

Capítulo 14

Noções de eletrônica

É perfeitamente possível montar e configurar um micro sem ter conhecimentos básicos de eletrônica. Por isso começamos este livro apresentando placas, memórias, processadores, etc. Por outro lado, muitos cursos de hardware começam apresentando conceitos básicos de microinformática e eletrônica. Atendendo a pedidos de diversas escolas que adotam nossos livros e começam seus cursos com esses conceitos, acrescentamos este capítulo, que no caso é ensinado nesses cursos, antes mesmo do capítulo 1.

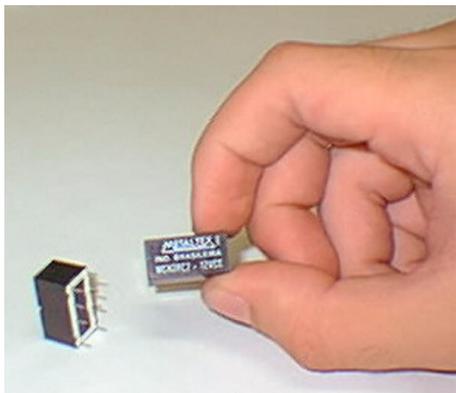
Noções de eletrônica é apenas o título, na verdade apresentaremos aqui uma introdução à microinformática, eletricidade, eletrônica analógica e digital. Nos cursos técnicos, cada um desses assuntos é uma matéria independente. Nesse capítulo faremos uma apresentação rápida desses assuntos.

Organização de computadores

A organização de computadores é uma matéria ministrada em todos os cursos de computação. Ensina uma série de princípios sobre o funcionamento interno dos computadores. Esses princípios são válidos para qualquer tipo de computador, não importa o tamanho, a marca, o modelo, ou se trata de um computador novo ou antigo. Vamos apresentar alguns conceitos, dando exemplos baseados nos microcomputadores.

Os primeiros computadores eletrônicos

Um computador precisa operar com bits. Um bit é a unidade mínima de informação, que pode assumir dois valores: 0 e 1. Podemos expressar os bits também como “falso e verdadeiro”, ou “desligado e ligado”, a idéia é sempre a mesma. Para criar um circuito que gere esses dois valores diferentes, foi usado no início do século XX um componente elétrico chamado *relé*. Os relés são usados até hoje em algumas aplicações, por exemplo, na lógica de controle de elevadores. O relé nada mais é que um interruptor comandado eletricamente. Um interruptor normal é comandado pelo toque. Por exemplo, acender ou apagar uma lâmpada, tocar uma campainha. O relé substitui o toque por uma corrente elétrica. Ou seja, enviando uma corrente elétrica, o relé liga. Desligando a corrente elétrica, o relé desliga. Parece simples, mas as máquinas ancestrais dos computadores eletrônicos eram construídas assim.

**Figura 1**

Relés.

**Figura 2**

Válvula eletrônica.

Os relés tinham várias desvantagens. Eram lentos e consumiam muita energia elétrica. Como eram eletromecânicos, sofriam desgaste com o uso. A partir da década de 1930 foram criadas as primeiras calculadoras a válvula. A válvula eletrônica (figura 2) também tinha elevado consumo de corrente, era muito grande e sofria desgaste (assim como lâmpadas, queimam depois de alguns milhares de horas de uso). A grande vantagem da válvula eletrônica era a sua velocidade elevadíssima de funcionamento em comparação com os relés.

Note que os primeiros computadores eram usados simplesmente para fazer cálculos, eram então chamados de “calculadores”. Eram usados para aplicações financeiras e militares. Aliás, o nome mudou para “computador” porque essas máquinas eram usadas para computar, calcular. Hoje ainda são chamados de computadores, mas são usados também para conversar, ouvir música, lazer, etc.

**Figura 3**

Uma parte do computador ENIAC (1939).

A figura 3 mostra parte de um dos primeiros computadores eletrônicos, o ENIAC (Electronic Numeric Integrator and Calculator). Era usado para cálculos de trajetórias de artilharia.

Computadores transistorizados

Uma grande melhoria em todos os aparelhos eletrônicos ocorreu após a invenção do transistor (1947). Esses pequenos componentes serviam para substituir as válvulas, mas com muitas vantagens. Eram muito menores, consumiam menos corrente elétrica e duravam muitos anos. Tornou-se possível a construção de computadores de menor tamanho, mais rápidos, mais confiáveis e mais baratos. Já no final dos anos 50, todos os computadores eram construídos com transistores. Também passaram a ser fabricados em série. Esses computadores ainda custavam milhões de dólares, mas passaram a ser usados em aplicações não militares:

- Aplicações comerciais em grandes empresas
- Controle de processos industriais

A indústria de computadores começou a crescer dando origem ao desenvolvimento dos grandes gigantes da informática mundial, como a IBM.

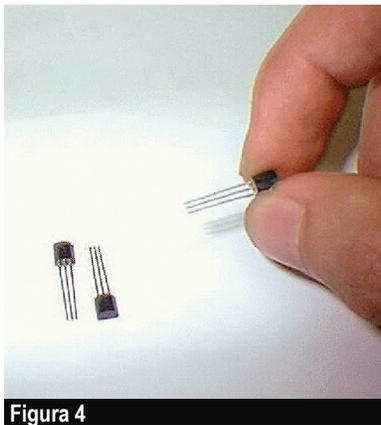


Figura 4
Transistores.

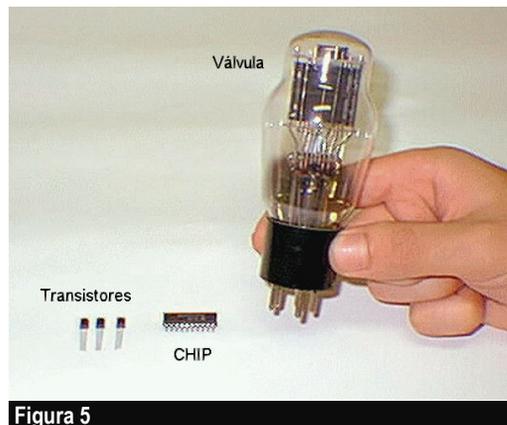


Figura 5
Válvula, transistores e chip.

Circuitos integrados

Ao mesmo tempo que os computadores transistorizados eram cada vez mais utilizados em todo o mundo, um outro grande avanço tecnológico ocorria: a corrida espacial. Americanos e soviéticos lançavam seus foguetes rumo ao espaço. A miniaturização de computadores era ainda mais importante, no caso de um computador a ser colocado a bordo de um foguete. Seria totalmente inviável levantar vôo carregando um enorme computador valvulado. Já para um computador transistorizado, isto era possível, mas se fosse conseguida uma miniaturização ainda maior, computadores mais poderosos ou então mais leves (ou ambas as coisas) poderiam ser embarcados nos foguetes.

A NASA (Agência Espacial Norte-Americana) gastou bilhões de dólares com seu programa espacial, contratou empresas fabricantes de transistores para que realizassem uma miniaturização ainda maior. Uma dessas empresas, até hoje uma líder mundial em microeletrônica, é a Texas Instruments. Foram então criados os primeiros *circuitos integrados*, também chamados de *chips*. Um circuito integrado é um pequeno componente eletrônico que possui em seu interior, centenas, ou até milhares de transistores. A figura 5 mostra a comparação de tamanhos entre uma válvula, um transistor e um chip dos mais rudimentares. Enquanto um transistor é equivalente a uma válvula e tem um tamanho muito menor, um chip dos mais simples tem aproximadamente o mesmo tamanho que um transistor comum, mas em seu interior existem, na verdade, centenas de transistores. Um processador moderno nada mais é que um chip com mais de 100 milhões de transistores.

Os primeiros microprocessadores

Você já sabe o que é um *processador*. Pentium 4, Core 2 Duo, Athlon 64, são exemplos de processadores. Antigamente eles eram chamados de microprocessadores. Foi assim até a época do microprocessador 80486 (1989-1995). O Pentium foi o primeiro *microprocessador* a ser chamado de *processador*. Tecnicamente não existe diferença, a mudança foi motivada pelo marketing. Os microcomputadores que antigamente eram quase sempre tratados como “brinquedos de nerds”, passaram a assumir um papel importante, inclusive no ambiente corporativo, substituindo os terminais de vídeo dos computadores de grande porte. Os fabricantes desses chips acharam por bem eliminar o prefixo “micro” dos seus chips, o que dá uma idéia de amadurecimento.

Não confunda essas duas palavras:

Microcomputador:

É um computador pequeno, de tamanho tal que pode ser colocado sobre uma mesa. Quando surgiram os microcomputadores, existiam apenas os computadores de grande porte (que ocupavam salas inteiras) e os minicomputadores, que eram do tamanho de uma geladeira.

Microprocessador:

É um pequeno chip que cabe na palma da mão. Podemos dizer que esse chip é o “cérebro” do computador. É ele que executa os programas, faz os cálculos e toma as decisões, de acordo com as instruções armazenadas na memória. Hoje esses chips são conhecidos como *processadores*.

Os microprocessadores formam uma parte importantíssima do computador, chamada de UCP (Unidade Central de Processamento), ou em inglês, CPU (Central Processing Unit). Antes da existência dos microprocessadores, as CPUs dos computadores eram formadas por um grande número de chips, distribuídos ao longo de uma ou diversas placas. Um microprocessador nada mais é que uma CPU inteira, dentro de um único CHIP.

Os computadores de pequeno porte, anteriores à era dos microprocessadores, eram chamados de *minicomputadores*. Tinham aproximadamente o tamanho de uma geladeira e eram formados por diversas placas. Uma placa só com memórias, uma placa com interfaces seriais, uma placa com interfaces para disco e fita magnética, e uma placa que formava a CPU. Esta era chamada *placa de CPU*. Os primeiros microcomputadores continuavam tendo uma placa chamada placa de CPU, na qual ficava o microprocessador e vários chips de apoio. Essas placas passaram a incluir também memória e algumas interfaces. Continuavam sendo chamadas de *placas de CPU*, mas aos poucos este termo foi sendo substituído por *placa de sistema* (system board) e *placa mãe* (motherboard).

Podemos ver na figura 6, um microprocessador e uma placa de circuito. Um microprocessador contém todos os circuitos que antigamente eram formados por diversas placas.

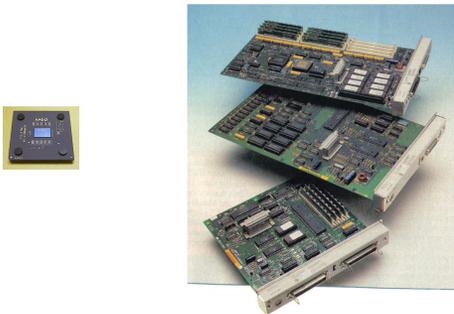


Figura 6
Microprocessador e placas de circuito.

O surgimento da Intel e o microprocessador 4004

A Intel foi uma das primeiras a produzirem microprocessadores. Seu primeiro microprocessador era chamado de “4004”. Era capaz de realizar operações com apenas 4 bits de cada vez. Para simplificar, um microprocessador de 4 bits só pode operar com números pequenos, de 0 a 15. Para usar números maiores, um microprocessador de 4 bits precisa dividir o número em várias partes e fazer as contas em várias etapas. Podemos exemplificar isto, fazendo uma analogia com o que acontece com o cérebro humano. Por exemplo, para fazer a conta “37 x 21”, fazemos por etapas:

37	$1 \times 7 = 7, \quad 1 \times 3 = 3$
x 21	$2 \times 7 = 14, \quad 2 \times 3 = 6, \quad 6 + 1 = 7$

37	$7 + 0 = 7, \quad 3 + 4 = 7, \quad 7 + 0 = 7$
+74	

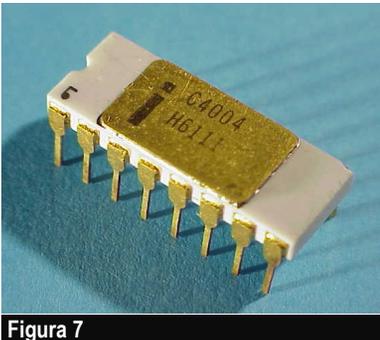
777	

Como nosso cérebro só sabe multiplicar números menores que 10, dividimos a operação em várias etapas, e encontramos assim o resultado 777. Esta multiplicação “complexa” precisou ser composta a partir de 4 multiplicações simples e 4 adições. Programas para o 4004 fazem esse mesmo tipo de desmembramento para operar com números maiores.

Depois do 4004, a Intel lançou o 8008, que era um microprocessador de 8 bits. Era muito mais rápido que o 4004, já que podia operar com números maiores. Com 8 bits, esse chip podia operar diretamente com números entre 0 e 255. O que o 4004 precisava de duas etapas para realizar, podia ser realizado em uma única etapa pelo 8008. Esses chips eram caríssimos. Custavam, na época do seu lançamento, mais de 1000 dólares!

Depois do 8008, a INTEL lançou um novo microprocessador de 8 bits, chamado de 8080. Era mais rápido e mais barato que o 8008. O 8080 foi o primeiro microprocessador a ser usado em larga escala nos chamados “computadores pessoais”. Antes deles, os microcomputadores eram usados apenas em laboratórios científicos, em fábricas e em universidades. O 8080 popularizou o uso de microcomputadores por pequenas empresas e até para uso pessoal.

Outros fabricantes também produziam processadores nessa época. Podemos citar a AMD, que era na ocasião parceira da Intel que produzia chips idênticos, mediante pagamento de royalties, a Motorola que produzia processadores como o MC68000 (usado nos primeiros computadores Macintosh) e a Zilog que produzia os processadores Z80 e Z8000. O grande salto da Intel ocorreu quando a IBM escolheu seu processador de 16 bits mais simples, o 8088, para usar no IBM PC. O 8088 era uma espécie de “Sempron” ou “Celeron” da época, já que o modelo mais avançado da Intel era o 8086, que operava com 16 bits internos e 16 bits externos. O 8088 operava internamente com 16 bits, mas acessava a memória em grupos de apenas 8 bits, sendo portanto mais lento, e um pouco mais barato.

**Figura 7**

Microprocessador Intel 4004.

**Figura 8**

Microprocessador Intel 8088.

CPU, Entrada e Saída

Todos os computadores possuem uma parte muito importante, chamada de *unidade central de processamento* (UCP). Em inglês, usamos a sigla CPU, que é abreviatura de *central processing unit*. Nos computadores de grande porte, a CPU é formada por uma ou várias placas. Cada uma dessas placas contém vários chips. Nos microcomputadores a CPU nada mais é que o próprio processador. É também comum chamar a placa que contém o processador de *placa de CPU* ou *placa mãe*. Não quer dizer que a CPU seja a placa inteira. A *placa de CPU* é a placa que contém a CPU, ou seja, que contém o processador.

Note que “CPU” não é a mesma coisa que “processador”, já que em computadores de maior porte (os supercomputadores, por exemplo), a CPU é formada por uma ou mais placas, com vários chips. Entretanto nos PCs é correto dizer que o processador é a CPU do computador, e que a placa de CPU de um microcomputador é a placa onde está localizado o processador.

Não importa de que tipo de CPU estamos falando, seja um microprocessador, ou uma das várias placas que formam a CPU de um computador de grande porte, podemos dizer que a CPU realiza as seguintes tarefas:

a) Busca e executa as instruções existentes na memória.

Os programas e os dados que ficam gravados no disco (seja ele um disco rígido, CD, *memory key* ou outro meio de armazenamento), são transferidos para a memória. Uma vez estando na memória, a CPU (no nosso caso, o processador) pode executar os programas e processar os dados.

b) Comanda todos os outros chips do computador.

A CPU é auxiliada por vários circuitos que desempenham diversas funções. Por exemplo, quando você pressiona uma tecla, faz com que o teclado transmita o código da tecla pressionada. Este código é recebido por um circuito chamado de *interface de teclado*. Ao receber o código de uma tecla, a interface de teclado avisa a CPU que existe um caracter recebido. Por outro lado, quando a CPU precisa enviar uma mensagem para o usuário, precisa que a mensagem seja colocada na tela. Isto é feito com auxílio de um circuito chamado de *interface de vídeo*. A CPU envia para a interface de vídeo, a mensagem, seja ela em forma de texto ou figura. A interface de vídeo coloca então a mensagem na tela.

Quando a CPU executa instruções e processa dados, dizemos que está *processando*. Quando um circuito recebe um dado e o transmite para a CPU, como no caso do teclado, dizemos que se trata de uma operação de *entrada de dados* (Input). Quando um circuito transmite um dado, como no caso do vídeo, ou da impressora, dizemos que se trata de uma operação de *saída de dados* (Output). Podemos dizer que o computador é uma máquina que passa o tempo todo realizando três operações:

Entrada / Processamento / Saída

A entrada de dados é realizada por diversos dispositivos coordenados pela CPU. Entre eles podemos citar o teclado e o mouse. O processamento é realizado pela própria CPU. A saída de dados é realizada por vários dispositivos, sob a coordenação da CPU. Entre eles podemos citar o vídeo e a impressora.

Vejam então um pequeno resumo dos conceitos apresentados aqui:

CPU - É a Unidade Central de Processamento. Em computadores de grande porte, a CPU é formada por uma ou mais placas. Nos microcomputadores, a CPU é o processador.

Placa de CPU - Todo microcomputador possui uma placa principal, chamada de *placa de CPU* ou *placa mãe*. Esta placa contém o processador, a memória e outros circuitos importantes.

Processamento - É a principal função da CPU. Além de realizar o processamento dos dados. A CPU também comanda as operações de entrada e saída, que são realizadas por circuitos auxiliares chamados de *interfaces*.

Bits e bytes

Começemos com uma explicação simplificada do que é bit, byte, kB, MB e GB.

BIT - Número que pode representar apenas dois valores: 0 e 1 (sua representação é o *b* minúsculo).

BYTE - Grupo de 8 bits. Pode representar valores numéricos entre 0 e 255. Pode também ser usado para representar caracteres (sua representação é o *B* maiúsculo).

kB (KILOBYTE) - Um grupo de aproximadamente 1.000 bytes (sua representação é o *k* minúsculo e o *B* maiúsculo).

MB (MEGABYTE) - Um grupo de aproximadamente 1.000.000 bytes (sua representação é o *M* maiúsculo e o *B* maiúsculo).

GB (GIGABYTE) - Um grupo de aproximadamente 1.000.000.000 bytes (sua representação é o *G* maiúsculo e o *B* maiúsculo).

As definições de kB, MB e GB acima não estão precisas. Na verdade os valores exatos são 1.024, 1.048.576 e 1.073.741.824, respectivamente. Vamos então entender porque são usados esses valores.

Estamos acostumados a utilizar o *sistema decimal de numeração*. Esse sistema usa 10 algarismos para formar todos os números: 0, 1, 2, 3, 4, 5, 6, 7, 8, e 9. O sistema de numeração decimal usa exatamente 10 algarismos, devido ao fato dos seres humanos terem 10 dedos. Historicamente o número 10 foi escolhido, pois os números eram usados na vida cotidiana para contar.

Imagine um número qualquer de 3 algarismos, como por exemplo, 732. Dizemos que esse número tem 3 dígitos decimais. O primeiro dígito, o das centenas, é o 7. O segundo dígito, o das dezenas, é o 3, e o terceiro dígito, o das unidades, é o 2. Existem muitos outros números de 3 dígitos. O menor deles é 100 e o maior deles é 999.

Os computadores podem receber valores decimais, através do teclado, e escrever valores decimais, através do vídeo, por exemplo. Mas internamente, ou seja, no interior da CPU e da memória, os valores são armazenados em um outro sistema, mas adequado aos circuitos do computador. Trata-se do *sistema binário*. Enquanto no *sistema decimal* cada dígito pode assumir 10 valores (0, 1, 2, 3, ..., 9), no *sistema binário* cada dígito pode assumir apenas 2 valores: 0 e 1. Para nós é complicado raciocinar com

números binários, mas para os circuitos do computador, esta é a forma mais simples. Por exemplo, o número 13, que no sistema decimal é representado apenas com dois dígitos (1 e 3), no sistema binário é representado com 4 dígitos, na forma: 1011.

BIT nada mais é que a abreviatura de **B**inary **digi**T, ou seja, *dígito binário*. Talvez os nossos conhecidos algarismos do sistema decimal devessem ser chamados de “DET” (**DE**cimal **digi**T). Poderíamos dizer nesse caso que os “DETS” existentes são 0, 1, 2, 3, ... 9. Não existe o termo “DET” que estamos apresentando aqui, isto é apenas para fazer uma analogia.

Dentro do computador, todos os dados que estão sendo armazenados ou processados são representados na forma de BITS. Como um BIT é muito pouco, já que pode representar apenas dois valores, os computadores trabalham com agrupamentos de bits. Por exemplo, os processadores antigos, como o 8080, podiam operar com 8 bits de cada vez. Os PCs que usavam os processadores 8088 e 80286 trabalham com 16 bits (apesar de aceitarem também instruções e dados de 8 bits). Os processadores 80386, 80486, Pentium e superiores, operam com 32 bits, apesar de também aceitarem dados de 8 e 16 bits, e de terem algumas instruções especiais para dados de 64 e 80 bits. Novos processadores já têm instruções nativas de 64 bits.

Sempre que um processador, uma memória ou outro chip qualquer precisa receber ou transmitir dados, esses dados são transferidos na forma de BITS. Entretanto, para que a transferência seja mais rápida, esses bits não são transferidos um de cada vez, e sim, vários de uma só vez. Com um único fio só é possível transmitir um bit de cada vez. Com 8 fios é possível transmitir 8 bits de cada vez, o que é muito mais rápido. Nos PCs, os bits são transmitidos em grupos de 4, 8, 16, 32 ou 64 bits simultâneos.

- Um grupo de 4 bits é chamado de NIBBLE
- Um grupo de 8 bits é chamado de BYTE
- Um grupo de 16 bits é chamado de WORD
- Um grupo de 32 bits é chamado de DOUBLE WORD (DWORD)
- Um grupo de 64 bits é chamado de QUAD WORD (QWORD)

Você não precisa decorar todas essas palavras. Basta saber que um BYTE (lê-se “báite”) é um grupo formado por 8 bits. Esses 8 bits caminham sempre juntos, como se fossem pássaros de um mesmo bando. Toda vez que um bit é transferido de um lugar para outro, os 8 bits seguem o mesmo caminho, cada um por um “fio” diferente.

Os bytes podem ser usados para representar números, caracteres, figuras, ou qualquer outro tipo de dado armazenado ou processado em um computador. Para representar caracteres, por exemplo, basta estabelecer um código que indique um número associado a cada caractere. Um código muito utilizado é o ASCII, no qual temos por exemplo:

01000001 - A	01000010 - B
01001010 - L	00100011 - #
01010100 - T	

Ninguém precisa decorar esses números. É importante que você saiba que, por exemplo, quando você pressiona a tecla “T”, o teclado transmitirá para o computador um código numérico que representa esta letra. Não é importante saber qual é o código, mas é importante saber que é formado por 8 bits, e que ficarão armazenados na memória do computador, ocupando exatamente 1 byte.

Vejamos agora o que é kB, MB e GB. Dissemos anteriormente que 1 kB é igual a *aproximadamente* 1000 bytes. Na verdade, 1 kB são 1024 bytes. Este número foi escolhido porque sua representação binária é muito mais simples que a representação do número 1000:

1000 = 01111101000 em binário

1024 = 10000000000 em binário

Por razões de simplificação de hardware, o número 1024 foi o escolhido para representar o “k” da computação. Na vida cotidiana e na física, o “k” vale 1000:

1 km = 1000 metros

1 kg = 1000 gramas

1 kV = 1000 volts

Entretanto, na informática, o multiplicador “k” (lê-se “quilo” ou “ká”) vale 1024. Da mesma forma, o multiplicador “M” (lê-se “mega”), que normalmente vale 1.000.000, na computação vale:

1 M = 1024 k = 1024x1024 = 1.048.576

Portanto, 1 MB (lê-se “um megabyte”) são exatamente 1.048.576 bytes. Mas para efeitos práticos, podemos dizer que 1 MB é aproximadamente 1 milhão de bytes.

O multiplicador “G” (lê-se “giga”), que normalmente vale 1 bilhão, na computação vale:

1 G = 1024 M = 1024x1024x1024 = 1.073.741.824

Portanto, 1 GB (lê-se “um gigabyte”) são exatamente 1.073.741.824 bytes, mas para efeitos práticos podemos dizer que 1 GB é aproximadamente 1 bilhão de bytes.

OBS: É errado usar os termos GIGA e MEGA no plural. O correto é “HD de 80 GIGA”, e não “80 GIGAS”. O correto é “512 MEGA de memória”, e não “512 MEGAS”.

Memória principal

A CPU é a parte mais importante de um computador. Essa importância é tão grande que é comum ouvir pessoas chamando seus computadores pelo nome do processador: “... possuo um Athlon 64...”.

Podemos dizer que depois da CPU, a parte mais importante de um computador é a memória. A *memória principal* é aquela que é acessada diretamente pelo processador.

É formada por diversos tipos de chips, e não somente os módulos de memória. Podemos incluir como memória principal, a Flash ROM onde fica gravado o BIOS, a memória de vídeo, etc.

Além da memória principal, que é diretamente acessada pela CPU, existe também a *memória secundária*, que será estudada na próxima sessão. A memória secundária não é acessada diretamente pela CPU. Seu acesso é feito através de interfaces ou controladoras especiais. Podemos citar como exemplo de memória secundária, o disco rígido. Em geral a memória secundária não é formada por chips, e sim, por dispositivos que utilizam outras tecnologias de armazenamento. O disco rígido, assim como os disquetes e as unidades de fita, usa a tecnologia magnética para armazenar dados. Os discos CD-ROM usam tecnologia ótica. Mais recentemente surgiram dispositivos de armazenamento baseados em chips do tipo FLASH ROM, como memory keys, memórias de câmeras digitais e MP3 Players, etc. Apesar de usarem chips de memória, esses dispositivos são acessados através de interfaces de entrada e saída (ex: USB), portanto são considerados como *memória secundária*.

No caso dos microcomputadores, quase toda a memória principal fica localizada na placa mãe. Entretanto, algumas outras placas, chamadas de *placas de expansão*, também podem conter mais memória. É o caso da placa de vídeo.

Os chips de memória podem ser divididos em duas categorias: ROM e RAM.

RAM

São chips de memória que podem ser lidos e gravados pela CPU a qualquer instante. A CPU usa a RAM para armazenar e executar programas vindos do disco, para ler e escrever os dados que estão sendo processados. Uma outra característica da RAM, é que se trata de uma memória *volátil*. Isso significa que quando o computador é desligado, todos os seus dados são apagados. Por essa razão, é necessário que os programas e dados fiquem gravados no disco, que é uma memória *permanente*.

ROM

É a abreviatura de *read only memory*, ou seja, “*memória para leitura apenas*”. São chips de memória que podem ser lidos pela CPU a qualquer instante, mas normalmente não podem ser gravados. Sua gravação é feita apenas pelo fabricante do computador, ou pelo fabricante de memórias. A outra característica importante de ROM é que se trata de uma memória *permanente*. Seu conteúdo nunca é perdido, mesmo com o computador desligado.

Nos microcomputadores existe um programa muito importante chamado de BIOS (Basic Input-Output System - Sistema Básico de Entrada e Saída). O BIOS tem várias funções, entre as quais, a de realizar a “partida” do computador. Quando ligamos o computador, o BIOS realiza a contagem de memória, faz uma rápida checagem do funcionamento do computador e realiza a carga do Sistema Operacional que deve estar armazenado no disco. O BIOS está gravado em uma memória ROM localizada na placa mãe. Existem tipos de ROM que podem ser gravados através de programas especiais. Por exemplo as do tipo *Flash ROM*. Também ficam disponíveis apenas para

leitura, mas utilizando um programa especial fornecido pelo fabricante da placa mãe, o BIOS pode ser gravado novamente.

Como já mostramos, a placa mãe contém quase toda a memória de um PC, mas outras placas também podem conter memórias, do tipo RAM e do tipo ROM. Por exemplo, as placas de vídeo contém uma ROM com o seu próprio BIOS, e contém uma RAM chamada de *memória de vídeo*, que armazena os caracteres e gráficos que são mostrados na tela.

Memória secundária

A *memória secundária* também é chamada de *memória de massa*. É uma memória do tipo *permanente* (não se apaga quando o computador é desligado), que tem uma alta capacidade de armazenamento, e um custo muito mais baixo que o da memória principal. Considere por exemplo, um computador com 512 MB de RAM (que faz parte da memória principal) e um disco rígido de 200 GB (que faz parte da memória secundária). Os 200 GB do disco rígido servem para armazenar diversos programas e diversos dados. Os 512 MB de RAM servem para manter apenas os programas e os dados que estão sendo processados em um dado instante.

O custo da memória secundária é muito mais baixo que o da memória principal. Enquanto 512 MB de RAM custam pouco mais de R\$ 100 (preços de 2007), um disco rígido de 200 GB custa cerca de quatro vezes mais, R\$ 400. Mas a capacidade desse disco é 400 vezes maior que essa quantidade de RAM. Logo o custo por gigabyte no disco é cerca de 100 vezes mais barato que o da memória.

Memória: R\$ 100 / 0,5 GB = R\$ 200,00 por gigabyte
Disco rígido: R\$ 400 / 200 GB = R\$ 2,00 por gigabyte

Cálculos semelhantes podem ser feitos para outros meios de armazenamento secundário, como CD, DVD e fitas magnéticas. Note que em muitos casos, é preciso levar em conta o custo não só da mídia de armazenamento, mas também da unidade. Por exemplo, para usar um DVD é preciso ter uma unidade de DVD.

A memória secundária é muito mais barata, de maior capacidade, e ainda é permanente, ou seja, não apaga os dados quando o computador é desligado. Por que então esse tipo de memória não é usado no lugar da memória principal? O problema é que infelizmente, os dispositivos de armazenamento secundário são lentos em comparação com a RAM. São lentos demais para serem acoplados diretamente ao processador. Além disso, não permitem acessos a seus bytes individuais, como um processador precisa realizar. Os meios de armazenamento secundário só permitem o acesso a blocos de dados. Em um disco rígido, por exemplo, as leituras são feitas em unidades mínimas chamadas de *setores*. Cada setor tem 512 bytes. Para ter acesso a um único byte, é preciso ler o setor inteiro. Isso faz com que seu acoplamento direto à CPU seja inviável. As memórias RAM e ROM são milhares de vezes mais rápidas e permitem que sejam feitos acessos a qualquer um de seus bytes, de forma individual. Por isso são usadas para formar a memória principal.

Os meios de armazenamento secundário são mais lentos porque envolvem movimento de suas partes mecânicas. O disco rígido, por exemplo, precisa mover as cabeças de leitura até o ponto onde será feita a leitura. A unidade de fita precisa girar a fita até o ponto a ser acessado. As memórias ROM e RAM não precisam desses movimentos, pois seu acesso é inteiramente eletrônico.

Dispositivos de entrada e saída

Já vimos que um computador passa o tempo todo realizando três tarefas:

- a) **Entrada** Feita por chips e dispositivos especializados em leitura de dados, sob a coordenação da CPU.
- b) **Processamento** Feito pela CPU
- c) **Saída** Feita por chips e dispositivos especializados em transmissão ou gravação de dados, sob a coordenação da CPU.

Existem portanto no computador, os chamados *dispositivos de entrada e saída*, também chamados de *periféricos*. Através desses dispositivos, o computador pode armazenar, ler, transmitir e receber dados. A *memória secundária*, já estudada na seção anterior, é formada por diversos dispositivos de entrada e saída.

O termo “Entrada e Saída” é abreviado por **E/S**, ou para quem preferir a língua inglesa, **I/O** (Input/Output). Entre os diversos dispositivos de E/S, existem alguns que são especializados apenas em entrada, outros especializados apenas em saída e outros em ambos. Podemos citar os seguintes exemplos:

Entrada:

- Teclado - Lê os caracteres digitados pelo usuário
- Mouse - Lê os movimentos e toques de botões
- Drive de CD-ROM - Lê dados de CDs
- Microfone - Transmite sons para o computador
- Scanner - Capta figuras e fotos

Saída:

- Vídeo - Mostra na tela caracteres e gráficos
- Impressora - Imprime caracteres e gráficos
- Alto-falante - Emite som

Entrada e saída:

- Disco rígido - Grava e lê dados
- Drive de disquete - Grava e lê dados em disquetes
- Unidade de fita - Grava e lê em fitas magnéticas
- Modem - Transmite e recebe dados pela linha telefônica
- Gravador de CD - Lê e grava CDs
- Gravador de DVD - Lê e grava CDs e DVDs
- Memory Key - Grava e lê dados

A CPU não pode comunicar-se diretamente com os periféricos. Esta comunicação é feita com a ajuda de circuitos chamados de *interfaces*. Por exemplo, os dados que são lidos e gravados em um *memory key* são transferidos através de uma interface USB.

Os PCs modernos saem da fábrica já acompanhados de vários dispositivos de entrada e saída:

- Monitor
- Drive de disquetes
- Disco rígido
- Teclado
- Mouse
- Alto falantes
- Modem
- Placa de som

O usuário pode, nesse caso, adquirir novos periféricos e placas e realizar sua instalação. Essa tarefa pode ser realizada por usuários mais experientes, ou então por técnicos especializados. Quando instalamos novos periféricos e placas em um computador, dizemos que estamos realizando um *expansão* ou *upgrade*. Nesse caso, podem ser instalados, por exemplo:

- Impressora
- Scanner
- Joystick
- ZIP Drive
- Gravador de CDs
- DVD

Arquivos

“Arquivo” é uma das palavras mais importantes em computação. Em inglês, sua tradução é “File”. *Arquivo* nada mais é que um conjunto de dados gravados na memória secundária (disco rígido, disquete, fita magnética, CD-ROM, etc). Os arquivos são uma forma de organizar os dados dentro da memória secundária. Se os dados estivessem todos espalhados, por exemplo, ao longo de um disquete, seu acesso seria extremamente complicado. Podemos fazer uma analogia entre dados, arquivos, casas e ruas. Os dados corresponderiam às casas, enquanto que os arquivos corresponderiam às ruas. Seria difícil localizar uma casa, sabendo apenas os nomes de seus moradores. Sabendo o nome da rua, o acesso é bem mais imediato. Por essa razão, os dados são agrupados em arquivos. Sabendo o nome do arquivo, fica mais fácil localizar os dados.

Os nomes completos dos arquivos são tradicionalmente divididos em duas partes, separadas por um ponto. Por exemplo:

CARTA.DOC

Neste caso, o “nome” é CARTA, e a *extensão* é DOC. É comum nesse caso, dizer que o nome do arquivo é CARTA, ou então CARTA.DOC. Da mesma forma, dizemos que o nome de uma pessoa é, por exemplo, JOÃO DA COSTA, mas podemos dizer também que o nome é apenas JOÃO.

A extensão serve para indicar o tipo de arquivo. Por exemplo, DOC significa que se trata de um documento de texto; JPG é um arquivo gráfico, EXE é um arquivo de programa, WAV é um arquivo de som, e assim por diante. No passado, os arquivos podiam usar no máximo 8 caracteres para o nome e 3 caracteres para a extensão (formato 8.3). A partir do lançamento do sistema operacional Windows 95 da Microsoft e do OS/2, da IBM, os PCs passaram a utilizar nomes longos. Hoje podemos usar para os arquivos, nomes como:

Relatório Mensal.DOC
Foto do passeio 001.JPG
Rugido do leão.WAV

Os arquivos podem armazenar diversos tipos de dados:

Instruções para a CPU:

Dizemos que se trata de um *arquivo executável*. Os arquivos executáveis, nos microcomputadores, normalmente usam a extensão EXE. Por exemplo:

DXDIAG.EXE
MPLAYER.EXE
EXPLORER.EXE

Documentos:

São textos digitados com o auxílio de um tipo de programa (aplicativo) chamado de *editor de textos*, ou *processador de textos*. Normalmente esses arquivos usam a extensão TXT ou DOC. Por exemplo:

CURRIC.DOC
LISTA.TXT
REUNIÃO.DOC

Gráficos:

São arquivos que representam figuras. Essas figuras podem ser vistas na tela ou na impressora, com o auxílio de programas apropriados.

Dados genéricos:

Muitas vezes os programas precisam manipular uma quantidade de dados tão grande que não cabem na memória principal. Nesse caso, esses dados são armazenados em arquivos que são lidos da memória secundária e processados por partes. Muitas vezes esses dados podem até caber na memória principal, mas por uma questão de organização ficam armazenados em arquivos. Por exemplo, podemos ter um arquivo que contém os nomes dos alunos de um colégio, assim como as notas que cada aluno obteve nas provas já realizadas.

Não podemos “ver” fisicamente os arquivos, já que são gravados em meio óptico ou magnético, ou mesmo eletrônico. Os “olhos” usados para “ver” os arquivos são as cabeças e circuitos de leitura, localizadas dentro dos respectivos periféricos. Através das cabeças de leitura de um disco rígido, por exemplo, seus dados podem ser lidos e transformados em impulsos eletrônicos e enviados para a CPU e para a memória. A CPU, por sua vez, pode acessar os dados originários dos arquivos e enviá-los para os dispositivos de saída, como o monitor e a impressora, para que possamos vê-los com nossos próprios olhos. Por exemplo, se um arquivo contém uma carta, só poderemos ver esta carta depois que o arquivo for lido e interpretado pela CPU (usando o programa apropriado) e enviado para a tela ou para a impressora.

Programas

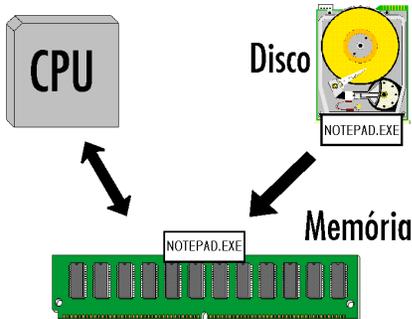
Os computadores passam o tempo todo executando *programas*. Os programas nada mais são que grupos de instruções e dados. Por exemplo, quando você está executando um jogo no computador, está na verdade executando um programa. Se você deixa o computador “parado”, por exemplo, na hora do almoço, pode pensar que nessa hora o computador não está executando nenhum programa. Engana-se. Na verdade, mesmo que você não tenha dado nenhum comando, o computador está na verdade executando inúmeros programas em segundo plano, como módulos do sistema operacional, anti-vírus, drivers, etc. A única hora em que o computador não está executando programa nenhum é quando está desligado ou em *estado de espera*.

Para que um programa possa ser executado, é preciso que seja transferido para a memória RAM. A maioria dos programas ficam armazenados em disco (disco rígido, CD-ROM, etc.), mas a CPU não pode executar nenhum programa diretamente a partir do disco. O programa precisa ser antes lido do disco e carregado na RAM. Por exemplo, para executar o programa *Bloco de Notas* (pequeno editor de textos que acompanha o Windows, cujo nome verdadeiro é NOTEPAD.EXE), é preciso que você clique os menus do Windows na seqüência:

Iniciar / Programas / Acessórios / Bloco de Notas

Uma vez usada esta seqüência de cliques, o NOTEPAD.EXE é lido do disco rígido e carregado na RAM. A CPU pode então executar o programa. A figura 24 simboliza a leitura do programa NOTEPAD.EXE a partir do disco para a memória RAM (essa operação é chamada de *carga*), e seu processamento pela CPU (essa operação é chamada de *execução*).

Você poderá estar pensando como que é feita a mágica da leitura do arquivo NOTEPAD.EXE do disco para a memória, e a seguir sua execução. Na verdade, quem leu o arquivo NOTEPAD.EXE e providenciou sua execução foi um outro programa. Trata-se de um programa que fica o tempo todo na memória, chamado de *sistema operacional*. No nosso exemplo, trata-se do sistema operacional Windows. Uma das várias funções do sistema operacional é ficar o tempo todo ativo na memória RAM, esperando que o usuário comande a execução de algum programa.

**Figura 9**

Carga e execução do programa NOTEPAD.EXE (Bloco de Notas).

Um sistema operacional é um grande conjunto de programas e arquivos auxiliares. O próprio Bloco de Notas é um programa que faz parte do sistema operacional Windows. Outro programa importante do Windows é o EXPLORER.EXE. Este programa é o responsável por, entre outras coisas, receber os comandos que o usuário envia o computador, através do teclado e do mouse. Por exemplo, quando clicamos em *Iniciar* e aparece um menu, no qual consta a opção *Programas*, depois *Acessórios* e finalmente *Bloco de Notas*, é o EXPLORER.EXE que está recebendo os comandos do mouse e apresentando os menus na tela. Portanto, quando você usa um comando ou programa, como o Bloco de Notas, o que ocorrer na verdade é o seguinte:

- 1) Inicialmente o EXPLORER.EXE está checando se você fornece algum comando pelo teclado ou pelo mouse.
- 2) Você clica em Iniciar / Programas / Acessórios / Bloco de Notas.
- 3) O programa EXPLORER.EXE identifica que o programa chamado *Bloco de Notas* é na verdade o NOTEPAD.EXE. Ele envia comandos para outros componentes do sistema operacional para que procurem no disco o arquivo NOTEPAD.EXE e para que façam sua carga na memória RAM.
- 4) O Windows é um sistema operacional que permite que vários programas possam ser executados ao mesmo tempo (multitarefa). Na verdade o processador dedica uma fração do seu tempo para cada um dos programas que se encontram em execução. Por exemplo, enquanto a área de trabalho do Windows continua na tela e novos comandos podem ser usados (ou seja, o programa EXPLORER.EXE está ativo), o processador também exibe a janela do programa NOTEPAD.EXE. Você pode então utilizar o NOTEPAD.EXE, digitando um texto e salvando-o em um arquivo.
- 5) Você finaliza o programa NOTEPAD.EXE, fechando a sua janela ou usando o comando Arquivo / Sair.

Podemos entender então que nenhum programa chega até a memória por mágica, e sim, através do controle feito pelo sistema operacional. Alguém mais observador pode então ficar com a seguinte dúvida: “Se é o sistema operacional quem lê para a RAM

todos os programas a serem executados, como é então que o próprio sistema operacional chegou à RAM?”.

No instante em que ligamos o computador, a RAM não contém programa algum (lembre-se que os dados existentes na RAM são apagados quando o computador é desligado). Nesse instante, o sistema operacional está armazenado no disco rígido e precisa ser carregado na memória. Quem faz a carga do sistema operacional para a memória é um programa chamado BIOS, que fica gravado na memória ROM. Lembre-se que a memória ROM não perde seus dados quando o computador é desligado.

Portanto, no instante em que ligamos o computador, o BIOS já está na memória, e é imediatamente processado pela CPU. O processamento do BIOS começa com uma contagem de memória, seguido de alguns testes rápidos no hardware, e finalmente a leitura do sistema operacional do disco para a memória RAM. Esse processo, ou seja, a carga do sistema operacional na memória RAM, é chamado de *boot*. A figura 10 mostra o processo de boot com maior clareza.

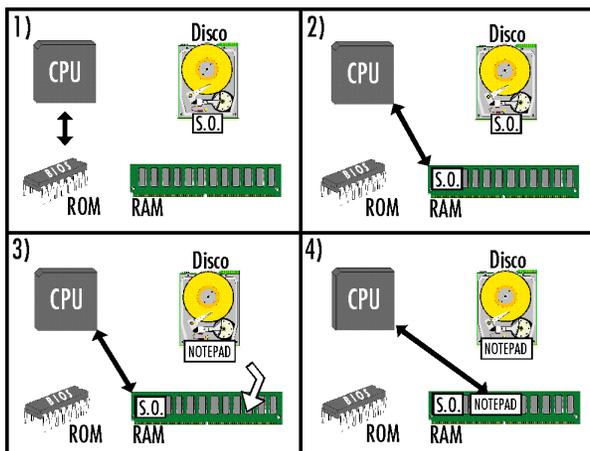


Figura 10

Boot e carga de um programa.

Acompanhe abaixo e na figura 10 os itens 1, 2, 3 e 4:

1) No instante em que o computador é ligado, o sistema operacional (S.O.) está armazenado em disco, a RAM está “vazia”, e a CPU executa o BIOS.

2) Mostra o instante em que termina a operação de boot. O sistema operacional já está carregado na memória e já está sendo executado pela CPU.

3) Mostra o que ocorre imediatamente antes da execução do programa NOTEPAD.EXE. O sistema operacional (mais especificamente, o programa EXPLORER.EXE, que faz parte do S.O.) recebe um comando do usuário para que leia o arquivo NOTEPAD.EXE do disco para a memória RAM.

4) O programa NOTEPAD.EXE está sendo executado pela CPU.

Sistema operacional

Vimos que o sistema operacional é um conjunto de programas que são carregados na memória quando o computador é ligado, e que tem como uma de suas responsabilidades, providenciar a execução dos comandos solicitados pelo usuário. Não fique achando que é só isso o que o sistema operacional faz. Essa é apenas uma de suas funções.

Em um passado recente, muitos sistemas operacionais de diversos computadores receberam o nome de D.O.S, que significa “Disk Operating System” (Sistema Operacional de Disco). Até meados dos anos 90, o sistema operacional mais utilizado nos PCs era o MS-DOS (Microsoft Disk Operating System). A partir de meados dos anos 90, o sistema operacional mais comum passou a ser o Windows 95, seguido pelas suas atualizações (Windows 98, Windows ME, Windows XP...). Ainda assim o chamado “modo MS-DOS” é oferecido juntamente com o Windows. Usuários de programas antigos podem, desta forma, utilizar este “MS-DOS” embutido no Windows.

Uma das atribuições do sistema operacional, como vimos, é fazer a carga e providenciar a execução dos programas que o usuário solicita. Mesmo quando um programa qualquer está em execução, o sistema operacional continua ajudando. Por exemplo, muitos programas precisam realizar acesso ao teclado, vídeo e impressora, assim como acessos ao disco para ler e gravar arquivos. Todos esses acessos são realizados pelo sistema operacional, que fica o tempo todo ativo, prestando serviços aos programas que estão sendo executados.

O sistema operacional também faz um gerenciamento dos recursos do computador, para evitar que os programas entrem em conflito. Por exemplo, o sistema operacional evita que dois programas simultaneamente acessem a mesma área da memória, o que poderia causar grandes problemas. O sistema operacional funciona como um “maestro”, providenciando para que todos os programas e todos os componentes do computador funcionem de forma harmônica.

Bases de numeração

Estamos acostumados a usar a base 10 por motivos históricos: temos 10 dedos, e os dedos foram a primeira tentativa de contar, há alguns milhares de anos. Nesta base são usados 10 algarismos: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Do ponto de vista matemático, podemos ter bases de numeração de qualquer tipo. As bases usadas em computação são as indicadas na tabela abaixo:

Base	Número de dígitos	Dígitos
Binária	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

A contagem de números sucessivos consiste em aumentar o dígito das unidades até o valor máximo (no caso da base decimal, o máximo é 9). Ao chegarmos ao máximo, o

dígito das unidades passa a ser zero, e o dígito das dezenas é aumentado. Quando o dígito das dezenas chega ao valor máximo, ele se torna zero e o das centenas é aumentado, e assim por diante. Sem muito aprofundamento matemático, note que só faz sentido em chamar as posições de unidades, dezenas e centenas se estivermos usando a base 10.

Vamos agora contar os 20 primeiros números em cada uma das bases citadas:

Base 10:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19.

Base 8:

0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23.

Note que o “10” na base octal é o mesmo que “8” na base decimal. O “23” em octal corresponde ao “19” em decimal.

Base 16:

Na base hexadecimal (às vezes chamada simplesmente de HEXA) temos 16 dígitos. São usados além dos tradicionais 0 a 9, novos dígitos com maiores valores. Convencionou-se utilizar letras do alfabeto latino, ao invés de criar novos algarismos. A contagem dos 20 primeiros números na base hexadecimal seria então:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13.

Base 2:

A mesma contagem usando a base 2 ficaria:

0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, 10010, 10011.

Parece estranho, mas todas as bases tem algo em comum, que é o método de contagem. Chegando ao dígito máximo (1 no caso da base binária, 7 no caso da octal, 9 no caso da decimal e F no caso da hexadecimal), ele se torna zero e fazemos o “vai 1”. Poderíamos assim construir uma tabela de equivalência:

Decimal	Binário	Octal	Hexa
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9

Decimal	Binário	Octal	Hexa
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13

Note que se não tomarmos cuidado, podemos fazer confusão entre as bases. Por exemplo, o número 13 tem valores diferentes quando usamos bases decimal, octal e hexadecimal. Quando não é feita ressalva alguma, consideramos que um número está expresso em decimal. Quando é usada uma base diferente, devemos usar um indicador apropriado. Por exemplo, o 13 em hexadecimal pode ser escrito de duas formas:

13h ou 13₁₆

Em octal usamos indicadores como:

13q ou 13₈

Em binário usamos um “b” ou um índice “2” ao seu final. Por exemplo:

1001b ou 1001₂

De acordo com a tabela, vemos por exemplo que:

19 = 10011b = 23q = 13h

Muitas calculadoras científicas possuem funções para conversões de bases de numeração. A própria calculadora virtual que acompanha o Windows tem essas funções. Use Iniciar / Programas / Acessórios / Calculadora para executar o programa. Use então o comando Exibir / Científica. Digite agora um número decimal qualquer e use as opções Hex, Dec, Oct e Bin para fazer as conversões. Quando não temos uma calculadora, podemos fazer mudanças de base através de cálculos manuais. Vejamos como.

Conversão de uma base qualquer para a base decimal

Dado um número expresso em uma base qualquer, tudo o que temos que fazer é somar as parcelas relativas a cada dígito. Cada parcela é igual ao valor do dígito multiplicado pelo seu peso. Os pesos são potências da base usada. Veja por exemplo o que significa o número 2375 na base 10:

$$2375 = 2 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 5 \times 10^0$$

Como vemos, cada algarismo é multiplicado por uma potência da base (lembrando que 10⁰ vale 1 e que 10¹ vale 10). Uma fórmula geral, considerando um número abcde expresso na base x, seu valor na base 10 seria:

$$a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$$

Usemos a fórmula para calcular quanto vale 7D2h (base 16). Ficaria:

$$7 \times 16^2 + D \times 16 + 2$$

Lembrando que o dígito D em hexadecimal vale 13 decimal, e trocando 16² por 256 (16x16), ficaríamos com:

$$7 \times 256 + 13 \times 16 + 2 = 1792 + 208 + 2 = 2002$$

Portanto $7D2h = 2002$.

Em um outro exemplo, vamos converter o número binário 10011101b para decimal. Ficariamos com:

$$1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1$$

Substituindo as potências de 2 ficamos com:

$$1 \times 128 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 1 = 157$$

Conversão de base decimal para uma base qualquer

A conversão de um número decimal para uma base qualquer consiste em realizar divisões sucessivas e tomar os restos dessas divisões. Os valores dos restos formarão os dígitos na nova base. Vamos mostrar o método através do exemplo de conversão do número 2002 decimal para a base 16. Devemos então fazer divisões sucessivas por 16. Cada quociente deve ser a seguir dividido por 16, até que o quociente fique menor que a base. A figura 11 ilustra o procedimento. O número decimal 2002 a ser convertido é dividido pela base desejada, no caso 16. O resultado da divisão foi 125, e o resto foi 2. O número 125 é agora dividido por 16, e encontramos o resultado 7 e resto 13. Enquanto o resultado é maior ou igual à base, continuamos a divisão. Terminadas todas as divisões, o resultado final (no caso, 7) é o primeiro dígito do valor convertido. Os dígitos seguintes são os restos das divisões. Note que o resto 13 é expresso como D em hexadecimal. Concluímos portanto que $2002 = 7D2h$.

$$\begin{array}{r|l}
 2002 & 16 \\
 40 & \underline{125} \\
 82 & \quad \underline{13} \\
 2 & \quad \quad \underline{7}
 \end{array}$$

Figura 11

Exemplo de conversão de base 10 para base 16.

Conversões simplificadas entre binário, octal e hexadecimal

Qualquer número pode ser base de um sistema de numeração. Podemos ter por exemplo, base 15 e base 6. Converter números diretamente de uma base para outra é muito trabalhoso. Por exemplo, para converter um número de base 15 para base 6, fazemos primeiro a conversão de base 15 para base 10, depois de base 10 para base 6. Em informática entretanto são mais usadas as bases 2 e 16, e em alguns casos a base 8. Felizmente é muito fácil fazer conversões diretas entre essas bases, com pouquíssimos cálculos.

Para converter de binário para hexadecimal, divida o número a partir da direita, em grupos de 4. Cada grupo de 4 dígitos deve ser então transformado em um dígito hexadecimal. Vamos tomar como exemplo o número binário 10011101b. Dividindo em grupos de 4 dígitos, da direita para a esquerda, ficamos com:

1001 1101

Agora basta saber que 1001 vale 9, e que 1101 vale D, e temos o valor convertido, 7D. Assim não é preciso fazer cálculos exaustivos, basta conhecer as representações binárias e hexadecimais de números de 0 a 15:

Binário	Hex	Binário	Hex
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Fica fácil fazer a conversão até mesmo de números grandes. Por exemplo, para converter 11011001011011101001101b para hexadecimal, temos:

110 1100 1011 0111 0100 1101 = 6CB74Dh

Note que o primeiro grupo ficou com apenas 3 dígitos, já que a separação é feita da direita para a esquerda. Devemos completar com zeros à esquerda até formar 4 dígitos, portanto 110 fica como 0110, que vale 6 de acordo com a tabela.

A conversão de hexadecimal para binário é ainda mais simples. Basta escrever cada dígito hexadecimal na sua forma binária. Por exemplo, para converter 57CFh para binário temos 5=0101, 7=0111, C=1100 e F=1111:

0101 0111 1100 1111

Podemos agora suprimir os espaços em branco e remover os zeros à esquerda, ficando com 101011111001111b

A base octal é pouco usada, mas suas conversões para binário são simples. Para converter de binário para octal, separamos o número em grupos de 3 dígitos, da direita para a esquerda, e a seguir usamos a tabela:

Binário	Octal	Binário	Octal
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

Por exemplo, para converter 10011101000110b para octal, ficamos com:

10 011 101 000 110 = 23506q

Para converter de octal para binário, basta escrever cada dígito octal na sua representação binária. Por exemplo para converter o número 32613q para binário, usamos 3=011, 2=010, 6=110, 1=001 e 3=011. Ficamos então com:

$$32613q = 011\ 010\ 110\ 001\ 011$$

O zero à esquerda é suprimido e ficamos com 11010110001011b.

Usando um multímetro digital

Um multímetro digital pode ajudar bastante nas atividades de hardware, principalmente em manutenção. Com ele você pode checar as tensões da fonte de alimentação e da rede elétrica, checar o estado da bateria da placa mãe, verificar se o drive de CD-ROM está reproduzindo CDs de áudio, acompanhar sinais sonoros, verificar cabos e várias outras aplicações. Atualmente é possível comprar um multímetro digital por menos de 30 reais, em lojas de material elétrico ou de eletrônica.

Um multímetro possui duas pontas de prova, uma vermelha e uma preta. A preta deve ser conectada no ponto do multímetro indicado com GND ou COM (este é o chamado “terra”). A ponta de prova vermelha pode ser ligada em outras entradas, mas para a maioria das medidas realizadas, a ligação é feita no ponto indicado com V- Ω -mA.

Uma chave rotativa é usada para selecionar o tipo de medida elétrica a ser feita: V para voltagem, Ω para resistência e mA para corrente. Uma chave é usada para a medição de voltagens em AC (corrente alternada) ou DC (corrente contínua). Por exemplo, para medir as tensões da fonte de alimentação, ou a tensão da bateria, usamos a chave em DC. Para medir a tensão presente na saída de áudio de um drive de CD-ROM ao tocar um CD musical (um tipo de corrente alternada), usamos a escala AC. Para medir as tensões da rede elétrica, também utilizamos a escala AC.



Figura 12

Multímetro digital.

Alguns multímetros possuem um único conjunto de escalas para voltagem, e uma chave adicional para escolher entre AC e DC. Outros modelos, como o da figura 12, não

possuem esta chave AC/DC, e sim grupos independentes de escalas para voltagens e correntes em AC e DC. A maioria dos multímetros não mede corrente alternada (ACA), apenas corrente contínua (DCA), tensão alternada (ACV) e tensão contínua (DCV).

Para cada grandeza elétrica existem várias escalas. Por exemplo, entre as várias posições da chave rotativa, podem existir algumas específicas para as seguintes faixas de voltagem: 200 mV, 2 V, 20 V, 200 V e 2000 V.

Se você pretende medir a tensão da bateria da placa mãe (em torno de 3 volts), não use a escala de 2V, pois tensões acima de 2V serão indicadas como 1,9999 V. Escolha então a escala de 20V, pois terá condições de fazer a medição esperada. Da mesma forma, para medir a tensão de uma rede elétrica de 220 volts (use AC, pois se trata de tensão alternada), não escolha a escala de 200 volts, pois a máxima tensão medida será de 199,99 volts. Escolha então a escala de 2.000 volts ou outra para tensões elevadas. Como regra geral, sempre que a leitura indicada tem valor máximo ou outra indicação que esteja fora da escala, devemos utilizar uma escala maior. Quando não temos idéia aproximada da tensão que vamos medir, devemos começar com a escala de maior valor possível, pois se medirmos uma tensão muito elevada usando uma escala baixa, podemos danificar o aparelho.

Para medir a tensão entre dois pontos, selecione a escala e encoste as pontas de prova nos terminais nos quais a tensão deve ser medida (figura 13). Muitas vezes queremos fazer medidas de tensão relativas ao terra (o terminal “negativo” da fonte de alimentação). Você pode então fixar a ponta de prova preta em um ponto ligado ao terra (por exemplo, os fios pretos do conector de alimentação da placa mãe) e usar a outra ponta de prova para medir a tensão no ponto desejado.

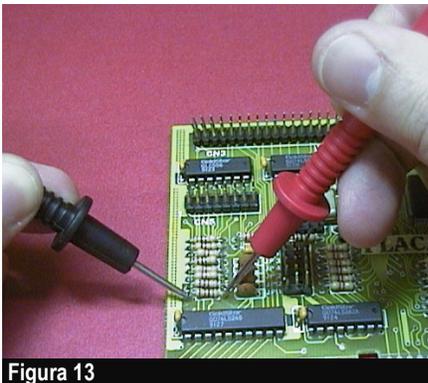


Figura 13
Medição de voltagem.

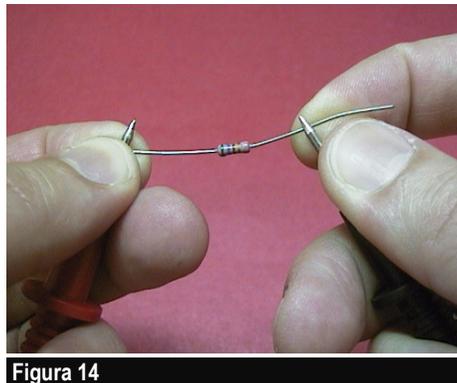


Figura 14
Medindo o valor de um resistor.

A medição de resistência também possui várias escalas, e você deve escolher uma escala que comporte a medida a ser realizada. Se você não tem idéia da escala a ser usada, escolha a maior delas. Por exemplo, se medir um resistor de cerca de 150 ohms em uma escala de 20.000, será apresentado o valor 150. Se quiser maior precisão pode

usar escalas menores. Por exemplo, na escala de 2000 ohms, o valor medido poderá ser 150,3 e na escala de 200 poderá ser 150,37.

Note que não podemos medir o valor de um resistor quando ele está em um circuito. O valor medido será influenciado pelos demais componentes do circuito ligados ao resistor. A medida correta é feita quando o resistor está desacoplado do circuito, como mostra a figura 14.

OBS: Para resistores com valores acima de 10k ohms, é recomendável não tocar as mãos nas pontas de prova do multímetro, pois a resistência do corpo humano provocará erro na medida.

Podemos usar o multímetro na escala de resistência para verificar se um cabo está partido ou se um fusível está queimado. Quando um fio ou fusível está em perfeitas condições, sua resistência é bem baixa, em geral inferior a 1 ohm. Colocamos então o multímetro na escala mais baixa de resistência e fazemos a medida. Quando o cabo está partido ou o fusível está queimado, a resistência é muito alta, e quando está bom é baixa. Note que para fazer essas medidas é preciso que o circuito esteja desligado.

A medição de corrente é feita de forma um pouco diferente. Precisamos escolher a escala mais adequada, assim como nas medidas de tensão e resistência, mas as pontas de prova devem ser colocadas em série com o fio por onde passa a corrente a ser medida. Em muitos casos é preciso cortar e desencapar o fio para fazer a medida, e soldar e isolar o corte posteriormente. Como é uma operação trabalhosa, devemos fazê-la apenas em caso de necessidade. O multímetro da figura 15 possui uma entrada para medir volts, ohms e Hertz (este mede também frequência), uma outra entrada para medir miliampères e outra para correntes de até 10 ampères. Alguns multímetros podem ainda medir transistores para verificar se estão bons ou queimados.

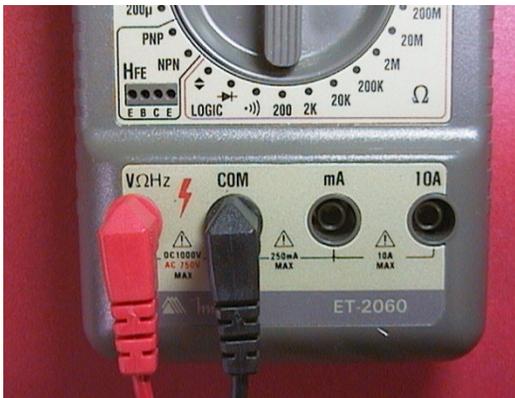


Figura 15

Os multímetros possuem entradas adicionais para medir altas tensões e altas correntes.

Alguns componentes eletrônicos

Vamos agora apresentar alguns componentes eletrônicos e suas propriedades elétricas. Não serão conhecimentos suficientes para você projetar e consertar circuitos complexos, como monitores e fontes, mas ainda assim poderão ajudar.

Bateria e fonte de alimentação

Nenhum circuito elétrico ou eletrônico pode funcionar sem um gerador de corrente elétrica. Os geradores nada mais são que baterias, pilhas ou fontes de alimentação. Possuem dois terminais, sendo um positivo e um negativo. O terminal positivo é aquele por onde “sai” a corrente, e o negativo é aquele por onde “entra” a corrente.

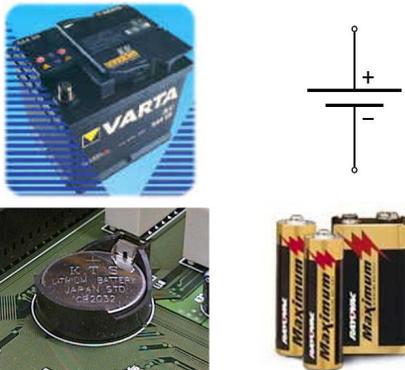


Figura 16

Baterias e o seu símbolo.

A figura 17 mostra o diagrama de um circuito de uma lanterna, no qual temos uma lâmpada alimentada por uma bateria. A corrente elétrica sai do terminal positivo da bateria e trafega através do fio. Chegando à lâmpada, a energia elétrica é transformada em energia luminosa e calor. Depois de atravessar a lâmpada, a corrente retorna à bateria através do seu terminal negativo. Uma bateria é na verdade um dispositivo que empurra a corrente elétrica através dos fios ligados aos seus terminais.

Toda bateria tem uma voltagem especificada. As pilhas, por exemplo, têm 1,5 volt. Também são bastante populares as baterias de 9 volts. Hoje em dia encontramos vários tipos de baterias com diversas voltagens, inclusive recarregáveis. É o caso das baterias de telefones celulares, por exemplo.

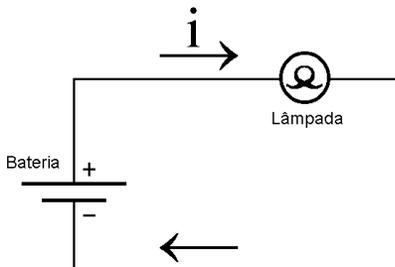


Figura 17

Esquema elétrico de uma lanterna. A letra “i” é usada para designar a corrente elétrica.

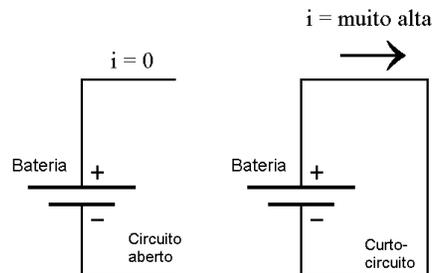


Figura 18

Circuito aberto e curto-circuito.

Em operação normal, uma bateria deve ter circuitos ligados aos seus terminais. A corrente elétrica faz com que esses circuitos funcionem. Por exemplo, se o circuito consistir em uma simples lâmpada, o funcionamento é caracterizado pelo acendimento desta lâmpada. É o que chamamos de *circuito fechado*. Uma bateria pode também estar desligada. Neste caso, existe tensão entre seus terminais, porém não existe corrente. A bateria não está portanto fornecendo energia elétrica ao circuito. É o que ocorre quando temos uma bateria isolada, fora do circuito, ou então quando o interruptor (ou chave) está desligado. Chamamos esta situação de *circuito aberto*.

Uma situação anormal é o chamado curto-circuito. Temos um fio ligando diretamente os dois terminais da bateria. A corrente atravessa o fio, porém como não existe circuito para alimentar, esta corrente tem enorme facilidade para tráfegar. Isto faz a corrente atingir um valor altíssimo, e gerando muito aquecimento. O fio pode até mesmo derreter e pegar fogo, a bateria pode esquentar até ser danificada. Para proteger equipamentos de curto-circuitos acidentais, usamos fusíveis. Se você ligar os dois terminais de uma pilha através de um fio, o curto circuito não será muito perigoso, mas se ligar os dois terminais de uma tomada elétrica, pode até provocar um incêndio.

A figura 18 mostra as características de uma bateria em aberto e outra em curto. Na bateria em aberto, a tensão entre os terminais é igual à tensão da bateria (vamos chamá-la de V_0), e a corrente vale 0. Quando a bateria está em curto, a tensão entre os terminais vale 0, e a corrente assume um valor elevadíssimo. Usando componentes teóricos, a corrente tenderia a ser infinita. Na prática isto não ocorre, mas atinge um valor alto, dependendo das características da bateria.

A fonte de alimentação é um circuito que tem a mesma função de uma bateria. Ela recebe a tensão da rede elétrica e realiza várias operações: redução, retificação, filtragem e regulação. O resultado é uma tensão contínua, semelhante à fornecida por baterias. Mais adiante neste capítulo mostraremos como uma fonte de alimentação realiza este processo.

Resistor

Este é o mais básico componente eletrônico. Muitos o chamam erradamente de *resistência*. Seu nome certo é resistor, e a resistência é a sua característica elétrica. Ainda assim o público leigo usa termos como “a resistência do chuveiro elétrico”, “resistência do aquecedor”, “resistência do ferro de passar”. Esses dispositivos são resistores formados por fios metálicos com resistência baixa. Ao serem ligados em uma tensão elétrica, são atravessados por uma elevada corrente, resultando em grande dissipação de calor. Note que nas resistências desses aparelhos, o objetivo principal é a geração de calor. Já nos circuitos eletrônicos, suas funções são outras, e não gerar calor. Os resistores usados nesses circuitos devem ter valores tais que possam fazer o seu trabalho com a menor geração de calor possível.

Os resistores usados nos circuitos eletrônicos são de vários tipos e tamanhos. Seus dois parâmetros elétricos importantes são a resistência e a potência. Resistores que irão dissipar muita potência elétrica são de maior tamanho, e vice-versa. Os mostrados na figura 19 são de 1/8 W, têm menos de 2 mm de largura por 10 mm de comprimento. Existem resistores de 1/4W, 1/2W, 1W, 2W, 5W, 10W e valores ainda mais elevados. A

figura 19 mostra também o símbolo usado para representar o resistor quando desenhamos um diagrama elétrico.

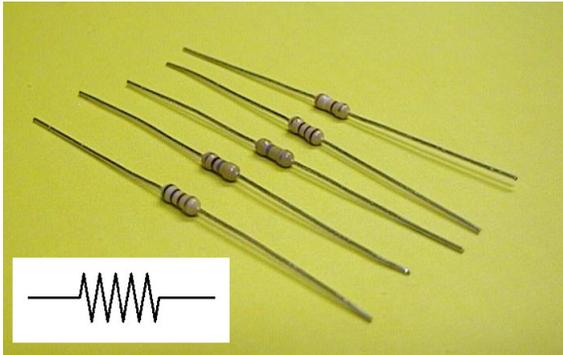


Figura 19
Resistores e o seu símbolo.

Todo resistor tem um valor, que é a chamada *resistência*. A unidade usada para medir a resistência é o *ohm*, cujo símbolo é Ω (a letra grega *ômega* maiúscula). A voltagem gerada por uma bateria tem seu valor dado em *volts*, cujo símbolo é V. A unidade usada para medir a corrente elétrica é o *ampère*, cujo símbolo é A.

Existe uma relação direta entre a tensão aplicada sobre um resistor, a corrente que o atravessa e o valor da sua resistência. Esta relação é a chamada *lei de Ohm*. Ela diz que se um resistor de valor R é ligado a uma tensão V, sua corrente i é dada por:

$$i = V/R$$

é o mesmo que escrever: $V = R \cdot i$

Por exemplo, na figura 20 ligamos uma bateria de 12 V em um resistor de 6Ω . De acordo com a lei de ohm, a corrente que atravessará o resistor será de:

$$i = 12V \div 6\Omega = 2A$$

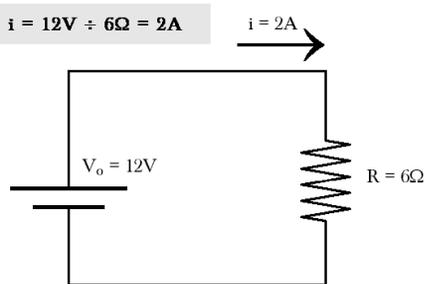


Figura 20
Relação entre corrente, tensão e resistência.

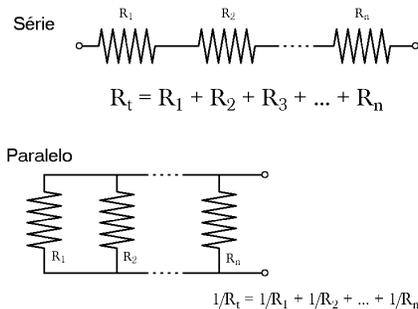


Figura 21
Associações de resistores.

Eventualmente podemos encontrar em circuitos, resistores ligados uns aos outros. Dizemos que os resistores estão *associados*. As duas principais formas de associação de resistores são as do tipo *série* e *paralela*. Ambas são mostradas na figura 21. Quando dois resistores estão em série, a resistência total é igual à soma das resistências de cada resistor. Portanto é calculada pela fórmula:

$$R_t = R_1 + R_2 + R_3 + \dots + R_n$$

Quando os resistores estão associados em paralelo, a fórmula da resistência equivalente é:

$$1/R_t = 1/R_1 + 1/R_2 + 1/R_3 + \dots + 1/R_n$$

Ou seja, o inverso da resistência equivalente é igual à soma dos inversos das resistências individuais.

Outra grandeza elétrica importante é a *potência*. Ela representa a quantidade de energia elétrica que está sendo consumida por um resistor quando é percorrido por uma corrente, e é medida em *watts*, cujo símbolo é *W*. Quando um resistor *R* é ligado a uma tensão *V* e percorrido por uma corrente *i*, a potência elétrica *P* pode ser calculada de várias formas equivalentes:

$$P = V \cdot i$$

$$P = R \cdot i^2$$

$$P = V^2/R$$

Por exemplo, um resistor de 6Ω ligado a uma fonte de 12 V dissipa uma potência de:

$$P = 12^2 / 6 = 144/6 = 24 \text{ watts}$$

É quantidade de calor suficiente para causar uma boa queimadura ao tocarmos neste resistor. Ao contrário do que ocorre na física do segundo grau, não usamos na prática resistores de valores tão baixos, nem operamos com correntes tão elevadas, pelo menos na maioria dos casos. Os resistores usados em eletrônica apresentam em geral resistências da ordem de milhares de ohms, e as correntes elétricas normalmente assumem valores da ordem de milésimos de Ampères.

Capacitor

O capacitor é um componente eletrônico capaz de armazenar cargas elétricas. Ele é formado por duas placas paralelas, separadas por um material isolante, chamado *dielétrico*. Quando o ligamos a uma tensão fixa, momentaneamente passa por ele uma pequena corrente, até que suas placas paralelas fiquem carregadas. Uma fica com cargas negativas (elétrons) e outra com cargas positivas (falta de elétrons).

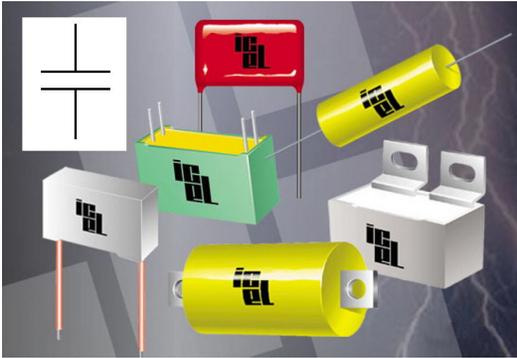


Figura 22

Capacitores e seu símbolo.

Existem vários tipos de capacitores, e as principais diferenças estão nos valores e nas tensões elétricas suportadas. Um capacitor que vai ser ligado a uma tensão de 50 volts deve ser maior que outro de mesmo valor mas que vai ser ligado a uma tensão de apenas 10 volts. Um capacitor sofre ruptura do dielétrico quando é ligado a uma tensão mais elevada que a especificada. Em outras palavras, ele explode!

O valor de um capacitor é chamado de *capacitância*. A grandeza usada para medi-la é o *faraday*, cujo símbolo é F. O faraday é uma unidade muito grande para medir os capacitores da vida real. Um capacitor de 1F seria imenso. Encontramos na prática capacitores medindo algo da ordem de milésimos ou milionésimos do faraday. Por isso é mais comum usar o microfaraday (μF) para medir os capacitores. Um capacitor de 4700 μF , por exemplo, é considerado de tamanho relativamente grande para um circuito eletrônico.

Os capacitores têm várias aplicações nos circuitos eletrônicos. Uma das principais é a *filtragem*. Eles podem acumular uma razoável quantidade de cargas quando estão ligados a uma tensão. Quando esta tensão é desligada, o capacitor é capaz de continuar fornecendo esta mesma tensão durante um pequeno período de tempo, funcionando portanto como uma espécie de bateria de curta duração.

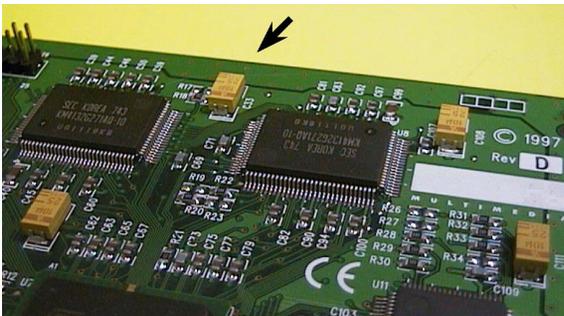


Figura 23

Capacitores de desacoplamento, um ao lado de cada chip.

Em qualquer placa de circuito impresso, encontramos pequenos capacitores ao lado de cada chip. São chamados de *capacitores de desacoplamento* (figura 23). Uma das características elétricas dos chips é que de um instante para outro podem aumentar substancialmente a quantidade de corrente consumida. A fonte de alimentação nem

sempre tem condições de responder ao fornecimento de corrente com a rapidez necessária (em geral, bilionésimos de segundo), e o resultado é uma pequena queda de tensão próxima ao chip que está solicitando este aumento de corrente. O capacitor de desacoplamento tem condições de fornecer rapidamente a corrente elevada que o chip exige, dando tempo à fonte para se adaptar ao novo patamar de corrente. Os capacitores de desacoplamento funcionam portanto como pequenas baterias auxiliares, ajudando a fonte de alimentação no fornecimento de corrente para os chips.

Um capacitor não precisa necessariamente ter placas paralelas e um dielétrico. Qualquer objeto possui uma capacitância. O corpo humano, por exemplo, pode funcionar como um capacitor de baixo valor, mas ainda assim capaz de armazenar cargas elétricas. É o que chamamos de *eletricidade estática*.

Capacitores também têm grandes aplicações em circuitos de rádio. Eles não permitem a passagem da corrente contínua, já que seu dielétrico é um isolante, mas permitem a passagem de tensões alternadas. Como a corrente alternada trafega ora no sentido direto, ora no sentido inverso, um capacitor pode ora se carregar positivamente, ora negativamente, deixando que a corrente alternada o “atravesse”. Quanto mais alta é a frequência da corrente alternada, mais facilmente ela atravessa o capacitor. Eles podem assim ser usados como filtros, barrando as frequências baixas e deixando passar as frequências altas.

Bobina

A bobina é um componente elétrico construído por um fio enrolado em várias voltas. Seu valor é a *indutância*, e a unidade de medida é o *henry* (H). Esta unidade é muito elevada para medir as bobinas da vida real, portanto são mais utilizados o milihenry (mH) e o microhenry (μH). A bobina é atravessada facilmente pela corrente contínua. Corrente alternada de baixa frequência também tem facilidade para atravessar uma bobina, mas quanto maior é a frequência, maior é a dificuldade. Esta característica é inversa à do capacitor. Por isso, associações de capacitores e bobinas são usados para formar filtros de vários tipos, como por exemplo, os sintonizadores. Quando giramos o botão sintonizador de estações de um rádio (DIAL), estamos na verdade atuando sobre um capacitor variável, associado a uma bobina, selecionando a frequência desejada.

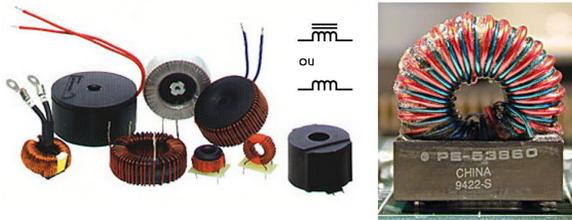


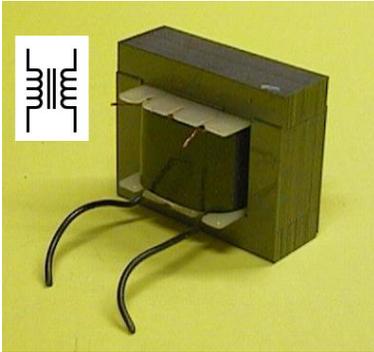
Figura 24

Bobinas e seus símbolos.

Transformador

Quando duas bobinas são enroladas sobre o mesmo núcleo, temos um componente derivado, chamado *transformador*. Cada uma das bobinas é chamada de *enrolamento*. Quando aplicamos uma tensão alternada no primeiro enrolamento (chamado de

primário), podemos retirar uma outra tensão, sendo gerada pelo segundo enrolamento (*secundário*). Isto pode ser usado para aumentar ou reduzir a tensão. Em uma fonte de alimentação convencional (não chaveada), o primeiro circuito é um transformador, que recebe a tensão da rede elétrica (110 ou 220 volts) e gera no secundário uma outra tensão alternada, porém de menor valor.

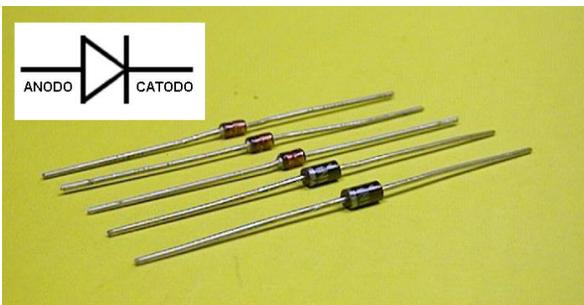
**Figura 25**

Transformador e seu símbolo.

Os transformadores têm muitas outras aplicações. São usados por exemplo como isoladores da linha telefônica em modems. Eles protegem (até certo ponto) o modem de eventuais sobretensões na linha telefônica. Pelo fato de terem uma indutância, eles também atuam como filtros de ruídos.

Diodo

O diodo é um componente classificado como *semicondutor*. Ele é feito dos mesmos materiais que formam os transistores e chips. Este material é baseado no silício. Ao silício são adicionadas substâncias chamadas genericamente de *dopagem* ou *impurezas*. Temos assim trechos tipo N e tipo P. A diferença entre os dois tipos está na forma como os elétrons são conduzidos. Sem entrar em detalhes sobre microeletrônica, o importante aqui é saber que quando temos uma junção PN, a corrente elétrica trafega com facilidade do trecho P para o trecho N, mas não consegue trafegar no sentido inverso. O diodo possui seus dois terminais ligados às partes de uma junção PN. A parte ligada ao P é chamada de *anodo*, e a parte ligada ao N é chamada de *catodo*. A corrente elétrica trafega livremente no sentido do anodo para o catodo, mas não pode trafegar no sentido inverso.

**Figura 26**

Diodos e seu símbolo.

Por causa desta característica, os diodos são usados, entre outras aplicações, como *retificadores*. Eles atuam no processo de transformação de corrente alternada em corrente contínua.

LED

O LED é um tipo especial de diodo que emite luz quando é atravessado por uma corrente elétrica. Como todo diodo, o LED (Light Emitting Diode) permite a passagem de corrente (quando acende) no sentido direto, do anodo para o catodo. No sentido inverso, a corrente não o atravessa, e a luz não é emitida.

Existem LEDs que emitem luz vermelha, verde, amarela e azul. Existem LEDs que emitem luz infravermelha, usados em sistemas de alarmes. Existem ainda os que emitem luz vermelha ou verde, dependendo do sentido da corrente. São na verdade dois LEDs, um vermelho e um verde, ambos montados sobre a mesma base, e ligados em paralelo, um no sentido direto e outro no inverso. Este tipo de LED é usado, por exemplo, em alguns gravadores de CD ou DVD. Quando estão lendo, emitem luz verde ou amarela. Quando estão gravando, emitem luz vermelha.

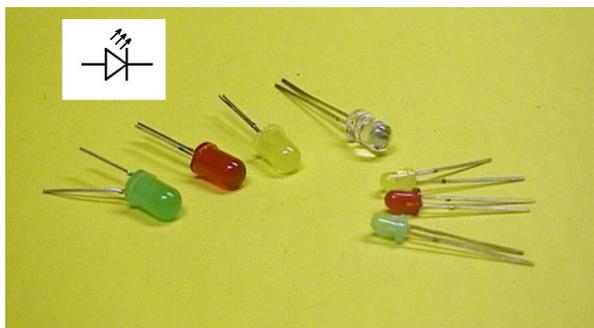


Figura 27

LEDs e seu símbolo.

Transistor

Este é sem dúvida o mais importante componente eletrônico já criado. Ele deu origem aos chips que temos hoje nos computadores. Um processador, por exemplo, tem no seu interior, vários milhões de microscópicos transistores. A figura 28 mostra alguns transistores e seu símbolo eletrônico. Note que existem vários tipos de transistores. Quanto ao sentido da corrente elétrica, os transistores são classificados como NPN e PNP, ambos mostrados na figura 28.

Os transistores realizam inúmeras funções, sendo que as mais importantes são como amplificadores de tensão e amplificadores de corrente. Por exemplo, o sinal elétrico gerado por um microfone é tão fraco que não tem condições de gerar som quando é aplicado a um alto falante. Usamos então um transistor para elevar a tensão do sinal sonoro, de alguns milésimos de volts até alguns volts. Seria tensão suficiente para alimentar um alto falante, mas ainda sem condições de fornecer a potência adequada (a tensão está correta mas a corrente é baixa). Usamos então um segundo transistor atuando como amplificador de corrente. Teremos então a tensão igual à gerada pelo primeiro transistor, mas com maior capacidade de fornecer corrente.

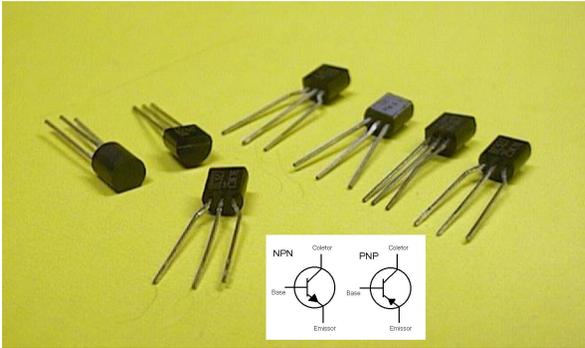


Figura 28
Transistores e seus símbolos.

Os aumentos de tensão e de corrente são no fundo, aumentos de energia. Esta energia não é gerada a partir do nada. O transistor retira a energia necessária a partir de uma bateria ou fonte de alimentação. A figura 29 mostra o diagrama do circuito simples, com dois transistores, para amplificar o sinal gerado por um microfone para que seja aplicado em um alto falante. Note que os transistores não trabalham sozinhos. Eles precisam ser acompanhados de resistores, capacitores, e dependendo do circuito, outros componentes, para realizar suas funções.

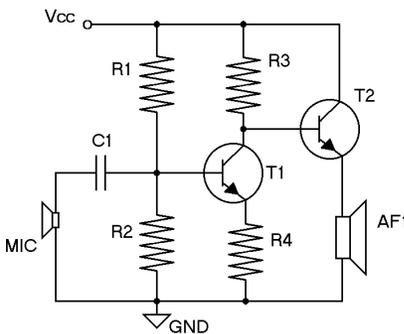


Figura 29

Amplificador transistorizado:

- MIC = Microfone
- AF1 = Alto falante
- R1, R2, R3 E R4 = resistores
- T1, T2 = transistores
- VCC = Terminal positivo da bateria
- GND = Terra, ou terminal negativo da bateria.

Existem transistores de baixa, média e alta potência. Quanto maior é a potência, maior é o seu tamanho. Os transistores de alta potência em geral precisam ser montados sobre dissipadores de calor (coolers). Existem transistores especializados em operar com frequências de áudio e outros especializados em altas frequências, usados em circuitos de rádio e TV. Existem transistores especializados em chaveamento, indicados para operar em circuitos digitais. Existem fototransistores, que amplificam o sinal gerado pelo seu sensor óptico. Enfim, vários outros tipos de transistores, para as mais variadas aplicações.

Regulador de voltagem

Todos os circuitos eletrônicos necessitam, para que funcionem corretamente, do fornecimento de corrente vinda de uma bateria ou fonte de alimentação com valor constante. Por exemplo, se um circuito foi projetado para funcionar com 5 volts, talvez possa funcionar com tensões um pouco maiores ou um pouco menores, como 5,5 V ou 4,5 V, mas provavelmente não funcionará corretamente com valores muito mais altos

ou muito mais baixos, como 6 V ou 4 V. Uma fonte de alimentação precisa portanto gerar uma tensão constante, independente de flutuações na rede elétrica e independente da quantidade de corrente que os circuitos exijam. Por isso, todas as fontes de boa qualidade utilizam circuitos reguladores de voltagem.

É possível criar um regulador de voltagem utilizando alguns transistores, resistores e um componente especial chamado *diódo Zener*, capaz de gerar uma tensão fixa de referência a ser “imitada” pela fonte. Os fabricantes construíram esses circuitos de forma integrada, semelhante a um chip, usando uma única base de silício. Os reguladores mais simples têm um encapsulamento parecido com o de um transistor de potência, com três terminais. Um dos terminais é o terra, que deve ser ligado ao terminal negativo da fonte. O outro terminal é a entrada, onde deve ser aplicada a tensão bruta, não regulada. O terceiro terminal é a saída, por onde é fornecida a tensão regulada. A tensão de entrada deve ser superior à tensão que vai ser gerada. O regulador “corta” uma parte desta tensão de modo a manter na saída uma tensão fixa. Por exemplo, para alimentar um regulador de +5 Volts, podemos aplicar na entrada uma tensão não regulada de +8 Volts, podendo variar entre +6 e +10. A saída fornecerá +5 V, e o restante será desprezado.

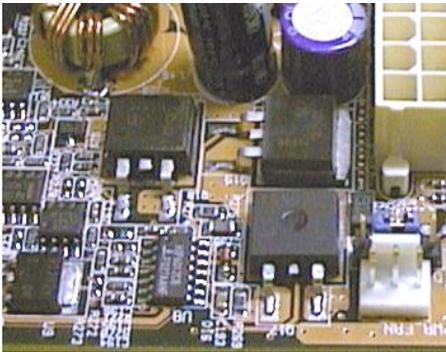


Figura 30

Reguladores de voltagem em uma placa mãe.

Muitos reguladores produzem tensões fixas, mas existem modelos que podem ser ligados a uma tensão de referência que pode ser programada. Nas placas mãe existe um circuito responsável por gerar as tensões exigidas pelo processador. A maioria dos processadores modernos requer uma fonte de +3,3 V para operações externas, e uma fonte de valor menor para as operações internas. Dependendo do processador, esta tensão pode ser de +1,3 V, +1,6V, +1,7V, +2,1V ou praticamente qualquer valor entre 1 V e 3,5 V. O circuito gerador de voltagem da placa mãe toma como base a tensão de +5V ou +12V fornecida pela fonte de alimentação do computador, e em função do valor indicado pelo processador, gera a tensão necessária. Trata-se de um regulador de tensão variável e programável. Note que esse tipo de circuito não vai simplesmente “cortar” a tensão deixando para o processador apenas o valor solicitado. Ele primeiro converte a tensão de +12 volts para um valor menor, em torno de 2 volts (mas com alta corrente), e finalmente corta o valor desejado, usando um regulador de voltagem.

Jumpers e microchaves

São dispositivos bastante comuns na maioria das placas, principalmente nas placas mãe. Vejamos inicialmente o que é um jumper. Eletricamente funciona como um minúsculo interruptor. Quando está instalado, permite a passagem de corrente. Quando é retirado, impede a passagem de corrente.

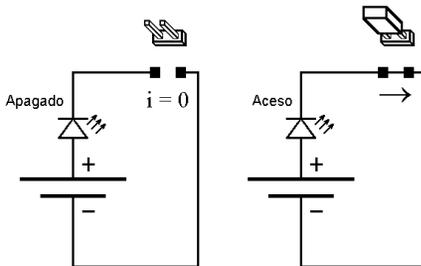


Figura 31

Funcionamento de um jumper.

O jumper é uma pequena peça plástica com dois orifícios metalizados e ligados internamente. Devem ser encaixados em pinos metálicos instalados nas placas. Quando o jumper está encaixado, a corrente o atravessa, passando de um pino para outro. Quando é retirado, a passagem de corrente fica desabilitada. A figura 31 mostra o acionamento de um LED através de um jumper. Note que o jumper é equivalente a uma chave ou interruptor. Fisicamente o jumper é a pequena peça, como as mostradas na figura 32.

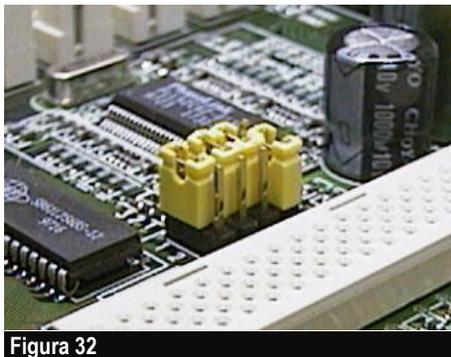


Figura 32

Jumpers.

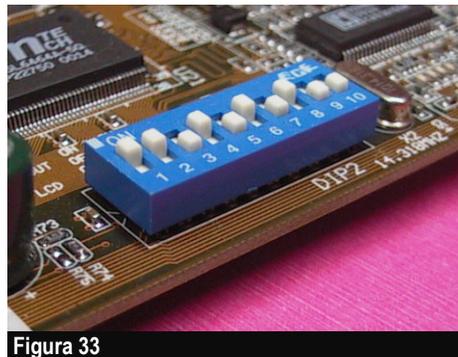


Figura 33

Microchaves.

A microchave ou dip switch é um dispositivo que desempenha a mesma função que o jumper. A diferença é que seu formato é similar ao de um chip. Além disso, as microchaves são apresentadas em grupos, em geral de 4 a 12 chaves. Cada chave pode ser posicionada nas posições ON e OFF, o que equivale a configurações do tipo *com jumper* e *sem jumper*, respectivamente.

Os jumpers e microchaves possuem várias aplicações. Nas placas mãe, servem para habilitar e desabilitar o funcionamento da bateria, selecionar o tipo e a velocidade das memórias, a velocidade e a tensão do processador, entre varias outras funções.

Voltagens e bits

Uma das características mais importantes dos circuitos digitais é a representação dos bits 0 e 1 através de dois valores de tensão. Em geral é usado um valor pequeno, entre 0 e 0,3 volts, para indicar o bit 0, e um valor um pouco maior, da ordem de alguns poucos volts, para indicar o bit 1. Por exemplo, chips de memória DDR usam cerca de 0,2 V para representar o bit 0 e em torno de 2,4 V para representar o bit “1”. Valores diferentes podem ser usados, dependendo da tecnologia. Por exemplo, no interior dos processadores modernos, os níveis de tensão são ainda mais baixos. São usados internamente valores em torno de 1,0 a 1,5 volts para representar o bit 1, e um valor sempre próximo de 0 V para representar o bit 0. Seja qual for o caso, o nível de tensão que representa o bit 0 será sempre um valor positivo, apesar de muito pequeno. Da mesma forma, o nível de tensão que representa o bit 1 será sempre um valor um pouco menor que a tensão da fonte de alimentação. A maioria dos chips existentes nas placas modernas opera com alimentação de 3,3 volts, mas muitos já operam com apenas 2,5 volts. Há alguns anos atrás a maioria dos chips operavam com 5 volts.

Teoricamente quaisquer níveis de voltagem poderiam ser usados para representar os bits 0 e 1. Na prática são usados valores pequenos, para que o consumo de energia seja pequeno. Valores maiores podem ser encontrados em alguns circuitos. Por exemplo, em um relógio despertador digital alimentado por uma bateria de 9 volts, o bit 1 pode ser representado por um valor superior a 8 volts, e o bit 0 por um valor menor, próximo de 0 volt.

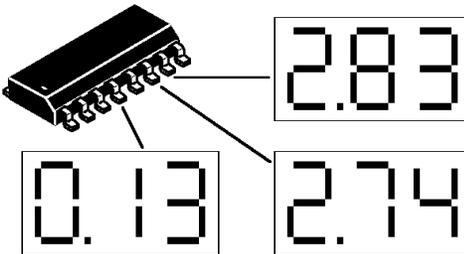


Figura 34

Medindo as tensões que representam os bits em um chip alimentado por 3,3 volts.

A figura 34 mostra uma medida teórica das tensões em pinos de um chip, representando bits 0 e 1. O pino que apresenta a tensão de 0,13 volts corresponde a um bit 0. Os outros dois pinos indicados, com tensões de 2,83V e 2,74V representam bits 1. Os valores de tensão que representam os bits podem variar sensivelmente de um chip para outro, ou mesmo de um pino para outro. Não existe um valor exato, e sim, uma faixa de valores.

Na prática esta medida nem sempre pode ser feita com um multímetro. Quando um chip está trabalhando, seus bits estão variando rapidamente, entre 0 e 1. Um multímetro não é capaz de medir tensões variáveis em alta velocidade, é adequado a medir apenas tensões constantes. Supondo que este chip esteja fornecendo bits constantes, mediríamos valores como os da figura 34. Em alguns casos um chip pode realmente apresentar valores constantes. Por exemplo, o chip que contém a interface de impressora pode transmitir bits variáveis enquanto está sendo produzida uma listagem, mas ao terminar, pode manter fixo em suas saídas o código binário do último dado

enviado para a impressora. Neste ponto poderíamos fazer uma medida usando o multímetro, como a apresentada na figura 34.

Tristate ou alta impedância

Quando um circuito digital está em operação normal, pode gerar na sua saída, tensões correspondentes aos bits 0 e 1. Existe entretanto um terceiro estado no qual um circuito pode operar. É o chamado *tristate* ou *alta impedância*. Em inglês são usados também os termos *high impedance* ou *float* (flutuar). É como se o circuito estivesse desconectado. Imagine por exemplo dois módulos de memória, cada um encaixado em seu respectivo soquete. Digamos que cada um desses módulos tenha 256 MB. Quando o processador acessa um endereço de memória entre 0 e 256 MB, o primeiro módulo está ativo e o segundo fica em tristate. Quando é acessado um endereço superior a 256 MB, e até 512 MB, o segundo módulo estará ativo e o primeiro estará em tristate. O uso do terceiro estado é necessário para que dois ou mais circuitos possam operar ligados ao mesmo ponto, ou ao mesmo barramento, porém apenas um de cada vez deverá entregar seus bits, e os demais devem ficar como se estivessem desligados.

Existem vários exemplos de uso do tristate (terceiro estado). Um deles é o uso de várias placas de expansão estão conectadas no barramento PCI de uma placa mãe, onde todas podem transmitir dados através do seu slot, porém não podem fazer transmissões no mesmo instante. No instante em que uma placa envia dados (ou que o processador comanda uma leitura dos seus dados), as demais placas mantêm suas saídas em tristate.

A maioria dos chips tem a capacidade de entrar em tristate. Eles possuem um pino (ou seja, uma “perninha”) chamado CS, ou *chip select*. Quando este sinal está ativado, o chip está em uso normal. Quando este sinal é desativado, o chip entra em tristate.

Diagramas de tempo

Como mencionamos, os bits representados pelos circuitos digitais variam bastante ao longo do tempo. Por exemplo, em um moderno chip de memória, os bits podem variar até 800 milhões de vezes a cada segundo (é o caso da DDR2/800), ora representando 0, ora representando 1. Um diagrama de tempo é um gráfico simplificado que mostra os valores dos bits ao longo do tempo, como o que vemos na figura 35.

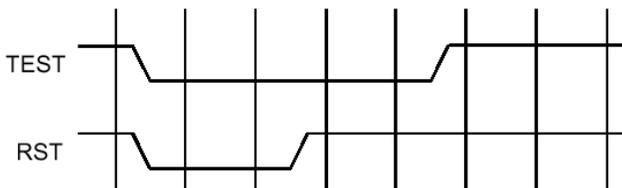


Figura 35

Diagrama de tempo.

Um diagrama de tempo pode representar um ou vários sinais digitais simultaneamente. Neste caso é usado um único eixo Y, representando a tensão (ou os bits 0 e 1), e vários eixos X independentes, cada um deles representando um sinal digital diferente ao longo do tempo. Cada sinal digital por sua vez assume valores 0 e 1 ao longo do tempo. O diagrama da figura 35 representa dois sinais digitais. Neste diagrama podemos

observar, além dos trechos nos quais o circuito gera bits 0 e 1, um pequeno intervalo de tempo em cada transição de 1 para 0 ou de 0 para 1, representados por trechos inclinados do gráfico. Esta transição deveria ser instantânea, do ponto de vista matemático, mas na prática leva um certo tempo, bastante pequeno. Por exemplo, um chip de memória que gera bits diferentes a cada 10 ns (10 bilionésimos de segundo) pode demorar entre 1 e 2 ns para mudar seu estado de 0 para 1 ou de 1 para 0.

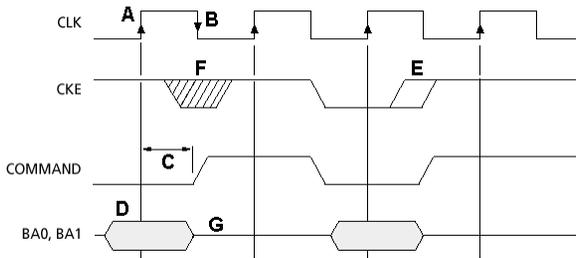


Figura 36

Convenções usadas em um diagrama de tempo.

A figura 36 mostra alguns símbolos de eventos encontrados em diagramas de tempo:

a) Trigger positivo

Este símbolo indica que no instante em que um sinal digital sofre uma transição de 0 para 1, um evento ou mudança em outro sinal digital será ativado.

b) Trigger negativo

Similar ao positivo, exceto que o evento é disparado na transição binária de 1 para 0.

c) Retardo entre dois sinais

Mostra a dependência temporal entre dois sinais relacionados. É usado quando é informação relevante saber que um determinado sinal será ativado depois de um determinado tempo a partir do qual o primeiro é ativado.

d) Indicação de barramento

Para evitar que um diagrama fique muito extenso, podemos agrupar vários sinais relacionados em um único eixo. Usamos para representar, por exemplo, o barramento de dados do processador ou memória, o barramento de endereços, o conjunto de dados que estão trafegando através de uma interface. Não existe interesse em especificar o valor individual de cada um dos sinais digitais. Eles formam um grupo, e alguns deles podem ser 1 e outros serem 0, e o circuito funcionará independentemente dos valores.

e) Mudança de estado em ponto indeterminado

Todos os circuitos digitais apresentam pequenas variações, mas os fabricantes sempre especificam valores máximos e mínimos. Por exemplo, um determinado circuito pode apresentar um tempo médio de resposta de 15 ns, mas alguns componentes podem chegar a 10 ns, outros a 20 ns. Em certos casos o projetista precisa compatibilizar seu circuito com componentes mais lentos e mais rápidos. Neste caso precisa levar em conta o primeiro instante e o último instante em que um sinal digital pode ser ativado.

f) Don't care

Significa “não importa”. O sinal digital poderá ter neste período, qualquer valor (obviamente, 0 ou 1), sem afetar o funcionamento do circuito. Por exemplo, se fizermos o diagrama da transmissão de dados por uma interface paralela, este diagrama deve começar indicando o dado que estava presente nas saídas da interface antes de começar a nova transmissão. Neste caso, não importa o dado que existia antes. Fazemos então a sua indicação como “don't care”.

g) Tristate

Este símbolo é usado para representar períodos de tempo nos quais um sinal digital encontra-se em tristate (terceiro estado, ou alta impedância).

Como exercício você poderá agora fazer o download de manuais de chips, memórias e processadores, encontrados nos sites dos seus fabricantes, e observar os diagramas de tempo mostrados. Poderá então entender melhor o funcionamento de vários desses chips.

OBS: Quando um sinal tem valor 1 quando está em repouso e valor 0 quando está ativo, dizemos que é um sinal de *lógica negativa*. Sinais com esta característica são indicados com um traço horizontal sobre o seu nome, ou então com um símbolo “#” à sua direita, ou um “n” à sua esquerda. Por exemplo, se um sinal RESET é ativo em 0, indicamos como $\overline{\text{RESET}}$ ou nRESET.

Microeletrônica

A microeletrônica consiste em projetar e produzir circuitos utilizando componentes de tamanho microscópico. Usando materiais e técnicas apropriadas, é possível construir transistores, resistores, capacitores, diodos e indutores, cada um deles com tamanhos menores que 1 milésimo de milímetro. Os componentes tradicionais, comprados em forma avulsa no comércio e usados nos circuitos de som, rádio e TV, são chamados de *componentes discretos*. Um *circuito integrado* ou *chip* é um circuito complexo porém de tamanho reduzido. É equivalente ao circuito de uma placa com componentes discretos, mas pelo fato de utilizar componentes integrados microscópicos, seu tamanho total é da ordem de 1 centímetro quadrado, ou mesmo menor.

Os componentes de um chip são como se fossem “pintados” na sua minúscula base, chamada *substrato*. O seu processo de fabricação é entretanto bem mais complexo que uma simples pintura. Trata-se de um processo um pouco químico, um pouco fotográfico, uma difusão de moléculas dentro da base de silício, formando camadas que compõem os circuitos.

A maioria dos materiais são divididos em duas categorias: *condutores* e *isolantes*. O condutor é um material que tem facilidade em conduzir corrente elétrica. Todos os metais são condutores. Já os isolantes são materiais que dificultam a passagem da corrente elétrica. A borracha é um exemplo típico de isolante, assim como o vidro, madeira, plásticos em geral, etc. Existem entretanto alguns materiais que ora se comportam como condutores, ora como isolantes. São os chamados *semicondutores*, e os principais deles são o *silício* e o *germânio*. A maioria dos transistores e chips utilizam o silício em sua fabricação. O germânio é utilizado em alguns componentes especiais, como transistores para altas frequências.

Para que os semicondutores possam variar sua resistividade, é preciso que lhe sejam adicionados materiais especiais, chamados de *dopagem*. Existem dopagens tipos N e P (negativa e positiva), e a sua combinação é usada na formação dos transistores, diodos e demais circuitos no interior de um chip.

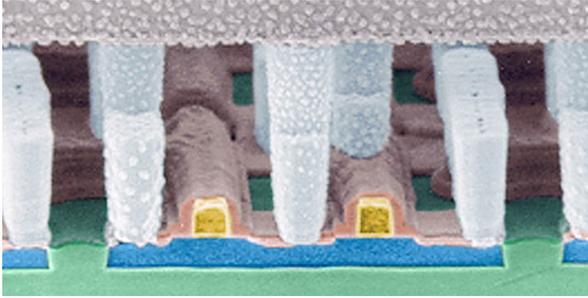


Figura 37

Foto ampliada do corte transversal de um microscópico trecho de um chip. A parte mostrada mede alguns poucos milésimos de milímetro.

Um chip é formado por sucessivas camadas de materiais diferentes. A base na qual um chip é construído (substrato) é feita de silício puro, ou seja, sem dopagem. Sobre esta base são aplicadas dopagens sucessivas, formando trechos tipos N e P. Eventuais ligações são feitas com camadas de alumínio ou cobre. Em certos trechos também são usadas camadas de óxidos como isolantes.

A figura 38 ilustra o funcionamento de um tipo especial de transistor usado para formar os chips. Trata-se do transistor MOS. É formado por camadas de metal, óxido e semicondutor (Metal Oxide Semiconductor). Este transistor possui três terminais, chamados de *source*, *drain* e *gate*. O terminal source é ligado à tensão positiva da fonte, através de um resistor. Ele é a saída do circuito, que pode representar bits 0 ou 1. O terminal drain é ligado ao terra, ou seja, o pólo negativo da bateria. O terminal de entrada é o gate, e é usado para controlar a corrente que passa entre source e drain.

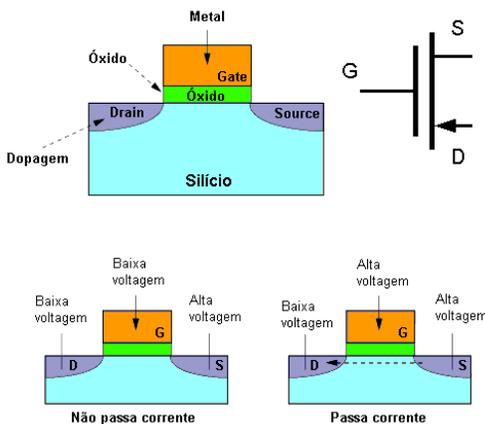


Figura 38

Funcionamento de um transistor MOS e seu símbolo.

Vejam como funciona o circuito da figura 39, formado por um transistor e um resistor. Quando o gate é ligado a uma tensão baixa (bit 0), não passa corrente entre source e drain. Sendo assim, o source terá uma tensão elevada (bit 1), já que fica ligado

ao polo positivo da bateria, através de um resistor. Quando o gate é ligado a uma tensão alta (bit 1), passará uma corrente entre source e drain. A resistência entre esses dois pontos será baixa, e a tensão medida no source será próxima de 0 volt. Teremos assim um bit 0 em sua saída. O circuito formado por este transistor e um resistor é o que chamamos de *inversor*. A operação lógica que realiza é a inversão de bits. Ao ser aplicado um bit 1 na sua entrada, produzirá um bit 0 na saída. Ao ser aplicado um bit 0 na entrada, produzirá um bit 1 na saída.

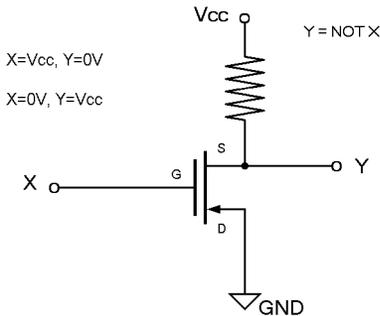


Figura 39

Inversor MOS.

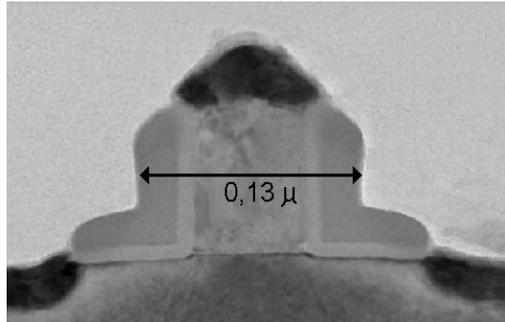


Figura 40

Foto ampliada de um transistor com 0,13 μ .

Um fator bastante importante é a medida dos microscópicos transistores que formam os chips, como os mostrados na figura 40. Com o passar dos anos, dimensões cada vez menores têm sido utilizadas. A unidade usada para medir esses transistores é o micron (símbolo μ). Cada micron é equivalente a um milésimo de milímetro. Outra unidade usada é o nanômetro (nm), que vale um milionésimo de milímetro. No início de 2007, a Intel já anunciava a produção de chips com tecnologia de 45 nm. Usar transistores menores significa:

- Menor voltagem
- Menor dissipação de calor
- Menor custo de produção

A tabela que se segue mostra as tecnologias de fabricação de chips dos últimos anos:

Ano	Tecnologia	Voltagem
1989	1 μ	5 V
1991	0,8 μ	5 V
1993	0,5 μ	3,3 V
1995	0,35 μ	2,5 V
1997	0,25 μ	1,8 V
1999	0,18 μ	1,5 V
2001	0,13 μ	1,3 V
2003	0,09 μ (90 nm)	1,3 V
2005	0,065 μ (65 nm)	1,2 V
2007	0,045 μ (45 nm)	1,2 V

Não apenas transistores podem ser construídos através de microeletrônica. Pequenos trechos de semicondutores podem formar resistores. Placas paralelas de metal formam capacitores, e trilhas de metal dispostas em forma espiral formam bobinas. Chips usados em telecomunicações utilizam no seu interior, bobinas e capacitores, além dos transistores e resistores. Chips usados em eletrônica digital (processadores, memórias, chipsets, etc) em geral apresentam apenas transistores e alguns resistores.

Os chips são produzidos em grandes pastilhas circulares de 20 ou 30 cm de diâmetro chamadas *waffers*. A indústria tem trabalhado durante os últimos anos com waffers de 20 cm, e apenas em 2001 começaram a ser adotados os waffers de 30 cm, com várias vantagens. Em cada waffer são construídas dezenas ou centenas de chips, como vemos na figura 41. Depois de prontos os chips são separados um dos outros através de corte. São testados e finalmente encapsulados.



Figura 41

Vários chips em um waffer.

O processo de *encapsulamento* consiste em alojar a pastilha do chip em uma carcaça externa, que pode ser de plástico ou cerâmica. Também é feita a ligação dos seus pontos de contato nos terminais externos (as “perninhas” do chip).

CMOS

Os circuitos integrados digitais devem ter o menor número possível de resistores. Esses componentes, mesmo no interior dos chips, ocupam áreas muito maiores que os transistores. Além disso produzem maior dissipação de calor e retardos que tornam os chips mais lentos. Por isso os projetistas tentam na medida do possível usar os próprios transistores para substituir os resistores. Daí surgiram os circuitos CMOS (Complementary Metal Oxide Semiconductor). Consiste em utilizar no circuito da figura 39, um segundo transistor no lugar do resistor. Este segundo transistor possui características inversas às do primeiro. São chamados *transistores complementares*. Um transistor é do tipo NMOS, e o outro é tipo PMOS. Quando um transistor conduz, o outro não conduz, e vice-versa. O resultado é o mesmo obtido com o uso do resistor, porém ocupando muito menos espaço, consumindo menos energia e com mais velocidade. O arranjo completo é mostrado na figura 42.

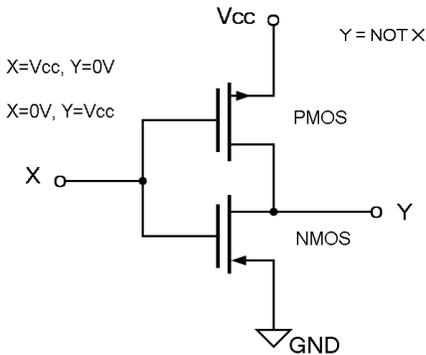


Figura 42

Circuito equivalente de uma célula CMOS.

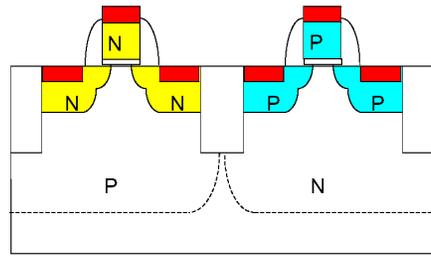


Figura 43

Camadas que formam o par CMOS.

Este circuito é o *inversor*, o mais simples dos operadores lógicos. Ele gera um bit 1 quando recebe um bit 0, e gera um bit 0 quando recebe um bit 1. Outras funções lógicas mais complexas são implementadas com arranjos parecidos. Observe que ambos os transistores possuem seus terminais *gate* interligados. Quando esta entrada recebe um bit 1, ou seja, um nível de tensão elevado, o transistor inferior conduzirá corrente, e o superior ficará *cortado*, ou seja, sem conduzir. Isto fará com que a saída fique com tensão baixa, ou seja, um bit 0. Quando a entrada receber um bit 0, o transistor inferior ficará cortado, sem conduzir, e o transistor superior irá conduzir, fazendo com que sua saída fique com uma tensão quase igual à da fonte de alimentação (bit 1). A figura 43 mostra como o par CMOS é construído em um chip.

A maioria dos chips modernos utilizam a tecnologia CMOS. Existem outras tecnologias que são utilizadas em aplicações nas quais o CMOS não pode ser aplicado. Por exemplo, os pares CMOS não são indicados quando é necessário fornecer correntes elevadas, como por exemplo, para alimentar os slots de um barramento. Nesses casos são usados circuitos lógicos TTL (Transistor-Transistor Logic), que consomem mais energia, mas também podem fornecer mais corrente. Muitos chips utilizam internamente células CMOS e externamente apresentam entradas e saídas TTL.

OBS: Muitas pessoas ouvem falar em CMOS pela primeira vez ao tomarem contato com o chamado *CMOS Setup* de placas mãe. Acabam conhecendo o "*chip CMOS*", no qual existe uma pequena área de memória para armazenar configurações do BIOS da placa mãe, além de um relógio permanente. O "*chip CMOS*" é alimentado por uma bateria que o mantém em funcionamento mesmo quando o computador está desligado. Aqui está um fato curioso: praticamente todos os chips do computador utilizam a tecnologia CMOS. É errado pensar que apenas o popular "*chip CMOS*" que armazena os dados do Setup e tem o relógio permanente utiliza esta tecnologia.

Circuitos lógicos

Toda a eletrônica digital é desenvolvida a partir da criação de circuitos capazes de executar *operações lógicas*, também chamadas de *operações booleanas*. Os três principais operadores lógicos são:

E (AND)

Ou (OR)

Não (NOT)

A partir desses operadores, circuitos ainda mais complexos são construídos:

Somadores e Subtratores

Multiplicadores e divisores

Células de memória

Registradores, multiplexadores, decodificadores

etc...

A reunião desses circuitos complexos forma chips bastante sofisticados, como processadores, memórias, chips gráficos, chipsets, etc. Parece incrível que equipamentos tão sofisticados possam ser construídos a partir de circuitos básicos tão simples.

Um operador lógico é algo que lembra um pouco um operador aritmético. Na aritmética temos operadores para Adição, Subtração, etc. Da mesma forma como na aritmética temos, por exemplo:

$$5 + 2 = 7$$

na lógica temos

$$1 \text{ AND } 1 = 1$$

$$1 \text{ OR } 0 = 1$$

$$\text{NOT } 1 = 0$$

Inicialmente, vejamos como funcionam os três operadores citados. Cada um deles pode ser definido através da sua *tabela verdade*. A seguir temos essas tabelas:

A	NOT A
0	1
1	0

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Como vemos na tabela, o operador NOT, também chamado de *inversor*, produz na sua saída o bit inverso daquele recebido na entrada. O operador AND possui duas entradas. Sua saída será 1 apenas quando as duas entradas também forem 1, simultaneamente. Já o operador OR produz uma saída 1 quando pelo menos uma das suas entradas tem o valor 1.

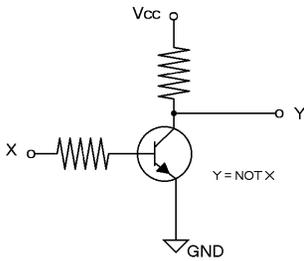


Figura 44

Inversor RTL.

É relativamente fácil produzir circuitos que realizam essas funções, usando transistores, resistores e outros componentes. O circuito mostrado na figura 44 implementa o operador lógico NOT. É formado a partir de um transistor e dois resistores. Este método de construção de circuitos é chamado RTL (Resistor-Transistor Logic). Optamos por mostrar circuitos RTL, e não CMOS, porque seu funcionamento é mais simples de entender, e também porque podem ser montados por alunos em laboratórios de eletrônica, utilizando resistores e transistores comuns. Use para os resistores ligados nas base dos transistores, 10 kΩ, e para os demais, 1 kΩ. Para tensão de alimentação, use 5 volts. Pode ser usado qualquer transistor de uso geral, como BC548 ou equivalente.

O funcionamento do inversor mostrado na figura 44 é bastante simples. Quando X é um bit 1, a tensão correspondente é um valor alto (porém menor que Vcc, a tensão da fonte de alimentação). Este valor alto faz com que exista uma corrente na base do transistor, que irá conduzir uma corrente elevada entre seus outros terminais. Ao mesmo tempo aparecerá uma baixa tensão (da ordem de 0,3 volts, dependendo do transistor) no seu coletor, que é a saída Y. Temos então um bit 0 na saída. Da mesma forma, quando X é um bit 0, a tensão na entrada do transistor será baixa. O transistor ficará então “cortado”, e praticamente não passará corrente por ele. A tensão na saída Y dependerá apenas do resistor ligado ao ponto Vcc. Teremos assim uma tensão alta em Y, o que corresponde a um bit 1.

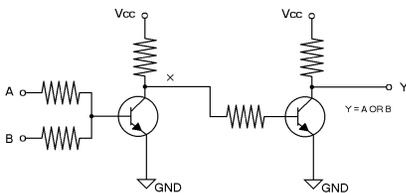


Figura 45

Circuito OR RTL.

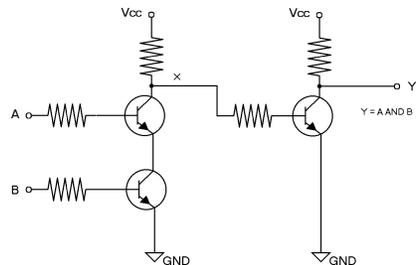


Figura 46

Circuito AND RTL.

A figura 45 mostra como é implementado o operador OR usando a lógica RTL (use os mesmos valores de resistores e tensão citados para a figura 44). O primeiro transistor vai conduzir corrente quando pelo menos uma das duas entradas, A ou B, estiver com

tensão alta (bit 1), ficando assim com um nível 0 no ponto X. Apenas quando ambas as entradas A e B estiverem em 0, o primeiro transistor ficará cortado e teremos um bit 1 no ponto X. Ora, este é exatamente o inverso da função OR. Temos portanto no ponto X um outro operador lógico chamado NOR (ou NOT OR), cuja tabela verdade é:

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Para que o circuito final tenha uma saída OR, e não NOR, temos que usar mais um inversor, representado pelo segundo transistor e seus dois resistores.

A figura 46 mostra o circuito que implementa um operador lógico AND, usando a técnica RTL. O primeiro estágio é formado por dois transistores, sendo que cada um deles tem ligada na sua base, uma das entradas (A ou B) do circuito. Para ter o valor 0 no ponto X é preciso que ambos os transistores estejam conduzindo, o que é conseguido apenas quando ambas as entradas A e B estão em 1. Se uma ou ambas as entradas estiver com o valor 0, o transistor correspondente estará cortado, e não passará corrente através de ambos. Isto fará com que o ponto X fique com o valor 1.

Esta é exatamente a função inversa do AND, chamada NAND. Sua tabela verdade é:

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Para que tenhamos na saída do circuito uma função AND, é preciso inverter o sinal presente no ponto X, para isso utilizamos mais um inversor, representado pelo terceiro transistor e seus resistores.

Circuitos lógicos como NOT, AND, OR, NAND, NOR e outros operadores, podem ser construídos utilizando várias técnicas. Mostramos aqui o método RTL, porém existem outras formas de criar circuitos equivalentes, como:

DTL: Diode-Transistor Logic

ECL: Emitter Coupled Logic

TTL: Transistor-Transistor Logic

CMOS: Complementary Metal Oxide Semiconductor Logic

As técnicas mais utilizadas são a TTL, para chips mais simples, e CMOS para chips mais complexos.

Operadores lógicos

Quando projetamos ou analisamos circuitos lógicos, não nos preocupamos com detalhes internos, como seus transistores, diodos e resistores. Levamos em conta apenas as entradas e saídas. Nos diagramas de circuitos digitais, desenhamos apenas os símbolos dos circuitos que implementam as funções lógicas. Chamamos esses circuitos de *portas lógicas*. A figura 47 mostra os símbolos das principais portas lógicas.

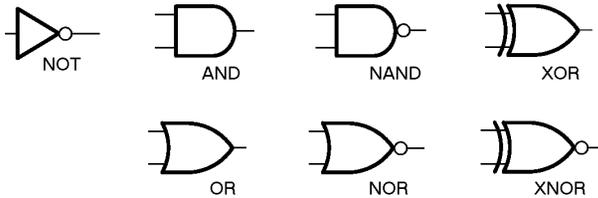


Figura 47
Símbolos das portas lógicas.

Nesta mesma figura apresentamos também as portas lógicas XOR (eXclusive OR – “ou exclusivo”) e XNOR (eXclusive NOR). A função XOR tem uma tabela verdade bastante parecida com a da função OR. Seu significado é o seguinte: o bit de saída será ligado se um dos bits de entrada estiver ligado, mas não ambos ao mesmo tempo. Portanto a única diferença entre as funções OR e XOR é que:

- 1 OR 1 = 1
- 1 XOR 1 = 0

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

Mostramos também acima a tabela verdade do operador XNOR, que é o inverso do operador XOR. Observe que a função XNOR funciona como um comparador. Seu resultado é 1 quando os dois bits de entrada são iguais, e 0 quando os dois bits de entrada são diferentes.

Circuitos lógicos complexos

A construção de circuitos lógicos complexos é uma simples questão de agrupar essas portas básicas, produzindo funções mais elaboradas. O exemplo da figura 48 mostra o circuito de um comparador binário. Este circuito faz a comparação de dois valores binários de 4 bits cada um. A saída do circuito será 1 quando os dois valores binários de 4 bits presentes nas entradas forem iguais. Digamos que esses valores sejam representados por $A_3A_2A_1A_0$ e $B_3B_2B_1B_0$. A saída Y do circuito será ativada em 1 quando tivermos iguais esses valores. Por exemplo $A=0110$ e $B=0110$. Este tipo de circuito é muito utilizado como decodificador de endereços nas placas mãe e nas placas de expansão. Os valores do endereço A podem ser originados no barramento de

endereços do processador, e os valores de B são originados em um grupo de microchaves ou jumpers, que dependendo da forma como são configurados, podem indicar bits 0 ou 1. O circuito comparador irá ativar sua saída em 1 quando o endereço recebido for igual ao endereço definido pelas microchaves ou jumpers. Obviamente para isto é necessário um comparador maior, operando com maior número de bits, mas seu princípio de funcionamento é o mesmo.

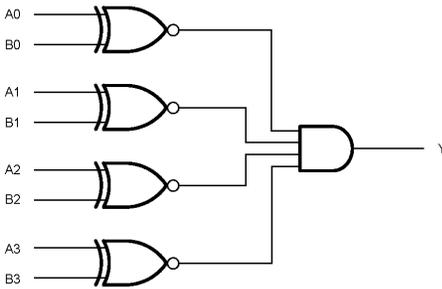


Figura 48

Comparador de 4 bits.

Utilizando um número maior de portas lógicas, podemos formar circuitos mais complexos. A figura 49 mostra o circuito de um contador binário de 4 bits. Este circuito recebe um sinal de clock e gera nas suas 4 saídas, números binários na seqüência 0000, 0001, 0010, etc. Pode ser programado para contar no modo decimal, ou seja, passando de 9 (1001) para 0 (0000), ou então no formato hexadecimal, passando de F (1111) para 0 (0000). Gera ainda um bit de “vai 1” e pode ser agrupado com outros circuitos iguais, formando assim contadores com qualquer número de dígitos. Pode ainda ser programado para fazer contagem crescente ou decrescente.

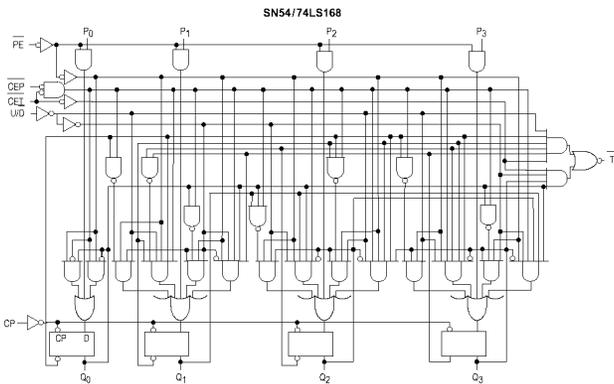


Figura 49

Contador binário.

Um projetista de hardware pode obter circuitos digitais de várias formas. A mais simples é utilizando chips padrões de mercado, que normalmente apresentam encapsulamentos como os da figura 50. Os encapsulamentos mostrados na figura são o DIP (Dual In-Line Package) e SOIC (Small Outline Integrated Circuit). Existem circuitos com portas AND, OR, NOR, NAND, inversores, e funções complexas mas de uso comum, como decodificadores, comparadores, contadores, registradores, etc.

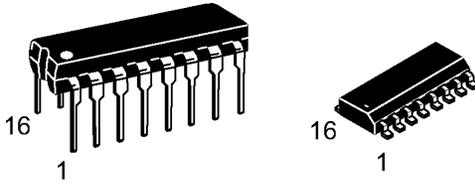


Figura 50

Chips com encapsulamento DIP plástico e SOIC.

Nos manuais dos chips que contém circuitos lógicos básicos, encontramos diagramas que indicam o que existe no seu interior, como nos exemplos da figura 51. Os chips deste exemplo têm o seguinte conteúdo:

- 4 portas NAND de 2 entradas
- 3 portas AND de 3 entradas
- 4 portas AND de 2 entradas
- 2 portas NAND de 4 entradas
- 4 portas XOR de 2 entradas
- 1 porta NAND de 8 entradas
- 4 portas NOR de 2 entradas
- 4 portas OR de 2 entradas
- 6 inversores (portas NOT)

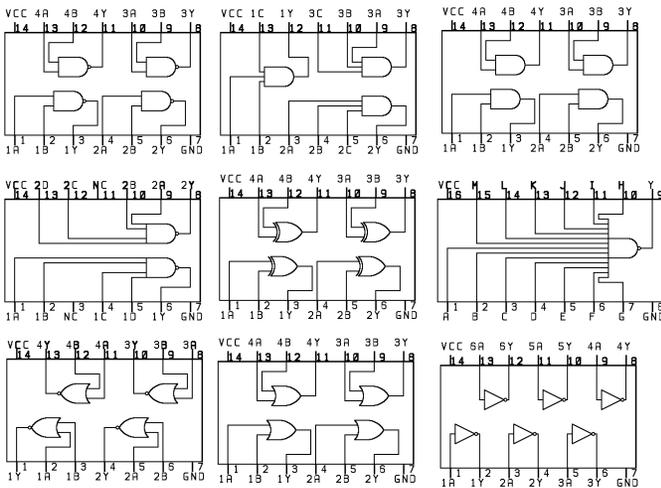


Figura 51

Diagramas de alguns chips TTL.

Ao projetar um circuito digital, usamos inicialmente as portas necessárias para implementar a função desejada. Depois contamos quantas portas de cada tipo são necessárias. Escolhemos os chips apropriados que contenham as portas desejadas, e finalmente realizamos as ligações entre os pinos desses chips.

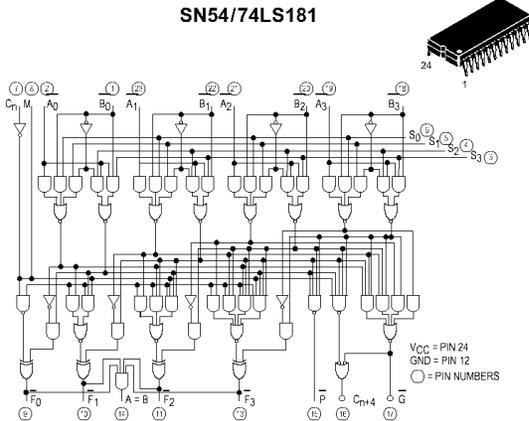
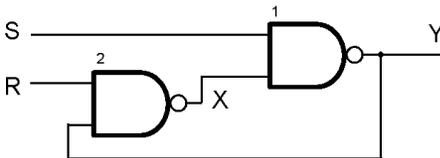
**Figura 52**

Diagrama interno do chip 74LS181 – unidade lógica e aritmética de 4 bits.

A figura 52 mostra o diagrama interno do chip 74LS181. Este chip é uma unidade lógica e aritmética de 4 bits, capaz de realizar 16 operações lógicas e aritméticas, entre adição, subtração, AND, OR, etc. Vários chips desses podem ser ligados em cascata para formar unidades com maior número de bits. Este chip tem pouco mais de 60 portas lógicas. Em um processador existem vários milhões de portas lógicas, executando entre outras, funções como as deste chip, porém com maior número de bits.

Célula de memória construída com portas lógicas

Para construir um computador, não basta utilizar operadores lógicos e aritméticos. É preciso também ter memória, uma característica fundamental dos circuitos digitais. Células de memória podem ser facilmente construídas a partir do diagrama básico mostrado na figura 53. Este circuito é chamado de FLIP FLOP.

**Figura 53**

Célula de memória.

Suas duas entradas R e S devem permanecer com valores 1. Para armazenar o valor 1 na célula, basta aplicar momentaneamente um bit 0 na entrada S (Set). Para armazenar o valor 0 na célula, basta aplicar momentaneamente um bit 0 na entrada R (Reset). Vejamos como isto ocorre, detalhadamente.

a) Suponha que as entradas estejam em repouso, ou seja, $R=1$ e $S=1$.

b) Aplicamos momentaneamente um bit 0 em S. A porta NAND ligada em S, ao receber 0 nesta entrada, produzirá uma saída $Y=1$ (lembre-se da tabela verdade da função NAND: se pelo menos uma das entradas é 0, a saída é 1).

c) A porta 2 está então recebendo as entradas $R=1$ e $Y=1$ (note que a saída Y do circuito funciona como entrada da porta 2). Como $1 \text{ NAND } 1 = 0$, teremos uma saída $X=0$ na saída da porta 2. Este zero, ao entrar na porta 1, continuará produzindo saída $Y=1$, e agora isto independe do valor de S, já que $0 \text{ NAND } 0 = 1$ e $0 \text{ NAND } 1 = 1$.

d) Agora a entrada S pode voltar ao seu valor de repouso 1, e a saída Y continuará sendo mantida em 1. Temos então um bit 1 armazenado.

Da mesma forma, o circuito também pode armazenar um bit 0, bastando manter S em 1, e momentaneamente levando a entrada R ao valor 0.

É um circuito extremamente simples, mas é realmente uma surpresa a sua capacidade de “lembrar” um bit. Circuitos como este são agrupados até formar células de memória com muitos bits. Milhões dessas células são encontradas em um chip de memória, formando vários megabytes.

Fonte de alimentação linear

A fonte de alimentação é um dispositivo que tem a mesma função que uma bateria. A diferença é que a energia elétrica não fica armazenada em células de voltagem (como ocorre com pilhas e baterias), e sim, é extraída da rede elétrica. Muitos aparelhos são alimentados diretamente a partir da rede elétrica, como é o caso de lâmpadas e motores. A voltagem da rede elétrica não é adequada para aparelhos eletrônicos, portanto esses aparelhos possuem fontes de alimentação. São circuitos que convertem a tensão da rede elétrica (110 volts em corrente alternada) para tensões adequadas ao seu funcionamento (em geral inferiores a 20 volts, em corrente contínua).

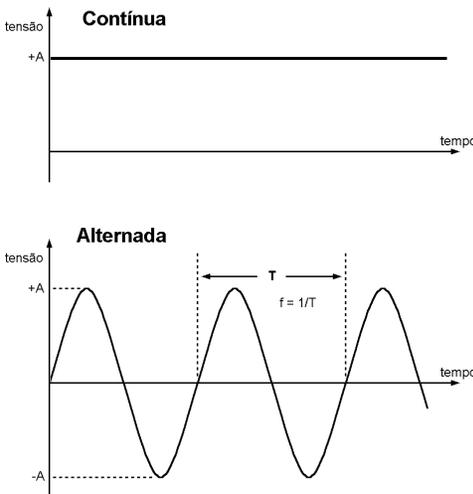


Figura 54

Tensão contínua e tensão alternada:

A = Amplitude
f = Frequência
T = Período

A figura 54 mostra a diferença entre uma fonte de tensão contínua e uma alternada. Na fonte de tensão contínua (CC), a corrente trafega sempre no mesmo sentido. O valor da tensão é constante, e se ligarmos um circuito de características constantes, como lâmpadas e resistores, a corrente também será constante. Como já mostramos, existem dois terminais, o positivo e o negativo. Na fonte de corrente alternada (CA), a corrente trafega, ora em um sentido, ora em outro sentido. A fonte CA empurra e puxa a corrente, indefinidamente.

A rede elétrica usada no Brasil opera com 60 ciclos por segundo, ou seja, empurra a corrente, depois puxa a corrente, e repete este ciclo 60 vezes a cada segundo. Dizemos que a tensão da rede é 60 Hz. Em alguns países, sobretudo na Europa, a rede opera com 50 Hz. O gráfico da tensão alternada tem a forma de uma senóide porque a geração é feita por eixos rotativos, existentes nos geradores das usinas de energia. Uma vantagem da tensão alternada é que pode ser facilmente convertida em valores mais altos ou mais baixos, através de transformadores.

Uma fonte de alimentação recebe tensão alternada a partir da rede elétrica, com frequência de 60 Hz e voltagem que pode ser de 110 ou 220 volts. Inicialmente esta tensão é reduzida para um valor menor, através de um transformador. Temos então corrente alternada, mas com um valor menor. A seguir é feita uma retificação, que consiste em fazer a corrente trafegar sempre no mesmo sentido. Isso é feito com *diódos*. O próximo passo é a filtragem (feita com *capacitores*), e finalmente a regulação (através de *diodo zener* e *regulador de tensão*). A figura 55 mostra as etapas da geração de tensão contínua em uma fonte.

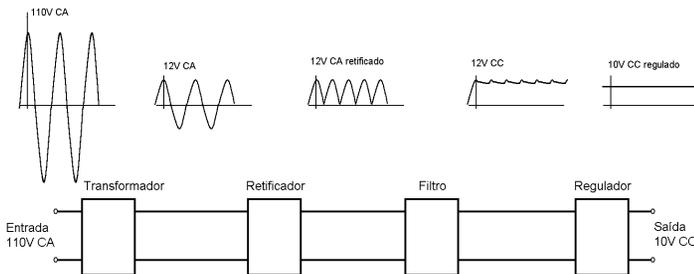


Figura 55

Operação de uma fonte linear.

As fontes que operam como mostramos na figura 55 são as chamadas “fontes lineares”. Sua principal desvantagem é que requerem transformadores muito grandes e pesados para fazer a redução de voltagem, e capacitores muito grandes para fazer a filtragem. São adequadas quando a potência a ser fornecida (potência = tensão x corrente) é pequena. Os chamados “adaptadores AC”, usados para alimentar caixas de som e dispositivos que não possuem fonte própria, consomem pouca potência. Eles são na verdade fontes lineares de alimentação, com operação similar ao mostrado na figura 55. Adaptadores AC extremamente leves e compactos são em geral formados por uma pequena fonte chaveada, e não do tipo linear.

Fonte de alimentação chaveada

Tanto os transformadores quanto os capacitores usados nas fontes de alimentação poderiam ser bem menores se a frequência da rede elétrica fosse mais elevada, ao invés de operar com apenas 60 Hz. Por isso foram criadas as fontes chaveadas, utilizadas nos PCs e em todos os equipamentos eletrônicos modernos. Elas não necessitam de transformadores e capacitores grandes, e por isso podem fornecer muita potência, porém mantendo peso e tamanho reduzidos.

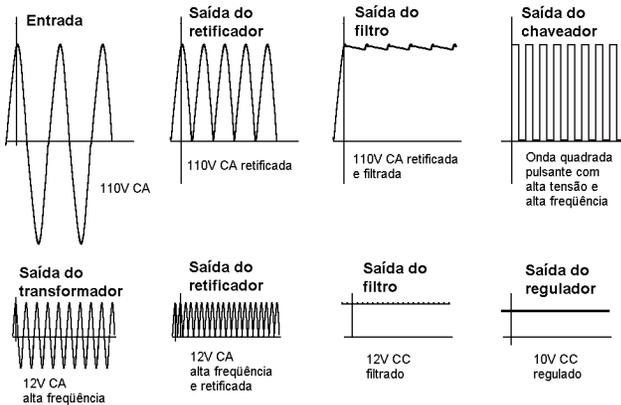


Figura 56

Operação de uma fonte chaveada.

A figura 56 mostra as etapas de funcionamento de uma fonte chaveada. Inicialmente a tensão da rede elétrica é retificada e filtrada. Não existe dificuldade técnica na retificação de tensões elevadas. Quanto à filtragem, podem ser usados capacitores de menor valor, pois a corrente é mais baixa, apesar da tensão ser elevada. O resultado é uma tensão contínua de valor elevado. Esta tensão passa por um transistor de chaveamento que a transforma em uma onda quadrada de alta frequência, entre 100 e 200 kHz. Este transistor opera como uma chave elétrica que abre e fecha o circuito para a passagem de corrente, em alta velocidade. Esta onda quadrada passa por um transformador e tem sua tensão reduzida, porém com valor de corrente maior. Este transformador pode ser pequeno, já que opera com frequência muito mais elevada, e quanto maior é a frequência, maior é a facilidade que um transformador tem para fazer o seu trabalho.

Temos então uma corrente alternada, mas com amplitude menor e frequência maior. Esta corrente é retificada e filtrada, desta vez usando capacitores de menor tamanho, já que a filtragem também é facilitada pela frequência elevada. Finalmente temos a etapa de regulação, na qual imperfeições são eliminadas, resultando em um valor constante na saída. Uma fonte de alimentação usada em um PC possui várias seções para a geração dos diversos valores de voltagem: +5V, +12V, +3,3V, etc.

Barramentos do processador

O processador é o principal responsável pelo desempenho de um PC. Todos os processadores usados nos PCs são descendentes do 8086, o primeiro processador de 16