Aucun autre cas ne pouvant se présenter, le théorème est établi.

Il admêt, entre autres conséquences, celle-ci :

Corollaire. $E_1$ *et* $E_2$ *sont deux corps convexes à points intérieurs dont l'un* $E_1$ , *par exemple, contient l'autre* $E_2$. *La condition nécessaire et suffisante pour que* $\overline{E_1 - E_2}$ *soit aussi un corps convexe est que le produit de fermeture* $E_1 \cdot (\overline{E_1 - E_2})$ *soit un corps convexe.*

1.°) La condition est évidemment nécessaire.

En effet, supposons $\overline{E_1 - E_2}$ conveve. Son produit avec $E_2$, $E_2 \cdot (\overline{E_1 - E_2})$ est convexe.

2.°) La condition est suffisante.

En effet, supposons que $E_2 \cdot (\overline{E_1 - E_2})$ soit une figure convexe, évidemment sans points intérieurs.

Dans ce cas, la somme $E_2 \dot{+} (\overline{E_1 - E_2}) = E_1$ étant convexe ainsi que le produit $E_2 \cdot (\overline{E_1 - E_2})$, il résulte du théorème précédent que $E_1$ et $\overline{E_1 - E_2}$ sont deux corps convexes.

Comme je l'ai indiqué au début de ce paragraphe il ne faut pas s'étonner du résultat précédent ni de sa simplicité, car les ensembles connexes, localement connexes continus jouissent d'une propriété analogue ([1]).

Je terminerai cet exposé par une question : «Est-il possible caractériser *logiquement* une propriété qui, appartenant à la réunion et à l'intersection de deux ensembles, appartient, de ce fait, à chacun de deux ensembles ?

---

([1]) — I. Soient $A$ et $B$ deux ensembles fermés (ou deux ensembles ouverts). Si les ensembles $A \dot{+} B$ et $A \cdot B$ sont connexes, les ensembles $A$ et $B$ le sont aussi. Voir une note de S. Janizewski et C. Kuratowski. Fundamenta Mathematicae. Tome I (1920) Nouvelle Ed. 1937, P. 211, th. I.

Voir aussi C. Kuratowski, Topologie. Vol. II, Chap. V Par 41, II, p. 83.

II. $A$ et $B$ étant deux ensembles compacts tels que $A \dot{+} B$ et $A \cdot B$ soient des continus, $A$ et $B$ sont aussi des continus. (C. Kuratowski. Topologie, Vol. II, Chap. V, par. 42, p. 1[illegible]8).

III. $A$ et $B$ étant deux ensembles fermés tels que les ensembles $A \dot{+} B$ et $A \cdot B$ soient localement connexes; les ensembles $A$ et $B$ sont aussi localement connexes. (C. Kuratowski. Topologie. Vol. II, Chap. VI. Parag. 44 ; II, 10, p. 164).

# Princípios fundamentais dos computadores digitais automáticos

por A. César de Freitas

(Conclusão)

## 4. Memória

A parte mais importante dum computador digital automático é, talvez, a sua memória, e a eficiência da máquina depende em grande parte da quantidade de informação que ela pode memorizar. Com efeito, como nos modernos computadores automáticos em geral há a necessidade de colocar na memória, logo de início, todos os números e instruções que conduzem à resolução dum problema de cálculo, se a capacidade dessa memória é pequena, a ordem dos problemas que podem ser tratados pela máquina é bastante limitada.

Toda a memória deve ter as três propriedades seguintes :

(1) Deve ser capaz de reter a informação durante o tempo que for necessário.

(2) Deve ser capaz de fornecer essa informação quando pedida.

(3) Deve permitir substituição de informação.

Além disso o tempo que a máquina leva a colocar e a extrair informação da memória — *tempo de acesso* — deve estar de acordo com o tempo que ela demora a efectuar as outras operações — a máquina não deve estar à espera para obter e pôr informação na memória.

Uma memória cujo tempo de acesso seja muito pequeno é sempre muito cara, e, principalmente por isso, numa grande maioria dos computadores digitais actualmente existentes aparecem dois tipos de memória: uma, de menor capacidade e cujo tempo de acesso é pequeno — *memória rápida* —; outra, de grande capacidade — *a memória principal* — onde é colocada a quáse totalidade da informação a fornecer à máquina. Esta informação é transferida (ou copiada), em pequenos blocos, para a memória rápida onde é tratada pela máquina. Consegue-se assim um tempo médio de acesso à informação relativamente pequeno, sem encarecer demasiadamente o custo da máquina.

Há uma grande variedade de materiais que, pelas suas propriedades físicas (ou químicas), podem ser explorados para a construção de memórias. Assim para memorizar informação no sistema binário, qualquer dispositivo onde se possam distinguir dois estados não simultâneos, serve para representar um dígito. É o que acontece, por exemplo, com o circuito bi-estável a que fizemos referência no parágrafo anterior.

Pode mesmo dizer-se, sem grande exagero, que não há nenhum sector da física onde não se possa descobrir algo que sirva de base à construção duma memória.

Actualmente os materiais magnéticos são os mais usados, quer pela sua durabilidade e economia, quer porque, em certas condições, permitem tempos de acesso bastante pequenos. São muito usuais, como memórias, os tambores e discos magnéticos, os fios e as fitas magnéticas, as matrizes de aneis magnéticos. Nestes dispositivos usam-se dois estados de magnetização para reter informação na forma binária.

Vamo-nos referir, porém, mais detalhadamente ao primeiro tipo de memória de grande capacidade que foi usado, e que se encontra em muitos computadores actualmente em funcionamento. É o tipo de memória formada pelos chamados *circuitos de atraso acústicos*. Cada elemento desta memória é constituído como está representado esquemáticamente na figura seguinte

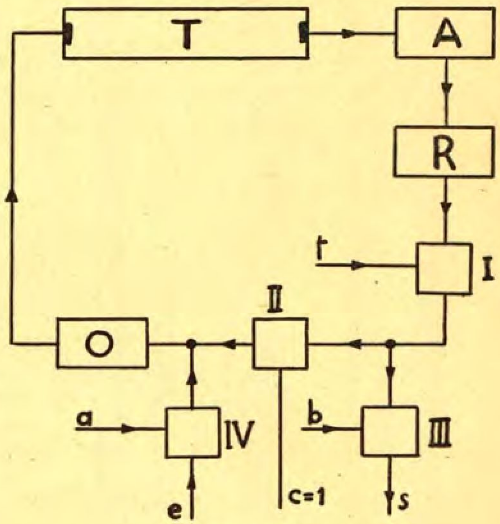


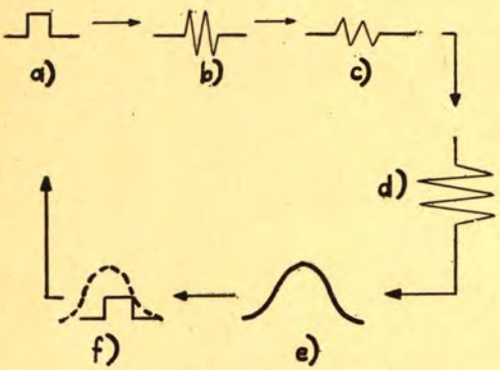

Fig. 10

$T$ — tubo cheio de mercúrio tendo interiormente, em cada uma das extremidades, um cristal de quartzo
$A$ — amplificador
$R$ — rectificador
$O$ — oscilador
$I$, $II$, $III$, $IV$ — circuitos de coincidência

Ao terminal $t$ chegam os impulsos dados pelo padrão de tempo, em geral um vibrador de quartzo.

Em $T$, no cristal de quartzo da esquerda, a energia eléctrica é transformada em energia acústica (ultra-sons) que viaja no mercúrio e é transformada novamente em energia eléctrica no cristal de quartzo da direita.

No dispositivo da figura anterior, um impulso que entrou no circuito sofre o ciclo de transformações a seguir esquematizado.




*a)* — à entrada do oscilador
*b)* — à saída do oscilador
*c)* — à saída do tubo de mercúrio
*d)* — à saída do amplificador
*e)* — à saída do rectificador
*f)* — o impulso inicial é reproduzido no circuito de coincidência $I$ com o auxílio dos impulsos do padrão de tempo. Para isso é necessária uma certa sincronização para que *e)* chegue a $I$ (Fig. 10) ao mesmo tempo que um impulso do padrão de tempo. Tal sincronização é obtida entrando em linha de conta com a velocidade de propagação do som no mercúrio e com o comprimento do tubo $T$.

A passagem de *a)* para *b)* faz-se com o fim de obter uma melhor propagação da energia no tubo $T$.

O impulso mantém-se, portanto, em circulação constante.

Se se tratasse dum grupo de impulsos (correspondente a um número ou a uma ordem) tudo se passaria da mesma maneira. Esse grupo de impulsos chegaria a $e$ (Fig. 10) e entraria em circulação desde que se tivesse $a=1$ durante o intervalo correspondente à passagem do grupo. Se quizessemos depois obter esse mesmo grupo de impulsos bastaria fazer $b=1$, em III, um pouco antes de ele chegar ao outro terminal de entrada e manter esse valor de $b$ até que o grupo fosse obtido em $s$. O circuito de coincidência II serve para «apagar» informação que esteja a circular.

Já se deixa ver que numa máquina que use este tipo de memória, deve haver uma sincronização que permita efectuar as operações anteriores, tanto mais que no mesmo circuito de atraso circulam em geral vários grupos de impulsos correspondentes a outros tantos números (ou ordens).

O nome de circuito de atraso que se dá ao circuito representado na Fig. 10 resulta do facto da propagação da energia eléctrica no circuito ser atrasada em virtude das transformações sofridas em $T$ (a velocidade do som no mercúrio é da ordem de 1.5 $mm/\mu s$). A amplitude do atraso é função do comprimento do tubo.

Note-se que no tubo $T$ podem ser usados outros líquidos além do mercúrio. Também se usam as propriedades magneto-estritivas de certos metais como o níquel, para a construção de circuitos de atraso acústicos.

É interessante referir que certas teorias modernas sobre a memória humana sugerem que tudo se deve passar de modo análogo ao que temos referido para os circuitos de atraso — grupos de impulsos nervosos circulam constantemente, possivelmente em vários circuitos em paralelo. O processo da aquisição de conhecimentos envolve, quer o estabelecimento de circuitos fechados onde circulam os grupos de impulsos de energia nervosa, quer a formação inicial desses grupos de impulsos.

## 5. Adição e subtracção

Por simplicidade e para não nos alongarmos demasiadamente, passaremos, de agora em diante, a considerar uma máquina com as características seguintes

(1) Tipo série, trabalhando com números $x$ tais que $-1 \leqq x < 1$

(2) Os números negativos são representados pelos complementos, para dois, dos seus módulos

(3) Cada número é representado por dezasseis dígitos binários, o primeiro sendo o dígito do sinal. Assim o número 0,1 (base dez) será representado por 0,000110011001101 e o número $-0{,}628$ por 1,011000000000000

(4) Possui um acumulador, isto é, um dispositivo onde se obtém o resultado das operações antes de o transferir para a memória e onde se pode colocar qualquer número vindo da memória.

Por se tratar duma máquina do tipo série todo o seu funcionamento é controlado, no tempo, por um padrão de tempo.

Vejamos então como nesta máquina se faz a adição e a subtracção.

a) *Adição.*

Para somar dois números basta fazê-los circular simultâneamente através de dois dos terminais de entrada de um somador e ligar o terminal de saída, $T$, correspondente ao transporte, ao outro terminal de entrada através dum circuito de atraso que dê um atraso de um impulso [1].

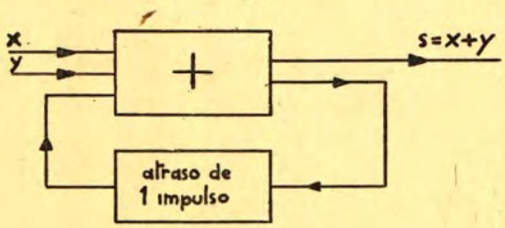



Fig. 11

[1] É daqui que vem a designação de somador dada ao circuito ($f$) do parágrafo 3.

Para verificar que de facto em $S$ se obtém a soma $X+Y$ basta ter em atenção as tabelas I e II do parágrafo 1 e as tabelas III e IV do parágrafo 2. Note-se que os números circulam de modo que os algarismos menos significativos se apresentam sempre primeiro.

b) *Subtracção.*

Neste caso pode usar-se um somador como para a adição, desde que o diminuidor seja complementado antes de atingir o seu terminal de entrada. Um circuito que serve para achar o complemento dum número é o seguinte

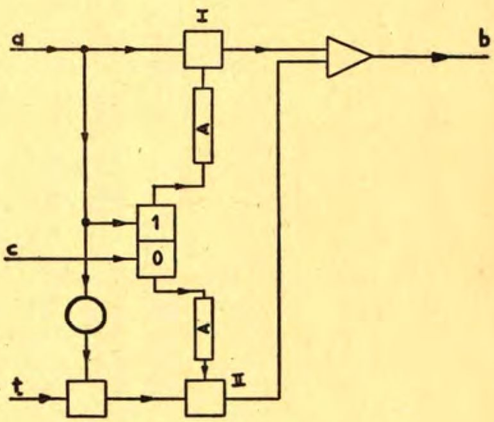



Fig. 12

$A$ — Circuito de atraso.
$t$ — Terminal de entrada de impulsos dados pelo padrão de tempo.

Suponhamos o circuito bi-estável no estado indicado na figura. Quando um número se apresenta no terminal de entrada $a$, se o seu primeiro algarismo da direita é *0* esse algarismo é reproduzido no terminal de saída $b$ e vai acontecendo o mesmo até que apareça o algarismo *1* no terminal de entrada. Este *1* passa em I mas ao mesmo tempo muda o estado do circuito bi-estável o que interrompe a passagem em I a qualquer outro *1* que se apresente. É fácil agora reconhecer que, para os restantes algarismos do número, onde está *1* fica *0* e onde está *0* fica *1*, devido à acção do inversor e porque em II se tem agora sempre *1* no terminal de entrada que vem do circuito bi-estável [1].

Note-se que é possível construir um circuito — *um subtractor* — capaz de fazer a subtracção directamente.

Uma máquina com dispositivos para fazer a adição e a subtracção está apta a efectuar também a multiplicação e a divisão [2] já que estas operações não são mais do que determinadas sequências das outras, mas é rara a máquina que não possui ainda um dispositivo para fazer a multiplicação directamente. Máquinas que façam directamente a divisão não são tão frequentes.

## 6. Representação e execução das ordens (instruções)

A máquina usa as instruções em código. Vamos supor que se trata dum *código de direcção simples*, isto é, cada ordem refere-se a um único compartimento da memória (quando se trate de ordens em que ela intervém). Para usar um código deste tipo a unidade aritmética da máquina deve possuir um acumulador — é o caso da máquina que estamos a considerar.

Cada ordem é representada por dezasseis dígitos binários, tal como os números. Desses dígitos alguns referem-se ao tipo de ordem (some, multiplique, copie, etc.) — é a *parte funcional* — e os restantes indicam o compartimento da memória visado por tal ordem (quando se trata duma ordem que faz intervir a memória) — é a *direcção* —.

Suporemos que, dos dezasseis dígitos binários, cinco se referem à parte funcional e os restantes aos compartimentos da memória

$$\underbrace{\boxed{d_{16}\,|\,d_{15}\,|\,d_{14}\,|\,d_{13}\,|\,d_{12}}}_{\text{função}}\underbrace{\boxed{d_{11}\,|\,d_{10}\,|\,d_{9}\,|\,d_{8}\,|\,d_{7}\,|\,d_{6}\,|\,d_{5}\,|\,d_{4}\,|\,d_{3}\,|\,d_{2}\,|\,d_{1}}}_{\text{direcção}}$$

Estamos portanto a supor a possibilidade de $32 = 2^5$ ordens e que a memória tem $2048 = 2^{11}$ compartimentos.

Passaremos a designar por $C(n)$ o conteudo do compartimento $n$ da memória, por $C(Ac)$ o conteúdo do acumulador, por $C(M)$ o conteúdo do multiplicador [1] e suporemos desde já que o código de ordens da máquina inclui as ordens seguintes

1) $A$ $n$ — adicione $C(n)$ a $C(Ac)$ colocando o resultado no acumulador;

2) $S$ $n$ — subtraia $C(n)$ de $C(Ac)$ colocando o resultado no acumulador;

3) $T$ $n$ — transfira $C(Ac)$ para o compartimento $n$ da memória (deixando limpo o acumulador);

4) $U$ $n$ — copie $C(Ac)$ no compartimento $n$ da memória;

5) $H$ $n$ — substitua $C(M)$ por $C(n)$;

6) $V$ $n$ — multiplique $C(n)$ por $C(M)$ e some o resultado a $C(Ac)$;

7) $F$ $n$ — tome $C(n)$ como a próxima ordem a ser executada [2];

8) $G$ $n$ — se $C(Ac) < 0$ tome $C(n)$ como a próxima ordem; caso contrário proceda normalmente [2];

9) $E$ $n$ — se $C(Ac) \geqq 0$ tome $C(n)$ como a próxima ordem; caso contrário proceda normalmente [2];

---

[1] Antes de achar o complemento de outro número o circuito bi-estável deve mudar de estado.

[2] Observe-se que o essencial na máquina é possuir um dispositivo apenas para fazer a subtracção.

[1] O multiplicador é um registo, semelhante a um compartimento da memória, onde se coloca um dos factores quando se pretende fazer uma multiplicação.

[2] No parágrafo seguinte perceberemos melhor o significado desta ordem.

10) $R$ $p$ — divida $C(Ac)$ por $2^p$;
11) $L$ $p$ — multiplique $C(Ac)$ por $2^p$;
12) $Z$ — pare.

Considere-se, por exemplo, a ordem *A 21* que significa, adicione o conteudo do compartimento *21* da memória ao conteúdo do acumulador. Se a função adicionar — $A$ — por representada pelo grupo de digitos *11100,* a ordem anterior é representada por (1) *1110000000010101*.

Vejamos agora, em linhas muito gerais, como é executada uma determinada ordem onde intervenha o conteúdo dum compartimento da memória. Os impulsos correspondentes a tal ordem viajam da memória até um grupo de circuitos bi-estáveis (em número de 16) onde a ordem é memorizada.

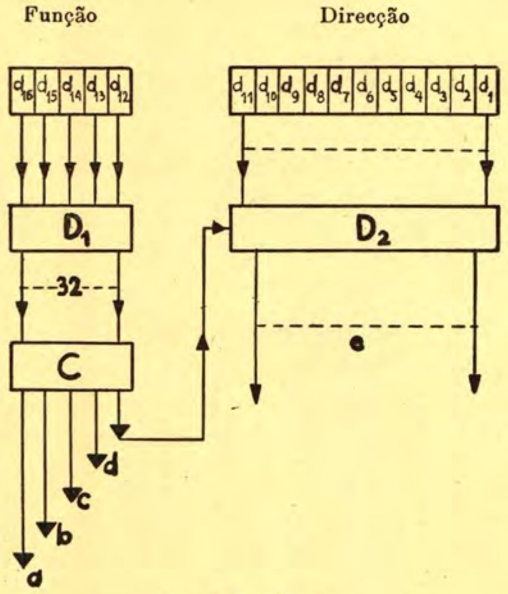



Fig. 13

$D_1$, $D_2$ — decifradores
$C$ — cifrador
$a$ — para a unidade aritmética
$b$ — para o control principal
$c$ — para a via de entrada
$d$ — para a via de saída
$e$ — para a memória

(1) Este grupo de dígitos também representa o número — 0,001111111101011.

O grupo de circuitos bi-estáveis correspondente à parte funcional da ordem está ligado a um *decifrador,* $D_1$, que tem trinta e dois terminais de saída, dos quais um, e só um, é activado para cada tipo de ordem. Este decifrador está por sua vez ligado a um *cifrador* que tem terminais de saída para as diferentes partes da máquina e que vão dar origem aos impulsos necessários à execução da ordem. Um destes terminais de saída está ligado a um decifrador $D_2$ que interpreta os dígitos correspondentes à direcção da ordem.

Considerámos uma máquina com um código de direcção simples, mas para o caso de códigos com várias direcções tudo se passa de maneira semelhante. Assim, para um código de três direcções, a ordem

$A\ l\ m\ n$ — adicione $C(l)$ a $C(m)$ e coloque o resultado no compartimento $n$,

continua a ser representada por um grupo de dígitos dos quais alguns se referem à operação a executar, e os restantes estão divididos em três grupos cada um deles referindo-se a um compartimento da memória.

## 7. Programação

A programação tem por fim traduzir na linguagem da máquina (isto é, por aplicação do respectivo código de ordens) a informação que conduzirá à resolução de determinado problema. Ela compreende duas fases distintas: na primeira, em geral bastante delicada e exigindo sólidos conhecimentos de análise numérica, estabelece-se a sequência de operações adaptáveis à máquina e que resolvem o problema; na segunda, traduz-se essa sequência de operações no código da máquina, obtendo-se o que se chama o *programa* correspondente ao problema em questão.

Na máquina que estamos a considerar (código de direcção simples) o programa é todo colocado[1] na memória em compartimentos sucessivos, digamos nos compartimentos 100, 101, 102, ..., e depois a máquina começa a executar esse programa começando pela ordem do compartimento 100, passando à do compartimento 101, depois à do 102 e assim sucessivamente. Este modo de actuar só poderá ser modificado quando for encontrada uma ordem $Fn$, $Gn$, $En$, ou quando a máquina parar. Antes de começar a execução do programa é necessário, evidentemente, que os números a que se referem as instruções também já estejam na memória.

Vejamos dois exemplos muito simples

1) Suponhamos que durante a resolução de certo problema um número[2] $N$ vai ocupar o compartimento 200 da memória. Pretende-se escrever a sequência de instruções que fazem com que $N$ seja substituído pelo seu módulo.

Se supusermos que o acumulador está limpo, as instruções seguintes permitem obter o que se pretende

| | |
|---|---|
| 100 | $A$ 200 → coloca $N$ no acumulador |
| 101 | $E$ 105 → se $N \geqq 0$ execute a ordem que está no compartimento 105, caso contrário siga normalmente (isto é, execute a ordem do compartimento 102) |
| 102 | $S$ 200 → zero no acumulador |
| 103 | $S$ 200 → $-N$ no acumulador |
| 104 | $T$ 200 → $C(200) = -N$ (e portanto positivo). |
| 105 | ...... |

Um outro conjunto de ordens que resolvem a questão, é o seguinte

| | |
|---|---|
| 100 | $S$ 200 → $-N$ no acumulador |
| 101 | $G$ 103 → se $-N$ é negativo execute a ordem que está no compartimento 103, caso contrário siga normalmente. |
| 102 | $T$ 200 → $C(200) = -N$. |
| 103 | ...... |

Este segundo programa é preferível ao anterior pois faz uso apenas de três ordens.

2) Seja agora calcular o valor do polinómio

$$a + ax + ax^2 + \cdots + ax^{11}$$

para $x = 0{,}5$ e $a = 0{,}25$, supondo que $C(200) = a$ e $C(201) = x$.

Não convém pôr $a$ em evidência porque se obtinha uma expressão da forma $a(1 + \cdots)$ e o número 1 não pertence ao intervalo em que trabalha a máquina. O melhor processo é escrever o polinómio na forma

$$\{[(ax + a)x + a]x + a\}x + a \cdots$$

pois corresponde a efectuar repetidamente o mesmo ciclo de operações: multiplicar um número por $x$ e somar $a$. Mais precisamente, se $S_n$ é o resultado parcial após $n$ repetições do ciclo, então

$$S_{n+1} = x S_n + a$$

Deve então colocar-se $x$ no multiplicador e repetir onze vezes o ciclo

$$(1)\begin{cases} V\,202 \longrightarrow \text{multiplicar } x \text{ por } C(202) \\ A\,200 \longrightarrow \text{somar } a \\ T\,202 \longrightarrow \text{colocar o resultado em 202,} \end{cases}$$

partindo com $C(202) = a$.

Para contar onze repetições do processo vamos colocar $-11 \times 2^{-15}$ num compartimento da memória e aumentar este número de $1 \times 2^{-15}$ depois de cada repetição do grupo de ordens (1):

Depois do primeiro ciclo fica $-10 \times 2^{-15}$
Depois do segundo ciclo fica $-\ 9 \times 2^{-15}$
. . . . . . . . . . . . . . . . .
Depois do décimo-primeiro ciclo fica 0

---

(1) Veremos mais adiante como isso é feito.

(2) Cujo sinal é desconhecido.

Suponhamos então que $C(2)=1\times 2^{-15}$ e que $C(3)=11\times 2^{-15}$.

O programa para o problema proposto será o seguinte (1):

| | | | |
|---|---|---|---|
| | 100 | $H$ 201 | → coloca $x$ no multiplic. |
| | 101 | $A$ 200 | → faz $C(202)=a$ e limpa o acumulador |
| | 102 | $T$ 202 | |
| | 103 | $S$ 3 | → faz $C(4)=-11\times 2^{-15}$ |
| 110 → | 104 | $T$ 4 | |
| | 105 | $V$ 202 | → $x\,S_n$ no acumulador |
| | 106 | $A$ 200 | → $x\,S_n+a$ no acumulador |
| | 107 | $T$ 202 | → $x\,S_n+a=S_{n+1}$ em 202 |
| | 108 | $A$ 4 | → Contagem do número de ciclos e verificação do final |
| | 109 | $A$ 2 | |
| | 110 | $G$ 104 | |
| | 111 | . . . . | |

## 8. Entrada e saída de informação.

Vamos finalmente ver como as ordens e os números são colocados na memória.

Considere-se por exemplo a ordem $A\,315$. Esta ordem deverá ser colocada num compartimento da memória na forma

$$\underbrace{1110}_{\text{função } A}\ \underbrace{000100111011}_{\text{direcção } 315=2^8+2^5+2^4+2^3+2+1}$$

Para isso, por meio dum dispositivo com um teclado semelhante ao de uma máquina de escrever, perfura-se numa fita de papel o que está indicado na figura

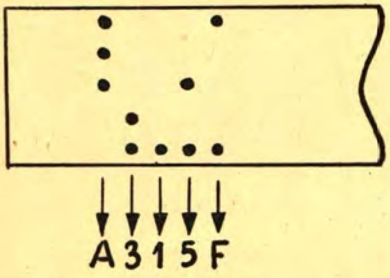



Fig. 14

Esta fita vai ser «lida» fotoelèctricamente pela máquina. O $F$ que agora aparece vai servir para indicar que terminou o que dizia respeito à ordem $A\,315$ e que portanto o que vier em seguida fará parte doutra ordem.

Na máquina existe um programa (1) que vai transformar o que é lido na fita naquilo que se pretende. Nesse programa intervem a ordem

$I\,n$ — Coloque no compartimento $n$ da memória o número $b\times 2^{-15}$, sendo $b$ o inteiro representado no código de entrada pela fila de furos da fita.

As ordens do programa podem ser as seguintes:

| | | | |
|---|---|---|---|
| | 0 | $T$ 28 | → Limpa o acumulador |
| | 1 | $H$ 25 | → Coloca $\frac{10}{16}$ no multiplicador |
| 24 → | 2 | $T$ 30 | → $C(30)=0$ |
| | 3 | $I$ 28 | → $C(28)=0000000000011100$ |
| | 4 | $A$ 28 | → $C(Ac)=C(28)$ |
| | 5 | $L$ 11 | → $C(Ac)=1110000000000000$ |
| | 6 | $T$ 29 | → Ficou «guardada» a função $A$ |
| 16 → | 7 | $I$ 28 | → Lê a fila de furos seguinte e coloca $b\times 2^{-15}$ no acumulador |
| | 8 | $A$ 28 | |
| | 9 | $S$ 26 | → Verifica se se trata dum algarismo ou da letra $F$ indicativa do fim da ordem; neste último caso muda control para o compartimento 17 |
| | 10 | $E$ 17 | |
| | 11 | $T$ 31 | → $C(Ac)=0$ |
| | 12 | $V$ 30 | → $C(Ac)=C(30)\times\frac{10}{16}$ |
| | 13 | $L$ 4 | → $C(Ac)=C(30)\times\frac{10}{16}\times 16=$ $=C(30)\times 10$ |
| | 14 | $A$ 28 | → $C(Ac)=C(30)\times 10+C(28)$ |
| | 15 | $T$ 30 | → $C(30)=C(Ac)$ |
| | 16 | $F$ 7 | → Para recomeçar o ciclo |

(1) Supondo que o acumulador está limpo.

(1) Pode dizer-se que este programa faz parte da máquina.

| | | |
|---|---|---|
| 10→17 | $T$ 28 | → $C(Ac) = 0$ |
| 18 | $A$ 29 | → $C(Ac) = 1110000000000000$ |
| 19 | $A$ 30 | → $C(Ac) = 1110000100111011$ |
| 20 | $T$ 32 | → No compartimento 32 da memória ficou a ordem que se pretende |
| 21 | $A$ 20 | → Tem por fim aumentar de uma unidade a direcção da ordem 20 |
| 22 | $A$ 27 | |
| 23 | $T$ 20 | |
| 24 | $F$ 2 | → Para começar a leitura da ordem seguinte |
| 25 | | → Neste compartimento está o número 10/16 |
| 26 | | → Neste compartimento está o número $10 \times 2^{-15}$ |
| 27 | | → Neste compartimento está o número $2^{-15}$ |
| 28 | | → Compartimentos auxiliares. |
| 29 | | |
| 30 | | |
| 31 | | |

Os números são colocados na memória utilizando um programa apropriado que é lá colocado como acabámos de indicar.

Para obter os resultados calculados pela máquina usa-se também um programa apropriado onde desempenha papel primordial a ordem

$On$ — perfure na fita de saída uma fila de furos que corresponda posicionalmente aos *uns* das cinco posições digitais mais significativas do compartimento $n$ da memória.

No que acabámos de referir suposemos que a entrada e saída de informação era feita através duma fita de papel perfurada. É também muito usual, para tal fim, o emprego de cartões perfurados e de fita magnética.

### BIBLIOGRAFIA


[1] V Belevitch, *Langage des Machines et Langage Humain*, Office de Publicité, S. A., Editeurs, Bruxelles.

[2] *Faster Than Thought* (a symposium on digital computing machines), Sir Isaac Pitman & Sons Ltd, London.

[3] M. V. Wilkes, *Automatic Digital Computers*, Methuen & Co. Ltd, London.

[4] D. R. Hartree, *Calculating Instruments & Machines*, Cambridge University Press.

[5] R. K. Richards, *Arithmetic Operations in Digital Computers*, D. Von Nostrand Co. Inc.

[6] Wilkes, Wheeler, Gill, *Programs for an Electronic Digital Computer* (2d edition), Addison-Wesley Publishing Co. Inc.

[7] R. Higonnet, R. Gréa, *Étude Logique des Circuits Electriques et des Systèmes Binaires*, Editions Berger-Levrault, Paris.


# MATEMÁTICAS SUPERIORES

## PONTOS DE EXAMES DE FREQUÊNCIA E FINAIS

## MATEMÁTICAS GERAIS

F. C. L. — Matemáticas Gerais — 2.º exame de frequência.

**5124** — 1) Estudar a curva $y = x^2 e^x$

**5125** — 2) Enunciar e demonstrar o teorema de Rolle. É o teorema aplicável à função $f(x) = \dfrac{x^2 - 4x}{x+2}$ no intervalo $[0, 4]$?

**5126** — 3) Determinar $\lim\limits_{n \to 0} \dfrac{e^x + e^{-x} + x^2 - 2}{\operatorname{sen}^2 x - x}$

**5127** — 4) Primitivar a) $(e^x + 1)^3 e^x$

b) $\dfrac{1}{x^2 + 10x + 30}$

c) $\dfrac{x^3 + x^2 - 5x + 1}{x^3 + x^2 - 6x}$