

# abc dos Computadores



T. F. FRY

COLECÇÃO CULTURA E TEMPOS LIVRES





## **CULTURA E TEMPOS LIVRES**

1. ABC DO XADREZ, por Petar Trifunovitch e Sava Vukovitch
2. FISHER/SPASSKI — Campeonato Mundial de Xadrez de 1972, por Petar Trifunovitch
4. ABC DO BRIDGE, por Pierre Jais e H. Lahana
5. GUIA PRATICO DE FOTOGRAFIA, por W. D. Emanuel
6. ABC DO JUDO, por E. J. Harrison
7. COMO FAZER CINEMA, por Paul Petzold
8. BRIDGE MODERNO, por Pierre Jais e H. Lahana
9. FOTOGRAFIA — Técnicas e Truques I, por Edwin Smith
10. ESTILOS DO MOBILIARIO, por A. Aussel
11. FOTOGRAFIA — Técnicas e Truques II, por Edwin Smith
12. A PESCA SUBMARINA, por António Ribera
13. TEORIA DOS FINAIS DE PARTIDA, por Yuri Averbach
14. APRENDA RADIO, por B. Fighiera
15. GUIA DO CAO, por Louise Laliberté-Robert e Jean-Pierre Robert
16. ABC DO AQUARIO, por Anthony Evans
17. INICIAÇÃO A ELECTRICIDADE E A ELECTRONICA por Fernand Huré

18. OS TRANSISTORES, por Fernand Huré
19. KARATE I, por Albrecht Pflüger
20. INICIAÇÃO AO RADIOCOMANDO DOS MODELOS REDUZIDOS, por C. Péricone
21. CONSTRUA O SEU RECEPTOR, por B. Fighiera
22. MONTAGENS ELECTRONICAS, por B. Fighiera
23. O BERBEQUIM ELECTRICO, por Villy Dreier
24. CACTOS, por J. Nilaus Jensen
25. INICIAÇÃO A ALTA FIDELIDADE, por Peter Turner
26. O AQUARIO DE AGUA DOCE, por Paulo de Oliveira
27. ABC DO TENIS, por Fonseca Vaz
28. KARATE II, por A. Pflüger
29. ABC DA CRIAÇÃO DE CANARIOS, por Curt Af Enehjelm
30. GINASTICA FEMININA, por Sonja Helmer Jensen
31. CARTOMANCIA, por Rhea Koch
32. CALCULADORAS ELECTRÓNICAS DE BOLSO, por E. Dam Ravn
33. O PASTOR ALEMAO, por Gilles Legrand
34. XADREZ/TEORIA DO MEIO JOGO, por I. Bondarevsky
35. MANUAL DO SUPER 8 - I, por Myron A. Matzkin
36. ABC DA CRIAÇÃO DE PERIQUITOS, por Cyril R. Rogers
37. O LIVRO DOS GATOS, por Bärbel Gerber e Horst Bielfeld
38. MANUAL DO SUPER 8 - II, por Myron A. Matzkin
39. ABC DO MERGULHO DESPORTIVO, por Walter Mattes
40. CIRCUITOS INTEGRADOS / APLICAÇÕES PRATICAS por F. Bergtold
41. A APICULTURA, por H. R. C. Riches
42. ABC DO CULTIVO DAS PLANTAS, por H. G. Witham Fogg
43. ABC DA CRIAÇÃO DE POMBOS, por Kai R. Dahl
44. CONSTRUÇÃO DE CAIXAS ACÚSTICAS DE ALTA FIDELIDADE, por R. Brault
45. RAÇAS DE CANARIOS, por Klaus Speicher
46. JOGOS DE CARTAS, por Graciano Dolma
47. COCKER SPANIELS, por H. S. Lloyd
48. ABC DA CACÁ, por Fabián Abril
49. APRENDA TELEVISÃO, por Gordon J. King
50. INICIAÇÃO A PESCA, por Juan Nadal
51. BASQUETEBOL, por Marius Norregard
52. CAES DE CACÁ, por Santiago Pons
53. APRENDA ELECTRONICA, por T. L. Squires e C. M. Deason
54. A AVICULTURA, por Jim Worthington
55. A PRODUÇÃO DE COELHOS, por P. Surdeau e R. Henaff
56. ABC DOS COMPUTADORES, por T. F. Fry

T. F. FRY

# ABC DOS COMPUTADORES

EDITORIAL PRESENÇA  
PORTUGAL

● LIVRARIA MARTINS FONTES  
BRASIL

Título original:  
**BEGINNER'S GUIDE TO COMPUTERS**  
© *Copyrigt by BUTTERWORTH & CO (PUBLISHERS) LTD.,*  
*1978 — LONDRES*

Tradução de Conceição Jardim e Eduardo Lúcio Nogueira  
Fotografia da capa gentilmente cedida pela Regisconta

Reservados todos os direitos  
para a língua portuguesa à  
**EDITORIAL PRESENÇA, LDA.**  
Rua Augusto Gil, 35-A — LISBOA

## PREFÁCIO

*Como o seu título sugere, o objectivo deste livro consiste em introduzir os seus leitores naquilo que é considerado por muitos uma área bastante misteriosa da tecnologia. É facto, no entanto, que todos sentimos de uma maneira ou de outra o efeito dos computadores na nossa vida de todos os dias, quer nos documentos que recebemos pelo correio e são preparados por computadores, ou no nosso envolvimento em decisões tomadas por uma máquina.*

*Este livro permitirá-nos descobrir um pouco do que se encontra por trás destes produtos finais dos computadores, procurando explicar em termos simples o que o computador é e a maneira como realiza as funções de que está encarregue. Não se pretende que o livro constitua um tratado altamente técnico e, de facto, apenas é possível cobrir, num tão diminuto número de páginas, um esboço geral do assunto. No entanto, poderá ser esta leitura que estimule o interesse dos leitores por um tema ainda tão ignorado, constituindo assim a obra um ponto de partida para estudo mais detalhado daquilo que se tornou uma das áreas mais importantes e de maior alcance da tecnologia do mundo actual.*

*Gostaria de exprimir os meus agradecimentos a Peter Fry pela ajuda que me deu através dos seus amplos conhecimentos de electrónica e a Marian Bentley que tão bem se encarregou do trabalho de preparação do livro.*

T. F. Fry



# 1

## IDEIAS BÁSICAS SOBRE COMPUTADORES

É difícil pensar em qualquer desenvolvimento técnico de maior importância nos últimos vinte anos que tenha tido um impacto na indústria, no comércio e na sociedade em geral tão extenso como os computadores. Raramente mencionada há apenas trinta anos, a palavra «computador» transformou-se num termo por assim dizer «doméstico». Todos ouvimos falar em computadores, muitos de nós ainda os olhamos com uma certa desconfiança, mas poucos temos uma ideia suficientemente adequada sobre o que de facto são e aquilo para que servem.

Sem entrarmos excessivamente em aspectos tecnológicos, procuramos neste livro dar uma ideia básica destas máquinas, examinar a maneira como realizam as suas tarefas e considerar o interesse que assumem na nossa sociedade. No entanto, através das discussões mais aprofundadas que serão realizadas nos capítulos seguintes, será conveniente manter presente a tarefa fundamental que se pretende que os computadores realizem. Na sua forma mais simples, esta tarefa poderia ser descrita do seguinte modo: «aceitar e tratar dados de tal modo que se torne possível obter informações com interesse». Esta afirmação introduz imediatamente os termos *dados*, *tratamento* destes e *informação*, e será bom procurarmos desde já esclarecer um pouco o significado de cada um deles. Talvez uma ilustração bastante simples baste para nos dar uma ideia.

Consideremos três números, 7, 78 e 8. Podemos chamar-lhes dados. Como tais, não têm um grande significado para nós. A fim de obter informações úteis, devemos submetê-los a algum tratamento adequado, por exemplo, colocá-los por uma dada ordem: 8 - 7 - 78. Esta indicação, se for incluída num contexto apropriado, por exemplo, no canto de um cheque, transforma-se numa informação com bastante interesse: a data, 8 - 7 - 78, 8 de Julho de 1978.

Os dados consistem então nos factos e valores relacionados com uma situação que, isolada, pode não ter qualquer significado. *Tratamento* é o processo através do qual se relacionam e interpretam os dados, de tal modo que sejam obtidas *informações* com interesse, que podemos utilizar com algum fim. Para reforçar este conceito básico daquilo que um computador faz, consideremos imediatamente dois exemplos.

1. Um fiscal lê o seu consumo de electricidade, anotando num caderno a leitura do contador. Esta leitura será um *dado*. É enviado à companhia de fornecimento de energia eléctrica, onde existem outros dados relacionados com o serviço que lhe é prestado. Estes outros dados serão por exemplo o seu nome, o seu endereço e o número de referência da sua conta bancária, a leitura anterior, pormenores quanto ao preço por unidade que deverá pagar e outras tarifas. Todos estes dados são então tratados (ou *processados*), sendo então determinado o valor do seu consumo de energia em função de duas leituras do contador, o pagamento que deverá ser executado em função dos preços e tarifas praticados, sendo depois todos estes resultados canalizados como *informação* impressa sob a forma de uma conta que alguém introduzirá na sua caixa do correio.

2. Numa fábrica, são enchidos automaticamente por uma máquina sacos de um quilograma de farinha. Enquanto o saco está a ser enchido, o seu peso é constantemente medido. Estes dados — o peso do saco — são

tratados por comparação com o peso normalizado do saco, neste caso um quilo. Quando ambos os pesos são iguais, é gerada informação sob a forma de uma instrução à máquina no sentido de cortar o enchimento do saco, remover este e começar a encher outro.

### *Computadores*

O primeiro computador completamente automático, ENIAC — Electronic Numerical Integrator and Computer — foi terminado na Pensilvânia, em 1946. Pesava 30 toneladas, continha 18 000 válvulas e diodos termiónicos e ocupava 45 m<sup>2</sup>.

Nos anos subsequentes, as máquinas aumentaram no entanto de potência e de velocidade de funcionamento, enquanto simultaneamente diminuiam de tamanho, de maneira drástica. A soma total de circuitos lógicos e a capacidade de memória do computador ENIAC poderiam, hoje, estar contidos em circuitos electrónicos transistORIZADOS que ocupam o espaço de uma pequena pastilha com 2,5 cm<sup>2</sup>, enquanto a velocidade a que as operações lógicas são realizadas é medida em bilionésimos de segundo.

Um computador não existe apenas por existir. Existe para realizar trabalho, realizar tarefas, num ambiente em que a sua necessidade é já reconhecida. Se agora pensarmos um pouco, verificaremos que virtualmente em qualquer trabalho por nós realizado é possível distinguir três factores básicos. Talvez isto possa também ser melhor explicitado se observarmos um exemplo que nos é bastante familiar — o tratamento de um jardim.

No início do ano, se pretendermos que a nossa actividade de jardineiros tenha algum sucesso, definimos um plano ou, se quisermos, um sistema, que nos propomos seguir durante todo o ano. Este sistema terá primeiramente em conta os objectivos que pretendemos atingir, em termos de flores, frutos e vegetais que desejamos cultivar. Definimos em seguida o nosso plano de opera-

ções — sementeira, etc. — de modo a que todas elas sejam realizadas pela ordem que se torna necessária para podemos alcançar os nossos objectivos. Portanto, realizamos primeiramente uma planificação, um *sistema*.

Estas operações não só devem em seguida ser realizadas pela ordem planeada, como ainda cada uma delas, para ser realizada com êxito, deverá obedecer a um certo número de regras, ou instruções.

A maior parte das pessoas estão tão habituadas a certas operações que de facto não parecem seguir um dado número de instruções detalhadas; com efeito, certas operações são realizadas automaticamente sem pensar no assunto. No entanto, se essas operações nunca tiverem sido executadas antes, somos obrigados a seguir um conjunto de instruções que poderá ter a aparência seguinte (no caso de desejarmos cavar a terra do jardim):

1. Colocar a pá verticalmente com o punho para cima, ficando o gume da pá encostado ao solo.
2. Segurar no punho com a mão direita.
3. Com a mão esquerda segurar o cabo da pá, a metade da distância entre o punho e a pá.
4. Coloque o pé esquerdo sobre o rebordo da pá... etc.

Este conjunto de instruções detalhadas define por-menoradamente a maneira de usar a pá. É este o segundo factor que, em trabalho de computador, será designado por *programa*. Os dois primeiros elementos, os sistemas e os programas, são muitas vezes designados pela expressão inglesa *software*.

O terceiro factor refere-se aos instrumentos de que necessitamos para realizar a tarefa, ou seja, no nosso exemplo, todos os apetrechos para o cultivo do jardim. Estes serão conhecidos, no caso dos computadores, pelo termo *hardware*, igualmente inglês. Como é evidente, no

que se refere às operações realizadas com computadores, o acessório principal será o próprio computador...

Software .....	Sistemas e programas
Hardware .....	Maquinaria e dispositivos

Talvez, neste momento, seja melhor concentrarmos a nossa atenção no computador propriamente dito, a fim de observar as suas funções básicas. Nos primeiros tempos, os computadores eram muitas vezes designados pela expressão enganadora de «cérebros electrónicos», e enganadora porque um computador é afinal apenas uma máquina bastante sofisticada que tem muitas limitações quando comparada a um cérebro. No entanto, as suas funções básicas podem ser bem explicadas por comparação com uma tarefa simples que todos nós podemos executar e, em grande parte, mentalmente.

Suponhamos que, no final de cada semana, desejamos verificar qual irá ser a soma que devemos pagar ao jornaleiro. Este sistema obrigar-nos-á a (a) identificar e quantificar os jornais e outros periódicos que tenham sido entregues, (b) verificar o preço unitário de cada um deles, (c) calcular os custos, multiplicando os preços unitários pelas respectivas quantidades, (d) acrescentar outros custos necessários (correios, etc.), (e) somar todas estas quantias, a fim de obter o valor total da conta que deveremos pagar.

Examinemos como deveremos realizar esta tarefa.

A primeira coisa de que necessitamos é um registo de todos os jornais e revistas entregues durante a semana. Este registo pode ser obtido por simples colocação de uma lista na parede da cozinha, e inscrição de um «visto» à frente de cada publicação assim que é recebida. No final da semana, este documento será a base a partir da qual vamos determinar o valor a pagar. Chamemos-lhe então a *entrada* do sistema.

A fase seguinte consiste em obter o preço de cada publicação considerada. Você poderá manter uma lista actualizada com o preço de cada uma delas, ou talvez

saiba os preços de memória. Mas, quer estejam escritos quer se encontrem na sua memória, estes preços estão «guardados», prontos a serem usados quando forem necessários. Isto traz outro elemento ao nosso sistema, neste caso a *memória ou registo*.

Em seguida, é necessário fazer algum trabalho, que consiste em contar o número de exemplares de cada publicação fornecidos durante a semana, multiplicá-los pelo preço unitário e somar todos os totais assim obtidos, incluindo outros custos, obtendo então toda a quantia que é necessário pagar. Isto é uma questão de simples aritmética, pelo que lhe podemos chamar precisamente isso — *aritmética*. Com toda a probabilidade, estas respostas serão então escritas na lista original, constituindo-se um registo da conta total e das suas diversas parcelas. Chamemos-lhe a *saída do sistema*.

Até agora, encontrámos quatro elementos básicos neste método de trabalho:

Entrada... Registo... Aritmética... Saída...

Existem no entanto ainda dois outros elementos, talvez não tão evidentes. O primeiro destes já foi mencionado no exemplo do jardim, e é a necessidade de seguir uma rotina de instruções, a fim de realizar a tarefa com êxito. Chamámos-lhe um *programa*. O segundo tem a ver com o facto de, tendo terminado a nossa aritmética e anotado a conta total, podermos ter a certeza de não ter feito qualquer erro. Para ter a certeza, seria bom pedir a outra pessoa que confirmasse os resultados. Isto introduz um elemento de controlo.

Finalmente, neste exemplo, observemos um pouco mais de perto aquilo que acontece entre os dois extremos, a entrada e a saída. Neste caso, estamos a tratar da determinação de parte do método de trabalho, utilizando o nosso cérebro e provavelmente um pedaço de papel onde assentamos as informações necessárias. Podemos chamar a isto, ao nosso cérebro e ao pedaço de papel, a nossa *área de trabalho*, ou seja, a área em que estamos envol-

vidos no *tratamento* da entrada de modo a obter a saída pretendida. Este tratamento envolverá, entre outras coisas, a leitura e o armazenamento temporário na área de trabalho de todos os artigos da lista de entrada, assim como o preço de cada um deles obtido a partir dos registos, a fim de realizar as nossas operações aritméticas e, enquanto o fazemos, consultando continuamente as instru-

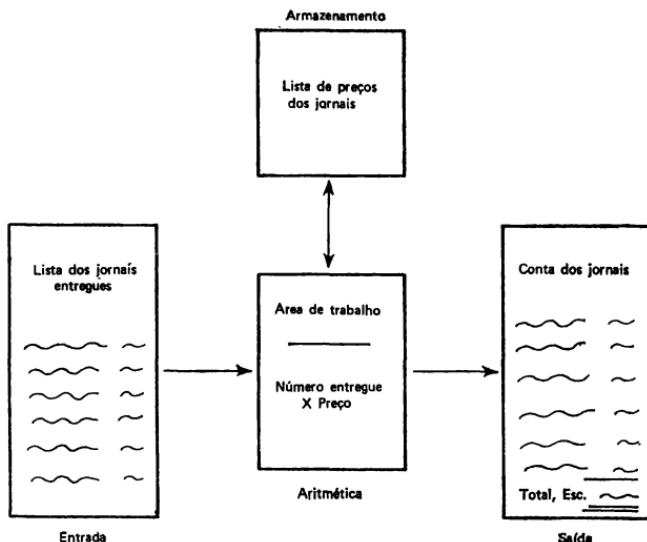


Fig. 1 — Rotina de avaliação do preço dos jornais a pagar.

ções a fim de verificar qual a ordem pela qual as operações devem ser realizadas. Simultaneamente, manter-nos-emos atentos à possibilidade de se verificarem erros e à necessidade de verificar o rigor das operações por um processo de confirmação dos resultados.

A figura 1 mostra a maneira como todos estes factores se encontram presentes e se conjugam, num simples sistema manual.

Acontece que um computador trabalha de maneira bastante semelhante a esta e incorpora os mesmos factores básicos:

A informação original, gravada de uma maneira ou de outra, é lida pela máquina.

A resposta completa é dada a conhecer pela máquina.

Entre ambas encontra-se a «área de trabalho», à qual chamaremos

*Processador central*

que armazena e realiza as instruções da maneira definida num

*Programa*

consultando igualmente dados armazenados

*Memória*

realizando os cálculos necessários e

*Unidade aritmética*

verificando todo o trabalho efectuado.

*Controlo*

O «coração» de um computador é então um dispositivo conhecido pelo nome de *processador central*. É neste que todo o trabalho é feito. Para lhe comunicar a informação sobre a qual deve trabalhar, necessitamos de um dispositivo que leia esta informação e a transfira de maneira aceitável. Chama-se-lhe *entrada*. Do mesmo modo, as respostas obtidas no processador central devem ser comunicadas ao exterior de maneira aceitável e, para tal, necessitamos de um dispositivo de *saída*.

Verificámos que o computador deveria manter ou «recordar» os dados enquanto está a trabalhar, e a esta capacidade chama-se *memória* do computador. No entanto, a quantidade de dados que normalmente devia ser mantida em registo é tão vasta que, como veremos num

capítulo mais adiante, é impraticável que o processador a possa manter em si próprio. Recorre-se então a registo externos, ligados ao processador, sendo este capaz de transferir para si próprio quaisquer dados de que necessite e que neles se encontrem. Isto dá origem à necessidade

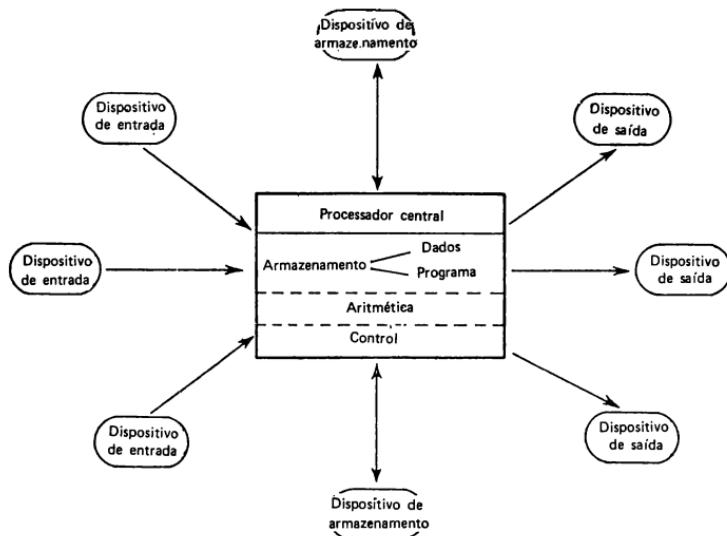


Fig. 2—Elementos básicos de um computador.

de acrescentar dispositivos de registo aos de entrada e saída já conhecidos.

Por conseguinte, ao observarmos um computador, não encontraremos uma única máquina mas sim várias, todas interligadas. A principal, como vimos, é o processador central, mas o seu trabalho apoia-se num certo número de dispositivos de entrada, saída e registo. Este conjunto constituirá uma *configuração de computador*, e os dispositivos que rodeiam e apoiam o processador central são conhecidos pelo nome de *periféricos* (figura 2).

## *O processador central*

As principais características do processador central são a sua capacidade para aceitar e manipular dados, armazená-los e executar instruções programadas quanto a estes dados, assim como verificar este procedimento através de um sistema de controlo. Além disso, como é evidente, deve ter capacidade para comunicar, ou apresentar na saída, os resultados deste processamento.

Os dados em que o processador central está a trabalhar podem ser lidos directamente para registo nele existentes sob a forma de informação primária vinda de um dispositivo de entrada, ou podem já existir em memória e serem transferidos desta para o processador.

A manipulação de dados dentro do processador pode envolver muitos processos diferentes, como veremos mais adiante com maior pormenor, mas incluem a capacidade de tratar funções aritméticas, comparar os dados, e inclusive chegar a decisões com base em critérios que lhe terão sido fornecidos anteriormente. No entanto, se bem que um computador tenha capacidade para realizar bastantes processos, apenas fará aquilo que as instruções lhe tiverem dito para fazer, e o veículo para a comunicação destas instruções é um programa de computador, temporariamente armazenado no processador, permitindo-lhe realizar uma dada tarefa específica.

A capacidade do processador para armazenar informação é por vezes designada por memória do computador. No entanto, um computador é apenas uma máquina e só pode reter a informação que lhe é apresentada nas formas mais simples. De facto, a memória do computador deve apenas recordar afirmações com dois algarismos distintos (0 e 1), e toda a informação que lhe é apresentada encontra-se codificada deste modo. Aplica-se assim um sistema numérico, conhecido pelo nome de *binário*, o qual será tratado com algum pormenor no Capítulo 3.

## *Entrada*

Infelizmente, não é normalmente possível alimentar dados ao computador na sua forma original. Se pudéssemos alimentar a tal lista de jornais, a que já nos referimos, directamente à máquina, levá-la a realizar os cálculos e escrever a resposta no lugar correcto, ejectando novamente a lista pelo outro lado, tudo seria mais simples. No entanto, existem dificuldades que podem impedir um computador de o fazer. Por exemplo, as listas podem ter diferentes formatos e dimensões, com as indicações apresentadas em posições diferentes. A forma da escrita manual pode diferir tanto de uma lista para outra que nos seria difícil a nós próprios lê-la, quanto mais a uma máquina. Para resolver estes problemas, a informação deve (a) ser preparada numa dimensão e numa forma normalizadas que a máquina possa ser capaz de aceitar, e (b) registada de tal modo que a máquina possa reconhecê-la e lê-la.

Existe então um certo número de técnicas que preenchem estes dois requisitos e que serão discutidas no Capítulo 4.

## *Saída*

Foi visto anteriormente que um computador receberá dados, relacioná-los-á e interpretá-los-á com o objectivo de fornecer informação útil. É esta informação útil que representa a saída da máquina. Para ser útil, deve ser comunicada ao exterior de uma maneira que seja compreensível. O capítulo 3 apresentará diferentes maneiras de um computador apresentar os seus resultados.

## *Tipos de computador*

Até agora, temos estado a usar o termo computador num sentido bastante geral, como uma máquina contro-

lada por um programa armazenado internamente, que tem a capacidade de registar informação e realizar um determinado numero de processos com base nesta informação. Existem no entanto dois tipos diferentes de computador, o *computador digital* e o *computador analógico*, se bem que possamos até acrescentar uma terceira categoria de *computadores híbridos*, que serão de facto uma combinação dos dois tipos anteriores, digital e analógico.

### *Computadores digitais*

Um computador digital é essencialmente, como o seu nome implica, uma máquina que trabalha com base em «dígitos» discretos, ou números. Todo o processamento é realizado em termos de representações numéricas da informação trabalhada. O sistema numérico normalmente usado, como já vimos, é o sistema binário, usando apenas dois dígitos, o 0 e o 1. Toda a informação, quer a sua origem sejam dígitos decimais, letras do alfabeto ou símbolos, é convertida numa série de dígitos binários e armazenada deste modo. Todos os dados de entrada são convertidos em expressões binárias, sendo todo o processamento realizado em termos de 0 e 1, e a saída reconvertida de novo para os caracteres convencionais alfanuméricos que todos usamos.

### *Computadores analógicos*

Talvez a melhor maneira de explicar a diferença entre os computadores digitais e os analógicos consista em empregar um exemplo muito simples. Se quiséssemos medir, em litros por minuto, o caudal de água numa torneira, uma maneira de o fazer consistiria em colocar uma série de recipientes de um litro sob a torneira, removendo cada um deles à medida que ficasse cheio. Se contarmos o número de recipientes enchidos em 30 segundos, e o multiplicarmos por dois, obteremos um valor

representando o caudal em litros por minuto. Esta é uma maneira «digital» de encarar o problema — estamos a trabalhar neste caso com números discretos. Isto, aliás, dar-nos-ia o caudal apenas para um determinado meio minuto, ignorando o facto de os caudais se poderem alterar de um minuto para o seguinte, devido a variáveis como sejam a velocidade da água e a abertura da torneira.

Uma outra maneira de resolver o problema consistiria em montar dispositivos de controlo que mediriam continuamente a velocidade da água e a abertura da torneira, converter estas medidas em impulsos eléctricos e relacioná-los entre si de modo a produzir o movimento de um ponteiro numa escala graduada, indicando o caudal em litros por minuto. Teremos então uma maneira «análogica» de encarar o problema. Com efeito, o computador analógico está relacionado com a medição contínua de propriedades físicas e a realização de um tratamento destas medidas, usando as propriedades físicas do próprio computador para fornecer uma analogia do problema que se pretende resolver.

Um outro exemplo muito simples desta função análogica muitas vezes referido é o de um velocímetro de automóvel. Neste caso, a posição do ponteiro, relativamente à escala na qual se move, representa a velocidade da viatura em quilómetros por hora, sendo obtida, não computando números, mas sim pelo controlo contínuo das velocidades de rotação do veio e pela conversão destes dados através das propriedades físicas do dispositivo, engrenagem e cabos, dando no final a leitura pretendida.

### *Computadores híbridos*

Não nos iremos referir agora a um outro tipo de computador, mas sim a uma máquina que incorpora elementos digitais e analógicos simultaneamente.

Tem a vantagem de possuir uma memória em que as instruções de programa podem ser armazenadas e executadas sem necessidade de uma regulação manual e,

como é evidente, pode armazenar variáveis físicas, convertendo-as em expressões digitais. Isto obriga à conversão de medidas de propriedades físicas em afirmações digitais, envolvendo o uso de conversores analógico-para-digital e digital-para-analógico. Por exemplo, poderemos estar interessados em medir a intensidade variável de luz numa dada área, por exemplo, uma gravura. Primeiramente, poderemos atribuir um valor numérico às intensidades variáveis, por exemplo 0 a 99. Se então dividirmos a área num certo número de «pontos» muito pequenos, «varrermos» fotoelectricamente estes pontos e relacionarmos os valores da corrente induzida, obteremos uma série de expressões digitais que podem ser armazenadas em registos e utilizadas a partir daí, a fim de reconstituir a gravura mais tarde. A operação de «varrimento» é o elemento analógico do processo, enquanto o armazenamento de números que representam a intensidade da luz é o elemento digital.

## 2

### OS SISTEMAS NUMÉRICOS E O COMPUTADOR

Como já vimos, um requisito essencial do computador é a sua capacidade para armazenar informações. Este processo de armazenamento é, evidentemente, algo com que todos nós estamos familiarizados, dado que armazenamos mentalmente informações desde que nascemos. Chamamos a isto memória, e esta característica de um computador é também normalmente conhecida pelo nome de memória do computador. No entanto, se compararmos a memória de um computador com a do nosso cérebro, aquele ficará muito mal visto... Podemos armazenar informação em função de uma vasta gama de factores. Um exemplo muito simples e um dos primeiros de que nos recordamos é o das letras do abecedário e dos dez dígitos 0 a 9. Conhecemos estes caracteres e podemos até manipulá-los de modo a formar palavras, expressões e números.

Comparado com o nosso cérebro, o processador do computador constitui um dispositivo muito simples e a gama de caracteres que o computador normal pode tratar é bastante limitada, de facto, apenas dois... Esta função do computador pode ser comparada com uma luz eléctrica que, num dado momento, só pode encontrar-se num de dois estados: acesa ou apagada. Se representarmos o estado «apagada» por um zero — 0 — e o estado «acesa» pelo número um — 1 — teremos a base do sistema numérico com que o computador pode trabalhar. O nome dado a um sistema numérico que só utiliza estes dois dígitos, 0 e 1, é o de *sistema binário*.

## Sistemas numéricos

São habitualmente usados diversos sistemas numéricos com que qualquer de nós está familiarizado. Por exemplo, quando medimos o tempo, sabemos que 60 segundos constituem em minuto, 60 minutos constituem uma hora e 24 horas um dia. No antigo sistema de medidas ingles, 12 polegadas constituem um pé, três pés constituem uma jarda, etc.

Em todos estes sistemas, verifica-se um «transporte» de uma unidade quando se atinge um dado valor.

$$50\text{ s} + 40\text{ s} = 30, \text{ transporte } 1 = 1\text{ m } 30\text{ s}$$

$$6 \text{ polegadas} + 9 \text{ polegadas} = 3, \text{ transporte } 1 = 1 \text{ pé}\\ \text{e } 3 \text{ polegadas.}$$

Na nossa utilização diária dos números, verificamos que o sistema mais utilizado é o *sistema decimal* ou *denário*. O valor de cada dígito na expressão é determinado pela sua posição relativamente aos outros. Chamamos-lhe por vezes *valor de posição*. Por exemplo, no número 27 142, sabemos que o algarismo mais à direita significa 2 unidades, o seguinte 4 dezenas, o seguinte 1 centena, o seguinte 7 milhares e o último 2 dezenas de milhar. Cada algarismo (ou dígito), quando lido da direita para a esquerda, representa um múltiplo de uma potência cada vez maior de 10, em que  $10^0 = 1$ ,  $10^1 = 10$ ,  $10^2 = 10 \times 10 = 100$ , etc.

$$\begin{aligned}27\ 142 &= (2 \times 10^4) + (7 \times 10^3) + (1 \times 10^2) + (4 \times 10^1) + (2 \times 10^0) = \\&= 20\ 000 + 7\ 000 + 100 + 40 + 2 = \\&= 27\ 142\end{aligned}$$

Um sistema numérico baseado em 1—esta *base* é, a propósito, referida em qualquer sistema numérico pelo nome de *raiz*—envolve a utilização de dez símbolos, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Como vimos anteriormente, um computador é incapaz de tratar uma gama tão vasta de algarismos, e está

de facto limitado à utilização de apenas dois, o 0 e o 1. Isto dá-nos um sistema binário, cuja raiz é 2.

Num sistema decimal, verifica-se o transporte de uma unidade de cada vez que se atinge o valor 10, facto que é indicado movendo para a esquerda um 1 seguido de um zero. Num sistema binário, verifica-se esse transporte de cada vez que é atingido o algarismo 2, e isto é indicado do mesmo modo, ou seja, movendo para a esquerda um 1 seguido de um 0.

Em decimal  $5 + 5 = 10$  (decimal 10)

Em binário  $1 + 1 = 10$  (decimal equivalente 2) e

$1 + 1 + 1 = 1$ , transporte  $1 = 11$  (decimal equivalente 3).

Num sistema decimal, os valores de posição aumentam de uma potência adicional de 10 quando os movemos numa expressão numérica da direita para a esquerda, enquanto num sistema binário estas posições aumentam por potências adicionais de 2. Os valores de posição num sistema decimal serão  $10^4$ ,  $10^3$ ,  $10^2$ ,  $10^1$ ,  $10^0$ , e num sistema binário  $2^4$ ,  $2^3$ ,  $2^2$ ,  $2^1$ ,  $2^0$  (recordre-se que  $2^0 = 1$ ,  $2^1 = 2$ ,  $2^2 = 4$ ,  $2^3 = 8$ , etc.). Isto significa que a expressão binária 11011 significará, da esquerda para a direita:

$$(1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = \\ = 16 + 8 + 0 + 2 + 1 = \text{decimal } 27.$$

### *Conversão de uma base para outra*

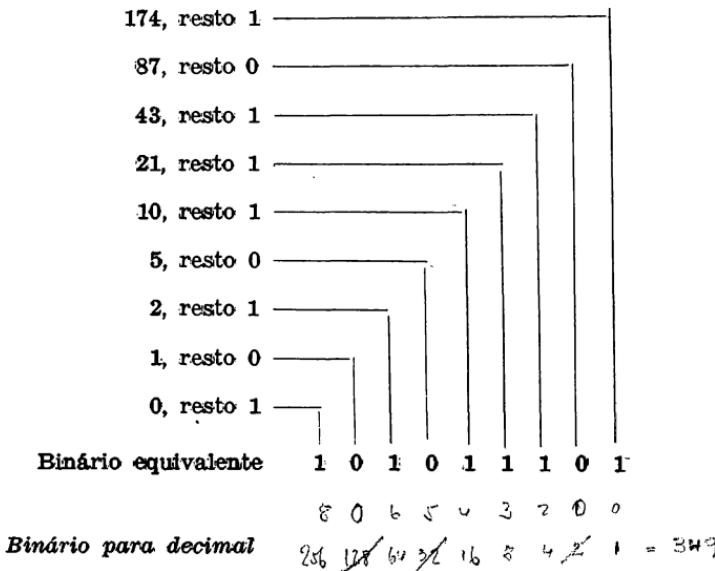
#### *Decimal para binário*

O princípio aqui usado consiste em dividir o número decimal sucessivamente por dois até estar reduzido a zero. Quando na divisão por dois se obtém resto igual a um, este transforma-se no dígito binário 1; quando o

resto é zero, transforma-se no dígito binário 0. A expressão é formada da direita para a esquerda.

Exemplo: converter o número decimal 349 para binário.

349/2:



O valor decimal de uma expressão binária é igual à soma dos valores decimais dos dígitos binários.

Exemplo: converter o número binário 101011 para decimal.

$$\begin{aligned} & 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \\ & = 2^5 + 0 + 2^3 + 0 + 2^1 + 2^0 \\ & = 32 + 0 + 8 + 0 + 2 + 1 = \text{decimal } 43 \end{aligned}$$

## *Fracções binárias*

Num inteiro binário, as posições sucessivas para a esquerda aumentam de valor de uma potência adicional de dois. Numa fração, as sucessivas posições para a direita diminuem o seu valor de uma potência negativa adicional de dois.

<i>Fracção binária</i>	<i>Fracção decimal</i>
$.1 = 2^{-1} = \frac{1}{2}$	$= 1/2 = 0,5 \checkmark$
$.01 = 2^{-2} = \frac{1}{2 \times 2}$	$= 1/4 = 0,25$
$.001 = 2^{-3} = \frac{1}{2 \times 2 \times 2}$	$= 1/8 = 0,125$
$.0001 = 2^{-4} = \frac{1}{2 \times 2 \times 2 \times 2}$	$= 1/16 = 0,0625$

## *Conversão da fração decimal em binária*

Multiplica-se a fração decimal sucessivamente por dois, contando cada 1 de transporte como um binário 1 e, se não houver transporte, como binário 0. Não se considera o transporte para efeitos da multiplicação seguinte. A expressão binária é formada da esquerda para a direita.

Exemplo: converter a fracção decimal 0,625 para fracção binária.

$$\begin{array}{r} 0,625 \\ \times 2 \\ \hline 1,250 \\ \times 2 \\ \hline 0,500 \\ \times 2 \\ \hline 1,000 \end{array}$$

1      0      1 ..... binário equivalente

#### *Conversão de fracção binária em fracção decimal*

O valor decimal da expressão binária é igual à soma dos valores decimais das posições binárias.

Exemplo: converter a fracção binária 0,1101 em fracção decimal.

$$\begin{aligned} & \quad \quad \quad 1 \quad \quad \quad 1 \quad \quad \quad 0 \quad \quad \quad 1 \\ = & \quad 2^{-1} \quad + \quad 2^{-2} \quad + \quad 0 \quad + \quad 2^{-4} \\ = & \quad \frac{1}{2} \quad + \quad \frac{1}{4} \quad + \quad 0 \quad + \quad \frac{1}{16} \\ = & \quad 0,5 \quad + \quad 0,25 \quad + \quad 0 \quad + \quad 0,0625 \\ = & \quad 0,8125, \text{ fracção decimal.} \end{aligned}$$

É evidente, a partir destes exemplos, que qualquer número decimal, inteiro, fracção ou misto pode ser convertido em expressão binária e que o contrário é igualmente verdadeiro, podendo qualquer binário ser convertido em decimal.

Por exemplo:

$$\begin{array}{rcl} 1\,821,703\,125 & = & 11100011101,101101 \\ & & 101101111,0101 = 367,3125 \end{array}$$

### Aritmética binária

Sem nos preocuparmos, por agora, com a «mecânica» do modo como o valor binário é armazenado num computador, nem com a maneira como é manipulado nos processos aritméticos, vejamos no entanto um pouco os princípios de que a máquina se serve para realizar as suas somas.

É verdade que, por muito complexo que seja um problema matemático, se tiver uma solução, pode ser resolvido pela simples aplicação de quatro operações aritméticas básicas, a saber, a soma, a multiplicação, a subtracção e a divisão. Em cálculo manual, a redução de um problema a estas operações simples obrigaría a empregar muito tempo na sua resolução, sendo portanto usadas técnicas mais avançadas para apressar o processo. No entanto, um computador trabalha tão rapidamente que pode realizar uma enorme quantidade de operações aritméticas simples, num período de tempo bastante curto.

Podemos assim utilizar a grande velocidade de funcionamento do computador, para simplificar ainda mais os processos aritméticos internos. Utilizando um método conhecido pelo nome de *soma complementar*, podemos tratar a subtração pelos métodos da adição.

Nos cálculos decimais, o complemento de um número é aquele que lhe deve ser somado para obter como total zero. Por exemplo, o complemento decimal de 473 é 527. Ao somar estes números, o resultado será 1000, obtendo-se portanto zeros nas três posições pretendidas. Para subtrair 473 de outro número, por exemplo 982, soma-se a este o complemento de 473, ou seja, 527 e temos portanto  $527 + 982 = 1509$ . Ignorando o algarismo mais significativo, obtém-se o resultado correcto: 509.

A utilidade deste método, ao calcular no sistema binário, decorre do facto de o complemento de um número ser extremamente fácil de encontrar. Basta inverter o valor de todos os algarismos (um transforma-se em zero e zero em um) e acrescentar um ao resultado.

Vejamos um exemplo de subtracção complementar, utilizando números expressos na base binária:

111010 — 100111

Determinar o complemento de	100111
inverter os algarismos	011000
somar	1
—————	
011001 = verdadeiro complemento	
depois a	111010
somar o verd. complemento	011001.
—————	
(1)010011	

*Resultado:*

111010 — 100111 = 10011

Por este processo de adição complementar, pode-se eliminar a subtracção das nossas operações básicas. A multiplicação pode evidentemente ser também eliminada por soma repetida. O resultado de  $35 \times 38$  pode ser determinado somando 35 a si próprio 38 vezes. É este o princípio básico que o computador utiliza na sua multiplicação binária, se bem que sejam possíveis cortes no tempo necessário, utilizando técnicas de «deslocamento». Estas serão tratadas mais adiante.

De um modo semelhante, a divisão pode ser realizada por subtracção repetitiva. O resultado de  $1330 : 38$  pode ser determinado contando o número de vezes que 38 pode ser subtraído de 1330, até se atingir o zero. Usando estas

técnicas num computador, acaba-se por reduzir todas as operações aritméticas a simples adições binárias.

Até agora, todas as expressões binárias usadas foram apresentadas em binário puro, ou seja, em expressões cujo valor é equivalente ao da expressão decimal original. Ao tratar expressões binárias num computador, no entanto, estas não são necessariamente armazenadas sob essa forma. Se bem que os computadores utilizem quase universalmente a base binária como sistema numérico, são muitas vezes usadas variações na construção das expressões binárias por razões de conveniência e de economia. Talvez seja bom notar que estas técnicas diferentes podem ser usadas em diferentes fabricos de máquinas. Segue-se uma listagem de algumas das maneiras mais vulgares de usar os binários de modo diferente da forma pura ou serial.

### *Codificação de decimal para binário*

Uma outra maneira de registar números, usando o princípio de representação binária, consiste em converter cada dígito decimal individual no seu binário equivalente, em vez de converter todo o número, obtendo uma expressão binária contínua. O número 359 por exemplo é representado em forma binária pura por 101100111; convertendo cada algarismo individual para a base binária, obtém-se

3	5	9
11	101	1001

Ao armazenar números sob esta forma num computador, é normal atribuir um número de dígitos sempre

igual a cada um dos dez algarismo 0 a 9, o que dará o seguinte conjunto de valores:

$$0 = 0000$$

$$1 = 0001$$

$$2 = 0010$$

$$3 = 0011$$

$$4 = 0100$$

$$5 = 0101$$

$$6 = 0110$$

$$7 = 0111$$

$$8 = 1000$$

$$9 = 1001$$

Voltando ao exemplo anterior, ou seja, 359 em código binário, é importante recordar que o valor de posição de cada algarismo binário em cada grupo é medido em termos de potências de 2, e os grupos propriamente ditos em termos de potências de 10:

$$[2^4 + 2^0]^{10^2} + [2^2 + 0 + 2^0]^{10^1} + [2^3 + 0 + 0 + 2^0]^{10^0}$$

### *Sistema numérico octal*

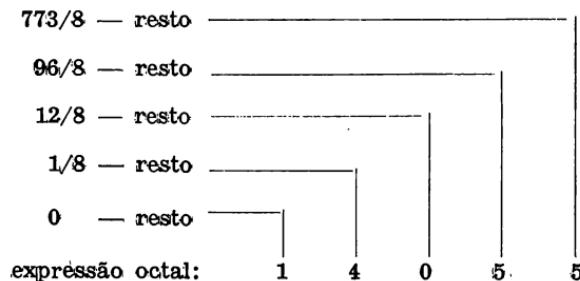
Verifica-se que são necessários quatro «bits» (do inglês «binary digits», ou seja, dígitos — ou algarismos — binários) variando entre 0000 e 1001 para exprimir um algarismo decimal entre 0 e 9. No entanto, este conjunto de quatro bits pode significar dezasseis valores diferentes, entre 0000 e 1111, o que significa que o uso da codificação binária não é económica no sentido de que seis dos arranjos possíveis não são utilizados. Se reduzirmos o número de bits de quatro para três, teremos possibilidades de exprimir oito valores, equivalentes aos números decimais 0 a 7, ou seja 000-111. A expressão de números utilizando a base 8 — a raiz 8 — é designada por *sistema*

*octal* e, quando cada algarismo do número octal está expresso em binário, chamamos-lhe *sistema octal codificado em binário*.

Um número decimal pode ser convertido em octal de uma maneira bastante semelhante à empregue para o converter em binário, excepto no que se refere ao facto de ser utilizada uma divisão por 8.

Exemplo: converter 6189 para a base octal.

6189/8



Em termos decimais esta expressão é agora igual a

$$(1 \times 8^4) + (4 \times 8^3) + 0 + (5 \times 8^2) + (5 \times 8^0) \\ = 4096 + 2048 + 0 + 40 + 5 = 6189$$

### *Adição binária*

Tal como no caso da adição decimal é transportado um para a esquerda sempre que é atingido o número 10, no caso da adição binária, é transportado um para a esquerda sempre que é atingido o valor 2. As regras simples são:

$$\begin{array}{r} 0 + 0 = 0 \\ 0 + 1 = 1 \\ 1 + 0 = 1 \\ 1 + 1 = 0, \text{ transporte } 1 \\ 1 + 1 + 1 = 1, \text{ transporte } 1 \end{array}$$

Exemplo:

Decimal		Binário
	42	101010
+	60	111100
	—	—
transporte	1	111
	102	1100110
	—	—

Note-se que não se verifica aqui um transporte até ao quarto par de dígitos binários a partir da direita. Neste caso,  $1 + 1$  é igual a 0 e transporte 1 é o dígito mostrado abaixo da linha que se encontra sob o quinto par de dígitos. Aqui  $0 + 1 + \text{transporte } 1$  é igual a 0 transporte 1, sendo o transporte de 1 apresentado debaixo da linha sob o sexto par de dígitos, onde  $1 + 1 + 1$  é igual a 1 transporte de 1.

### *Subtração binária*

Neste caso, interessa-nos determinar o verdadeiro complemento do subtractivo, realizando em seguida a adição normal como se mostrou anteriormente.

Exemplo:

Decimal	Binário
150	10010110
— 108	1101100
—	
42	
—	
Primeiro: determinar o verdadeiro complemento de inverter dígitos somar	1101100 0010011 1
—	
Sumar então:	0010100 10010110 0010100
—	
	(10)101010

Ignorar o dígito mais significativo.

Resultado:

$$101010 = 42$$

## *Caracteres alfabéticos em representação binária*

Até agora, interessámo-nos apenas em observar a maneira como os números podem ser expressos em termos binários. Mas, como é evidente, um computador deve tratar igualmente caracteres alfabéticos e símbolos, e cada um destes deve ser apresentado nos registos por uma expressão binária única. Não existe uma maneira normalizada de o fazer, no sentido de que não existe um método único empregue por todos os fabricantes de computadores, mas vamos considerar em seguida uma maneira possível de resolver o problema.

Como já vimos, os dez dígitos 0 a 9 podem ser expressos por quantidades decimais codificadas em binário, 0000 a 1001. Pode-se então adoptar este princípio, dividindo o alfabeto em três grupos de 9, 9 e 8 caracteres, consistindo em A a I, J a R e S a Z respectivamente. A posição numerada da letra em cada grupo pode ser indicada usando a mesma gama de representação utilizada no caso dos números, ou seja, A = 0001, I = 1001. O grupo pode então levar um prefixo de dois dígitos adicionais que indicam o grupo a que pertence a letra. Por exemplo, poderemos usar 01 para o grupo A — I, 10 para o grupo J — R e 11 para o S — Z, deixando o prefixo 00 para representar um número 0 — 9. Neste caso, a letra A seria representada por 010001, K por 100010 e U por 110011, assim como 7 por 000111. No entanto, se bem que isto explique o princípio usado na codificação de caracteres, torna-se evidente que, dado que num grupo de quatro bits existem dezasseis combinações possíveis, 0001—1111, usar apenas nove de cada grupo constitui um processo não económico. É muito melhor usar todas as combinações num grupo de seis bits entre 000000 e 111111, obtendo-se então um total de 64 que nos permite usar representações únicas para dez dígitos, 26 letras e até 28 símbolos.

Tendo assim observado o sistema numérico básico, que é o binário, utilizado no processador de um computador, e a maneira como este pode ser adaptado e

utilizado de diferentes maneiras, talvez devamos examinar agora como, em princípio, o computador armazena toda esta informação.

Sem nos preocuparmos, de momento, com os aspectos técnicos da maneira como tal é feito, basta dizermos que um registo de computador consiste basicamente num grande número de dispositivos, muitos milhares num computador de tamanho médio, sendo cada um destes capaz de representar um 0 ou um 1, e tendo a possibilidade de ser invertido ou alterado, se tal for conveniente. Quaisquer que sejam os caracteres que estamos a usar, estes terão evidentemente de ser ordenados e agrupados de acordo com algum sistema previamente determinado, de modo a transmitir alguma coisa que tenha significado. Por exemplo, se esta página estivesse coberta por uma sucessão contínua de caracteres alfabéticos, não teria qualquer sentido. Estes deveriam ser agrupados em palavras, sendo cada uma delas diferenciada das seguintes por um espaço, sendo ainda necessário compor estas palavras em frases significantes e definir o início e o final de cada uma dessas frases.

Assim, com todos estes 0s e 1s na memória do computador, é necessário introduzir alguma ordem no conjunto para que todos eles ganhem um significado; mas estabelecer esta ordem não é tão fácil como no caso dos caracteres comuns.

Vejamos primeiro os problemas que temos de resolver:

1. É necessário um grupo de bits binários (0s e 1s) para definir cada dígito, carácter ou símbolo.

2. É necessário um certo número destes grupos para formar aquilo a que chamamos uma palavra ou uma expressão numérica. Em termos de computadores, chamamos a isto um *campo*.

3. Um certo número destas «palavras» ou campos constituem aquilo a que usualmente chamamos uma frase

mas, em termos de computadores, isso é designado por um *registo de dados*.

Tudo isto é expresso em dígitos binários e, para que faça sentido, é necessário distinguir cada carácter do seguinte, cada campo de qualquer outro e cada registo de qualquer outro. Tendo igualmente em mente, como se disse anteriormente, que a memória do computador conterá possivelmente centenas de milhares destes bits, torna-se necessário saber onde se encontra qualquer grupo particular de informações.

Começamos portanto por dois problemas principais: (a) identificar a gama de bits que representa cada carácter, campo e registo, e (b) poder localizar qualquer dado armazenado na memória. Uma solução para o primeiro problema, que pode surgir por si própria, consiste em marcar de algum modo o ponto em que cada grupo de bits começa e acaba, mas isto não é tão fácil como parece. Que marca iremos usar? Só temos duas alternativas, 0 e 1, e ambas são utilizadas para constituir o próprio carácter.

Tendo posto de parte o uso dos próprios bits como marcadores, poderemos pensar em seguida que uma maneira de resolver o problema consistiria em dividir todos os bits em grupos de uma dimensão «standard», contendo cada grupo um número suficiente de bits para registar um carácter. Fixa-se assim o limite de cada grupo, o que significa que deixa de haver uma necessidade de marcar o ponto em que cada grupo começa e termina. Aceitando isto, podemos agora ir um pouco mais adiante. Se dermos a cada grupo uma referência única, teremos então um meio de localizar qualquer grupo que se torne necessário para os cálculos.

Talvez neste momento devamos recordar que são actualmente usados muitos tipos diferentes de computador. Não existe nenhuma maneira normalizada de organizar os dados armazenados que seja comum a todos os processadores centrais, e de facto os métodos de agru-

pamento variam consideravelmente. Tendo isto em vista, iremos referir apenas as ideias básicas.

O princípio básico consiste em usar, na memória do processador, grupos normalizados de bits. Talvez possamos considerar o assunto como se tivéssemos um grande número de caixas pequenas, tendo cada uma delas um número de referência, como se mostra na figura 3. Cada caixa contém bits suficientes para registar um ca-



Fig. 3 — Maneira como a memória de um computador é dividida em palavras endereçáveis.

rácter e este é localizado por referência ao número da caixa. O nome que damos a este número é *endereço*.

Tendo decidido dividir a memória em grupos de bits, o problema seguinte consiste em determinar a dimensão do grupo. A figura 3 sugere seis bits para cada grupo, e esta é de facto uma dimensão conveniente porque, como vimos anteriormente, permitirá acomodar qualquer dígito, carácter ou símbolo e dá uma gama de 64 combinações possíveis, 000000 — 111111 (0 — 63). Mas agora surge outra complicação. Suponhamos que pretendemos armazenar números em expressões binárias puras. O maior número que poderemos representar neste caso será 63 (111111), sendo já necessários sete bits até ao número 127, oito até 255, etc.

Devemos mencionar agora três pontos relativamente ao armazenamento de números no processador:

1. O armazenamento em codificação binária obriga a utilizar um espaço maior que o ocupado pelo mesmo número na base binária. Por exemplo, 271 em forma binária pura necessita de nove bits, 100001111, enquanto em codificação binária já obriga a usar pelo menos 12 bits: 0010 0111 0001. Seria portanto mais económico armazenar os valores em forma binária.

2. É habitual, num computador, exprimir os números na forma binária a fim de realizar as operações aritméticas. São necessários circuitos electrónicos bastante mais complexos para realizar estas operações em codificação binária.

3. A maior parte dos dados armazenados, tanto numéricos como alfabéticos, não serão sujeitos a processamento aritmético. Trata-se dos dados que têm uma natureza descritiva em vez de quantitativa, como sejam números de referência, datas, descrições, etc. Estes dados mistos *alfa/numéricos* só podem ser armazenados na forma de grupo de caracteres.

Isto conduz-nos a uma situação em que alguns dados são melhor armazenados carácter a carácter — de facto isto representa frequentemente a maior parte dos dados, e nesse caso o grupo de seis bits é ideal—enquanto os outros dados quantitativos devem ser armazenados em forma binária pura, para o que um grupo de apenas seis bits é demasiado pequeno. Uma solução consiste em realizar um compromisso entre estes dois requisitos, tendo um grupo que conterá um número exacto de caracteres com seis bits e seja, ao mesmo tempo, suficientemente vasto para acomodar um número relativamente grande expresso em forma binária. Por exemplo, podemos dispor de um grupo de 24 bits. Este conterá quatro grupos de caracteres de seis bits ou uma expressão binária pura. Na figura 4, apresentam-se alguns exemplos.

Talvez fosse bom, neste ponto, tentar racionalizar a nossa terminologia. Já usámos diversas vezes o termo «bit», que significa um único dígito binário, 0 ou 1.

Sugerimos que os dígitos binários, ou seja, os bits, podem ser organizados em grupos de uma dimensão normalizada, por exemplo, 24. O nome que daremos a cada grupo destes é *palavra*. Por definição, cada palavra possui um unico *endereço*. Sugerimos ainda que cada palavra,

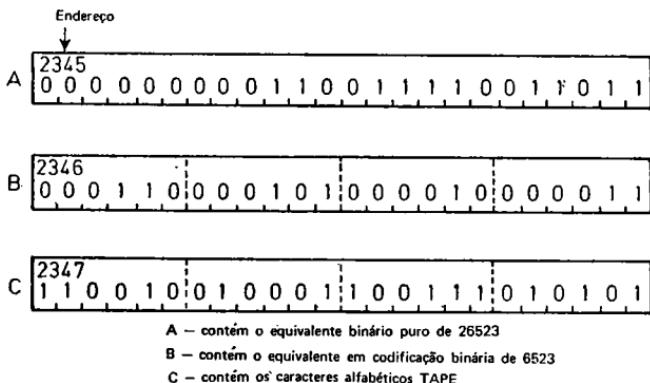


Fig. 4—Exemplos de uma palavra de 24 bits.

contendo por exemplo 24 bits, possa ser subdividida num certo número de localizações de caracteres, por exemplo, quatro de seis bits cada. Estas subdivisões são normalmente conhecidas pelo nome de *bytes*. Talvez conviesse mencionar agora que, ao contrário de cada palavra que é endereçável, ou seja, possui uma referência de endereço único, os caracteres individuais, ou bytes, já não a têm.

No entanto, se os dados se encontrarem organizados de tal modo que o primeiro deles a ser lido apareça sempre no primeiro grupo (ou seja byte) da palavra, e os caracteres seguintes dispostos numa sequência estrita, a máquina necessitará apenas de conhecer o endereço da palavra, lendo o primeiro carácter continuando a fazê-lo até receber uma instrução para parar a leitura. Esta ins-

trução de paragem pode ser dada dizendo à máquina para ler um certo número bem definido de caracteres, por exemplo.

Um outro método de levar a máquina a interromper a leitura dos dados consistirá em colocar um grupo especial de bits com o significado «parar» (nas instruções do programa, utiliza-se neste caso a palavra inglesa «stop») depois do último carácter que deva ser lido.



# 3

## LÓGICA DE COMPUTADORES

O objectivo deste capítulo consiste em observar alguns dos princípios em que é baseada a concepção de um computador. Hoje em dia, ouve-se falar muito de automação, o que de facto se refere a um processo que é realizado automaticamente. As máquinas distribuidoras foram em tempos consideradas «automáticas». Quando nelas se introduz uma moeda, é-se servido automaticamente do produto, em vez de se ter de falar com um vendedor agarrado a uma máquina registadora. Quando estamos a falar de automação pensamos numa tarefa que, depois de iniciada, é continuada até ao fim sem necessidade de qualquer outra intervenção humana. Esta propriedade automática é uma característica do computador, se bem que, evidentemente, não baste por si só para o definir na medida em que se devem referir além disso a velocidade, o rigor e a capacidade para realizar operações bem definidas.

No entanto, observemos um pouco mais de perto este princípio da automação. Se voltarmos ao exemplo da máquina distribuidora, verificaremos que entre o momento em que a moeda é inserida e aquele em que o produto é ejectado se verifica um certo número de passos lógicos realizados automaticamente. Quando a moeda é inserida, a primeira coisa a fazer é realizar uma decisão Sim - Não no sentido de decidir se esta é ou não aceite. Isto envolve algum tipo de medição ou pesagem que

determinará o passo lógico seguinte, que será rejeitar ou aceitar a moeda. De facto, pode ser necessário inserir mais que uma moeda para que a máquina funcione, caso em que é necessário realizar uma adição lógica. Em seguida, vem a referência à parte da máquina onde se encontra o produto em causa. É novamente realizada uma decisão: se a caixa do produto está vazia é necessário rejeitar a moeda, se está cheia deve ser accionado o mecanismo que fornece o produto ao exterior.

Tudo isto constitui evidentemente um processo automático mecânico ou electromecânico, envolvendo alavancas, rodas, relés, etc. O processador central de um computador é automático num sentido muito semelhante. Depois de um dado processamento ter sido iniciado, continuará sem qualquer intervenção humana até ter terminado. O primeiro computador foi construído com o mesmo tipo de alavancas, rodas, etc., mas estas, como é evidente, impediam a sua característica mais interessante — a velocidade.

Actualmente, os computadores são integralmente electrónicos e a sua capacidade lógica e automática é melhorada através do controlo do caudal dos impulsos eléctricos. Por muito eficientes que os antigos dispositivos mecânicos fossem, os seus tempos de funcionamento eram medidos em segundos. Mesmo os primeiros computadores integralmente automáticos permitiam tempos de microssegundos (milionésimos), e nas máquinas mais recentes já pensamos em termos de nanossegundos (bilionésimos). Um potente computador moderno pode realizar a soma de dois números de 18 bits em menos de 500 nanossegundos, o que significa que pode realizar cerca de dois milhões de cálculos por segundo. Com o recente desenvolvimento dos circuitos integrados, devemos agora começar a pensar em termos de picossegundos (trilionésimos).

Nos primeiros tempos, os computadores eram por vezes designados por «cérebros electrónicos». Este conceito não é muito apropriado. No entanto, se o objectivo de um computador consiste em substituir-nos, de resto com aumento de velocidade, na realização de certas tare-

fas que de outro modo deveriam ser realizadas pelo nosso cérebro, deve de qualquer modo ser capaz de simular os nossos processos mentais. Ora uma das capacidades que nós possuímos num grau maior ou menor é a do pensamento lógico. Esta é a capacidade de resolver um problema seguindo os seus passos lógicos até chegar à solução. Quando analisamos os aspectos essenciais, como vimos no Capítulo 2, um computador é apenas capaz de tratar conceitos muito básicos. Todo o seu «pensamento» é de facto realizado em termos de 0 e 1; isto não significa que não seja capaz de resolver problemas extremamente complexos, mas significa que estes problemas devem, em última análise, ser reduzidos a termos de zeros e uns.

### *Lógica dos computadores*

Mencionámos anteriormente o termo automação relativamente a um computador. A fim de realizar automaticamente uma sequência de acontecimentos em que diferentes acções devem resultar de condições variáveis, as acções devem ser baseadas na lógica das condições em causa. Por exemplo, parte da lógica da máquina distribuidora é, se a moeda não tem o peso correcto, rejeitá-la. O computador, ao realizar as suas operações, será concebido de modo a aplicar técnicas lógicas, entre as quais se encontram a lógica da adição, da comparação (verificando coincidências ou grandezas), a lógica da contagem e a lógica da complementação (inversão ou negação). Todas estas operações devem ser realizadas em termos de dois únicos valores, 0 e 1. Esta lógica binária tem como base uma ciéncia matemática conhecida pelo nome de *álgebra booleana*, desenvolvida pelo matemático George Boole (1815-1864).

#### *A Álgebra Booleana*

Na álgebra convencional que todos nós usamos, as variáveis de qualquer afirmação do tipo  $A + B = C$  podem

ter um número infinito de valores. Em álgebra booleana, as variáveis representam condições e cada afirmação booleana indica a relação lógica entre estas condições utilizando «ligadores» lógicos como sejam o «e» e o «ou» (em inglês «and» e «or» — são estes os termos usados na escrita das expressões booleanas). A aplicação da álgebra booleana ao computador deve-se à restrição das variáveis, neste caso, a apenas duas condições possíveis, «verdadeiro» e «falso», ou, numericamente, 1 e 0. A maneira de aplicar esta álgebra booleana no computador fundamenta-se na utilização de circuitos e dispositivos electrónicos. Dado que, nestas condições, o número de variáveis de uma afirmação booleana tem um número finito de valores (dois apenas no caso da lógica de computadores), então o número de funções a que estas variáveis podem dar origem é igualmente finito. Este número pode ser expresso pela quantidade  $2^n$ , onde  $n$  é o número de variáveis binárias. Com duas variáveis, o número de funções é então de  $2^2 = 4$ .

Neste capítulo, não se pretende apresentar pormenorizadamente as proposições da álgebra booleana, mas sim ilustrar a maneira como esta matemática fornece a base de funções específicas de computadores e, a partir daí, de dispositivos electrónicos específicos.

Consideremos por exemplo uma proposição bastante simples. «Se estiver vento e estiver a chover, levarei o meu casaco». Esta afirmação contém três variáveis. As duas primeiras, «se estiver a fazer vento» e «se estiver a chover» podem ser designadas por variáveis *independentes* — podem ser verdadeiras ou falsas. O valor da terceira «levarei o casaco», se bem que deva também ser verdadeira ou falsa, dependerá já do valor das outras duas. Podemos portanto descrevê-la como uma função ( $f$ ) das outras duas variáveis independentes. Se designarmos as variáveis em causa por A, B e C, poderemos exprimir a situação sob a forma  $C = f(A, B)$ . Como vimos anteriormente, o número de combinações de duas variáveis binárias é  $2^n$ . Neste caso  $n = 2$  e portanto as funções possíveis entre estas serão 4.

As quatro combinações possíveis no nosso exemplo são:

Não chove	Não faz vento
Chove	Não faz vento
Não chove	Faz vento
Chove	Faz vento

As funções produzidas por estas combinações serão respectivamente:

Não levo o casaco  
Não levo o casaco  
Não levo o casaco  
Levo o casaco

A	B	C	
0	0	0	0
0	1	0	2
1	0	0	4
1	1	1	7

Fig. 5 — Tabela de verdade da função lógica AND.

Como é evidente, é muito mais fácil exprimir estes resultados sob a forma de uma tabela (figura 5). Este tipo de tabela é designado por *tabela de verdade*, dado que dá o valor de verdade de C para todas as combinações de variáveis independentes (variáveis constitutivas) A e B. Poderá ser mais simples afirmar apenas que a coluna C representa a função de saída de todas as combinações

de *entrada*. De facto, é esta a terminologia que usamos quando aplicamos estes princípios aos computadores.

Em termos booleanos, teremos uma proposição cujas funções são determinadas pela ligação lógica AND («e», em português). Chama-se a esta relação uma *intersecção*, o que significa que C só é verdadeiro quando A e B o são igualmente. Poderemos exprimir este facto algebricamente pela expressão  $C \equiv A \cdot B$ .

Outra maneira de exprimir esta relação será através de um *diagrama de Venn*, que constitui de facto o equi-

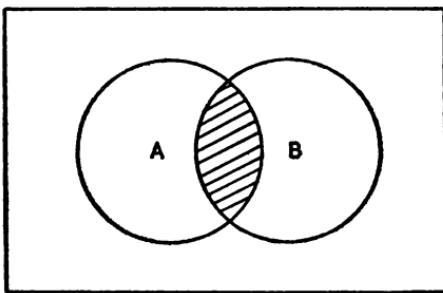


Fig. 6 — Diagrama de Venn para a função lógica AND.

valente, em gráfico, de uma tabela de verdade. Um diagrama de Venn consiste essencialmente em dois círculos no interior de um rectângulo. Aqueles representam as duas variáveis de entrada A e B. A área branca em qualquer dos círculos ou no rectângulo representa o valor falso ou 0, enquanto que a área tracejada representa o valor verdade ou 1. O diagrama da figura 6 representa a nossa proposição original.

Tudo o que se encontra no interior do círculo A representa a variável de entrada A, enquanto que tudo o que se encontra no interior do círculo B representa a variável de entrada B. Do mesmo modo, tudo o que se encontra fora do círculo A representa o complemento de A, ou seja o NAO A (NOT A, em inglês), normalmente indicado por  $\bar{A}$ , e tudo o que está fora do círculo B é  $\bar{B}$ . Tudo o que

está fora da área em que os círculos se sobrepõem é comum às variáveis A e B que, como já vimos, pode ser representado pela notação booleana  $A \cdot B$ . É esta área a tracejada que representa a função de saída C como verdadeira (1), verificando-se este resultado apenas quando

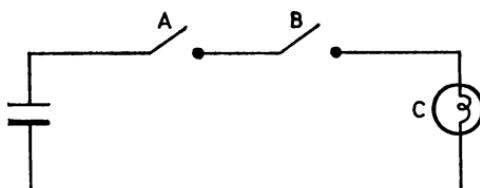


Fig. 7 — Um circuito eléctrico simples ilustrando a função AND.

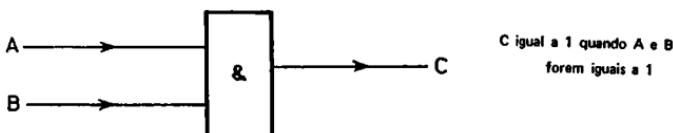


Fig. 8 — Porta AND.

as duas variáveis de entrada A e B constituem uma condição verdadeira.

Esta proposição é conhecida pelo nome de função lógica AND («e» em português), que de facto estipula que, no caso de duas variáveis binárias de entrada, a saída só será verdadeira (1) quando ambas as entradas o forem. Esta função AND pode ser realizada por um simples circuito eléctrico apresentado na figura 7. Para que a lâmpada C esteja acesa (valor lógico 1), ambos os interruptores A e B devem estar ligados (terem ambos o valor 1).

Os dispositivos que realizam electronicamente as proposições da álgebra booleana são conhecidos pelo nome de portas (em inglês, «gates»). Uma porta AND, como o nome implica, é concebida para realizar uma função AND.

Emitirá um impulso de saída sempre que forem aplicados impulsos simultaneamente às duas entradas (figura 8).

Voltemos agora à nossa proposição original e alteremos-a um pouco: «Se estiver a chover ou a fazer vento, levarei o meu casaco». A ligação lógica é agora realizada por um OU, o que evidentemente irá alterar as funções que derivam da condição das variáveis. Se bem que as combinações de condições das variáveis de entrada continuem a ser as já citadas, as funções serão agora:

Não levo o casaco  
Levo o casaco  
Levo o casaco  
Levo o casaco

A tabela de verdade desta proposição OR («ou» em inglês, que constitui o termo usado em álgebra booleana) e o seu diagrama de Venn são apresentados na figura 9. Em termos booleanos poderia ser expresso como  $C \equiv A + B$ .

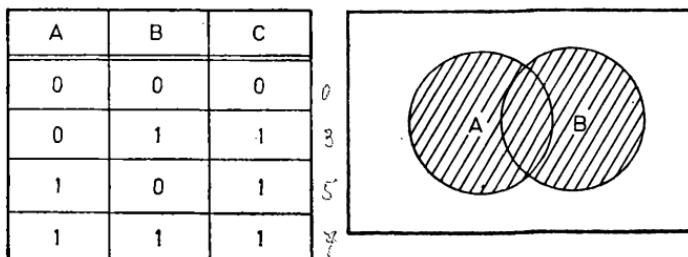


Fig. 9 — Tabela de verdade e diagrama de Venn para a função lógica OR.

Do mesmo modo, as áreas tracejadas do círculo do diagrama de Venn indicam uma posição de verdade para A e para B. Isto significa que se tanto A como B forem verdade (1), então a função de saída C deve ser igualmente verdade (1). Isto é conhecido pela designação de propo-

sição lógica OU e pode ser demonstrado por um circuito eléctrico simples (figura 10). Para que a lâmpada C esteja ligada, é necessário que pelo menos um dos interruptores A ou B o esteja.

Tal como no caso da proposição AND, pode-se construir uma porta electrónica para produzir as funções de

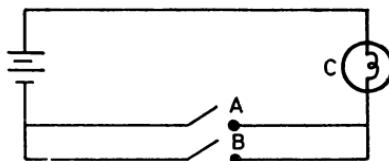


Fig. 10 — Um circuito eléctrico simples que mostra a realização da proposição OR.

uma proposição OR. Conhecida pelo nome de porta OR, emitirá um impulso sempre que esteja presente um impulso em qualquer das entradas ou em ambas simultaneamente (figura 11).

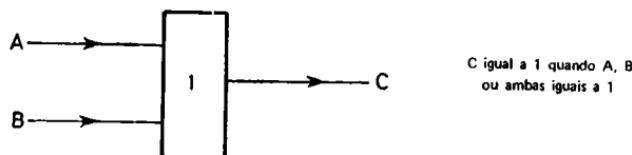


Fig. 11 — A porta OR.

Pensando agora no que se disse no Capítulo 2, as regras básicas da adição são:

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0, \text{ transporte } 1.$$

Esquecendo por agora o transporte, é ainda evidente que nenhuma das portas discutidas até agora produzirá exactamente estes resultados. A porta OR satisfaz as três primeiras condições, mas nenhuma respeita a quarta. Para tal, é necessário introduzir uma outra proposição booleana. Esta estipula que quando uma variável, por exemplo B, é verdadeira, então tudo o que não é B, ou é não-B ( $\bar{B}$ ), será falso e, reciprocamente, quando B é falso,  $\bar{B}$  será verdadeiro. Isto constitui de facto uma *proposição de inversão*, conhecida pelo nome de função NOT («não» em português). Este inversor possui apenas uma entrada e uma saída e significa, em termos práticos, que a saída será 1 quando a entrada é 0 e 0 quando a entrada for 1.

Podemos agora juntar estas três proposições, AND, OR e NOT sob a forma de um dado arranjo de portas que permitirá realizar a operação de adição. Chama-se a este arranjo ou montagem um *semi-somador*.

### *Semi-somador*

A figura 12 mostra um arranjo típico de portas constituindo um semi-somador, através do qual é possível obter as quatro combinações de valores de A e B como se segue:

<i>Entrada</i>	<i>Entrada porta AND 1</i>	<i>Entrada porta AND 2</i>	<i>Entrada porta OR</i>	<i>Saída</i>
0 0	1 0	0 1	0 0	0
1 0	0 0	1 1	0 1	1
0 1	1 1	0 0	1 0	1
1 1	0 1	1 0	0 0	0

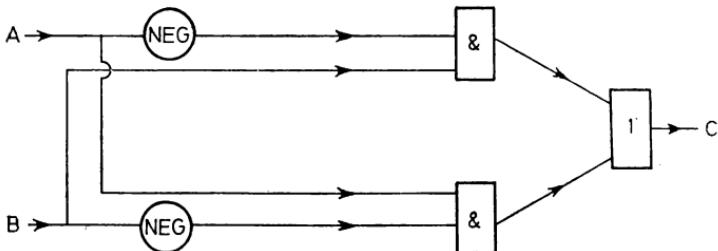


Fig. 12 — Semi-somador binário de um bit (função OR exclusiva).

Esta montagem dá as condições daquilo que é designado pela OR exclusiva, escrita sob a forma

$$C = \bar{A} \cdot B + A \cdot \bar{B}$$

o que significa que C (saída) é verdade (1) quando B o é e A não, e quando A o é mas não B. Este dispositivo, um semi-somador de um bit, é designado por porta OR exclusiva (figura 13).

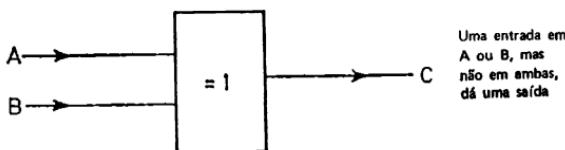


Fig. 13 — Porta Or exclusiva.

### *Somador de um bit*

Devemos considerar agora o problema do transporte do um, como no caso do  $1 + 1 = 0$  transporte 1 e  $1 + 1 + 1 = 1$  transporte 1. Por definição, quando se verifica o transporte no processo de adição, teremos de utili-

zar uma terceira entrada. Na tabela seguinte, C1 representa a terceira entrada, de transporte, e C2 o segundo impulso de saída que irá surgir como dígito de transporte. A gama de combinações de entrada e de saída é apresentada a seguir:

C1	Entradas		Saídas	
	A	B	S	C2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Acrescentando mais portas à porta exclusiva OR, poderemos conceber um somador completo de um bit que corresponderá a estes requisitos. A figura 14 mostra a maneira como este pode ser obtido.

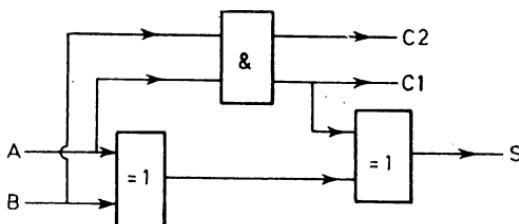


Fig. 14 — Somador binário de um bit.

1. As entradas A e B são aplicadas simultaneamente à porta AND e à primeira porta exclusiva OR.
2. A saída da primeira porta exclusiva OR transforma-se na entrada da segunda porta exclusiva OR.

3. Qualquer saída da porta AND é transportada ao conjunto seguinte de portas, como segunda entrada à segunda porta exclusiva OR, e ainda como uma entrada adicional à porta AND. Esta porta AND dará um impulso de saída se quaisquer dois dos três impulsos de entrada estiverem presentes.

O arranjo da figura 14 constitui um exemplo do circuito necessário para a adição de dois dígitos binários — um somador de um bit. Na prática, estamos a tratar de expressões binárias contendo um certo número de dígitos. Existem duas maneiras de tratar a adição de dois números deste tipo. Um consiste em usar apenas um circuito de soma e alimentar os bits em pares, um de cada número, a partir da direita. Chama-se a isto *adição em série* mas, neste caso, o dígito de transporte deve ser atrasado de um ciclo, de modo a que entre no somador ao mesmo tempo que o par de bits sucessivo. A outra maneira consiste em montar um certo número de portas interligadas, através das quais todos os bits das expressões possam passar simultaneamente. Isto é conhecido pelo nome de *adição paralela*. Um exemplo deste modo de adição, utilizando duas expressões binárias de sete bits, é mostrado nas figuras 15 e 16.

A parte do processador central do computador concebida para realizar as operações aritméticas utilizando estas portas electrónicas é conhecida pelo nome de *unidade aritmética*. Em muitas máquinas, existem localizações especiais de registo, conhecidas pelo nome de *acumuladores*, para as quais são transferidos os números que devem ser tratados nestas operações aritméticas e onde são igualmente guardados os resultados dos cálculos. Por exemplo, pretende-se somar X e Y, que se encontram na memória do computador nos endereços 496 e 723 respectivamente. Primeiramente, copia-se a expressão X, passando esta do endereço 496 para um acumulador e em seguida transporta-se a expressão Y do endereço 723 através da unidade aritmética juntamente com o conteúdo do acumulador (X). A resposta resultante é então colo-

Porta AND			Primeira porta OR exclusiva.			Segunda porta OR exclusiva		
Posição do bit	Entrada		Saída		Saída (saída AND do andar anterior)	Entrada		Saída (saída AND do andar anterior)
	1	2	1	2		1	2	
1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	1	1	1
3	1	1	0	1	1	0	0	0
4	0	0	1	0	0	0	0	1
5	0	1	0	0	0	1	1	0
6	1	1	0	1	1	0	0	0
7	0	0	1	0	0	0	0	1

Fig. 15 — A adição de 110110 e 100100 utilizando uma porta AND e duas portas OR exclusivas.

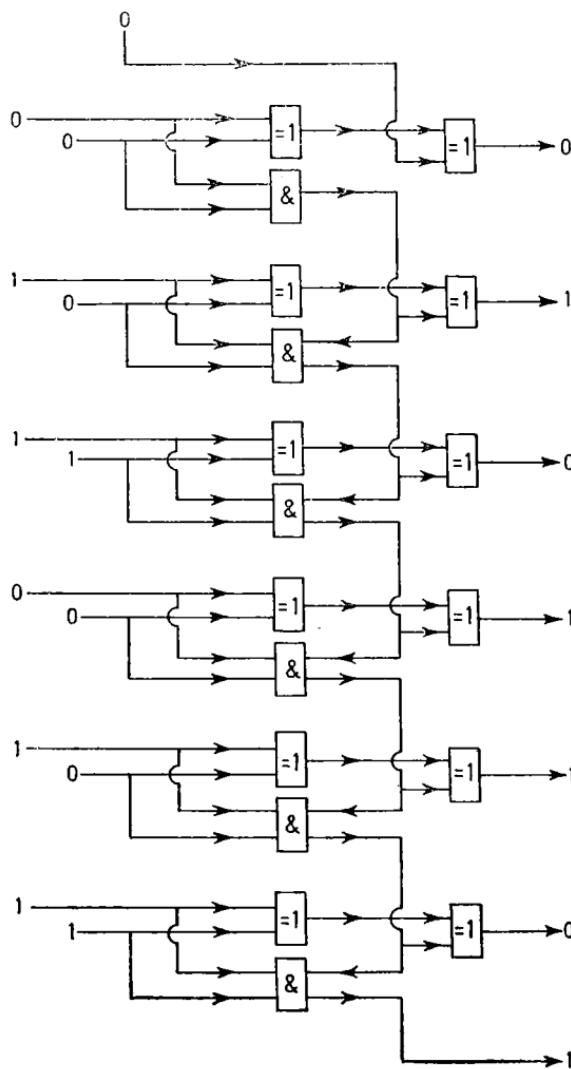


Fig. 16 — Somador de seis bits, ilustrando a adição de 110110 e 100100.

cada no acumulador substituindo a expressão original X. Este total pode então ser transportado para um dos endereços originais, colocado num endereço novo ou retido no acumulador para outros cálculos.

Existem no entanto algumas máquinas que não utilizam acumuladores especiais como tais, antes passando os dados directamente dos endereços de memória, através da unidade aritmética, e movimentando o resultado directamente para um dos endereços originais, «apagando» o dado original. Este modo de funcionamento é designado por «soma à memória».

### *Temporização*

Um computador é uma massa de equipamento electrónico, através do qual se passa constantemente informação, sob a forma de dados e instruções, a velocidades extremamente grandes. Toda esta informação se encontra na forma de impulsos eléctricos, representando dígitos binários. A fim de realizar operações lógicas e aritméticas, é necessário sincronizar o movimento destes bits de tal modo que sejam reunidos na posição correcta nos muitos milhares de endereços de memória existentes, exactamente no momento indicado, medido com um rigor de milionésimos de segundo. Esta temporização é sincronizada por um dispositivo que emite impulsos que viajam através de todas as partes do computador a uma frequência constante durante todo o tempo em que o computador está a trabalhar. O nome dado a este dispositivo é o de *gerador de impulsos de temporização*, assumindo muitas vezes a forma de um oscilador de alta frequência controlado por cristal que pode produzir impulsos quadrados.

A frequência dos impulsos de temporização varia de máquina para máquina. É chamada razão de temporização e encontra-se na região dos vários megaciclos (milhões de ciclos completos) por segundo. O impulso de temporização é a menor unidade indivisível de tempo do computador e pode, pela sua presença, representar um binário

1 ou, através da sua ausência, um valor lógico 0. A razão de temporização determina toda a escala de tempos do funcionamento do computador, demorando cada operação um número bem definido de impulsos. Este intervalo mínimo de tempo entre operações sucessivas de computador é designada por *tempo cíclico* da máquina, um factor que é normalmente referido nas especificações da máquina, fornecidas pelo fabricante. O tempo cíclico é medido em microssegundos ou, em computadores muito rápidos, em nanossegundos.

### *Construção das portas*

As portas constituem de facto um tipo de interruptor, sendo muitas vezes designadas por *interruptores lógicos*. Podem ser construídas com qualquer dispositivo capaz de iniciar ou interromper o fluxo de corrente quando tal se pretende; com efeito, como vimos anteriormente, podem ser construídas com circuitos eléctricos vulgares, utilizando interruptores manuais que serão abertos ou fechados à vontade do utilizador. Em alguns dispositivos iniciais eram usados relés, ou seja, interruptores que podem ser accionados aplicando a corrente de um electroíman. Este tipo de dispositivo electromagnético, apesar de ter bastante segurança de funcionamento, é também bastante lento, e portanto não pode ser usado nas operações de comutação no interior do computador. Alguns dos primeiros computadores usavam válvulas termiónicas para controlar a emissão de corrente, mas no caso dos computadores modernos só se utilizam transístores.

Basicamente, um transístor tem três elementos, um *emissor*, uma *base* e um *colector*. Pode trabalhar a uma tensão muito baixa e, ao contrário de uma válvula, não requer qualquer circuito de aquecimento. Alimenta-se corrente ao transístor através do emissor e obtém-se outra corrente a partir do colector. A base controla o fluxo de corrente entre ambos. Uma corrente aplicada à base

provocará a passagem de uma corrente entre o emissor e o colector, e a ausência dessa corrente na base impedirá a passagem de corrente entre os outros dois elementos. O dispositivo é conhecido pelo nome de transístor *semicondutor*. Em termos bastante simples, se for aplicada uma entrada A à base do transístor, um impulso 1 provocará a passagem de corrente através do dispositivo, enquanto que um impulso 0 a impedirá.

A figura 17 mostra alguns circuitos transistorizados, concebidos para realizar a lógica das portas AND, OR e NOT. O circuito de porta AND contém dois transístores montados em série, o circuito OR dois transístores em paralelo e a acção do transístor único existente na porta NOT consiste em retirar a corrente da saída quanto o transístor está ligado, isto é, quando se encontra no estado condutor.

O tipo de interruptor lógico electrónico que temos estado a discutir é apenas isto, ou seja, um dispositivo que serve para realizar operações de comutação, isto é, emitir um impulso de saída quando lhe é apresentado um determinado conjunto de impulsos de entrada. De modo nenhum armazenará impulsos que representem dígitos binários por qualquer período de tempo. No entanto, os impulsos que são apresentados aos interruptores lógicos devem vir de algum lado, e o único sítio de onde podem vir é algum dispositivo existente no processador central, no qual possam encontrar-se armazenados. Além disso, o dispositivo de armazenamento deve possuir duas qualidades, a saber, (a) a capacidade para emitir impulsos representando a informação binária armazenada e reter esta informação (que de outro modo seria destruída e perdida) e (b) a capacidade para alterar a informação que contém, quando esta já não é necessária.

O armazenamento no processador é basicamente de dois tipos. A maior parte dos registos é concebida para armazenar a massa de informação, dados e instruções de programa, que deve ser utilizada mais cedo ou mais tarde, enquanto que um pequeno número de dispositivos de memória é concebido para guardar informações para

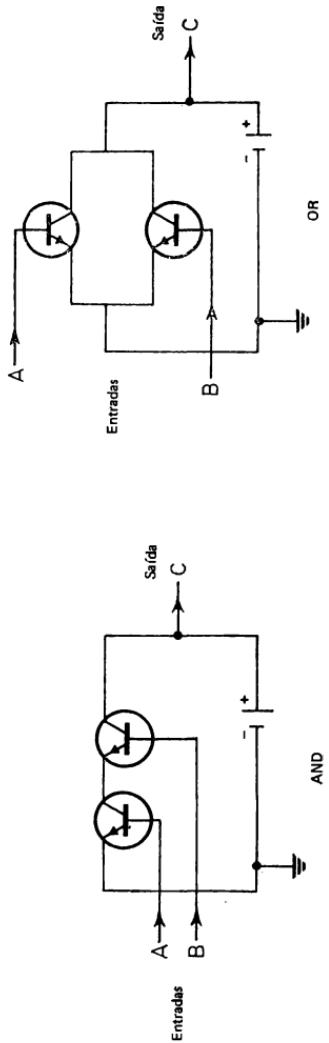


Fig. 17 — Portas transistorizadas AND, OR e NOT.

utilização imediata. Esta técnica será discutida em maior pormenor no capítulo 5, pelo que agora apenas referiremos que a informação que se torna necessária é transferida da memória para um de um pequeno número de registos de trabalho e, quando a operação é completada, transferida de novo para a memória. Na maior parte das máquinas, esta *memória*, como é usualmente chamada, utiliza um princípio de funcionamento magnético — os *núcleos de ferrite* (ver o capítulo 5). Este tipo de memória não permite realizar da maneira mais conveniente todos os pontos mencionados em (a) e (b), e portanto é usado um princípio de funcionamento diferente na construção dos registos de trabalho. Não só estes armazenarão informações sob a forma de uma representação estável de estados 0 e 1 durante um período indefinido, como também actuarão como interruptores electrónicos capazes de emitirem impulsos, quando tal se torna necessário, representando o seu estado actual, e sendo ainda capazes de alterar esse estado sempre que preciso, ou seja, destruindo a informação existente e substituindo-a por nova. Estes dispositivos são designados pelo nome de elementos bi-estáveis ou *flip-flops*.

### *Flip-Flops*

Na sua forma mais simples, este dispositivo terá duas entradas, S e R (ver a figura 18), e duas saídas que designaremos por F e  $\bar{F}$  (F e não F). Uma saída em F será o equivalente do binário 1, enquanto que uma saída em  $\bar{F}$  será equivalente ao binário 0. O aspecto mais importante é que a saída se manterá constante em F ou  $\bar{F}$  (1 ou 0) até ser introduzida uma variação nas entradas S ou R. Por outras palavras, o seu estado será estável, mantendo a mesma informação, até ser introduzida uma variação nas entradas.

Sem entrar nos aspectos detalhados da electrónica empregue na sua construção, um flip-flop poderá ser

constituído por dois transístores ligados por um circuito de realimentação. Isto significa que só um dos transístores estará a conduzir em qualquer momento, sendo o segundo bloqueado através do circuito de realimentação.

Esta realimentação significa que, no caso de ser alimentada uma tensão à base do transístor (a) levando-o a conduzir —no sentido do emissor para o colector—, simultaneamente, através de uma série de resistências,

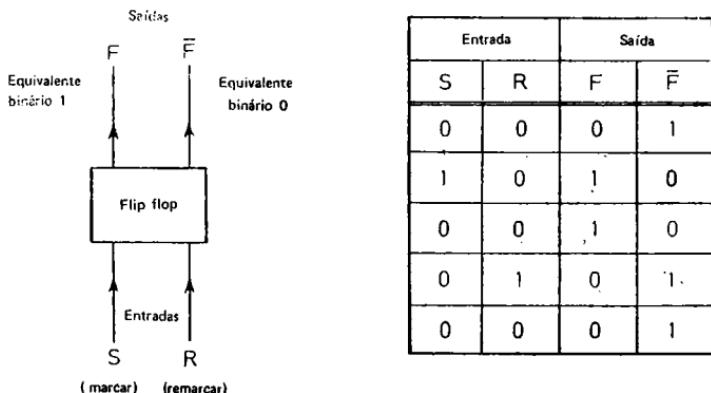


Fig. 18 — Símbolo lógico e tabela de verdade.

essa tensão bloqueará a base do transístor (b), não o deixando conduzir. Inversamente, uma tensão aplicada à base de (b) impedirá a condução em (a). Recordando que está aplicada continuamente ao emissor uma corrente e que a sua passagem para o colector é determinada pelo estado da base, se se pretende que as entradas S e R influenciem o funcionamento do flip-flop, deverão ser ligadas à base dos dois transístores. Prevalecem então as seguintes condições:

1. Partindo do princípio de que  $F = 0$ , se for aplicado um impulso a S ( $S = 1$ ), então a condição será alterada e F transforma-se em 1. Manter-se-á neste estado,

mesmo no caso de S voltar a 0. Isto é conhecido como acção de *marcar*.

2. Se for aplicado um impulso a R ( $R = 1$ ), enquanto  $S = 0$ , então o dispositivo voltará ao seu estado original  $F = 0$ . Isto é conhecido como acção de *remarcar*.

### *Aplicação dos flip-flops*

Como foi sugerido anteriormente, são necessários registos especiais na parte de trabalho do processador central, a fim de guardar informações imediatamente utilizadas. Cada registo é constituído por um certo número de flip-flops, sendo cada um deles capaz de guardar um dígito binário. Segue-se uma listagem de algumas das tarefas que estes registos cumprem.

### *Contador binário*

A contagem é uma técnica utilizada na maior parte dos sistemas de tratamento de dados. No capítulo 7 podem-se encontrar exemplos da sua utilização. Existe um certo número de dispositivos lógicos diferentes que podem ser usados para este objectivo, sendo o apresentado em seguida apenas um exemplo utilizando flip-flops.

Baseia-se num princípio de divisão por dois, o que significa que para cada duas entradas ao circuito lógico contador é dada apenas uma saída. Um conjunto de circuitos deste tipo é ligado de tal modo que a saída do primeiro se transforma na entrada do segundo, a saída do segundo na entrada do terceiro, etc. Deste modo, cada circuito sucessivo representará uma potência de dois cada vez maior, e os respectivos valores representarão uma expressão binária pura. Na figura 19, apresenta-se um elemento único deste tipo de contador, conhecido pelo nome de *divisor binário*. Imagine-se que no flip-flop A,  $\bar{F}$  representa o armazenamento do valor lógico 1. A recicla-

gem deste impulso, coincidente com o impulso seguinte na porta AND 1, permitir-lhe-á emitir um impulso de «marcação» ao flip-flop, que por sua vez emitirá um impulso de saída que se transformará no impulso de entrada do circuito seguinte. A porta AND 2 será bloqueada dado que se  $\bar{F} = 1$ , F deverá ser igual a 0 e portanto a

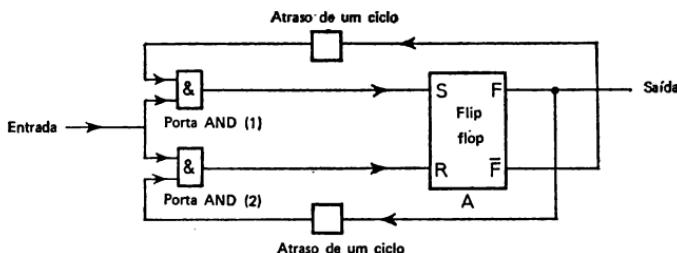


Fig. 19 — Elemento de contagem de «divisão por dois».

entrada a esta porta será 0 e 1. O impulso seguinte levará a porta AND 2 a emitir um novo impulso de marcação que tornará  $F = 0$  e portanto não motivará qualquer novo impulso ao circuito seguinte. Deste modo, cada segundo impulso de entrada dará uma saída.

### *Complementação*

Como já vimos, a complementação constitui uma função necessária em aritmética de computadores para execução das funções de subtracção e divisão. Utiliza o circuito NOT, ou princípio de inversão, convertendo 0s em 1s e 1s em 0s. Mantém-se o problema de somar 1 a fim de obter o complemento verdadeiro. Como já vimos, ao discutir a subtracção complementar anteriormente, o dígito mais significativo do resultado, o algarismo mais à esquerda, é sempre um 1 que deve ser ignorado quando se lê a resposta correcta. Na prática, se este 1 for remo-

vido e transportado de modo a ser somado ao dígito menos significativo, a resposta será então o complemento verdadeiro.

### *Registos de deslocamento*

Um *registro de deslocamento* consiste num certo número de flip-flops interligados, suficiente para acomodar uma palavra, no qual a representação binária pode ser movimentada, ou deslocada, para o flip-flop seguinte sempre pela mesma ordem, podendo este deslocamento ser realizado, flip-flop a flip-flop, ao longo de todo o conjunto destes em caso de necessidade. Quando o conteúdo de um flip-flop é movido para o seguinte, chamaremos ao processo *deslocamento directo*; quando movido para o anterior, chamar-lhe-emos *deslocamento inverso*. Dois exemplos do uso de registos de deslocamento serão a conversão do movimento em série dos bits em movimento paralelo, como se torna muitas vezes necessário nos processos de entrada/saída, e as operações aritméticas necessárias para dividir ou multiplicar um número pela sua raiz. A figura 20 ilustra a diferença da transferência de dados nos modos em série e paralelo.

Na figura 21, apresenta-se um exemplo de circuito de registo de deslocamento. Considere-se que no flip-flop A,  $F = 1$ . Este impulso de saída é submetido, ao mesmo tempo que um impulso de temporização (ou de comando, como também é chamado), a uma porta AND, que por sua vez emitirá um impulso de marcação para A + 1, tornando F, neste segundo flip-flop, igual a 1. Dado que F = 0, não será emitido qualquer impulso de remarcação a partir da outra porta AND. Reciprocamente, se  $\bar{F}$  em A for igual a 1, então F deverá ser igual a 0. Neste caso, é emitido um impulso de remarcação a partir da segunda porta AND, tornando F em A + 1 igual a 1 e portanto F = 0. São incorporados elementos de atraso entre o flip-flop e as portas AND, permitindo que o deslocamento

de um impulso seja efectuado antes de o impulso seguinte ser inserido em A.

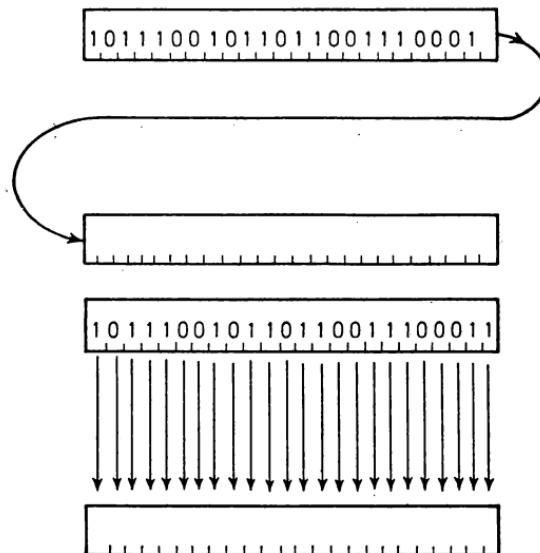


Fig. 20 — Transferência de palavras em série e em paralelo.

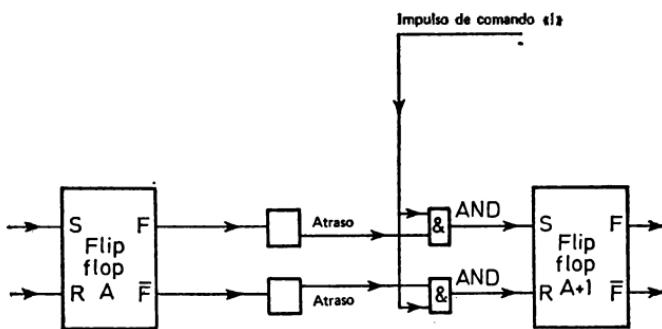


Fig. 21 — Elemento de registo de deslocamento.

## *Codificação decimal/binária*

Na maior parte dos casos, os dados primeiramente comunicados a um computador são expressos em termos decimais utilizando os símbolos 0 a 9 ou uma representação destes símbolos. No caso dos cartões perfurados, são representados por orifícios em posições bem definidas e, no caso da entrada através de teclas, por teclas específicas.

Uma maneira de converter a representação decimal para binária consiste em passar o impulso criado ao sentir um orifício ou premir um tecla, através das portas que representam valores binários. A saída destas portas é então movimentada para registo de flip-flops que conterão agora a representação binária dos dados de entrada, prontos para transferência para endereços de memória. A figura 22 mostra em princípio a maneira como isto é feito. Para obtenção de saídas, existem descodificadores de binário para decimal que realizam a mesma tarefa. Neste caso, os dados são movimentados da memória para registo de flip-flops, cujas saídas são submetidas simultaneamente a um certo número de portas, sendo cada uma delas capaz de aceitar apenas o padrão binário do dígito decimal representado (figura 23).

## *Instruções de descodificação*

Vimos anteriormente que cada função que o processador deve realizar é identificada por um dado padrão binário, permanentemente armazenado na memória. Vejamos agora a maneira como a máquina escolhe o circuito necessário para realizar uma dada função.

Como exemplo simples, consideremos apenas quatro funções, sendo cada uma delas representada por um código binário de dois dígitos da seguinte maneira:

Sumar	00
Subtrair	01
Dividir	10
Multiplicar	11

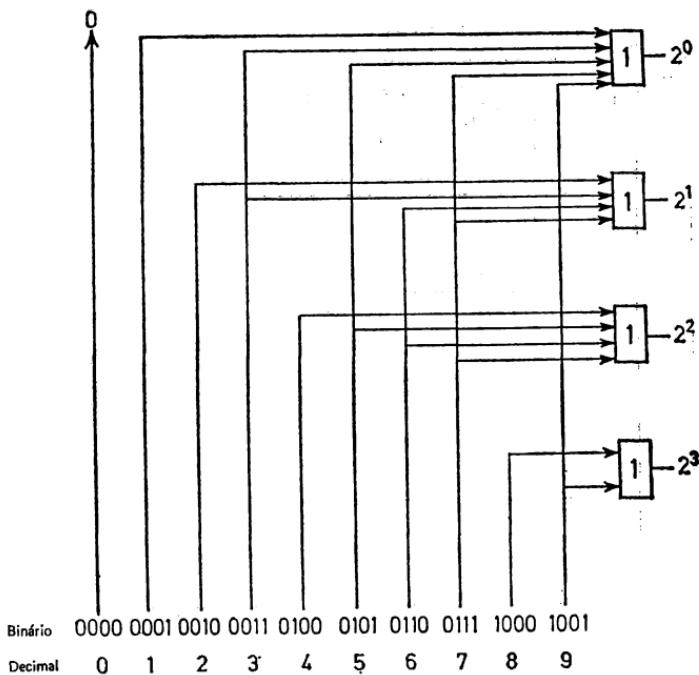


Fig. 22 — Codificação decimal/binário.

O código binário é marcado num registo, neste caso de dois flip-flops, e as saídas dos flip-flops conduzem a quatro portas AND, cada uma delas controlando uma das anteriores quatro funções.

Se, por exemplo, os flip-flops contêm o código de divisão — 10 — então a saída de A será F (binário 1) e a saída de B,  $\bar{F}$  (binário 0) (figura 24). Estes dois impulsos coincidirão apenas numa das portas AND, a qual controlará a operação de divisão. Por sua vez, esta operação levará a porta a emitir um impulso que activa o circuito necessário.

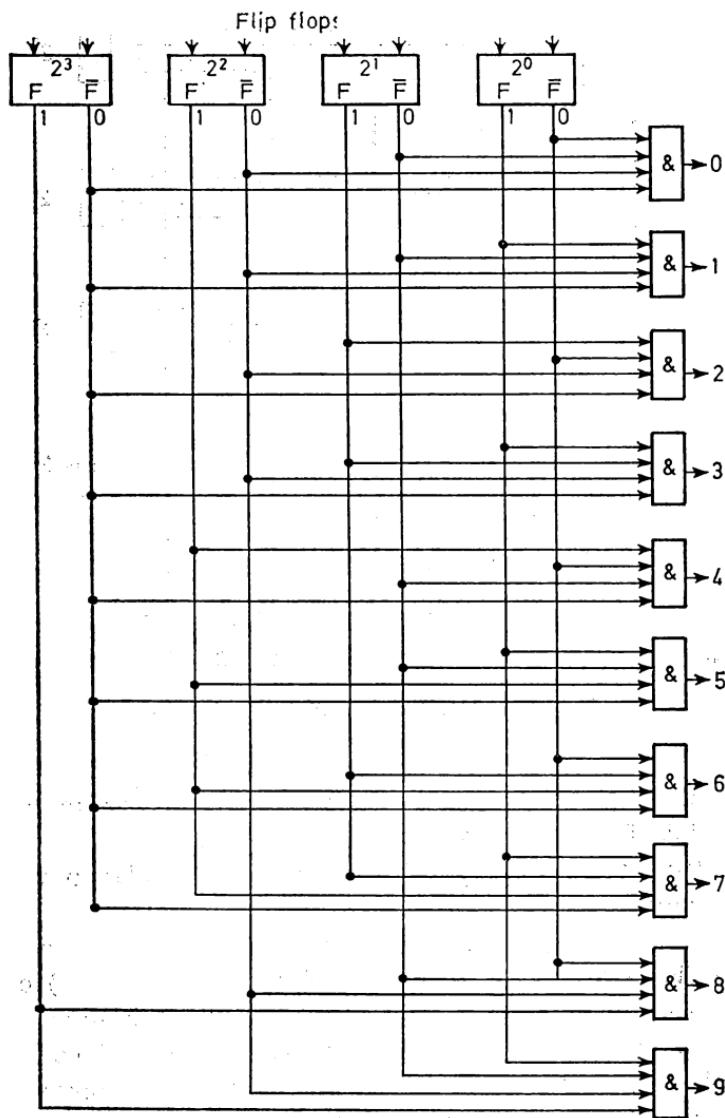


Fig. 23 — Descodificação binário/decimal.

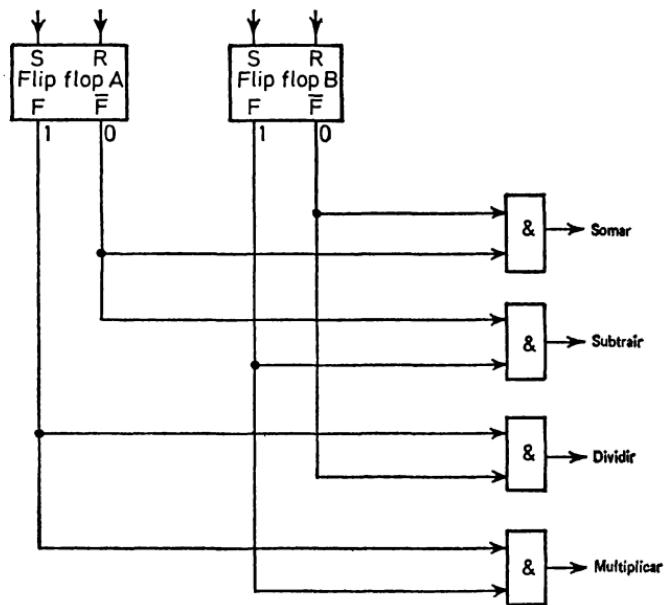


Fig. 24 — Descodificação das instruções.



## 4

### ENTRADAS E SAÍDAS DE COMPUTADORES

Vimos anteriormente que o «cérebro» de um computador é o seu processador central. Este processador realiza um certo trabalho sobre dados que lhe são submetidos, armazena os dados em que está a trabalhar e armazena também o programa que determina os processos a realizar. Realiza ainda uma função de controlo, regulando o caudal de dados e o seu tratamento, fazendo tudo isto a velocidades incrivelmente altas.

O objectivo deste capítulo consiste em considerar a maneira como estes dados, com os quais trabalha o processador central, são conduzidos até ele, e rever a maneira como os resultados do tratamento são comunicados ao exterior. Chamaremos a estes dois elementos as funções de *entrada* e *saída* de um computador.

#### *Dispositivos e meios de entrada*

Em última análise, o processador central de um computador digital só pode aceitar, compreender e manipular informação que lhe seja apresentada na forma digital e, como já vimos, isto significa uma série de

dígitos binários. No que toca ao computador, estes dígitos binários são-lhe conduzidos sob a forma de uma série de impulsos eléctricos (figura 25).



Fig. 25 — Corrente de dígitos binários.

É no entanto difícil pensar em quaisquer dados obtidos nesta forma e apresentados tal qual ao computador. Torna-se assim necessário construir um dispositivo, ligado ao processador, capaz de aceitar os dados numa outra forma e de os converter numa série de impulsos binários que o processador central possa aceitar. Vimos no capítulo 1 que estes são conhecidos como dispositivos de entrada e fazem parte do grupo de máquinas que suportam o processador e são conhecidas por *periféricos*.

No entanto, se pensarmos no assunto durante uns momentos, verificaremos que o dispositivo de entrada é, em si, apenas uma máquina e só será capaz de aceitar dados na forma para a qual foi concebido. Uma máquina de escrever, por exemplo, só aceitará dados se premirmos as suas teclas. Não vale a pena falar-lhe e esperar que ela escreva o que lhe dizemos, ou mostrar-lhe um conjunto de notas manuscritas e esperar que ela automaticamente as transcreva. Como é evidente, os dados não são necessariamente originados numa forma aceitável pelo dispositivo de entrada. Devemos portanto interpor entre os dados originais e o dispositivo de entrada algum mecanismo de conversão que os torne aceitáveis. Poderíamos, assim, conceber os estados seguintes, conducentes

ao ponto em que os dados são lidos para a memória do computador:

**OCORRÊNCIA**



**REGISTO ORIGINAL**

**DA OCORRÊNCIA** —————→ **DOCUMENTAÇÃO DE FONTE**



**CONVERSÃO PARA UMA**

**FORMA ACEITAVEL**

**PELA MÁQUINA** —————→ **PREPARAÇÃO DOS DADOS**



**DISPOSITIVOS DE**

**ENTRADA** —————→ **ENTRADA DE DADOS**



**COMPUTADOR**

Como veremos, alguns meios de entrada utilizarão todos estes dados, enquanto outros são concebidos para curto-circuitar parte deles. Isto tornar-se-á mais aparente à medida que examinarmos os métodos de entrada um a um. No entanto, um outro método que deveremos observar antes de os discutirmos em pormenor é que, essencialmente, são utilizados dois tipos distintos de dispositivo para transferir dados para o processador. Um destes será constituído pelos dispositivos de armazenamento e o outro pelos dispositivos de entrada que estamos agora a considerar. Antes de os dados poderem ser mantidos num dispositivo de armazenamento, devem ser primeiramente lidos através de um dispositivo de entrada. Estes dispositivos de armazenamento estarão relacionados com a entrada primária.

Falando em geral, os dispositivos de entrada podem ser grosseiramente subdivididos nas seguintes categorias:

1. Leitores de documentos
2. Dispositivos de teclas
3. Meios magnéticos
4. Dispositivos ópticos e acústicos.

### *Leitura de documentos*

Já vimos anteriormente que os dados registados na sua forma inicial podem não ser apropriados para entrada directa através de um dispositivo de leitura ao processador, pelo que pode tornar-se necessário algum estado intermédio de preparação destes dados. Talvez possamos agrupar os documentos nas seguintes três categorias:

- a) Caso em que não é aceitável pela máquina nem o documento original nem o modo como os dados são registados. Por exemplo, uma anotação manual num livro de contas. Neste caso, os dados devem ser transcritos para um meio e numa forma que sejam aceitáveis. A maneira mais vulgar de o fazer consiste em utilizar um meio do tipo dos cartões ou fita perfurada.
- b) Caso em que o registo original e o modo de registo são ambos aceitáveis pela máquina, como acontece com os caracteres em tinta magnética e caracteres ópticos.
- c) Caso em que o documento original é aceitável mas a forma de registo não, o que envolve a conversão para uma forma utilizável pela máquina, mas usando o documento original. Um exemplo disto consiste em usar técnicas de sensibilização a marcas em cartões perfurados.

Naturalmente, o número de estados através dos quais os dados deverão passar antes de atingirem eventualmente o computador dependerá dos três factores sublinhados. Por exemplo, os dados registados num documento

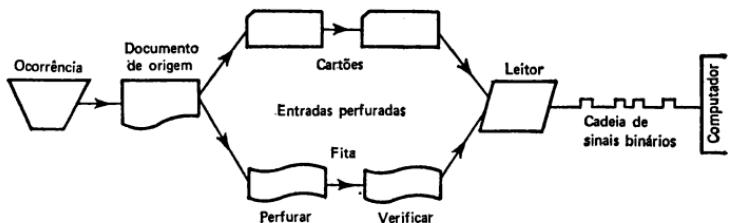


Fig. 26 — Conversão da fonte para a máquina.

escrito à mão poderiam ter de passar por todos os estados ilustrados na figura 26, antes de se iniciar o seu processamento.

### *Cartões perfurados*

Um cartão perfurado é um pedaço de cartão de alta qualidade, fabricado com uma dimensão exacta e uma espessura bem definida. O rigor destas dimensões é determinante no que se refere à sua aceitação por uma máquina de leitura de cartões.

Se bem que tenham sido desenvolvidos alguns tipos e capacidades de cartões diferentes, o mais usado actualmente em trabalho de computadores é o conhecido «cartão de 80 colunas» (figura 27).

O cartão encontra-se dividido em oitenta colunas verticais, tendo cada coluna 12 posições nas quais se podem abrir orifícios. Cada coluna do cartão pode registrar um carácter, dígito, letra ou símbolo, através de um padrão de um, dois ou três orifícios, único para cada carácter. Os padrões de orifícios representando caracteres são conhecidos pelo nome de *código de per-*

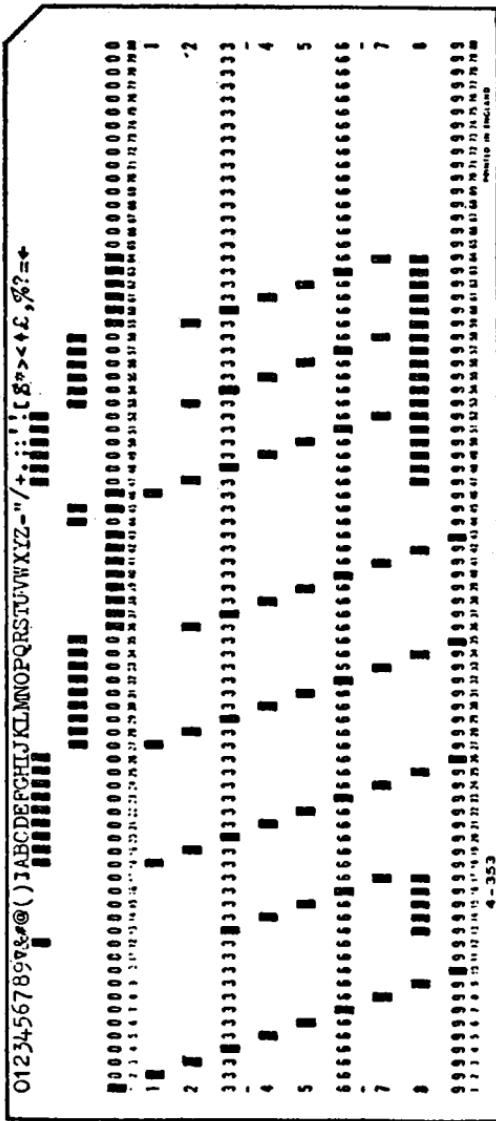


Fig. 27 — Cartão perfurado de oitenta colunas.

furação. Este código varia um pouco de sistema para sistema.

Para registar os dígitos 0 a 9, é perfurado um único orifício na posição conveniente, como se mostra nas colunas 1 a 10 da figura 27. A representação de letras é um pouco mais difícil, dado que existem 26 no nosso alfabeto (considerados o K, o W e o Y) mas apenas 12 posições de perfuração em cada coluna. Este problema é resolvido dividindo o alfabeto em três grupos, A—I, J—R e S—Z. O grupo em que se encontra a letra a representar é indicado por um orifício numa das três posições superiores de perfuração, enquanto que um segundo orifício em qualquer das restantes nove posições indica a posição da letra no interior do seu grupo. Isto é ilustrado pelas colunas 18—43 da figura 27. O grupo A—I é indicado por um orifício na posição mais acima, J—R na seguinte e S—Z na posição O.

Não se pode registrar apenas informação numérica e alfábética, mas também uma gama de símbolos, tendo cada um deles o seu padrão único de orifícios. Nas colunas 11—17 e 44—64 da figura 27 apresentam-se exemplos.

Como veremos, o princípio básico utilizado nos cartões perfurados consiste em dar a cada posição vertical um dado valor previamente determinado e codificar toda a informação que desejamos apresentar em termos destes valores. O mesmo princípio se aplica às 80 colunas, ao longo de todo o comprimento do cartão. Para dar algum sentido aos orifícios abertos, devemos atribuir a cada coluna um dado significado prévio. Por exemplo, se desejamos representar as seguintes informações num cartão:

Número de referência	Descrição	Quantidade	Valor
devemos atribuir um número fixo de colunas para representação do número de referência, um outro número bem determinado para conter a descrição, etc. Estes grupos de colunas existentes nos cartões são conhecidos pela			

designação de *campos*, e distinguem-se normalmente entre si por linhas verticais impressas no cartão e um cabeçalho de campo, por exemplo «Descrição», igualmente impresso na parte superior do cartão.

O conjunto da informação mencionado é conhecido pelo nome de *registo de dados*. Quando um cartão contém

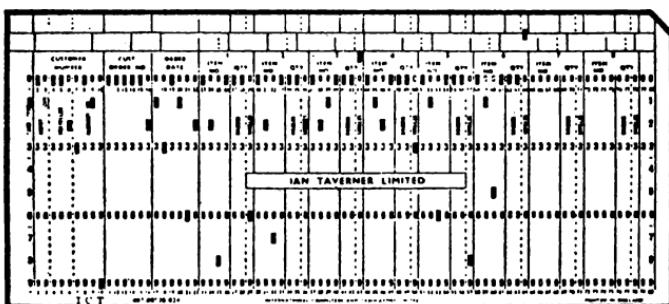


Fig. 28 — Um cartão de registo múltiplos. (International Computers, Ltd.).

apenas um destes registo, é normalmente conhecido por um *cartão de registo único*. Em certas circunstâncias, no entanto, pode tornar-se possível manter dois ou mais registo num único cartão, caso em que nos referimos a este como *cartão de registo múltiplos*. Na figura 28 apresenta-se um exemplo destes cartões.

#### *Preparação dos cartões perfurados*

Existem duas maneiras principais de preparar os cartões para constituírem entradas adequadas. A primeira destas surge quando a informação original é preparada de uma forma não aceitável pela máquina, como acontece no caso da maior parte dos registo manuscritos. Neste caso, a preparação envolve a leitura pelo operador do documento de origem e a entrada desta informação atra-

vés do teclado de uma máquina perfuradora de cartões. São automaticamente alimentados cartões virgens a partir de um depósito acoplado à máquina em causa, sendo cada um deles perfurado de acordo com a informação batida no teclado e em seguida ejectado quando a sua perfuração se encontra terminada. Para excluir o uso de um documento manuscrito a fim de registar as informações originais, em alguns sistemas os cartões são concebidos de modo que o registo original seja escrito no próprio cartão, lido deste pelo operador com o cartão já inserido na máquina, sendo então os dados batidos no teclado e perfurados no mesmo cartão. Os cartões usados deste modo são designados por *cartões de dupla função*.

Para eliminar este processo bastante laborioso de registar e em seguida transcrever para uma forma perfurada, foi desenvolvida uma técnica conhecida por *sensibilização por marcas*. Trata-se novamente de um processo de registo de dados no próprio cartão, mas em vez de utilizar os caracteres manuscritos normais, os dados são indicados realizando, no cartão, uma marca com um lápis de grafite na posição requerida. Depois de isto ser feito, o cartão é passado por uma máquina de perfuração a alta velocidade que «lerá» a posição das marcas e perfurará o cartão nos mesmos pontos. Se bem que este método tenha a vantagem de eliminar a necessidade de perfurar o cartão manualmente, na prática a sua aplicação é limitada à informação numérica que envolva apenas uma marca em cada coluna.

#### *Rigor dos cartões perfurados*

É evidente que o rigor total constitui uma característica essencial de qualquer meio de entrada de um computador, senão, como é evidente, os resultados obtidos no processamento dos dados não serão de confiança. A fim de assegurar o rigor em causa, é realizado um processo de «verificação» na preparação dos cartões perfurados. Estes, tendo sido perfurados inicialmente a partir

da informação original, são então passados a um segundo operador que repete a operação voltando a perfurar o cartão a partir do mesmo original. No caso de existirem duas perfurações que não correspondam, ou seja, se na segunda operação se tentar realizar uma perfuração num ponto onde ainda não existe qualquer orifício, ou se não for ensaiada uma perfuração num ponto onde já existe um orifício, a máquina assinalará a existência de um erro. A razão desta diferença entre as duas perfurações deve então ser verificada, preparando-se um novo cartão sem erro. Este cartão corrigido deve por sua vez ser verificado de modo a assegurar o rigor da nova operação.

#### *Leitura de cartões perfurados*

Tendo sido verificado o rigor dos cartões, a informação neles contida deve então ser transferida para o computador. Isto é feito através de um dispositivo da máquina, ligado directamente ao processador central, conhecido pelo nome de *leitor de cartões*. Neste, os cartões são passados a uma velocidade bastante grande entre uma fonte de luz e uma fila de células foto-eléctricas, uma para cada posição vertical do cartão. Chama-se-lhe *posição de leitura*. Quando existe um orifício, a luz penetrará no cartão e provocará o aparecimento de uma carga na correspondente célula foto-eléctrica. A posição e o padrão das células em que se verifica o aparecimento de cargas são reconhecidos pela máquina e traduzidos por impulsos electrónicos apropriados em codificação binária para transmissão à memória do computador.

Se bem que a verificação anterior tenha assegurado o rigor da informação contida nos cartões, devem-se agora adoptar medidas que garantam que a leitura é igualmente correcta. A maneira de o fazer consiste normalmente em utilizar uma segunda posição de leitura, conhecida pelo nome de *posição de verificação*. Isto significa que os cartões são lidos duas vezes durante o seu progresso através da máquina, sendo as duas leituras comparadas e os dados

fornecidos à memória do computador, a menos que as leituras não sejam idênticas. Uma maneira de realizar esta comparação consiste em contar o valor total de todos os orifícios existentes no cartão na primeira fase de leitura e colocar este total num registo. Na fase de verificação é realizada uma contagem semelhante e o seu resultado é igualmente colocado no registo da máquina com um sinal menos. O conteúdo do registo só é passado para o processador central no caso de ser obtido um total de zero das duas leituras.

As velocidades de leitura dos cartões variam de máquina para máquina, desde cerca de 300 cartões por minuto nos tipos mais antigos e mais lentos, até mais de 1500 cartões por minuto no caso das máquinas mais recentes. Uma velocidade média de leitura será de cerca de 900 cartões por minuto, dando uma velocidade de leitura de caracteres de cerca de 1200 por segundo (c.p.s.).

#### *Fita de papel perfurada*

Este meio de entrada utiliza o mesmo princípio de funcionamento para registo de dados, mas usa uma fita contínua de papel em vez de cartões separados.

Cada carácter é representado por um padrão único de orifícios, conhecido por *armação*, disposto transversalmente na fita de papel. O número de orifícios que podem ser abertos transversalmente na fita varia de sistema para sistema. Cada posição de perfuração ao longo do comprimento da fita é conhecida por *banda*. Assim, uma fita de cinco bandas pode acomodar cinco orifícios em largura. A maior parte dos sistemas modernos de fita perfurada utilizam uma fita de oito bandas que tem a largura de 25,4 mm. Destas oito bandas, sete são usadas para registar caracteres e a oitava, conhecida pelo nome de *banda de paridade*, é usada para verificação. Na figura 29, mostra-se parte de uma fita de papel de oito bandas. O número de diferentes combinações de orifícios que podem ser abertos em sete bandas é de 128, o que

permite o uso de um código de perfuração de 128 caracteres. A figura 30 mostra um código de perfuração de 128 caracteres em oito bandas.

Talvez seja conveniente neste ponto explicar o que é entendido por «paridade» no que se refere ao registo e armazenamento de dados em forma binária. À medida que os dados são movimentados num computador e do processador para os periféricos, existe sempre a possibilidade de erro devido à perda de um bit de dados ou o seu



Fig. 29 — Fita de papel perfurada com oito bandas.

aparecimento num ponto onde não se devia encontrar. Para dar algum elemento de salvaguarda no caso de isto se verificar, todas as afirmações binárias são exprimidas de tal modo que cada uma delas tenha um número par ou ímpar de bits. Por exemplo, se for usada uma paridade par, quando existe um carácter constituído por um número ímpar de bits, é acrescentado um bit adicional que tornará o total par. Isto é ilustrado na figura 29, onde se pode notar que a letra A tem dois bits, a letra B igualmente dois, mas a letra C três bits e um quarto acrescentado na banda de paridade de modo a obter um resultado par. Durante o processamento, de cada vez que um grupo de bits é movimentado, realiza-se uma operação de contagem. No caso de se optar por uma paridade par,

se o resultado da contagem for ímpar, é assinalado um estado de erro. O bit de paridade, ou orifício de paridade no caso da fita de papel perfurada, é acrescentado automaticamente quando a fita é perfurada.

#### *Preparação da fita de papel perfurada*

Esta preparação é realizada através de um perfurador de fita em papel, uma máquina à qual é alimentada fita virgem a partir de uma bobina, sendo esta passada através de uma posição de perfuração e daí para uma nova bobina de recepção. O teclado contém teclas que representam um conjunto total de caracteres semelhante ao de uma máquina de escrever. Quando se prime uma tecla, a máquina escolhe automaticamente o padrão correcto de orifícios e perfura a fita de maneira apropriada.

Tal como no caso dos cartões perfurados, existe um controlo de rigor, uma vez mais designado por verificação. Em princípio, este controlo é semelhante, sendo realizada uma segunda operação de perfuração do mesmo material por um segundo operador, mas neste caso a segunda perfuração é realizada num novo comprimento de fita. A fita originalmente perfurada e uma nova fita virgem são alimentadas simultaneamente ao verificador e, à medida que se prime cada tecla, realiza-se uma comparação com a fita originalmente perfurada. Se as duas acções coincidirem, a nova fita é perfurada. Se assim não acontecer, é fornecido um sinal de erro.

#### *Modo de registo*

Como vimos anteriormente, os campos e registos num cartão perfurado têm um comprimento definido. O número de caracteres que o cartão conterá está fixado em oitenta, e a dimensão de cada campo é previamente determinada quando o cartão é concebido. Esta consideração não se aplica à fita de papel, dado que é contínua.

No entanto, pode ser desejável tratar a fita da mesma maneira que o cartão, atribuindo um número fixo de caracteres para cada campo, se bem que isto possa provocar uma perda de espaço. Se, por exemplo, for neces-

		< Banda nº		< Valor de banda				
						Significado		
Zona zero	O	Tc	Controllo de transmissão		O	SPACE		
	O	SOH	Início de título		O	! Exclamação		
	●	STX	Início de texto		●	" Citação		
	O	ETX	Final de texto		O	# Número		
	●	EOT	Final de transmissão		●	€ Libra		
	O	ENQ	Inquérito		●	% Por cento		
	O	ACK	Tomar conhecimento		●	• E		
	●	BEL	Campainha, alarme (soar)		●	' Apóstrofo		
	●	BS	Espaço atrás		●	( Parêntesis esquerdo		
	●	HT	Tabulação horizontal		●	) Parêntesis direito		
	●	FE <sub>1</sub>	Efectuador de formato		●	* Asterisco		
	●	LF	Alimentação de linha		●	. Mais		
	●	FF	Alimentação de forma		●	, Comas		
	●	CR	Retorno da cabeça		●	- Hifen ou menos		
	●	SO	Deslocamento (saída)		●	. Período		
		SI	Deslocamento (entrada)		●	/ Barra		
Zona um	●	OLE	Escapamento de ligação de dados		●	0		
	●	DC	Controllo de serviço		●	1		
	●	DC <sub>1</sub>			●	2		
	●	DC <sub>2</sub>			●	3		
	●	DC <sub>3</sub>			●	4		
	●	DC <sub>4</sub>			●	5		
	●	NACK	Conhecimento negativo		●	6		
	●	SYNC	Pausa síncrona		●	7		
	●	E1B	Final de bloco de transmissão		●	8		
	●	CNCL	Cancelar		●	9		
	●	EM	Final do meio de entrada		●	:	Dois pontos	
	●	SS	Início de sequência especial		●	;	Ponto e vírgula	
	●	ESC	Escapamento		●	<	Menor do que	
	●	FS	Separador de fila		●	=	Igual	
	●	GS	Separador de grupo		●	>	Maior do que	
	●	RS	Separador de registo		●	?	Pergunta	
	US	Separador de unidade						
Zona três								
		</td						

sário um campo máximo de cinco caracteres, dando uma gama de quantidades até 99999, uma quantidade real de, por exemplo, 23 teria de ser expressa acrescentando três zeros sem qualquer significado, ou seja, pela expressão

Zona quatre		< Banda nº < Valor de banda		Zona seis	
Significado				Significado	
● ●	○	● E m		● ●	○
●	○	● A		● ●	○
●	○	● U		● ●	○
● ●	○	● C		● ●	○
●	○	● D		● ●	○
● ●	○	● E		● ●	○
● ●	○	● F		● ●	○
●	○	● G		● ●	○
● ●	○	● H		● ●	○
● ●	○	● I		● ●	○
● ●	○	● J		● ●	○
●	○	● K		● ●	○
● ●	○	● L		● ●	○
●	○	● M		● ●	○
● ●	○	● N		● ●	○
● ●	○	● O		● ●	○
● ●	○	P		● ● ●	○
● ●	○	Q		● ● ●	○
● ●	○	R		● ● ●	○
● ●	○	S		● ● ●	○
● ●	○	T		● ● ●	○
● ●	○	U		● ● ●	○
● ●	○	V		● ● ●	○
● ●	○	W		● ● ●	○
● ●	○	X		● ● ●	○
● ●	○	Y		● ● ●	○
● ●	○	Z		● ● ●	○
● ●	○	l	Paréntesis recto esquierdo	● ● ●	○
● ●	○	đ	Dólar	● ● ●	○
● ●	○	l	Paréntesis recto derecho	● ● ●	○
● ●	○	l		● ● ●	○
● ●	○	-		● ● ●	○
Zona cinco					
Orifícios de engate				Orifícios de engate	

00023. No entanto, esta maneira de encarar o problema tem a vantagem de se poder especificar à máquina o número de caracteres de cada campo, e o número de campos em cada registo, pelo que ela saberá exactamente onde começar e terminar. Este tipo de registo é conhecido pela designação de *formato fixo de comprimento de campo*.

Por outro lado, podemos perfurar na fita o número exacto de caracteres que de facto surgem no campo, ou seja, no exemplo anterior, os dígitos 23, se bem que da próxima vez que este campo surja possa ter um número diferente de dígitos, por exemplo 1234. Se for utilizado este método, segue-se que se torna necessário identificar na fita onde se inicia e termina cada campo e registo. Isto pode ser feito terminando cada grupo com um padrão especial de orifícios designados por *marcador*. Como é evidente, serão utilizados diferentes marcadores para indicar o final de um campo e o final de um registo. A utilização da fita deste modo é designada por *formato variável de comprimento de campo*. Na prática, pode tornar-se vantajoso combinar ambos os formatos, particularmente quando certos campos, por exemplo um número de referência, são fixos por natureza e outros, por exemplo uma descrição, são variáveis. Este modo de registo é designado por *formato fixo-variável de comprimento de campo*. Uma vantagem que convirá mencionar, na utilização de um formato de comprimento fixo, é que permite a verificação de validade. Ou seja, dado que a máquina conhece quantos caracteres surgirão em cada campo, pode indicar um estado de erro se, ao contá-los, o número não corresponde.

#### *Leitura de fita de papel perfurada.*

A leitura é realizada através de um leitor de fita perfurada, consistindo basicamente num mecanismo de bobina de alimentação que movimenta a fita através da máquina, e uma posição de leitura. A fita passa entre uma fonte de luz e uma fila de células fotoeléctricas, uma

para cada banda da fita. São produzidos impulsos nas células sempre que a luz penetra na fita e convertidos na codificação binária relevante para armazenamento no computador. As velocidades de leitura variam consideravelmente com os diferentes tipos de leitor de fita desde cerca de 1000 a 2000 c.p.s.

Como vimos anteriormente, o uso de cartões e fita de papel perfurados como entradas envolverá normalmente duas fases distintas:

1. Preparação da documentação de fonte.
2. Transferência destes dados para o cartão ou a fita.

Este processo é caro e laborioso, e assim, numa tentativa de eliminar a primeira destas duas fases, foram desenvolvidas técnicas para registar dados na fonte de uma maneira comprehensível pela máquina.

As técnicas mais vulgarmente usadas são:

«Reconhecimento de caracteres em tinta magnética» (m.i.c.r., em inglês).

«Reconhecimento de caracteres ópticos» (o. c. r., em inglês).

#### *Reconhecimento de caracteres em tinta magnética*

Este método utiliza o princípio seguinte: se um meio condutor for passado através de um campo magnético, será induzida uma corrente eléctrica no meio proporcionalmente à força do campo magnético existente. Assim, se imprimirmos um carácter com uma tinta contendo uma substância ferro-magnética, a magnetizarmos e a passarmos sobre um fio, será induzida uma corrente neste fio com uma forma de onda que reflectirá o formato do carácter.

Os caracteres mais vulgares usados em muitos países para este efeito são conhecidos pela designação E13B e

podem ser encontrados na parte inferior dos cheques. Na figura 31 são ilustrados estes caracteres.

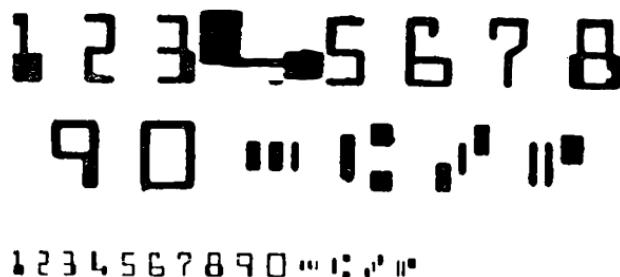


Fig. 31 — Caracteres M.I.C.R.

*Preparação de documentos m.i.c.r.*

As considerações básicas na preparação do m.i.c.r. são:

1. Dado que o documento de fonte constitui igualmente a forma de entrada, a sua dimensão deve manter-se dentro das tolerâncias impostas pela máquina usada para o seu processamento.
2. A leitura com êxito dos caracteres depende da reprodução rigorosa do formato do carácter e de uma regularidade da densidade da tinta. Isto obriga a realizar uma impressão de qualidade óptima.
3. Os caracteres devem surgir numa posição predefinida no documento, a fim de assegurar a sua posição correcta na cabeça de leitura.

Existem duas fases na codificação dos caracteres em tinta magnética no documento. A primeira delas verifica-se quando o documento é originalmente impresso. Esta é limitada a dados não variáveis, como seja, no

campo dos cheques, o número de referência da filial, o número de cheque e a referência da conta do cliente. Chama-se-lhe *pré-codificação*.

A segunda fase consiste em captar os dados variáveis. No caso dos cheques, estes serão constituídos pela quantidade a pagar e pela entidade a quem a importância deve ser paga, ambos escritos pelo utente da conta. Chama-se a isto *pós-codificação*. O método usado consiste em passar os cheques um a um por uma máquina de teclado, onde cada cheque se mantém visível enquanto o operador lê os dados e os escreve. O cheque é então passado a uma posição de impressão, na qual os caracteres são reproduzidos utilizando tinta magnetizável.

#### *Leitura de caracteres m.t.c.r.*

O elemento ferro-magnético existente na tinta é magnetizado, e os caracteres são passados sobre uma cabeça de leitura onde o magnetismo induz uma corrente no circuito de leitura. Esta corrente variará com o formato do carácter. Comparando o padrão da corrente induzida com as ondas «standard» mantidas no circuito de leitura, aquele poderá ser reconhecido como representando determinado carácter, sendo então produzidos os impulsos binários apropriados. As velocidades de leitura variarão novamente conforme a máquina e dependerão, como é evidente, do número de caracteres existentes no documento. Uma velocidade razoável de leitura será de cerca de 1200 documentos por minuto, e presumindo que existem, por exemplo, 75 caracteres em cada documento, obteremos uma velocidade de leitura de 1500 c.p.s.

#### *Reconhecimento de caracteres ópticos*

Este modo de registo é bastante mais popular e está muito mais generalizado que o anterior. O princípio usado aqui consiste em «varrer» cada carácter com uma

fonte de luz e dirigir a luz reflectida através de um sistema de lentes para uma célula fotoeléctrica. A forma de onda da corrente assim induzida é usada para identificar o

1234567890-\*

1234567890-\*

Fig. 32 — Caracteres O.C.R.

carácter de uma maneira bastante semelhante à usada no m.i.c.r.

Os caracteres o.c.r. não são tão estilizados como os caracteres m.i.c.r. e são mais fáceis de ler pela vista humana. O tipo o.c.r. n.º 2 é apresentado na figura 32.

#### *Documentos o.c.r.*

Dado que aqui nos encontramos perante um sistema de reconhecimento de caracteres que se fundamenta apenas na forma do carácter e não nas propriedades da tinta usada, temos uma situação em que o próprio computador, através dos seus periféricos, pode preparar os documentos. Estes documentos podem, por sua vez, ser reciclados através do sistema do computador numa nova fase de processamento. Um exemplo disto será uma conta de electricidade. Esta será então preparada pelo computador, impressa em caracteres ópticos, e enviada ao cliente com um pedido de que seja devolvida uma parte da factura quando for realizado o pagamento. Esta parte, que muitas vezes tem escrito «não dobrar por favor», pode ser alimentada directamente, através de um leitor o.c.r., ao computador de modo a registar o pagamento.

Outra aplicação muito importante no campo da leitura óptica pode ser encontrada nas técnicas de sensibilização por marcas. Um dos métodos utilizados é bastante

semelhante ao descrito anteriormente para os cartões perfurados. São impressas colunas verticais dos dígitos 0 a 9 num formulário e uma marca é realizada nos dígitos de modo a representar um dado número. Um segundo método, de introdução mais recente, baseia-se no princípio de escolha entre várias respostas, ou seja, selecção de factores previamente determinados. Consideremos por exemplo a pergunta seguinte:

Quantos segundos existem num minuto: 15, 120, 37, 60, 210?

A resposta pode ser dada colocando um X sobre o valor correcto ou sublinhando-o. Teremos então uma situação sim-não para todas as variáveis referidas: não, não, não, sim, não. É agora apenas necessário um leitor que diferencie entre as variáveis marcadas e as não marcadas, e obteremos assim uma entrada directa de computador. Naturalmente, a resposta correcta encontrar-se-á no programa de tal modo que este armazene a informação correcta. Este sistema, normalmente designado por *leitura de marca óptica*, encontra muitas aplicações em sistemas comerciais computarizados. Um exemplo será uma avaliação de «stocks» utilizando uma lista que é comparada a uma gama de níveis de stock, sendo o valor apropriado marcado depois de uma contagem directa dos stocks.

#### *Leitura de caracteres o.c.r.*

Os caracteres e as marcas ópticos são lidos ambos por varrimento por uma fonte de luz, associada a células foto-eléctricas. No caso dos caracteres, o reconhecimento é realizado em função da sua forma, e no das marcas em função da sua posição. A moderna leitura o.c.r. dará uma velocidade de cerca de 3 000 c.p.s.

#### *Terminais de computadores*

Estamos a pensar aqui em dispositivos que se encontram normalmente situados longe do computador, forne-

cendo a este uma ligação directa ou indirecta. Os termos que usamos para significar uma ligação directa ao computador será *on-line* (em linha, em inglês), ou seja, a informação é dirigida directamente para a memória do computador à medida que entra nos terminais, sem se recorrer a qualquer intervenção manual (figura 33).

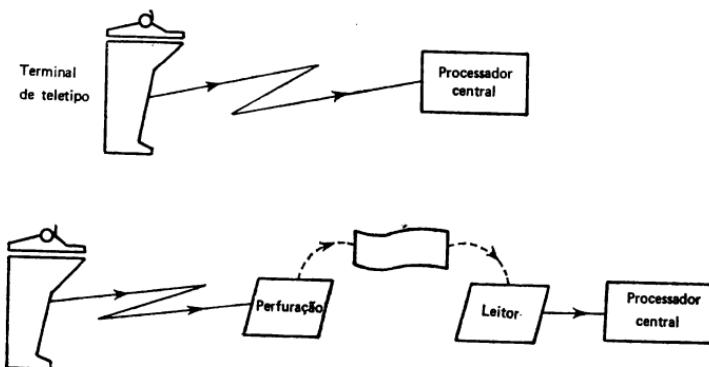


Figura 33 — Transmissão de dados «on-line» e «off-line».

Um terminal que seja usado para transmitir informação para um centro computador remoto, no qual esta informação seja reproduzida numa entrada de computador (por exemplo, sob a forma de fita perfurada) e em seguida lida para aproveitamento pelo computador numa operação separada, será designado por terminal *off-line*. Considerações de *hardware* do computador determinarão o modo de funcionamento do terminal, *on-line* ou *off-line*. Estas considerações serão discutidas num capítulo ulterior.

### *Transmissão de dados*

O processo de transporte de dados à distância, de um terminal para o computador correspondente, é conhecido

pelo nome de *transmissão de dados*. Num sentido estrito, o termo foi já associado a três métodos diferentes:

1. Quando os dados são fisicamente transferidos da fonte para a instalação do computador. Por exemplo, quando é preparada fita em papel perfurada num ponto remoto e enviada por correio ao centro computador.
2. Quando são usadas linhas de comunicação para transmitir dados para uma máquina off-line na instalação do computador.
3. Quando os dados são transmitidos por linhas de comunicação directamente para o computador.

No entanto, o termo transmissão de dados, muitas vezes designado por «transmissão automática de dados», é agora mais geralmente aceite como designação do terceiro caso, o da entrada directa ao processador central. Vimos anteriormente, neste mesmo capítulo, que quando se preparam dados sob forma de perfurações em qualquer meio, ao passar por um leitor os orifícios codificados, estes são convertidos numa série de impulsos eléctricos conhecidos pelo nome de bits ou dígitos binários que, por sua vez, são armazenados no computador. A transmissão automática de dados é basicamente uma extensão deste princípio, aplicada a maiores distâncias. A transmissão é realizada através de linhas de comunicação, e é fácil compreender que o tempo necessário para transmitir um dado volume de dados dependerá da eficácia de funcionamento do meio em que são transmitidos. A unidade usada para definir a velocidade de transmissão é conhecida por *baud*. Um baud representa a transferência de um bit por segundo.

A transmissão automática de dados utiliza normalmente as linhas de comunicação de telegrafo ou telefone, se bem que em alguns casos sejam usadas linhas directas privadas. As transmissões podem igualmente ser realiza-

das utilizando comunicações de rádio, inclusivamente por via satélite.

O uso de linhas telegráficas não é normalmente considerado óptimo para transmissão de dados de computadores. Como se poderia esperar, a velocidade de transmissão é pequena, cerca de 100 bauds, e em geral apresenta uma qualidade bastante deficiente. A rede telefónica constitui um meio mais rápido e de melhor qualidade,

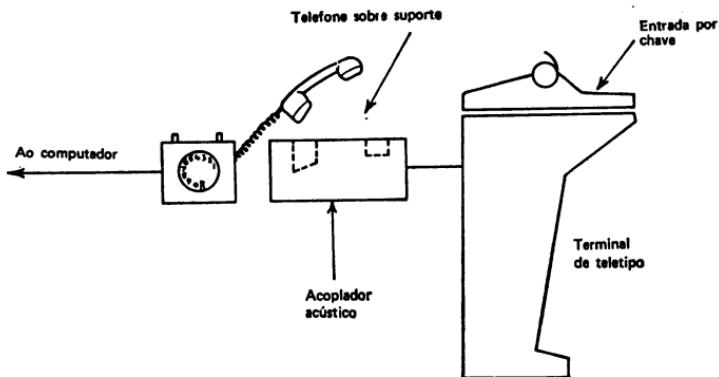


Fig. 34 — Ligação terminal — computador.

que pode ser usado a diferentes velocidades, 200, 300, 600 ou 2400 bauds.

Pode ser utilizado qualquer código de caracteres empregue pelos fabricantes nestes serviços telefónicos a alta velocidade, desde que se usem terminais de transmissão apropriados. Estes terminais encontram-se ligados à rede telefónica através de um dispositivo telefónico designado por *modem*, abreviatura de modulador-desmodulador. A modulação é o processo através do qual se convertem os dados digitais (ou seja, dados em forma binária) numa representação analógica para efeitos de transmissão, sendo a desmodulação a técnica de inversão deste processo no computador. É portanto necessário montar «modems» em ambas as extremidades da ligação (figura 34).

Existe uma segunda maneira de transmitir dados a partir dos terminais, se bem que esta funcione a uma pequena velocidade de entrada/saída. Consiste em utilizar um dispositivo designado por *acoplador acústico* (Fig. 35).

Neste caso, coloca-se o chamado «auscultador» de um telefone normal sobre um descanso concebido especial-

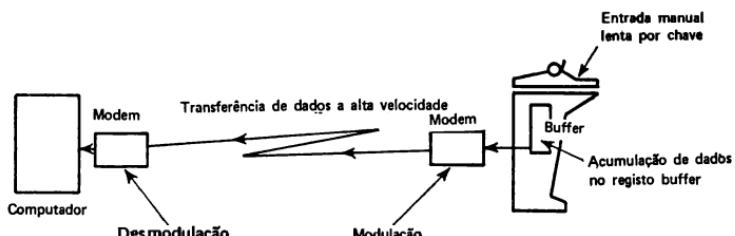


Fig. 35 — Utilização de um acoplador acústico ao terminal.

mente para este fim e que constitui parte integrante do terminal. Os sinais digitais do terminal são convertidos em sinais audio e transmitidos pela linha de uma maneira bastante semelhante à dos sinais das conversas telefónicas. Na extremidade onde se encontra o computador, são novamente convertidos em impulsos digitais prontos para entrarem no processador central.

A transmissão por acoplador acústico apresenta duas desvantagens principais: (a) baixa velocidade de transmissão — o máximo possível é 200 bauds; (b) existe o risco de os dados serem alterados devido a ruídos estranhos captados pelo sistema. No entanto, é também evidente uma importante vantagem no que se refere ao facto de o acoplador acústico poder ser usado com qualquer instalação telefónica normal, permitindo assim o uso de terminais portáteis.

### *Terminais de computador*

Essencialmente, a maior parte dos terminais de computadores tem uma função dupla, na medida em que

podem ser utilizados tanto para entrada como para saída. Se bem que exista uma extensa gama de terminais, estes podem ser grosseiramente classificados nas seguintes quatro categorias principais:

Terminais de teletipo

Terminais de processamento

Terminais de apresentação visual

Terminais especializados

Todos estes têm em comum a possibilidade de serem ligados on-line por linha de transmissão directa a um processador central.

#### *Terminal de teletipo*

De longe o tipo de terminal mais usado, consiste num teclado semelhante ao de uma máquina de escrever no qual se podem introduzir dados manualmente. Este teclado contém não só a gama habitual de caracteres alfanuméricos como ainda um certo número de teclas de «comando», através das quais se podem transmitir instruções ao computador (figura 36).

O terminal incorpora ainda um dispositivo de impressão, de tal modo que os dados introduzidos através do teclado possam ser reproduzidos numa cópia, e que serve ainda para registar a informação automaticamente transmitida do computador para o terminal.

Por definição, uma impressora de teletipo de caracteres em série constitui um dispositivo lento de entrada//saída, com uma velocidade máxima de cerca de 300 caracteres por minuto. Utilizando este tipo de dispositivo, a transmissão de dados à medida que vão sendo passados ao teclado conduz a uma utilização muito pouco económica do computador, dado que o processador central

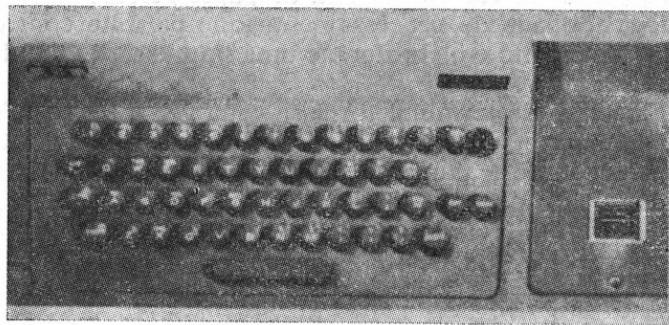
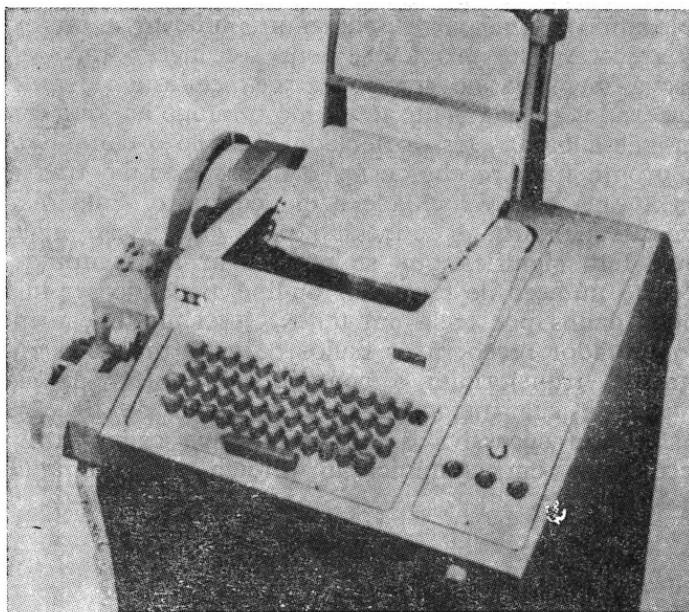


Figura 36 — Terminal de computador e respectivo teclado  
(Westrex Company, Ltd.).

se encontra parado durante extensos períodos de tempo, funcionando a uma velocidade muito inferior à possível. Para resolver este problema, pode ser incorporada uma pequena memória no terminal, que aceitará e manterá armazenados os dados durante todo o tempo em que estão a ser passados através do teclado. Quando o computador está pronto para receber estes dados, são então transmitidos a uma alta velocidade e em conjunto. Este dispositivo de memória é conhecido pela designação inglesa *buffer*. Isto significa que se pode ligar ao computador um certo número de terminais on-line, podendo ser introduzidos dados por tecla em todos eles simultaneamente. O computador receberá os dados de cada um deles rotativamente, transferindo o conteúdo do buffer de cada um para a sua memória. Este processo de ligação e transferência sucessiva é tão rápido que cada terminal parece estar em comunicação directa ao computador constantemente.

#### *Terminal de processamento*

Como o nome sugere, este é concebido para tratar dados. Na sua forma mais completa, consiste num leitor, normalmente de cartões ou fita de papel perfurados, uma pequena unidade de processamento que permite controlo e armazenamento em «buffer», e uma impressora de linhas relativamente rápida.

Este tipo de terminal torna-se cada vez mais popular, dado que torna disponível, em pontos bastante remotos, a capacidade de tratamento de dados de um grande computador.

#### *Terminal de apresentação visual*

Trata-se de um dispositivo em que a informação pode ser apresentada visualmente num visor constituído por um tubo de raios catódicos. Muitos destes terminais

incorporam um teclado para realização de entradas. A entrada de dados é automaticamente apresentada no visor, tal como os resultados do processamento realizado na unidade central.

As unidades de apresentação visual têm a vantagem de um curto tempo de resposta, isto é, a informação é apresentada quase imediatamente quando é pedida, em vez de se tornar necessário esperar que o registo seja impresso, se bem que, evidentemente, não se obtenha assim uma apresentação permanente.

### *Terminais especializados*

Esta categoria cobre uma vasta gama de dispositivos concebidos para aceitarem dados de uma ou outra forma e transmiti-los directamente para um computador. Exemplos:

1. Dispositivos de controlo usados em verificação.
2. Dispositivos de escrutínio magnético ou óptico para leitura de dados codificados relativos a produtos, para inventário imediato de stocks.
3. Terminais de actualização de documentos. Por exemplo, actualização de um livro de movimentação de contas bancárias (caderneta bancária).

### *Meios e dispositivos de saída*

Quando pensamos em termos de saídas de computadores, estamos a considerar o processo pelo qual a máquina comunica ou regista de qualquer maneira os resultados do seu processamento. Em princípio, podemos considerar esta saída como assumindo uma de duas formas: (a) saída intermédia, e (b), saída final.

No primeiro caso, a saída seria normalmente passada para algum tipo de memória onde seria mantida até ser necessária, num passo ulterior do processamento. Um exemplo disto é o caso da rotina de inventariação de stocks, em que são acumulados valores diários na memória do computador até, periodicamente, ser requerido o total.

No segundo caso, estamos a pensar no tipo de saída concebida para uma apresentação no exterior do computador, a fim de informar o utilizador ou fundamentar as atitudes a tomar. Por exemplo, uma lista impressa dos valores em stock, como se sugeriu anteriormente.

Os dispositivos de armazenamento onde podem ser guardadas as saídas do computador são discutidos no Capítulo 6, enquanto que em seguida se dão algumas indicações sobre os dispositivos usados para apresentar os resultados ao exterior, numa forma que possa ser compreendida.

#### *Saída impressa*

Como forma de saída final, esta é provavelmente a mais usada. A maior parte dos dispositivos de impressão final são de dois tipos, *de carácter único* ou *impressoras em série* e *impressoras de linha*.

*Impressoras de carácter único.* Estas máquinas já foram discutidas anteriormente neste mesmo capítulo, na secção dedicada aos terminais de computadores. São no entanto designadas muitas vezes por *tele-impressoras*, *máquinas de interrogação* ou *de consola*. É difícil indicar linhas absolutas de demarcação entre as áreas de trabalho de cada um destes subtipos. As máquinas interrogadoras são normalmente as impressoras em série, relacionadas com a comunicação on-line ao computador em ambos os sentidos. São por vezes designadas por *terminais inteligentes*. As máquinas de consola são utilizadas como um canal para comunicação computador/operator, enquanto

o termo tele-impressora se refere à transmissão de dados off-line, normalmente sob a forma de fita de papel perfurada.

*Impressoras de linha.* Como o nome sugere, a característica mais importante de uma impressora de linha consiste na sua capacidade para imprimir toda uma linha de tipos simultaneamente. Os dois tipos de impressora de linha mais usados para trabalho de computador on-line são designados por *impressoras de cadeia* e *impressoras de cilindro*.

O mecanismo de impressão de uma impressora de cadeia consiste numa cadeia metálica fechada, na qual se encontram montados um ou dois conjuntos completos de impressão. A cadeia roda continuamente, paralelamente à linha de impressão. Atrás do papel, encontra-se uma fila de martelos que são libertados individualmente quando o carácter pretendido atinge a posição de impressão desejada. A velocidade de batimento do martelo é suficiente para se obter uma impressão bem definida no papel, apesar de o conjunto impressor se movimentar a alta velocidade.

Um mecanismo de impressora de cilindro (figura 37) utiliza o mesmo princípio de movimentação contínua de faces de impressão, mas os caracteres são montados na superfície de um cilindro metálico e não numa cadeia. Todos os caracteres se encontram repetidos ao longo de todo o cilindro, uma vez para cada posição de impressão (ver a figura 37).

O cilindro roda a alta velocidade, de tal modo que cada carácter seja apresentado ciclicamente à linha de impressão. Quando, por exemplo, a linha dos A's está alinhada pela posição de impressão, são libertados simultaneamente os martelos em todas as posições de linha em que se pretende imprimir a letra A, comprimindo o papel contra a face do tipo. Em seguida, é colocada em posição a fila dos B's, etc., passando a vez a todos os caracteres. Isto significa que toda uma linha de impressão é obtida por cada revolução do cilindro.

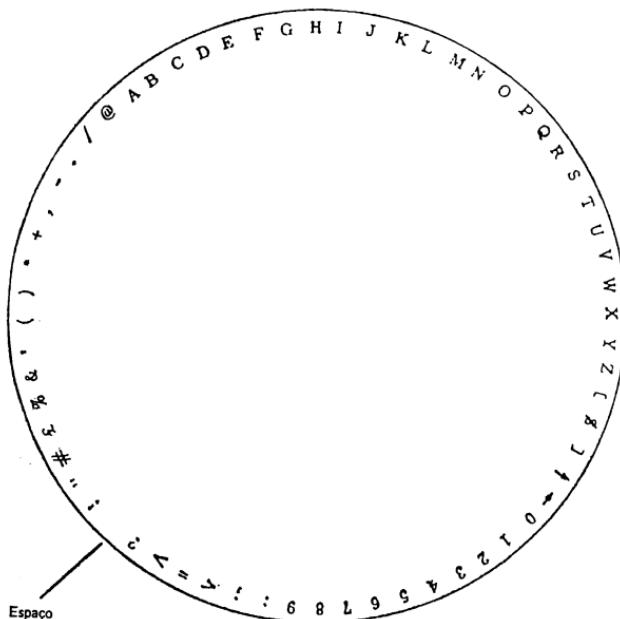
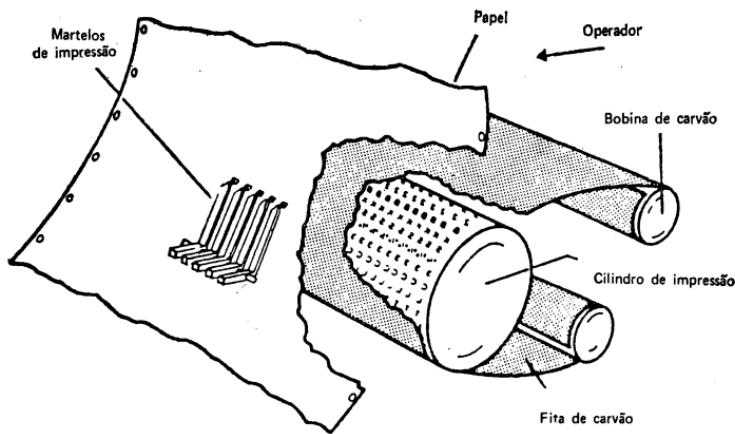


Fig. 37 — Impressora de cilindro — arranjo geral e disposição dos caracteres no cilindro de impressão.

As velocidades de impressão das impressoras de cadeia ou de cilindro, se bem que dependam dos caracteres existentes na linha e do número de caracteres usados, são bastante altas. Uma impressora em linha rápida pode produzir 1200 linhas por minuto, cada uma com 120 caracteres.

O conjunto de caracteres de uma impressora em linha tem normalmente um máximo de 64 caracteres diferentes: 26 alfabéticos, e 10 numéricos, sendo os restantes símbolos (ver a figura 37). A linha de impressão pode consistir num conjunto de caracteres num número máximo de 160, se bem que pareçam hoje ser mais populares as linhas de impressão de 120 caracteres.

Se bem que, como já vimos, não seja rara uma velocidade de impressão de 2400 c.p.s., esta velocidade não é ainda suficiente para acompanhar a velocidade de processamento de um computador. Isto significa que é perdida uma grande parte do tempo de processamento, enquanto o computador se ocupa da transferência para o exterior e da impressão dos dados de saída. No tempo necessário para imprimir uma linha — cerca de 50 milisegundos — o computador poderia facilmente realizar milhares de operações. Uma maneira muito utilizada de reduzir este tempo perdido consiste em dar à impressora a sua memória pequena, de dimensões reduzidas, a qual será designada por «buffer» e para a qual é possível transmitir toda uma linha de uma só vez. A impressora encarrega-se então do controlo da operação de impressão, extraindo os caracteres do buffer à medida que vão sendo necessários na posição de impressão. Isto deixa a área de saída da memória do processador livre para receber a linha seguinte de saída e permite igualmente que o processamento continue enquanto se realiza a impressão.

Se bem que o uso de um buffer permita utilizar o tempo de processamento de uma maneira mais eficiente, mantém-se ainda o facto de, mesmo nestas condições, os dados não poderem ser processados mais depressa que a velocidade a que podem ser fornecidos pela máquina. A saída pode no entanto ser temporariamente

inscrita numa fita magnética, à velocidade de 150 c.p.s. A fita magnética é mais tarde impressa durante um período de paragem do processamento.

#### *Cartões e fita de papel perfurados*

Ambos constituem formas de saída bastante lenta, consistindo em dados de saída perfurados numa forma codificada. Estes dois meios não se prestam a situações em que se tratem grandes volumes de dados, como por exemplo no caso de sistemas de processamento de dados comerciais, nos quais os volumes de saída e de entrada são grandes relativamente à quantidade de processamento a que os dados são submetidos. No entanto, a fita de papel é muitas vezes usada em aplicações científicas e matemáticas, nas quais se obtém uma saída relativamente pequena, em relação ao volume das operações de processamento envolvidas.

#### *Saída visual*

Este termo é geralmente usado para significar a apresentação visual de informações num tubo de raios catódicos. Uma unidade de apresentação visual (em inglês v.d.u.), já mencionada na secção sobre terminais de computadores, constitui um meio bastante conveniente para a obtenção temporária de dados. A informação é pedida através de uma entrada de teclado e em seguida apresentada, quase imediatamente, no visor da unidade.

Pode-se realizar a modificação dos dados assim apresentados, por entrada através do teclado. Isto permite actualizar o registo existente na memória e a alteração é reflectida no visor.

Do ponto de vista do utilizador, um v.d.u. tem a vantagem de possuir um rápido tempo de resposta, ou seja, a informação pode ser apresentada quase imediata-

mente após ser pedida e torna-se portanto ideal numa situação em que não é necessário possuir uma cópia dos dados de saída. No entanto, são necessários um processador relativamente potente e um «software» bastante complexo para suportarem o uso de v.d.u.'s em qualquer escala (figura 38).

Se bem que não entre verdadeiramente na categoria da apresentação visual, a saída em *microfilme* constitui



Fig. 38 — Unidade de apresentação visual (Data Dynamics).

uma maneira bastante prática de registar permanentemente os dados de saída, fornecendo o meio de obtenção de uma cópia permanente quando necessário e, simultaneamente, permitindo economia em espaço de armazenamento e facilidade de obtenção da saída. Nos primeiros tempos da utilização da saída em microfilme, realizava-se um processo em duas fases, primeiramente de apresentação visual ou em cópia permanente e em seguida de

fotografia ou fotocópia. No entanto, as técnicas foram desenvolvidas no sentido da transferência directa da fita magnética para microfilme, um método extremamente rápido de produzir uma cópia permanente.

### *Saída diagramática*

É muitas vezes necessário apresentar a saída sob a forma de um diagrama ou gráfico. As duas maneiras de o fazer consistem em utilizar um *traçador de incremento digital* que forma um registo permanente ou um v.d.u., o qual fornece um registo visual temporário. No último caso, o v.d.u. tem a vantagem de permitir a manipulação da imagem (aumentada ou diminuída, ou até alterada em perspectiva). O feixe electrónico que cria a apresentação pode ser redirigido, modificando automaticamente os dados digitais mantidos na memória e que definem o formato do diagrama.

Um traçador de incremento digital é um dispositivo usado para comunicar a saída do computador numa forma gráfica impressa como sejam diagramas, mapas ou desenhos a traço. A máquina consiste basicamente num tambor, que se movimenta para a frente ou para trás, no qual é colocado papel. Para obter um registo rigoroso, o papel é controlado por apertos. Acima do tambor, encontra-se suspensa uma ponta traçadora, capaz de se movimentar para a esquerda e a direita a toda a largura do papel. Obtém-se assim um movimento em quatro direcções básicas. Alterando as velocidades relativas do papel e da ponta, pode-se obter no papel uma linha com qualquer direcção e curvatura. A máquina é controlada por um programa, sendo as dimensões e a forma das linhas controladas por dados de saída. Se bem que a programação deste tipo de dispositivo seja bastante complexa e a produção da apresentação final lenta relativamente a um v.d.u., constitui mesmo assim um método rápido de produção de uma cópia permanente sob uma forma gráfica.

<i>Dispositivo</i>	<i>Possibilidades de utilização</i>	<i>Cópia física</i>	<i>Distribuição externa</i>	<i>Interrogação</i>	<i>Registo permanente</i>	<i>Saída intermediária</i>	<i>Nova entrada para subsequente utilização</i>	<i>Obtenção de informações</i>
Impressora de linha	Extensa	✓	✓		✓		✓	✓
Terminal (teletípico)	Relativamente extensa.	✓		✓	✓			✓
Unidade de apresentação visual	Menos extensa.			✓				
Tracedor de gráficos	Limitada	✓		Limitada	✓			
Meio magnético	Extensa					✓		
Meio perfurado	Limitada					✓	✓	✓
Microfilme	Em desenvolvimento		✓	(Duas fases)	✓			
Resposta audio	Em desenvolvimento (limitada)				✓			✓

Fig. 39 — Lista de dispositivos de saída.

### *Saída acústica*

Antes de terminar esta listagem dos dispositivos de saída, devemos mencionar um dos desenvolvimentos mais recentes, o da *saída acústica*. Se bem que tenham sido usados durante algum tempo *sintetizadores de voz* sob a forma de gravadores sofisticados de fita ou disco, e seja um processo relativamente simples reproduzir sons escolhidos sob controlo do computador, os desenvolvimentos mais recentes referem-se ao armazenamento de sons, palavras, etc. sob a forma digital, em circuitos do «estado sólido». Estes sinais digitais são lidos a partir da memória quando conveniente e alimentados através de um conversor digital para analógico, produzindo assim um sinal audio analógico que é passado por um amplificador e um alto-falante.

A figura 39 apresenta a lista de potencialidades e utilizações dos dispositivos de saída.

# 5

## O PROCESSADOR CENTRAL

Foram anteriormente mencionados dois termos relacionados com a instalação de computadores. Esses termos eram *hardware* e *software*. Software é o nome dado colectivamente ao sistema e aos programas, e hardware ao conjunto de máquinas que constituem o computador, incluindo o processador central, os dispositivos de entrada e de saída e as memórias.

O pormenor mais importante do hardware do computador é o processador central. Nos primeiros tempos, os computadores eram muitas vezes referidos pelo nome de «cérebros electrónicos». Se bem que esta expressão dê ao computador uma importância que não tem, dado que se trata apenas de uma máquina e, como tal, inclusivamente bastante limitada no que realiza relativamente às capacidades do nosso cérebro, o processador central é no entanto o local onde todo o trabalho é feito, onde se realiza a função de direcção e onde são tomadas decisões. Sob alguns pontos de vista, as suas funções são semelhantes a algumas que realizamos com o nosso cérebro. Estas podem ser sumarizadas do seguinte modo:

1. Uma capacidade de memória que armazenará e permitirá ir buscar, quando necessário, registos de informação.

2. A capacidade de «recordar» uma série de instruções.

3. A capacidade de realizar funções lógicas, incluindo a realização de operações aritméticas.

4. A capacidade de verificar e controlar as suas operações.

5. A capacidade de executar e «recordar» as instruções recebidas. De facto até mais que isto, dado que é capaz de decidir quais das instruções devem ser efectuadas e continuar este processo sem intervenção humana, até ser realizado o objecto das instruções.

6. A capacidade de, através do controlo das actividades dos dispositivos periféricos, transferir informações como e quando for necessário, manter estas informações na sua memória e finalmente comunicar os resultados do seu trabalho.

Talvez neste ponto, antes de considerar em detalhe estas funções do processador central, seja útil rever brevemente a relação entre o processador e o resto do computador. Já vimos anteriormente que, enquanto todo o trabalho de processamento é realizado no processador central, os dados sobre os quais este deve trabalhar e os programas que especificam o trabalho a realizar devem ser alimentados ao computador a partir de algum tipo de dispositivos de armazenamento ou entrada. Do mesmo modo, quando o trabalho está completado, os resultados devem ser retirados do processador para um dispositivo de armazenamento ou de saída. Isto significa que o processador deve possuir um certo número de «canais», através dos quais esta informação pode passar para e dos periféricos.

De facto, designamo-los por *canais de entrada e saída*. Um canal é pois uma via de entrada ou saída do processador central. Em termos simples, existe um cabo,

vindo do dispositivo periférico e com uma ficha na extremidade, que pode ser ligado a um destes canais de entrada/saída. O ponto de ligação ao processador é designado por *interface*. Um sistema de computador no qual qualquer tipo de periférico possa ser ligado como se pretenda terá aquilo que se poderá designar por uma *interface «standard»*.

Alguns processadores têm *canais lentos e rápidos*. Os canais lentos são usados para periféricos que transportam dados lentamente, como sejam os leitores de cartões, ou os leitores ou impressoras de fita de papel. Os canais rápidos são usados para periféricos rápidos, como sejam os de fita magnética e discos magnéticos.

Um computador é então normalmente uma «configuração» de máquinas, ligadas através de uma interface, sendo o processador central a parte central de controlo. O nome que damos a todo este conjunto de hardware é *configuração de computador*. A inter-relação dos periféricos com o processador central é ilustrada na figura 40.

Observemos agora as funções do próprio processador central e examinemos com algum detalhe as funções anteriormente mencionadas.

### *A memória*

Na capítulo 2, vimos que um computador trabalha com o mais simples dos sistemas de numeração — o sistema binário. Vimos igualmente que toda a informação existente no computador, quer tenha uma forma alfabetica ou numérica, é expressa digitalmente em termos de 0 ou 1. O segredo da memória de um processador deve portanto residir em algum tipo de dispositivo que possa, em qualquer momento concreto, encontrar-se num de dois estados, seja suficientemente estável para manter esse estado indefinidamente mas, simultaneamente, tenha a capacidade de alterar o seu estado de 0 para 1 ou de 1 para 0, conforme as necessidades. Outra condição ainda é que deve ser possível discriminar entre estes dois esta-

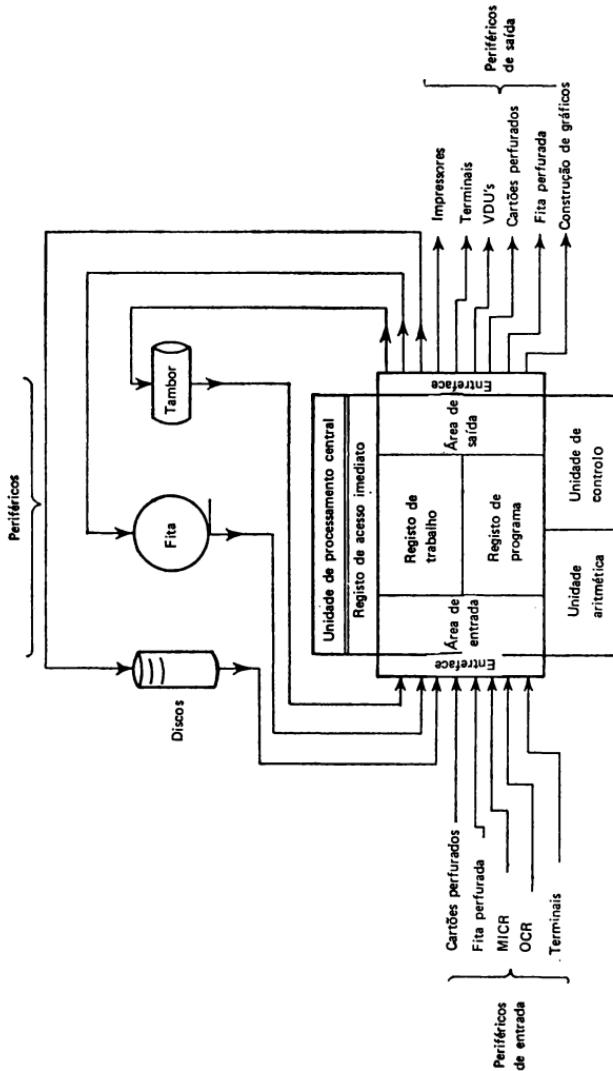


Fig. 40 — Periféricos e processador central.

dos, de tal modo que o estado binário representado possa ser «lido».

Talvez o exemplo mais simples seja neste caso o de uma lâmpada eléctrica que represente o 0 quando está desligada e 1 quando está ligada, e que, quer esteja ligada quer desligada, mantenha esse estado sem alteração. Como é evidente, pensar em usar circuitos deste tipo numa memória de computador é perfeitamente impossível.

Existem dois outros aspectos essenciais da memória de um computador. São eles o acesso e a velocidade. Há pouco interesse em desenvolver a mecânica de memórias de grandes dimensões, a não ser que nelas seja possível descobrir qualquer registo particular quando é necessário. Dado que a potência de um computador reside em grande parte na velocidade extremamente rápida a que é possível localizar qualquer informação, seja qual for o sistema de armazenamento usado na memória do processador, este deve prestar-se a uma busca muito rápida dos dados convenientes. Existem evidentemente muitas maneiras diferentes de armazenar informação. Um exemplo bem conhecido é o armazenamento de música, para o qual usamos discos ou fitas gravadas. Se pretendemos reproduzir uma dada frase da fita, é necessário bobiná-la num gravador até atingirmos a frase que nos interessa ouvir. No caso de um disco, se soubermos qual a faixa onde a frase em causa se encontra gravada, podemos colocar a agulha de reprodução no local certo. Podemos dar nomes a estes dois métodos, designando o primeiro por *acesso serial*, dado que é necessário passar a fita toda até ao ponto desejado, e o segundo por *acesso directo*, dado que podemos alcançar imediatamente o que pretendemos em qualquer dos lados do disco.

Como é evidente, o segundo método de localização é bastante mais rápido. São estas duas qualidades, a capacidade de se dirigir directamente à informação pretendida, e a capacidade de o realizar a grande velocidade,

que dão origem à designação da memória como o *registo de acesso imediato* do computador.

Em seguida, devemos considerar alguns dos dispositivos que podem ser usados para constituir a memória do processador, e isto de acordo com os critérios exigidos. Alguns merecer-nos-ão apenas uma curta menção, mas os mais utilizados serão observados em maior pormenor.

#### *Linha de atraso de mercúrio*

Este foi um dos primeiros dispositivos utilizados para armazenamento de informação em processadores. Sem sermos demasiado «técnicos», diremos que uma linha de atraso de mercúrio é um tubo de mercúrio com um cristal piezeléctrico em cada extremidade. Este tipo de cristal tem a propriedade de se expandir e contrair quando é passada uma corrente eléctrica variável através dele.

Reciprocamente, produzirá uma corrente pulsante quando é vibrado mecanicamente. Isto significa que uma série de impulsos aplicados numa das extremidades do cristal levam-no a vibrar, provocando uma série de ondas de pressão através do mercúrio contido no tubo. Estas ondas batem no cristal na outra extremidade e levam-no assim a vibrar, emitindo por sua vez uma série semelhante de impulsos, se bem que consideravelmente mais fracos que o original. Se bem que este processo atrase os impulsos de, no máximo, cerca de um milissegundo, se os sinais de saída forem amplificados e apresentados novamente à entrada do tubo, é obtida uma maneira de circular, ou armazenar, a informação indefinidamente. Quando se pretende usar a informação em causa, esta pode ser desviada para um canal de saída. Se se pretende armazenar uma nova informação no circuito regenerativo em causa, este é aberto, a fim de limpar o tubo de atraso de linha, e a nova informação alimentada através do amplificador (figura 41).

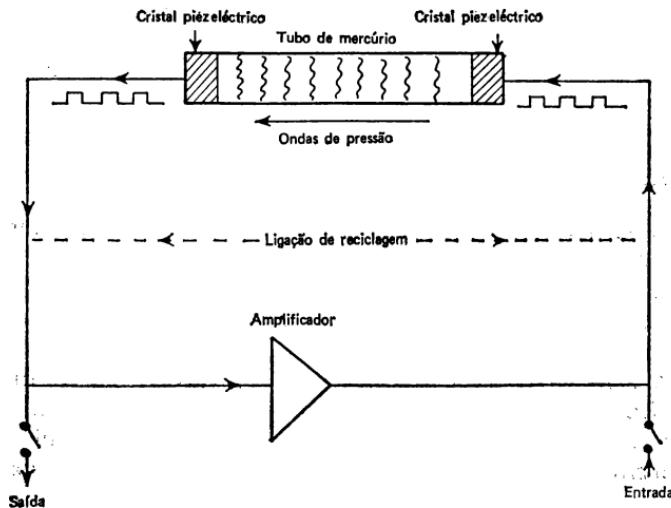


Fig. 41 — Linha de atraso de mercúrio, indicando-se a série de impulsos em circulação.

### *Tambores magnéticos*

Outro tipo de armazenamento utilizado nos primeiros processadores era o *tambor magnético*. Este não constitui verdadeiramente um registo de acesso imediato, dado que o movimento mecânico do tambor é necessário antes de poder ser localizada a informação pretendida. Isto ocupa um período apreciável de tempo — uma média de cerca de 10 milissegundos. O método constitui no entanto um grande passo em frente no sentido do aumento de volume de dados que podiam ser mantidos no processador. Se bem que os tambores tenham agora sido ultrapassados como registos dos processadores centrais pelas memórias electrónicas e magnéticas de «estado sólido» (uso de semicondutores), foram de facto reintroduzidos nos últimos anos como forma de armazenamento de apoio. Os tambores magnéticos são tratados com algum detalhe no Capítulo 6.

O maior desenvolvimento seguinte foi a introdução de armazenamento em anéis ferromagnéticos. É este que se usa como registo de acesso imediato na maior parte dos processadores centrais actuais.

#### *Registo de núcleo de ferrite*

Este registo utiliza o princípio eléctrico elementar que consiste no facto de, no caso de ser passada uma corrente eléctrica através de um fio, se formar um campo magnético na direcção dos ponteiros do relógio relativamente à direcção da corrente. Se se inverter a direcção da corrente, a direcção do campo magnético será invertida. Se passarmos o fio pelo centro de um pequeno anel capaz de ser magnetizado, como por exemplo um anel feito de ferrite, ao passar uma corrente através do fio, as extre-

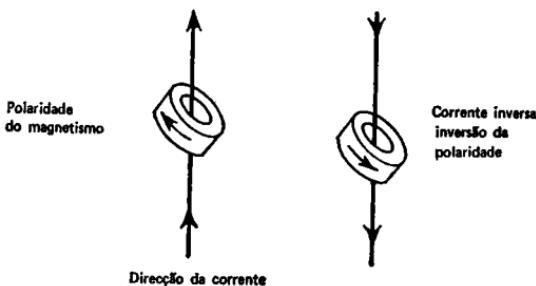


Fig. 42 — Princípio da memória em núcleo de ferrite.

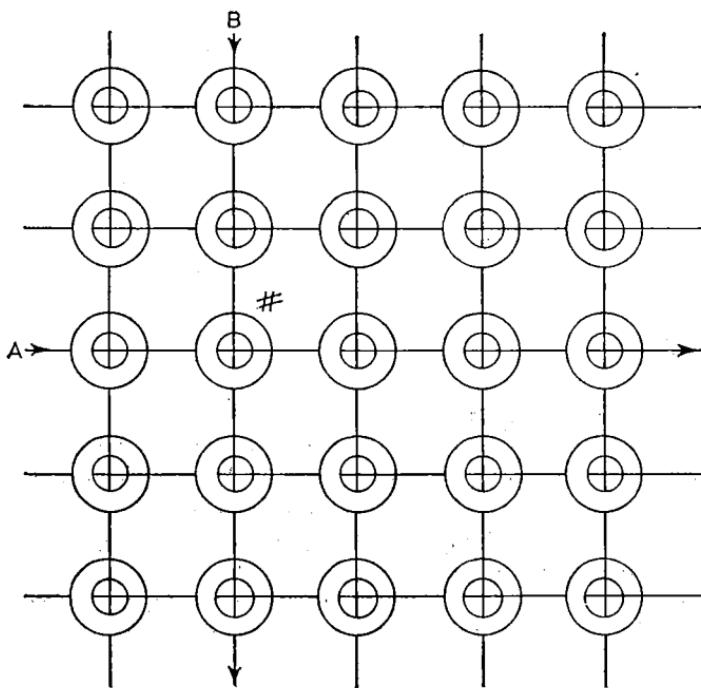
midades das partículas de ferrite que se orientam para o Norte do campo magnético apontarão na direcção dos ponteiros do relógio segundo as linhas de força do campo magnético e manter-se-ão nesse estado até ser introduzida uma força que o altere. Esta força poderia ser introduzida fazendo passar uma corrente de potencial suficiente na direcção oposta e ao longo do fio, invertendo-se então a polaridade do magnetismo. Isto dá-nos pois um dispositivo bastante prático e fácil de accionar, com apenas dois estados (figura 42).

Um núcleo de ferrite constitui então um pequeno anel de ferrite, com um diâmetro interior de 0,3 a 1,3 mm, que por virtude da direcção do seu magnetismo pode ser levado a representar um dígito binário 0 ou 1. No entanto, necessitamos de dezenas de milhares destes anéis para armazenar toda a informação necessária ao processador, e por sua vez estes anéis devem ser montados em grupos que representem palavras no computador. A capacidade de armazenamento do processador é definida pelo número de palavras que contém. Isto é expresso em termos de K, constante que representa 1024 palavras. As memórias são normalmente construídas em múltiplos de 4K, constituindo portanto memórias de 4 K, 8 K, 16 K, 32 K, etc. Isto significa, partindo de uma dimensão da palavra de 24 bits, que uma memória de 4 K seria constituída por, ignorando por agora quaisquer outros factores, cerca de 24 000 anéis de ferrite.

Uma complicação adicional é que cada anel individual deve ser acessível electricamente de tal modo que a direcção do seu campo magnético, e portanto também a sua representação binária, possam ser controladas.

Os núcleos de ferrite são montados numa matriz (figura 43), sendo cada linha de núcleos ligada por fios em duas direcções e perpendiculares entre si. Isto significa que existe apenas um núcleo na intersecção de quaisquer dois fios. Dado que a corrente necessária para alterar a possibilidade do magnetismo num núcleo deve ser de um valor mínimo necessário, se forem passadas correntes de metade deste valor através de dois fios perpendiculares entre si, o valor de corrente necessário só será obtido no núcleo, onde os dois fios se intersectarem. Ou seja, neste ponto a soma dos dois semi-impulsos é suficiente para alterar a condição deste núcleo. Isto significa que qualquer núcleo individual existente na matriz poderá ser escolhido de acordo com a sua referência única na grelha de fios.

Isto dá então acesso a qualquer núcleo individual numa matriz, se bem que se deve ter em conta que, nestas condições, só um núcleo pode ser atingido de cada vez.



O estado do núcleo, marcado #, pode ser alterado fazendo passar impulsos pelos fios A e B

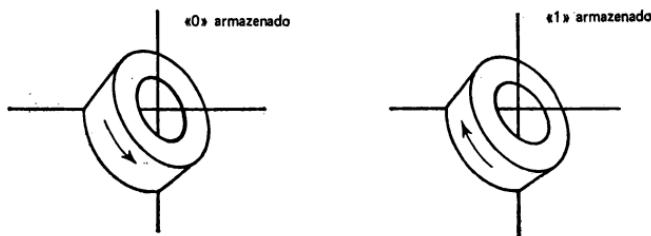


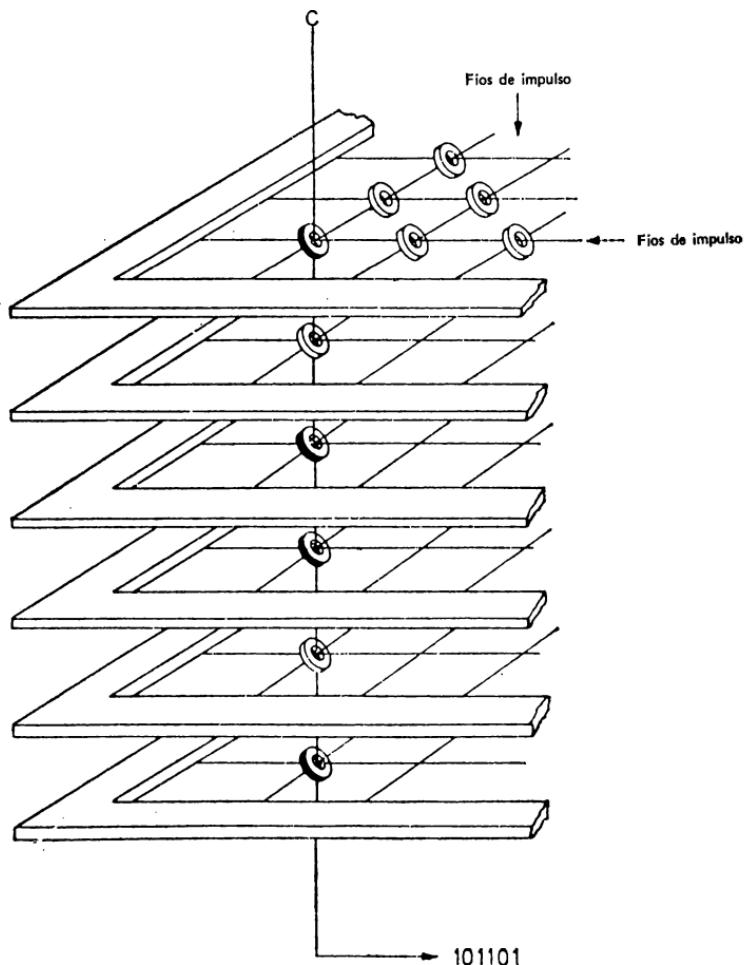
Fig. 43 — Secção de uma matriz de memória de núcleos magnéticos.

Qualquer tentativa para atingir dois núcleos simultaneamente, fazendo passar correntes através de dois conjuntos de fios perpendiculares entre si, encontraria quatro núcleos nos pontos de intersecção. Isto significa que, no caso de desejarmos ter acesso simultâneo a um grupo de núcleos representando uma palavra, isto não poderá ser feito no caso de todos eles se encontrarem na mesma matriz. Este problema é resolvido montando um certo número de matrizes verticalmente umas por cima das outras (figura 44), de tal modo que os núcleos aparecem agora em colunas verticais, e cada coluna pode ser considerada como uma palavra de computador. Dado que os núcleos existentes na coluna se encontram montados em grelhas de fios separadas, torna-se assim possível ter acesso a um certo número de núcleos simultaneamente, fazendo passar impulsos através do mesmo conjunto correspondente de fios em cada grelha.

Partindo do princípio de que começamos por um núcleo magnetizado numa direcção que represente zero, se fizermos passar semi-impulsos através dos fios A e B (figura 43), então a polaridade será invertida, dando origem a um estado 1. Como é evidente, é agora necessário fornecer a capacidade para interpretar ou «ler» o estado em cada núcleo. Esta função é novamente fundamentada num princípio eléctrico elementar. Se for movimentado um campo magnético relativamente a um fio, será produzida uma corrente induzida no fio.

Para captar esta corrente induzida, passamos um terceiro fio S (figura 45) por cada núcleo. Se agora forem passados simultaneamente semi-impulsos através de A'—A e B'—B, no núcleo (a) da figura 45 a direcção do magnetismo, no sentido dos ponteiros do relógio relativamente à direcção da corrente, manter-se-á inalterável e nenhum impulso será constituído em S, mas no núcleo (b) o efeito será o inverso da direcção do campo e esta inversão provocará um impulso induzido em S.

Temos agora no fio sensor S a representação do estado binário dos dois núcleos (a) e (b) — nenhum impulso = zero, impulso = um — mas deste modo destruí-



$\bullet = 1$   
 $\circ = 0$

Fig. 44 — Seção tridimensional de um registo de núcleos com seis planos. Cada coluna vertical representa uma palavra de seis bits que pode ser lida pelo fio sensível C.

mos a informação que se tinha originalmente, dado que ambos os núcleos têm agora um valor lógico zero. Torna-se agora necessário regenerar o estado original dos núcleos. Chama-se a isto o *ciclo de regeneração*. Para tal,

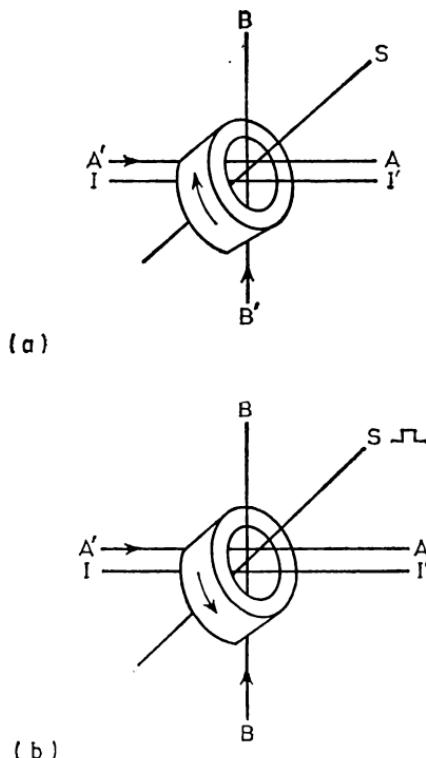


Fig. 45 — Leitura de um núcleo de ferrite.

são passados impulsos no sentido oposto ao longo dos fios  $A - A'$  e  $B - B'$  mas, como é evidente, isto fará passar ambos os núcleos (a) e (b) para o estado 1 quando, de facto, o núcleo (a) se encontrava originalmente no estado 0. É então necessária uma outra operação para

impedir um núcleo, originalmente no estado zero, de ser passado para 1 no ciclo de regeneração. Isto é feito passando um semi-impulso através do fio I'—I no sentido oposto a A—A', de tal modo que os dois impulsos dos dois fios coincidam apenas no núcleo que não se pretende alterar. O efeito disto consiste evidentemente em ter dois semi-impulsos deslocando-se em sentidos opostos, que se anularão mutuamente, e um terceiro impulso deslocando-se perpendicularmente, ao longo de B—B', que é insuficiente por si só para alterar o estado do núcleo.

Uma desvantagem importante das memórias de núcleos de ferrite é que o seu fabrico se torna caro. A sua construção emprega muita força de trabalho, devido à necessidade de montar manualmente os núcleos, passando por cada um deles os fios de leitura/escrita e sensor.

Os avanços técnicos realizados nos últimos anos nos circuitos integrados abriram caminho ao uso de semicondutores como elementos básicos de memória.

### *Os semicondutores*

Já discutimos anteriormente o uso de semicondutores bipolares na construção de dispositivos bi-estáveis — flip-flops — quando tratámos dos dispositivos de armazenamento temporário (capítulo 3).

Como um flip-flop constitui um dispositivo de armazenamento estável, é habitual usar nas memórias principais semicondutores metal-óxido. Este dispositivo consiste numa sanduíche de óxido metálico entre uma placa metálica e uma banda de silício. A resistência, e portanto também a condutividade, da pastilha de silício pode ser alterada aplicando uma carga à placa metálica. Temos então uma situação em que se poderia levar um baixo grau de condutividade a representar um binário 1 e uma alta condutividade um binário 0. Estando de facto fios ligados a ambas as extremidades de uma pastilha de silício — respectivamente um fio de fonte e um de *drain*, o fio de fonte pode ser verificado através de um amplifi-

cador sensor no que se refere a uma possível passagem de corrente. Um problema deste tipo de memória de semicondutores é que se torna bastante «volátil»; isto significa que, sendo aplicada a carga à placa metálica — a «porta» — a maior condutividade da pastilha de silício tem apenas uma curta duração, de cerca de 2 a 20 milisegundos. É portanto necessário reconstituir continuamente o estado dos semicondutores, reciclando a informação que representam em períodos de alguns milisegundos. Poderemos chamar a isto *ciclo de reanimação*.

As memórias semicondutoras deste tipo, contrariamente à memória de núcleos ferromagnéticos, dependem da corrente — a informação representada desaparece assim que a fonte de corrente é desligada. É argumento que, no computador moderno, isto não constitui uma grande desvantagem, dado que os periféricos muito rápidos evitam a necessidade de armazenar dados ou um programa na memória do processador por períodos significativos de tempo.

As memórias semicondutoras são montadas seguindo um princípio muito semelhante ao das memórias de núcleos: uma matriz de semicondutores num plano, com fios para aceder a cada «bit» individual no respectivo ponto de intersecção. São montadas algumas matrizes paralelas entre si, de modo que o conjunto de semicondutores que surgem nas mesmas posições relativas em todas as matrizes constituirão uma palavra de computador. Cada matriz (figura 46) contém uma série de amplificadores de leitura e amplificadores sensores, sendo o ciclo de reanimação realizado contínua e automaticamente através de circuitos que constituem parte integrante da matriz. Com os desenvolvimentos actuais dos microcircuitos integrados, é possível construir memórias semicondutoras muito pequenas mas bastante rápidas e de funcionamento bastante seguro, por exemplo, memórias de 4K numa pastilha de  $50 \times 75$  mm.

Todos os tipos de dispositivos de memória discutidos até agora têm uma característica em comum: o seu estado pode ser alterado pela entrada de um impulso eléctrico.

Esta é evidentemente uma característica essencial, dado que se pretende que as memórias armazenem temporariamente informação variável.

#### *Memórias de leitura*

Parte da informação mantida no processador central, no entanto, não varia e é permanente. Referimo-nos por

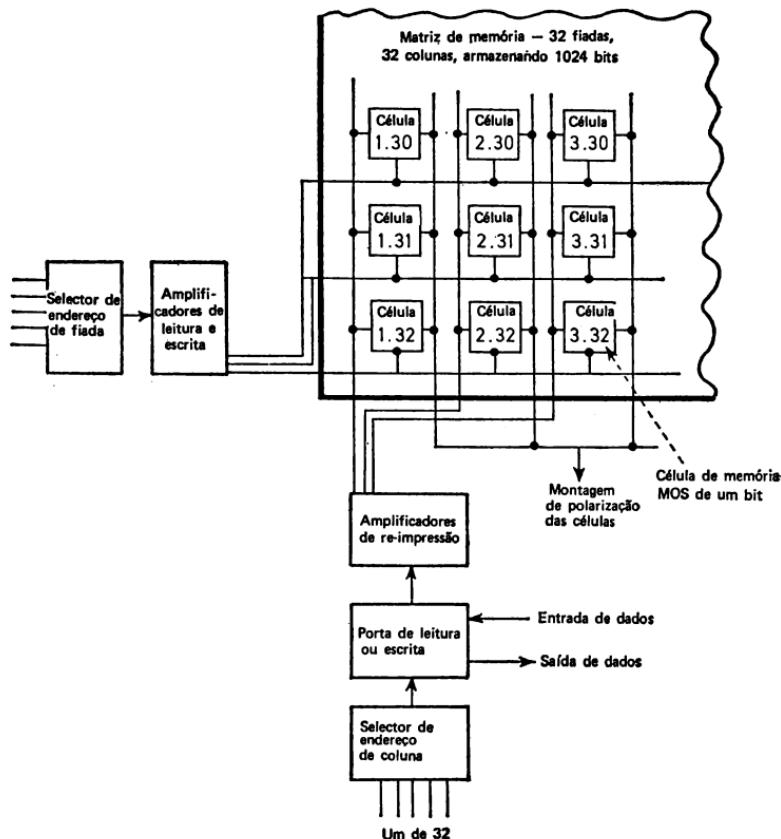


Fig. 46 — Exemplo de memória semicondutora de acesso cíclico.

exemplo à informação de controlo, como seja o código de funções da máquina ou o seu sistema de funcionamento, ou ainda, por outro lado, os dados necessários em funções de processamento, como por exemplo tabelas de valores das constantes. Um exemplo deste último tipo é uma tabela aritmética consistindo num certo número de posições reservadas na memória, nas quais se mantêm permanentemente a soma e a diferença entre qualquer par possível de dígitos binários. Isto fornece aliás uma maneira alternativa de realizar operações aritméticas.

Para manter estes dados estáticos, são necessários dispositivos de memória cuja representação binária não possa ser alterada depois de os registos terem sido construídos. Chama-se-lhes *memórias de leitura apenas*. O estado destes registos, em termos binários, é constante e a sua saída será sempre a mesma, quando lhes forem submetidos impulsos de entrada.

### *Organização da memória*

Até agora, discutimos neste capítulo diversos dispositivos capazes de armazenarem bits individuais e vimos a maneira como é possível alterar o estado destes dispositivos, de modo a que possam representar 0 ou 1 conforme as necessidades, e igualmente que deve existir um método de discriminar entre ambos — um processo de leitura. Além disso, no Capítulo 2, foi mencionado que uma memória que continha apenas uma amálgama destes bits não teria qualquer significado, a menos que estes se encontrassem organizados em grupos definidos, permitindo não só a diferenciação de partes desses dados entre si como ainda fornecer um meio de localizar qualquer dado particular em termos de um endereço único. Neste ponto, talvez devêssemos olhar um pouco mais de perto estes grupos, conhecidos pela designação de *palavras de computador*.

Tendo dito que uma palavra contém um número standard de bits, é necessário reconhecer que a dimensão

de uma palavra varia de uma máquina para outra. Em algumas máquinas, poderá conter o número suficiente de bits para registar um carácter, normalmente seis bits, o que nos dá uma situação em que cada carácter tem um endereço único. A maior parte das máquinas, no entanto, utilizam uma palavra de maior dimensão que acomodará um certo número de caracteres ou uma expressão binária relativamente extensa. Trata-se das palavras de 16, 24, 36, 48, etc., bits. Para efeitos das explicações seguintes será usada uma palavra de 24 bits.

### *A palavra de computador*

Aquilo que é conhecido por palavra de 24 bits acomodará, de facto, 25 bits. O vigésimo quinto bit é usado como verificação de paridade a fim de salvaguardar o rigor

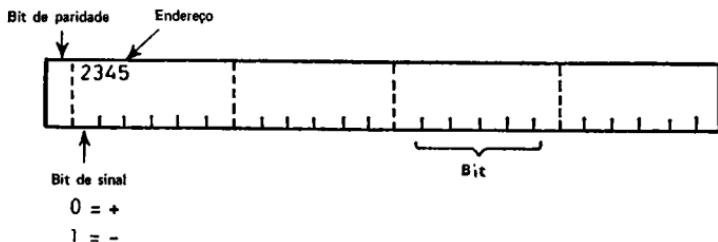


Fig. 47 — Palavra de vinte e quatro bits.

dos dados quando estes são deslocados dentro do computador (figura 47).

Cada palavra, excepto as reservadas para objectivos especiais, como sejam registadores ou acumuladores, é capaz de guardar uma instrução de programa ou um dado. Cada palavra pode ser dividida num certo número de subsecções designadas por *bytes*. No caso de uma palavra de 24 bits, cada byte conterá seis bits, o que é suficiente para um carácter — número, letra ou símbolo. Se bem

que a palavra propriamente dita tenha um endereço, os bytes individuais não o têm.

O primeiro dos 24 bits de dados (figura 47) é utilizado como bit de «sinal». Um bit 0 nesta posição indica um número positivo e um bit 1 indica um número negativo. O número positivo maior que pode ser guardado numa palavra de 24 bits é o representado por 23 bits «1»:

011111111111111111111111

Isto equivalerá a  $2^{23} - 1$ , o que é igual a 8 388 607. O número mais negativo que pode ser guardado é representado por um «1» seguido de 23 zeros:

100000000000000000000000,

ou seja  $-2^{23}$ , ou ainda,  $-8\,388\,608$ .

Recordemos que os números negativos na forma completa são tais que 24 bits «1»:

111111111111111111111111

será o complemento de

000000000000000000000001,

ou seja, — 1. Uma palavra de 24 bits armazenará portanto qualquer número sob a forma de uma expressão binária pura ou serial, na gama entre  $-8\,388\,608$  e  $+8\,388\,607$ . Se é necessário armazenar um número que ultrapasse esta gama, pode-se fazê-lo utilizando duas palavras adjacentes. Neste caso, o bit de sinal da palavra menos significativa não é usado como parte da expressão, sendo colocado no valor zero. O sinal é indicado pelo bit de sinal da palavra mais significativa. Esta prática de junção a uma segunda palavra é conhecida por *aritmética de comprimento duplo*.

Os dados mantidos na memória assumirão uma de duas formas — alfabética ou numérica. Como já vimos

anteriormente, a codificação alfabética é realizada em grupos de seis bits e será portanto armazenada sob a forma de quatro caracteres por palavra. Os números podem ser alternativamente descritivos ou quantitativos, como por exemplo na afirmação «54 peças em stock de referência BG167C». Os algarismos no número de referência não se encontram sujeitos a qualquer processo automático, enquanto que a quantidade «54» o pode perfeitamente estar. Enfrentamos então a escolha entre armazenar afirmações numéricas quantitativas em forma binária pura ou em codificação binária.

Em muitos computadores, ambas as formas são utilizadas, mas em fases diferentes. Quando se lê primeiramente os dados para armazenamento na memória, estes encontram-se normalmente em codificação binária. Qualquer parte que deva passar por uma operação aritmética é então convertida numa afirmação binária pura para este fim e, quando os cálculos estão terminados, a resposta é novamente reconvertida em pura codificação binária, para execução da saída.

### *Realização das operações aritméticas*

Já vimos no capítulo 2 a maneira como as operações aritméticas podem, em aritmética binária, ser reduzidas a simples adições, e no capítulo 3 a maneira como um somador pode ser construído utilizando uma combinação de portas lógicas. A próxima questão a considerar, em princípio, é a forma como o processador realiza as operações aritméticas. Dado que dois números, ou operandos, deverão estar presentes, são necessários dois registos especiais na unidade aritmética para os guardar. Podemos chamar-lhes registos A e B (figura 48), se bem que na maior parte dos sistemas o registo B seja igualmente um acumulador, no qual podem ser mantidos os resultados intermédios ou finais do processamento.

As operações são controladas através de um registo de controlo aritmético que governará a série de fases necessárias para realizar os cálculos, a primeira das quais será uma instrução para «limpar» (em inglês «clear») a fim de «esvaziar» todos os registo.

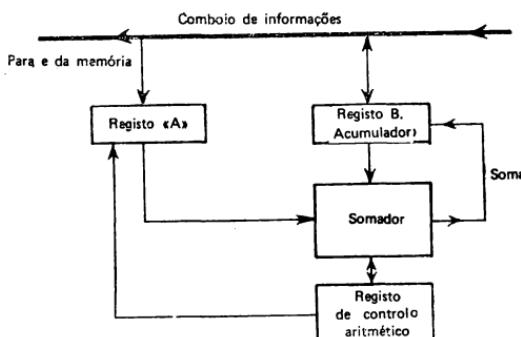


Fig. 48 — Função «adição».

### *Adição*

Sob uma instrução do programa, é dado acesso a um «acumulador de carga» ao endereço do operando relevante na memória e a informação por este contida é circulada, sendo abertas as portas necessárias para que esta possa entrar no acumulador.

Uma ordem «somar ao acumulador» dará acesso ao endereço do segundo operando e o seu conteúdo será movimentado do mesmo modo para o registo A. O controlo aritmético leva agora os dois operandos a circularem através do somador e o resultado é então movimentado de novo para o acumulador, sendo sobreposto ao seu conteúdo anterior. Finalmente, será executada uma instrução para movimentar o resultado para o endereço de destino na memória.

## Subtracção

Já vimos anteriormente que a máquina não subtrai realmente, antes realizando esta operação por adição complementar. Isto significa que um outro dispositivo poderá ser necessário na unidade aritmética — um complementador. Este é colocado em circuito pelo controlo, quando se pretende executar uma ordem de subtracção, convertendo o subtraendo mantido no registo A, antes da sua circulação através do somador (figura 49). A partir

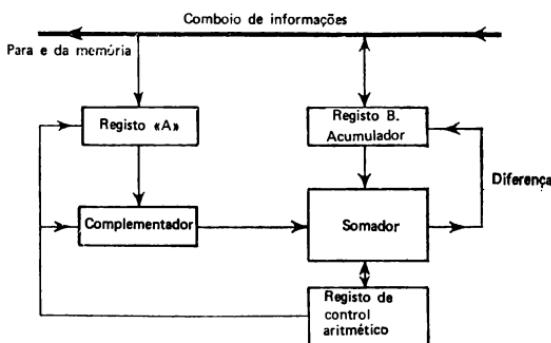


Fig. 49 — Função «subtracção».

deste ponto, realiza-se uma adição normal, sendo a «diferença» guardada novamente no acumulador e lançada em seguida no respectivo endereço de memória.

Surge uma nova complicação no caso de a máquina dever realizar uma adição ou subtração algébrica, dado que ambos os números podem ser negativos. No caso da adição, a regra consiste em somar sinais iguais e subtrair sinais diferentes (complementação e soma). Por exemplo:

Soma de — a e — b ..... soma

Soma de — a e + b ..... complementação e soma

Isto significa que se torna necessário um complementador adicional, trabalhando juntamente com o acumulador, e um comparador de sinal, trabalhando em conjunto com os complementadores, a fim de decidir se é ou não necessária a complementação.

### *Multiplicação*

Como já vimos, a multiplicação pode ser efectuada por adição repetida. Este processo é, no entanto, relativamente lento. Por exemplo, o cálculo  $111011\ (59) \times 101011\ (43)$  obrigaria a somar 111011 a si próprio 43 vezes. A multiplicação pode ser consideravelmente encurtada, utilizando um princípio de deslocamento, idealmente apropriado à aritmética binária. Este deslocamento seguirá as seguintes regras:

1. Se o primeiro dígito do multiplicador for 1, o multiplicando é copiado, movido para o acumulador e deslocado de um espaço para a direita. Se o primeiro dígito do multiplicador é 0, então o multiplicando não é copiado, mas as posições para o produto no acumulador são deslocadas um espaço para a direita.
2. Se o segundo dígito do multiplicador é 1, então o multiplicando é circulado através do somador juntamente com o conteúdo do acumulador, sendo o produto parcial resultante movido de novo para o acumulador e deslocado mais um espaço para a direita. Se for zero, o conteúdo do acumulador é apenas deslocado um espaço para a direita.
3. Este processo é repetido até que o dígito final do multiplicador seja tratado, e nesta fase será guardado no acumulador o produto final.

Multiplicando	1 1 1 0 1 1	
Multiplicador	1 0 1 0 1 1	
1.º dígito	1 1 1 0 1 1	— se 1, copia ao acumulador.
	1 1 1 0 1 1	— deslocamento de um espaço
2.º dígito	<u>1 1 1 0 1 1</u>	— soma ao acumulador
	1 0 1 1 0 0 0 1	— produto parcial ao acumulador
	1 0 1 1 0 0 0 1	— deslocamento de um espaço
3.º dígito	1 0 1 1 0 0 0 1	— não soma
4.º dígito	<u>1 1 1 0 1 1</u>	— deslocamento de um espaço
	1 0 1 0 0 0 1 0 0 1	— soma ao acumulador
	1 0 1 0 0 0 1 0 0 1	— produto parcial ao acumulador
5.º dígito	1 0 1 0 0 0 1 0 0 1	— deslocamento de um espaço
6.º dígito	<u>1 1 1 0 1 1</u>	— não soma
	1 0 0 1 1 1 1 0 1 0 0 1	— deslocamento de um espaço
	1 0 0 1 1 1 1 0 1 0 0 1	— soma ao acumulador
	1 0 0 1 1 1 1 0 1 0 0 1	— produto final ao acumulador

= 2537.

Em termos do «hardware» da unidade aritmética, isto significará que se tornam necessários dois elementos adicionais. Um registo adicional para manter o multiplicador (o multiplicando ficará no registo A) e uma porta para testar cada dígito do multiplicador, verificando se o algarismo em causa é, em cada caso, um «zero» ou um «um», e para ordenar uma operação de soma no segundo caso ou uma operação de deslocamento no primeiro

(figura 50). Depois de armazenar o multiplicando e o multiplicador nos registo A e Q respectivamente, a ope-

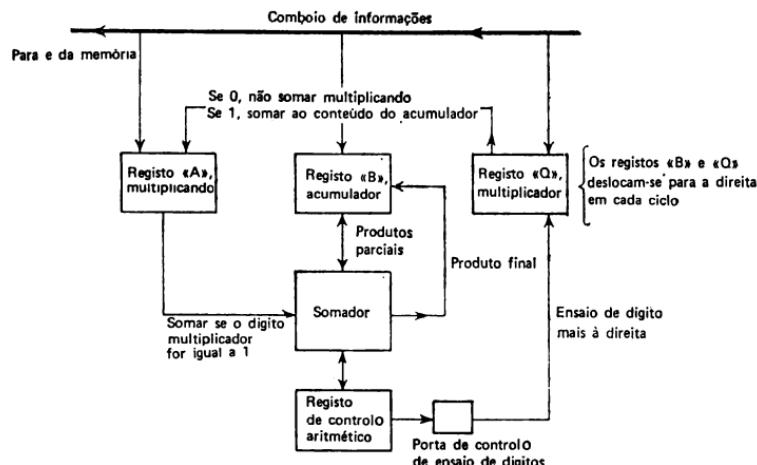


Fig. 50 — Função «multiplicação».

ração de multiplicação continuará, de acordo com o seguinte:

1. Verificação do dígito mais à direita do multiplicador, a fim de saber se é 1 ou 0. Se for 1, movimentar o multiplicando para o acumulador. O acumulador contém agora um produto parcial, ou seja, o multiplicando vezes o dígito menos significativo do multiplicador.

2. Segue-se uma operação de deslocamento. Existem diversas maneiras de a realizar, mas consideremos neste caso que a operação em causa movimenta o conteúdo do acumulador um espaço para a direita. Isto é equivalente a movimentar o produto do dígito seguinte do multiplicador para a esquerda.

3. O dígito seguinte do multiplicador, para a esquerda, é agora verificado. Se for 1, o multiplicando é movimentado para o somador, somado ao conteúdo do acumulador e o resultado armazenado como um novo produto parcial

no acumulador. Se for 0, o multiplicando é ignorado e não se realiza qualquer adição.

4. Realiza-se um novo deslocamento para a direita como em (2).

5. É verificado o dígito seguinte do multiplicador como anteriormente, sendo as operações repetidas até o cálculo estar completo, momento em que o produto final é mantido no acumulador, pronto a ser movimentado para um endereço de memória.

### *Divisão*

Tal como no caso da multiplicação, a divisão pode ser realizada por subtração repetitiva, o que com efeito significa a realização de uma adição complementar repetitiva. O uso de técnicas de deslocamento encurtará no entanto novamente este procedimento. O dividendo é mantido num acumulador e o divisor no registo B. Será necessário um complementador para encontrar o complemento do divisor, a fim de somar através de um somador paralelo. O dividendo e o complemento do divisor são circulados através do somador, alinhando pela extremidade esquerda das expressões. Se, como resultado da adição, resultar um resto negativo, o dividendo é deslocado um espaço para a esquerda, e o divisor é «subtraído» nesta posição. Um resto positivo provocará o armazenamento de um 1 no lado direito do registo de quociente (figura 51).

Tanto o dividendo como o quociente são agora deslocados um espaço para a esquerda e realiza-se a «subtração» do divisor. Um resto negativo dará um 0 na posição mais à direita do registo do quociente, e um resto positivo um 1. Este processo de deslocamento para a esquerda e «subtração» é repetido até o cálculo ser completado. O registo do quociente contém agora o resultado, pronto a ser novamente movimentado para a memória.

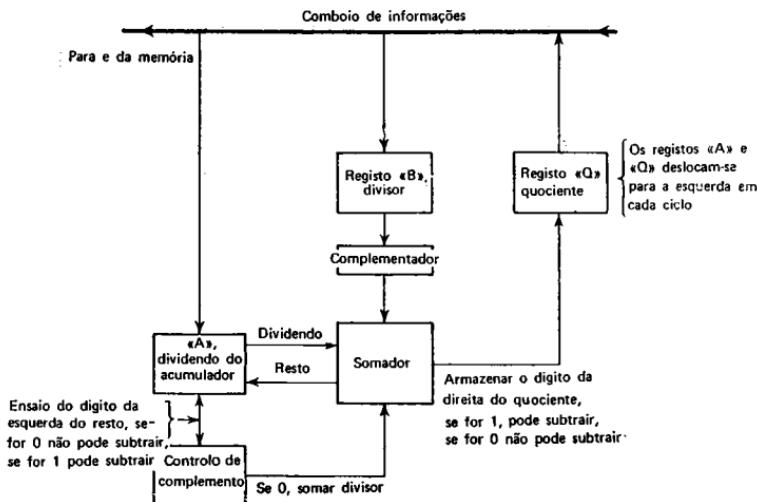


Fig. 51 — Função «divisão».

### *Sumário de funções do processador central*

A figura 52 mostra, numa forma muito simplificada, a relação entre as partes funcionais de um processador central. Nos termos mais simples possíveis, a rotina no interior do processador central pode ser definida nas fases seguintes:

1. O endereço da primeira instrução de programa é enviada para o contador do registo de programa.
2. O endereço da instrução é movido para um registo de endereço.
3. É activado um selector de endereços, de modo a dar acesso ao endereço de memória referido.
4. O conteúdo da palavra existente no endereço é movimentado para o registo buffer da memória.

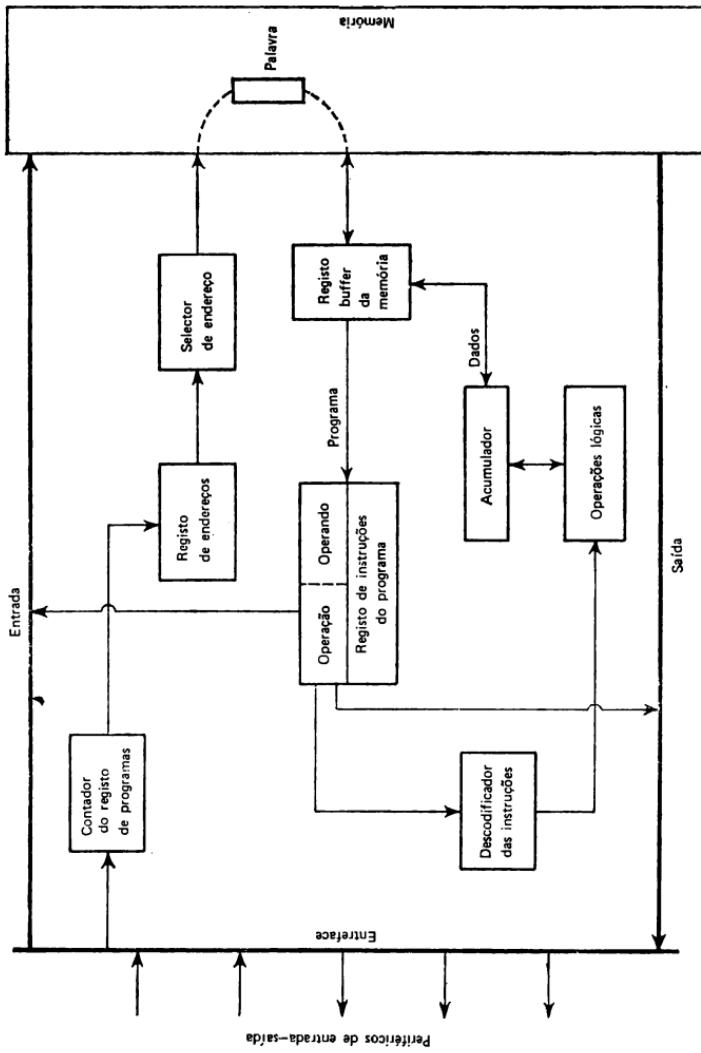


Fig. 52 — Vista simplificada do processador central.

5. Se a palavra contém uma instrução de programa, é copiada e armazenada no registo de instruções de programa.

6. A parte de codificação da operação, existente na instrução, é circulada através de um descodificador da instrução, o qual por sua vez liga os circuitos lógicos que irão realizar a operação em causa.

7. A parte de endereço do operando, existente na instrução, é movimentada para o registo de endereços e em seguida, através do procedimento de selecção, é obtida a palavra em causa no endereço apropriado da memória.

8. Esta palavra, contendo os dados que devem ser utilizados nos cálculos, é agora movimentada através de um buffer de memória para um acumulador.

9. São realizadas as operações lógicas requeridas sobre os dados.

10. Os dados resultantes das operações lógicas são agora conduzidos novamente para a posição de memória onde devem ficar, através do procedimento de selecção de endereços.

11. Ao completar cada instrução, o contador do registo do programa é actualizado, de modo a fornecer o endereço da memória onde se encontra a instrução seguinte.

12. As operações de entrada e saída, para e dos periféricos, são comandadas pelo programa da maneira especificada pelo código de operações existente na instrução. Isto obrigará provavelmente a uma codificação ou descodificação decimal/binária, obrigando ainda ao uso dos processos de escolha de endereços, para colocar ou retirar da memória as palavras necessárias.



## 6

**ARMAZENAMENTO E OBTENÇÃO DE INFORMAÇÃO**

Na nossa vida de todos os dias, verificamos que existe um grande número de factos e valores que somos obrigados a consultar. Não seria realista sugerir que toda esta informação pode ser armazenada em nós próprios, ou seja, na nossa própria memória, e portanto estamos habituados a memorizar os dados que se tornam mais importantes para nós e confiamos na possibilidade de descobrir o resto que nos interessa em notas, livros, etc. Desde que saibamos onde procurar, somos normalmente capazes de encontrar qualquer informação pretendida. Este processo pode ser designado por *obtenção de informação*.

Estamos portanto a usar dois métodos básicos de armazenamento de informação, um numa forma interna — a nossa própria memória — e outro numa forma externa a nós próprios. Existe evidentemente uma grande diferença de tempo na recolha de informação a partir destas duas fontes. Se lhe perguntarem o seu número de telefone, poderá provavelmente indicá-lo imediatamente, dado que o tem na memória. Devido a esta facilidade, a sua memória poderia ser designada como um *registo de acesso imediato*. É este com efeito o nome dado ao registo existente no processador central de um computador, posto que contém informação passível de uso imediato. No entanto, tal como na nossa memória, existe uma severa limitação ao volume de dados que a memória do pro-

cessador poderá guardar. Se você necessitar do número de telefone de uma pessoa de quem não tenha ouvido falar anteriormente, é obrigado a recorrer a uma lista telefónica, dado que a memorização de todos os números nela existentes estaria, em princípio, para além das suas capacidades. Em termos de computadores, o nome dado ao meio em que esta massa de informação é mantida e existe para referência, quando e como for necessária, é o de *memórias auxiliares* (memórias de apoio).

Como é evidente, a procura de um número numa lista telefónica levará um tempo considerável, bastante mais longo que o necessário para recordar um número existente na nossa memória. Do mesmo modo, se bem que as unidades de tempo envolvidas sejam muito menores, medidas neste caso em milissegundos, será necessário um período significativo de tempo para localizar e obter um dado mantido nas memórias auxiliares.

No entanto, ao estabelecer esta analogia entre os registos de um computador e a nossa memória, devemos notar uma diferença bastante importante. A informação é mantida na nossa memória durante um longo período de tempo, aliás permanentemente em certos casos. A memória do processador central mantém a sua informação durante talvez apenas uma fracção de segundo. Toda a informação é mantida em memórias auxiliares e o processador limita-se a escolher aquilo que pretende utilizar em qualquer momento, transfere-o para a sua própria memória, trabalha com os dados em causa, e em seguida transfere novamente os resultados para as mesmas memórias auxiliares, repetindo este procedimento quantas vezes for necessário até terminar o processamento.

Em conclusão, são usados dois tipos básicos de registos num computador:

1. O *registo interno* que é a memória do processador, na qual temos um acesso imediato a qualquer dado.

2. O *registro externo*, normalmente designado pelo nome de memórias auxiliares; se bem que não constitua uma parte integrante do processador, está-lhe directamente ligado, de tal modo que a informação nele contida pode ser consultada pelo processador em caso de necessidade.

### *Registro interno (memória principal)*

Já considerámos, nos capítulos 3 e 5, os tipos de meios de registo mais geralmente usados em processadores centrais. Como veremos, a característica mais significativa que distingue este tipo de armazenamento das memórias auxiliares é a de ser puramente electrónica, não tendo qualquer parte móvel. É isto que dá ao processador a possibilidade de um acesso imediato a qualquer parte dos registos, independentemente do ponto onde a informação se encontra.

### *Memórias auxiliares*

A diferença essencial entre estas e a memória principal é a da velocidade a que se recolhe a informação e a facilidade em o fazer. Virtualmente, todos os tipos de memórias auxiliares têm como base alguma forma de meio de registo magnético. Isto presta-se de uma maneira bastante prática ao armazenamento em forma binária, através da simples presença ou ausência de áreas magnetizadas. As formas principais destes meios de armazenamento magnético, geralmente usadas, são:

- Fitas magnéticas
- Tambores magnéticos
- Discos magnéticos
- Cartões magnéticos
- Faixas magnéticas
- Barras magnéticas

A natureza da construção física destes tipos de registos determinará quantos dados individuais podem ser encontrados em cada um deles. Existem basicamente duas maneiras de localizar um determinado dado, que talvez possam ser melhor ilustradas pelo exemplo seguinte.

Se pretendemos encontrar uma palavra específica, existente nesta página impressa, e se sabemos qual a linha em que se encontra e a sua posição nessa linha, pode-se ir directamente à sua posição sem referência a qualquer outra parte do texto existente na página. Incidentalmente, o número da linha e a sua posição na linha poderiam ser designados por *endereço* da palavra em causa. Este tipo de localização é designado por *acesso directo*. Se, por outro lado, a informação contida nesta página estivesse gravada numa fita magnética, e não em forma impressa, seria necessário, em alguma fase, reproduzir a fita e ouvi-la até reconhecer a palavra particular procurada.

Dos tipos de meios de armazenamento magnético referidos na lista anterior, as fitas magnéticas entram nesta última categoria, a dos registo de acesso serial, enquanto os restantes, num grau maior ou menor, constituem dispositivos de acesso directo. Talvez aqui se deva mencionar, no interesse da terminologia de computadores, que estes dispositivos de acesso directo são muitas vezes designados por *registos de acesso livre*. E isto por duas razões. Primeiro porque qualquer dos dados pode ser consultado independentemente daquilo que está registado antes ou depois dele, e em segundo lugar porque não é necessário armazenar dados por qualquer ordem particular, por exemplo, uma sequência alfabética ou numérica.

Como exemplo simples do que acabamos de dizer, consideremos que desejamos armazenar 100 dados numerados serialmente de 0 a 99. A maneira mais prática de o conseguir consistiria em fazê-lo por uma ordem numérica. Se o fizermos num meio de acesso serial, e em seguida procurarmos ter acesso livre a qualquer deles, por exemplo, primeiro ao 98, em seguida ao 2 e depois ao 53, tornar-se-ia muito trabalhoso passar por todos os dados até

atingir o 98.<sup>o</sup>, voltar em seguida ao 2.<sup>o</sup> e depois novamente ao 53.<sup>o</sup>. A memória de acesso serial só é de facto prática quando desejamos colher os dados pela ordem em que estão armazenados. Se no entanto armazenarmos os 100 dados por ordem num dispositivo de acesso livre, não haverá qualquer problema em localizar um deles. Desde que saibamos onde se encontra, poderemos ir directamente ao seu encontro. Este último ponto introduz o conceito de «endereçamento» dos registo de acesso livre, um princípio que discutiremos com algum pormenor mais adiante.

### *Fita magnética*

A fita magnética é um processo muito prático de armazenar grandes volumes de dados num espaço relativamente pequeno. Em essência, o princípio aqui usado é muito semelhante ao de um gravador de fita magnética para uso doméstico, sendo os dados armazenados em bobinas de fita que podem ser carregadas num reproduutor de fita como e quando se pretende.

### *Características físicas*

A fita magnética usada em armazenamento de computador é feita num plástico bastante forte, como por exemplo o designado por Mylar. Este é revestido de um material que pode ser magnetizado e é mantido em bobinas, cujo comprimento varia entre 37 e 110 metros. Se bem que a largura da fita varie de sistema para sistema, a mais vulgarmente usada tem 12,7 mm de largo.

A fita é tratada num reproduutor que possui três elementos principais (ver a figura 53):

1. Um dispositivo de gravação, leitura e apagamento.

2. Um mecanismo para passagem da fita em frente das cabeças de leitura e escrita.

3. Duas bobinas entre as quais se pode movimentar a fita.

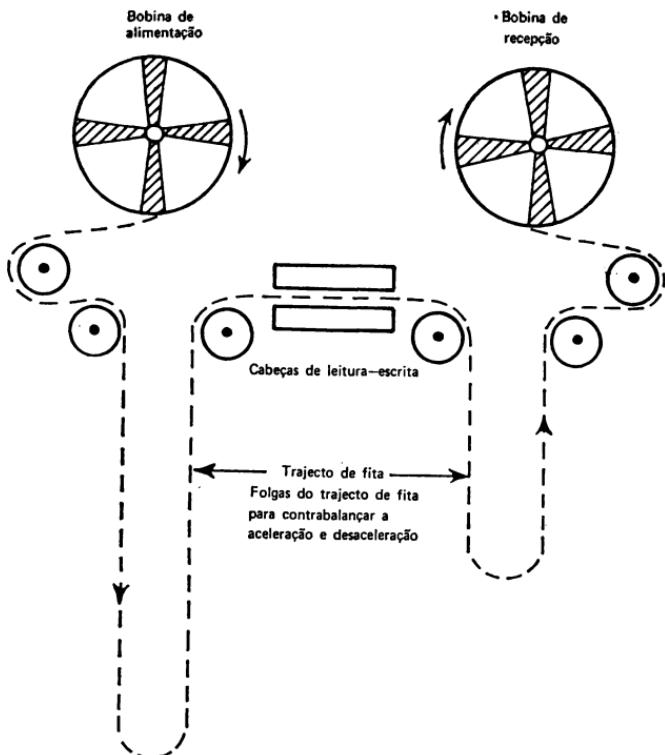


Fig. 53 — Reprodutor de fita magnética (esquema).

#### *Modo de gravação*

Os caracteres são gravados na fita utilizando um modo bastante semelhante ao já descrito para o caso da fita de papel perfurada, excepto no que se refere ao facto de,

em vez dos orifícios representando bits, serem magnetizadas pequenas áreas ou pontos em toda a largura da fita, na direcção oposta ao campo magnético permanente da mesma. Os bits são posicionados longitudinalmente na fita, naquilo que é designado por bandas (figura 54). O número de bandas existentes na fita varia de sistema para sistema. A figura 54 mostra uma secção de uma fita de oito bandas, das quais sete são usadas para registo de caracteres e a oitava para verificação de paridade. O número de caracteres que podem ser acomodados num dado comprimento de fita varia igualmente de sistema para sistema. Este dado é referido pelo nome de *densidade* e é expresso em termos de caracteres por milímetro ou polegada. As densidades mais vulgares são de 8, 22, 31 e 63 por milímetro (200, 550, 800 e 1600 por polegada).

A informação só pode ser escrita e lida na fita, através do processador central. Com efeito, isto significa que os dados são lidos para o processador central, mantidos temporariamente na sua memória, processados e, em seguida, escritos novamente em fita magnética. É impraticável ler dados a partir de uma fita, processá-los e escrevê-los novamente na mesma fita, dado que, durante a fase de processamento, o comprimento físico do registo de dados pode perfeitamente aumentar. Haveria então um espaço insuficiente para eles na fita anteriormente ocupada. A operação deve pois ser lida numa fita, realizada, e os seus resultados inscritos numa segunda fita. A dimensão da memória do processador determinará o volume máximo de dados que podem ser tratados numa operação de leitura escrita. A expressão que designa este volume de dados transferidos numa operação é *bloco de dados*, mas deve-se mencionar que a dimensão da memória principal (a do processador central) não constitui o único factor limitativo; considerações de sistema em termos de números e dimensão dos registos de dados desempenham aqui igualmente um papel.

Tendo transferido um bloco de dados, o processador necessitará de tempo para executar as operações necessárias. Continuar a rodar a fita durante este tempo con-

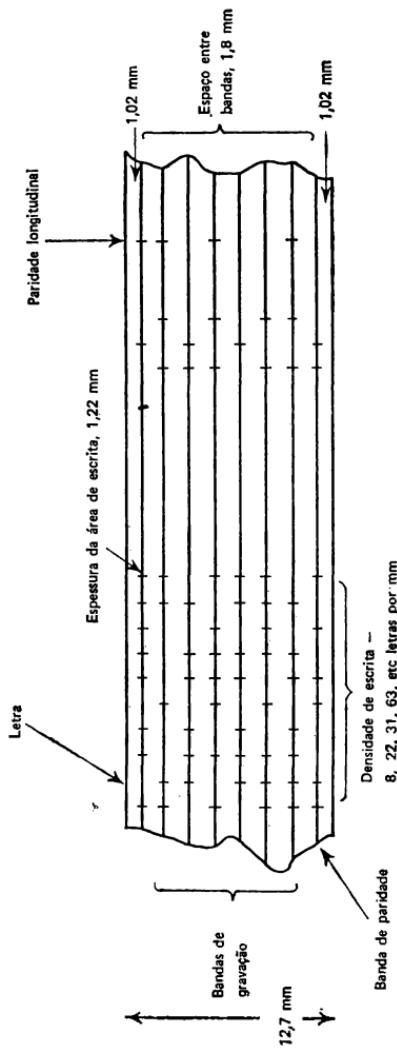


Fig. 54 — Exemplo de uma secção de fita magnética.

duziria evidentemente à perda de longas secções de fita não gravada. Para o evitar, a fita é parada entre cada operação de leitura e escrita. Mesmo assim, subsiste ainda um espaço em branco, se bem que muito pequeno, provocado pelo processo de desaceleração antes de parar e de aceleração até à velocidade necessária de gravação. Este espaço em branco entre dois blocos de dados é designado por *espaço entre blocos*.

### *Organização de dados*

Como é evidente, uma sucessão contínua de filas de dígitos binários, representando caracteres, numa fita não teria em si própria qualquer significado. Seria semelhante a uma página de livro coberta de linhas com sucessões contínuas de letras. Para que as letras tenham um sentido, devem-se separar as palavras, sendo assim cada uma destas distinguida das outras por um espaço; as palavras devem ainda ser arranjadas em frases, estas em parágrafos, etc. Do mesmo modo, na fita, cada grupo de caracteres representando um campo de dados deve ser bem definido, e o conjunto de campos que constituem um registo de dados deve ainda ser indicado. Isto é feito inserindo «marcadores», ou seja, padrões especiais de bits que o computador saberá reconhecer, mostrando assim onde é que cada campo, registo e bloco termina. Estes marcadores serão designados por *marcador de fim de campo*, *de fim de registo* e *de fim de bloco*. A figura 55 ilustra este arranjo dos dados numa fita magnética.

Como vimos anteriormente, é impraticável escolher uma fita magnética para gravar registos de tratamento livre. O tempo envolvido na passagem de uma fita de uma ponta à outra, até ser encontrada uma dada palavra de computador, seria proibitivo. É portanto vulgar utilizar o armazenamento em fita apenas quando os registos são processados sequencialmente, isto é, por uma ordem previamente determinada, numérica ou alfabética.

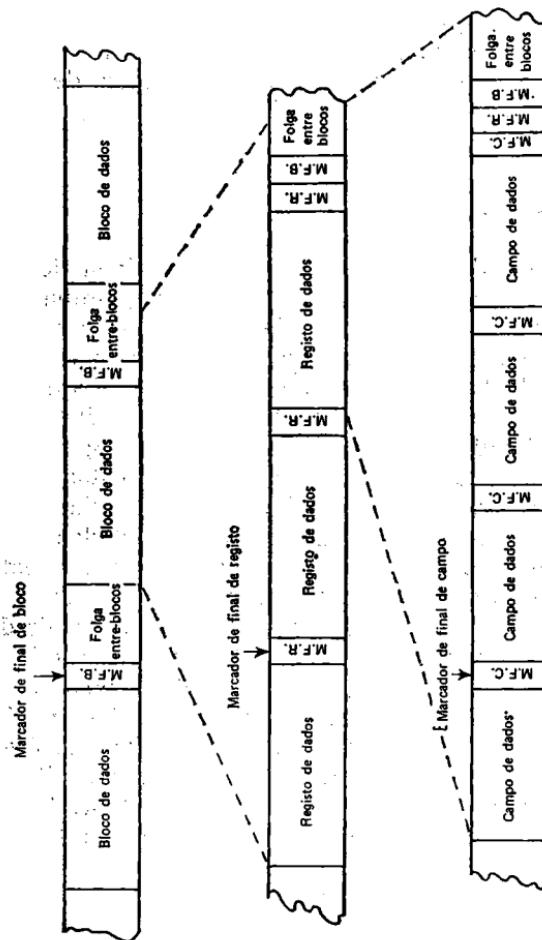


Fig. 55 — Organização de dados em fita magnética.

### *Velocidades de funcionamento e capacidade*

Os dois factores principais que determinam a transferência de dados para e do processador central são:

- a) a velocidade da fita sobre a cabeça de leitura/escrita,
- b) a densidade.

Em teoria, a máxima velocidade de transferência de uma fita que se movimenta a 305 centímetros por segundo (120 polegadas por segundo) com uma densidade de 31 caracteres por milímetro (800 por polegada) é de 96 K c.p.s. No entanto, isto não toma em consideração o tempo necessário para as operações de arranque e paragem nos espaços entre blocos. Na prática, a velocidade de transferência será consideravelmente inferior a esta.

Os sistemas de fita variam tanto no que se refere à densidade permitida e à velocidade de movimento da fita, que se torna impraticável definir uma velocidade «standard» de leitura/escrita. Esta poderá ser qualquer coisa entre 10 K até 300 K c.p.s. Provavelmente, uma média aceitável estará na vizinhança de 150 a 200 K c.p.s.

A capacidade de uma bobina de fita dependerá evidentemente de um certo número de variáveis. Estas incluirão a densidade, considerações de concepção como sejam as dimensões dos blocos de dados e a frequência resultante de espaços entre blocos, e evidentemente o comprimento da própria fita. Como exemplo, pode perfeitamente acontecer que uma fita com 84 metros de comprimento (2400 pés) contenha mais de 15 milhões de caracteres.

### *Memória de acesso directo*

Antes de rever os tipos de memórias auxiliares que entram na categoria das memórias de acesso directo,

talvez seja bom definir duas expressões que surgem frequentemente. São elas:

*Tempo de acesso.* Trata-se do tempo médio necessário para localizar qualquer dado escolhido existente no dispositivo. Esta consideração não se aplica à fita magnética, dado que esta só pode ser tratada serialmente.

*Velocidade de transferência.* A velocidade a que os dados, depois de localizados, podem ser transferidos do e para o processador central. Isto pode ser expresso em termos de caracteres por segundo, abreviado na forma de K c.p.s., onde K é um prefixo que designa 1000.

#### *Tambores magnéticos*

Antes do aparecimento dos registos de núcleos de ferrite, os tambores magnéticos eram usados em muitos dos primeiros computadores como tipo de memória de acesso imediato. Actualmente, aumenta a sua utilização como forma de armazenamento auxiliar de dados em massa.

Basicamente, um tambor magnético consiste num cilindro feito num material não magnético, cuja superfície se encontra coberta com um revestimento magnético. Esta superfície está dividida num certo número de pistas circulares paralelas, sendo cada uma delas por sua vez dividida num certo número de secções de dimensão normalizada e capaz de registar um dado número de dígitos binários.

Os termos *banda* e *palavra* são utilizados em geral como sinónimos de pista e secção, respectivamente. Cada banda tem um número de referência único e, por sua vez, cada palavra é numerada no interior da banda onde se encontra (figura 56). Isto significa que o conteúdo de cada posição de registo pode ser descoberto por referência ao seu número de banda e de palavra.

O tambor encontra-se montado num eixo e roda a alta velocidade sob cabeças de leitura/escrita montadas, uma para cada banda, acima da superfície do tambor. Assim, para um tambor que rode à velocidade de 3000 revoluções por minuto, todos os dados existentes na sua superfície passarão sob as cabeças de leitura/escrita em 20 milissegundos. Tendo em conta que a distância média

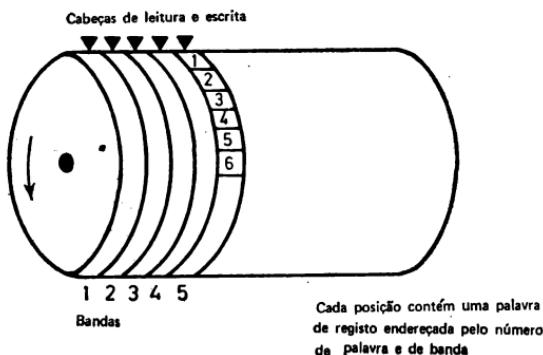


Fig. 56 — Representação diagramática de um tambor magnético.

de qualquer palavra à cabeça será equivalente a meia rotação, o tempo médio de acesso para qualquer dado seria de cerca de 10 milissegundos.

Como no caso da fita magnética, a capacidade e a velocidade de transferência de dados dos tambores variam consideravelmente de sistema para sistema. Um exemplo típico seria uma capacidade de oito milhões de caracteres com uma velocidade de transferência de 100 K c.p.s., e um tempo de acesso de 10 milissegundos.

#### *Discos magnéticos*

Estes constituem a forma mais popular de registo de acesso directo actualmente utilizada em computadores.

*Características físicas.* Um disco possui, em cada um dos lados, uma superfície gravável na qual podem ser magnetizadas pequenas áreas ou pontos, de modo a representarem dígitos binários, seguindo um princípio muito semelhante ao do tambor. A superfície do disco é dividida num certo número de círculos concéntricos, designados por *bandas*, ao longo dos quais se realiza a gravação. A superfície é igualmente dividida em sectores por um certo número de linhas não gravadas que se estendem do centro do disco para a sua periferia (figura 57). Cada sector de cada banda pode registar um número máximo fixo de caracteres.

Existem dois sistemas principais de discos. Um utiliza um único disco grande que se encontra permanentemente montado numa unidade designada por *transporte do disco*. Este sistema é considerado como de *disco fixo*. O outro, bastante mais popular, utiliza um certo número de discos menores, montados paralelamente uns aos outros num eixo central, fornecendo assim um número múltiplo de superfícies de gravação. Estes conjuntos de discos podem ser montados à vontade no transporte ou dele removidos. É designado por *sistema de discos intermutáveis*. O transporte é mantido permanentemente «on-line» com o processador, ao passo que os conjuntos de discos («disc packs») que armazenam dados ou programas podem ser guardados longe do computador para utilização quando necessário.

No caso do sistema de discos intermutáveis, existe um certo número de cabeças de leitura/escrita no transporte, uma para cada superfície de gravação. Estas encontram-se fixas nas extremidades de braços móveis, dois por disco, de modo que podem ser movidas nos espaços entre os discos, varrendo cada cabeça uma superfície de disco. Estes braços encontram-se por sua vez fixos a uma montagem conhecida pelo nome de *braço de arrasto*. As cabeças individuais de leitura e escrita não se podem movimentar independentemente umas das outras, movendo-se todas simultaneamente com o braço de arrasto. Isto significa que, em qualquer momento, todas as bandas

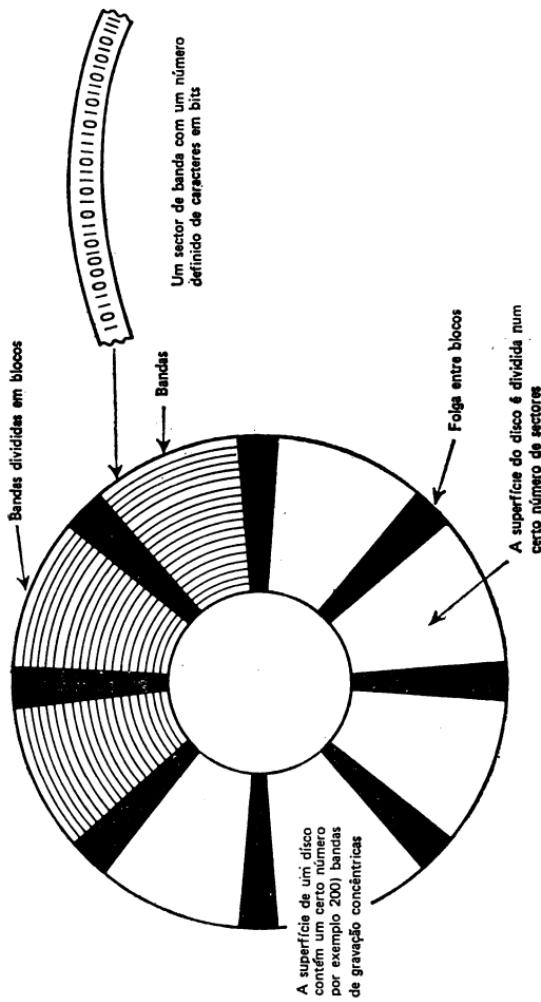


Fig. 57 — Uma superfície de disco magnético.

que se encontram na mesma posição relativa em cada disco são lidas simultaneamente. O conjunto de discos roda a alta velocidade relativamente às cabeças de leitura/escrita, dirigindo cada sector das bandas para a cabeça respectiva (figura 58).

Cada carácter é gravado sob a forma de uma sucessão de bits ao longo da banda. Para distinguir cada carácter do seguinte, a banda encontra-se subdividida, em cada sector, em pequenos grupos de bits, normalmente seis, sendo cada um desses grupos capaz de guardar um carácter. As operações de leitura/escrita são realizadas por comando do programa, transferindo dados entre o disco e a memória do processador. No entanto, deve-se incorporar um factor importante nessas instruções, a saber, o endereço ou localização do sector do disco do qual ou para o qual se deve transferir informações. Este endereço é determinado por referência a três factores: a superfície do disco, o número da banda na superfície

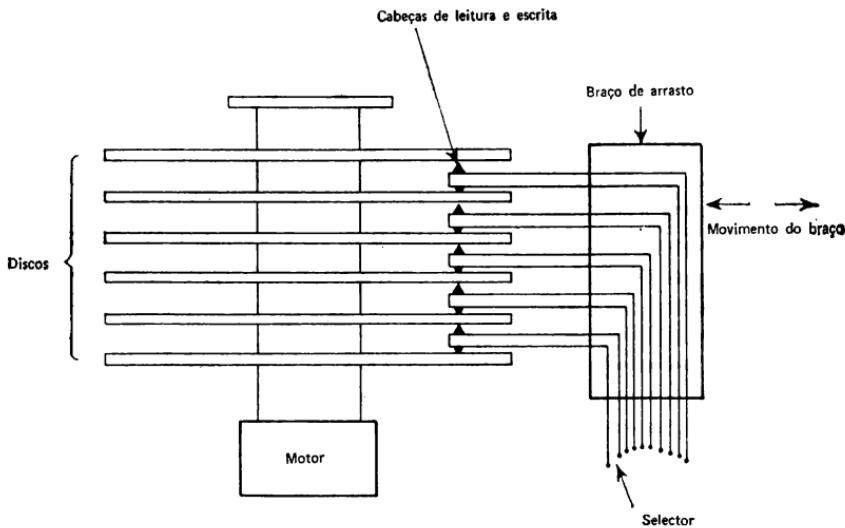


Fig. 58 — Unidade intermutável de discos mostrando a selecção de bandas e de superfícies.

e o número do sector dentro de cada banda. No caso de estar ligado ao computador mais de um transporte, torna-se necessário um quarto elemento no endereço, que identificará o transporte onde se encontra o disco.

*Organização dos dados.* Como vimos, um disco é dividido num certo número de unidades de armazenamento de dimensão fixa. Num determinado sistema, por exemplo, a capacidade de gravação de um sector é de 512 caracteres. O espaço ocupado por registo de dados individuais pode variar bastante. Em alguns casos, pode ser possível montar mais que um num único sector, outros talvez seja necessário todo um sector para acomodar um registo e outros ainda o registo pode ser tão comprido que requeira dois ou mais sectores. No entanto, qualquer que seja o espaço atribuído a um registo de dados, devemos ter em conta a probabilidade de, durante o processamento, o comprimento do registo ser aumentado pela adição de novos dados.

Surge assim o problema do que poderá acontecer no caso de o registo se tornar demasiado grande para o espaço originalmente disponível. Não podemos tentar forçar a gravação, senão a parte final do registo de grandes dimensões apagará o início do registo seguinte. Para resolver esta situação, é habitual reservar um certo número de bandas designadas por «área livre». Quando um registo ultrapassa a sua dimensão original, é passado para esta área livre e é deixada uma «mensagem» na sua localização anterior, dizendo para onde foi deslocado o registo. A procura do registo iniciar-se-á pela banda original, sendo então enviada, pela indicação nesta existente, para a banda livre.

Os registos de dados em disco podem ser armazenados sequencialmente ou livremente. Cada registo terá um factor indicativo único, conhecido por *chave de registo*. Esta é normalmente apenas um número, por exemplo, um número de conta bancária. Quando os registos são armazenados sequencialmente, são-no pela ordem de registo. Em armazenamento livre, não existe evidentemente uma

ordem particular. Se se pretende processar sequencialmente os registos, haverá uma óbvia vantagem de tempo se forem armazenados pela mesma ordem, dado que as localizações de memória são encontradas serialmente. Além disso, depois de localizado o endereço do primeiro registo da sequência, não é necessário preocuparmo-nos com o endereço do segundo, etc. dado que se seguem uns aos outros.

*Organização da gravação.* Se desejamos gravar, por exemplo, 100 registos em disco, poderá à primeira vista parecer que a melhor maneira de o fazer consistiria em começar a gravar na banda exterior de um disco escolhido, quando esta se encontrasse cheia passar à banda seguinte, etc., utilizando talvez dez bandas com dez registos em cada uma.

Porém, se pensarmos melhor, verificaremos que, nestas condições, de todas as vezes que tentarmos procurar um registo noutra banda, a cabeça de leitura/escrita deve ser movimentada. Isto levará cerca de 80 milisegundos. Se, no entanto, usarmos a banda correspondente em cada uma das dez superfícies graváveis, obtendo uma montagem vertical das bandas em vez de horizontal (figura 59), teremos uma situação em que todos os registos são acessíveis sem qualquer movimento da cabeça de leitura/escrita. O nome dado a um conjunto de bandas utilizadas deste modo é um *cilindro*. Como é evidente, pode-se tornar necessário um certo número de cilindros adjacentes para acomodar um conjunto de dados, o que significará que deverá haver um certo movimento da cabeça para aceder a cada cilindro, mas a técnica permite no entanto obter uma área de utilização que maximiza a quantidade de dados que podem ser investigados em qualquer movimento.

*Velocidade de funcionamento e capacidade.* É difícil fornecer velocidades de acesso, velocidades de transferência ou capacidades de armazenamento típicas, no caso dos discos, devido à vasta gama dos sistemas

actualmente usados. O número de bandas por disco varia entre 100 e 200 e a densidade, ou seja, o número de bits que pode ser registado, por exemplo, numa polegada de banda, poderá ser de apenas 250 ou atingir cerca de 3000. No entanto, deve-se notar que estas densidades também variam ao longo de uma mesma superfície de disco, dado

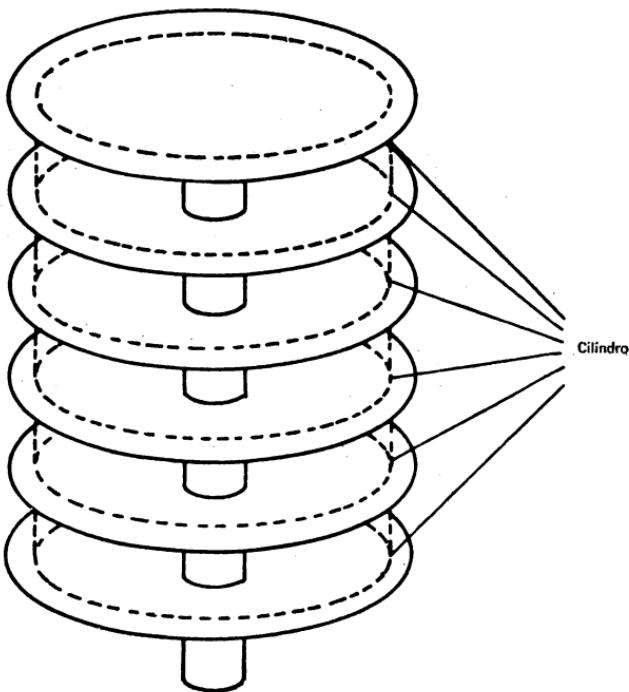


Fig. 59 — Esquema de cilindro de armazenamento.

que o comprimento de uma banda perto do centro deste será bastante inferior ao mesmo comprimento na periferia. As velocidades de transferência são normalmente de 200 a 350 K c.p.s. O tempo de acesso é representado pelo tempo levado pela cabeça de leitura/escrita para se movimentar sobre a superfície do disco até localizar a

banda correcta (*tempo de procura*) mais o tempo necessário para o disco rodar até ser localizado o endereço debaixo da cabeça (*latência*). Estes dois factores estarão evidentemente relacionados com a velocidade de rotação do disco e também com o número de bandas por superfície. No parágrafo seguinte, pretendemos apenas dar um exemplo de uma memória de discos intermutáveis.

Um transporte de disco contém um conjunto de seis discos de 14 polegadas (35 cm) dando um total de dez superfícies de gravação (as duas superfícies exteriores, em cima e em baixo, não são usadas para registo de dados). Existem 200 bandas por superfície, cada uma dividida em oito blocos. Cada bloco registará 512 caracteres, dando 4096 caracteres por banda, 819 200 por superfície e 8 192 000 por conjunto de discos. Cada carácter é guardado em seis bits, constituindo quatro caracteres a unidade básica de armazenamento — uma palavra de 24 bits. O disco roda a 2400 rev/min. A velocidade de transferência é de 208 K c.p.s. e o tempo médio de acesso é de 97,5 milissegundos, com uma latência de 12,5 milissegundos e um tempo médio de procura de 85 milissegundos. Pode-se manter «on-line» simultaneamente um certo número de transportes de discos. Oito unidades destas dariam um máximo de capacidade de armazenamento on-line de 65 536 milhões de caracteres (o que equivale, aproximadamente, a 250 livros do tamanho deste).

#### *Cartões magnéticos*

O meio de gravação neste caso assume a forma de um cartão, se bem que o princípio de gravação magnética de bits na sua superfície seja semelhante ao utilizado no caso do disco ou da fita. Os cartões variam em dimensão de sistema para sistema, com uma superfície de gravação de cerca de 320 a 450 cm<sup>2</sup> — por exemplo, 335 × 89 mm. A superfície de gravação é dividida num certo número de bandas paralelas, sendo cada uma delas dividida por sua vez num determinado número de segmentos, dando

uma unidade de armazenamento de capacidade fixa (figura 60).

Os cartões são montados em cartuchos onde se pode escolher um cartão individual, transportá-lo e movimentá-lo em torno a um eixo rotativo, sobre o qual estão suspensas cabeças de leitura/escrita. A identificação do

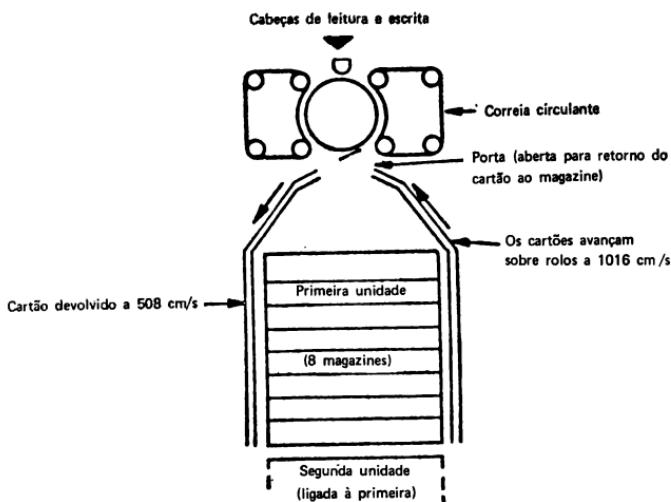


Fig. 60 — Esquema de dispositivo de cartões magnéticos.

cartão é realizada através de um padrão único de ranhuras no seu lado superior.

É habitual montar um certo número de cabeças de leitura/escrita sobre o eixo, cada uma capaz de se movimentar e localizar assim um certo número de bandas — por exemplo, 36 cabeças movendo-se para qualquer uma de quatro posições, cobrindo assim operações em 144 bandas. A capacidade de armazenamento de um cartão magnético pode ser bastante grande. Sendo cada cartão capaz de armazenar 200 K caracteres, um dispositivo que contenha oito cartuchos de 256 cartões cada terá uma capaci-

dade de armazenamento total de mais de 5.000 milhões de caracteres.

No entanto, se bem que os sistemas de cartões tenham a vantagem do seu grande volume de armazenamento num meio de gravação relativamente barato, não obtiveram a popularidade do disco ou da fita. Devido aos processos mecânicos envolvidos — movimento do cartão e posicionamento da cabeça de leitura — os tempos de acesso tendem a ser grandes, variando entre 200 e 500 milissegundos. O aspecto mecânico do dispositivo apresenta aliás alguns problemas. Se os cartões se movimentarem a, por exemplo, 960 cm/s, há a possibilidade de deterioração do cartão com a consequente perda de informação gravada.

#### *Faixas magnéticas*

Um processo em princípio bastante semelhante ao dos cartões magnéticos é o sistema IBM Datacell, utilizando como meio de gravação faixas magnetizáveis de cerca de  $57 \times 330$  mm. Cada uma é dividida em 100 bandas com uma capacidade de 2000 bits cada. As faixas são montadas em grupos de dez, constituindo uma subcélula; um conjunto de 20 subcélulas constitui uma célula. A célula tem a forma de um segmento de cilindro, obtendo-se um carretel de forma cilíndrica quando se juntam dez células. Isto dá uma capacidade total do carretel de 2.000 faixas. As células constituem unidades de armazenamento que podem ser inseridas ou retiradas do carretel conforme as necessidades.

Para operações de leitura/escrita é escolhida uma faixa individual por referência a uma identificação codificada, extraída da célula e passada por um eixo rotativo como no caso anterior. Este roda a 1200 rev/min, formando efectivamente um pequeno tambor magnético. Sobre o eixo, encontram-se suspensas 20 cabeças de leitura/escrita, uma para cada cinco bandas de gravação, susceptíveis de ser colocadas sobre qualquer das cinco, de

modo a que, em qualquer momento, cada quinta banda se encontre numa posição de gravação, e seja possível aceder simultaneamente a 20 bandas. Este arranjo significa que, ao armazenar dados, é possível utilizar o sistema do cilindro, através do mesmo princípio já discutido anteriormente para os discos magnéticos. Uma faixa conterá cinco cilindros, cada um com 20 bandas.

A capacidade total de um carretel completo, contendo dez células de dados, é de 400 milhões de caracteres, e o tempo máximo de acesso é de 600 milissegundos.

#### *Barras magnéticas*

Trata-se de uma forma de armazenamento extensivamente usada em computadores de registo visível de pequenas dimensões e não nas máquinas maiores. São discutidas no Capítulo 7.



# 7

## CÓMO FUNCIONAM OS PROGRAMAS DE COMPUTADOR

Um programa de computador é apenas uma lista de instruções que diz ao computador o que deve fazer. Naturalmente, aquilo que podemos esperar que faça será aquilo que pode fazer... O conjunto de coisas que pode fazer é restringido pela especificação e construção da máquina. Assim, as instruções do programa devem ser limitadas às funções específicas da máquina. Por outro lado, em qualquer tarefa por nós executada, com ou sem o auxílio de máquinas, se se pretende que as operações sejam executadas convenientemente, deve-se seguir uma sequência lógica. Do mesmo modo, as instruções contidas num programa de computador devem ser ordenadas e apresentadas à máquina por uma sequência lógica que lhe permita realizar a tarefa que dela se espera.

### *Uma instrução de programa*

Para ter significado, qualquer instrução em qualquer área de actividade deve incorporar dois factores. Por exemplo, se pretendemos abrir um buraco na parede a fim de colocar um parafuso no qual possamos pendurar um quadro, uma simples instrução «esburacar» teria como resultado a abertura de um bom número de buracos

na parede sem servirem para nada... A instrução em causa deve conter uma especificação quanto ao local onde se pretende abrir o buraco. Uma instrução não deve referir apenas o que há a fazer, mas também especificar onde se deve realizar a operação. Estas duas partes de uma instrução podem ser designadas por *operação* e *operando*, respectivamente.

### *Operação*

Esta será pois a parte da instrução que define *o que* se pretende fazer. Como se sugeriu, em qualquer máquina ou peça de equipamento há um limite quanto à gama de operações que pode realizar. Uma máquina de escrever imprimirá letras, números, espaços, voltará atrás, etc., ou seja, realizará um certo número de tarefas, mas não servirá, por exemplo, para somar. Acontece também com qualquer dispositivo que nenhuma das suas funções será executada sem uma instrução, mesmo que esta se traduza apenas em premir uma tecla ou fechar um interruptor.

O mesmo princípio se aplica ao computador. É construído para realizar uma gama finita de funções específicas e a realização de cada uma delas depende das instruções necessárias. No exemplo da máquina de escrever, cada operação é identificada por uma tecla tendo uma inscrição única. Num computador, cada operação é identificada por um código binário único, armazenado permanentemente no processador. Este é conhecido pelo *código de instruções* da máquina, *código máquina* ou *código de funções*.

É este código que deve ser referido nas instruções do programa. Quando o processador está pronto para executar uma instrução, compara este código de operação com os que se mantêm permanentemente na sua memória até encontrar uma equivalência. Liga então automaticamente os circuitos que executarão a operação pretendida.

### *Operando*

A parte «operando» das instruções identifica os dados com que a operação deve ser executada. Não o faz definindo o que são os dados, mas especificando onde devem ser encontrados. Todos os dados existentes na memória de um computador se encontram em posições (palavras) da memória, cada uma delas com um endereço único. É este endereço da palavra que contém os dados que é designado como operando.

Este arranjo de operação e operando é conhecido pelo nome de *formato da instrução* (figura 61). Na sua

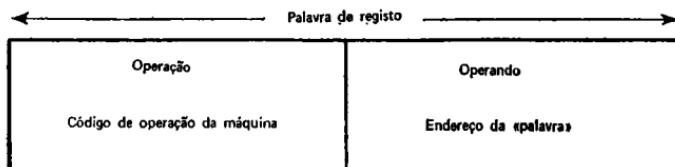


Fig. 61 — Um formato de instrução de endereço único.

forma mais simples, este formato conterá o código da instrução e um endereço, sendo portanto referido como um *formato de instrução de endereço único*. No entanto, os formatos de instrução variam de máquina para máquina e podem conter dois, três ou mesmo quatro endereços na parte operando. Por exemplo, se desejamos que o computador realize o seguinte: «somar o conteúdo da posição 123 ao conteúdo da posição 234 e registar o resultado na posição 345», as instruções dadas em três diferentes formatos, de um, dois ou três endereços poderiam ser as seguintes:

*Formato de endereço único* (serão necessárias neste caso três instruções diferentes):

1. Colocar no acumulador o conteúdo de 123.
2. Somar o conteúdo de 234 ao acumulador.

3. Movimentar o conteúdo actual do acumulador para 345.

Usando codificações de funções numéricas imaginárias, teríamos:

Código da operação	Endereço do operando
1	24
2	30
3	34

*Formato de dois endereços* (serão então necessárias duas instruções):

1. Somar o conteúdo da posição 123 ao conteúdo da posição 234.

2. Movimentar o conteúdo da posição 234 para 345:

Código da operação	Endereço do 1.º operando	Endereço do 2.º operando
1	30	123
2	34	234

*Formato de três endereços* (requerendo apenas uma instrução):

Somar o conteúdo da posição 123 ao da posição 234 e guardar o resultado em 345.

Código da operação	Endereço do 1.º operando	Endereço do 2.º operando	Endereço do 3.º operando
30	123	234	345

## *Instruções de armazenamento*

Para que uma dada tarefa seja realizada pelo computador, devem estar presentes dois factores essenciais: (a) a instrução de programa; (b) os dados com os quais se pretende trabalhar. Ambos estes factores são mantidos na memória do processador central, enquanto são imediatamente necessários. A sua «casa» permanente será provavelmente nas memórias auxiliares. O programa, cujos elementos são as instruções, deve ser transferido para a memória, quando necessário, sendo consultado da mesma maneira que os dados são transferidos da e para a memória quando o programa os pede.

As palavras em que a memória é dividida servem um objectivo duplo: armazenar dados ou uma instrução de programa, e não tem qualquer interesse que palavras são usadas para cada uma dessas funções. Como vimos anteriormente, a mecânica da localização dos dados assenta na localização do seu endereço como operando da instrução. Não podemos localizar uma instrução deste modo, dado que a instrução não pode ser o veículo do seu próprio endereço. As instruções do programa devem portanto ser armazenadas em endereços consecutivos de tal modo que, ao terminar a execução de uma delas, o processador passe automaticamente para o endereço imediato, encontrando a instrução seguinte. Como é evidente, o endereço da primeira instrução deve ser comunicado ao processador antes de esta sequência se poder iniciar. Isto é realizado directa ou indirectamente pelo operador. Surgirá outro problema no caso de a distribuição de dados na memória ser tal que deixe insuficientes palavras consecutivas para acomodarem todo o programa. Neste caso, não há qualquer razão para não armazenar o programa num certo número de grupos separados de palavras, desde que todas as palavras em cada um desses grupos sejam consecutivas e a instrução da última palavra indique o endereço da primeira palavra do grupo seguinte. Isto é ilustrado na figura 62.

11110 Dados	11111 Dados	11112 Instrução do programa	11113 Instrução do programa	11114 Instrução do programa	11115 Instrução do programa
11116 Instrução do programa.	11117 Instrução do programa	11118 Dados	11119 Dados	11120 Dados	11121 Dados
11122 Dados	11123 Dados	11124 Dados	11125 Dados	11126 Dados	11127 Instrução do programa
11128 Instrução do programa.	11129 Instrução do programa	11130 Instrução do programa	11131 Instrução do programa	11132 Instrução do programa	11133 Instrução do programa

As instruções do programa são trabalhadas sequencialmente, começando pela palavra 11112. A palavra 11117 conterá uma indicação de passagem para a palavra 11127.

Fig. 62 — Armazenamento de instruções de programa e de dados no processador central.

## *Execução de uma instrução*

Consideremos que temos um certo número de instruções armazenadas numa sequência de endereços e um número de dados numa outra sequência. Pretendemos que o processador realize as instruções, pela ordem em que estas se encontram armazenadas, sobre dados escondidos.

A primeira coisa que o processador deve fazer consiste em encontrar o endereço da primeira instrução que, como vimos anteriormente, será provavelmente comunicada pelo operador da máquina. Este endereço é então introduzido e armazenado num registo especial, conhecido pelo nome de *registo de programa*. A função deste registo consiste agora em controlar cada passo do ciclo de operações necessário para executar a instrução. Em seguida, fornece-se um exemplo que explicará a maneira como estes passos devem ser executados, se bem que não sejam necessariamente comuns a todos os tipos de máquina.

1. O endereço da instrução contida no registo de programa é movimentado para um registo de endereço, que por sua vez o introduz num selector de endereços, onde é comparado ao endereço das palavras armazenadas até ser descoberta uma coincidência.
2. O acesso a esta palavra é agora estabelecido e o seu conteúdo, ou seja, a instrução propriamente dita, é movimentada para outro registo designado por *registo de instrução*. Este conterá o código da operação e o endereço do operando.
3. O código da operação é movido para um *descodificador da operação*. Este contém uma série de portas, uma para cada operação que a máquina deve executar. Uma porta emitirá um impulso de activação quando o código da máquina que representa a operação por ela controlada é submetido a uma entrada. O código de entrada, idêntico ao código da máquina, é submetido a todas

as portas e será rejeitado por todas, excepto as designadas para activar a operação que o mesmo código representa. O impulso então emitido provoca a execução da operação.

4. O endereço do operando é copiado do registo de instrução para o registo de endereço, que por sua vez o introduz no selector de endereços como no passo 1.

5. O conteúdo do endereço do operando, ou seja, o conjunto de dados a trabalhar, é transferido para um acumulador ou para a unidade aritmética.

6. É então executada a operação pretendida na unidade aritmética e o resultado é lançado novamente no acumulador.

7. O endereço de instrução contido no registo de programa é agora aumentado de 1 num contador, obtendo-se assim o endereço da instrução seguinte. A rotina volta agora ao passo 1, e recomeça-se o ciclo da instrução seguinte.

A figura 63 ilustra este ciclo de execução de uma instrução de programa.

### *Níveis de programa dos computadores*

Até agora temos pensado em instruções de programa em termos de afirmações numéricas codificadas que são mantidas na memória sob a forma de expressões binárias. De facto, em última análise, é esta a única forma que a máquina reconhecerá e aceitará. Um programa escrito deste modo, indicando a operação na instrução sob a forma de um código de operação numérico e o operando como endereço numérico absoluto da palavra, é conhecido pelo nome de *programa de código-máquina* e é por vezes referido como *codificação absoluta*. Como é evidente, se um programador estivesse a escrever um programa nesta

forma, não utilizaria uma representação binária dos números mas sim os dígitos decimais normais. As instruções seriam então perfuradas num cartão ou numa fita e a conversão decimal para binário realizar-se-ia automaticamente na leitura.

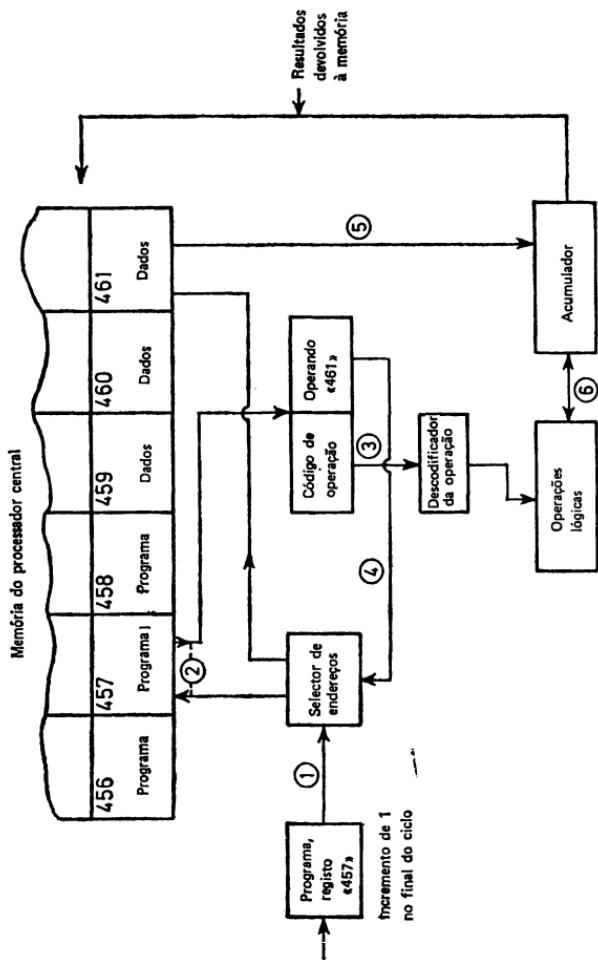


Fig. 63 — Execução de uma instrução de programa.

Se pensarmos na parte operativa da instrução, verificaremos que, se bem que a indicação desta por um número tem a vantagem de ser directamente reconhecível pela máquina, do ponto de vista do programador é inconveniente, dado que deve memorizar ou seguir uma lista procurando a codificação de cada função, escrita no seu próprio programa. Seria muito simples se o código da operação tivesse alguma semelhança com a descrição da função. Nos primeiros tempos do trabalho em computadores, isto conduziu ao uso de códigos alfabéticos que, em si, poderiam ser reconhecidos como uma função. Com efeito, este princípio ainda hoje é usado em algumas linguagens de computador. Este tipo de código é designado por *mnemónica* (da palavra grega que significa memória). Se bem que não exista qualquer código mnemónico comum a todas as máquinas — os fabricantes tenderam a desenvolver os seus próprios códigos individuais — o princípio geral consiste em relacionar o código com a função. Por exemplo, o código de multiplicar é MULT, o código de transferência é TRAN, o código de soma é ADD (somar, em inglês), etc. Isto significa que uma instrução em formato de endereço único poderia ser escrita:

*Operação*  
LOAD

*Operando*  
3456

(mnemónica de acumulador de carga — «load accumulator»).

Utiliza-se assim ainda um endereço de operando absoluto. Surge no entanto o problema de o computador ser incapaz de compreender o termo MULT. Só é capaz de aceitar e actuar de acordo com esta instrução se for representada, por exemplo, por 11101. Isto significa que é necessário instalar qualquer tipo de dispositivo de tradução que converta a primeira expressão na segunda. Recordando que quando MULT atinge o processador pode ser expresso por 100100 110011 100011 110010 (decimal

em codificação binária), devemos ter no processador alguma indicação de que

100100 110011 100011 110010 = 11101

É suficientemente simples preparar uma indicação deste género, mostrando a mnemónica equivalente para cada código de operação, e armazenar todas essas indicações na memória de modo que a mnemónica possa ser traduzida no código absoluto da máquina, que é então armazenado sob esta forma como parte do formato de instrução.

Podemos levar a questão um pouco mais longe e concentrar a nossa atenção na parte de operando da instrução, argumentando que se deveria aplicar aí o mesmo princípio. A especificação das posições de endereço como endereços absolutos significa que o programador deve manter uma lista das posições com uma nota do que nelas se encontra. Isto é um processo muito laborioso. Do ponto de vista da programação, seria muito mais simples indicar o nome dos dados em vez do endereço em que se encontram. Utilizando uma mnemónica para o operando, a instrução anterior poderia assumir a forma seguinte:

MULT

CASH

(«dinheiro» em inglês; será o nome dos dados existentes na posição 3456)

Neste caso, estamos a usar um endereço simbólico em vez de um endereço absoluto. Surge novamente o problema de a máquina dever traduzir o endereço simbólico em endereço absoluto e, além disso, dever produzir sozinha este endereço absoluto. Por outras palavras, ao receber uma instrução que mande ler qualquer coisa chamada «cash», o processador deve procurar na sua memória, encontrar um lugar livre e tomar nota de onde «arrumou»

a informação. Relaciona então o endereço simbólico com o endereço absoluto. Quando mais tarde recebe a instrução em causa, MULT CASH, pode procurar «cash» na sua «lista» para descobrir o sítio onde colocou a informação.

Um programa usado utilizando estes princípios de mnemónicas e endereços simbólicos é considerado como tendo sido escrito numa *linguagem assembly* («montada»). Não se afasta do formato base da máquina e não diminui o número de instruções que devem ser escritas, havendo uma equivalência entre ambas de um para um. O programa que é usado para traduzir em código de funções da máquina é um *programa-máquina*. Neste ponto, seria conveniente introduzir mais dois termos. Trata-se de *programa primário* e *programa secundário*. O programa primário é o programa tal como foi escrito pelo programador, que neste caso o será numa linguagem *assembly*. O programa secundário é o programa em código-máquina que resulta da tradução do primeiro. Os passos da conversão de um no outro são os seguintes (figura 64):

1. O programa primário é gravado num meio de entrada, como por exemplo os cartões perfurados ou a fita de papel perfurada.
2. O programa-máquina é lido na memória do processador (este programa é fornecido pelo fabricante juntamente com a máquina e é mantido em algum tipo de memória auxiliar, por exemplo, em cartões perfurados ou fita magnética).
3. O programa primário é agora lido para a memória, onde cada afirmação é comparada com o programa-máquina, sendo então determinado o seu equivalente código-máquina.
4. As afirmações em código-máquina, que agora constituem o programa secundário, são incorporadas nas memórias auxiliares, ficando em estado de serem utilizadas pelo processador sempre que necessário.

A obtenção de um programa secundário é uma tarefa realizada uma só vez. Depois de o programa ter sido produzido, é ele que será utilizado em todos os processamentos futuros.

Um programa-máquina pode realizar as seguintes funções:

- a) Traduzir uma mnemónica de função no seu equivalente numérico em linguagem máquina.

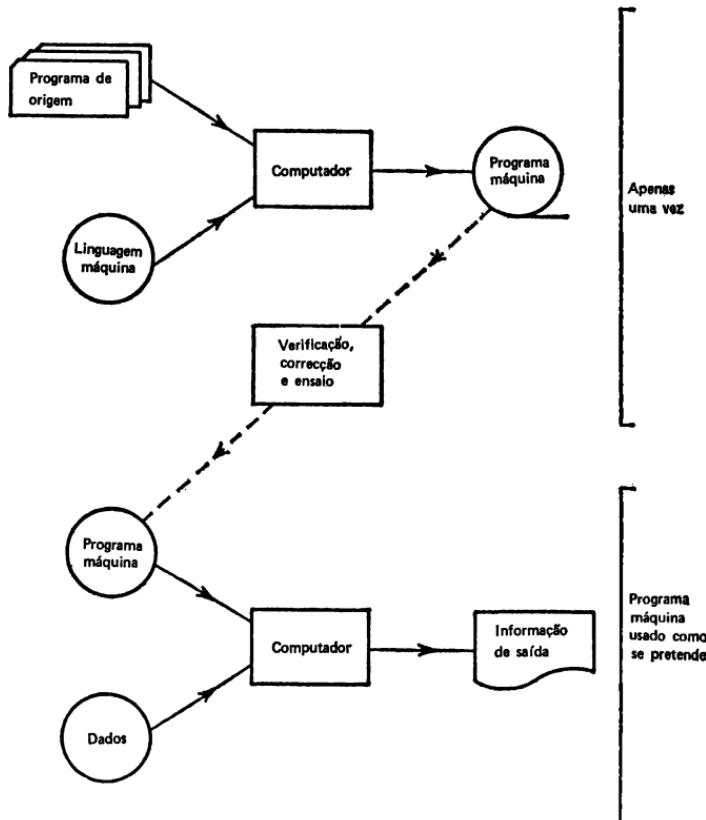


Fig. 64 — «Montagem» de um programa primário.

b) Traduzir os endereços simbólicos em endereços absolutos.

c) Traduzir as instruções de acordo com o necessário formato da máquina.

d) Atribuir áreas de armazenamento.

e) Realizar a equivalência entre endereços simbólicos e endereços absolutos.

As técnicas de programação discutidas até agora não se afastaram muito do formato básico da máquina — de facto apenas substituímos mnemónicas por números. Ao escrever estes programas, tornou-se evidente para os programadores que certas combinações de instruções ocorriam muitas vezes. Era portanto lógico escrever o nome da função que representava o grupo de instruções e dar ao programa-máquina a capacidade de reconhecer esse nome e reconvertê-lo automaticamente na série original de instruções que lhe deram origem. O tipo de instrução simples de programa que produz um conjunto de instruções em código-máquina é designado por *instrução macro*. Trata-se agora de uma relação de um para muitos.

### *Linguagens de computador*

O passo seguinte no desenvolvimento das linguagens consistia em evitar escrever informações no formato da máquina. Isto simplifica novamente o trabalho do computador, permitindo-lhe estruturar o seu programa na mesma linguagem usada para definir o problema, o que permitiu passar de linguagens orientadas em função da máquina para linguagens orientadas em função do problema. No entanto, mesmo nesta forma, existia ainda uma tendência para os fabricantes desenvolverem linguagens concebidas especificamente para as suas próprias máquinas. Isto conduziu ao desenvolvimento de linguagens universais que podem ser usadas em qualquer máquina.

No final dos anos 50, os fabricantes de computadores e os seus utilizadores começaram a congregar esforços no sentido de conceberem linguagens «standard» ou universais que fossem independentes das máquinas. Inicialmente, foram desenvolvidas linguagens universais separadas de modo a cobrirem duas áreas principais de processamento, os sistemas comerciais e os problemas matemáticos. Eram elas a linguagem *Cobol* (Common Business Oriented Language) no primeiro caso, e a *Fortran* (Formula Translation) para aplicações no sector das matemáticas. Mais tarde, foi acrescentado um outro programa para processamento de problemas matemáticos, tendo sido desenvolvido com o nome de *Algol* (Algorithmic Language).

Em 1966 foi introduzida a linguagem PL/1 como linguagem universal capaz de tratar simultaneamente aplicações comerciais e matemáticas. Um exemplo das afirmações escritas numa destas linguagens standard, a Cobol, é dado na figura 65.

Estas linguagens standard que se afastam da construção do formato da máquina são muitas vezes designadas por linguagens de *alto nível*, enquanto que as linguagens assembly e de codificação-máquina referidas anteriormente são conhecidas por linguagens de *baixo nível*.

Mas continuava a haver um problema na utilização das linguagens standard — não existia ainda uma normalização no campo da concepção do hardware. O formato básico de instrução diferia de máquina para máquina e portanto tornou-se necessário fornecer o meio de adaptar um programa escrito numa linguagem standard para uso num dado computador. Isto foi realizado por fabricantes que produziam os seus próprios programas de conversão separados, semelhantes em princípio ao programa-máquina, que traduziriam o programa standard primário no seu equivalente programa secundário. A este nível de linguagem, o programa de conversão é designado por *compilador* e o processo de conversão por *compilação*.

**Fig. 65 —** Afirmações orientadas em função do problema, em linguagem Cobol.

Antes de abandonar esta súmula das linguagens de computador, deve-se mencionar um outro tipo importante. Trata-se da *linguagem interrogativa*, cujo exemplo mais utilizado é a *Basic*. No capítulo 4 encontra-se uma descrição do uso de terminais situados a grande distância do computador. Estes terminais são normalmente meios de

```
10 LET A = 0
20 LET B = 0
30 FOR I = 1 TO 75
40 READ M
50 IF M > 100 THEN 140
60 LET A = A + M
65 PRINT A
70 IF M > 39 THEN 90
80 LET B = B + 1
90 NEXT I
100 LET C = A / 4
110 PRINT «NÚMERO DE FALTAS», B
120 PRINT «ÍNDICE MÉDIO», C
130 GOTO 160
140 PRINT «ÍNDICE ERRADO», M
145 GOTO 90
150 DATA «75 NÚMEROS»
160 END
```

Fig. 66 — Afirmações de programa em Basic.

ligação nos dois sentidos, permitindo de facto ao operador do terminal conduzir uma conversação com o computador. Por esta razão, este tipo de linguagem é por vezes designado por linguagem «conversacional». A linguagem *Basic*, concebida para utilização deste modo, tem uma construção bastante simples e adapta-se portanto idealmente para uso por pessoas sem um treino pormenorizado de programação. Um exemplo de afirmações em *Basic* é dado na figura 66.



## 8

### GRANDES E PEQUENOS SISTEMAS DE HARDWARE

O objectivo deste capítulo consiste em rever os diferentes tipos de computador digital actualmente usados e os seus diferentes modos de funcionamento.

Os primeiros computadores digitais, concebidos para processamento de dados em larga escala, pouco mais fizeram que encarregar-se de enormes quantidades de trabalho rotineiro que anteriormente era realizado manualmente. Este tipo de trabalho pode ser feito por um computador rapidamente e com grande rigor.

A maior parte dos primeiros sistemas comerciais recebiam registo de entrada durante um período de tempo — um dia, uma semana ou um mês — e em seguida processavam-nos em conjunto de uma só vez. Os registo de dados tendiam a ser obtidos de modos não aceitáveis pelas máquinas o que significava que se tornava primeiramente necessário convertê-los numa forma que a máquina pudesse ler, como seja em cartões perfurados ou fita de papel perfurada. Liam-se então grupos destes cartões para algum tipo de memória auxiliar, normalmente de tipo magnético. Isto era feito periodicamente à medida que esses grupos de cartões iam sendo obtidos e assim, durante um certo período de tempo, todos os registo de entrada — normalmente conhecidos por registo de movimento — eram concentrados para uma data de movimento, por exemplo, o final de cada mês. Esta

fita de movimento era então comparada com outras contendo dados de comparação, com informações sobre compras, vendas ou stock, resultando no produto final desejado — dados sobre vendas, listas de movimento dos stocks, etc.

De um ponto de vista do sistema, este processamento por «lotes» tinha uma desvantagem importante, na medida em que os registos nunca se encontravam exactamente em dia. A concentração de lotes de registos num dado período de tempo significa que o computador está sempre a tratar com acontecimentos que se verificaram no passado — dados históricos. Do lado das vantagens, estes sistemas de processamento por lotes tinham uma concepção bastante simples, um padrão de acontecimentos claramente definido sem a necessidade de um hardware ou software altamente sofisticados.

No entanto, apesar destes factores, o processamento por lotes é ainda usado, mesmo nas máquinas mais sofisticadas e mais rápidas actualmente existentes. Em muitos sistemas, não há necessidade de dispor de informação imediata absolutamente em dia, por exemplo, num sistema concebido para produzir contas de electricidade.

Do ponto de vista do hardware, as primeiras máquinas eram apenas capazes de realizar uma operação de cada vez, não só ficando restringidas a um único programa, como ainda a um único procedimento dentro desse programa. O grau de eficiência da utilização da máquina era portanto pobre. O processamento era efectuado num modo serial, ou seja, cada procedimento devia ser completado antes de se iniciar o seguinte. Recorde-se que o objecto do processador central consiste em processar dados. Numa situação em que deve ocupar-se primeiramente com a leitura e armazenamento de dados, e em seguida com o seu processamento, terminando por juntar e dar saída a um conjunto de dados para, por exemplo, uma impressora, a maior parte do seu tempo é ocupada nos processos de entrada e saída e só uma pequena parte no verdadeiro processamento. Isto deve-se, por um lado, a velocidades de transferência de entrada/saída muito

lentas e, por outro, à sua potência de processamento extremamente rápida.

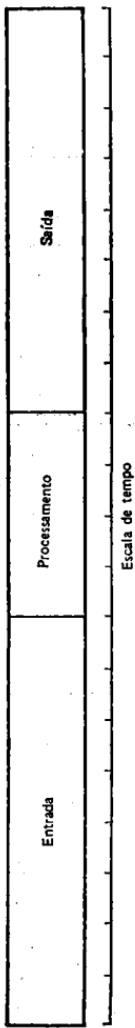
### *Simultaneidade*

Uma maneira de melhorar o rendimento do processador consiste em fornecer meios que lhe permitam realizar dois procedimentos simultaneamente, ou seja, realizar a sua função de processamento ao mesmo tempo que se estão a verificar processos de entrada ou saída de dados, o que pode ser feito utilizando técnicas «buffer». Isto significa que os dados de entrada são primeiramente colocados em áreas não endereçáveis de armazenamento e, do mesmo modo, os dados de saída são reunidos em áreas não endereçáveis. É então possível sobrepor funções de entrada, saída e processamento, aumentando a proporção de tempo gasto em processamento (figura 67). No entanto, isto representa apenas um pequeno aumento marginal do rendimento do processador.

### *Multi-programação*

É pois evidente que um factor limitativo importante do grau de rendimento a que o processador trabalha são as comparativamente lentas velocidades operacionais dos dispositivos de entrada e de saída. É um pouco como trabalhar numa linha de produção em que os objectos com que se trabalha chegam a intervalos de um minuto, mas em que só são necessários seis segundos para realizar o trabalho necessário nesse ponto. Nestas circunstâncias, pode-se argumentar que o mais inteligente a fazer será iniciar uma segunda linha de produção de tal modo que se possam alternar no fornecimento do material, em vez de se ter de esperar 54 segundos em cada minuto. Isto teria, pelo menos, o efeito de duplicar o rendimento de 10 % para 20 %. Pode-se ainda dizer que, se montarmos dez linhas de produção, o rendimento pode ser maxi-

(a) Processamento serial



(b) Processamento por buffer

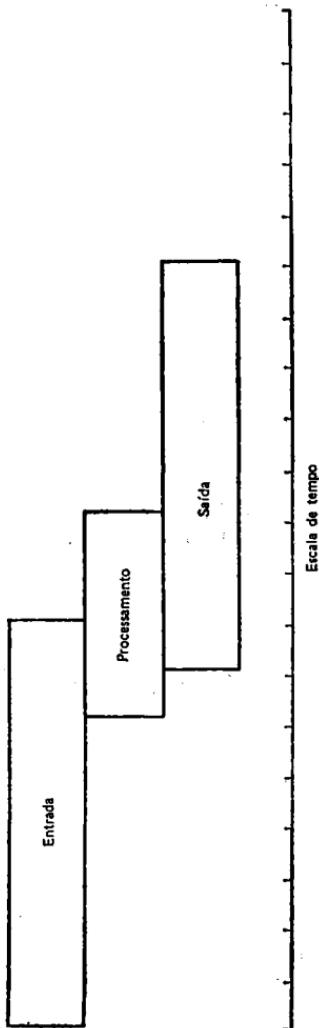


Fig. 67 — Simultaneidade. (a) Processamento serial, (b) Processamento buffer.

zado para 100 %. Isto teoricamente, pois na prática raramente é possível.

O mesmo acontece com o processador central. Se enquanto está a trabalhar um programa, existirem longos intervalos à espera de dados, porque não lhe dar outro trabalho durante estes intervalos? Isto pode ser feito armazenando no processador dois ou mais programas simultaneamente, de tal modo que, no caso de não poder trabalhar num deles devido à lentidão da actividade periférica, comute para outro onde possa realizar o processamento. Este tipo de funcionamento é conhecido pelo nome de *multi-programação*.

A multi-programação pode no entanto ser apenas efectuada numa máquina relativamente potente que tenha sido especificamente concebida para funcionar deste modo. A máquina incorporará um sistema operatório ou de controlo, capaz de decidir qual o programa que deve ser processado. Outro requisito é que exista um número suficiente de periféricos — entrada, saída e armazenamento — para apoiar a realização desses processamentos múltiplos.

Como exemplo simples, consideremos dois programas, A e B, armazenados no processador. O programa A pede uma leitura de cartões perfurados, mantendo o seu conteúdo na memória até terem sido lidos dez cartões e passando depois tudo para fita magnética. Consideremos que necessitará de 100 milissegundos para ler um cartão mas apenas de 5 milissegundos para juntar o seu conteúdo na memória, pronto para ser utilizado. O programa B obriga a um cálculo demorado, absorvendo 200 milissegundos de tempo de processamento. Trabalhando com o programa A, o processador, depois de ler o primeiro cartão, ocupar-se-á durante 5 milissegundos do tratamento dos seus dados, o que significa que terá de esperar 95 milissegundos até terminar a leitura do cartão seguinte. No entanto, no final do seu processamento de 5 milissegundos, terá enviado uma mensagem ao sistema operatório pedindo dados do segundo cartão.

O sistema operatório comunicará então com o leitor de cartões, verificando que este está ocupado a ler um

cartão. Neste ponto, parará as operações no programa A e procurará qualquer outra coisa para manter o processador ocupado. Esta operação é conhecida pelo nome de *suspensão do programa*. Descobre-se então o programa B à espera de tempo disponível e o sistema comuta o controlo para ele, dando assim ao processador alguma coisa que fazer. Depois de passarem 5 milissegundos, o leitor de cartões terá terminado a leitura do segundo cartão, enviando uma mensagem nesse sentido ao sistema operatório. Poderá haver dois programas pedindo tempo de processamento. O sistema terá então de decidir qual deles vai ser trabalhado. Esta decisão é realizada de acordo com as prioridades que lhe foram previamente comunicadas no início das operações. Se considerarmos que A tem prioridade relativamente a B, então o controlo é novamente comutado para A neste ponto, interrompendo o trabalho que se está a fazer em B mesmo que este não tenha sido completado. Chama-se a isto uma *interrupção do programa*. O controlo pode ser novamente comutado para B quando houver mais algum tempo de espera no processamento de A.

Este exemplo baseia-se apenas em dois programas — uma situação normalmente designada por *programação dupla*. Existe uma multi-programação quando diversos programas estão a ser realizados simultaneamente, tendo cada um deles uma determinada prioridade. Em teoria, a multi-programação poderia permitir um rendimento de 100 % do processador central. Na prática, a sequência de actividade de processamento entre os programas não é provavelmente contínua e portanto a utilização é de facto inferior à máxima. No entanto, este modo de funcionamento permite um aumento bastante notório de rendimento do processador.

Até agora, temos pensado num computador como uma máquina, colocado no centro de uma grande actividade, com dados referentes a esta actividade que lhe são conduzidos para efeitos de processamento. Esta é de facto uma situação bastante conveniente, quando todos os dados

são originados numa área concentrada perto do computador.

No entanto, em muitos casos, esta situação não se verifica. Os escritórios, oficinas, etc., podem estar espalhados por uma área bastante vasta, talvez em todo o território de um país. Isto deixa-nos duas alternativas. Ou montar um certo número de pequenos computadores nestes diferentes pontos e processar os dados localmente, ou instalar um sistema de processamento central que satisfaça todas as necessidades de processamento da organização.

Se bem que a segunda alternativa tenha a desvantagem inerente de ter de movimentar dados a longa distância, beneficia em muitos casos da importante vantagem da economia em hardware, de constituir uma fonte centralizada de informações e de dar um maior controlo de toda a organização. De facto, nos últimos anos, tendeu-se para o estabelecimento de computadores centrais, grandes e potentes, capazes de tratarem dados originados numa vasta área. Sabemos agora evidentemente que algumas destas máquinas potentes trabalham com dados enviados ao centro por correio a partir de todo o país, por exemplo, um computador centralizado de atribuição de licenças de veículos.

No sector comercial, muitos computadores trabalham com dados que lhes são transmitidos por terminais remotos, situados em filiais e escritórios locais. O sistema usado nos bancos é um exemplo disto. Este sistema tem a vantagem de não só permitir a transmissão muito rápida dos dados originados numa filial ao computador, como ainda permitir à filial o acesso à informação contida no computador central.

No capítulo 4, deu-se uma panorâmica da transmissão de dados e uma descrição dos terminais usados para tal. A sua utilização introduz outro princípio no uso do computador, conhecido por *acesso múltiplo*. Isto significa que um certo número de pontos de entrada e saída longe do computador têm um acesso à máquina directo e, para todos os efeitos práticos, simultâneo.

### *Partilha do tempo*

A essência de um modo operacional de partilha do tempo consiste em fornecer as condições em que um dado número de utilizadores tem acesso a um computador — uma situação de acesso múltiplo. Os utilizadores comunicam com o computador através de teletipo ou de um dispositivo terminal de apresentação visual de dados. Para ser eficaz, o tempo de resposta, ou seja, o tempo necessário para que o computador comece a devolver informações, deve ser curto. E de facto inferior, normalmente, a 5 segundos.

Um sistema de partilha de tempo fornece uma situação em que um certo número de utilizadores têm acesso ao computador virtualmente ao mesmo tempo. Os utilizadores têm terminais remotos ligados por linhas de transmissão ao computador. Cada terminal tem o seu pequeno registo de dados, conhecido pelo nome de buffer, no qual são mantidos os dados enquanto se estão a realizar as operações de entrada e saída. A razão disto é que, se os dados fossem transmitidos carácter por carácter à medida que entrassem através do teclado do terminal, este processo seria demasiado lento para o computador ligado na outra extremidade. Quando a entrada por teclas está terminada, o conteúdo do buffer pode ser transferido para o centro a alta velocidade. Do mesmo modo, a saída do computador é também transferida para o buffer terminal, onde é o próprio terminal que controla a saída da sua impressora serial. Enquanto este processo de entrada/saída se realiza num dado terminal, o processador pode continuar o seu trabalho, submetido a outros terminais.

Estando um grande número de utilizadores ligados a uma máquina, todos eles pedindo tempo de processamento, é necessário existir um número correspondente de programas para satisfazer todas estas necessidades. Um processador pode trabalhar apenas num programa de cada vez, tornando-se portanto necessário armazenar todos os programas dos utilizadores em algum tipo de dispositivo de acesso directo do qual possam ser transfe-

ridos para o processador à medida das necessidades. Na prática, o que acontece é que é atribuído a cada terminal de utilização um período de tempo durante o qual o seu programa é apresentado ao processador e obtém, de facto, um uso exclusivo do processador durante esse período. Este período de tempo é bastante pequeno, provavelmente de apenas cerca de 10 milissegundos. Tendo

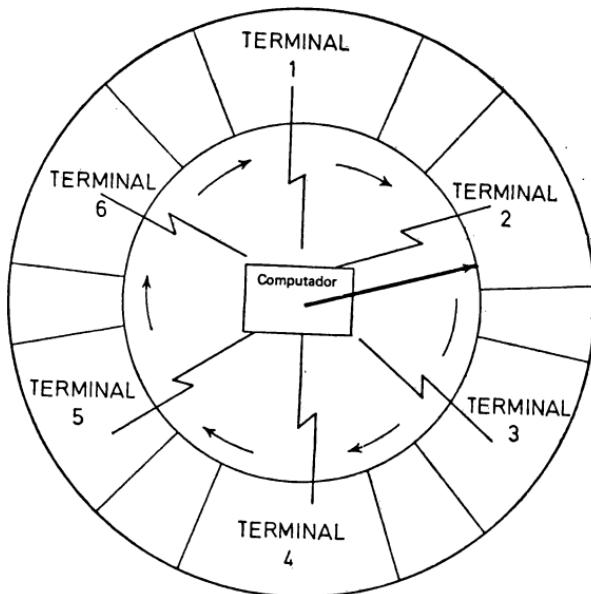


Fig. 68 — Sistema de partilha do tempo.

em conta que o processador é provavelmente capaz de completar mais de um milhão de cálculos por segundo, é evidentemente possível realizar um grande trabalho mesmo naquele reduzido período de tempo.

O sistema operatório numa situação de partilha de tempo fará circular todos os terminais, atribuindo uma «fata» de tempo a cada um deles rotativamente (figura 68). Num segundo, um sistema moderno de partilha de tempo

pode atender muitos terminais. Se bem que o processador possa trabalhar apenas num programa de cada vez, em multi-programação não há razões para que não estejam guardados em memória vários programas simultaneamente. Este princípio é utilizado em partilha de tempo. Por exemplo, consideremos que estão armazenados os programas dos utilizadores A e B simultaneamente. Quando termina a fracção de tempo atribuída a A, o controlo é imediatamente comutado para o programa B. Enquanto B está a ser processado, A é retirado da memória do processador para um disco e C é transferido do disco ocupando o seu lugar. Quando termina o tempo de B, passa a ser processado C, B é movimentado para fora da memória e D acede a esta. Este processo continua até todos os utilizadores terem tido uma fracção de tempo de processamento e, em seguida, o ciclo inicia-se novamente.

### *Sistemas de tempo real*

A característica mais importante dos sistemas de tempo real é que devem ser capazes de aceitar e processar dados no momento da sua ocorrência e realimentar os resultados a tempo de alterar a ocorrência em causa. No entanto, o tempo de resposta, ou seja, o atraso de tempo entre os dados originais e a realimentação dos resultados, variará conforme a aplicação.

Por exemplo, para afectar o comportamento de um míssil em voo, o tempo de resposta deve ser da ordem dos milissegundos. Para obter informações do estado da conta de um cliente, resultante da apresentação de um cheque, já será possível tolerar tempos de realimentação maiores.

Outra característica de um sistema de tempo real é que os dados existentes em memória são imediatamente actualizados pelos novos dados de entrada, à medida que são produzidos. Um exemplo clássico de um sistema de tempo real é o das reservas de lugares em aviões. Neste, o pedido de um lugar é apresentado num terminal e pro-

vocará a apresentação dos lugares ainda existentes no voo. O lugar agora reservado deve ser imediatamente marcado, ou seja, os dados existentes em memória devem ser actualizados, senão será fornecida uma informação incorrecta ao próximo cliente. Os sistemas de tempo

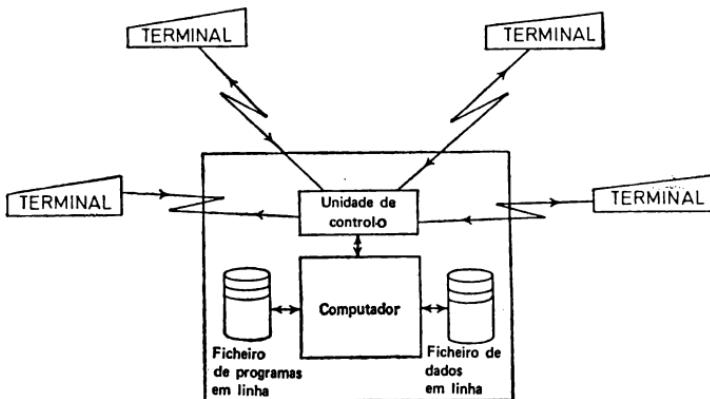


Fig. 69 — Funcionamento em tempo real.

real (figura 69) trabalham através de terminais com acesso directo e imediato ao computador e, através deles, a registos de dados e de programas. Isto significa um requisito de hardware em sistemas comerciais, em termos de memórias auxiliares de acesso directo, em larga escala.

### *Computadores pequenos*

Nos últimos anos, observou-se uma revolução na maquinaria de escritório utilizada para cálculos e actualização de contas. Há dez anos, a maior parte destes dispositivos eram puramente mecânicos ou electromecânicos. As máquinas de calcular mecânicas eram bastante comuns e a maior parte das máquinas de actualização de contas fundamentavam-se no uso de barras de funcionamento mecâ-

nico, para realizar o reduzido processamento de que eram capazes. Tudo isto já foi ultrapassado e a electrónica ocupou, também neste campo, o lugar da mecânica. Já existem inclusivamente máquinas de escrever automáticas de armazenamento em fita magnética e controladas eletronicamente.

Com o desenvolvimento de circuitos integrados de reduzidas dimensões, foram introduzidas máquinas de cálculo electrónicas com uma gama fixa de funções aritméticas. Foi-lhes mais tarde acrescentada uma memória de fraca capacidade, e em seguida sucederam-se máquinas programáveis capazes de armazenar e executar programas apresentados através de um teclado ou de um meio de armazenamento magnético. As calculadoras electrónicas tornaram-se entretanto bastante populares, tendo chegado a substituir completamente os velhos dispositivos mecânicos. São baratas, têm um funcionamento seguro, e são fáceis de utilizar e muito rápidas.

Simultaneamente, a electrónica foi aplicada a máquinas de contabilidade. As registadoras mecânicas de cálculo rotativo foram substituídas por circuitos somadores electrónicos e as barras mecânicas pelo programa armazenado electronicamente. Foram acrescentadas possibilidades de armazenamento de dados, tendo estas máquinas recebido o nome de «computadores de secretaria» ou «de escritório», se bem que haja fortes dúvidas de que muitas destas máquinas pudessem ser de facto designadas por computadores no sentido lato da palavra; no entanto, o nome pelo qual são conhecidas as máquinas de contabilidade electrónicas modernas é o de *computador de registo visível* (ver a figura 70).

#### *Computadores de registo visível*

Uma característica importante de um sistema de computador, considerada por muitas pessoas como uma desvantagem, é que os registos relacionados com os sistemas que processa são mantidos em registos magnéticos ou

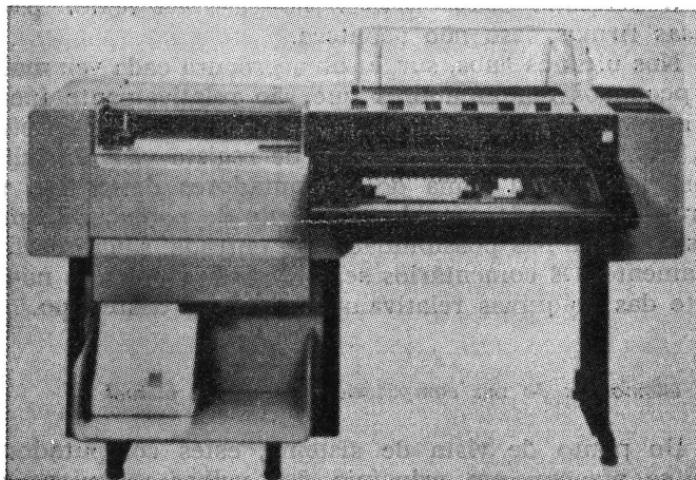


Fig. 70 — Um computador de registo visível (Kienzle Data Systems).

electrónicos, evidentemente impossíveis de ler directamente. Para receber os registos de uma maneira lisible, é necessário pedir ao computador que os imprima ou apresente ao exterior de outro modo. Isto contrasta com as antigas máquinas de contabilidade, nas quais era fornecido um papel impresso ou cartão com um registo da situação de cada cliente ou produto. Este cartão fornecia não só balanços actualizados, como ainda continha a história das transacções realizadas durante um longo período de tempo. Constituía um meio muito conveniente e simples de dispor de um registo fácil de consultar. Muitas firmas têm uma certa relutância em dispensar esta facilidade.

Outro factor, particularmente para as empresas de dimensões médias ou pequenas, é o capital necessário e os custos de manutenção para a aquisição de um computador verdadeiro. Este valor é justificado no caso de o volume de trabalho a processar ser suficiente para man-

ter a máquina ocupada mas, no caso da maior parte dessas firmas, isso não acontece.

Nos últimos anos, surgiu uma procura cada vez maior de pequenos computadores que são relativamente fáceis de instalar e de utilizar, e fornecendo um registo visível das transacções — o computador de registo visível. Existe agora uma vasta gama de computadores deste tipo no mercado, variando consideravelmente de potência de processamento e nas possibilidades de entrada, saída e armazenamento. Os comentários seguintes são comuns à maior parte das máquinas relativamente baratas deste tipo.

*Elementos de um computador de registo visível*

Do ponto de vista de sistema, estes computadores não se afastam em princípio das velhas máquinas de contabilidade, sendo mantido um ficheiro de cartões e tendo cada um deles um registo impresso de transacções passadas. A medida que eram realizadas novas transacções, depois de serem primeiramente registadas na documentação — requisições de stock, etc. — eram apresentadas através de um teclado, actualizando o cartão ou ficha e calculando automaticamente os balanços actuais. De um ponto de vista de hardware, a alteração mais importante consistiu no fornecimento de um meio para entrada automática de dados estáticos referidos a cada ficha. Isto é realizado armazenando os dados numa fita magnética que corre ao longo do comprimento da ficha.

Por exemplo, num sistema de salários, seria armazenaada a informação seguinte:

*Identificação do empregado:* número, nome, departamento.

*Detalhes da remuneração:* salário ou horas normais, salário em horas extraordinárias.

*Deduções no salário:* previdência, etc.

*Pormenores de impostos:* imposto a pagar, código do imposto.

Ao inserir a ficha na máquina, estes dados são automaticamente inscritos na memória, constituindo a base do cálculo do salário líquido do mês anterior.

A entrada primária é executada através de um teclado accionado manualmente. Este contém normalmente três secções. Primeiramente, uma secção alfa/numérica para entrada de dados descritivos (nomes, endereços, descrição de produtos, etc.). A identificação dos dados, por exemplo, o número da conta, é verificada por entrada de teclado comparando com a versão armazenada na barra magnética, sendo a ficha rejeitada se estes dados não coincidirem. A segunda parte do teclado, numa secção numérica de dez dígitos, é usada para a entrada de dados quantitativos, e a terceira secção contém as teclas que controlam as funções da máquina. Muitas máquinas incorporaram uma linha visual alfa/numérica de apresentação de dados acima do teclado, na qual são mostrados os dados à medida que entram na máquina, permitindo realizar uma verificação visual. Serve igualmente para controlar o programa, mostrando a fase por ele atingida. Existe uma entreface em muitas máquinas de registo visível para entrada automática através de cartões perfurados ou de fita de papel perfurada.

Nos modelos básicos a saída é realizada de duas formas:

1. Como registo impresso nas fichas.

2. Os balanços podem ser registados na barra magnética da ficha. Isto fornece automaticamente uma avaliação do balanço quando a ficha é usada de novo. Os dados quantitativos são igualmente registados em memória, de modo a fornecer totais de controlo e permitir as análises necessárias no final.

Se bem que numa escala menor, o processador central executa funções bastante semelhantes às do processador central de um computador de maiores dimensões. É típica a utilização de uma memória de acesso imediato

de 8 K palavras de 16 bits, capaz de guardar um programa e igualmente dados necessários, enquanto estes são processados. Incorpora uma unidade aritmética que realiza o processamento numérico normalizado, e um sistema de controlo que vigia a execução do programa e das operações de hardware.

Os programas são construídos num código mnemónico bastante simples. Podem ser lidos para a memória ou para um meio de armazenamento magnético, através do teclado da máquina, se bem que os fabricantes forneçam normalmente programas como parte do software de origem da máquina. Uma maneira bastante generalizada de guardar estes programas é em fita magnética, mantida em cassettes em circuito fechado. Esta disposição da fita magnética evita a necessidade de a rebobinar no sentido inverso. Ao inserir a cassette na máquina e ao premir a tecla da função desejada, o programa é passado à memória para execução.

Entre as vantagens atribuídas aos computadores de registo visível, encontram-se as seguintes:

1. São fáceis de instalar num escritório vulgar. Isto significa que não é necessário proceder a grandes alterações na organização do departamento. A máquina torna-se parte do departamento em vez de ser este que se transforma numa parte do sistema da máquina.
2. No caso de muitas aplicações comerciais, um computador de registo visível realizará o seu trabalho de uma maneira tão adequada e eficaz como um computador maior, se bem que a uma menor velocidade.
3. O funcionamento da máquina é relativamente simples e, até certo ponto, não obriga a utilizar um operador especializado. Um operador de máquinas tradicionais de contabilidade não deve ter qualquer dificuldade em utilizá-la, depois de um curto período de treino.

4. Muitos computadores, a fim de serem rendosos, devem ser utilizados numa base de 24 horas por dia. Os computadores de registo visível podem trabalhar entre as 9 e as 17 horas, sendo provavelmente mais barato, a longo prazo, instalar uma segunda máquina que utilizar a anterior no dobro das horas.

5. Não são necessários sistemas especializados nem pessoal de programação, não existindo normalmente, nestas instalações, um director de serviço e todo o resto do pessoal que se encontra necessariamente em instalações de computadores de maiores dimensões. Falando em geral, tanto os sistemas de computadores de registo visível como os programas necessários são fornecidos pelo próprio fabricante.

6. O computador de registo visível moderno tende a constituir uma peça de equipamento de grande segurança de funcionamento. Os custos de manutenção são bastante baixos, e as perdas de tempo devidas aos períodos de pouco trabalho tendem a ser pequenas. Excluindo o salário do operador, o custo total de manutenção de um computador de registo visível pode ser bastante baixo.

### *Minicomputadores e microprocessadores*

Não ficaria completa a nossa apresentação dos dispositivos existentes, sem uma breve menção às duas áreas de tecnologia de computadores actualmente em mais rápido desenvolvimento — os minicomputadores e os microprocessadores.

#### *Minicomputadores*

Trata-se de computadores com um processador central de dimensões bastante reduzidas, tornadas possíveis pela utilização de circuitos integrados sob a forma de pastilhas

semicondutoras. Em geral, estas máquinas são capazes de controlar uma gama de entradas, saídas e periféricos de armazenamento de dados de maneira bastante semelhante à de um computador «convencional». A sua principal desvantagem consiste nas suas velocidades operacionais comparativamente lentas, devido à limitada capacidade de armazenamento de palavras neste tipo de máquina.

#### *Microprocessadores*

Estes representam provavelmente o desenvolvimento mais significativo em dispositivos automáticos dos últimos anos, cujo potencial, particularmente na indústria, começa apenas a ser sentido. Não são computadores no sentido que temos estado a discutir até agora, constituindo antes dispositivos de controlo, facilmente programáveis, e sendo parte integrante do dispositivo que controlam. Isto contrasta com um computador que controla actividades exteriores a si próprio e até, possivelmente, remotas. O microprocessamento foi tornado possível devido ao desenvolvimento de pastilhas semicondutoras «micro» que podem ser programadas para iniciar e controlar uma gama de funções previamente determinada e continuar a repetir a rotina até receberem uma instrução para parar.

Um exemplo comum do uso de um microprocessador é nas consolas de jogos de televisão, onde é utilizado para simular e controlar o movimento de uma bola num visor de televisão.

No entanto, o seu maior potencial reside no campo dos processos industriais. Nos primeiros tempos da automação de máquinas controladas por computadores, o comportamento das máquinas era governado por um programa armazenado num computador situado longe da máquina. Um microprocessador, no entanto, constitui parte da própria máquina, o que significa que temos uma espécie de «robot» tal que a máquina ferramenta

executará um certo número de actividades automaticamente.

Uma implicação grave do que acaba de ser dito é o deslocamento dos operadores que, de outro modo, deveriam accionar a máquina manualmente. Na medida em que os micro-processadores são agora bastante baratos e uma máquina «robot» por eles controlada pode muitas vezes realizar o trabalho que anteriormente requeria um certo número de operadores, o seu uso generalizado parece surgir como economicamente sólido mas o valor social a pagar deverá ser medido em termos de uma procura cada vez menor de emprego manual.



## O COMPUTADOR E A SOCIEDADE

Não é objectivo deste capítulo fornecer uma explcação técnica detalhada do funcionamento dos sistemas de computador que influenciam a nossa vida de todos os dias, mas sim dar uma curta descrição daquilo que podem fazer.

### *Aplicações comerciais*

A maioria dos computadores actualmente usados são-no muito particularmente em firmas comerciais e industriais, para processar a quantidade de dados utilizados na actividade comercial. Seria de facto razoável afirmar que mais de 80 % dos computadores usados o são nesta área. Entre os sistemas que vulgarmente utilizam o processamento por computadores, encontram-se os seguintes.

### *Listas de pagamentos*

Este foi talvez o primeiro dos sistemas comerciais a ser computarizado — de facto, uma das primeiras máquinas LEO (Lyon's Electronic Office) era usada para este fim. A folha de pagamento que recebemos juntamente com o nosso salário no final de cada semana ou mês foi, muito provavelmente, produzida por um computador.

Uma lista de pagamentos, devido ao grande número de cálculos simples envolvidos e o grande número de variáveis que contém relacionadas com o valor dos salários, deduções, etc., é o tipo de sistema que pode utilizar idealmente as qualidades oferecidas por um computador.

Existem dois elementos básicos num sistema de listas de pagamentos. Primeiramente, os dados «standard» ou «estáticos» relacionados com cada empregado. Estes incluirão detalhes pessoais e de identificação, uma indicação do pagamento bruto, detalhes das contribuições, etc., assim como balanços de pagamento em bruto, total de impostos pagos e de pagamentos líquidos actualizados. Em segundo lugar, haverá dados correntes relacionados com o período para o qual o pagamento é calculado. Estes poderão incluir o número de horas de trabalho, subsídios de horas extraordinárias, bónus especiais, etc. O primeiro conjunto de dados será mantido na memória auxiliar do computador para uso cíclico, dado que as listas de pagamento são preparadas periodicamente. O segundo é preparado semanalmente ou mensalmente e transforma-se na entrada do processamento da lista. A medida que são processados dados relacionados com cada empregado, é feita referência aos dados de comparação existentes em memória, são realizados os cálculos necessários e feitas as deduções, obtendo-se uma indicação do pagamento líquido que o empregado receberá. Entretanto, os dados contidos na ficha de comparação serão actualizados, dando valores revistos do pagamento bruto actualizado, etc. Os impostos e as outras deduções terão sido acumulados em registos, de tal modo que, no final da lista de pagamentos, existirá um total de controlo para cada um destes elementos. Durante o seu funcionamento, o computador imprimirá as listas de pagamento e, possivelmente, cheques e envelopes para conter os quantitativos apropriados, assim como um sumário do pagamento total dando custos em bruto, deduções totais e pagamento líquido total. Quando o pagamento é realizado em dinheiro, fornecerá também uma lista mostrando as quantidades de notas de dife-

rente valor facial e de moedas requeridas para fazer os pagamentos.

#### *Controlo de stocks*

Este outro sistema presta-se também muito facilmente à utilização de métodos computarizados, fornecendo não só uma função de apreçamento dos produtos que pode por sua vez ser usada para análise de custos como, ainda mais importante, manter os níveis de stocks nos seus valores óptimos do ponto de vista económico. Stocks excessivos conduzem a perdas por deterioração, despesas excessivas de armazenamento e capital parado que pode ser utilizado com vantagem noutras sectores. Se os stocks forem demasiado reduzidos, há a possibilidade de perder vendas, devido à ausência de material em stock.

Uma ficha de controlo de stocks conterá:

A identificação e descrição de cada produto em stock.  
Níveis normais de stock.

Preços do produto.

Uma definição dos níveis mínimo e máximo de stock.

Uma ordem de preenchimento de stocks, baseada no tempo necessário para substituição dos mesmos.

Os dados de movimento consistirão essencialmente em:

Identificação de cada produto em stock.

Quantidade de cada produto que entra ou sai de armazém.

O processamento dos dados de movimento, comparado com a ficha de stock, produzirá relatórios de saídas, que incluirão:

Identificação, descrição, quantidade, preço unitário e valor total do movimento de cada produto em armazém — informação necessária para facturação de vendas.

Uma lista de todos os produtos em stock que estejam no nível mínimo.

Uma lista dos produtos que estão a atingir os níveis mínimos de armazenamento, a fim de ser lançada uma ordem de preenchimento dos níveis normais de stock.

Totais de controlo, mostrando os valores totais de entrada e saída dos produtos num dado período.

Quando desejável, será também produzida uma lista completa de todos os produtos em stock, suas quantidades e valores, para avaliação dos stocks.

E também uma lista de produtos com pequena movimentação, que permitirá tomar certas medidas, por exemplo, sob a forma de redução de preços (saldos), a fim de os eliminar.

#### *Actividade bancária*

É este talvez o exemplo clássico da utilização comercial de computadores. O volume de papel tratado diariamente pelos bancos — cheques, ordens de pagamento, etc. — é vasto. Os registo de todos os bancos actuais estão agora computarizados, dado que é esta a única maneira prática de copiar a enorme massa de papel, a tempo de corresponder às transacções comerciais por ela representadas. Os milhões de cheques tratados pelo banco até à hora de fecho em qualquer dia devem ser lançados nas contas dos clientes e nos balanços diários que devem ser usados pelas direcções e administrações em cada dia.

As contas dos clientes são mantidas nos computadores centrais, num sistema de armazenamento de acesso directo, normalmente utilizando discos. A informação relativa a cada cliente incluirá a identificação do número da conta, o nome e a morada, detalhes das transacções anteriores desde o último extracto de conta, informação sobre balanços e pormenores sobre pagamentos à ordem, etc.

Cada filial do banco receberá durante o dia cheques lançados em muitos outros bancos, de tal modo que a primeira tarefa consiste em obter os dados necessários. As dívidas entre bancos devem ser calculadas, o que é realizado através dos computadores centrais de cada um deles. Em seguida, os cheques devem passar pelo centro computador do banco, onde são distribuídos pelas filiais e usados para actualizar a ficha do cliente.

Os relatórios de saída incluirão uma lista dos balanços de todos os clientes das filiais que deve ser apresentada à direcção da filial na manhã seguinte e, rotativamente, os extractos de conta a serem enviados aos clientes.

Um desenvolvimento mais recente na automação bancária é o uso de terminais interrogativos, nos quais é possível obter imediatamente todos os detalhes pretendidos sobre a conta de cada cliente.

É interessante notar que os sistemas bancários constituem um bom exemplo do uso de documentação primária, imediatamente utilizável pela máquina. É usado um sistema m.i.c.r. com caracteres impressos na parte inferior do cheque. Isto evita a necessidade de preparar grandes quantidades de dados, constituindo um documento que pode ser lido directamente pelo computador.

Um outro serviço bancário dependente dos computadores é o sistema de cartão de crédito. Seria completamente impraticável utilizar este sistema sem computadores que processem a massa de vendas realizadas em muitos milhares de pontos de venda em todo um país.

Outro aspecto essencial do sistema é o acesso imediato por telefone às contas dos clientes para efeitos de concessão de crédito. Isto só pode ser realizado através de memórias de acesso directo de grande volume.

#### *Registo de vendas*

Nos últimos anos, a influência do computador tem sido cada vez mais sentida, mesmo ao nível do comércio

local. Isto deu origem a técnicas diferentes das usadas em grandes instalações de computadores, concebidas para tratarem o movimento de um grande volume de produtos de baixo preço unitário. Os desenvolvimentos iniciais utilizavam o princípio de mini-cartões perfurados. O cartão continha, sob forma perfurada, dados que identificavam o produto e outras informações necessárias para controlo de vendas e de stocks. Quando cada artigo é vendido, o cartão é destacado do artigo em causa e apresentado a um leitor de cartões, sendo os dados transferidos para uma fita que pode ser apresentada na entrada de um computador.

Um desenvolvimento mais recente foi a computarização de notas de caixa. Estas, que constituem a maneira tradicional de registar as vendas a dinheiro, incorporaram sempre um certo grau de análise, realizado pela pressão de teclas, sendo os totais impressos em notas gerais no final do dia. O registo de vendas em forma lisível pelo computador é agora possível através do teclado de entrada que passa os dados para cassettes de fita magnética. Este processo é ainda, no entanto, um processo laborioso, dado que a identificação do processo e o preço devem continuar a ser comunicados por teclas, com um elevado risco de erro.

O desenvolvimento mais recente em registos de vendas consiste em eliminar o teclado das notas de caixa e fornecer um meio de comunicar os dados sobre vendas de modo completamente automático. Os métodos variam, mas o princípio básico consiste ainda em imprimir dados de maneira codificada em cada artigo, normalmente um grupo de linhas ou barras de espessuras diferentes que representam uma codificação binária. Esta é então escrutinada opticamente ou magneticamente num dispositivo que produzirá uma série de impulsos correspondentes ao padrão codificado e os transmitirão directamente ao computador. Os procedimentos de validação no interior do dispositivo de leitura assinalarão qualquer incorrecção de leitura, de modo que seja possível corrigi-la. O dispositivo de leitura é um pequeno instrumento, um pouco

como uma caneta, ligada por fio eléctrico e que se move sobre o código impresso. O sistema obriga evidentemente a que cada produto individual contenha a sua própria codificação impressa. Esta pode, em muitos casos, ser incluída na impressão da etiqueta, o que por sua vez permite poupar uma grande parte do trabalho necessário para atribuir preços a cada produto, como acontece correntemente em muitos postos de venda a retalho.

#### *Serviços de gás e electricidade*

Basta lançar uma vista de olhos à conta de gás ou electricidade, para verificar a grande dependência destas indústrias em relação aos computadores. Foram também desenvolvidas técnicas especializadas de processamento para corresponder aos requisitos necessários neste campo. O uso generalizado do princípio de entrada de dados o.c.r. e das técnicas de leitura de marcas podem facilitar as leituras de contadores.

Como em muitas outras condições comerciais, o sistema fundamenta-se em fichas de computador que contêm dados de comparação relacionados com cada conta individual e movimentos de dados produzidos na fonte, ou seja, o contador do cliente. A ficha de comparação contém um registo da leitura anterior e, ao entrarem novos dados, estes são comparados à leitura em causa, sendo determinado o consumo no tempo decorrido e definidos os preços a pagar. São então acrescentadas quaisquer deduções ou impostos, sendo a conta remetida ao consumidor.

Uma característica única destes sistemas é que a nota de pagamento junta à conta de gás ou electricidade constitui um documento que deve ser devolvido. Contém caracteres lisíveis pela máquina, de tal modo que o destacável que deve ser devolvido pode ser alimentado directamente aos dispositivos de leitura para comparação com a ficha do cliente disponível no computador. Daí a instrução, por vezes existente na factura, no sentido

de «não dobrar», dado que o leitor em causa não aceitará documentos mutilados ou vincados.

### *Polícia*

Os computadores provaram constituir um meio ideal para armazenamento de dados relativos a prisões, actuações típicas de criminosos, impressões digitais, descrições pessoais, etc. O seu valor não consiste apenas na grande velocidade de registo, mas ainda na sua capacidade para realizar comparações e escolher registos partindo de critérios conhecidos. Por exemplo, uma descrição geral comunicada à máquina produzirá uma lista de pessoas cujos registos existem em memória e que correspondem à descrição apresentada.

### *Impostos locais*

Este constitui também um dos primeiros usos dos computadores na área das autoridades públicas. É necessário manter em ficheiros uma grande quantidade de informações relacionadas com propriedades individuais, as quais constituem a base do cálculo de valores rateáveis. Num sistema de computador, estes dados são mantidos em fichas de comparação, normalmente do tipo magnético, e incluirão a identificação da propriedade, identidade do ocupante, base do rateamento, identificação do dono, no caso de este não ser o próprio ocupante, codificação que indica o tipo de propriedade e uma conta individual da pessoa responsável pelos pagamentos de impostos.

Isto dá todas as informações necessárias para calcular as taxas devidas, depois de ter sido fixado o imposto unitário em causa. O computador imprime então as taxas devidas que são seguidamente enviadas ao ocupante.

## *Aplicações industriais*

As aplicações que discutimos até agora foram essencialmente sistemas de processamento de dados, principalmente relacionados com questões financeiras. No entanto, o computador é também um instrumento muito válido e prestável nos próprios processos industriais.

### *Planificação de produção*

A maximização dos rendimentos de produção fundamenta-se no uso óptimo dos recursos produtivos existentes. Para a obter, devem ser tidos em conta os seguintes factores:

Existência de suficientes quantidades de matérias primas e peças sobresselentes, a fim de evitar paragens de produção devido à escassez desses produtos.

Existência de materiais e peças do tipo necessário na altura correcta e no local certo na cadeia de produção.

Definição das possibilidades da máquina, de modo a obter um andamento regular da produção e evitando que, numa zona de trabalho, este se acumule sem ser despatchado, enquanto na fase seguinte as máquinas esperam por trabalho.

Distribuição eficaz de recursos humanos.

Para reunir todos estes factores de modo a obter o plano mais eficaz de produção, o computador necessitará dos seguintes tipos de informação:

1. *Características dos materiais*, não só em termos de descrição e quantidades de matérias primas e peças requeridas para produzir um determinado produto, como ainda a fase da produção em que estes devem estar disponíveis.

2. *Características de instalação*, identificando os tipos, capacidades, velocidades, etc., das máquinas existentes e outro equipamento, assim como os seus requisitos em termos de, por exemplo, tempo de manutenção, ferramentas e operadores.

3. *Características de funcionamento*, dando detalhes quanto a processos de fabrico, tempo de operação, etc.

Com toda esta informação, o computador será capaz de produzir «modelos» para diferentes níveis de produção, mostrando os estrangulamentos que se podem verificar ou as potencialidades que podem vir a ser subutilizadas. Podem então ser realizados ajustamentos nos tempos de funcionamento, aumentando ou diminuindo potencialidades ou ainda modificando processos, com vista a conseguir as melhores condições de trabalho, a melhor acumulação de potencialidades possível, e o melhor fluxo de materiais e peças de modo a obter o maior nível prático de rendimentos.

#### *Controlo de processos*

Como o nome implica, este controlo envolve a utilização de computadores para fiscalizar a produção efectiva ou processos industriais. Trata-se de uma área que se presta a um processamento de computador em tempo real, como numa situação de produção industrial contínua, sendo os dados relativos ao processo determinados pelo computador e as suas conclusões realimentadas de modo a modificar, se necessário, factores que controlam o fluxo de produção.

Um exemplo disto é a produção de papel, na qual é necessário um controlo estrito da espessura, à medida que vai sendo produzido. Um dispositivo de controlo on-line medirá constantemente a espessura e transmitirá esta informação ao computador. No caso de a espessura sair fora de determinadas tolerâncias, o computador en-

viará imediatamente uma mensagem à máquina, iniciando a acção correctora necessária.

Outro exemplo ainda é o controlo da produção de electricidade. Neste caso, não se trata apenas de um processo de controlo, mas sim de um sistema que determinará previamente as necessidades prováveis de electricidade. O computador terá de registar na sua memória os dados importantes em relação a todos os factores que influenciam esta produção — condições de tempo, flutuações dependentes do momento do dia, o dia da semana, a época do ano, etc. Estes registo s são então tidos em conta à luz das condições existentes em cada momento, sendo feita uma previsão da procura provável de electricidade. Além disto, se existirem em ficha dados estatísticos relacionados com a produção de electricidade de todas as estações, e se estes dados estiverem por sua vez relacionados com as várias áreas de distribuição, o computador será capaz de avisar com antecedência quais as estações que devem ser desligadas ou postas em funcionamento, de modo a assegurar um equilíbrio entre a procura de corrente e a capacidade de produção desta. Um sistema deste tipo ajuda ainda, por um lado, a evitar a sobrecarga e consequentes cortes de energia, e por outro lado, a obviar a uma perda indevida de combustível, resultante de uma situação de produção excessiva.

### *Educação*

Nos últimos anos, aumentou rapidamente o uso de computadores em processos de aquisição de conhecimentos. Virtualmente todas as universidades, escolas técnicas e muitos colégios de tecnologia têm o seu próprio computador, ou pelo menos acesso a um.

Como é evidente, uma proporção significativa do uso destas máquinas é dedicado ao aprendizado de manuseamento dos próprios computadores, ou seja, de processamento de dados, técnica de computadores, cursos de pro-

gramação, etc. No entanto, é também feito um uso extensivo dos computadores em outras disciplinas.

Em trabalho experimental que envolva temas de física, biologia ou sociologia, podem ser elaborados programas que produzam «modelos» (ou seja, simulacros de situações) que constituem uma oportunidade de ensaiar teorias. Dados variáveis e estatísticas relacionadas com diferentes condições são então aplicadas ao modelo, cujo efeito será determinado pelo computador. Isto evita a tarefa quase impossível de produzir situações experimentais reais.

Os computadores podem ser igualmente utilizados como ferramenta de ensino, mesmo no caso de estudantes de pouca idade, num processo designado por *ensino programado*. Este sistema é ainda muito limitado em vários países, mas usado de maneira extensiva na América. O princípio utilizado consiste em armazenar material educativo em fichas de computador numa progressão lógica de uma dada matéria. São então projectadas unidades de informação num visor para consulta e aprendizado pelo estudante. Depois são projectadas perguntas sobre esta informação apresentando várias respostas das quais apenas uma é correcta.

Espera-se que o estudante indique a resposta correcta utilizando uma «caneta luminosa» ou premindo uma tecla que representa essa resposta. Se for realizada uma escolha correcta entre as respostas, o computador passará ao tema seguinte, enquanto que uma escolha errada provocará a apresentação de informações adicionais sobre o mesmo tema. Deste modo, um estudante é colocado numa situação de aprendizagem, cuja evolução pode ser controlada por ele próprio individualmente, sem necessidade de vigilância constante por um professor.

Uma interessante aplicação educacional é a revisão de provas, caso em que as respostas às perguntas serão feitas por escolha entre um certo número de possibilidades.

Se, por exemplo, considerarmos a pergunta:

«A relva é verde, amarela, vermelha, violeta, azul?»

a resposta pode ser dada riscando com uma caneta ou sublinhando a palavra «verde». Teremos assim uma condição sim-não para todas as variáveis apresentadas; por outras palavras, a pergunta foi respondida por sim, não, não, não, não. Apenas necessitamos agora de um leitor que diferencie entre as variáveis marcadas e não marcadas, obtendo-se assim uma entrada directa em termos binários. Naturalmente, a tecla de resposta será indicada no programa de modo a que se possam tomar decisões entre a resposta correcta e a errada.

Uma outra utilização de computadores em educação é a que se torna possível no campo da administração. A quantidade de papel utilizada numa escola ou colégio é imensa, dado que o estabelecimento pode ser frequentado por milhares de estudantes. Um computador constitui um sistema muito conveniente e de referência rápida para registar dados sobre os estudantes, incluindo resultados de exames, e para análise estatística destes registo.

### *Sistemas de informação de gestão*

Duas áreas particularmente importantes no campo da gestão são (a) o controlo de actividades correntes, e (b) a planificação de actividades futuras. Para realizar estas em boas condições, deve haver informação disponível sob a forma correcta e na altura mais conveniente para consulta pelos gestores. As decisões baseadas em informações a menos (ou inclusivamente a mais) podem ser incorrectas, e as fundamentadas em dados atrasados podem ser demasiado tardias para poderem ter alguma utilidade.

Nestas condições, uma vantagem evidente do uso dos computadores consistirá no aumento da velocidade do ciclo de apresentação dos dados necessários, não só reduzindo o número de estágios processuais envolvidos, como

ainda realizando funções básicas na determinação desses dados a grande velocidade. Em muitos sistemas manuais, a produção de informação de controlo pode constituir um processo muito lento e laborioso, devendo muitas vezes percorrer toda uma hierarquia de níveis operacionais antes de ser possível obter uma ideia geral, utilizável pela administração. Na altura em que se dispõe finalmente da informação, já passou o momento em que devia ser tomada a decisão eficaz.

Para um computador que contenha todas as fichas relativas às actividades consideradas, trata-se no entanto de um processo único. De facto, se pensarmos em termos de capacidade interrogativa dos terminais, é evidentemente possível dispôr de informação completamente actualizada num visor montado na própria secretaria do gestor.

No entanto, se bem que a rapidez de obtenção de dados seja uma consideração crucial num sistema de informação de gestão, o grau de pormenor em que esta informação é apresentada é também muito importante. A gestão a alto nível não está interessada em receber enormes quantidades de dados com detalhes de todas as transacções e actividades. Depois de definidos objectivos e orçamentos, os gestores estão principalmente interessados naqueles sectores em que a evolução não correspondeu às previsões. A gestão a níveis inferiores, no entanto, já requererá uma maior quantidade de pormenor, no que se refere à sua esfera de responsabilidade. Para que um director de produção tome as acções necessárias, pode ser suficiente saber que a fábrica A não está a cumprir os seus objectivos. O director da fábrica A já necessitará, no entanto, de uma descrição mais detalhada das actividades nessa unidade de produção, a fim de poder identificar as deficiências existentes, enquanto que o chefe de um departamento poderá ser obrigado a conhecer, em todos os pormenores, cada um dos factores de produção que lhe dizem respeito.

É nestas áreas que o computador se torna mais importante. A sua capacidade para ser programado de modo a

discriminar entre a excepção e a regra e produzir por sua vez a informação estatística relevante no grau de pormenor necessário para cada nível da administração, constitui uma ferramenta de controlo de gestão que de outro modo não seria possível.

Ao nível da planificação futura, o computador pode igualmente desempenhar um papel muito importante na determinação dos resultados prováveis do desenvolvimento ulterior da actividade. O processo de simulação significa construir modelos matemáticos que representam uma situação real, aplicar estes diversos parâmetros de modo a representar condições variáveis e levar o computador a determinar os resultados prováveis. A simulação constitui uma ferramenta muito útil na tentativa de determinação de tendências futuras dos negócios e actividade produtiva em função dos recursos existentes, alterações de concepção, procura do produto e condições gerais do mercado nacional e internacional.

Um outro auxiliar importante de gestão em que o computador pode também desempenhar um importante papel são as *técnicas de planeamento de instalações*, das quais as duas mais importantes actualmente são a PERT (Program Evaluation and Review Techniques) e a CPA (Critical Path Analysis). Num projecto complexo, que utilize uma vasta gama de recursos que devem ser coordenados de acordo com um programa de actividades em tempos críticos, estas técnicas procuram especificar o programa de actividades com recursos óptimos que resultam na realização e funcionamento mais eficazes do projecto. Existem programas normalizados para aplicação destas técnicas.

### *Outras aplicações*

#### *Controlo de tráfego*

O computador tem permitido o desenvolvimento de sistemas de controlo de tráfego utilizando luzes que acendem em função de volumes de tráfego e não de tempos. Em vez

de tempos fixos de variação das luzes de tráfego, existem dispositivos de controlo nas ruas que, quando se aproxima tráfego, assinalam este facto a um monitor, transmitindo essa informação a um computador, no qual se encontram registados dados quanto à maneira de controlar o tráfego em cada ponto concreto. Os sinais podem então ser accionados por controlo computarizado, permitindo obter um caudal óptimo de veículos.

### *Aviação*

Os computadores têm sido muito utilizados para aumentar a segurança das viagens aéreas, tanto do ponto de vista do voo individual como do controlo do tráfego aéreo geral. O desenvolvimento de minicomputadores e a sua instalação a bordo de aviões forneceu um meio de controlo contínuo das condições de voo e do funcionamento mecânico, permitindo ainda facilitar a relacionação de todos estes factores. A tripulação do avião dispõe assim de informações necessárias para efeitos de navegação ou sobre erros de funcionamento e, por vezes, é até possível desencadear automaticamente uma acção correctiva. O aumento muito rápido, nos últimos anos, do tráfego aéreo conduziu ao uso de computadores para processar dados relacionados com o movimento dos aviões, a fim de computerizar rotas, níveis de voo, trajectos de aproximação, etc., aumentando a segurança do voo.

### *Medicina*

Os pequenos computadores, além de serem bastante usados no modo de processamento de dados para manter registos clínicos, estão a encontrar um uso cada vez maior na própria prática da medicina. Dispositivos de varriamento ligados on-line ao computador podem ser ligados a um doente de modo a vigiar as suas condições clínicas. Qualquer variação no estado do doente que obrigue a

realizar um tratamento pode assim ser imediatamente assinalada.

### *Utilizações sociais dos computadores*

O facto de os computadores terem tido um impacto notório na sociedade nos últimos anos não pode evidentemente ser negado. Mas que tenham trazido muitas vantagens à sociedade já poderá ser contestável. Quase todos tivemos já a experiência daquelas, felizmente raras, ocasiões em que um computador erra e produz uma conta com um total «ultrajante», e para muitos de nós o tal papelinho que diz «por favor não dobrar» é uma constante fonte de irritação. Muitas pessoas estão preocupadas, e talvez até um pouco amedrontadas, com o efeito que os computadores estão a ter na nossa sociedade, particularmente nas seguintes áreas.

#### *Emprego*

Um argumento com algum valor, que muitas vezes é lançado em apoio dos computadores, é que a máquina elimina a necessidade de utilizar muitos dos instrumentos anteriormente associados a trabalhos de rotina nos escritórios. De facto, esta é uma área em que o computador pode trabalhar muito mais depressa e eficientemente que o equipamento tradicional. O reverso desta situação é que, simultaneamente, se desaloja assim uma grande parte dos empregados, provocando desemprego.

A experiência mostrou, nos primeiros anos do processamento de sistemas comerciais, que o medo de desemprego era até certo ponto infundado, devido à necessidade de preparar intensivamente os dados para utilização pela própria máquina, por exemplo, na preparação de cartões perfurados. No entanto, nos últimos anos, com a introdução de técnicas mais directas de captação de dados, o problema tornou-se mais evidente.

É nas áreas em que os dispositivos computadores foram introduzidos para fornecer um certo grau de automação de processos industriais que o deslocamento de trabalho manual parece constituir um factor de maior importância. Como se mencionou no capítulo 8, o uso de um microprocessador para controlar máquinas-ferramentas «robot» constitui uma tecnologia em crescimento rápido que deve eventualmente conduzir ao desalojamento de números significativos de anteriores operadores do equipamento.

#### *Impessoalidade*

Com a introdução dos computadores, algumas relações humanas foram alteradas. Antigamente, era possível entrar pela porta de uma repartição para resolver directamente um problema com o inspector local. Hoje, os registos de impostos estão contidos num computador que se encontra talvez a centenas de quilómetros de distância e é necessário esperar algum tempo para que o nosso registo seja retirado do computador e enviado para o local. Em alguns países, a renovação da licença de televisão vem de um ponto do país, a da carta de condução de outro, a do imposto automóvel de outro, etc. Os serviços encontram-se cada vez mais centralizados, com um sentimento crescente de que nos encontramos cada vez mais longe das pessoas que se ocupam desses serviços.

#### *Vida privada*

Muitas pessoas encontram-se cada vez mais preocupadas com o volume de dados pessoais, relacionados com cada um de nós, mantido em locais centralizados. Se bem que isto possa conduzir a uma administração mais eficaz dos serviços, é um facto que detalhes sobre os nossos rendimentos, impostos, registos de saúde, créditos, educação, etc., são mantidos em fichas de computador e

as pessoas estão preocupadas com a possibilidade de poder ser dado acesso não autorizado a esta informação, infringindo a vida privada. Acontece de facto que a confidencialidade dos registos existentes em computadores é mais salvaguardada que a dos registos escritos em ficheiros locais. São tomadas rígidas precauções em todos os centros de computadores para impedir o acesso não autorizado aos registos.

Se bem que haja deficiências no uso cada vez maior de computadores, é certo que estes trouxeram grandes benefícios à sociedade.

### *Administração*

Parece inevitável que o crescimento, quer em termos de qualidade quer de dimensão, traga consigo um aumento desproporcionado da quantidade de papel escrito que o acompanha. Isto tem sido verdadeiro na sociedade em geral, à medida que os níveis de vida têm aumentado, provocando um número cada vez maior de situações que obrigam a preencher formulários e fornecer um número cada vez maior de informações pessoais. O mesmo pode acontecer no nosso emprego, à medida que cresce a organização onde trabalhamos. A comunicação escrita tende a sobrepor-se às antigas relações pessoais; o preenchimento de um formulário torna-se necessário nos casos em que um simples interrogatório pessoal bastava. Na administração do comércio e da indústria, a multiplicidade de legislações centrais e locais conduziu a aumentos no volume de «papelada» necessária.

Mesmo com a introdução de computadores, a proporção de força de trabalho aplicada em administração e trabalho em escritórios tem aumentado continuamente. Sem os computadores, o aumento do número de pessoas empregadas em trabalho de escritório teria sido muito mais nítido. Os computadores não só eliminaram grande

parte do equipamento que de outro modo seria necessário, como ainda criaram novas áreas de actividade muito mais estimulante.

#### *Produtividade*

A utilização de computadores em actividade industrial, controlo de produção, investigação técnica, controlo de stocks, investigação de mercados, etc., pode conduzir a um nível de rendimento muito superior no uso dos recursos existentes. O desgaste pode ser minimizado, o tempo necessário para muitos projectos de investigação e planeamento diminuído significativamente e a eficácia da produção optimizada através da coordenação de processos de produção. Tudo isto beneficia a sociedade, permitindo obter produtos mais baratos, maior qualidade técnica e uma gama cada vez maior de produtos que podem enriquecer os nossos tempos livres, aumentar a nossa saúde e proporcionar locais de trabalho mais estimulantes.

#### *Inovação*

Sem o computador, é difícil ver como poderiam ser realizados alguns serviços que todos exigimos. O sistema de cartões de crédito não seria possível sem controlo por computador. Se bem que a previsão meteorológica ainda seja uma arte imperfeita, sem o computador e a aparelhagem que lhe está associada sê-lo-ia ainda mais. Sem o poder de cálculo de um computador e a sua capacidade para simular e resolver situações problemáticas existentes, muitos dos produtos que actualmente usamos no nosso dia-a-dia teriam levado muito mais tempo a surgir.

Finalmente, se as pessoas pretendem a sua televisão a cores com transmissões por satélite para todo o mundo;

se pretendem uma gama cada vez mais vasta de produtos que facilitem o seu trabalho diário; se querem uma gama de serviços cada vez mais vasta no campo das finanças, da medicina e dos serviços sociais, então devem aceitar as ferramentas necessárias para tal. E, entre estas, o computador é provavelmente a mais perfeita hoje existente.



## **ÍNDICE**

<b>PREFACIO ...</b>	<b>7</b>
<b>1 — Ideias básicas sobre computadores ... ... ... ...</b>	<b>9</b>
<b>2 — Os sistemas numéricos e o computador ... ... ...</b>	<b>23</b>
<b>3 — Lógica de computadores ... ... ... ... ...</b>	<b>43</b>
<b>4 — Entradas e saídas de computadores ... ...</b>	<b>73</b>
<b>5 — O processador central ...</b>	<b>111</b>
<b>6 — Armazenamento e obtenção de informação ... ...</b>	<b>141</b>
<b>7 — Como funcionam os programas de computador ...</b>	<b>165</b>
<b>8 — Grandes e pequenos sistemas de «hardware» ... ...</b>	<b>183</b>
<b>9 — O computador e a sociedade ... ... ... ...</b>	<b>203</b>

Este livro acabou de se imprimir  
em 1979  
para a  
EDITORIAL PRESENÇA, LDA.  
na  
Tipografia Nunes, Lda. — Porto

Tiragem 3 000 exemplares





**EDITORIAL PRESENÇA / MARTINS FONTES**

No campo do actual desenvolvimento tecnológico, o computador ocupa destacada posição pela importância crescente da sua aplicação na indústria, no comércio, na sociedade em geral. Escrito por um dos mais conceituados divulgadores do assunto, este livro representa uma lúcida, completa e concisa introdução à técnica de computadores, desde os seus princípios básicos de funcionamento até à programação. Trata-se pois de uma obra que, além do leitor curioso, interessa muito particularmente todos aqueles que se iniciam numa profissionalização neste campo.