARITHMÉTIQUE ÉLECTRONIQUE ZEBRA ET DE SES PERFORMANCES \*\*

par C. JEANNERET, physicien diplômé, Standard Téléphone et Radio S. A., Zurich

Avant de commencer la description d'une machine à calculer électronique, il est bon de donner tout d'abord une définition d'une telle calculatrice et de situer la machine décrite dans l'ensemble des machines existantes.

Du point de vue de ses fonctions, une machine à calculer ou « computer » est un dispositif qui accepte des instructions et des nombres donnés dans une certaine forme, effectue à partir de ces informations les opérations logiques et mathématiques commandées et délivre les résultats sous la forme désirée.

Quant à la classification des calculatrices, il faut faire une première distinction entre les deux groupes suivants: les machines analogiques et les machines digitales.

Les machines analogiques sont des machines pouvant effectuer un nombre limité de calculs dans lesquels les grandeurs des nombres sont représentés par des quantités physiques telles que longueurs, poids, tensions, résistances, etc. Ces calculatrices sont spécialement

appropriées pour résoudre des problèmes dans lesquels se retrouvent toujours les mêmes fonctions et opérations mathématiques et où seules les valeurs numériques changent. La précision des calculs effectués par ces machines est limitée principalement par la précision avec laquelle ces grandeurs physiques peuvent être mesurées.

La calculatrice ZEBRA ne fait pas partie de ce groupe de machines, mais appartient au groupe des calculatrices dites « calculatrices digitales ».

Les calculatrices digitales sont des machines qui utilisent des nombres et des instructions représentés par des symboles ou digits et dans lesquelles les opérations arithmétiques sont effectuées d'une façon

\* Le précédent numéro spécial « Réglage et calculatrices électroniques » qui porte le n° 21 du *Bulletin technique*, constitue en quelque sorte une introduction au présent numéro spécial; nous prions donc nos lecteurs de bien vouloir s'y référer.

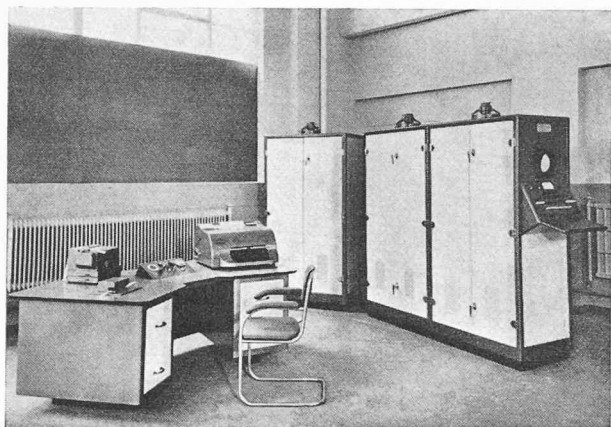
\*\* Conférence prononcée lors des 3<sup>es</sup> Journées d'Information de l'Association Suisse pour l'Automatique les 25 et 26 juin 1958 à Lausanne.

logique semblable aux opérations effectuées normalement lors d'un calcul. La précision des résultats obtenus est proportionnelle au nombre de digits que la machine peut accepter dans un mot.

Dans ce groupe, nous pouvons distinguer trois grandes catégories de calculatrices.

Dans la première catégorie sont comprises les calculatrices capables de calculer des problèmes mathématiques compliqués. Elles possèdent une grande mémoire dont les cellules sont facilement accessibles et calculent rapidement. Les machines de cette catégorie trouvent leurs applications principalement dans les calculs mathématiques, scientifiques et techniques.

Les machines de la deuxième catégorie sont destinées à calculer des problèmes relativement simples se répétant souvent, mais où la quantité des valeurs numériques est très grande. Ces calculatrices devront être équipées de mémoires extérieures additionnelles, calculer rapidement et posséder des dispositifs d'entrée et de sortie rapides en raison du grand nombre d'infor-



Vue d'ensemble de la calculatrice électronique ZEBRA.

mations à manipuler. Ces machines sont plutôt utilisées pour des calculs statistiques et commerciaux dans lesquels une grande quantité de nombres doit être prise en considération.

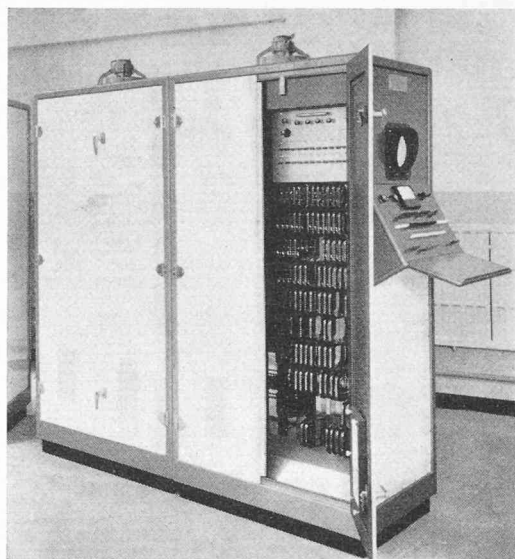
La troisième catégorie comprend les machines mécaniques et électromécaniques qui sont de petites calculatrices aux possibilités limitées comparativement aux machines des deux premières catégories.

La ZEBRA, de par ses caractéristiques, appartient à la catégorie désignée par 1, mais sa position dans cette catégorie peut encore être précisée. En effet, une distinction peut être faite entre deux classes de machines, celle des « calculateurs mathématiques » et celle des « calculateurs pour buts spéciaux ».

Les premiers sont les calculateurs capables de calculer des problèmes compliqués de tout genre, tandis que les seconds sont, comme l'indique le nom de leur type, destinés à faire un calcul ou un groupe de calculs bien défini pour lequel ils ont été spécialement étudiés. Ces derniers possèdent des circuits particulièrement rapides et des dispositifs qui physiquement leur permettent de faire automatiquement certaines opérations. Dans cette nouvelle subdivision, la ZEBRA prend place dans les machines du type « Calculateurs mathématiques ».

*Une brève définition de la ZEBRA peut être la suivante :*

*C'est un calculateur mathématique digital capable de traiter et de résoudre des problèmes généraux et qui trouve son application principalement dans la résolution de problèmes mathématiques, scientifiques et techniques.*

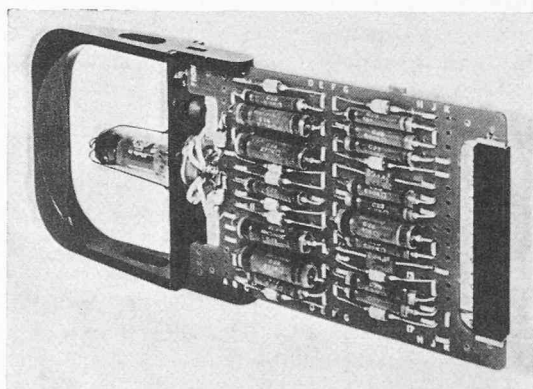


Vue de l'armoire du calculateur montrant à l'intérieur quelques « plug-in ».

Cependant, une des caractéristiques de la ZEBRA étant son système logique particulièrement souple, elle peut, légèrement transformée, être adaptée à la résolution de problèmes commerciaux. Une telle transformation portera principalement sur l'adjonction de mémoires supplémentaires, qui pourront être des bandes magnétiques ou des ferrites.

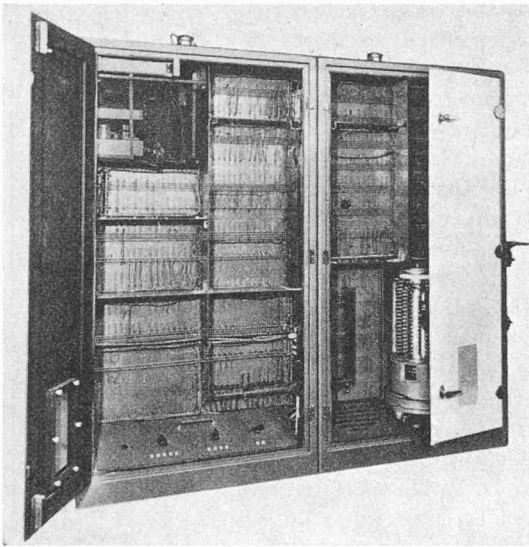
Il est difficile de donner une liste complète des nombreuses applications de la ZEBRA, mais par exemple, elle sera utilisée avec avantage dans les domaines suivants :

- en mathématique, pour la résolution d'équations différentielles, les calculs de transformations de matrices, l'établissement de tables, etc. ;
- en électricité ou en électronique, pour le calcul de transformateurs et des bobines, pour les calculs des transmissions et des filtres ;
- en optique, pour les calculs des systèmes de lentilles.



Vue d'un « plug-in » sur lequel sont montées les composantes.

La ZEBRA trouvera également de nombreuses applications dans la résolution des problèmes concernant l'industrie mécanique, aéronautique, et rendra de grands services dans les calculs de génie civil.



Armoire du calculateur montrant l'ensemble des unités et à droite le tambour magnétique.

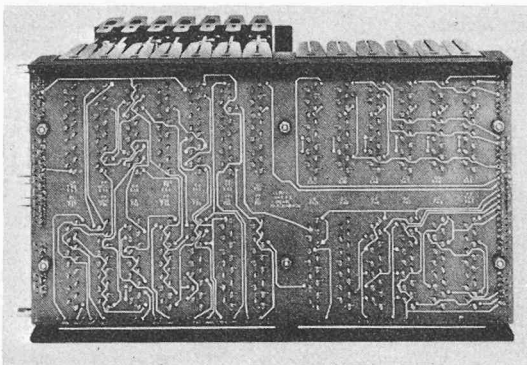
Fabriquée par la « Standard Telephones and Cables Ltd. », à Newport, en Angleterre, la calculatrice ZEBRA, ou de son vrai nom la « Stantec Zebra », a été développée par la maison susmentionnée en collaboration avec les ingénieurs de l'Administration hollandaise des P.T.T.

Vue de l'extérieur, la ZEBRA est composée des trois groupes suivants :

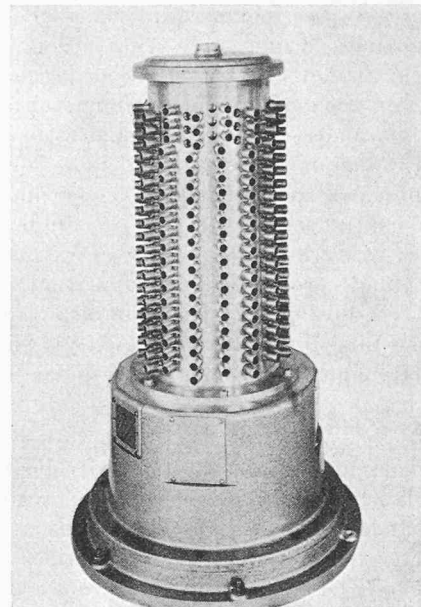
*Le calculateur*, qui contient le système logique de la machine, c'est-à-dire la mémoire magnétique, les unités arithmétiques et de contrôle, ainsi qu'un dispositif de test de la machine.

*Le pupitre de commande*, sur lequel sont montés les dispositifs d'entrée et de sortie de la machine. Pour l'entrée, un lecteur de bandes perforées et pour la sortie un téléscripteur et un perforateur rapide. Le téléscripteur ou le perforateur rapide sont utilisés suivant que les résultats doivent être imprimés ou perforés dans une bande. Quelques clés de commande sont également montées sur le pupitre.

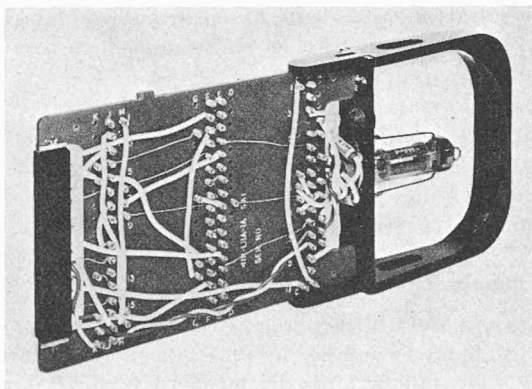
*Le groupe d'alimentation*, fournissant les sources de tensions nécessaires au fonctionnement de la machine. Ces tensions sont de 230 V pour les tubes et de 12 V pour les transistors. La puissance totale consommée est d'environ 5 kilowatts.



Ensemble de « plug-in » montés sur un tableau et reliés par circuits imprimés.



Tambour magnétique.



« Plug-in », face montrant les joints par bobinage.

Dans la description d'une calculatrice électronique telle que la ZEBRA, il est bon de dire quelques mots des composantes et du principe de montage de ces éléments. Le groupe arithmétique et le groupe de contrôle sont composés de petites unités ou « plug-in » connectés entre eux pour former des unités complètes. Ces petites unités sont branchées dans un tableau et les connexions entre elles sont assurées par des circuits imprimés. Ce principe de montage facilite énormément l'entretien de la machine et la recherche des défauts, si celles-ci devaient se produire. Un simple remplacement du « plug-in » défectueux permet à nouveau un bon fonctionnement de la machine, qui n'a été immobilisée que peu de temps. La réparation du « plug-in » peut être faite à l'extérieur.

Il est intéressant de mentionner que dans les « plug-in » et sur les tableaux il n'y a que très peu de soudures. Les fils à rassembler en un point sont placés autour d'un noyau rectangulaire et ligaturés ensemble par une

machine spéciale, qui fait un bobinage de 6 à 8 spires autour du noyau et des fils à rassembler. Par rapport à la soudure, ce système a les avantages suivants :

La qualité du joint est indépendante de l'opérateur, il n'y a pas de pièces chauffées, pas de déchets de soudure, pas de soudure froide et l'opération est faite beaucoup plus rapidement que par soudure normale. L'expérience a montré que d'une qualité excellente, le joint est encore meilleur après un certain temps, car une diffusion se fait entre les métaux.

Quant au tambour magnétique, qui est la mémoire principale de la machine, c'est dans les grandes lignes une feuille cylindrique de nickel tournant à une vitesse de 6000 tours/min et sur laquelle les cellules des mémoires sont réparties dans 256 pistes. A chacune de ces pistes correspond une tête qui permet de lire ou d'écrire dans la mémoire les informations désirées. Ces têtes sont vissées sur les petits cylindres perpendiculaires à l'axe de rotation du tambour.

Si les composantes de base de la ZEBRA, qui sont principalement des résistances, des capacités, des tubes, des diodes et des transistors, sont les mêmes que celles utilisées fréquemment dans de nombreux autres appareils, la qualité de celles-ci doit être particulièrement haute, afin qu'une sécurité de fonctionnement et qu'une durée de vie maximum soient assurées malgré les grandes exigences demandées à une calculatrice électronique moderne. Il a donc fallu adapter ces composantes aux exigences de ces machines et certaines fois ce sont le développement de nouvelles techniques et de nouvelles composantes qui ont fait faire de grands progrès au développement des calculatrices. C'est ainsi qu'on a passé des dispositifs électromécaniques aux dispositifs électriques puis aux dispositifs électroniques, beaucoup plus rapides.

### Dispositifs d'entrée et de sortie

Examinons maintenant les dispositifs d'entrée et de sortie de la ZEBRA. Ceux-ci sont montés sur le pupitre de commande. A l'entrée, la ZEBRA lit à raison de 100 caractères à la seconde les informations contenues sur une bande perforée. Cette bande, qui est identique à celle d'un téléscripteur, est préparée à l'aide d'une machine à perforer qu'on appelle en anglais un « Keyboard perforateur ». Cette machine a une certaine ressemblance avec une machine à écrire électrique, mais au lieu d'imprimer des caractères sur une feuille de papier, elle perce des trous dans une bande. Dans la bande, chaque caractère est représenté par une combinaison de cinq signes, répartis en ligne sur la largeur de la bande perforée. Ces cinq signes sont la présence ou l'absence de trous. A chaque disposition de ces trous correspond une lettre ou un chiffre bien déterminé, déjà donné dans le système binaire.

Tout le programme d'un calcul ainsi que les valeurs numériques sont perforés sur une telle bande et celle-ci est ensuite placée dans le dispositif de lecture de la machine. Les cinq signes d'un caractère, c'est-à-dire la présence ou l'absence de trous dans la bande, sont lus parallèlement et transmis à la machine sous forme de cinq courants parallèles. Celle-ci traduit automatiquement ces caractères dans son propre code.

Pour la sortie, la ZEBRA offre deux possibilités : impression des résultats par téléscripteur à raison de

sept caractères par seconde ou perforation des résultats dans une bande par perforateur rapide à raison de cinquante caractères par seconde. Dans la pratique, le dispositif de sortie, relativement lent, a déjà tendance à ralentir le processus de calcul, si bien qu'il est avantageux d'utiliser le plus rapide de ces deux dispositifs, à savoir le perforateur. Afin de tout de même pouvoir imprimer les résultats contenus dans la bande perforée, celle-ci est passée par un reproducteur qui, en connexion avec un téléscripteur, imprime ces résultats. Le téléscripteur incorporé à la machine sera de préférence utilisé uniquement pour l'impression des résultats intermédiaires d'un calcul.

### Informations préliminaires

Avant d'étudier le principe de fonctionnement de la ZEBRA, il faut dire quelques mots des informations à donner à la machine. Une calculatrice en elle-même n'est capable ni de penser ni de faire des opérations si elle n'a pas reçu au préalable certaines instructions qui lui permettent d'accepter et de comprendre les informations qui lui seront transmises. Ces instructions préliminaires sont :

1. Le « short input programme », qui permet à la machine de comprendre le code dans lequel les instructions lui sont fournies.
2. Le « normal input programme », qui permet à la machine d'accepter des instructions ou des nombres donnés dans ce code.

La machine ainsi préparée est capable d'accepter les informations qui lui sont fournies et d'effectuer les opérations commandées, mais il faut encore lui donner les instructions nécessaires pour qu'elle puisse sortir les résultats de ses calculs. Ces instructions font l'objet d'un troisième programme qu'on appelle le « normal output programme ».

Les informations d'entrée que la machine reçoit et traduit dans son propre code de travail prennent toutes la même longueur de 33 bits. L'ensemble de ces 33 bits forme ce qu'on appelle un « mot » et indique soit un « nombre » ou une « instruction ».

### Les nombres

Les nombres utilisés par la ZEBRA comprennent donc 33 bits, dont le premier indique le signe et les trente-deux autres le nombre lui-même. Exprimés dans le système binaire et en virgule fixe, ils ont la forme :

TELEPRINTER Lett. Fig.	VALUES	ZEBRA Input code
all space	0	0
E 3	1	1
line feed	2	2
A -	3	3
space	4	4
S ,	5	5
I 8	6	6
U 7	7	7
carr. ret.	8	8
D WRU	9	9
R 4	10	K
J bell	11	Q
N ,	12	.
P :	13	L
C :	14	R
K (	15	I
T 5	16	B,H
Z +	17	C,S
L )	18	D
W 2	19	E
H	20	T
Y 6	21	U
P 0	22	V
Q 1	23	N
O 9	24	A
B ?	25	X
G -	26	+
figures	27	-
M .	28	Y
X /	29	Z
V =	30	P
letters	31	corr.

$$p = \overline{p_0 p_1 p_2 \dots p_j \dots p_{32}} = -p_0 + \sum_{j=1}^{j=32} p_j \cdot 2^{-j}$$

et  $p$  en valeur absolue doit être plus petit que 1.

La valeur de  $p_0$  exprime le signe du nombre. Si celui-ci est positif,  $p_0 = 0$ , si le nombre est négatif, il est mis sous la forme

$$-p = -1 + (1 - p) \text{ et } p_0 = 1.$$

C'est ainsi que le nombre + 0,6875 sera représenté dans le code de travail de la ZEBRA par

$$+ 0,6875 = 0.1011000000000000000000000000$$

Le premier bit après le point a la valeur  $2^{-1}$ , le second la valeur  $2^{-2}$ , le troisième  $2^{-3}$ , et ainsi de suite jusqu'à  $2^{-32}$ . Les places non utilisées sont automatiquement complétées par des « zéro », pour que la longueur du « mot » soit bonne. Suivant la même convention, le nombre fractionnaire négatif aura la forme

$$- 7/8 = 1.1110000000000000000000000000$$

Les trente-deux digits permettent d'exprimer un nombre de neuf chiffres. Comme le nombre doit être mis sous une forme telle qu'il soit plus petit que 1, il pourra donc avoir neuf décimales. Il est toutefois possible d'indiquer à la machine tous les chiffres sous forme de nombres entiers et de programmer la multiplication par  $10^{-9}$  pour les ramener à la forme précédente. Il faudra donc faire attention lors de la préparation de valeurs numériques d'un calcul que les nombres soient tous exprimés avec la même position de la virgule et que les résultats intermédiaires et finals aient cette même position.

### Les instructions

Quant aux « instructions », elles possèdent également 33 bits. 15 bits sont dits « bits fonctionnels » et les 18 autres indiquent l'adresse de l'instruction (5 pour les registres, et 13 pour l'adresse des cellules dans la mémoire principale). Les bits fonctionnels sont désignés par les lettres A K Q L R I B C D E V V<sub>4</sub> V<sub>2</sub> V<sub>1</sub> W et commandent, comme nous le verrons plus loin, chacune une opération bien définie.

Nous travaillons dans le système binaire, donc chacun des bits peut prendre deux valeurs. Nous donnons la valeur 1 au bit qui est présent dans une instruction et la valeur zéro à celui qui est absent. La machine permet de combiner ces bits, si bien qu'une seule instruction peut commander plusieurs opérations différentes et simultanées. Ces quinze digits permettent par combinaison de composer  $2^{15}$  instructions différentes, mais remarquons que les combinaisons utiles et ayant un sens pratique sont beaucoup moins nombreuses.

### Le système logique de la ZEBRA

Pour comprendre l'action des différents digits, il faut d'abord décrire les quatre unités essentielles du système logique de la ZEBRA, qui sont :

- l'unité arithmétique ;
- l'unité de contrôle ;
- la mémoire principale ;
- les registres à accès immédiat.

L'unité arithmétique est composée de deux accumulateurs A et B ayant chacun un dispositif d'addition et de soustraction. C'est dans les accumulateurs que se font les opérations arithmétiques, c'est-à-dire les additions et les soustractions, seules opérations que la machine effectue directement. Toutes les autres opérations doivent être programmées pour être ramenées à ces deux opérations fondamentales. Toutefois, les instructions de ces sous-programmes sont contenues en permanence dans la machine et on les appelle les « sub-routines ». A titre d'exemples, citons les quelques sous-programmes suivants : extraction d'une racine carrée, calcul d'une exponentielle, calcul d'un logarithme naturel ou d'un logarithme décimal, calcul des sinus et des cosinus, etc.

Chacun des accumulateurs a une capacité de 33 bits, c'est-à-dire place pour 1 mot. En cas de besoin, les deux accumulateurs peuvent être combinés pour n'en former qu'un seul d'une capacité de 66 bits. Le fait d'avoir cette possibilité permet de faire une multiplication de deux nombres d'une longueur normale de 33 bits. La combinaison de ces deux accumulateurs permet également le débordement du contenu d'un des accumulateurs dans l'autre, au cas où la capacité d'un seul n'est plus suffisante. De plus, il est possible de faire une translation vers la gauche ou vers la droite du contenu de chacun des accumulateurs séparément ou du contenu du grand accumulateur. Une telle translation revient, dans le système binaire, à une multiplication ou à une division de ce contenu par deux.

La mémoire principale est un tambour magnétique, d'une capacité de 8192 mots répartis sur 256 pistes ayant place chacune pour 32 nombres ou instructions. Ce tambour tourne à une vitesse de 6000 t/min. Pendant le temps d'une révolution, qui est de 10 ms, la machine peut lire ou écrire 32 mots dans la mémoire, ce qui donne un temps de base par opération de 312 µs. Les cellules du tambour sont désignées par les adresses 0 à 8191. Quant à la répartition de l'occupation de ces cellules, elle est la suivante : d'une part, la machine doit contenir trois programmes pour être capable de recevoir et de comprendre les informations qui lui sont données et de sortir les résultats de ces calculs et que, d'autre part, toutes les opérations à part l'addition et la soustraction doivent être programmées et que les instructions des « sub-routines » sont contenues dans la machine, c'est-à-dire dans le tambour. Nous verrons plus loin que les informations données à la machine peuvent être exprimées dans deux codes différents, le « code normal », qui est le code d'entrée normal de la machine, et un code simplifié dit « code simple ». La traduction du code simple nécessite un programme d'interprétation qui peut également être contenu dans le tambour.

Les instructions de ces différents programmes occupent au total 2148 cellules, si bien que la mémoire

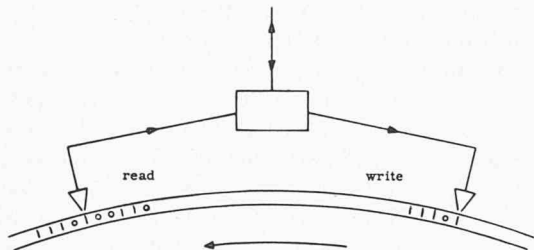
Function Digits															Register Address			Main Store Address																			
A	K	Q	L	R	I	B	C	D	E	V	V <sub>4</sub>	V <sub>2</sub>	V <sub>1</sub>	W																							

Place des bits dans une « instruction ».

ne contient plus que 6044 cellules réellement disponibles pour les calculs. Toutefois, si on renonce aux programmes établis en code simple ou à certaines « sub-routines », cette capacité peut être sensiblement augmentée.

Les registres sont au nombre de 32, numérotés de 0 à 31. Douze d'entre eux sont des mémoires possédant les mêmes caractéristiques que celles du tambour magnétique ; du reste, elles sont placées également sur celui-ci, mais elles sont d'un accès immédiat. De telles mémoires sont nécessaires pour raccourcir le temps de calcul. En effet, pour pouvoir lire ou écrire dans une cellule déterminée du tambour, il faut attendre que lors de la rotation du tambour la cellule passe devant la tête de lecture, ce qui en moyenne demande environ 5 ms. Ce temps d'accès peut être réduit sensiblement par une répartition judicieuse des adresses et on fait alors ce qu'on appelle en anglais « l'optimum programming ». Les douze mémoires immédiates sont chacune sur une piste séparée du tambour et le principe de ces mémoires est le suivant.

Tout d'abord, on a deux têtes au lieu d'une (une



Principe des mémoires à accès immédiat.

pour la lecture et l'autre pour écrire) et l'espace entre la tête de lecture et le dispositif pour écrire dans la mémoire a exactement la longueur d'un mot, c'est-à-dire 33 bits. Dans ces conditions, lors de la rotation du tambour, le bit lu est, d'une part, transmis à l'unité arithmétique ou à l'unité de contrôle et, d'autre part, écrit à nouveau sur la piste, si bien qu'il est de nouveau à disposition.

Parmi les vingt registres restants, signalons encore le contenu ou la signification de certains d'entre eux. Trois d'entre eux contiennent des constantes qui sont 0, -1,  $2^{-32}$ . Deux des registres sont les accumulateurs A et B et les autres sont utilisés pour les dispositifs d'entrée et de sortie.

La dernière des unités du système logique de la ZEBRA qu'il reste à examiner est l'unité de contrôle. De façon générale, l'unité arithmétique reçoit et travaille avec des nombres, tandis que l'unité de contrôle reçoit des instructions. C'est cette unité qui fait exécuter les différentes instructions et assure leur exécution dans le bon ordre et au bon moment. Cette unité contient entre autres trois registres (qui ne sont pas à confondre avec les précédents). Dans l'un, l'instruction en exécution est conservée, tandis que dans les deux autres les prochaines instructions à exécuter sont mises en attente.

Maintenant que nous connaissons la nature des instructions et dans les grandes lignes le rôle des différentes unités du système logique de la ZEBRA, nous pouvons examiner la fonction individuelle de certains digits qui sont à la base des instructions données en code normal.

### Instructions en code normal

Rappelons les 15 digits fonctionnels. Ce sont A K Q L R I B C D E V  $V_4$   $V_2$   $V_1$  W, et donnons quelques règles générales pour l'établissement des programmes dans le code normal de la Zebra. Plusieurs digits peuvent être combinés dans une même instruction, ce qui va permettre à la machine d'exécuter plusieurs opérations simultanément. Si le digit est présent dans l'instruction, la fonction correspondante est exécutée, mais si une fonction ne doit pas être exécutée, il n'est pas nécessaire d'écrire 0 ou de donner un digit particulier, sauf pour le digit A. Quand celui-ci n'est pas présent, on écrit X ; donc

X = pas de digit A.

Chaque instruction va donc commencer soit par le digit A, soit par le digit X. La présence de l'un de ces digits au début d'un mot indique à la machine qu'elle va lire une instruction et non un nombre. De même, si au début d'un mot la machine lit le signe + ou -, elle sait qu'elle va lire un nombre et en même temps elle prend note de son signe. Les digits A et K déterminent les connexions entre les unités logiques de la machine en donnant la position des clés des commutateurs portant les mêmes lettres que les digits. (Dans le dessin, la position des clés correspond à la valeur zéro du bit.)

#### Digit A

Si A est présent, il y a connexion entre le tambour magnétique et l'unité arithmétique.

Si A est absent (digit X), il y a connexion entre le tambour magnétique et l'unité de contrôle.

#### Digit K

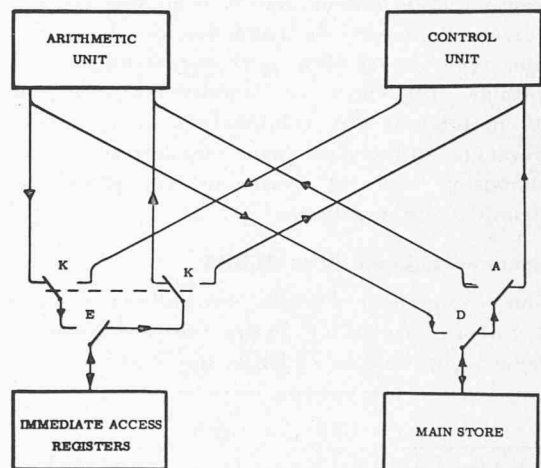
Si K est présent, il y a connexion entre les registres et l'unité de contrôle.

Si K est absent, il y a connexion entre les registres et l'unité arithmétique.

#### Digits E et D

Ces digits déterminent si la machine lit ou écrit dans les registres à accès immédiat ou dans la mémoire principale.

Les digits A et K combinés offrent quatre possibilités qui donnent quatre opérations de base pouvant être



Système logique de la ZEBRA.

effectuées par la machine. Comme indiqué plus haut, en désignant par 1 la présence d'un digit et par 0 son absence, il y a les combinaisons suivantes :

Digit A	Digit K	Opérations
0	0	adding jump
0	1	double jump
1	0	double addition
1	1	jumping addition

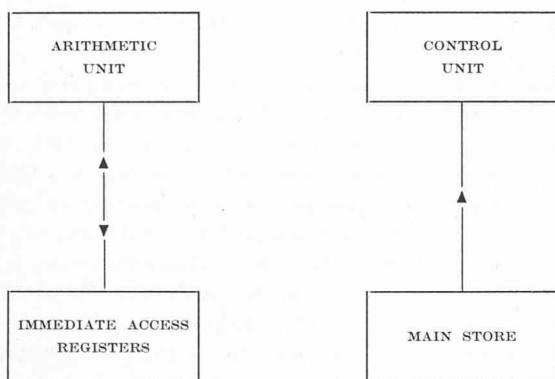
Le terme « addition » se réfère dans cette description aussi bien à l'opération d'addition qu'à celle de la soustraction et le terme « jump » se réfère aux choix de l'instruction suivante.

### Adding jump

A = 0, K = 0, supposons D = 0.

Il y a, comme le montre le schéma simplifié du système logique, d'une part, connexion entre les registres et l'unité arithmétique et, d'autre part, entre la mémoire principale et l'unité de contrôle.

- Le contenu d'un des registres est transmis à l'un des accumulateurs, c'est-à-dire additionné au contenu de ce dernier, ceci si E = 0, tandis que si E = 1, c'est le contenu d'un des accumulateurs qui est transmis à l'un des registres.
- L'instruction suivante est lue dans le tambour magnétique et transmise à l'unité de contrôle.



Adding jump.

### Exemple d'instruction

X 100 . 12

Il n'y a pas de A (X = pas de A) et pas de K, c'est-à-dire qu'on a une opération du type « adding jump ». En prenant les symboles suivants :

( ) = contenu  
A = accumulateur A  
12 = registre n° 12

l'opération peut s'écrire

$$(A) + (12) \rightarrow A$$

$$(100) \rightarrow \text{U.C.}$$

ce qui veut dire : additionner au contenu de l'accumulateur A le contenu du registre n° 12 et prendre l'instruction suivante dans la cellule n° 100 du tambour magnétique.

### Double jump

A = 0, K = 1, supposons D = 0, E = 0.

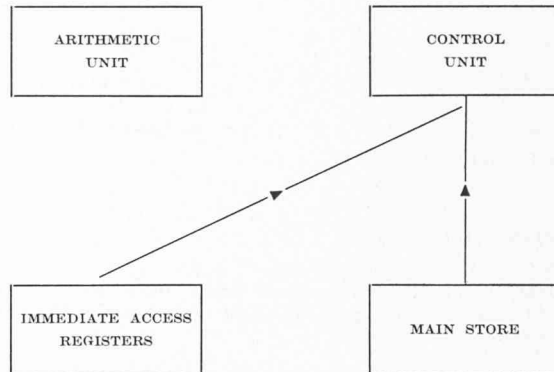
Dans ce cas, les registres et le tambour magnétique sont tous deux reliés à l'unité de contrôle. Il s'agit donc de l'addition de deux instructions transmises à l'unité de contrôle. Une telle opération conduit à la modification d'une instruction par une autre pour obtenir une nouvelle instruction.

Exemple : X 125 K 13

L'opération s'exprime par :

$$(125) + (13) \rightarrow \text{U.C.}$$

et s'énonce : modifier l'instruction contenue dans la cellule n° 125 du tambour par l'instruction contenue dans le registre n° 13 et transmettre cette nouvelle instruction à l'unité de contrôle.



Double jump.

### Double addition

A = 1, K = 0.

Cette opération est symétrique à la précédente. Les registres et la mémoire principale sont tous deux reliés à l'unité arithmétique. Quatre combinaisons sont possibles suivant la présence ou l'absence des digits D et E.

Si E = 0 D = 0.

Un nombre des registres et un nombre de la mémoire principale sont additionnés et leur total est ajouté au contenu de l'accumulateur.

Exemple : A 320 . 11

L'opération s'écrit

$$(A) + (320) + (11) \rightarrow A$$

et s'énonce : ajouter au contenu de l'accumulateur A le contenu de la cellule 320 et le contenu du registre 11.

Si E = 1 D = 1.

Le contenu d'un accumulateur est écrit à la fois dans un des registres et dans une des cellules du tambour.

Exemple : A 100 D E 12

L'opération s'écrit

$$(A) \rightarrow A$$

$$(A) \rightarrow 100$$

$$(A) \rightarrow 12$$

et s'énonce : le contenu de l'accumulateur A reste dans A. Le contenu de l'accumulateur A est écrit à la fois dans la cellule 100 du tambour et dans le registre n° 12.

Il y a encore deux combinaisons : E = 1 D = 0 et E = 0 D = 1.

Exemple d'instruction de la première de ces combinaisons :

A 1767 E 12

L'opération s'écrit

$$(A) \rightarrow 12$$

$$(A) + (1767) \rightarrow A$$

et s'énonce : le contenu de l'accumulateur A est écrit dans le registre n° 12 et le contenu de la cellule n° 1767 est additionné au contenu de l'accumulateur A.

Exemple d'instruction de la deuxième combinaison :

A 1000 D 11

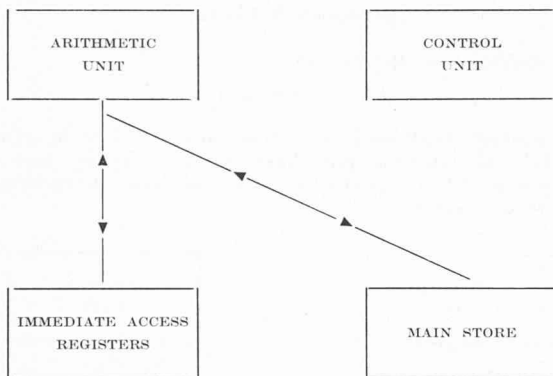
L'opération s'écrit

$$(A) \rightarrow 1000$$

$$(A) + (11) \rightarrow A$$

et s'énonce : copier le contenu de l'accumulateur A dans la cellule n° 1000 et ajouter le contenu du registre n° 12 au contenu de l'accumulateur A.





Double addition.

### Jumping Addition

$A = 1$   $K = 1$  et supposons  $E = 0$  et  $D = 0$ .

Dans ce cas, il y a connexion entre l'unité arithmétique et le tambour, d'une part, et entre les registres et l'unité de contrôle, d'autre part.

Une instruction du registre est transmise à l'unité de contrôle pour modifier l'instruction déjà contenue dans celui-ci et en même temps le contenu d'une cellule du tambour est additionné au contenu de l'accumulateur A.

Exemple :  $A\ 100\ K\ 12$   
L'opération s'écrit  $(12) + (D) \rightarrow U.C.$   
 $(A) + (100) \rightarrow A$

Ainsi, nous avons la signification des digits A K D E ; quant aux autres digits Q L R I B C V, ils agissent tous sur les opérations de l'unité arithmétique.

*Digit B* : Si B est absent dans l'instruction, les opérations se font dans l'accumulateur A. Si B est présent, elles se font dans l'accumulateur B.

Les exemples précédents n'ont porté que sur des additions mais il est clair que la machine effectue également les soustractions.

Si le digit I est absent, les opérations sont des additions.

Si le digit I est présent, elles sont des soustractions.

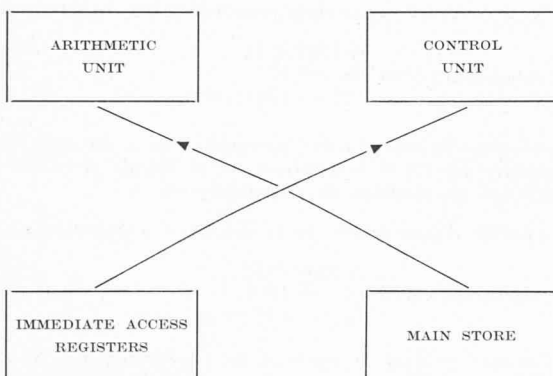
Exemple :  $X\ 100\ B\ I5$

$A = 0$ ,  $K = 0$ , c'est donc une opération du type « adding jump ». E et D sont également absents.

La signification d'une telle instruction est :

$(100) \rightarrow U.C.$   
 $(B) - (5) \rightarrow B$

c'est-à-dire, prendre l'instruction suivante dans la cellule n° 100. Soustraire le contenu du registre n° 5 du contenu de l'accumulateur B.



Jumping addition.

Le digit C indique que l'accumulateur spécifié dans l'instruction doit être vidé ( $C = \text{clear}$ ). Si toutefois l'instruction mentionne que le contenu de l'accumulateur doit être copié dans un registre ou dans une cellule de la mémoire, cette dernière opération est faite avant la mise à zéro de l'accumulateur.

Exemple :  $A\ 100\ B\ C\ D\ E\ 12$   
signifie  $(B) \rightarrow 100$   
 $(B) \rightarrow 12$   
zéro  $\rightarrow B$

Dans la description de l'unité arithmétique, il a été mentionné qu'il était possible de faire une translation du contenu des accumulateurs vers la gauche ou vers la droite et que celle-ci correspondait dans le système binaire à une multiplication ou à une division par 2 de ce contenu. Une translation vers la gauche est indiquée par le digit L (left) et une translation vers la droite par le digit R. L'opération d'addition ou de soustraction, qui est commandée dans une instruction contenant L ou R, se fait toujours sur le contenu de l'accumulateur après la translation.

Quant au digit Q, lorsqu'il est présent, 1 est ajouté ou soustrait (ceci dépend du digit I) au dernier bit du contenu de l'accumulateur B, et ceci quelle que soit l'opération arithmétique effectuée au même moment. Ce digit Q sera utilisé certaines fois pour des opérations de comptage.

Les derniers digits qui restent à examiner sont les digits V. Ceux-ci sont utilisés lorsque des fonctions, opérations ou instructions ne sont à effectuer que lorsque certaines conditions sont remplies. Ces digits ordonnent à la machine des tests qui portent en particulier sur le signe du contenu des accumulateurs, sur la valeur du dernier digit de l'accumulateur B, si le contenu d'un accumulateur est nul ou sur la position de certains commutateurs. Chaque instruction contenant un digit V est dite instruction test et les opérations indiquées par l'instruction ne sont effectuées que si les conditions prévues par le test sont remplies.

Exemple d'une telle instruction test :

$A\ 100\ I\ 5\ V_2$ .

Par définition, la signification de  $V_2$  est la suivante : Si le contenu de l'accumulateur B est nul ou positif, les opérations indiquées par l'instruction ne sont pas effectuées, mais elles ne le sont que si le contenu de B est plus petit que zéro.

On peut donc écrire :

Si  $B \geq 0$  l'opération est ignorée.  
Si  $B < 0$   $(A) - [(100) + (5)] \rightarrow A$

ce qui s'énonce : du contenu de l'accumulateur A, soustraire le contenu de la cellule 100 du tambour et le contenu du registre n° 5.

Ces exemples montrent les connexions entre les différentes unités du système logique et, dans les grandes lignes, le genre d'opérations effectuées par la ZEBRA. Il est à noter que la forme des instructions et la logique simple et souple de la machine permettent d'effectuer plusieurs opérations simultanées, ce qui raccourcit sensiblement les programmes et les temps de calcul.

Les temps de calcul pour les quelques opérations suivantes sont :

Addition et la soustraction :	312 $\mu$ s
Multiplication :	11 ms
Division :	25 ms
Extraction d'une racine carrée :	80 ms
Calcul d'un sinus ou d'un cosinus :	80 ms

Ces temps ne comprennent pas le temps d'accès moyen au tambour, qui est, comme déjà vu, de 5 ms environ.

### Instructions en code simple

La programmation d'un calcul en code normal est une opération assez longue et délicate, qui demande de la part de l'utilisateur une connaissance approfondie des possibilités de la machine et surtout une grande habileté s'il veut utiliser la machine au maximum de ses possibilités. Cependant, si un problème ne doit être résolu qu'une seule fois, si le programme doit être établi rapidement et facilement ou si le programmeur n'a pas une grande expérience, le programme peut être établi dans un code simple facile à préparer. Lorsqu'on travaille en code simple, on peut oublier tout ce qui concerne le code normal, la machine faisant elle-même la traduction du code simple dans son propre code. Ce programme d'interprétation et ses instructions sont contenus dans la mémoire de la machine. Cette interprétation va prendre un certain temps, si bien que les temps pour les calculs programmés en code simple seront à peu près cinq fois plus longs que pour ceux programmés en code normal.

En code simple, les nombres ne sont plus exprimés avec virgule fixe mais avec virgule flottante, c'est-à-dire mis sous la forme

$$a \cdot 10^b$$

où  $a$  en valeur absolue doit être plus grand que 0,1 et plus petit ou égal à 1.  $b$  sera compris entre -900 et +900.

Chaque nombre est donc exprimé par les deux nombres  $a$  et  $b$ , c'est-à-dire qu'il utilisera deux cellules de la mémoire. Chaque instruction du code simple nécessite également deux cellules, si bien que la capacité d'informations du tambour devient deux fois plus petite qu'en code normal et elle ne contient pratiquement plus que 3000 adresses, dont en principe 1500 sont réservées pour les instructions et 1500 pour les nombres. Par convention, les adresses des nombres sont indiquées par  $n$  et celles des instructions par  $p$ .

Voici quelques exemples d'instructions arithmétiques en code simple :

Instructions	Significations	Interprétations	Temps en ms
An	additionner	(A) + (n) $\rightarrow$ A	40
Sn	soustraire	(A) - (n) $\rightarrow$ A	40
Hn	prendre dans	(n) $\rightarrow$ A	20
Tn	transférer et annuler	(A) $\rightarrow$ n 0 $\rightarrow$ A	20
Un	transférer	(A) $\rightarrow$ n	20
Dn	diviser	(D) / (n) $\rightarrow$ A	55
Vn	multiplier	(A) . (n) $\rightarrow$ A	35

### Exemple :

Il s'agit de programmer le calcul suivant :

$$r = \sqrt{x^2 + y^2}$$

La valeur de  $x$  est supposée contenue dans la cellule n° 1 et celle de  $y$  contenue dans la cellule n° 2.

Plan du programme	Signification	Instruction	Temps ms
Start	Début du calcul	Y	
$x \rightarrow A$	Introduire la valeur de $x$ (cellule 1) dans l'accumulateur A	H 1	20
$x^2 \rightarrow A$	Multiplier le contenu de l'accumulateur A par le contenu de la cellule n° 1	V 1	35
$x^2 \rightarrow 3$ $0 \rightarrow A$	Introduire le contenu de l'accumulateur A dans la cellule 3. Vider l'accumulateur A	T 3	20
$y \rightarrow A$	Introduire la valeur de $y$ (cellule 2) dans l'accumulateur A	H 2	20
$y^2 \rightarrow A$	Multiplier le contenu de l'accumulateur A par le contenu de la cellule n° 2	V 2	35
$x^2 + y^2 \rightarrow A$	Additionner le contenu de la cellule 3 au contenu de l'accumulateur A	A 3	40
$r = \sqrt{x^2 + y^2} \rightarrow A$	La subroutine Z1 indique qu'il faut extraire la racine carrée du contenu de l'accumulateur A	Z 1	100
$r \rightarrow 3$ $0 \rightarrow A$	Introduire le contenu de l'accumulateur A dans la cellule n° 3	T 3	20
Stop	Fin du calcul	Z	

Le temps total pour ce calcul est de 290 ms.

C'est l'exemple d'un petit programme établi en code simple qui montre avec quelle facilité et rapidité celui-ci peut être préparé.

Cet exemple montre que le programme s'établit suivant un processus analogue à la suite des calculs effectués normalement avec un papier et un crayon et qu'il n'est pas nécessaire d'avoir une formation spéciale pour programmer des calculs dans ce code, mais qu'il suffit de connaître la liste des instructions. Quelques heures d'explications à un débutant sont suffisantes pour lui permettre de préparer les programmes dans ce code simple. Si le temps de calcul est plus long que lorsque le programme est établi en code normal, la préparation du programme en code simple est par contre beaucoup plus rapide, ce qui fait que la rentabilité pour la résolution des problèmes par programme en code simple est tout de même bonne.

L'idée directrice suivie dans le développement de la Zebra a été d'obtenir une calculatrice rapide à grande mémoire, d'une logique simple, capable de traiter et de résoudre des problèmes mathématiques, scientifiques et techniques de tout genre.