Capítulo 4

Arquitetura de um PC

O que é a arquitetura de PCs

O hardware é uma área onde nos preocupamos com todos os aspectos de um computador, chegando até o nível de portas lógicas e componentes eletrônicos em geral, correntes e tensões, glitches, overshoot e outros efeitos elétricos. Na arquitetura, nosso objeto de estudo está um nível acima. Não importa saber de forma detalhada como os circuitos são construídos, e sim, como se conectam e como funcionam. Na arquitetura de computadores apresentamos conceitos como CPU, memória, dispositivos de entrada e saída. Sempre que possível exemplificamos os conceitos usando PCs.

Neste capítulo vamos estudar a arquitetura de forma mais profunda, entretanto voltada exclusivamente para PCs. Para trabalhar com montagem, manutenção e expansão de PCs não é preciso conhecer hardware de forma tão detalhada, chegando ao nível de portas lógicas, chips, correntes e tensões, mas é preciso conhecer a fundo a arquitetura dos PCs. Falaremos neste capítulo sobre processadores, memórias, chipsets, dispositivos de entrada e saída, interfaces, canais de DMA, interrupções e outros conceitos importantes.

Processadores

Este componente é o principal responsável pelo desempenho de um PC. Exemplos de processadores usados nos PCs são o Pentium 4, Athlon, Pentium III e Duron, além de outros, é claro. Todos os processadores usados nos PCs são descendentes do 8086, o primeiro processador de 16 bits lançado pela Intel, no final dos anos 70. Na discussão que faremos a seguir, encontraremos diversos termos técnicos relacionados com os processadores, por exemplo:

- Barramento de dados
- Barramento de endereços
- Acesso à memória
- Acesso a entrada e saída

Para facilitar a compreensão desses termos, apresentaremos aqui uma descrição simplificada de um processador. Esta descrição não irá reproduzir diretamente as características dos processadores usados nos PCs, mas dará ao leitor, o embasamento necessário para entendê-los.

Um processador é um chip que contém o que chamamos de Unidade Central de Processamento (em inglês, Central Processing Unit, ou CPU). É responsável por buscar e executar instruções existentes na memória. Essas instruções são o que chamamos de "linguagem de máquina". São comandos muito simples, como operações aritméticas e lógicas, leituras, gravações, comparações e movimentações de dados. Essas instruções simples, quando agrupadas, formam o que chamamos de programas.

Um processador precisa realizar operações de leitura da memória. Nessas leituras o processador recebe as instruções a serem executadas e os dados a serem processados. Também é preciso realizar gravações de dados na memória, para guardar os resultados intermediários e finais do processamento.

Não basta ser capaz de realizar leituras e gravações na memória. Um processador também precisa ser capaz de comunicar-se com o mundo exterior. Neste mundo exterior está o usuário que opera o computador. É preciso ler dados provenientes do teclado, mouse e outros dispositivos de entrada, bem como transferir dados para o vídeo, impressora e outros dispositivos de saída. Essas operações são chamadas de "entrada e saída", ou E/S (em inglês, Input/Output, ou I/O). Portanto, além de processar dados, um processador deve ser capaz de realizar operações de entrada e saída, bem como realizar leituras e gravações na memória.

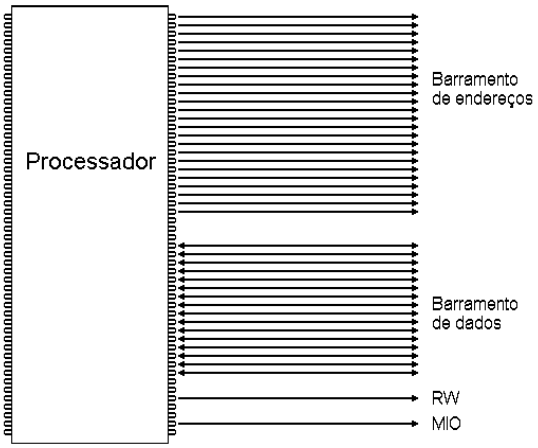


FIGURA 4.1

Representação simplificada de um processador.

A figura 1 mostra, de forma bem simplificada, alguns dos sinais digitais existentes em um processador. Temos o chamado "barramento de dados", através do qual trafegam os dados que são transmitidos ou recebidos pelo processador. Os dados transmitidos podem ser enviados para a memória ou para um dispositivo de saída, como o vídeo. Os dados recebidos podem ser provenientes da memória, ou de um dispositivo de entrada, como o teclado. Cada uma das "perninhas" do processador pode operar com um bit. No processador da figura 1, temos um barramento de dados com 16 bits. Observe que as linhas desenhadas sobre o barramento de dados possuem duas setas, indicando que os bits podem trafegar em duas direções, saindo e entrando no processador. Dizemos então que o barramento de dados é bidirecional.

O barramento de endereços serve para que o processador especifique qual é a posição de memória a ser acessada, ou qual é o dispositivo de entrada e saída a ser ativado. Na figura 1, temos um barramento de endereços com 24 bits, já que são usadas 24 "perninhas" do processador para a formação deste barramento. Observe ainda que o barramento de endereços é unidirecional, ou seja, os bits "saem" do processador.

Além desses dois barramentos, a figura mostra ainda dois sinais de controle que servem para definir se a operação a ser realizada é uma leitura ou uma gravação, e se deve atuar sobre a memória ou sobre um dispositivo de E/S. São eles:

MIO: Este sinal indica se a operação diz respeito à memória ou a E/S

RW: Este sinal indica se a operação é uma leitura ou uma gravação

Através desses dois sinais, podem ser definidas 4 operações básicas:

- Leitura da memória
- Escrita na memória
- Leitura de E/S (Ex: do teclado)
- Escrita em E/S (Ex: no vídeo)

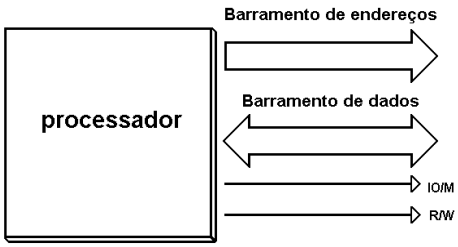


FIGURA 4.2

Outra forma de representar os barramentos de um processador.

Note que o processador representado na figura 1 tem 20 linhas que indicam os endereços e 16 que indicam os dados. São ao todo 36 linhas. Processadores mais modernos operam com um número ainda maior de bits. Por exemplo, 32 bits de enereços e 64 bits de dados. O número de linhas é tão grande que sua representação torna-se confusa. Por isso é comum utilizar a representação da figura 2. Usamos setas maiores para representar um conjunto de bits que têm a mesma função, como o barramento de dados e o barramento de endereços.

Os processadores possuem, além do barramento de dados e de endereços, o chamado barramento de controle, no qual existe uma miscelânea de sinais digitais com diversas finalidades. Os sinais RW e MIO exemplificados na figura 1 são parte do barramento de controle. Outros exemplos de sinais deste barramento são os que descrevemos a seguir.

INT

Este sinal é uma entrada que serve para que dispositivos externos possam interromper o processador para que seja realizada uma tarefa que não pode esperar. Por exemplo, a interface de teclado interrompe o processador para indicar que uma tecla foi pressionada. Esta tecla precisa ser lida, e seu código deve ser armazenado na memória para processamento posterior. As interfaces de drives e do disco rígido interrompem o processador para avisar o término de uma operação de leitura ou escrita. Vários outros dispositivos

também precisam gerar interrupções. Como existe apenas uma entrada INT, o processador opera em conjunto com um chip chamado controlador de interrupções. Este chip é encarregado de receber requisições de interrupção de vários dispositivos e enviá-las ao processador, de forma ordenada, através do sinal INT.

NMI

Este é um sinal de interrupção especial para ser usado em emergências. Significa Non Maskable Interrupt, ou Interrupção não mascarável. Em outras palavras, esta interrupção deve ser atendida imediatamente. Ao contrário do sinal INT, que pode ser ignorado pelo processador durante pequenos intervalos de tempo (isto se chama mascarar a interrupção), o sinal NMI é uma interrupção não mascarável. Nos PCs, o NMI é usado para informar erros de paridade na memória e outras condições catastróficas.

INTA

Significa Interrupt Acknowledge, ou seja, reconhecimento de interrupção. Serve para o processador indicar que aceitou uma interrupção, e que está aguardando que o dispositivo que gerou a interrupção identifique-se, para que seja realizado o atendimento adequado.

VCC

Esta é a entrada de corrente elétrica que alimenta os circuitos internos do processador. Processadores antigos operavam a partir de uma tensão de 5 volts. A partir de meados dos anos 90, passaram a utilizar tensões mais baixas, como 3,5 volts. Todos os processadores modernos operam com duas tensões (VCC1 e VCC2). A tensão externa é sempre de 3,3 volts (já existem modelos mais recentes que operam externamente com 2,5 volts), e é usada para alimentar os circuitos que se comunicam com o exterior do processador. A tensão interna é usada para alimentar o interior (núcleo) do processador, e é sempre mais baixa. Nos processadores recentes, a tensão interna é inferior a 2 volts. Note que cada tensão de entrada não ocupa um único pino do processador, e sim, vários pinos. Como a corrente total é relativamente alta, os processadores usam vários pinos para a entrada da tensão do núcleo (Core) e para a tensão externa (I/O).

GND

Significa Ground, ou Terra. Deve ser ligado ao polo negativo da fonte de alimentação. Assim como ocorre com as entradas de VCC, os processadores possuem diversos pinos de terra, para que o fornecimento de corrente seja melhor distribuído.

Reset

Este é um sinal que está ligado ao botão Reset do painel frontal do gabinete. Ao ser ativado, o processador para tudo, e atua como se tivesse acabado de ser ligado. Este sinal é também conectado a um circuito chamado Power on Reset. Sua função é gerar, no instante em que o computador é ligado, um pulso eletrônico similar ao criado pelo pressionamento deste botão. Aproveitando os conhecimentos de eletrônica apresentados no capítulo 3, mostramos no final deste capítulo, o funcionamento de um circuito de Reset.

Clock

Esta entrada deve receber um sinal digital que será usado internamente para sincronizar todo o funcionamento do processador. Explicando de forma simplificada, se um processador recebe um clock de 100 milhões de ciclos por segundo, ele executará 100 milhões de operações por segundo.

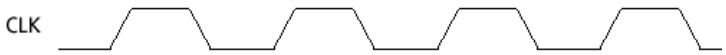


FIGURA 4.3
Diagrama de tempo de um sinal de clock.

A figura 3 mostra o diagrama de tempo de um sinal de clock. Seus bits se alternam de forma periódica, entre 0 e 1. Um trecho com valor 1, seguindo por um trecho com valor 0, é o que chamamos de período do clock. O período é calculado em função do valor do clock, pela seguinte fórmula:

$T = 1/f$

Na fórmula, T é o período, dado em segundos, e f é a freqüência do clock, medida em Hz (hertz). Por exemplo, se tivermos um clock de 100 MHz (100.000.000 Hz), o período será de:

$T = 1/100.000.000 = 0{,}000\ 000\ 01s$

Para evitar o uso de casas decimais, toma-se o hábito de usar a unidade ns (nano-segundo, ou bilionésimo de segundo). Para fazer a conversão basta andar com a vírgula, 9 casas decimais para a direita. Portanto temos:

$0{,}000\ 000\ 01s = 10\ ns$

A maioria dos circuitos digitais opera a partir de uma base de tempo, um clock. São chamados de circuitos síncronos. Os processadores são circuitos síncronos, já que são comandados por sinais de clock. As transições se positivas e negativas (0 para 1 e 1 para 0) do sinal de clock indicam aos circuitos digitais que o utilizam, o momento certo de realizar suas operações.

Processadores antigos (até o 486DX-50) utilizavam um único sinal de clock para suas operações internas e externas. A partir daí, e até os dias atuais, os processadores passaram a operar com dois clocks, sendo um interno e um externo. O clock interno é sempre mais alto, e é usado para sincronizar as operações de processamento. Quando falamos, por exemplo, sobre um "Pentium III/800", estamos dizendo que o seu clock interno é de 800 MHz. O clock externo tem um valor menor, e é usado para sincronizar as operações de comunicação entre o processador, a memória, o chipset e outros circuitos externos.

Não só o processador opera a partir de um clock. Vários outros circuitos e barramentos do computador têm suas operações sincronizadas por um clock. Por exemplo:

- Memórias PC133 operam a partir de um clock de 133 MHz
- Memórias PC100 operam a partir de um clock de 100 MHz
- O barramento PCI opera a partir de um clock de 33 MHz
- O barramento AGP utiliza um clock de 66 MHz
- O barramento ISA utiliza um clock de 8 MHz
- Placas de som fazem digitalizações em até 44 kHz

A princípio, quanto maior é o clock de um processador, maior é o seu desempenho. Por exemplo, um processador de 800 MHz é seguramente mais veloz que um de 200 MHz. Os fabricantes de processadores se esforçam para criar modelos capazes de operar com clocks cada vez mais elevados. Não devemos entretanto levar ao pé da letra, a relação entre desempenho e clock. Por exemplo, em certas condições, um processador de 700 MHz pode ser mais veloz que um de 900 MHz. O motivo desta discrepância é que além do clock, existem outros fatores que influenciam no desempenho, como por exemplo:

- Velocidade das memórias
- Desempenho da cache L2
- Arquitetura avançada

Quando as memórias não são suficientemente velozes, podem demorar muito na entrega de dados e instruções para o processador, que acaba ficando parte do tempo ocioso, tendo seu desmpenho prejudicado pela lentidão da memória. Também a cache L2 tem papel fundamental. A cache L2 do processador Pentium III Coppermine, por exemplo, é mais eficiente que a do processador Pentium III Katmai. Portanto existem diferenças de desempenho, se comparamos essas duas versões do Pentium III, mesmo quando ambas operam com o mesmo clock. O tamanho da cache L2 também tem influência no desempenho. Processadores Athlon e Duron são idênticos, exceto pelo tamanho da cache L2 (256 kB para o Athlon e 64 kB para o Duron). Por isso ao compararmos os desempenhos desses dois processadores, mesmo operando com o mesmo clock, o Athlon leva vantagem. A arquitetura mais avançada também tem influência direta no desempenho. Um processador de 1200 MHz de 7ª geração, por exemplo, tende a ser mais rápido que um de 1200 MHz, mas de 6ª geração. Processadores de gerações mais avançadas são capazes de executar mais instruções ao mesmo tempo e operam com mais eficiência, tirando assim maior proveito do seu clock. É como comparar um carro com motor 2.0 produzido no ano 2000 com outro de motor 2.0 produzido em 1980. Os motores de geração mais nova têm maior rendimento, e tendem a obter maior desempenho em relação à potência do motor.

O clock de um processador está diretamente relacionado com o número de instruções que podem ser executadas a cada segundo. O 8086 e o 8088, nas suas primeiras versões, operavam a 5 MHz. Isto não significa exatamente 5 milhões de instruções por segundo, e sim, 5 milhões de CICLOS por segundo. Algumas instruções mais simples podiam ser executadas em apenas dois ciclos. Desta forma, em um segundo seria possível executar 2.500.000 dessas instruções. Outras instruções mais complexas, como a multiplicação e a divisão, eram muito mais demoradas. Suponha por exemplo uma instrução que precise de 10 ciclos para ser executada. Operando a 5 MHz, esses processadores poderiam executar 500.000 dessas instruções por segundo.

Com o passar do tempo e a evolução da tecnologia foi possível desenvolver processadores capazes de operar com clocks mais elevados, e o que é mais importante: executar instruções em um reduzido número de ciclos. Os processadores mais modernos são capazes de executar a maioria das instruções em apenas um ciclo. A partir do Pentium, passaram a executar instruções de forma simultânea, tornando possível, por exemplo, executar duas instruções em um único ciclo. Isto faria com que, teoricamente, operar a 200 MHz resulte em 400 milhões de instruções por segundo.

Algumas características dos processadores

Ao compararmos processadores novos e antigos, simples e sofisticados, vemos que eles possuem muitas características comuns, e as diferenças estão no nível dessas características. Um exemplo simples é o clock, já abordado neste capítulo. No ano 2000 já existiam processadores com clocks superiores a 1000 MHz. Em 1980, os modelos típicos operavam na faixa dos 5 MHz. Mesmo com esta diferença tão grande, o significado do clock é exatamente o mesmo nos dois casos. Passemos então a apresentar as principais características de um processador genérico, e vejamos como se aplicam aos modelos atuais.

Número de bits internos

Daqui vêm as terminologias "micro de 8 bits", "micro de 16 bits" e "micro de 32 bits", etc. Dentro de um processador, existem vários circuitos que armazenam, transportam e processam dados. Na maioria dos processadores atuais, tais circuitos operam com 32 bits de cada vez. Nos antigos processadores dos anos 80, todos esses circuitos operavam com 16 bits, enquanto os modelos dos anos 70 operavam com 4 ou 8 bits.

Quanto maior é o número de bits de um processador, mais veloz poderá realizar cálculos e processamento de instruções em geral. Veja por exemplo, os limites de números inteiros positivos que podem ser manipulados com 8, 16 e 32 bits:

8 bits	0 a 255
16 bits	0 a 65.535
32 bits	0 a 4.294.967.295

Suponha por exemplo que um processador de 16 bits precise realizar a operação 245.818.768 + 978.798.423. Ambas as parcelas desta adição não podem ser representadas em um grupo de 16 bits. Portanto, deve ser realizada por etapas. Um processador de 32 bits é capaz de representar e operar tais números de forma direta, o que faz com que o cálculo seja feito, no mínimo duas vezes mais rápido. Este é apenas um exemplo no qual um processador de 32 bits leva vantagem sobre um de 16 bits. Praticamente em todas as instruções, esta vantagem existe.

Os processadores 8086, 8088 e 80286, usados nos PCs do início dos anos 80 e ainda encontrados no mercado até o início dos anos 90, operavam com 16 bits. A partir do 80386, os processadores usados nos PCs passaram a operar com 32 bits. Por incrível que pareça, os processadores mais modernos, como

Pentium III, Pentium 4, Athlon e diversos outros atuais, também operam com 32 bits.

Número de bits externos

Para que um processador seja rápido, é preciso que ele seja capaz de manipular instruções em alta velocidade. Essas instruções são armazenadas na memória, e portanto, é preciso que a memória seja acessada em alta velocidade. Ao mesmo tempo em que executa instruções, o processador também lê e armazena dados na memória, o que é mais uma razão para que a memória seja rápida. A velocidade de transferência de dados entre o processador e a memória depende de diversos fatores, e um dos principais é o número de bits do seu barramento de dados (data bus). O barramento de dados é um conjunto de sinais digitais que ligam o processador à memória e aos dispositivos de entrada e saída de dados.

Os processadores de 8 bits utilizavam um barramento de dados também com 8 bits. O processador 8086, operava com 16 bits, tanto internamente como externamente, ou seja, utilizava um barramento de dados também com 16 bits. Até então, o número de bits internos era igual ao número de bits externos do processador, mas isto nem sempre ocorre. Por exemplo, o processador 8088, usado nos primeiros PCs, operava internamente com 16 bits, e externamente com apenas 8. Já com os processadores modernos (a partir do Pentium), ocorre o inverso: operam internamente com 32 bits e externamente com 64. A tabela abaixo apresenta o número interno e o número externo de bits dos processadores usados nos PCs.

Processador	Número interno de bits	Número externo de bits
8086	16	16
8088	16	8
286	16	16
386SX	32	16
386DX	32	32
486	32	32
486DLC / SLC	32	32
Pentium, Pentium MMX	32	64
Pentium Pro	32	64
Cyrix 5x86 e AMD 5x86	32	32
Cyrix 6x86	32	64
AMD K5, K6, K6-2, K6-III	32	64
Pentium II, Pentium III	32	64
Celeron	32	64
Pentium 4	32	64
AMD Athlon, Duron	32	64

Como vemos, os processadores mais recentes operam com 32 bits internos e 64 bits externos, ou seja, barramento de dados com 64 bits. Uma nova configuração foi introduzida com o processador Intel Itanium, inaugurando a era dos processadores de 64 bits. São 64 bits internos e 64 bits externos.

Capacidade de endereçamento

Aqui está um fator que não está exatamente relacionado com a velocidade, e sim, com a capacidade de manipular grandes quantidades de dados. A capacidade de endereçamento nada mais é que o máximo tamanho que pode ter a memória, ou, seja, o número máximo de células de memória que um processador consegue acessar. Para acessar uma célula (ou posição) de memória, o processador precisa informar qual é o endereço desta célula. Cada célula armazena um byte. Processadores com barramento de dados de 16 bits podem acessar duas células de uma só vez. Aqueles com barramentos de dados com 32 e 64 bits podem acessar até 4 e 8 células, respectivamente.

O 8086 e o 8088 possuíam barramentos de dados com 20 bits, e por isto podiam acessar 1 MB de memória. Para saber a quantidade máxima de memória que um processador pode acessar, basta saber o número de bits do seu barramento de endereços e calcular 2 elevado a esta número. Portanto:

2^{20} bytes = 1.048.576 bytes = 1 MB
2^{24} bytes = 16.777.216 bytes = 16 MB
2^{32} bytes = 4.294.967.296 bytes = 4 GB

Para a época do 8086 e do 8088 (em torno de 1980), a capacidade de endereçar 1 MB era considerada bem elevada. Os primeiros PCs nem mesmo chegavam a usar toda esta capacidade. Eram comuns modelos com 64 kB, 128 kB e 256 kB de memória RAM. Apenas em meados dos anos 80 começaram a ser comuns os PCs com 512 kB e 640 kB de RAM.

O processador 286, com sua capacidade de endereçar até 16 MB de memória (usava um barramento de endereços com 24 bits) foi um grande avanço em relação ao 8086 e ao 8088. Mesmo no início dos anos 90, a maioria dos PCs usava entre 1 MB e 2 MB de memória, apenas uma fração da capacidade de endereçamento do 286.

O 386, com seu barramento de endereços com 32 bits, possibilitava endereçar até 4 GB de memória, uma quantidade espantosamente alta até para os dias atuais. Um PC com 256 MB de RAM, por exemplo, não chega a usar 10% da sua capacidade máxima de memória. Por isto, mesmo os processadores mais modernos, em sua maioria, ainda utilizam barramentos

de endereços com 32 bits. A tabela abaixo apresenta o número de bits do barramento de endereços, bem como a capacidade máxima de endereçamento de memória para os processadores usados nos PCs:

Processador	Número bits de Endereço	Capacidade de endereçamento
8086	20	1 MB
8088	20	1 MB
286	24	16 MB
386SX	24	16 MB
386DX	32	4 GB
486	32	4 GB
486DLC	32	4 GB
486SLC	24	16 MB
Pentium e similares	32	4 GB
Pentium Pro e superiores	36	64 GB

Memória cache

Os processadores experimentaram ao longo dos anos, grandes avanços na velocidade de processamento. Um já ultrapassado Pentium II de 300 MHz, por exemplo, é mais de 1000 vezes mais veloz que o velho 8088 usado no IBM PC XT. As memórias também experimentaram avanços significativos, porém mais modestos. No início dos anos 80, eram comuns as memórias DRAM com 250 ns de tempo de acesso. Em meados dos anos 80, este tempo de acesso chegou à casa dos 60 ns, e no final dos anos 90, aos 10 ns. Portanto essas memórias são apenas cerca de 25 vezes mais rápidas que há 20 anos atrás, enquanto os processadores são no mínimo 1000 vezes mais rápidos. O resultado disso é um grande desequilíbrio entre a velocidade do processador e a velocidade da memória.

Este problema é antigo, pois já ocorria com os computadores de grande porte durante os anos 60. Com os processadores, só passou a existir tal problema a partir de 1990, aproximadamente. Antes disso os processadores, sendo mais lentos, ficavam perfeitamente sintonizados com a velocidade das memórias. As memórias, mesmo sendo relativamente lentas, ainda eram capazes de entregar dados na velocidade exigida pelos processadores. Somente quando o seu clock chegou a 25 MHz (por volta de 1990), os processadores passaram a ter seu desempenho penalizado pela baixa velocidade das memórias.

A memória RAM usada em larga escala nos PCs é chamada de DRAM (Dynamic RAM, ou RAM Dinâmica). Suas principais características são:

- Preço relativamente baixo

- Grande capacidade em pequeno espaço
- Velocidade relativamente baixa

O preço baixo e o alto grau de miniaturização fizeram com que a DRAM fosse o tipo de memória mais indicado para os microcomputadores. A sua baixa velocidade não chegava a ser um problema, pelo menos até 1990.

Existe um outro tipo de memória RAM que apresenta uma velocidade de operação muito mais alta. É chamada de SRAM (Static RAM, ou RAM Estática). Suas principais características são:

- Preço elevado
- Grande capacidade requer um grande espaço
- Alta velocidade

Tecnicamente seria possível equipar um PC com memória SRAM, mas teríamos duas grandes desvantagens. Uma delas é o preço. A SRAM é cerca de 10 vezes mais cara que a DRAM de mesma capacidade. A outra desvantagem é o seu baixo grau de compactação. Seriam necessárias placas de circuito enormes para dotar um PC com uma razoável quantidade de memória.

A solução utilizada pela indústria de PCs foi a mesma usada nos computadores de grande porte nos anos 60. Esta solução é a memória cache. É formada por uma pequena quantidade de SRAM, usada para acelerar uma grande quantidade de DRAM. Quando o processador precisa ler dados da DRAM, estes são antes transferidos para a cache (isto não é feito pelo processador, e sim, por um circuito especial chamado controlador de cache, que faz parte do chipset). O processador obtém os dados diretamente da cache, e enquanto esses dados estão sendo lidos, o controlador de cache se antecipa e acessa mais dados da DRAM, transferindo-os para a memória cache. O resultado é que na maior parte do tempo, o processador encontra dentro da cache os dados que procura. Este processo funciona bem porque, mesmo com grandes quantidades de memória, um processador passa bastante tempo utilizando trechos pequenos de memória. Por exemplo, ao executar um programa com o tamanho de 200 kB, todo ele cabe dentro de uma cache com apenas 256 kB. Ao executá-lo, os dados estariam, praticamente o tempo todo, sendo obtidos da rápida memória cache.

O primeiro processador a utilizar memória cache foi o 486. Em seu interior existem 8 kB de memória estática super veloz, operando como cache. Este tipo de cache, localizada dentro do processador, é chamada de:

- Cache interna
- Cache primária
- Cache de nível 1 (L1)

Apesar de ter apenas 8 kB, a cache interna do 486 podia acelerar consideravelmente o desempenho do acesso à memória.

Os processadores 386 não tinham cache interna, e nem precisavam dela, enquanto operavam com até 20 MHz. Com o lançamento de versões de 25, 33 e 40 MHz, o baixo desempenho da memória DRAM obrigou os fabricantes a acrescentarem memória cache. Esta cache não era localizada dentro do processador, como ocorria com o 486. Era formada por chips de memória SRAM, e era chamada de:

- Cache externa
- Cache secundária
- Cache de nível 2 (L2)

OBS: Note que só é correto usar o termo cache secundária ou cache L2 quando existe cache primária (ou L1), como no caso do 486.

Foram lançadas placas de CPU baseadas no 386, equipadas com 8 kB, depois com 16, 32, 64 e finalmente 128 kB de memória cache externa (isto ocorreu entre 1990 e 1993). Um computador baseado no 386DX-40, com 128 kB de cache externa, era mais veloz que um 486 de 25 MHz sem cache externa.

Hoje em dia, tanto a cache primária como a secundária são importantes para o desempenho. A tabela que se segue apresenta a quantidade de memória cache interna existente nos processadores usados nos PCs.

Processador	Cache L1
386 e anteriores	Sem cache L1
486DX / DX2 / SX / SX2	8 kB
486 DX4 *	16 kB
486DLC, 486SLC	1 kB
Pentium	16 kB
Pentium Pro	16 kB
Pentium MMX	32 kB
Cyrix 5x86 e AMD 5x86	16 kB

Cyrix 6x86	16 kB
AMD-K5	24 kB
AMD K6, K6-2, K6-III	64 kB
Cyrix 6x86MX, MII	64 kB
Cyrix III	128 kB
Pentium II, Celeron	32 kB
Pentium III	32 kB
Athlon, Duron	128 kB
Pentium 4	20 kB

Os primeiros processadores a utilizarem cache (486) tinham uma única área para dados e instruções. Novas versões do 486 e todos os processadores seguintes passaram a utilizar uma cache L1 dividida em duas áreas iguais, sendo uma para dados e uma para instruções (data cache e instruction cache). Isto tornou a cache L1 mais eficiente. Notáveis são as caches L1 dos processadores Cyrix e AMD, normalmente maiores que as de processadores Intel de mesmo poder de processamento. Por exemplo, o AMD Athlon tem 128 kB de cache L1, enquanto o Pentium III tem apenas 32 kB. Também notável é o caso da cache L1 do processador Pentium 4. Esta cache não armazena dados e instruções vindos da memória, e sim, micro-instruções já decodificadas. Isso significa que as instruções existentes na cache L1 podem ser executadas mais rapidamente.

Evolução da cache

Os primeiros processadores usados nos PCs não necessitavam de memória cache. A memória DRAM disponível na época era suficientemente veloz para aqueles processadores. O IBM PC XT, por exemplo, usava memórias com 250 ns de tempo de acesso, mas o seu processador operava com ciclos de 800 ns para realizar os acessos, portanto 250 ns era um tempo de acesso mais que satisfatório. Apenas computadores de grande porte, aqueles que custavam alguns milhões de dólares, utilizavam memória cache.

Em 1989 surgiu o processador Intel 80486, o primeiro a utilizar cache. Com clock de 25 MHz e ciclos de 80 ns, necessitava de memórias com menor tempo de acesso, porém na época as mais rápidas eram de 100 ns, tempo muito grande para aquele processador. Os 8 kB de cache, localizadas dentro do próprio processador (cache interna) permitiam o funcionamento do processador com bom desempenho, mesmo com a memória DRAM mais lenta que o necessário.

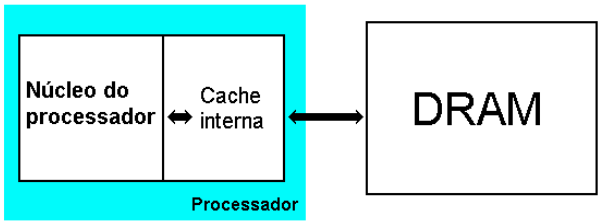


FIGURA 4.4
Cache interna do 486.

Processadores 386 produzidos pela AMD na época (1991-1993) eram concorrentes do 486, até então produzidos apenas pela Intel. Assim como ocorria no 486, os processadores 386 daquela época também necessitavam de cache para melhorar o seu desempenho. Como o 386 não tinha cache interna, foram produzidas placas de CPU 386 com cache externa, ou seja, formada por chips SRAM (RAM estática) localizados na placa de CPU. Um processador 386 de 40 MHz e 128 kB de cache externa era praticamente tão veloz quanto um 486 de 25 MHz e 8 kB de cache interna, mas a opção do 386 era muito mais barata.

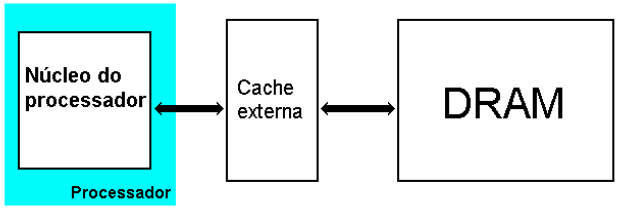


FIGURA 4.5
Cache externa de placas de CPU para 386.

A cache externa realmente acelerava bastante o desempenho, e assim foram criadas placas de CPU para processadores 486, também com cache externa. Eram comuns placas para 486 com 256 kB de cache externa, além dos 8 kB de cache interna existentes no processador.

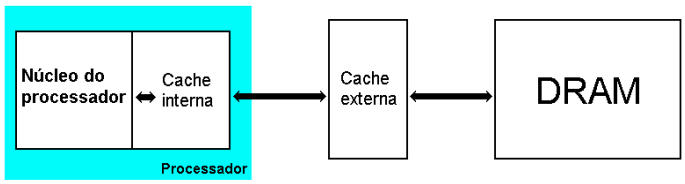


FIGURA 4.6
Cache interna e externa.

Este esquema de dupla cache (interna e externa) utilizada em processadores 486 foi mantido em processadores mais modernos, como o 586, o Pentium e

todos os demais processadores para Soquete 7, com exceção do AMD K6-III, que operava com 3 caches.

Os termos "cache interna" e "cache externa" caíram em desuso. Atualmente ambas as caches ficam localizadas dentro do próprio processador, portanto não faz mais sentido classificá-las como interna e externa. A cache interna é agora chamada de cache primária ou cache L1 (level 1 ou nível 1). A cache externa é agora chamada de cache secundária ou cache L2 (level 2 ou nível 2).

Na época em que o Pentium e o Pentium MMX eram utilizados em computadores de uso pessoal, a Intel produzia o Pentium Pro, utilizado em aplicações de nível profissional e em servidores (1995-1997). Este foi o primeiro processador a embutir a cache L2. Em outras palavras, dentro do processador Pentium Pro encontrávamos a cache L1 e 256 kB de cache L2.

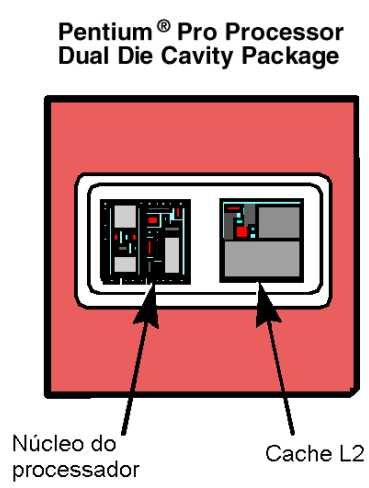
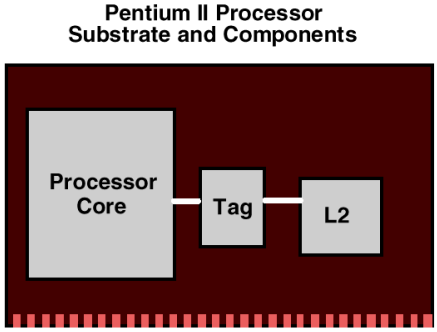


FIGURA 4.7
Cache L2 do Pentium Pro.

O Pentium Pro era construído em uma pastilha de silício (die) com dupla cavidade, ou seja, como se fossem dois chips montados em um mesmo substrato. Um deles é o núcleo do processador, o outro é a cache L2. Este método permitiu a construção de uma cache L2 bastante veloz, entretanto tinha um elevado custo de produção. O núcleo do Pentium Pro utiliza a arquitetura Intel P6, usada nos processadores seguintes (Pentium II, Celeron e Pentium III). A cache L2 entretanto nunca mais foi produzida com o sistema de dupla cavidade.

O Pentium II foi lançado em 1997, utilizando um núcleo similar ao do Pentium Pro, ou seja, ele também usa a microarquitetura P6. A principal

diferença está na sua cache L2. Ao invés de utilizar uma única pastilha de silício contendo o processador e a cache L2, o Pentium II é montado em uma placa de circuito, juntamente com chips de memória que formam a cache L2. O conjunto inteiro é montado em um cartucho metálico. Do ponto de vista do núcleo do processador, esta cache L2 é externa, mas considerando o cartucho como um todo, a cache L2 é interna. Para evitar confusão, os termos interna e externa não são mais usados, e em seu lugar usamos hoje, L1 e L2.



*** 35% ***
FIGURA 4.8

Cache do Pentium II e das primeiras versões do Pentium III e do Athlon.

Este sistema de cache L2 foi também utilizado nas primeiras versões do Pentium III e do AMD Athlon.

Cache L2 integrada no núcleo

Integrar a cache L2 no núcleo significa produzir um processador contendo na mesma base de silício, com uma única cavidade, o núcleo e a cache L2. Integrar a cache no núcleo foi possível com a adoção de tecnologia de 0,18 mícron, no lugar da antiga tecnologia de 0,25 mícron, possibilitando a construção de transistores menores, e em conseqüência, chips menores e com menor aquecimento. Além do menor custo, a cache L2 integrada ao núcleo do processador resulta em maior desempenho, já que os acessos à cache podem ser feitos com maior velocidade.

O primeiro processador a integrar a cache L2 no seu núcleo foi o Celeron. Posteriormente a mesma técnica passou a ser usada pelo Pentium III. A Intel utiliza vários nomes para diferenciar seus modelos de processador. O Pentium III versão Katmai era o original, que tinha a cache L2 formada por chips SRAM adicionais. A versão chamada Coppermine é a que integra a cache L2 no núcleo. Apesar de ter apenas 256 kB, contra os 512 kB do

Pentium III Katmai, a nova versão do Pentium III oferece maior desempenho, pois sua cache L2 opera com um clock duas vezes maior.

Também os processadores Athlon passaram a utilizar cache L2 integrada no núcleo. Assim como ocorre com os processadores Intel, são usados nomes adicionais para designar as versões do Athlon. A versão com cache L2 embutida no núcleo é chamada de Thunderbird, ou simplesmente T-Bird. Ao mesmo tempo em que foi lançado o Athlon T-Bird, com 256 kB de cache L2 integrada no núcleo, foi também lançado o Duron, utilizando a mesma tecnologia. A diferença é a cache L2, que no Duron tem apenas 64 kB. Entretanto, sua cache L1 de 128 kB (encontrada tanto no Athlon quanto no Duron) oferece um bom desempenho, mesmo com uma cache L2 de apenas 64 kB.

Velocidades das caches

Um dos principais melhoramentos introduzidos nos processadores modernos foi o aumento de velocidade da cache L2. Quando um processador se torna mais rápido, a memória DRAM não necessariamente precisa acompanhar este aumento de velocidade (e na prática não acompanha), mas a cache L2 precisa acompanhar. Se o processador se tornar mais veloz mas a cache L2 mantiver velocidade constante, o desempenho será prejudicado.

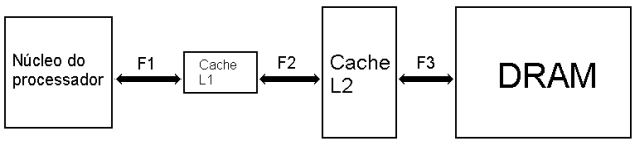


FIGURA 4.9
Relação entre o processador e as caches.

A figura 9 mostra a relação entre o processador, as caches e a memória DRAM. Para que o sistema tenha um bom desempenho, deve ocorrer o seguinte:

a) O processador encontra na maior parte das vezes, os dados e instruções que precisa na própria cache L1.

b) Os dados a serem transferidos para a cache L1 estão na maior parte das vezes, localizados na cache L2

Desta forma, a cache L2 acelera o desempenho da DRAM. Ao mesmo tempo, a cache L1 acelera o desempenho da cache L2. Note que na figura estão indicadas as freqüências F1, F2 e F3.

F1: Velocidade na qual os dados trafegam entre a cache L1 e o núcleo
F2: Velocidade na qual os dados são transferidos entre as caches L1 e L2
F3: Velocidade de transferência entre a DRAM e a cache L2

Veja como ficam essas velocidades em alguns processadores produzidos em um passado recente:

Processador	F1	F2	F3
Pentium-200	200 MHz	66 MHz	66 MHz
AMD K6-2/300	300 MHz	100 MHz	100 MHz
AMD K6-2/500	500 MHz	100 MHz	100 MHz
Pentium II/400	400 MHz	200 MHz	100 MHz

Em todos os casos, o clock usado na transferência de dados entre a cache L1 e o núcleo do processador é o próprio clock do núcleo. Por exemplo, em um núcleo de 500 MHz, esta transferência é feita a 500 MHz.

Observe o que ocorre com os valores de F2, que representa a velocidade da cache L2. Nos processadores Pentium, K6-2 e similares, a cache L2 opera com freqüência fixa, igual à freqüência do barramento externo. Um K6-2/500 tem condições de processar dados mais rapidamente que um K6-2/300, entretanto ambos possuem caches L2 com velocidades semelhantes. Aumentar mais ainda o clock do processador e manter fixa a velocidade da cache L2 é a mesma coisa que usar em um carro de Fórmula 1, pneus de Fusca.

Finalmente observe o valor de F2 para o Pentium II. Este processador possui uma cache L2 capaz de transferir dados em uma velocidade maior que a do seu barramento externo. É usado um barramento dual, um de 100 MHz para acessar a DRAM e um de 200 MHz para acessar a cache L2. No caso geral, a cache L2 do Pentium II e das primeiras versões do Pentium III (Katmai) opera com a metade da freqüência do núcleo do processador. Um Pentium III/600, por exemplo, tem cache L2 operando a 300 MHz.

O aumento do valor de F2 foi uma das prioridades nos processadores lançados recentemente. Veja o que ocorre com os modelos mais novos:

Processador	F1	F2	F3
Pentium IIIE	F	F	100 MHz
Pentium IIIB	F	F/2	133 MHz

Pentium IIIEB	F	F	133 MHz
Athlon original	F	F/2, F/2.5, F/3	200 MHz
Athlon T-bird	F	F	200/266 MHz
Duron	F	F	200 MHz
Pentium 4	F	F	400 MHz

Na tabela usamos F para indicar a freqüência do núcleo do processador. Por exemplo, em um Pentium III/1000, F vale 1000 MHz. Observe que nos processadores mais modernos, F2 (freqüência da cache L2) é igual à freqüência do núcleo do processador. Núcleo a 1000 MHz significa cache L2 a 1000 MHz. Isto resulta em um grande aumento de desempenho, em comparação com versões mais antigas. Nas primeiras versões do Pentium III, bem como no Pentium IIIB (clock externo de 133 MHz), a cache L2 operava com a metade da freqüência do núcleo. Isto também ocorria com as primeiras versões do Athlon, onde a cache L2 operava com a metade, e até com 1/3 da freqüência do núcleo. Nas versões mais novas do Pentium III (Coppermine) e nas versões T-Bird do Athlon e Duron, a cache L2 também opera com a freqüência do núcleo. Esta é uma característica que será mantida em todos os processadores modernos: cache L2 em full speed., integrada no núcleo do processador (on-die).

Finalmente observe na tabela que melhoramentos têm sido feitos na freqüência da DRAM. Novas tecnologias como DDR e RDRAM estão aos poucos sendo implantadas para tornar mais elevada a taxa de transferência dos dados que chegam da DRAM.

Cache L3

Durante aproximadamente um ano (meados de 1999 a meados de 2000), a AMD produziu o processador K6-III. Foi lançado apenas nas versões de 400 e 450 MHz, mas foi logo retirado de linha, devido ao seu custo de produção relativamente alto, o que dificultava a concorrência com os processadores Intel. O K6-III tinha uma cache L2 de 256 kB full speed integrada no seu núcleo. Processador a 450 MHz, cache L2 a 450 MHz. Seu desempenho era muito bom, bem mais veloz que o K6-2 e igualando-se ao Pentium III de mesmo clock. A AMD achou melhor descontinuá-lo e dedicar sua linha de produção ao Athlon.

O processador K6-III tinha no seu núcleo, caches L1 e L2. Podia ser instalado em placas de CPU para K6-2, que já tinham cache externa. Sendo assim, a cache existente na placa de CPU era de nível 3 (L3). A figura 10 mostra a relação entre as três caches do K6-III.

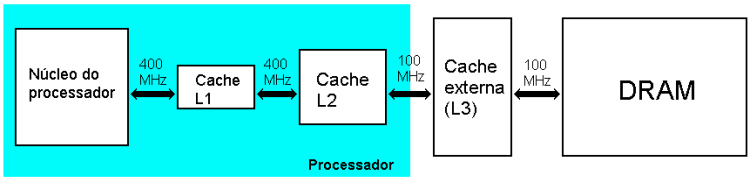


FIGURA 4.10
Relação entre as caches de um K6-III/400.

Na figura 10 foi tomado como exemplo um K6-III de 400 MHz. Estando o núcleo operando a 400 MHz, as transferências feitas entre o processador, a cache L1 e a cache L2 (internas) são feitas na mesma freqüência. Para o modelo de 450 MHz, essas transferências são feitas a 450 MHz. Em ambos os modelos, as transferências entre a cache L2 e a L3 (externa), e entre a cache L3 e a DRAM são feitas a 100 MHz.

Desempenho

Todos os esforços no sentido de melhorar a tecnologia dos processadores giram em torno de um ponto chave: o desempenho, ou seja, a velocidade de processamento. Entre as técnicas implantadas visando obter maiores velocidades, podemos citar:

- Aumento do clock
- Aumento do número interno de bits
- Aumento de número externo de bits
- Redução do número de ciclos para executar cada instrução
- Uso de cache L1 e L2 mais eficientes
- Execução de instruções em paralelo

Avanços em todas essas áreas têm possibilitado obter velocidades cada vez maiores. Para avaliar essas velocidades, é fundamental que existam métodos precisos para medir o desempenho de um processador. No tempo do PC XT, quando apenas o processador 8088 era usado, bastava indicar o seu clock, e automaticamente poderíamos ter uma idéia da sua velocidade de processamento. Por exemplo, um XT de 10 MHz era duas vezes mais veloz que um XT de 5 MHz.

OBS: O primeiro PC XT não operava com 5 MHz, e sim, com 4,77 MHz. Portanto, um XT de 10 MHz era cerca de 2,09 vezes mais veloz que o XT original.

Durante muitos anos, o desempenho dos processadores usados nos PCs foi estimado através de comparações com o IBM PC XT. Por exemplo, o 80286 de 6 MHz usado no IBM PC AT era cerca de 5,7 vezes mais rápido que o

IBM PC XT. Esta comparação é realizada através de programas chamados de benchmarks. A idéia é relativamente simples. Colocava-se um XT para executar uma grande quantidade de instruções, todas elas envolvendo apenas o processador e a memória, isto, é não eram levados em conta acessos a disco, ao vídeo e demais dispositivos. Marcava-se o tempo que o XT levava para executar esta miscelânea de instruções. Digamos que o XT tenha demorado, por exemplo, 10 segundos. Este tempo era registrado dentro do programa de benchmark. Ao usarmos este programa em um computador de teste, são executadas as mesmas instruções processadas pelo XT, e o tempo total de processamento sendo registrado. Suponha por exemplo que o tempo de execução foi de 2 segundos. Portanto, dividindo o tempo de execução do XT (10 segundos) pelo tempo de execução do computador em teste (2 segundos), encontramos como resultado o índice de velocidade. Neste exemplo, o computador em teste mostrou ser 5 vezes mais veloz que o XT.

Vários programas de benchmark foram criados ao longo dos últimos anos. Todos eles são baseados na execução de uma miscelânea de instruções, a contagem do tempo para esta execução, e a comparação com o tempo requerido por um computador tomado como referência, normalmente o IBM PC XT. Sem dúvida, um dos programas mais usados na medição do desempenho de processadores é no Norton Sysinfo. Outro menos cotado, mas também muito conhecido é Checkit. Ambos fazem medidas e apresentam resultados comparativos com o IBM PC XT. A tabela que se segue apresenta os índices de velocidades de vários processadores, medidos com o Norton SI 8.0 e com o Checkit 3.0. Hoje os testes de desempenho feitos com esses dois programas são obsoletos, por isso não faz mais sentido usá-los para processadores novos. Apresentamos portanto os índices de velociodade para processadores até o Pentium-200.

Processador e clock	Norton Si 8.0	Checkit 3.0
Pentium-200	640	460
Pentium-166	525	380
Pentium-150	475	340
Pentium-133	420	300
Pentium-120	380	273
Pentium-100	317	228
Pentium-90	285	205
Pentium-75	235	170
Pentium-66	209	150
Pentium-60	190	136
486DX4-120	261	166
486DX4-100	218	139
486DX2-80	174	114
486DX4-75	163	105
486DX2-66	144	91.8

486DX2-50	108	69.5
486DX-40	87.0	57.0
486DX-33	72.0	45.9
486DX-25	54.0	34.7
386DX-40	43.2	31.6
386DX-33	35.6	26.1
386DX-25	27.0	19.8
386DX-20	21.6	15.8
386SX-40	40.6	25.1
386SX-33	33.5	20.7
386SX-25	25.4	15.7
386SX-20	20.3	12.6
386SX-16	16.2	10.0
286-25	18.4	13.9
286-20	14.7	11.1
286-16	11.8	8.9
286-12	8.8	6.7
286-10	7.4	5.6
286-8	5.9	4.4
8088 10 Mhz	2.1	2.1
8088 8 Mhz	1.7	1.7

Observando esta tabela, notamos um fato muito interessante que ocorre com os programas medidores de desempenho. Observe que os índices são iguais quando dizem respeito ao 8088. Em todos os outros processadores, o Norton SI e o Checkit encontram índices diferentes. Porque esses índices são diferentes? Qual dos dois está correto?

Os índices são diferentes porque esses dois programas usam "miscelâneas de instruções" diferentes. Nos processadores mais modernos, as multiplicações e divisões são incrivelmente mais rápidas que nos antigos. Entretanto, operações que realizam acessos à memória são penalizadas pelo fato das memórias não serem tão velozes quanto os processadores exigem. Um programa de benchmark que utiliza operações de multiplicação e divisão tende a apresentar índices muito mais altos que outro que realiza muitos acessos à memória. O resultado é que um processador pode ser muito veloz ao processar programas que fazem muitos cálculos, mas pode não ser tão veloz quanto executa programas que manipulam grandes quantidades de dados na memória.

Hoje em dia faz pouco sentido medir o desempenho usando programas que fazem comparação com o IBM PC XT. O fato de um Pentium-166 apresentar o índice 525 medido com o Norton SI não significa que ele realmente será 525 vezes mais veloz que o XT para qualquer tipo de processamento. Por exemplo, quando desabilitamos a memória cache L2 em um PC equipado com um Pentium-166, ele continua apresentando um índice de 525 medido pelo Norton SI, graças à eficiência da sua cache interna. Esta

eficiência não é tão grande assim quando é preciso acessar grandes quantidades de memória. A cache L1 não consegue dar conta do serviço, e o desempenho cai consideravelmente.

Sem a cache L2, um Pentium-166 apresenta um desempenho similar ao de um Pentium-90, apesar do seu índice de velocidade medido com o Norton SI (ou com o Checkit) permanecer inalterado. Para medir de forma mais realista o desempenho do Pentium e processadores mais avançados, é preciso usar programas que são baseados na execução de uma miscelânea de instruções mais comuns nos programas mais sofisticados para o ambiente Windows. Exemplos de programas adequados são o Norton Sysinfo para Windows 9x, o Winbench e o Winstone.

A tabela que se segue mostra índices de velocidade para alguns processadores na faixa de 200 a 500 MHz. Nesses testes usamos os programas Norton Sysinfo para Windows 9x e o Winbench 99, que apresenta por sua vez dois índices: CPUMark32, para processamento não numérico, e o FPUWinMark, para processamento numérico. Observe como processadores de gerações mais novas apresentam desempenho muito superior ao de processadores mais antigos porém com o mesmo clock. Por exemplo, o Pentium II/233 tem índice CPUMark32 igual a 560, enquanto o Pentium MMX/233 tem índice de apenas 440.

Processador e clock	Norton	CPUMark32	FPUWinMark
Pentium II, 450 MHz	210	1100	2290
Pentium II, 400 MHz	190	1000	2060
Pentium II, 350 MHz	170	900	1800
Pentium II, 300 MHz	150	800	1500
Pentium II, 333 MHz	150	780	1600
Pentium II, 300 MHz	140	730	1500
Pentium II, 266 MHz	130	650	1300
Pentium II, 233 MHz	110	560	1150
Pentium MMX, 233 MHz	61	440	830
Pentium MMX, 200 MHz	55	400	750
AMD K6-2, 400 MHz	140	860	1250
AMD K6-2, 350 MHz	130	800	1100
AMD K6-2, 300 MHz	120	760	950
AMD K6, 300 MHz	110	620	930
AMD K6, 266 MHz	100	580	850
AMD K6, 233 MHz	90	550	750
AMD K6, 200 MHz	80	520	640
Celeron 300 MHz	100	610	1500
Cyrix MII PR300	85	560	520

6x86MX PR266	75	540	460
6x86MX PR233	64	470	420
6x86MX PR200	60	430	390
6x86 PR200	52	420	380

Processadores mais com clocks mais elevados apresentam índices de desempenho ainda maiores. A tabela que se segue mostra os índices CPUMark32, medidos com o programa Winbench 99 versão 1.2, para alguns processadores acima de 500 MHz. Note que os índices do Winbench 99 versão 1.2 não têm relação com os índices do Winbench 99 versão 1.0, usados na tabela anterior.

Processador e clock	CPUMark32 (ver 1.2)
Athlon 1 GHz	90
Athlon 800 MHz	72
Athlon 600 MHz	55
Pentium III 1 GHz	85
Pentium III 800 MHz	70
Pentium III 600 MHz	45
Duron 800 MHz	65

Unidade de ponto flutuante

Os velhos processadores 8086 e 8088 podiam operar em conjunto com um chip auxiliar chamado 8087. Este chip era chamado de processador (ou coprocessador) matemático. Era uma espécia de processador secundário, especializado em realizar cálculos com números reais em alta velocidade. Enquanto o 8086 e o 8088 faziam apenas adição, subtração, multiplicação e divisão de números inteiros de 32 bits, o 8087 podia realizar essas mesmas operações, e ainda uma grande quantidade de funções algébricas (raiz quadrada, logaritmo, exponencial, etc), trigonométricas (seno, tangente, arco tangente, etc) e hiperbólicas (seno hiperbólico, cosseno hiperbólico, etc), com números reais de 80 bits de mantissa (lembrando que um número real pode ser representado por uma base, ou mantissa, e um expoente). Programas que utilizam grandes quantidades de cálculos deste tipo ficavam incrivelmente mais velozes quando usavam o 8087. Normalmente, os softwares eram fornecidos simultaneamente em duas versões, uma para operar através do 8086/8088, e outra para usar o 8087. Quando o PC não tinha o 8087 instalado, mesmo assim podia realizar esses cálculos, mas estes eram feitos por etapas, o que era muito mais demorado. Os programas que se beneficiam de um coprocessador matemático são os seguintes:

- CAD (Computer Aided Design)

- Programas para engenharia
- Programas científicos
- Programas que geram figuras tridimensionais

Ao lançar os seus novos processadores, a Intel sempre lançava também um coprocessador matemático compatível:

Processador	Coprocessador matemático
8086 / 8088	8087
80286	80287
80386SX	80387SX
80386DX	80387DX

Ao lançar o 486, a Intel finalmente colocou o coprocessador matemático dentro de próprio processador. O chamado 486DX possui um coprocessador matemático interno, enquanto o 486SX não o possui. Outros processadores mais avançados como o Pentium e o Pentium Pro também possuem o coprocessador matemático interno. O mesmo ocorre com todos os processadores produzidos depois do 486, ou seja, todos possuem um coprocessador matemático embutido. Esta parte do processador é chamada atualmente de unidade de ponto flutuante (FPU, ou Float Point Unit).

Antigamente apenas engenheiros, arquitetos e cientistas precisavam de um coprocessador matemático. No tempo em que reinava o processador 486, a sua unidade de ponto flutuante ficava praticamente ociosa, pois os softwares da época quase não a utilizavam. Hoje em dia, além das aplicações sérias já citadas, existe uma categoria de programas que faz uso intensivo da unidade de ponto flutuante: os jogos tridimensionais. A geração de imagens tridimensionais demanda uma grande quantidade de cálculos, portanto a unidade de ponto flutuante passou a ser um ítem essencial, mesmo para os usuários domésticos.

Mapas de memória e de E/S

Um bom conhecedor de hardware deve entender não apenas o que se passa dentro de um processador, mas também a forma como ele se comunica com o seu exterior. É preciso entender como o processador envia e recebe dados para a memória e para os dispostivos a ele ligados. Vamos então começar estudando a forma como o processador "vê" a memória e os demais dispositivos.

Como vimos no início deste capítulo, um processador é capaz de realizar operações como:

- Ler um dado da memória
- Escrever um dado na memória
- Receber um dado de dispositivos de E/S
- Enviar dados para dispositivos de E/S

De um modo geral podemos dizer que o processador é capaz de ler e escrever dados em duas categorias de circuitos:

a) Memória: São as ROMs e RAMs localizadas na placa de CPU e nas placas de expansão.

b) Entrada e saída: Em inglês "Input/Output" (I/O). São circuitos representados pelas interfaces de diversos dispositivos como drives, disco rígido, teclado, impressora, monitor, mouse, etc.

Nas operações de acesso à memória, o processador escreve e lê dados, praticamente sem intermediários. Nos acessos a dispositivos de E/S, existem circuitos intermediários, que são as interfaces. Por exemplo, quando é feita a leitura de um caracter proveniente do teclado, não existe uma ligação direta entre o processador e o teclado. Esta ligação é feita por um circuito chamado Interface de Teclado (esta interface fica localizada na placa de CPU). O código do caracter proveniente do teclado é transferido para esta interface, que por sua vez, avisa ao processador que existe um código para ser lido. O processador pode então fazer a leitura do caracter ou comando de teclado recebido. Da mesma forma, quando é feita a impressão de um caracter na impressora, o processador não envia dados diretamente para a impressora. Os dados são colocados em um circuito chamado Interface Paralela, que por sua vez, encarrega-se de transmitir os dados para a impressora.

Cada dispositivo de E/S possui a sua própria interface, como mostram os exemplos da tabela a seguir:

Dispositivo	Interface
Monitor	Placa de vídeo
Teclado	Interface de teclado
Alto falante	Interface de alto falante
Impressora	Interface paralela ou USB
Mouse	Interface serial, PS/2 ou USB
Drive de disquete	Interface para drives de disquete
Disco rígido IDE	Interface IDE
Disco rígido SCSI	Interface SCSI
Joystick	Interface para jogos ou USB
Scanner	Placa de interface de scanner, paralela ou USB
Câmera digital	Interface serial, paralela ou USB

ZIP Drive	Existem modelos SCSI, paralelos, USB e IDE.

Para controlar um dispositivo de E/S, o processador precisa realizar acessos de leitura e escrita na sua interface. Observe que quando o processador escreve dados na memória, nada acontece fisicamente. Apenas o valor escrito fica armazenado na posição de memória que o processador indicou. Por outro lado, quando o processador escreve no circuito de uma interface, ações físicas ocorrem.

Para realizar a leitura e escrita de dados na memória e nas interfaces, o processador possui diversos sinais digitais, como mostra a figura 11:

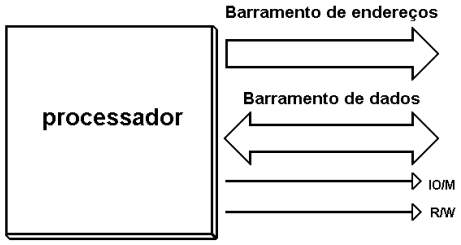


FIGURA 4.11

Barramentos e sinais de controle envolvidos nas operações de leitura e escrita do processador.

a) Barramento de endereços

Em inglês, Address BUS. Nos processadores 386DX e no 486 este barramento é um conjunto de 32 sinais digitais, representados por 32 terminais do processador. Na maioria dos processadores mais avançados este barramento opera com 36 bits. Através desses sinais o processador especifica o endereço de memória ou de E/S que deseja ter acesso. Este barramento é do tipo unidirecional, ou seja, os valores que representa trafegam em uma única direção, que é a indicada pela seta.

b) Barramento de dados

Em inglês, Data BUS. Nos processadores 386DX e 486, este barramento possui 32 bits, e é representado também por 32 terminais do processador. No Pentium e superiores, possui 64 bits. É através deste barramento que trafegam os dados que o processador lê e escreve na memória e nas interfaces. Este barramento é do tipo bidirecional, ou seja, seus dados podem trafegar em duas direções: para dentro do processador (nas operações de leitura) e para fora do processador (nas operações de escrita).

c) IO/M

Significa Input-Output/Memory, ou seja, Entrada e Saída / Memória. Com este sinal digital o processador indica se está acessando uma posição de memória ou uma posição de E/S.

d) R/W

Significa Read/Write, ou seja, Leitura/Escrita. Este é um sinal digital através do qual o processador informa se está realizando uma operação de leitura ou de escrita.

Através dos sinais IO/M e R/W, o processador define uma das 4 operações de transferência de dados possíveis:

- Leitura da memória
- Leitura de E/S
- Escrita na memória
- Escrita em E/S

Um exemplo de leitura de E/S é a recepção do código de uma tecla pressionada pelo usuário no teclado. Um exemplo de escrita em E/S é a transmissão de um caracter para a impressora.

Observe bem a figura 11 e veja como existe uma grande similaridade (pelo menos do ponto de vista do processador) entre as operações de acesso à memória e as operações de acesso a E/S. Em ambos os tipos de operação o processador precisa fornecer o endereço desejado. Em ambos os tipos podem ser feitas leituras e escritas através do barramento de dados. O sinal R/W indica se a operação é de leitura ou escrita, tanto no caso do acesso à memória como no acesso a E/S. O sinal IO/M é o único diferenciador que o processador fornece para distinguir entre as operações de acesso à memória e as de acesso a E/S.

O processador "enxerga" a memória como uma grande seqüência de bytes. Esta seqüência, quanto representada em um gráfico, é chamada de mapa de memória. Da mesma forma, o processador "enxerga" os dispositivos de E/S como uma seqüência de bytes, que ao serem representados graficamente, formam o que chamamos de "mapa de E/S". O mapa de memória é uma representação gráfica de todos os bytes presentes em todos os chips de memória do computador. O mapa de E/S é uma representação gráfica de todos os bytes existentes nos diversos chips que formam as diversas interfaces existentes no computador. A figura 12 mostra o mapa de memória e o mapa de E/S de um PC equipado com 256 MB de RAM.

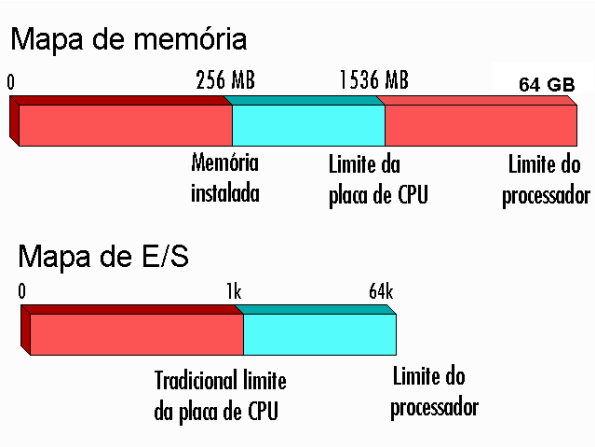


FIGURA 4.12
Mapa de memória e mapa de E/S.

Observe o mapa de memória da figura 12. Vemos que neste mapa existem vários "finais de memória".

1. Memória instalada. Na figura está sendo usado um limite de 256 MB, mas poderia ser qualquer quantidade suportada pela placa de CPU. Esta é a memória que os programas poderão acessar.

2. Limite da placa de CPU. Este limite é imposto pelo projetista da placa de CPU, que possui previsão para a instalação futura de novos módulos de memória. Muitas placas de CPU modernas possuem 3 soquetes para instalação de módulos de até 512 MB, portanto seu limite máximo é 1536 MB. Outras placas poderão ter limites ainda maiores, assim como placas um pouco mais antigas têm limites menores.

3. Limite do processador. Com um barramento de endereços de 36 bits, o máximo endereço que pode ser utilizado é 64 GB de memória.

A figura 12 mostra também o mapa de E/S e dois limites:

1. Limite da placa de CPU. Este limite foi imposto pela IBM quando projetou o IBM PC, e era seguido pelas placas de CPU até poucos anos atrás. Nessas palcas são usados apenas 1024 endereços de E/S (1k), apesar do processador poder chegar até 64k. Na figura, chamamos este limite de "Tradicional Limite da Placa de CPU", pois nas modernas placas de CPU, este limite é maior, ou seja, é usado um espaço de endereçamento maior que 1 kB.

2. Limite do processador. Nas operações de E/S, os processadores usados na família PC usam apenas 16 bits do seu barramento de endereços, limitando o endereço máximo de E/S em 64k. As placas de CPU modernas utilizam todos os 16 bits para especificar endereços de E/S, portanto seu limite máximo é o próprio limite de endereçamento do processador, ou seja, 64 kB.

Os bytes do mapa de E/S ficam localizados em diversos chips existentes nas diversas interfaces instaladas no PC. Quando o processador escreve valores nesses bytes, os dispositivos conectados às interfaces recebem automaticamente os comandos correspondentes a esses valores.

Através do Gerenciador de Dispositivos do Windows, podemos visualizar o mapa de E/S, com as indicações das faixas de endereços ocupadas por cada interface. No Gerenciador de Dispositivos, clique em Computador, depois em Propriedades. No quadro apresentado (figura 13), marque a opção "Entrada/saída (E/S)".

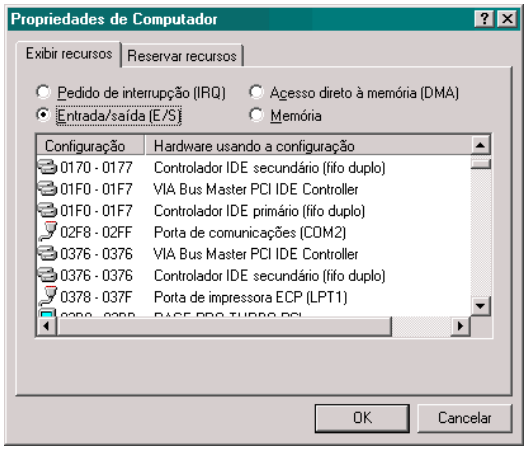


FIGURA 4.13

Mapa de E/S apresentado pelo Gerenciador de Dispositivos do Windows.

No mapa apresentado, vemos por exemplo que a porta COM2 ocupa os endereços entre 2F8 e 2FF, e que a porta paralela está ocupando os endereços entre 378 e 37F. Usando a barra de rolamento deste quadro podemos visualizar todos os endereços de E/S usados pelo computador.

Você certamente já ouviu falar em drivers de dispositivos de E/S. Temos por exemplo o driver da placa de som, o driver da placa de vídeo, o driver do modem, e assim por diante. Uma das coisas que o driver faz é ler e escrever valores apropriados nos endereços apropriados do mapa de E/S correspondentes ao dispositivos que está controlando.

Ao projetar o IBM PC, a IBM definiu diretrizes para o uso do mapa de E/S. Esta definição é uma reserva de faixas de endereços específicos para interfaces específicas. Todos os fabricantes de hardware para PCs devem obedecer este padrão. Por exemplo, em qualquer placa que tenha uma interface serial COM1, esta interface deve ocupar os endereços entre 3F8 e 3FF. A tabela que se segue mostra como a IBM definiu esses endereços de E/S. Até os dias atuais as interfaces mantém esses endereços, por questões de compatibilidade.

Endereços	Interface que os utiliza
000-01F	Controlador de DMA (placa de CPU)
020-03F	Controlador de interrupções (placa de CPU)
040-05F	Timer (placa de CPU)
060-06F	Controlador de teclado do AT
070-07F	Chip CMOS
080-09F	Registro de página de DMA (placa de CPU)
0A0-0BF	Segundo controlador de interrupções (CPU)
0C0-0DF	Segundo controlador de DMA (placa de CPU)
0F0-0F1	CLEAR e RESET do coprocessador
170-177	Controladora IDE secundária
1F0-1F7	Controladora IDE primária
200-207	Interface de joystick
278-27F	Porta paralela
2E8-2EF	Porta serial COM4
2F8-2FF	Porta serial COM2
370-377	Interface de drives secundária
378-37F	Porta paralela
3B0-3BF	Placa de vídeo MDA e HÉRCULES
3C0-3CF	Placa VGA
3D0-3DF	Placas CGA e VGA
3E8-3EF	Porta serial COM3
3F0-3F7	Interface de drives primária
3F8-3FF	Porta serial COM1

Interrupções

As interrupções são um método bastante eficiente para realizar operações de entrada e saída. Imagine uma situação da vida real em que uma secretária fica o tempo todo ao lado do chefe, esperando que ele solicite um serviço. Ela não pode executar outras tarefas porque está "monitorando" o seu chefe. Imagine agora que a secretária está realizando normalmente o seu trabalho, sem se preocupar com o chefe. Quando o chefe deseja algum serviço, chama a secretária, que irá interromper o que estava fazendo para atender o chefe. Da mesma forma, um processador não precisa ficar constantemente monitorando os seus dispositivos de E/S. Pode fazer o seu trabalho normalmente, e quando um dispositivo necessitar de atenção do processador, irá interrompê-lo para que a operação de E/S seja realizada.

A interrupção é uma operação de hardware na qual o processador suspende provisoriamente a execução de um programa para o atendimento de um determinado evento. Essa suspensão dura tão pouco que o usuário não chega a perceber que o programa parou. Na maioria dos casos este tempo é inferior a alguns milésimos de segundo. Entre os diversos pinos do processador, um deles é chamado de INT, e serve para que os diversos circuitos existentes no computador possam requisitar interrupções. Assim que o processador recebe um sinal INT, guarda na memória informações que permitem mais tarde saber exatamente onde parou. A seguir, determina qual foi o dispositivo que gerou a interrupção e faz o seu atendimento. Ao terminar de atender a interrupção, volta a processar o programa original exatamente do ponto de onde parou. Existem diversos dispositivos que necessitam interromper o processador. Alguns exemplos são:

- A interface de teclado, para avisar que o usuário pressionou uma tecla
- A interface paralela, para avisar que ocorreu algum problema com a impressora
- A interface serial, para indicar que recebeu um byte, ou que terminou de transmitir um byte
- A interface de drives de disquetes, para avisar que já terminou a operação de leitura ou gravação solicitada
- Idem para a interface de disco rígido

Sem as interrupções, o funcionamento do computador seria muito mais complicado. Além de processar o programa principal, o processador precisaria periodicamente checar todos os dispositivos para verificar se existem eventos pendentes. Por exemplo, precisaria checar se alguma tecla foi pressionada, se a impressora está pronta para receber dados, se existe algum dado chegando das interfaces seriais, etc. Alguns computadores muito antigos operavam desta forma, uma técnica chamada de pooling. Era o caso da maioria dos micros de 8 bits. O uso de interrupções faz com que o computador opere de forma muito mais eficiente, podendo se ocupar do processamento do programa principal, e sendo interrompido apenas quando necessário.

Vários dispositivos, através das suas interfaces, precisam freqüentemente enviar um comando de interrupção para que o processador lhes dedique a atenção necessária, normalmente relacionada com a transmissão e recepção de dados. Como o processador possui apenas uma entrada de interrupção e existem vários dispositivos que necessitam interrompê-lo, a placa de CPU

utiliza um circuito chamado de controlador de interrupções. Uma das funções deste circuito é receber requisições de interrupções de vários dispositivos e interromper o processador através do sinal INT. Outra função é informar ao processador qual foi o dispositivo que gerou a interrupção.

A figura 14 mostra de forma muito simplificada, a estrutura de interrupções no IBM PC XT. A figura 15 mostra a estrutura de interrupções usada nos PC mais modernos. Por simplicidade, comecemos a analisar como eram as interrupções no XT.

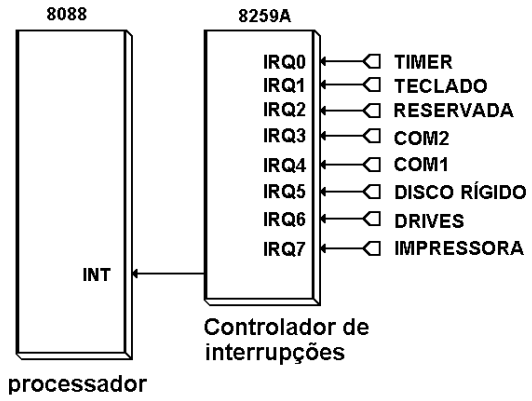


FIGURA 4.14

Uso das interrupções no IBM PC XT.

Os vários circuitos que precisam gerar interrupções enviam as requisições ao chip controlador de interrupções, que por sua vez, interrompe o processador e o informa qual foi o dispositivo que requisitou a interrupção.

Observe na figura 14 os seguintes sinais digitais:

INT	Sinal que serve para interromper o processador.
IRQ0 a IRQ7	Essas são as oito entradas de interrupções, ligadas a diversos dispositivos. A sigla IRQ significa "Interrupt Request" (Requisição de interrupção). Quando o controlador de interrupções recebe um IRQ de algum dispositivo, gera um sinal INT para o processador. Além disso, o controlador de interrupções informa ao processador qual dos oito sinais IRQ foi ativado. Este controlador também é capaz de gerenciar prioridades entre as interrupções (qual interrupção é atendida em primeiro lugar quando duas ou mais ocorrem no mesmo instante), e também leva em conta interrupções aninhadas (quando uma nova interrupção ocorre antes do final do atendimento de uma prévia interrupção). Entre os diversos chips controladores de interrupções existentes no mercado, a IBM optou pelo 8259A, fabricado pela Intel.

Para que todos esses dispositivos possam gerar interrupções, é preciso que suas interfaces tenham acesso físico aos respectivos sinais IRQ. Observe que tanto o processador como os controladores de interrupções ficam localizados

na placa de CPU. Por outro lado, a maioria das interfaces ficam localizadas em placas de expansão. Portanto, os sinais IRQs são originados em placas de expansão e precisam chegar até a placa de CPU. Por esta razão, os sinais IRQ estão presentes nos slots, que são a ligação física entre a placa de CPU e as placas de expansão.

As oito entradas de interrupções disponíveis são descritas na tabela que se segue:

IRQ0	Timer. Trata-se de um circuito que gera uma interrupção a cada 55 milésimos de segundo. Esta interrupção é usada para manter a data e a hora. Nos XTs, o usuário precisava fornecer a data e a hora durante o boot. Depois disso, o timer fazia a contagem do tempo a partir da data e hora iniciais, e das suas interrupções. Nos PCs modernos, apesar da existência do relógio permanente localizado no chip CMOS, o timer continua sendo usado pelos programas que necessitam saber a data e a hora. A única diferença é que nas operações de boot o timer não precisa mais ser acertado pelo usuário, pois este acerto é feito automaticamente pelo BIOS a partir da data e hora existentes no chip CMOS.
IRQ1	Teclado. É gerada pelo chip de interface de teclado sempre que o usuário pressiona alguma tecla.
IRQ2	Reservado. Inicialmente a IBM deixou esta interrupção reservada para uso futuro, e não fez nela nenhuma conexão. Entretanto, diversas placas de interface passaram a usá-la, já que estava disponível, apesar da recomendação da IBM de reservá-la para uso futuro.
IRQ3	COM2. É usada pela segunda interface serial, para sinalizar o final da transmissão e da recepção de dados. A cada byte recebido pela interface serial, uma interrupção é gerada. Ocorre também no final da transmissão de cada byte, indicando ao processador que o próximo byte já pode ser transmitido.
IRQ4	COM1. Tem o mesmo uso da IRQ3, porém é usada pela interface serial COM1.
IRQ5	Disco rígido. As placas controladoras de disco rígido para XT usavam esta interrupção para indicar a finalização de operações de acesso ao disco rígido. Assim o processador saberia que é hora de enviar o próximo comando. Entre essas operações podemos citar: Leitura de setor, gravação de setor, posicionamento sobre uma trilha, formatação de trilha, etc.
IRQ6	Drive de disquetes. Esta interrupção era, e ainda é usada pela interface de drives de disquetes. Serve para sinalizar o término de operações de acesso ao disquete, como leitura, gravação, posicionamento, formatação, etc.
IRQ7	Impressora. Através desta interrupção, a impressora pode informar a ocorrência de erros (falta de papel, carro de impressão preso, etc). É também usada para controlar o fluxo de dados entre o computador e a impressora. Quando a impressora está com o seu buffer cheio, gera esta interrupção para informar esta condição, fazendo com que o computador suspenda a transmissão de dados. Quando o buffer fica parcialmente descarregado, gera outra interrupção para informar que o computador já pode enviar mais dados.

Com o lançamento do IBM PC AT, equipado com o processador 80286, a IBM passou a utilizar dois controladores de interrupções ligados em cascata, como mostra a figura 15. Este arranjo continua sendo utilizado da mesma forma, com mínimas modificações, nos PCs atuais.

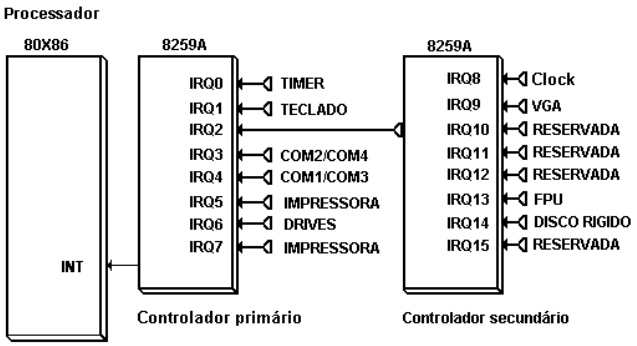


FIGURA 4.15

Uso de interrupções nos PCs modernos.

As interrupções passaram a ser usadas da seguinte forma:

IRQ0	Timer. Mesmo uso que tinha no XT.
IRQ1	Teclado. Mesmo uso que tinha no XT.
IRQ2	CASCADE. Ligação com o segundo controlador de interrupções (ligação em cascata).
IRQ3	COM2 / COM4. A IBM aumentou o número de portas seriais para um máximo de quatro, mas não reservou interrupções exclusivas para a COM3 e a COM4. A COM4 deve usar a mesma interrupção que a COM2, enquanto a COM3 deve usar a mesma interrupção que a COM1. Esta é uma das principais razões de conflitos de hardware envolvendo as interfaces seriais.
IRQ4	COM1 / COM3.
IRQ5	Paralela. A IBM reservou esta interrupção para a segunda interface paralela.
IRQ6	Interface de drives. Mesmo uso do XT.
IRQ7	Paralela. Mesmo uso do XT. Normalmente esta interrupção é usada pela primeira interface paralela (LPT1), ficando a IRQ5 para a LPT2. Entretanto, nem sempre esta regra é seguida. Existem placas em que a LPT1 está ligada na IRQ5.
IRQ8	Alarm clock. Esta interrupção está ligada ao chip CMOS, que pode ser programado para gerar uma interrupção após um período pré-programado.
IRQ9	EGA / VGA. Originalmente esta interrupção era utilizada pela placa de vídeo EGA, que deu lugar às placas VGA. As placas VGA podem opcionalmente, por questão de compatibilidade com a EGA, usar também esta interrupção.
IRQ10	Reservado. A IBM nunca diz "livre", e sim, "reservado". Normalmente esta interrupção está livre, e pode ser usada por novas placas, como por exemplo, placas de rede e placas de som.
IRQ11	Reservado. Mesmo caso da IRQ10.
IRQ12	Reservado. Mesmo caso da IRQ10.
IRQ13	Coprocessador matemático. Esta interrupção é reservada para uso exclusivo do coprocessador matemático. Através dela o processador pode ser informado sobre condições anormais do cálculo, como por exemplo, a divisão por zero e a raiz quadrada de um número negativo.
IRQ14	Disco rígido. No IBM PC XT, a interface do disco rígido usava o IRQ5. Nos ATs, esta foi substituída pelo IRQ14, ficando a IRQ5 destinada à segunda interface paralela.
IRQ15	Reservado. Mesmo caso da IRQ10.

Ao lançar o IBM PC AT, a IBM passou a utilizar não apenas um número maior de interrupções, mas também um número maior de bits de dados e de endereços. Veja algumas diferenças entre os modelos XT e AT:

	XT	AT
Bits de dados	8	16
Bits de endereços	20	24
Número de IRQs	8	15

Canais de DMA	4	7

Os bits de dados, endereços, linhas de IRQ e canais de DMA (mais adiante estudaremos o DMA) são ligados às placas de expansão através dos slots. Como o AT possuía mais bits de endereços, dados, sinais de IRQ e DMA que o XT, a IBM teve que aumentar os seus slots. O XT usava os slots "ISA de 8 bits", e o AT passou a usar os slots "ISA de 16 bits".

B1	Terra	A1	I/O channel Check
B2	Reset Driver	A2	Data 7
B3	+5 volts	A3	Data 6
B4	IRQ2	A4	Data 5
B5	-5 volts	A5	Data 4
B6	DMA REQ 2	A6	Data 3
B7	-12 volts	A7	Data 2
B8	Card select	A8	Data 1
B9	+ 12 volts	A9	Data 0
B10	Terra	A10	I/O Channel Ready
B11	Memory Write	A11	Address Enable
B12	Memory Read	A12	Address 19
B13	I/O Write	A13	Address 18
B14	I/O Read	A14	Address 17
B15	DMA ACK 3	A15	Address 16
B16	DMA REQ 3	A16	Address 15
B17	DMA ACK 1	A17	Address 14
B18	DMA REQ 1	A18	Address 13
B19	DMA ACK 0	A19	Address 12
B20	Clock	A20	Address 11
B21	IRQ7	A21	Address 10
B22	IRQ6	A22	Address 9
B23	IRQ5	A23	Address 8
B24	IRQ4	A24	Address 7
B25	IRQ3	A25	Address 6
B26	DMA ACK 2	A26	Address 5
B27	Terminal Count	A27	Address 4
B28	Address Latch Enable	A28	Address 3
B29	+5 volts	A29	Address 2
B30	OSC	A30	Address 1
B31	Terra	A31	Address 0

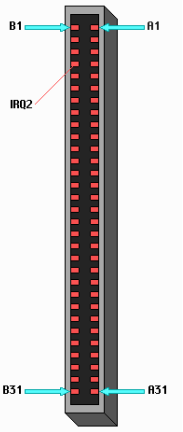


FIGURA 4.16

Sinais de um slot ISA de 8 bits.

A figura 16 mostra um slot ISA de 8 bits. Seus sinais são numerados como A1, A2... A32 (parte direita) e B1, B2, ... B31 (parte esquerda). Com exceção do IRQ0 e IRQ1, todos os outros sinais de interrupção estão presentes neste slot. O IRQ0, como sabemos, está ligado ao Timer, um circuito localizado na placa de CPU e que não é usado por placas de expansão. O IRQ1 está ligado na interface de teclado, também localizada na placa de CPU. Como esses dois circuitos nunca ficam localizados em placas de expansão, não há necessidade da sua presença nos slots. As linhas de IRQs ficam localizadas nos pinos B4, B21, B22, B23, B24 e B25.

B1	Terra	A1	I/O channel Check
B2	Reset Driver	A2	Data 7
B3	+5 volts	A3	Data 6
B4	IRQ9	A4	Data 5
B5	-5 volts	A5	Data 4
B6	DMA REQ 2	A6	Data 3
B7	-12 volts	A7	Data 2
B8	Zero Wait State	A8	Data 1
B9	+ 12 volts	A9	Data 0
B10	Terra	A10	I/O Channel Ready
B11	Real Memory Write	A11	Address Enable
B12	Real Memory Read	A12	Address 19
B13	I/O Write	A13	Address 18
B14	I/O Read	A14	Address 17
B15	DMA ACK 3	A15	Address 16
B16	DMA REQ 3	A16	Address 15
B17	DMA ACK 1	A17	Address 14
B18	DMA REQ 1	A18	Address 13
B19	REFRESH	A19	Address 12
B20	Clock	A20	Address 11
B21	IRQ7	A21	Address 10
B22	IRQ6	A22	Address 9
B23	IRQ5	A23	Address 8
B24	IRQ4	A24	Address 7
B25	IRQ3	A25	Address 6
B26	DMA ACK 2	A26	Address 5
B27	Terminal Count	A27	Address 4
B28	Address Latch Enable	A28	Address 3
B29	+5 volts	A29	Address 2
B30	OSC	A30	Address 1
B31	Terra	A31	Address 0

D1	Memory 16-bit Select	C1	System Bus High Enable
D2	I/O 16-bit Select	C2	Unlatched Address 23
D3	IRQ10	C3	Unlatched Address 22
D4	IRQ11	C4	Unlatched Address 21
D5	IRQ12	C5	Unlatched Address 20
D6	IRQ15	C6	Unlatched Address 19
D7	IRQ14	C7	Unlatched Address 18
D8	DMA ACK 0	C8	Unlatched Address 17
D9	DMA REQ 0	C9	Memory Read
D10	DMA ACK 5	C10	Memory Write
D11	DMA REQ 5	C11	Data 8
D12	DMA ACK 6	C12	Data 9
D13	DMA REQ 6	C13	Data 10
D14	DMA ACK 7	C14	Data 11
D15	DMA REQ 7	C15	Data 12
D16	+5 volts	C16	Data 13
D17	Master	C17	Data 14
D18	Terra	C18	Data 15

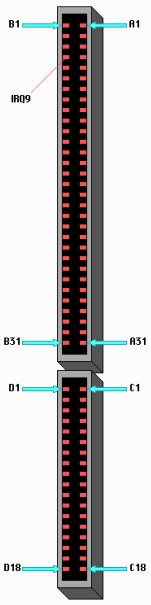


FIGURA 4.17

Sinais de um slot ISA de 16 bits. Observe que as novas linhas de interrupção (IRQ10-IRQ15) ficam localizados no conector menor. Observe ainda que no lugar da IRQ2, no pino B4, temos a IRQ9.

Observe a figura 17, que mostra um slot de 16 bits, típico de micros 286 e superiores, e a descrição de todos os seus sinais. Note que mesmo placas de CPU de fabricação recente, equipadas com processadores de última geração, mas que tenham slots ISA de 16 bits, seguem este mesmo padrão, herdado do IBM PC AT. A figura mostra que no conector maior estão presentes os mesmos IRQs encontrados no XT, exceto o IRQ2. No XT, o IRQ2 era ligado no pino B4 do slot (veja a figura 17). Nos PCs 286 e superiores, o IRQ2 passou a ser usado internamente pela placa de CPU, realizando a ligação em cascata dos dois controladores de interrupções (figura 15). Para que as antigas placas de expansão que usavam o IRQ2 pudessem continuar funcionando, a IBM colocou exatamente no seu lugar (pino B4) o IRQ9. Em outras palavras, o IRQ9 faz nos PCs 286 e superiores o mesmo papel que fazia o IRQ2 nos antigos micros XT. Por isso, muitos manuais costumam usar os termos "IRQ2" e "IRQ9" como sinônimos.

O uso da IRQ9

Cabe aqui chamar a atenção para um detalhe muito importante a respeito do uso da IRQ9. Originalmente esta interrupção era utilizada por placas de vídeo EGA, sinalizando um evento chamado retraço vertical. Esta sinalização era necessária para evitar um efeito indesejável na tela chamado “snow”.

As placas de vídeo antigas eram muito lentas. Tão lentas que sua memória de vídeo não podia ser simultaneamente acessada pelo processador e pelos circuitos que enviam os sinais para o monitor. Se este acesso fosse feito de forma simultânea, fazia com que surgissem momentaneamente pequenos traços pretos horizontais na tela sempre que o processador precisava colocar dados na memória de vídeo. Este efeito indesejável é chamado de snow. Para evitar este problema, os programas faziam acesso à memória de vídeo apenas durante o retraço vertical, que é o período no qual o feixe eletrônico do monitor atinge a parte inferior da tela e é reposicionado na sua parte superior. Como neste período o feixe eletrônico do monitor é apagado, não ocorre o snow. A placa de vídeo EGA gerava interrupções através do IRQ9 para indicar o início e o fim do retraço vertical, e muitos programas utilizavam este recurso. As modernas placas SVGA são bem mais velozes, e podem ao mesmo tempo enviar sinais de vídeo para o monitor e serem acessadas pelo processador, sem a ocorrência de snow. Por isso os programas atuais não precisam mais esperar pelo retraço vertical para acessá-la.

Por questões de compatibilidade com as placas EGA, as placas VGA e SVGA podem opcionalmente utilizar o IRQ9. Entretanto, a esmagadora maioria dos softwares modernos não necessita desta interrupção. Podemos tranqüilamente deixar o IRQ9 na placa VGA desabilitado. Assim esta interrupção ficará livre para ser usada por novas placas que desejemos instalar. A única desvantagem de desativar o IRQ9 na placa VGA é que programas gráficos bem antigos (criados entre 1985 e 1990, em geral), escritos para a placa EGA, deixarão de funcionar, e o micro "travará" sempre que forem executados. Como é pouco provável que você utilize algum programa desta época, a melhor coisa a fazer é realmente desabilitar o uso da IRQ9 na sua placa VGA. Nas placas VGA antigas, esta desabilitação era feita através de um jumper. Nos PCs modernos, podemos encontrar no CMOS Setup, um comando para ativar ou desativar o uso da IRQ9 para a placa de vídeo.

Uso das IRQs nos PCs atuais

Nos PCs modernos, sejam eles equipados com slots ISA ou não, o uso das interrupções é muito parecido com o que ocorria no IBM PC AT. Por exemplo, o teclado continua usando a IRQ1, o timer continua usando a IRQ0, a IRQ13 continua sendo usada para indicar a ocorrência de cálculos inválidos pela unidade de ponto flutuante. Vejamos quais são as diferenças presentes nos PCs atuais:

IRQ5 e IRQ7	Essas duas interrupções são reservadas para interfaces paralelas. Como na configuração básica existe apenas uma interface paralela, apenas uma dessas interrupções, normalmente a IRQ7, estará sendo usada. A outra, normalmente a IRQ5, estará livre. Tome cuidado, pois em certos casos ocorre exatamente o contrário, ou seja, a IRQ5 está em uso e a IRQ7 está livre. Outro dado interessante é que muitas impressoras podem funcionar sem o uso de interrupções. Portanto em caso de necessidade, podemos configurar o Windows para que não use uma IRQ para a porta parlela, deixando assim mais uma IRQ livre para ser usada por novas placas de expansão.
IRQ9	Esta interrupção poderá estar sendo usada pela placa de vídeo. Podemos desabilitar seu uso através de um jumper (nas placas antigas), de acordo com as instruções existentes no manual da placa de vídeo. Em PCs modernos, pode ser possível desabilitar o uso de interrupções pela placa de vídeo, através de um comando do CMOS Setup.
IRQ10, IRQ11, IRQ12	Essas interrupções estarão livres, já que não são usadas pelos dispositivos que fazem parte da configuração básica de um PC. Poderão ser usadas por placas de expansão, como modems, placas de som, placas de rede, etc. Nos PCs atuais essas interrupções são normalmente destinadas às placas de expansão que estão ligadas ao barramento PCI.
IRQ15	Esta interrupção normalmente é usada pela interface IDE secundária, enquanto a IRQ14 é usada pela interface IDE primária.

Podemos facilmente visualizar o uso das interrupções usando o Gerenciador de Dispositivos do Windows. Para chegar a ele basta clicar em Meu Computador com o botão direito do mouse e no menu apresentado escolher a opção Propriedades. No quadro apresentado clicamos na guia Gerenciador de Dispositivos. Clicamos em Meu Computador e Propriedades, e finalmente marcamos a opção Pedido de interrupção (IRQ). O quadro assumirá o aspecto mostrado na figura 18.

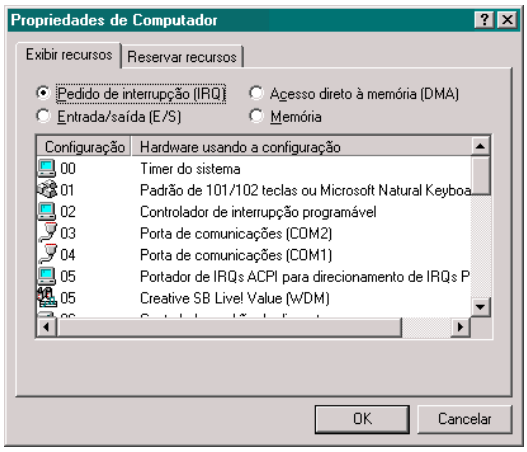


FIGURA 4.18

Relatório de uso das IRQs.

Este relatório informa como as IRQs estão sendo utilizadas, e indica também quais IRQs ainda estão livres para serem usadas em novas placas a serem instaladas.

Compartilhamento de interrupções

A princípio não podemos ter dois dispositivos usando o mesmo recurso de hardware. Quando isto ocorrer, o Gerenciador de Dispositivos colocará um ponto de exclamação sobre os dispositivos em conflito. É o caso da IRQ5 e IRQ10, indicadas na figura 19. O ponto de exclamação indica que pode existir um conflito de hardware, ou então que o dispositivo não está corretamente instalado. Note que este conflito de hardware pode ser devido a IRQ (ambos usariam a mesma IRQ), ou de DMA, ou de endereços de memória, ou de endereços de E/S.

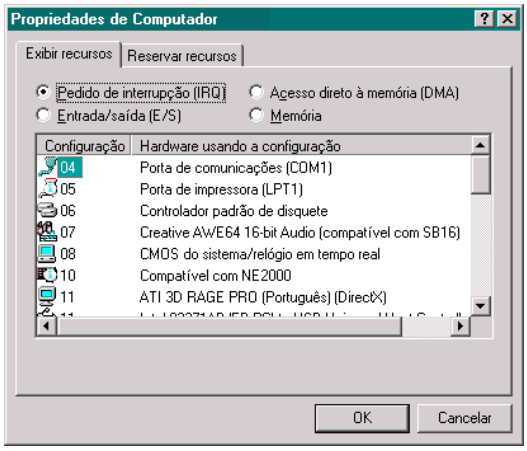


FIGURA 4.19

Dispositivos em conflito de hardware.

Note entretanto que existem casos de dispositivos usarem a mesma IRQ, e mesmo assim não estar ocorrendo conflito. Observe na figura 20 que existem três dispositivos usando a IRQ11, e mesmo assim não existe indicação de conflito. O que isso significa?

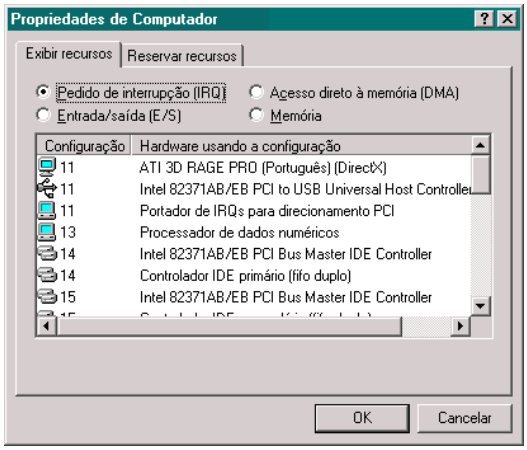


FIGURA 4.20

Interrupções compartilhadas.

Dois dispositivos podem usar a mesma IRQ (pelo menos se considerarmos as IRQs como sendo de IRQ0 a IRQ15) desde que seja usado um "controlador de interrupções terciário". Os chipsets modernos possuem este terceiro controlador, que em geral é diferente dos dois primeiros. Seu uso é mostrado na figura 21. Este controlador terciário é na verdade chamado de roteador de interrupções do barramento PCI.

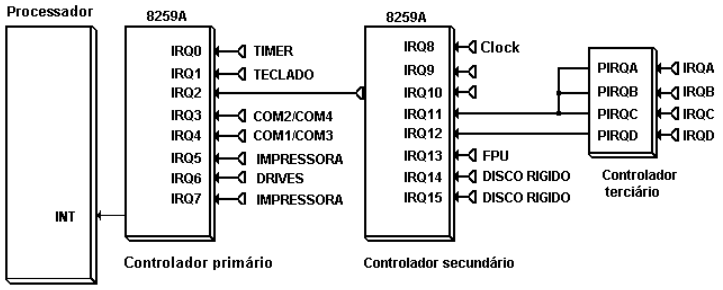


FIGURA 4.21

Interrupções nos PCs atuais, equipados com controlador terciário.

Assim como o controlador de interrupções secundário usa a entrada IRQ2 para gerar novas interrupções (8 a 15), um controlador terciário existente nos chipsets modernos e permite a geração de 4 novas linhas de interrupção, chamadas IRQA, IRQB, IRQC e IRQD. Essas linhas são ligadas nos slots do barramento PCI. Cada uma dessas novas IRQs pode estar conectada diretamente em outras IRQs convencionais, mas podemos ter mais de um deles usando a mesma IRQ. No exemplo da figura 21 temos IRQA, IRQB e IRQC ligadas em IRQ11. Para o Windows, todas essas três entradas estão ligadas em IRQ11, porém isto não é considerado um conflito de hardware, já que o Windows sabe que esses três dispositivos estão conectados neste controlador terciário. Além disso existe mais um fato importante: as interrupções no barramento PCI podem ser compartilhadas, coisa que não era permitida no barramento ISA.

Apesar do Gerenciador de dispositivos não indicar explicitamente os recursos IRQA, IRQB, IRQC e IRQD, não indicará conflitos quando mais de uma dessas IRQs estiver associada à mesma IRQ dos controladores primário e secundário. Fica então caracterizado que não existe conflito de hardware, mesmo que no Gerenciador de Dispositivos estiver indicado que mais de um dispositivo usa a mesma IRQ.

Acesso direto à memória

Vejamos agora outro ponto importante no funcionamento de um PC, que é o DMA (Acesso direto à memória, ou Direct Memory Access). Para entender o funcionamento do DMA, observe inicialmente a figura 22. Estão sendo representadas operações de E/S feitas através do processador, ou seja, sem usar DMA. Em uma operação de saída, o processador obtém da memória o dado a ser transmitido e logo a seguir o envia para a interface, que por sua vez faz com que chegue ao dispositivo de saída. Este é o caso, por exemplo, do funcionamento da interface de impressora. Nas operações de entrada, o dispositivo envia o dado para a sua interface. A seguir o processador lê o dado da interface (a interface poderá usar uma IRQ para avisar o processador que existe um dado pronto para ser lido) e o coloca na memória para que seja posteriormente processado. Este é o caso, por exemplo, do funcionamento da interface de teclado. As operações de entrada e saída nas quais existe um envolvimento direto do processador são chamadas de Entrada e Saída Programada.

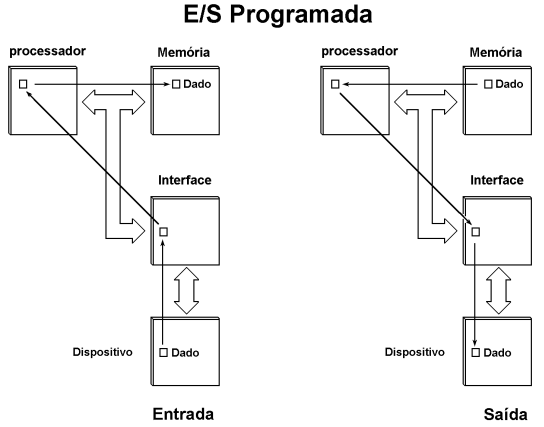


FIGURA 4.22

E/S programada. O processador controla o "transporte" dos dados entre a memória e a interface.

As operações de entrada e saída programada são usadas pela maioria dos dispositivos, mas sua eficiência não é boa quando é necessário transmitir uma grande quantidade de dados em alta velocidade. Nessas condições, o processador precisa ficar exclusivamente dedicado a esta transferência, o que impede que possa realizar qualquer outro processamento e também de realizar E/S em outros dispositivos. Por exemplo, durante a leitura de um setor de um disco rígido IDE, o processador não pode sofrer interrupções e nem transmitir ou receber dados de outros dispositivos que operem com E/S programada. Isto não chega a ser um problema na maioria das vezes, pois normalmente os programas não têm o que processar enquanto não estiverem disponíveis os dados provenientes do disco. Por outro lado, imagine o caso da reprodução de um arquivo sonoro através da placa de som. Se a placa de

som operasse também com E/S programada, não poderia ser usada em conjunto com o disco IDE. Para ouvir um arquivo sonoro seria preciso transferi-lo integralmente para a memória (o que nem sempre é possível no caso de arquivos muito grandes) para depois transferi-lo para a placa de som. Esta operação poderia ser inviabilizada pela limitação no tamanho da memória. O arquivo também não poderia ser lido por partes, pois seu som ficaria com diversas pausas.

Devido às limitações da E/S programada, os PCs podem operar também com um outro tipo de operação de E/S. Trata-se da entrada e saída por DMA. Nessas operações, um circuito especial chamado de controlador de DMA faz o controle dos barramentos do processador. Para receber um dado por DMA, este controlador faz o seguinte:

1. Desabilita momentaneamente o processador, colocando-o em tristate
2. Faz a leitura do dado da interface que requisitou a transferência
3. Grava este dado na posição de memória pré-programada
4. Habilita o processador para funcionamento normal

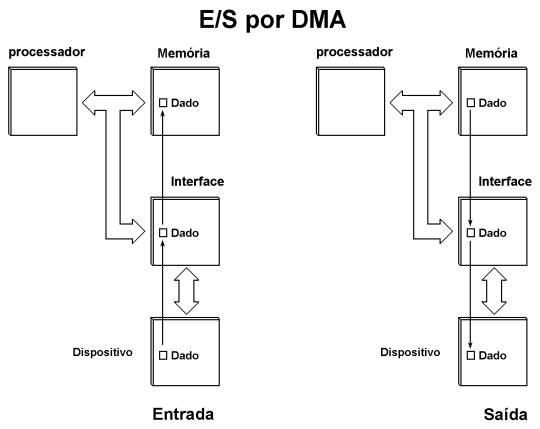


FIGURA 4.23

E/S por DMA. O processador fica em tristate enquanto o controlador de DMA assume o controle dos barramentos e faz as transferências.

Nas operações de saída, o controlador de DMA faz o seguinte:

1. Desabilita momentaneamente o processador, colocando-o em tristate
2. Faz a leitura do dado da memória
3. Transmite o dado para a interface apropriada
4. Habilita o processador para funcionamento normal

As operações de DMA são sempre feitas em blocos. Por exemplo a leitura de um setor vindo do disquete é feita desta forma. O controlador de DMA é

antes programado com o número de bytes a serem recebidos (que neste caso é 512) e com o endereço de memória a partir do qual os dados serão armazenados. O controlador de DMA automaticamente conta o número de bytes recebidos e gera os endereços consecutivos onde os 512 bytes serão armazenados. Podemos ver as operações de E/S por DMA ilustradas na figura 23.

A grande vantagem do DMA é que o processador não precisa se ocupar diretamente da operação de recepção e transmissão de cada byte, ficando livre para executar outros processamentos. Normalmente as interfaces que utilizam DMA, utilizam também uma interrupção para avisar o processador sobre o término da transferência do número de bytes pré-programado.

Entre as interfaces que utilizam DMA podemos citar:

- Interface de drives de disquetes
- Placas controladoras SCSI
- Placas de som
- Placas de interface de scanner
- Placas digitalizadoras de vídeo
- Interface paralela operando no modo ECP

Entre as interfaces que NÃO usam DMA, podemos citar:

- Interfaces seriais
- Interfaces paralelas (exceto quando operam no modo ECP)
- Interfaces para joystick
- Interfaces de teclado

Durante as transferências de DMA, o processador não fica o tempo todo desabilitado. Entre a chegada de dois dados consecutivos de uma transferência, o processador opera normalmente. Suponha que uma determinada interface transmita dados de 1000 em 1000 ns, e que a recepção de cada um desses dados leve 100 ns. Após receber um dado, o processador tem mais 900 ns para processamento normal antes da chegada do próximo dado. Por isso o processador pode realizar, por exemplo, a leitura de um arquivo sonoro do disco rígido e ao mesmo tempo enviá-lo para a placa de som. Ao mesmo tempo em que um trecho do som está sendo tocado, o trecho seguinte estará sendo lido do disco. Isto só é possível porque as placas de som operam com DMA.

Os PCs derivados do IBM PC AT (baseados no 286, 386, 486, Pentium e superiores) podem operar com até 7 dispositivos utilizando DMA. Dizemos então que o circuito controlador de DMA implementa 7 canais de DMA. Na verdade, este circuito é formado por dois controladores de DMA, cada um sendo capaz de gerenciar 4 canais. Esses dois controladores estão ligados em cascata, e um dos canais é utilizado nesta ligação, sobrando apenas 7. Os oito canais e seus usos são os seguintes:

DMA0: Livre
DMA1: Livre
DMA2: Interface de drives
DMA3: Livre
DMA4: [CASCADE]
DMA5: Livre
DMA6: Livre
DMA7: Livre

Um PC que ainda está com a sua configuração básica, possui os canais 0, 1, 3, 5, 6 e 7 livres. À medida que placas de expansão vão sendo instaladas, é preciso escolher canais de tal forma que não ocorram conflitos, ou seja, nunca devemos deixar que duas placas utilizem o mesmo canal de DMA. Placas Plug-and-Play (PnP), quando usam DMA, têm seus canais escolhidos de forma automática pelo Windows, durante o processo de instalação. No caso de placas que não são PnP (modelos antigos), cabe ao usuário fazer a escolha dos canais.

No Windows podemos visualizar os canais de DMA que estão em uso, através do Gerenciador de Dispositivos, como mostra a figura 24. Neste exemplo, além dos canais DMA2 e DMA4, que estão sempre ocupados em qualquer PC, temos ainda os canais DMA1 e DMA5 sendo utilizados pela placa de som.

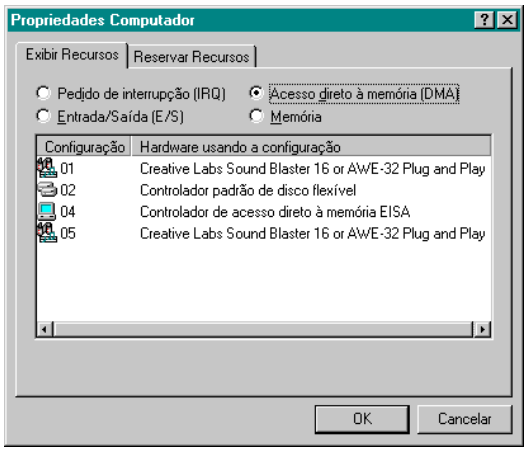


FIGURA 4.24

Visualizando o uso dos canais de DMA com a ajuda do Gerenciador de Dispositivos.

Bus Mastering

O barramento PCI não opera com DMA. Ao invés disso, utiliza um outro método de transferência de dados com características parecidas com o DMA, porém com velocidade muito mais elevada. Este método é o Bus Mastering. Várias interfaces ligadas ao barramento PCI utilizam este recurso, por exemplo:

- Placas de rede
- Placas de som
- Placas de vídeo AGP
- Interfaces IDE
- Interfaces USB
- Controladoras SCSI
- Digitalizadoras de vídeo

Na técnica de Bus Mastering, uma interface qualquer assume o controle do barramento, passando a operar como Master, e envia ou recebe os dados diretamente de uma outra interface ou dispositivo que opera como Target, que pode ser por exemplo, a memória. Enquanto uma transferência está sendo realizada desta forma, o processador fica com o barramento que o liga à memória livre na maior parte do tempo, podendo assim continuar trabalhando ao mesmo tempo em que a transferência é feita. Apresentaremos essas informações em detalhes quando estudarmos o barramento PCI.

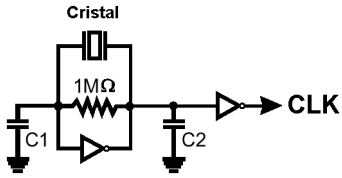
Para efeitos de detecção e eliminação de conflitos de hardware, aqui está uma notícia boa: o Bus Mastering não gera conflitos de hardware, como pode ocorrer com o DMA.

Circuitos de clock e reset

Aproveitando nossos conhecimentos de eletrônica, mostraremos agora como funcionam dois circuitos importantes de uma placa de CPU. São os circuitos de clock e reset. O circuito gerador de clock fornece em sua saída uma onda quadrada com uma freqüência específica. Nas placas de CPU, o processador deve receber uma onda quadrada que representa o seu clock externo. Internamente esta freqüência será multiplicada, resultando no seu clock externo. O circuito de RESET gera um pulso que é enviado para a entrada RESET do processador. Este pulso deve ser ativado quando pressionamos a tecla Reset do gabinete, e também quando ligamos o computador (Power on Reset).

Como funciona um gerador de clock

O cristal de quartzo é o principal componente usado na geração de um clock. Ele tem a capacidade de entrar em ressonância em determinadas freqüências, quando ligado a amplificadores apropriados. A freqüência de ressonância pode ser determinada a partir das dimensões do cristal. Quanto mais fino, mais elevada é a freqüência. A figura 25 mostra um circuito gerador de clock simples, que utiliza um cristal, dois inversores, dois capacitores e um resistor. O sinal de clock gerado por este circuito terá freqüência igual à freqüência de ressonância do cristal. Está fora do escopo deste livro analisar este circuito e provar que ele realmente oscila. Isto exigiria conhecimentos de eletrônica e matemática ainda mais profundos que os propostos neste livro.



*** 35% ***
FIGURA 4.25
Circuito gerador de clock.

O circuito da figura 25 é capaz de oscilar em uma única freqüência. Ele não pode ser usado em casos nos quais é preciso ter um clock variável. Por exemplo, as placas de CPU modernas, para processadores Celeron e Pentium III, devem ser capazes de operar com clocks externos de 66, 100 e 133 MHz. Um oscilador para esta placa deveria ser capaz de gerar essas três

freqüências, de acordo com o processador utilizado. Um método para fazer isso seria utilizar 3 osciladores independentes, um para 66, outro para 100 e outro para 133 MHz. Um método ainda melhor, e que é realmente aplicado na prática, é utilizar um gerador de clock programável.

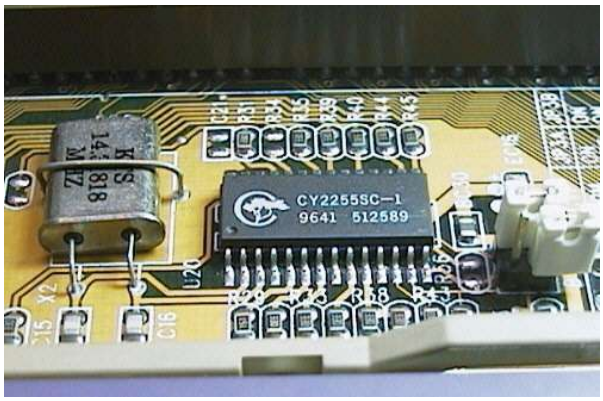


FIGURA 4.26
Gerador de clock e cristal de referência.

A figura 26 mostra um chip gerador de clock. Esses chips sempre trabalham ligados a um cristal que é usado como referência para as freqüências que serão geradas. Eles geram clocks independentes para o processador e para os diversos barramentos usados na placa de CPU.

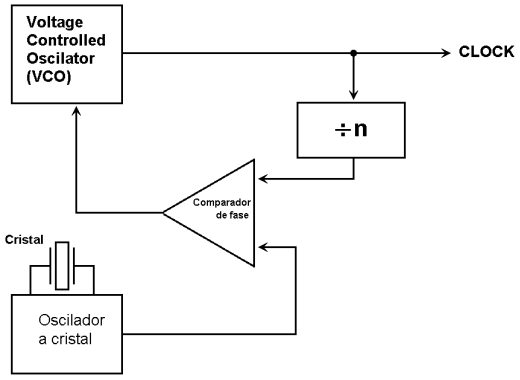


FIGURA 4.27
Gerador de clock programável.

A figura 27 mostra o funcionamento de um gerador de clock. Ele usa um oscilador a cristal como referência. Normalmente o cristal é de 14,31818 MHz, por razões históricas. Este era o cristal usado na geração da freqüência de 4,77 MHz do IBM PC original, e dos 3,58 MHz usados na geração de vídeo composto NTSC, pelas antigas placas de vídeo que eram ligadas a TVs. O clock desejado é gerado por um VCO (oscilador controlado por

voltagem). O clock gerado é dividido digitalmente por um número programável n, e o resultado é comparado com a freqüência gerada pelo oscilador a cristal. O resultado da comparação é usado para controlar o VCO. Quando a freqüência do cristal e a freqüência gerada, dividida por n, estão em fase, o VCO ficará estável e manterá a freqüência fixa. Portanto o circuito ficará estável quando:

$F/n = f_{cristal}$

Ou seja:

$F = n \times f_{cristal}$

Programando o valor de n podemos fazer com que o circuito gere qualquer freqüência, múltipla de 14,31818 MHz. Por exemplo, com n=7 temos

F=100,22726 MHz

O circuito mostrado tem um pequeno inconveniente. Como o valor de n só pode ser um número inteiro, a freqüência gerada será sempre múltipla de 14,31818 MHz. Não seria possível desta forma gerar freqüências como 66 MHz, por exemplo (na verdade os "66 MHz" são 66,666 MHz). Isso pode ser resolvido facilmente, adicionando um divisor na saída do oscilador a cristal. O valor enviado para o comparador de fase não seria 14,31818 MHz, e sim um valor bem menor. Por exemplo, se usamos na saída do oscilador a cristal um divisor 256, teremos:

$F/n = f_{cristal}/256$

$F = n \times f_{cristal}/256$

Se fizermos n=1788 teremos F=100,0035 MHz, valor bem mais próximo dos 100 MHz ideais. Se fizermos n=1192 teremos F=66,669 MHz, valor bem próximo dos 66,666 MHz ideais.

Em um chip gerador de clock existem vários circuitos como o da figura 27, sendo um para cada freqüência gerada. Apenas o oscilador a cristal é comum a todos esses circuitos.

Como funciona o Reset

Todas as placas de CPU possuem um circuito de RESET. Este circuito tem como finalidade enviar um sinal RESET para o processador em duas situações:

1) Quando o usuário pressiona o botão RESET do gabinete
2) No instante em que o computador é ligado

É necessário gerar um RESET automático quando o computador é ligado (Power on Reset) porque neste instante os bits armazenados no interior do processador e dos demais circuitos têm valores aleatórios. O Reset faz com que todos esses bits sejam preenchidos com valores conhecidos, assim o processador não fica "perdido".

Quando o computador está em uso normal, o capacitor C1 estará carregado com uma tensão igual a Vcc. Seu carregamento foi feito pela corrente que passa pelo resistor R1. O ponto X estará representando um bit 1, e este mesmo bit 1 será enviado ao ponto de saída do circuito. O componente em forma de triângulo é um buffer. Trata-se de um operador lógico que gera na saída um bit igual ao da entrada. Portanto em uso normal o sinal RESET estará com o valor 1. Normalmente o comando de RESET ocorre quando o processador recebe na sua entrada RESET, um bit 0.

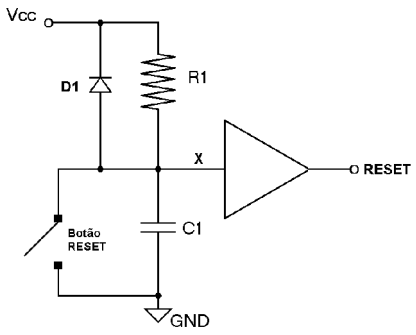


FIGURA 4.28
Circuito de RESET.

Digamos que o usuário pressiona o botão RESET do gabinete. Isto fará com que o capacitor C1 seja imediatamente descarregado, e a tensão no ponto X será zero volts, o que corresponde a um bit 0. Este bit 0 será transmitido pela saída do circuito, resetando o processador e os demais componentes do computador. Quando o usuário solta o botão Reset, o capacitor C1 será carregado através do resistor R1. O tempo de carregamento do capacitor depende dos valores de R1 e C1. Quanto maiores forem seus valores, maior será o tempo de carga. Durante o carregamento do capacitor, sua tensão atinge um valor que passa a ser considerado como um bit 1, o que irá

colocar a saída RESET também em 1. Este é o fim do período de Reset, que dura cerca de 1 segundo, mas pode variar um pouco de uma placa para outra.

Quando o computador é desligado, a tensão Vcc passa a assumir um valor de 0 volts. Isto fará com que o capacitor C1 seja rapidamente descarregado através do diodo D1. Este capacitor passará a ter uma voltagem de 0 volts.

Digamos que agora o computador é ligado. Neste exato instante o capacitor está descarregado, o que representa um bit 0. O sinal RESET na saída do circuito será um bit 0, o que vai resetar o processador e demais circuitos do computador. Como o resitor R1 está ligado a Vcc, passará por ele uma corrente que irá aos poucos carregar o capacitor C1, elevando o valor da sua tensão. Quando esta tensão ultrapassa o valor mínimo para um bit 1, o sinal RESET passará a fornecer também um bit 1, e estará terminado o pulso de RESET. A duração deste pulso depende dos valores de R1 e C1.

Você já deve ter visto computadores que não resetam corretamente quando são ligados, obrigando o usuário a pressionar o botão Reset logo assim que o PC é ligado. O motivo da falha é que o pulso de Reset não tem duração suficiente para resetar o processador e os demais circuitos do computador. Uma solução para este problema é descobrir na placa de CPU onde estão localizados os componentes R1 e C1, e trocar um deles por outro de valor maior. Por exemplo, se usamos ao invés de R1 um resistor duas vezes maior, o pulso de Reset terá uma duração também duas vezes maior, aproximadamente, o que pode resolver o problema. Para encontrar os componentes R1 e C1 é preciso seguir o circuito a partir do conector de Reset da placa de CPU, com o auxílio de um multímetro.

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