

Capítulo 13

Barramentos de E/S e interfaces

Os barramentos PCI, AGP e ISA fazem parte da placa de CPU. Todos eles têm uma característica em comum: servem para conectar placas. Os barramentos que mostraremos neste capítulo são ditos de entrada e saída. Servem para conectar periféricos, e não placas. A maioria deles é externa ao computador, ou seja, existem conectores do computador, normalmente disponíveis na sua parte traseira, nos quais ligamos diretamente os periféricos. Nem todos são assim, como são os casos dos barramentos IDE (abordado neste capítulo) e SCSI.

Barramento USB

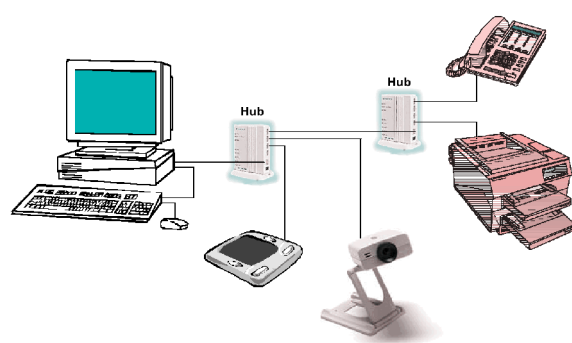
O USB (Universal Serial Bus) existe desde meados dos anos 90 e foi provido principalmente pela Intel. No final dos anos 90 o USB finalmente começou a conquistar espaço no mercado de PCs. Serve para conectar de forma extremamente simples, eficiente e com bom desempenho, vários tipos de periféricos que antes eram espalhados por uma miscelânea de interfaces. Alguns exemplos de periféricos que podem ser ligados no barramento USB são:

Teclado	Alto falante	Drive de disquetes
Mouse	Joystick	Disco rígido / CD-ROM
Impressora	Câmera digital	ZIP Drive
Scanner	WebCam	Modem
Microfone	Tablet	Gravadores de CDs

Existe uma tendência de nos próximos anos todos esses periféricos deixarem de ser produzidos nas suas interfaces tradicionais e passem a utilizar a interface USB. Hoje todos eles são disponíveis tanto com interface USB como com as interfaces tradicionais.

O cabo USB possui 4 fios, sendo 2 para transmissão de dados e 2 para alimentação. Desta forma os dispositivos podem obter a alimentação diretamente do cabo (+5 volts) e não precisam de fonte própria. Dispositivos USB que exigem correntes elevadas podem utilizar suas fontes de alimentação próprias.

A maioria dos PCs modernos possui duas interfaces USB. Em cada uma delas podemos conectar até 127 dispositivos. Para que vários dispositivos possam ser conectados na mesma porta, é preciso utilizar um hub (figura 1). Cada hub permite ligar vários dispositivos, e para aumentar o seu número, podemos ligar os hubs uns nos outros.

**FIGURA 13.1**

Conexão de dispositivos USB.

A figura 2 mostra os conectores USB encontrados na parte traseira de uma placa de CPU. Normalmente encontramos nas placas modernas, duas portas USB. Algumas possuem 4 portas, e através da instalação de placas de expansão, podemos aumentar ainda mais o número de portas. Em cada porta podem ser ligados até 127 dispositivos usando hubs, mas quando queremos ligar apenas um ou dois dispositivos, não precisamos de hub. Podemos ligá-los diretamente nas portas.

**FIGURA 13.2**

Conectores USB na parte traseira de uma placa de CPU ATX.

Normalmente as interfaces USB ficam localizadas no SouthBridge, que é o componente do chipset no qual estão também as interfaces IDE e o controle do barramento ISA ou LPC. Existem chipsets com 2 interfaces USB e outros com 4, como o AMD 766, mostrado no diagrama da figura 3. Neste caso, dois conectores ficam normalmente no painel de conectores existente na parte traseira da placa de CPU, e os outros dois são acessíveis através de conectores auxiliares (figura 4).

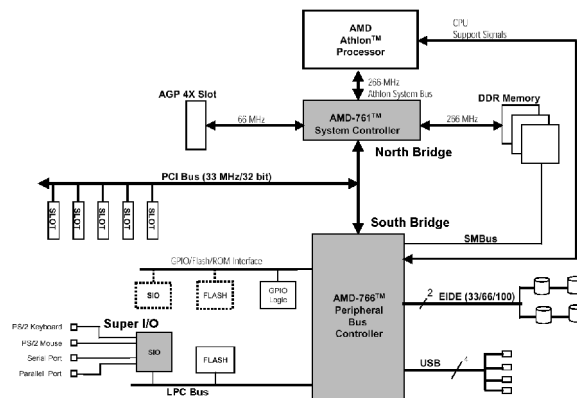
**FIGURA 13.3**

Diagrama de uma placa de CPU contendo 4 portas USB no chipset.

**FIGURA 13.4**

Conectores auxiliares para a 3ª e 4ª portas USB de uma placa de CPU.

Quando um PC não possui portas USB (é o caso típico dos modelos antigos), ou quando possui mas queremos aumentar o seu número, podemos instalar placas de interface USB, como a da figura 5. São ligadas ao barramento ISA e com elas podemos ter portas USB similares às existentes nos chipsets modernos. A placa da figura 5 tem 4 portas.

**FIGURA 13.5**

Placa de interface USB com 4 portas.

Dispositivos USB

Não existe grande diferença no aspecto de dispositivos USB e dispositivos que usam interfaces tradicionais, exceto pelos seus conectores. A figura 6 mostra um mouse USB. Observe o detalhe do seu conector, ampliado à direita.

**FIGURA 13.6**

Mouse USB.

Na figura 7 vemos um teclado USB. Não existe diferença aparente em relação aos teclados comuns, exceto pelo conector.

**FIGURA 13.7**

Teclado USB.

Os dispositivos USB são Plug and Play, portanto são automaticamente detectados pelos sistemas operacionais com este recurso. Algumas deles já possuem drivers nativos no próprio Windows, como é o caso do mouse e do teclado. Esses dois dispositivos têm inclusive suporte pelos BIOS modernos, permitindo que funcionem mesmo no modo MS-DOS. Outros dispositivos são reconhecidos mas requerem a instalação de drivers apropriados.

Características do USB

Cada porta USB permite ligar até 127 dispositivos. Os cabos utilizados nessas conexões devem ter até 5 metros. Além de ser compatível com o padrão Plug and Play, o barramento USB traz ainda o recurso Hot Plugging. Significa que os dispositivos podem ser conectados e desconectados com o computador ligado. No instante em que um dispositivo USB é conectado, ele informa ao computador sobre a sua presença. O sistema operacional o

detecta e ativa seus drivers. O dispositivo passa então a ficar disponível para uso. Portanto podemos destacar as seguintes características:

- Plug and Play
- Hot Plugging
- Até 127 dispositivos por porta
- Cabos de até 5 metros
- Alimentação no barramento dispensa fontes de alimentação

USB 1.1 e USB 2.0

As primeiras interfaces USB atendiam à especificação 1.0. Posteriormente foram introduzidas pequenas modificações que deram origem à especificação 1.1. Essas especificações são seguidas pelos chipsets existentes nas placas de CPU produzidas antes de 2000. Esta especificação suporta duas velocidades:

Modo	Taxa de transferência
Low Speed	1,5 Mbits/s
Full Speed	12 Mbits/s

Placas de CPU novas, produzidas a partir de meados de 2000, já possuem chipsets que dão suporte ao USB 2.0. A principal diferença é uma nova taxa de transferência, de 480 Mbits/s. Esta taxa é bastante elevada, e viabiliza a utilização de periféricos mais velozes, como discos rígidos e câmeras de vídeo de alta resolução. Além disso o USB 2.0 é compatível com o USB 1.1. Interfaces USB 2.0 operam com os mesmos cabos e suportam qualquer periférico originalmente criado para a versão 1.1. O modo de 480 Mbits/s é chamado de High Speed. Portanto as taxas oferecidas pelo USB 2.0 são:

Modo	Taxa de transferência
Low Speed	1,5 Mbits/s
Full Speed	12 Mbits/s
High Speed	480 Mbits/s

Note que essas taxas são medidas em Mbits/s. Para converter para MB/s, temos que dividir por 8. Portanto as taxas de transferência obtidas nesses 3 modos são de cerca de 200 kB/s, 1,5 MB/s e 60 MB/s, respectivamente.

O modo Low Speed é utilizado por dispositivos que não necessitam de altas velocidades, tais como teclado, mouse e joystick. O modo Full Speed (12 Mbits/s) é usado por dispositivos que operam com som e imagem, como câmeras, microfones e alto falantes, e também com os dispositivos para

armazenamento de dados. O modo High Speed é usado por dispositivos que necessitam de elevadas taxas de transferência, como os que manipulam vídeo e operam com armazenamento de dados.

Os cabos certificados para o modo Full Speed do USB 1.1 (12 Mbits/s) funcionam perfeitamente no modo High Speed do USB 2.0. Esta é uma vantagem dos barramentos seriais. Permitem obter taxas de transferência mais elevadas com mais facilidade que nos barramentos paralelos. Portanto ao comprar cabos USB, verifique se na embalagem está especificado USB 2.0 ou se está indicada a taxa de 12 Mbits/s, o que indica que funcionarão também a 480 Mbits/s.

Cabos e conectores USB

Os cabos USB possuem no seu interior, 2 pares de fios. Um par é trançado e formado por fios branco e verde. Através desses dois fios trafegam os dados, no formato serial. Esses dois sinais formam o que chamamos de par diferencial. Ao invés de ter um fio de terra e outro de sinal, os dois levam o mesmo sinal, mas com polaridades invertidas, portanto são chamados de D+ (verde) e D- (branco). A vantagem do par diferencial é a alta imunidade a ruídos elétricos e interferências em geral.

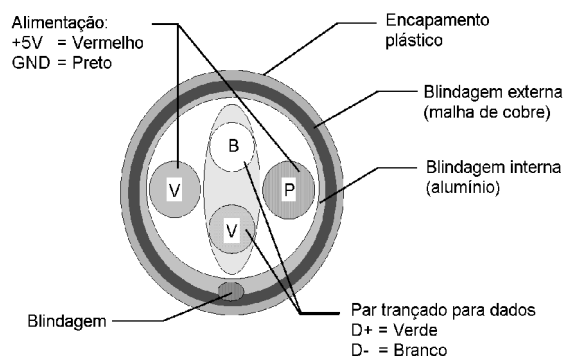
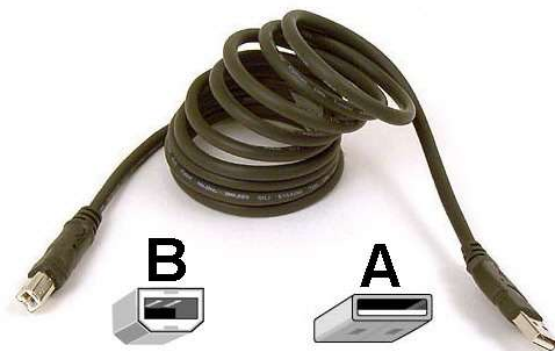


FIGURA 13.8

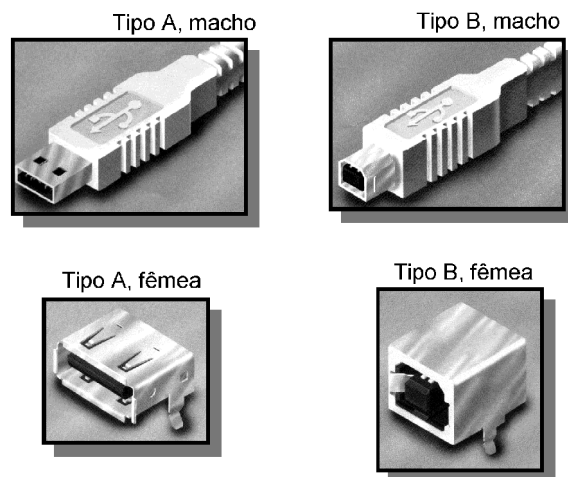
Corte transversal de um cabo USB.

O outro par é usado para alimentação dos dispositivos ligados ao barramento. O fio preto é o terra e o vermelho traz uma tensão de +5 volts. Envoltendo esses dois pares de fios temos uma camada formada por uma folha de alumínio e uma blindagem externa, formada por uma malha de cobre. Envoltendo tudo temos o encapsulamento plástico.

**FIGURA 13.9**

Cabo USB com conectores tipos A e B.

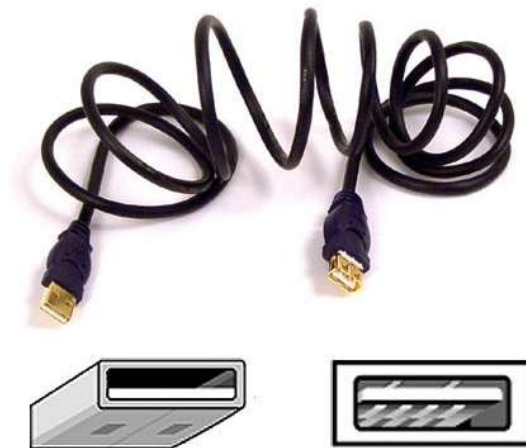
Os cabos USB sempre possuem dois conectores, que podem ser dos tipos A e B, como vemos na figura 9. O conector tipo A fica sempre voltado para a direção do computador, e o tipo B voltado para a direção do periférico. Na figura 10 vemos esses dois conectores em detalhes. Os conectores machos são usados no cabo. Os conectores fêmea são usados no computador, nos hubs e nos periféricos.

**FIGURA 13.10**

Detalhes dos conectores tipos A e B.

O cabo mostrado na figura 9 é o mais comum, do tipo AB. Podemos ainda encontrar cabos do tipo AA, que servem como extensões. Digamos que o cabo AB com o qual estamos tentando ligar uma impressora ao computador é muito curto, e não alcança o computador. Podemos então ligar neste cabo AB, um cabo AA. O conector A macho do cabo AB deverá ser ligado no conector fêmea do cabo AA. Ficamos então com a seqüência

A-A-A-B, eletricamente igual a um cabo AB de maior comprimento. Isto pode ser feito desde que o comprimento total do cabo não seja superior a 5 metros.

**FIGURA 13.11**

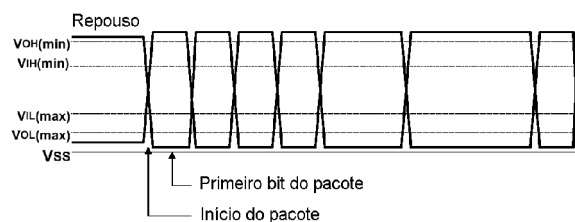
Cabo USB tipo AA.

Sinais do barramento

É bastante interessante o funcionamento do par diferencial. Quando desejamos transmitir uma voltagem V , enviamos V pelo fio positivo e $-V$ pelo fio negativo. Ao longo do caminho, digamos que ambos recebem uma interferência i , passando a ficar com valores $V+i$ e $-V+i$. Ao chegar ao destino, um amplificador diferencial calculará a diferença desses dois sinais e dividirá o resultado por 2. Ficará então com:

$$[(V+i) - (-V+i)]/2 = V$$

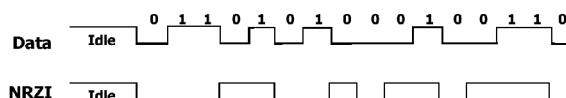
Não importa o valor da interferência i , ela será cancelada e o valor obtido na saída do amplificador receptor será V , o mesmo valor transmitido originalmente. Pares diferenciais são utilizados por várias interfaces que necessitam operar com alta velocidade por longas distâncias e com alta imunidade a interferências.

**FIGURA 13.12**

Voltagens ao longo do cabo USB.

A figura 12 mostra o aspecto das voltagens no par diferencial (D+ e D-) em um cabo USB. As tensões são sempre simétricas, uma positiva e uma negativa, com amplitudes entre 200 mV e 500 mV.

Essas tensões são geradas de acordo com os bits que devem ser transmitidos. Esquecendo agora as tensões e pensando nos bits, vejamos o gráfico da figura 13. Os dados a serem enviados pelo cabo, formando uma seqüência de bits, passam antes por um processo de codificação chamado NRZI (Non Return do Zero Invert). A cada bit que desejamos enviar, o método NRZI envia 0 quando o próximo bit é diferente do anterior, e envia 1 quando o próximo bit é igual ao anterior. Na figura 13, o trecho superior indica os valores dos bits que trafegam pelo cabo USB, e o trecho inferior indica os dados que realmente estão sendo transmitidos.

**FIGURA 13.13**

Codificação NRZI em um cabo USB.

Os dados trafegam pelo barramento USB em grupos chamados pacotes. Cada pacote tem um byte inicial chamado PID (Packet Identification), seguido de informação e um campo de CRC para checagem de erros. Os pacotes de dados no USB 1.1 podem ter até 1024 bytes, e no USB 2.0 podem ter até 8192 bytes.

Field	PID	DATA	CRC16
Bits	8	0-8192	16

FIGURA 13.14

Um pacote de dados no barramento USB.

Conexões entre o PC e dispositivos USB

Quando precisamos ligar muitos dispositivos USB em um computador, devemos utilizar um hub USB, como o mostrado na figura 15. O hub deste

exemplo tem 4 saídas. Note que existem 4 conectores tipo A, nos quais podemos ligar 4 dispositivos, ou então ligar outros hubs. Existe ainda um conector tipo B que deve ser ligado através de um cabo A/B em uma porta USB do computador, ou no conector A de outro hub.

**FIGURA 13.15**

Um hub USB.

Note que as 4 portas do hub mostrado na figura 15 não são 4 interfaces. As interfaces são as existentes no computador. Os conectores de um hub fazem parte de uma mesma interface, ou seja, não é permitido ligar 127 dispositivos em cada uma das portas do hub. O número total de dispositivos deve ser contado partindo do computador, somando todos os dispositivos ligados em uma interface e nos hubs ligados a partir dela. Observe que o hub da figura 15 é acompanhado de um adaptador AC, ou seja, ele tem alimentação própria. Um hub também pode funcionar a partir da tensão de +5 volts fornecida pelo cabo, dispensando o adaptador AC. O problema é que quando ligamos muitos dispositivos, a corrente exigida pode ser muito elevada e o barramento não dará conta. Usar o adaptador AC resolve o problema.

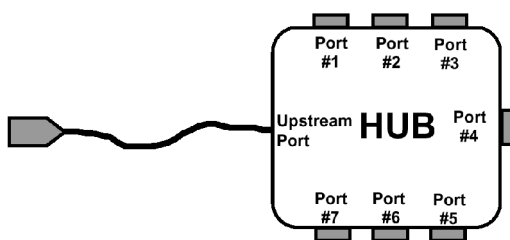
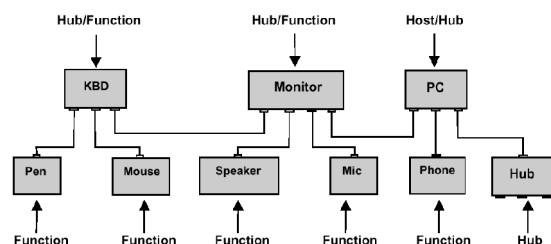
**FIGURA 13.16**

Diagrama simplificado de um hub USB.

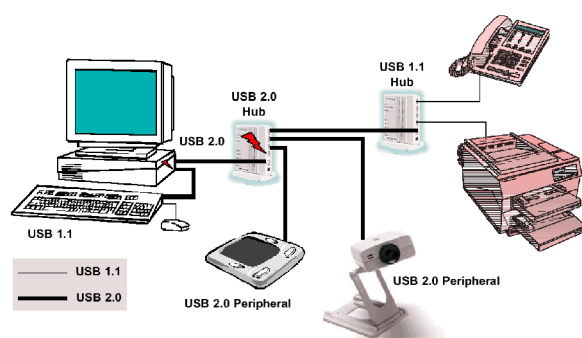
A figura 16 mostra o diagrama simplificado de um hub USB de 8 portas. Existem dispositivos que possuem hubs embutidos. Por exemplo, podemos encontrar monitores com conexão USB. Esses monitores possuem o cabo normal para ligação na placa de vídeo, mas possuem também uma conexão USB. Podem possuir portas para ligação de teclado, mouse, microfone e alto falantes.

**FIGURA 13.17**

Ligação de dispositivos que funcionam como hub.

A figura 17 mostra alguns dispositivos USB que têm esta característica. Note que o monitor é ligado ao computador, e nele existem conexões nas quais estão ligados o microfone, alto falantes e o teclado. O teclado deste exemplo também funciona como hub. Nele estão ligados um mouse e uma caneta eletrônica.

O USB 2.0 é compatível com o USB 1.1. Dispositivos de ambos os tipos podem ser misturados, mas devemos fazê-lo de tal forma que as ligações em USB 2.0 não sejam interrompidas por ligações em USB 1.1. Em outras palavras, entre cada dispositivo USB 2.0 e o computador (que também deve ter interface USB 2.0) devem existir apenas caminhos em USB 2.0, como mostra a figura 18. Se ligarmos na interface USB 2.0 de um computador, um hub USB 1.1, todas as conexões daí em diante irão operar no modo 1.1, mesmo que sejam dispositivos USB 2.0.

**FIGURA 13.18**

Mistura de dispositivos USB 1.1 e 2.0.

A figura 18 mostra a forma correta de intercalar dispositivos USB 1.1 e 2.0. As linhas mais finas representam conexões em USB 1.1 e as mais largas em USB 2.0. O computador é tem interfaces USB 2.0, portanto as conexões para o teclado e para o hub são feitas em 2.0. O teclado é deste exemplo é 1.1, portanto a conexão entre ele e o mouse é feita em 1.1. As três portas do hub USB 2.0 são também do tipo 2.0. Neste hub ligamos um modem USB 2.0, uma câmera USB 2.0 e um hub USB 1.1. Note que a conexão entre os dois hubs é 2.0, mas a taxa de transferência será de no máximo 12 Mbps, já que o segundo hub é 1.1.

Processo de enumeração

Quando o sistema operacional é inicializado, é feita uma busca em todos os dispositivos ligados ao barramento USB. A cada um deles é atribuído um endereço de 1 a 127. Esta distribuição de números é chamada processo de enumeração. Os dispositivos USB não possuem números fixos. Ao invés disso, eles dão ao sistema informações como o identificador do fabricante, identificador do modelo / revisão, e o número de série. Desta forma nunca existirão dois dispositivos USB com os mesmos parâmetros. Ao buscar esses parâmetros, o sistema poderá identificar exatamente quantos dispositivos existem instalados e dar números apropriados para cada um. A enumeração também ocorre no instante em que um dispositivo é ligado ao barramento já em funcionamento (hot plugging).

Este processo é diferente do utilizado por outros barramentos como IDE e SCSI, que utilizam números fixos para seus dispositivos, e de barramentos como PCI, ISA e AGP, que usam endereços de E/S bem determinados para suas placas.

Graças ao processo de enumeração, o usuário não precisa se preocupar com distribuir endereços para os dispositivos. Basta conectá-los e eles automaticamente receberão um endereço identificador.

Dispositivos especiais

Já é possível encontrar dispositivos USB dos mais exóticos. O dispositivo da figura 18 é um conversor de USB para áudio e vídeo analógico. Podemos conectar nele uma TV, câmera ou vídeo cassete. Através dele podemos capturar pelo computador, na forma digital, as imagens e o som dos dispositivos de áudio e vídeo nele ligados. Podemos ainda enviar para uma saída de vídeo, imagens geradas pelo computador. Este dispositivo opera como uma placa digitalizadora, a diferença é que os circuitos estão

embutidos em uma caixa que faz parte do próprio cabo. O software que acompanha este dispositivo permite fazer as conversões e gerações de som e vídeo.

**FIGURA 13.19**

Adaptador USB para áudio e vídeo.

O adaptador da figura 20 é útil para ligar dispositivos não USB em computadores que não possuem as interfaces apropriadas (normalmente notebooks). Ligamos este adaptador à interface USB do computador, e nele podemos diversos dispositivos, pois ele tem uma interface paralela, uma interface serial, uma interface para teclado e uma interface para mouse.

**FIGURA 13.20**

Adaptador de USB para outras interfaces.



Da mesma forma encontramos adaptadores de USB para paralela, de USB para SCSI e até alguns como o ligado no mouse da figura 21. É um adaptador que permite ligar um mouse USB em uma interface de mouse PS/2.



*** 35% ***

FIGURA 13.21

Mouse USB com adaptador para PS/2.

Barramento Firewire

O Firewire é usado principalmente para transmissão de som e vídeo digital e armazenamento de dados. Já podemos encontrar câmeras digitais para fotos, câmeras digitais para vídeo, discos rígidos, gravadores de CDs, drives de DVD, scanners de alta resolução e vários periféricos utilizando este padrão.

Como surgiu o Firewire, padronização, nomenclatura

Computadores produzidos pela Apple sempre foram melhores que os PCs para manipulação de vídeo. Enquanto nos PCs era preciso instalar dispositivos, placas e drivers, os Macs já saíam de fábrica com conexões próprias para scanners, câmeras, etc. Mantendo esta linha, a Apple desenvolveu há alguns anos o barramento Firewire, hoje presente na configuração padrão dos seus computadores (assim como o USB).

Em 1995 o Firewire tornou-se um padrão do IEEE (Instituto de engenheiros eletricitas e eletrônicos) sob o código 1394. Portanto Firewire e IEEE 1394 são sinônimos, assim como i.Link, o nome dado pela Sony para este padrão. O nome Firewire é marca registrada da Apple, portanto os fabricantes de periféricos não podem utilizar livremente esta marca sem a sua permissão. Por isso a maioria prefere utilizar o nome universal, que é IEEE 1394. O nome do padrão original é portanto IEEE 1394-1995. Posteriormente sofreu pequenas revisões e passou a ser designado como 1394a. A maioria dos dispositivos Firewire existentes hoje no mercado seguem a especificação 1394a.

Está em vias de ser lançada a especificação 1394b, que tem como principais características, o uso de taxas de transferência mais elevadas e conexões feitas com cabos mais longos.

Principais características do Firewire

Para apresentar o Firewire em poucas palavras, temos as seguintes características:

- Dados digitais transmitidos em formato serial
- Taxas de transmissão de 12.5, 25 e 50 MB/s (1394a)
- Taxas de transmissão de 100, 200 e 400 MB/s (1394b)
- Plug and Play, tem suporte nativo no Windows 9x/ME/2000/XP
- Ideal para transmissão de dados em altíssimas velocidades
- Utiliza cabos com até 4,5 metros entre dispositivos (1394a)
- Utiliza cabos com até 100 metros (1394b)
- Dispositivos são ligados em daisy-chain (em cascata)
- Permite conectar até 63 dispositivos
- Permite hot swapping – conexão e desconexão sem desligar o PC



FIGURA 13.22

Um disco rígido externo Firewire.

Transmissão serial

Todos aprendemos que a transmissão paralela é mais rápida que a serial. Nada mais lógico. Ao transmitir 8, 16, 32 ou mais bits de cada vez, teremos muito maior velocidade que se transmitíssemos um bit de cada vez. Isto é verdade quando as transmissões serial e paralela são realizadas com o mesmo clock. Na prática a transmissão serial tem algumas vantagens que a torna mais rápida que a paralela.

O primeiro problema da transmissão paralela é o espalhamento temporal dos bits transmitidos. Todos são enviados ao mesmo tempo a partir do transmissor, mas devido a problemas de propagação, nem todos os bits chegam simultaneamente ao destino. Quanto maior é o comprimento do cabo, maior é este espalhamento. Não é problema quando os sinais trafegam alguns centímetros ao longo de uma placa, mas é sério quando trafegam alguns metros ao longo de um cabo. Isto faz com que o comprimento total de um cabo paralelo não possa ser muito grande. Surge ainda a necessidade do uso de sinais de STROBE. Já que alguns bits demoram mais que os outros, o circuito transmissor precisa esperar um tempo suficiente para que todos cheguem ao destino. A seguir envia um pulso de STROBE, informando ao receptor que todos os bits estão disponíveis e podem ser lidos. Existem ainda problemas de reflexão, resolvidos com o uso de terminações. A interferência entre sinais adjacentes é resolvida com o uso de blindagens e pares diferenciais. Ambas as soluções aumentam muito a espessura do cabo, tornando-o caro e de difícil instalação. Cabos seriais são mais baratos, mais leves, de manuseio mais fácil, de blindagem mais fácil e não sofrem a maioria das distorções que ocorrem em cabos paralelos. É possível obter com eles taxas de transmissão muito mais elevadas e comprimentos bem maiores que os obtidos com cabos paralelos. Daí vem o sucesso de barramentos seriais para conexões externas, como o USB e o Firewire.

Taxas de transmissão

O padrão 1394a especifica o uso de três taxas de transmissão: 100 Mbits/s (12,5 Mbytes/s), 200 Mbits/s (25 Mbytes/s) e 400 Mbits/s (50 Mbytes/s). O uso dessas taxas de transmissão dependerá dos dispositivos e da interface utilizada. É permitido utilizar dispositivos de velocidades diferentes no mesmo barramento. Prevalecerá sempre a velocidade do menor dispositivo envolvido na comunicação. Por exemplo, se um dispositivo de 200 Mbits/s vai enviar dados a outro de 100 Mbits/s, a transmissão será feita a 100 Mbits/s. Se a seguir o mesmo dispositivo de 200 Mbits/s vai enviar dados a um dispositivo de 200 ou 400 Mbits/s, a transmissão será feita a 200 Mbits/s. Observe que as transmissões podem ser feitas entre o Host (PC) e os dispositivos, ou diretamente entre dispositivos, sem intervenção do Host – uma característica não disponível no USB. Por exemplo, uma câmera de vídeo Firewire pode enviar as imagens captadas diretamente a um VCR Firewire, ao mesmo tempo em que o Host realiza acessos a outros dispositivos. Uma banda de 5 Mbytes/s é suficiente para transmitir vídeo digital em MPEG com alta resolução (como em DVDs), utilizando apenas uma parcela do tráfego total permitido no barramento Firewire.

As placas de interface Firewire disponíveis hoje no mercado podem operar com máximo de 200 Mb/s ou 400 Mb/s. Se você quer maior flexibilidade futura, dê preferência a um modelo de 400 Mb/s. Quanto aos dispositivos Firewire disponíveis, são mais comuns os que operam a 100 e 200 Mb/s. Os de 400 Mb/s estão chegando ao mercado mais recentemente. Note que esses três modos são chamados de S100, S200 e S400, respectivamente.

1394a

Modo	Taxas
S100	100 Mb/s
S200	200 Mb/s
S400	400 Mb/s

A nova versão do padrão Firewire está em desenvolvimento, chamada 1394b. Será totalmente compatível com a versão atual e ainda permitirá velocidades de 800 Mb/s, 1600 Mb/s e 3200 Mb/s. As distâncias máximas e as taxas dependerão do tipo de cabo utilizado.

1394b

Modo	Taxas	UTP-5	POF	HPCF	MMF	STP
S100	100 Mb/s	100 m	50 m	100 m	X	X
S200	200 Mb/s	X	50 m	100 m	X	X
S400	400 Mb/s	X	X	X	100 m	4,5m
S800	800 Mb/s	X	X	X	100 m	4,5m
S1600	1600 Mb/s	X	X	X	100 m	4,5m
S3200	3200 Mb/s	X	X	X	X	X

UTP-5: Cabo trançado categoria 5

POF: Plastic Over fiber

HPCF: Hard Polymer Clad Fiber

MMF: Multi Mode Fiber

STP: Short Twisted Pair

O cabo trançado não blindado (UTP-5), muito usado em redes, permitirá operar com 100 Mb/s a distâncias de até 100 metros. O mesmo tipo de cabo permitirá operar com taxas de 400, 800 e 1600 Mb/s, mas apenas a distâncias de 4,5 metros. Outros cabos baseados em fibras óticas permitirão operar com distâncias máximas de 50 ou 100 metros, dependendo da taxa de transmissão. Até o momento não foram especificados os cabos para 3200 Mb/s, já que o 1394b será inicialmente liberado em uma versão de até 1600 Mb/s, e posteriormente sofrerá uma revisão para 3200 Mb/s.

Suporte no Windows

O Windows 98 e posteriores (já na sua versão original) bem como o Windows 2000, oferecem suporte nativo ao barramento IEEE 1394. Este suporte é mantido nas versões mais novas, como 98SE, ME e XP. Sendo assim, placas de interface Firewire podem ser instaladas sem a necessidade de drivers (pelo menos é o que ocorre com os modelos atuais). Essas placas também podem ser instaladas sob o Windows 95, mediante o uso de drivers que devem acompanhá-las. A instalação de interfaces Firewire ocorre da mesma forma que outras interfaces Plug and Play. Ao ser ligado o computador e dada partida no Windows, a interface é detectada. É feita então a instalação de drivers padrão Windows. A interface passa a constar no Gerenciador de Dispositivos. No caso de futuras interfaces IEEE 1394 que utilizem chips não suportados pelos drivers nativos do Windows, ou então em PCs equipados com o Windows 95, devemos logo após a detecção usar o botão Com Disco para instalar os drivers que o fabricante fornece.

Dependendo do dispositivo que for instalado, pode ser necessário ou não o uso de programas específicos para o seu controle. Por exemplo, discos rígidos, DVD, drives de CD-ROM e gravadores de CDs são automaticamente reconhecidos pelos drivers nativos do Windows, assim como ocorre com drives ligados nas interfaces IDE. Note que no caso de gravadores de CDs, o reconhecimento automático é feito como se fossem drives de CD-ROM. Para usar as funções de gravação de CD-R e CD-RW é preciso instalar um software de gravação, como o Adaptec CD Creator.

Dispositivos de alta velocidade

À primeira vista o IEEE 1394 pode parecer redundante em relação ao barramento USB, já que ambos destinam-se à conexão de periféricos externos e utilizam a transmissão serial. A diferença está na velocidade. O barramento USB 1.1 opera com 12 Mbits/s (1,5 Mbytes/s). É adequado para conexão de mouse, teclado, joystick, impressora, discos removíveis (existem por exemplo versões do ZIP Drive para interface USB), caixas de som, câmeras de vídeo conferência, scanners de média resolução, câmeras digitais para fotos, modems, etc. Alguns dispositivos disponíveis com interface Firewire também são encontrados com interfaces USB. Sua taxa de 1,5 Mbytes/s a torna adequada ao uso de gravadores de CD-R e CD-RW, por exemplo. É velocidade suficiente para fazer gravações em até 8x.

O USB 2.0, com sua taxa de até 480 Mbits/s, excede a velocidade do 1394a. Entretanto o USB não permite transmissões entre os periféricos, como ocorre

no Firewire. Em termos de velocidade, o 1394b ultrapassará o USB 2.0, oferecendo taxas de 800, 1600 e 3200 Mbits/s.

Outros dispositivos necessitam de taxas de transferência ainda mais elevadas, e portanto o barramento USB 1.1 é inadequado. Tradicionalmente utilizam os barramentos mais rápidos, como IDE e SCSI. O barramento IEEE 1394 é suficientemente veloz para permitir a instalação da maioria dos dispositivos rápidos que hoje operam com os barramentos IDE e SCSI (assim como o USB 2.0). O padrão IEEE 1394b irá ultrapassar a velocidade máxima permitida por essas tradicionais interfaces rápidas. Assim como encontramos hoje, discos óticos e discos rígidos SCSI externos, encontramos também discos externos que usam a interface IEEE 1394. Sua instalação é extremamente simples. Basta conectar o dispositivo na interface através do cabo e ligá-lo. O Windows reconhece automaticamente o novo drive. É uma instalação mais fácil que a de dispositivos SCSI. Não é preciso usar terminadores nem definir endereços.

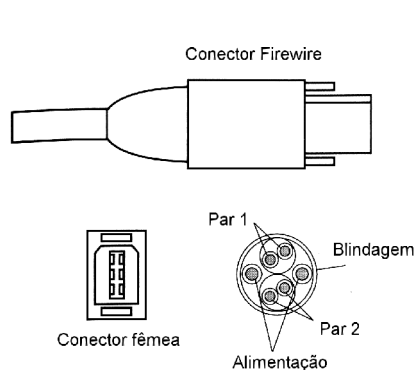
A Sony é outra empresa que adotou o Firewire (a Sony utiliza para este barramento o nome i.Link) para seus dispositivos de áudio e vídeo. Hoje oferece uma boa variedade de câmeras de vídeo, gravadores de vídeo e sistemas de som, totalmente digitais, conectados através de Firewire.

Apple, Compaq, Dell, NEC e Sony estão oferecendo PCs equipados com interfaces e periféricos Firewire. Apple, Gateway e Sony oferecem notebooks equipados com esta interface. A partir desses grandes fabricantes, aumentará cada vez mais o número de outros fabricantes adotando este barramento. Podemos encontrar ainda impressoras (Epson) Firewire, scanners de alta resolução, discos rígidos, gravadores de CDs, gravadores e editores de vídeo digital, TVs interativas de alta definição... Aparelhos de som da Pioneer, Kenwood, Philips e Yamaha também estão adotando o Firewire. A Kodak oferece vários modelos de câmeras digitais para fotos usando esta rápida interface.

Antigos aparelhos analógicos podem ser convertidos para Firewire. É possível encontrar no mercado alguns conversores para este fim. Sinais de vídeo analógico provenientes de câmeras e VCRs, por exemplo, podem ser digitalizados e transmitidos pelo barramento Firewire. Da mesma forma, dados de som e vídeo digital que trafegam pelo Firewire podem ser convertidos para o formato analógico e enviados a equipamentos de áudio e vídeo comuns.

Cabos e conectores

O barramento Firewire utiliza um conector blindado de 6 vias. Duas vias são de alimentação. As demais 4 vias formam dois pares diferenciais, sendo um para dados e um para clock. Os conectores são também de 6 vias, como vemos na figura 23. Existem ainda conectores de 4 vias, que não utilizam alimentação, apenas dados e clock.



*** 35% ***
FIGURA 13.23

Conector e cabo Firewire.

A tensão de alimentação fornecida através do cabo pode variar de 8 a 40 volts, e a corrente máxima deve ser de 1,5 A. Placas de interface para PC normalmente fornecem a tensão de +12 volts. Através de conversores DC/DC, cada periférico pode obter a partir da tensão do cabo, a alimentação de que necessita. Isto pode ser suficiente para manter em funcionamento periféricos mais simples e de baixo consumo. O ideal entretanto é que cada periférico utilize sua própria alimentação.



FIGURA 13.24

Cabo Firewire com conectores de 4 e de 6 vias.

Os cabos 1394a que ligam dois dispositivos vizinhos devem ter no máximo 4,5 metros. Distâncias maiores são obtidas quando os dispositivos são ligados em cadeia. Na parte traseira de cada dispositivo existem dois conectores, que podem ser de 4 ou 6 vias, dependendo do dispositivo. A diferença é que o

conector de 6 vias traz alimentação, e o de 4 vias não. Um deve ser ligado ao dispositivo anterior e o outro ao dispositivo posterior, caso exista um. Não são utilizados terminadores, como ocorre no barramento SCSI.

O conector da interface Firewire no PC possui 6 vias, ou seja, traz alimentação. O cabo a ser usado deverá ser 6/6 ou 6/4, dependendo do periférico ligado. Devemos ligar inicialmente os periféricos que obtêm alimentação do cabo, usando apenas cabos 6/6. Depois do último dispositivo que usa alimentação do cabo, podemos utilizar os dispositivos que possuem alimentação própria, todos com cabos de 4 vias.

Além de podermos ligar os dispositivos Firewire em cascata, podemos ainda utilizar hubs. Existem hubs com conectores de 4 e de 6 vias, e sua utilização é similar à dos hubs USB. A figura 25 mostra um hub Firewire de 4 portas de 6 vias, com alimentação própria.

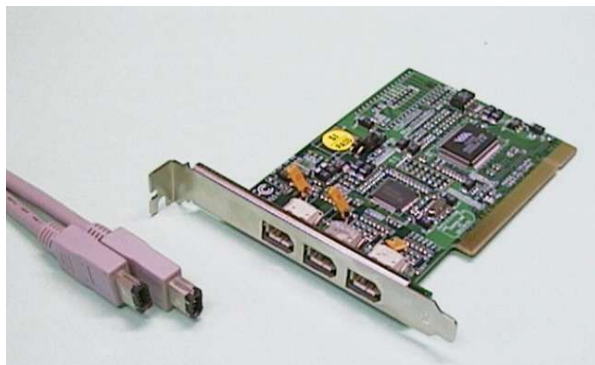


FIGURA 13.25

Hub Firewire.

Interface Firewire

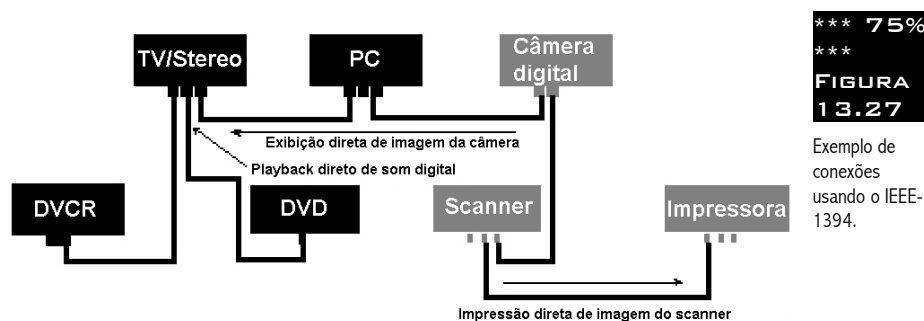
Em um futuro próximo, as placas de CPU terão interfaces Firewire integradas, assim como ocorre hoje com as interfaces USB. Encontraremos interfaces Firewire integradas aos chipsets. Por enquanto isto não ocorre, e para usar este barramento é preciso utilizar placas de expansão PCI/Firewire, como a mostrada na figura 26. Essas placas são detectadas automaticamente pelo Windows (98 e superiores), que já possui drivers IEEE 1394 nativos.

**FIGURA 13.26**

Placa de interface Firewire.

Conexão de dispositivos

O barramento IEEE 1394 permite conectar até 63 dispositivos. Os dispositivos podem operar de forma independente, não necessariamente dependendo do controle do computador (Host). Na figura 27, uma câmera de vídeo digital está enviando imagens diretamente a uma TV digital, enquanto o som de alta fidelidade vinda de um DVD é enviado aos circuitos de som da TV digital, ao mesmo tempo em que a imagem capturada por um scanner é impressa.



Hot Swapping

Esta é uma característica também presente no barramento USB. Dispositivos podem ser conectados e desconectados sem a necessidade de reiniciar o computador. Abra a janela Meu Computador, conecte um drive Firewire (ou USB) externo e ligue-o. Depois de poucos segundos o novo drive aparecerá automaticamente na janela Meu Computador. Alguém poderá dizer "... faço isso com o teclado e o mouse há muitos anos...". Conectar e desconectar equipamentos ligados não é coisa que se deva fazer. Os equipamentos têm grande chance de estragar com essas operações. Muitos usuários leigos já estragaram teclados, impressoras e suas interfaces devido a esta prática. Já os

barramentos Firewire e USB têm sua interface elétrica projetada para suportar este tipo de operação sem causar dano.

Comparação entre Firewire e USB

Praticamente todos os PCs novos já contam com o barramento USB, graças à presença de duas interfaces USB embutidas na maior parte dos chipsets atuais. Este caminho também será seguido pelo Firewire, que hoje normalmente necessita de uma placa de expansão, mas em um futuro muito próximo estará embutido nas placas de CPU. O barramento USB 1.1 destina-se à conexão de periféricos de baixa velocidade, e o Firewire a velocidades mais elevadas. O USB 2.0 se equipara ao 1394a em termos de taxa de transferência, mas não de funcionalidade. A tabela abaixo resume as principais características desses barramentos.

	1394a	1394b	USB 1.1	USB 2.0
Número de dispositivos	63	63	127	127
Hot swap	SIM	SIM	SIM	SIM
Comprimento máximo do cabo entre dois dispositivos	4,5 m	100 m	5 m	5 m
Velocidades	100, 200 e 400 Mb/s	800, 1600 e 3200 Mb/s	1,5 Mb/s e 12 Mb/s	1,5 Mb/s, 12 Mb/s e 480 Mb/s
Comunicação direta entre dispositivos	SIM	SIM	NÃO	NÃO

Barramento ATA

Não é um barramento de E/S tão versátil quanto o USB, Firewire e SCSI, no que diz respeito ao número de dispositivos que podem ser ligados. Um barramento ATA permite ligar até dois dispositivos. Podem ser discos rígidos, drives de CD-ROM, unidades de fita, gravadores de CDs, drives de DVD, discos removíveis e vários outros dispositivos. Um PC típico tem duas interfaces ATA (AT Attachment), mais conhecidas popularmente como IDE. Os dois dispositivos ligados em uma interface são designados como Master e Slave. É preciso atuar sobre seus jumpers para indicar qual deles é o Master (Device 0) e qual deles é o Slave (Device 1).

A interface IDE é uma das mais simples existentes em um PC. O processador “enxerga” esta interface como um conjunto de 9 endereços de E/S. A interface IDE primária ocupa os endereços 1F0-1F7 e 3F6. A interface IDE secundária ocupa os endereços 170-177 e 376 (hexadecimal). O interessante é que os circuitos representados por esses endereços não ficam na interface, e sim no dispositivo IDE (o disco rígido, por exemplo). A figura

28 mostra esses endereços de E/S, como são “vistos” pelo processador e como realmente estão implementados no dispositivo.

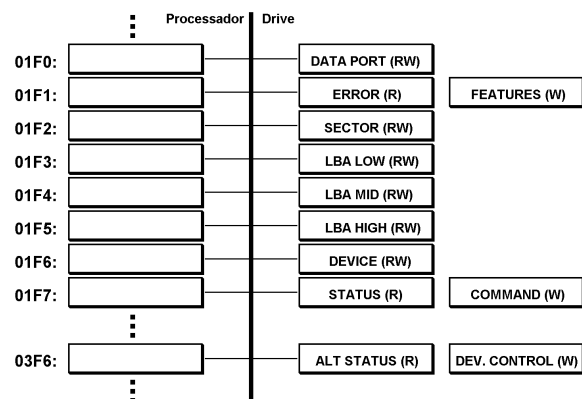
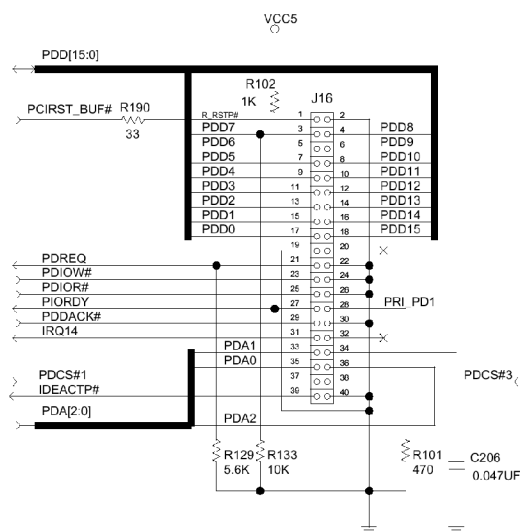


FIGURA 13.28

Uma interface IDE, vista pelo processador.

Esses circuitos são chamados de registradores, e cada um deles tem 8 bits. Note que alguns registradores são usados em operações de leitura e escrita, outros são somente para leitura, outros são somente para escrita. Por exemplo, ao escrevermos um valor no endereço 1F7, estamos enviando um comando para o dispositivo (leitura de setor, gravação de setor, etc.). Ao lermos este mesmo endereço 1F7, estamos verificando o status do dispositivo, ou seja, obtendo informações sobre as operações em andamento ou finalizadas. Para o BIOS ou os drivers dos dispositivos IDE fazerem seus acessos, precisam executar operações de entrada e saída (IN e OUT) sobre os endereços apropriados da sua interface.

**FIGURA 13.29**

Conector IDE.

O barramento ATA utiliza um conector de 40 vias como o mostrado na figura 29. Desses sinais, 16 são relativos aos dados, indicados na figura como PD0-PD15. Os demais são sinais de controle, como leitura, escrita, ready, IRQ, DMA, etc.

Na figura 30 vemos uma lista desses sinais. Note que vários deles são bidirecionais, outros trafegam no sentido do Host (interface) para o barramento (dispositivos), outros trafegam no sentido inverso.

Signal name	Connector contact	Connector contact	Signal name
RESET-	1	2	Ground
DD7	3	4	DD8
DD6	5	6	DD9
DD5	7	8	DD10
DD4	9	10	DD11
DD3	11	12	DD12
DD2	13	14	DD13
DD1	15	16	DD14
DD0	17	18	DD15
Ground	19	20	(keypin)
DMARQ	21	22	Ground
DIOW-STOP	23	24	Ground
DIOR-:HDMARDY-:HSTROBE	25	26	Ground
IORDY:DDMARDY-:DSTROBE	27	28	CSEL
DMACK-	29	30	Ground
INTRQ	31	32	Obsolete (see note)
DA1	33	34	PDIAG-:CBLID-
DA0	35	36	DA2
CS0-	37	38	CS1-
DASP-	39	40	Ground

NOTE – Pin 32 was defined as IOCS16 in ATA-2, ANSI X3.279-1996.

FIGURA 13.30

Sinais de um barramento ATA.

A tabela abaixo mostra a descrição dos sinais do barramento ATA.

Sinal	Descrição
CS0-CS1	Formam junto com DA0-DA2, o endereço do registrador interno do dispositivo a ser acessado.
DA0-DA2	Formam o endereço do registrador interno a ser acessado no dispositivo desejado, juntamente com CS0-CS1.
DASP	Indica dispositivo ativo e Slave Present.
DB0-DB15	São os 16 bits de dados usados nas leituras e escritas.
DIOR / HDMARDY / HSTROBE	Este pino tem tripla função. Quando operando em modo PIO (entrada e saída programada), indica operação de leitura. Nas leituras em modo Ultra DMA, o dispositivo indica à interface que um dado está pronto. Nas escritas por Ultra DMA, a interface indica ao dispositivo que o dado está pronto.
DIOW / STOP	Tem dupla função. Ao operar no modo PIO, indica operação de escrita. No modo Ultra DMA, este sinal causa a finalização de uma transferência.
IORDY / DDMARDY / DSTROBE	Também tem tripla função. Ao operar em modo PIO, faz com que o ciclo de leitura ou escrita seja estendido até os dados estarem prontos, o que equivale a uma espécie de wait state. Também faz a sincronização nas leituras e escritas em modo Ultra DMA.
DMACK	A interface responde à requisição de um dispositivo que quer fazer uma transferência por DMA.
DMAREQ	O dispositivo informa à interface que está pronto para fazer uma transferência por DMA.
INTRQ	O dispositivo solicita uma interrupção à interface. Na interface IDE primária, este pino está ligado diretamente em IRQ14, e na interface secundária é ligado diretamente em IRQ15.
PDIAG / CBLID	Este sinal é gerado pelo Device 1 (Slave) durante o processo de power on, indicando ao Device 0 (Master) que seu processo de inicialização terminou com sucesso. No padrão ATA-3 e superiores, este pino tem dupla função: serve também para identificar o tipo de cabo (40 ou 80 vias).
RESET	Este sinal provoca o RESET dos dispositivos ligados na interface. É conectado diretamente no RESET OUT da placa de CPU.
CSEL	Faz o selecionamento de Master/Slave sem necessidade de utilizar jumpers. Normalmente este recurso não é utilizado.

Curiosa é a forma usada pelo endereçamento dos dispositivos Master e Slave em uma interface IDE. A própria interface não tem registradores, apenas transmite os comandos para o cabo. Tanto o Master como o Slave possuem registradores idênticos, como os mostrados na figura 28. A diferença é que o registrador DEVICE (endereço 1F6 ou 176 na interface primária e secundária) é usado para especificar se os comandos atuais dizem respeito ao Master ou ao Slave. Dos 8 bits deste registrador (bit0-bit7), o bit 4 indica se os comandos são para o Master ou para o Slave. O valor 0 seleciona o Master e o valor 1 seleciona o Slave. Quando a interface transmite um comando ou um dado, tanto o Master como o Slave o recebem, mas apenas aquele que estiver selecionado o reconhecerá.

Transferências em modo PIO

As primeiras interfaces IDE operavam em modo PIO. Esta modalidade usa a técnica de entrada e saída programada, ou seja, realizada diretamente pelo processador. A operação de escrita é mostrada na figura 31. É feita através da instrução `IN AX,1F0` (no caso da interface primária). Esta operação de entrada faz com que o Address Bus indique o endereço correto para o dispositivo selecionado e comanda um pulso de DIOR (Data I/O Read), como mostra a figura. O dado será lido e colocado nos pinos D0-D15 da interface. Note que também podem ser realizadas operações de 8 bits, mas isto não é usual.

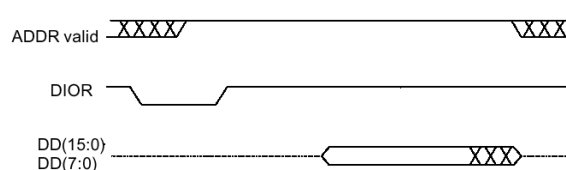


FIGURA 13.31

Leitura em modo PIO.

A operação de escrita é similar, e é mostrada na figura 32. É baseada na instrução `OUT AX,1F0` (interface primária). O valor de 16 bits presente no registrador AX do processador é enviado para a interface, juntamente com um pulso de DIOW (Data I/O Write).

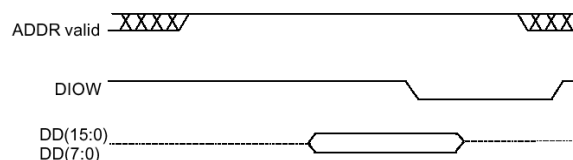


FIGURA 13.32

Operação de escrita em modo PIO.

Como os setores dos discos rígidos têm 512 bytes, essas operações devem ser executadas 256 vezes seguidas, já que cada operação envolve 2 bytes. Já os setores de um CD-ROM têm 2048 bytes, portanto a sua leitura é feita por 1024 operações seguidas.

Tanto na operação de leitura como na escrita, o sinal IORDY é usado para sincronizar a velocidade do dispositivo com a velocidade do processador. Isto significa que o processador fará uma breve pausa entre a chegada ou a saída de dados consecutivos.

Os modos PIO sofreram uma evolução ao longo dos anos 90. Na primeira versão da especificação ATA, era usado o PIO Mode 0. Depois surgiram os modos 1, 2, 3 e 4. Cada um desses modos tem um período padrão para a transferência de cada dado. O inverso do período resulta na frequência, e multiplicando a frequência por 2 (já que são transferidos 2 bytes por período) temos a taxa de transferência máxima teórica. A tabela que se segue mostra esses valores para os diversos modos PIO.

PIO Modo	Ciclo	Frequência	Taxa de transferência
0	600 ns	1,66 MHz	3,33 MB/s
1	383 ns	2,6 MHz	5,2 MB/s
2	330 ns	3 MHz	6 MB/s
3	180 ns	5,5 MHz	1,11 MB/s
4	120 ns	8,33 MHz	16,6 MB/s

Transferências em modo Multiword DMA

Note na figura 32 que as transferências em modo PIO são comandadas pelos sinais DIOR e DIOW (Data I/O Read e Data I/O Write), que são originados em comandos IN e OUT executados pelo processador. As transferências no modo Multiword DMA utilizam o recurso Bus Mastering das interfaces IDE. Nesta modalidade, o processador não se envolve diretamente com a transferência dos dados individuais, apenas dá o comando para a interface, que fará todo o trabalho. Quando um disco rígido está programado para operar em modo Multiword DMA, os sinais DMAREQ e DMACK são utilizados para estabelecer uma conexão de DMA entre o dispositivo e a interface. Os sinais DIOR e DIOW são usados para sincronizar a transferência de cada grupo de 16 bits, a diferença é que esses sinais não serão gerados pelo processador, e sim pela interface IDE.

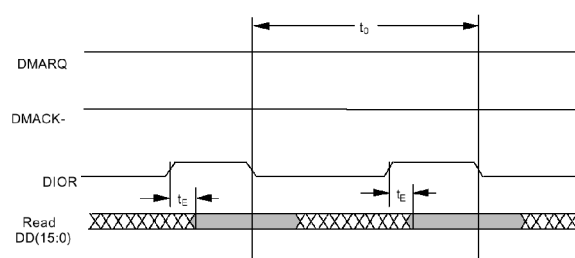
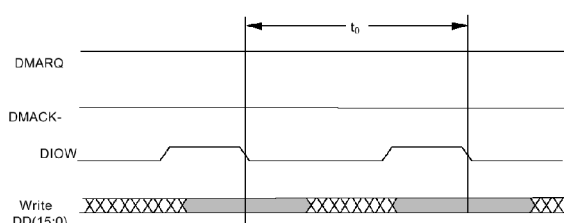


FIGURA 13.33

Leitura em Multiword DMA.

A figura 33 mostra o trecho de uma operação de leitura em modo Multiword DMA. Ao receber o comando de leitura, o dispositivo ativa o sinal DMARQ (DMA Request) e a interface responderá com o sinal DMACK (DMA

Acknowledge). A partir daí a interface irá gerar sinais DIOR (Data I/O Read) consecutivos para que sejam feitas as leituras, em grupos de 16 bits (DD0-DD15). Cada vez que o sinal DIOR faz uma transição de 0 para 1, o dispositivo enviará os dados após um tempo t_E especificado. A unidade mínima de leitura é o setor de 512 bytes, portanto um comando de leitura nesta modalidade terá 256 ciclos. Na prática são transferidos múltiplos setores, o que é feito pela programação do registrador SECTOR (veja a figura 28). Ao término da transferência, os sinais DMAREQ e DMACK são desativados (voltam ao valor 0) e a interface irá gerar uma interrupção (IRQ14 ou IRQ15) para avisar o processador que a leitura terminou. Durante a transferência por DMA, o processador fica liberado para executar outras tarefas.

**FIGURA 13.34**

Escrita em Multiword DMA.

A figura 34 mostra uma operação de escrita em Multiword DMA. É bastante similar à operação de leitura no que diz respeito ao uso de sinais e interrupções. A diferença é que neste caso, a interface envia os dados e ativa o sinal DIOW. A transição de 1 para 0 em DIOW ocorre no instante em que os dados estão prontos, sendo então recebidos pelo dispositivo. Observe que tanto na leitura como na escrita (figuras 33 e 34) em Multiword DMA, está especificado como t_0 , a duração do ciclo, e em cada ciclo 16 bits simultâneos são transferidos em DD0-DD15. Existem 3 modos Multiword DMA, e as diferenças estão nas durações desses ciclos. Quanto menor é a duração, mais elevada é a taxa de transferência máxima teórica, como mostra a tabela a seguir:

Multiword DMA	Ciclo	Frequência	Taxa de transferência
Modo 0	480 ns	2,08 MHz	4,16 MB/s
Modo 1	150 ns	6,66 MHz	13,3 MB/s
Modo 2	120 ns	8,33 MHz	16,6 MB/s

As interfaces e dispositivos IDE antigos podiam operar em um outro modo que já caiu em desuso, o Singleword DMA. É similar ao Multiword DMA, exceto pelo fato dos sinais DMARQ e DMACK serem ativados e

desativados para cada dado transferidos. Com este protocolo perde-se mais tempo e as taxas de transferência obtidas são menores. O modo Multiword é mais eficiente, pois não perde em cada ciclo, tempo com a ativação e desativação desses sinais. A tabela abaixo mostra as características das transferências em Singleword DMA, modos 0, 1 e 2:

Singleword DMA	Ciclo	Frequência	Taxa de transferência
Modo 0	960 ns	1,04 MHz	2,08 MB/s
Modo 1	480 ns	2,08 MHz	4,16 MB/s
Modo 2	240 ns	4,16 MHz	8,33 MB/s

Dispositivos e interfaces IDE modernas não suportam mais a operação em modo Singleword DMA.

Transferências em modo Ultra DMA

As transferências em modos Ultra DMA são muito parecidas com as do modo Multiword DMA. A principal diferença é que em cada ciclo são feitas duas transferências de dados (o que é uma espécie de DDR – Double Data Rate). A outra diferença importante é que os modos Ultra DMA operam com ciclos de menor duração, resultando em taxas de transferência maiores.

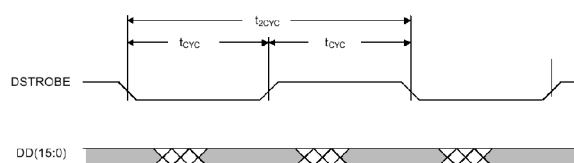
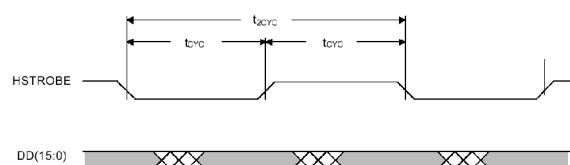


FIGURA 13.35

Leitura em Ultra DMA.

A figura 35 mostra o trecho de uma leitura em modo Ultra DMA. Neste modalidade, os sinais DMAREQ e DMACK são usados da mesma forma como no Multiword DMA, mas não estão mostrados na figura. A sincronização dos dados entre a interface e o dispositivo é feita pelo sinal DSTROBE. Este sinal é o mesmo IORDY (tem tripla função, e no modo Ultra DMA funciona como Data Strobe). É gerado pelo dispositivo para avisar a interface de que o dado a ser lido já está pronto. Note que os dados são fornecidos tanto na subida quanto na descida de DSTROBE, ou seja, este modo de transferência opera com Double Data Rate – duas transferências por ciclo. Observe ainda que o sinal DSTROBE trabalha com períodos idênticos em 0 e em 1.

**FIGURA 13.36**

Escrita em Ultra DMA.

A figura 36 mostra uma operação de escrita em modo Ultra DMA. Assim como na leitura, são feitas duas transferências por ciclo. A diferença é que desta vez a sincronização é feita pelo sinal HSTROBE (que é o sinal DIOR, gerado pela interface, mas que opera como HSTROBE no modo Ultra DMA). As transições de 0 para 1 e de 1 para 0 geradas pela interface em HSTROBE avisam ao dispositivo que o dado está pronto para ser lido (pelo dispositivo) em DD0-DD15.

Podemos portanto afirmar que os modos Ultra DMA são variantes do Multiword DMA, exceto pelo fato de utilizar duas transferências por cada ciclo, sinais de sincronização diferentes (DSTROBE e HSTROBE) e operarem com ciclos de duração menor, resultando em maiores taxas de transferência. A tabela que se segue mostra as características dos diversos modos Ultra DMA:

Ultra DMA	Ciclo	Frequência	Taxa de transferência
Modo 0	240 ns	4,16 MHz	16,6 MB/s
Modo 1	160 ns	6,25 MHz	25 MB/s
Modo 2	120 ns	8,33 MHz	33 MB/s
Modo 3	90 ns	11,1 MHz	44,4 MB/s
Modo 4	60 ns	16,6 MHz	66,6 MB/s
Modo 5	40 ns	25 MHz	100 MB/s

As versões do padrão ATAPI e os modos suportados

Todos os modos PIO, Multiword DMA e Ultra DMA fazem parte dos diversos padrões ATA/ATAPI liberados ao longo dos anos. A primeira especificação foi a ATA-1, a atual é a ATA-6. A tabela abaixo mostra os modos e as especificações ATA nas quais são suportados:

Modo	Taxa de transferência	ATA-1	ATA-2	ATA-3	ATA-4	ATA-5	ATA-6
PIO 0	3,33 MB/s	X	X	X	X	X	X
PIO 1	5,2 MB/s	X	X	X	X	X	X
PIO 2	6 MB/s	X	X	X	X	X	X
PIO 3	11,11 MB/s	-	X	X	X	X	X
PIO 4	16,6 MB/s	-	X	X	X	X	X
Singleword DMA 0	2,08 MB/s	X	X	-	-	-	-

Singleword DMA 1	4,16 MB/s	X	X	-	-	-	-
Singleword DMA 2	8,33 MB/s	X	X	-	-	-	-
Multiword DMA 0	4,16 MB/s	X	X	X	X	X	X
Multiword DMA 1	13,3 MB/s	-	X	X	X	X	X
Multiword DMA 2	16,6 MB/s	-	X	X	X	X	X
Ultra DMA 0	16,6 MB/s	-	-	-	X	X	X
Ultra DMA 1	25 MB/s	-	-	-	X	X	X
Ultra DMA 2	33 MB/s	-	-	-	X	X	X
Ultra DMA 3	44,4 MB/s	-	-	-	-	X	X
Ultra DMA 4	66,6 MB/s	-	-	-	-	X	X
Ultra DMA 5	100 MB/s	-	-	-	-	-	X

Note alguns modos apresentam taxas de transferência iguais, apesar de terem características completamente diferentes. Por exemplo o PIO modo 4, o Multiword DMA modo 2 e o Ultra DMA modo 0 apresentam taxas iguais a 16,6 MB/s, mas são três coisas completamente diferentes. Inclusive o Multiword DMA modo 2 costuma ser confundido com o Ultra DMA modo 0. O PIO modo 4 e o Multiword DMA modo 2 operam com ciclos de 120 ns, e fazem uma transferência a cada ciclo. O modo PIO é operado pelo processador, e o modo Multiword DMA é operado pela interface. Já o Ultra DMA modo 0 tem ciclos de 240 ns, mas faz duas transferências a cada ciclo, e também é operado pela interface, e não pelo processador.

O primeiro grande melhoramento sofrido pelo padrão ATA foi o lançamento da versão ATA/ATAPI-2. As taxas de transferência chegaram a 16,6 MB/s com o PIO modo 4 e o Multiword DMA modo 2. Entretanto a maioria das interfaces IDE da época não operava com o modo DMA, portanto a operação ficou restrita ao PIO modo 4. A especificação ATA/ATAPI-3 não introduziu novos modos, mas eliminou os modos Singleword, já que eram menos eficientes que os modos Multiword.

Os modos Ultra DMA foram introduzidos na especificação ATA/ATAPI-4, inicialmente até o modo 2 (33 MB/s). Este modo também é chamado de ATA-33, Ultra IDE-33 ou Ultra DMA-33. As versões 5 e 6 do padrão ATA/ATAPI ofereciam novos modos Ultra DMA de até 66 MB/s (ATA-66) e 100 MB/s (ATA-100).

Normalmente o sistema operacional, o BIOS e a interface IDE interrogam os dispositivos e determinam os modos de operação suportados, e operam com o mais rápido disponível, mas nem sempre isto ocorre. No Windows é preciso ativar a operação em DMA, através do Gerenciador de Dispositivos. Se esta configuração não for feita, os dispositivos irão operar em modo PIO. No CMOS Setup podemos indicar se desejamos ou não que sejam usados modos Ultra DMA, e ainda indicar qual o máximo modo a ser usado por

cada dispositivo IDE. Quando é usada a opção AUTO, o modo mais veloz é utilizado.

Master e Slave

Como sabemos, Master e Slave são os nomes que recebem os dois dispositivos ligados em uma interface ATA. Esses nomes são muito mal escolhidos, inclusive nas especificações ATA/ATAPI não são utilizados. Ao invés deles, são usados os termos Device 0 (Master) e Device 1 (Slave). O uso dos nomes Master e Slave dá uma idéia errada de que um dispositivo é controlado pelo outro. Isto não é o que ocorre. Uma interface ATA não é na verdade uma interface como as outras, e sim, um conjunto de circuitos que passam e recebem dados pelo cabo. As interfaces propriamente ditas ficam localizadas em cada dispositivo. Existem ainda uma duplicação de interfaces, ou seja, todos os circuitos existentes no Device 0 também estão no Device 1. Todos os comandos enviados pelo barramento chegam aos dois dispositivos. Antes de enviar comando, é feita a indicação do dispositivo desejado, através do Device Register (figura 28). Antes de enviar comandos, o sistema programa este registrador com 0 (Device 0) ou 1 (Device 1). Ambos os dispositivos ligados no cabo receberão este valor. A partir daí, somente o dispositivo selecionado irá responder e executar os comandos, mas mesmo o dispositivo que não está selecionado continuará interpretando os comandos recebidos, aguardando que seja selecionado.

Os dois dispositivos operam exatamente da mesma forma. A única situação em que ocorre uma diferença física entre os dois é durante a inicialização, após um Reset. O dispositivo Slave informa ao Master, através do sinal PDIAG (pino 34 do cabo) se ele completou com sucesso sua inicialização. Ao receber um erro, caberá ao Master informar esta condição ao sistema.

O cabo de 80 vias

Para permitir o funcionamento em Ultra DMA modo 3 e superiores, é necessário utilizar um cabo flat especial, com 80 vias. Este cabo tem, além dos 40 condutores normais, mais 40 condutores alternados e ligados ao terra. Esses novos 40 condutores funcionam como uma blindagem, e evitam as distorções nos sinais que trafegam pelo cabo quando operam em frequências muito altas.

O funcionamento desta blindagem é fácil de entender. Quando dois fios são colocados lado a lado, como ocorre nos cabos flat, as tensões existentes em um fio podem ser captadas pelos fios vizinhos (esta interferência é chamada de crosstalk, uma espécie de “linha cruzada”).

Detecção do tipo de cabo

Quando uma interface é capaz de operar em Ultra DMA modo 3 (44 MB/s) ou superiores, é necessário que seja utilizado um cabo flat de 80 vias. Como o cabo de 80 vias utiliza também um conector com apenas 40 vias (por questões de compatibilidade), é preciso que o sistema detecte qual é o tipo de cabo utilizado. Se for um cabo de 40 vias, automaticamente a velocidade máxima permitida será a do Ultra DMA modo 2 (33 MB/s, ou ATA-33). Quando for detectado o cabo de 80 vias, os modos mais rápidos estarão liberados, desde que é claro, sejam suportados pela interface e pelos dispositivos.

A detecção do tipo de cabo é feita através do pino 34. Nas implementações ATA-4 e anteriores, a velocidade máxima permitida é de 33 MB/s, obtida com o cabo IDE de 40 vias. Nessas especificações o pino 34 do cabo é utilizado pelo sinal PDIAG. Este sinal é usado para que o Slave informe ao Master que seu diagnóstico foi concluído com sucesso. Esta informação não precisa ser passada para a interface, é usada apenas na comunicação entre os dispositivos. Nos cabos de 80 vias, o pino 34 no conector que é ligado à interface deve ser ligado ao terra (0 volts). Nos dois conectores, a ligação entre os pinos 34 é normal, como mostra a figura 37.

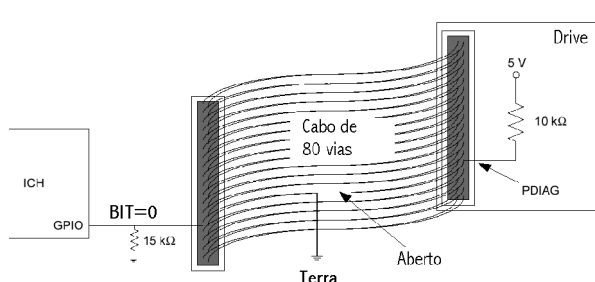


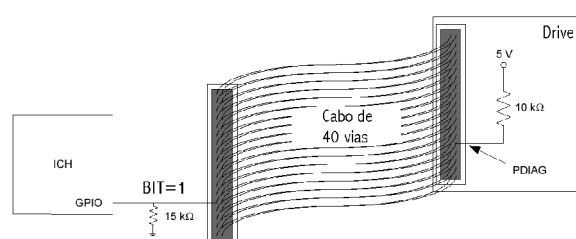
FIGURA 13.37

Detecção de um cabo flat IDE de 80 vias.

O chip ICH (Southbridge do chipset) tem além das suas várias interfaces, bits de entrada e saída que podem ser usados para monitorar e controlar certos sinais da placa de CPU. Esses pinos são chamados de GPIO (general purpose I/O). Um desses bits pode ser usado para ler o valor digital existente no pino 34 de cada interface IDE. Se o valor lido for um bit zero, significa que trata-se de um cabo de 80 vias. Por isso os cabos de 80 vias normalmente possuem um conector azul, enquanto os outros dois são cinza. O conector azul é diferente dos outros dois, já que possui esta conexão do pino 34 com o terra, e deve ser ligado no conector da placa de CPU. Se este

cabo for ligado na posição invertida, irá funcionar quando ligamos um só dispositivo, mas se ligarmos 2, o sinal PDIAG não passará do Slave para o Master, e assim o sistema irá ignorar o Slave.

O cabo IDE de 40 vias não tem esta conexão ao terra no pino 34. Quando é feita a leitura neste bit, é recebido o valor 1, já que este pino é ligado ao VCC (tensão da fonte, normalmente +5 volts) através de um resistor de 10k. Portanto quando é lido um bit 1, significa que trata-se de um cabo de 40 vias.

**FIGURA 13.38**

Detecção de um cabo flat IDE de 40 vias.

Interface para drive de disquetes

Esta é uma das interfaces que dá menos trabalho a um técnico. Não necessita de configuração alguma e é totalmente padronizada. Todas as interfaces para drives de disquetes, assim como todos os drives de disquetes (exceto os muito antigos) são idênticos. No início dos anos 80, os PCs usavam 5 tipos de drives de disquetes:

Drive	Disquetes suportados				
	5.25" DD	5.25" HD	3.5" DD	3.5" HD	3.5" ED
5.25" DD (360kB)	X	-	-	-	-
5.25" HD (1.2 MB)	X	X	-	-	-
3.5" DD (720 kB)	-	-	X	-	-
3.5" HD (1.44 MB)	-	-	X	X	-
3.5" ED (2.88 MB)	-	-	X	X	X

Os drives de disquetes de 360 kB, 1.2 MB e 720 kB tornaram-se obsoletos a partir de 1993. O drive de 2.88 MB (3.5" ED – Extra High Density) prometia ser o sucessor dos modelos de 1.44 MB, mas infelizmente isto não ocorreu por razões comerciais. Seu custo era elevado em comparação com outros meios de armazenamento mais eficientes, já disponíveis na época. De todos esses sobrou apenas o drive de 1.44 MB (3.5", high density). Entretanto as interfaces para drives de disquetes são capazes de controlar todos os tipos de drives, exceto os de 2.88 MB. Para checar se uma interface para drives é

compatível com os drives de 2.88 MB, basta checar se esta opção aparece no Standard CMOS Setup.

O drive de 2.88 MB utiliza um método especial de gravação, chamado gravação perpendicular, para dobrar a taxa de dados em comparação com a utilizada pelo modo high density.

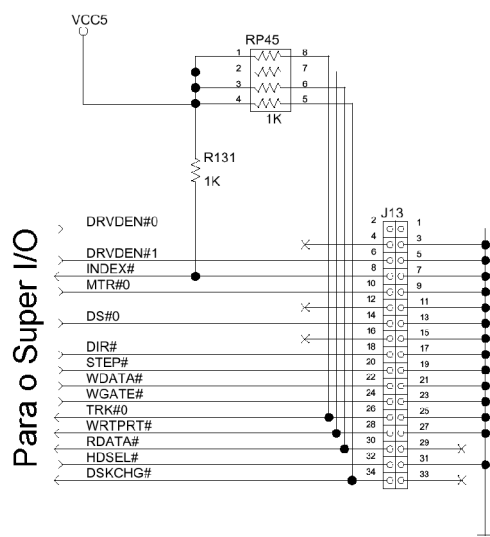


FIGURA 13.39

Conexões de uma interface para drives de disquetes.

A figura 39 mostra as conexões internas de uma placa de CPU com a interface para drive de disquetes. Todos os sinais são gerados pelo chip Super I/O e são enviados ao conector de 34 vias, no qual é ligado o cabo flat que vai até o drive de disquetes. Note que vários pinos são ligados ao terra. São todos os pinos ímpares, portanto neste cabo flat, os sinais (pinos pares) são intercalados por blindagens, que são os pinos ímpares aterrados. Alguns sinais são ligados a uma tensão de +5 volts através de resistores de pull-up com 1 k Ω . Esses resistores levam seus sinais a um nível lógico 1 quando o dispositivo ligado no cabo está inativo. São os seguintes os sinais que trafegam neste cabo:

Pino	Sinal	Descrição
2	DRV DEN0	Indicador de densidade
6	DRV DEN1	Indicador de densidade
8	INDEX	Gera pulsos a cada rotação do disquete. É ligado ao sensor de índice do drive, através do qual o <i>index hole</i> (furo de índice) marca o início de cada trilha.
10	MTR 0	Liga motor do drive 0
12	SELEC 1	Seleciona drive 1

14	SELEC 0	Seleciona drive 0
16	MTR 1	Liga motor do drive 1
18	DIR	Indica a direção do movimento a ser realizado pelas cabeças de leitura e gravação (para as trilhas internas ou para as trilhas externas)
20	STEP	Faz as cabeças avançarem ou voltarem uma trilha, de acordo com a direção indicada por DIR.
22	WDATA	Envia os dados a serem gravados no disquete, em formato serial, codificado em MFM.
24	WGATE	Ativa a gravação no drive.
26	TRACK 0	Indica se as cabeças estão posicionadas na trilha zero
28	WPROT	Indica se o disquete está protegido contra gravação.
30	RDATA	Recebe dados nas operações de leitura
32	HEADSEL	Seleciona cabeça 0 ou 1
34	DSKCHANGE	Indica se o drive foi aberto, o que normalmente indica troca de disquete. Mesmo quando é mantido o mesmo disquete, este sinal é ativado com a abertura da tampa ou o pressionamento do botão Eject.

O sistema de leitura e gravação nos drives de disquetes é extremamente simples. A figura 40 mostra de forma simplificada, uma operação de escrita. Após localizado o início de um setor, ele é identificado. Se for o setor desejado, o sinal Write Gate é ligado no início da área de dados. A seguir são transmitidos todos os bytes do setor, no formato serial, através do sinal Write Data. Depois dos dados são enviados dois bytes de CRC, usados para checagem de erros. Finalmente o sinal Write Gate é desligado.

**FIGURA 13.40**

Gravação em um disquete.

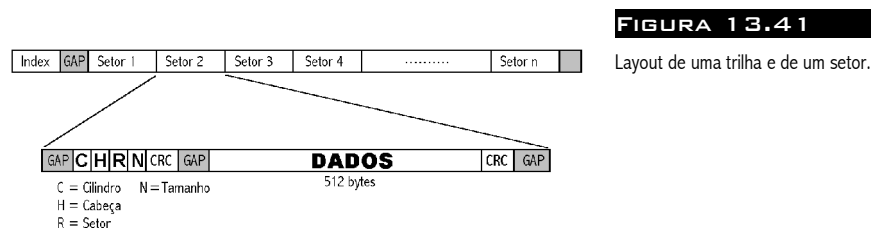
A operação de leitura também é simples. Consiste em buscar o setor desejado, que ao ser encontrado, terá seus dados lidos e transferidos para a memória. O CRC do setor é verificado e a interface informa ao sistema se ocorreu erro ou não. A transferência dos bytes é feita por DMA, usando o canal DMA2. Ao término das operações de leitura, escrita, formatação ou outras realizadas pelo drive, é gerada uma interrupção através da linha IRQ6. O uso do DMA2 e do IRQ6 é padrão em todas as interfaces de disquetes dos PCs.

Layout das trilhas do disquete

É interessante entender como é feita a gravação em um drive de disquetes, pois o processo é praticamente o mesmo empregado em outros meios

magnéticos, como discos rígidos. Também é usado com mínimas modificações nos meios óticos como os CD-ROMs.

Todas as trilhas de um disquete são divididas em setores. Os setores são criados durante o processo de formatação do disquete. A figura 41 mostra o layout de uma trilha.



A trilha começa com uma marca de índice, que não é na verdade gravada. É marcada fisicamente no disco, através de um furo chamado index hole (furo de índice). Um sensor ótico e um LED são colocados em lados opostos do disquete, e quando o furo de índice passa entre eles, o sensor irá gerar um pulso que será transmitido pelo cabo flat. Assim a interface saberá o início da trilha. Após o índice existe uma pequena pausa de segurança, chamada GAP. A partir daí são encontrados os setores.

Na figura 41 vemos também o layout de cada setor da trilha. No seu início existe um gap que o separa do setor anterior. A seguir existe um campo de identificação, composto de 4 valores: C=número do cilindro (ou trilha); H=número da cabeça; R=número do setor e N=tamanho do setor. Os tamanhos permitidos são 1, 2 e 3, que correspondem a setores de 256, 512 e 1024 bytes. Nos disquetes de PCs é padronizado o tamanho de 512 bytes por setor.

Depois desses 4 valores existem dois bytes de CRC, usados para a checagem dos valores de C, H, R e N. Após o CRC existe um GAP que separa o campo de identificação do campo de dados. Este gap é necessário para que gravações na área de dados não afetem o campo de identificação. Depois dos dados existem dois bytes de CRC para checagem de erros. Quando existe erro na checagem de CRC relativo ao campo de identificação, é retornado um código de erro que significa “setor não encontrado”. Quando existe erro na checagem do CRC da área de dados, é retornado um código de erro que significa “erro de leitura”.

A demarcação magnética da trilha, que cria os setores com os respectivos campos de identificação, é criada durante o processo de formatação do disquete. Note que não é qualquer modo de formatação que faz isso. O processo normal de formatação não cria os setores nas trilhas. Ele apenas apaga os diretórios e faz a leitura de todos os setores do disco, à procura de erros. Para que seja feita uma nova demarcação magnética dos setores, é preciso usar a formatação incondicional ou física. O comando de formatação do Windows só opera neste modo quando o disquete nunca foi formatado. O comando FORMAT do MS-DOS precisa ser usado com a sintaxe "FORMAT A: /U".

Movimentação das cabeças

A interface de drives de disquete sempre informa qual é a trilha onde as cabeças estão posicionadas. Ao receber um comando para fazer o acesso a um setor de uma outra trilha, a interface determina por quantas trilhas precisa mover as cabeças, e em qual direção. Por exemplo, para ir da trilha 15 até a trilha 20 é preciso mover as cabeças 5 trilhas no sentido interno. O sinal DIR é ativado de acordo, e são enviados pulsos através do sinal STEP. Esses pulsos devem ter uma duração compatível com o mecanismo dos drives. A sua duração é o que chamamos de stepping rate.

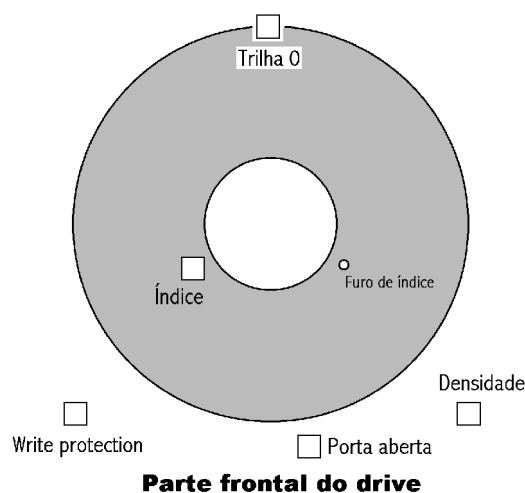
Ao chegar na trilha desejada, é feita uma leitura do campo de identificação de um setor qualquer, e é determinado qual é o cilindro (ou trilha) atual. Em caso de erro, ou seja, se a trilha atual não for igual à trilha desejada, é preciso executar uma operação chamada recalibrate. Consiste em mover as cabeças até a trilha zero, e mover novamente as cabeças até a trilha desejada. Ao contrário das demais trilhas, a trilha zero não é detectada pela leitura de campos de identificação, e sim pela ativação de um sensor de trilha zero, que envia este sinal de detecção para a interface. Se a operação de recalibrate não tiver sucesso, são feitas mais algumas tentativas (retries). Podemos ouvir claramente os sons emitidos pelo drive quando tentar ler um disquete com erros. São feitas operações de recalibrate sucessivas, emitindo um som característico de disquete com problemas.

Sensores do drive

Os drives de disquetes são dispositivos, digamos, sem inteligência. Realmente eles não possuem processador ou controlador. Seus circuitos são extremamente simples, e os sinais de sua interface estão diretamente relacionados com a movimentação das cabeças e as leituras e gravações na mídia magnética. Todo o controle do drive é feito graças às informações fornecidas por 5 sensores, ligados diretamente em pinos da interface:

Pino	Sinal	Descrição
2	DRV DEN0	Indicador de densidade
8	INDEX	Gera pulsos a cada rotação do disquete. É ligado ao sensor de índice do drive, através do qual o <i>index hole</i> (furo de índice) marca o início de cada trilha.
26	TRACK 0	Indica se as cabeças estão posicionadas na trilha zero
28	WPROT	Indica se o disquete está protegido contra gravação.
34	DSKCHANGE	Indica se o drive foi aberto, o que normalmente indica troca de disquete.

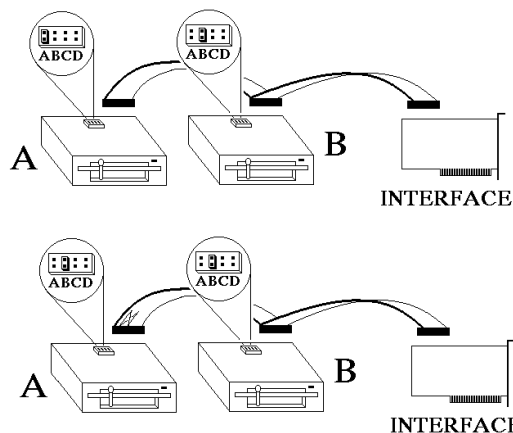
Todos esses sensores precisam funcionar corretamente, caso contrário o drive não funcionará. A figura 42 mostra a localização dos sensores em um drive de disquetes.

**FIGURA 13.42**

Sensores de um drive de disquetes.

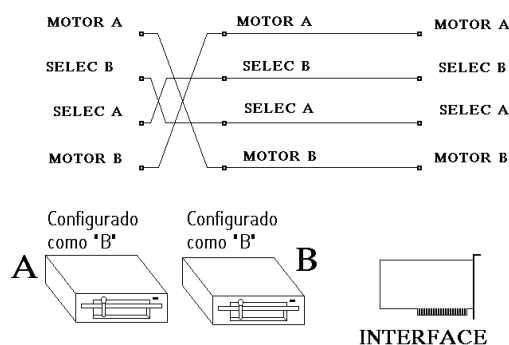
Funcionamento do cabo trançado

Os drives usados nos primeiros PCs utilizavam jumpers para selecionamento. Podiam ser configurados como A ou B. Ao instalar um drive configurado como A, o seu jumper deveria ser colocado na posição “A”, e ao instalar um drive como “B”, o jumper deveria ser configurado com a opção B. No final dos anos 80 começou a ser usado um sistema muito mais prático, que é o Cable Select. Neste sistema, todos os drives são configurados como B. Um trançamento no conector da extremidade do cabo flat faz com que um drive B, ao ser ligado neste ponto, seja reconhecido pela interface como A. A figura 43 mostra as configurações usando o cabo sem inversão (antigo) e o cabo com inversão (atual).

**FIGURA 13.43**

Uso do cabo flat com inversão.

O uso do cabo flat com inversão é muito vantajoso, já que com ele não é preciso configurar jumpers nos drives. Todos os drives são configurados na fábrica como B, e a inversão do cabo fará com que aquele ligado na extremidade passe a operar como “A”. O funcionamento desta inversão é mostrado na figura 44. Ele atua sobre os fios 10, 12, 14 e 16 do cabo, que correspondem aos sinais Motor A, Selec B, Selec A e Motor B.

**FIGURA 13.44**

Funcionamento do cabo flat com inversão.

Ambos os drives são configurados como B. A ligação entre a interface e o drive B (o do conector no meio do cabo) é direta, ou seja, não tem inversão. Sendo assim o drive ligado neste conector responderá a todos os comandos relacionados a B, enquanto o outro drive irá ignorá-los. A inversão no cabo faz com que os papéis de Motor A e Motor B, Selec A e Selec B sejam trocados para este drive. Quando a interface enviar comandos Motor A e Selec A, o drive do meio do cabo irá ignorá-los, enquanto a inversão fará com que o drive da extremidade do cabo responda a esses comandos.

Note que existem algumas placas de CPU que suportam apenas um drive de disquetes. Nesses casos os pinos Selec B e Motor B ficam sem conexão.

////// FIM //////////////////////////////////