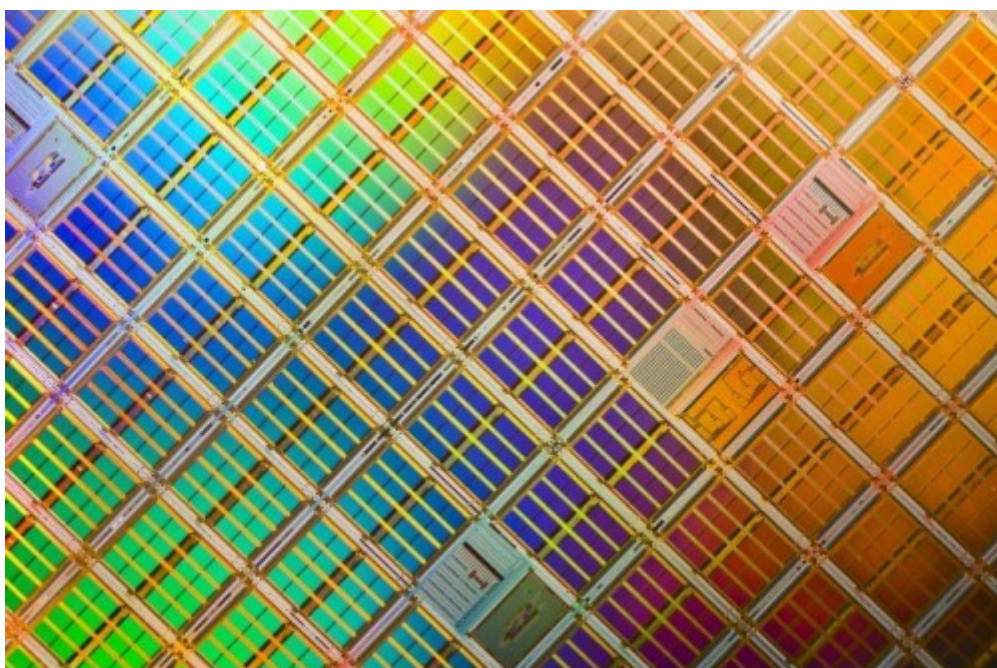


PRIMEIRA PARTE

Introdução

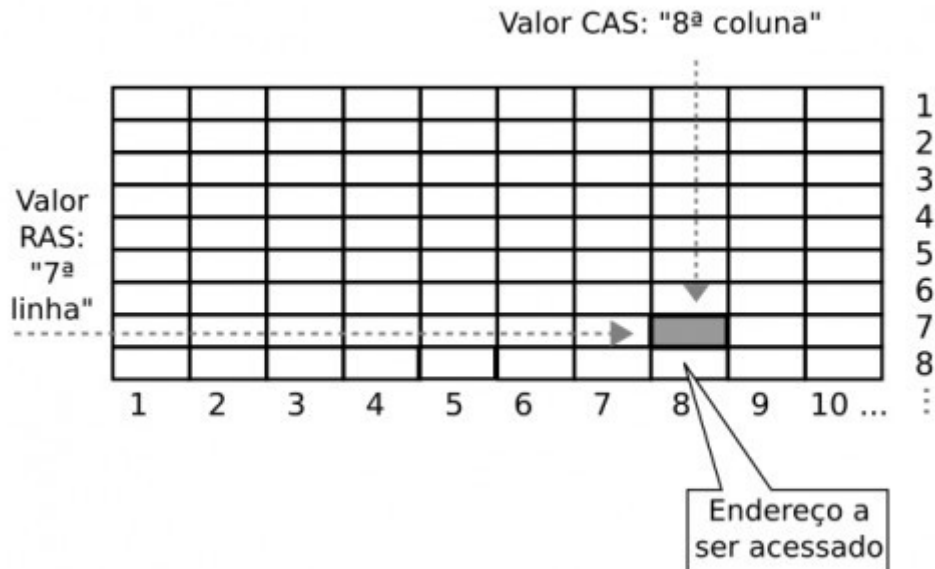
Embora seja brutalmente mais rápida que o HD e outros periféricos, a memória RAM continua sendo muito mais lenta que o processador. O uso de caches diminui a perda de desempenho, reduzindo o número de acessos à memória; mas, quando o processador não encontra a informação que procura nos caches, precisa recorrer a um doloroso acesso à memória principal, que em um processador atual pode resultar em uma espera de mais de 150 ciclos. Para reduzir a diferença (ou pelo menos tentar impedir que ela aumente ainda mais), os fabricantes de memória passaram a desenvolver um conjunto de novas tecnologias, a fim de otimizar o acesso aos dados, dando origem aos módulos de memória DDR2 e DDR3 utilizados atualmente.



Começando do básico, um chip de memória é um exército de clones, formado por um brutal número de células idênticas, organizadas na forma de linhas e colunas, de forma similar a uma planilha eletrônica. O chip de memória em si serve apenas para armazenar dados, não realiza nenhum tipo de processamento. Por isso, é utilizado um componente adicional, o controlador de memória, que pode ser incluído tanto no chipset da placa-mãe quanto dentro do próprio processador, como no caso dos processadores AMD a partir do Athlon 64 e dos processadores Intel a partir do Core i7. Para acessar um determinado endereço de memória, o controlador primeiro gera o valor RAS (Row Address Strobe), ou o número da linha da qual o endereço faz parte, gerando em seguida o valor CAS (Column Address Strobe), que corresponde à coluna. Quando o RAS é

MEMÓRIAS

enviado, toda a linha é ativada simultaneamente; depois de um pequeno tempo de espera, o CAS é enviado, fechando o circuito e fazendo com que os dados do endereço selecionado sejam lidos ou gravados:



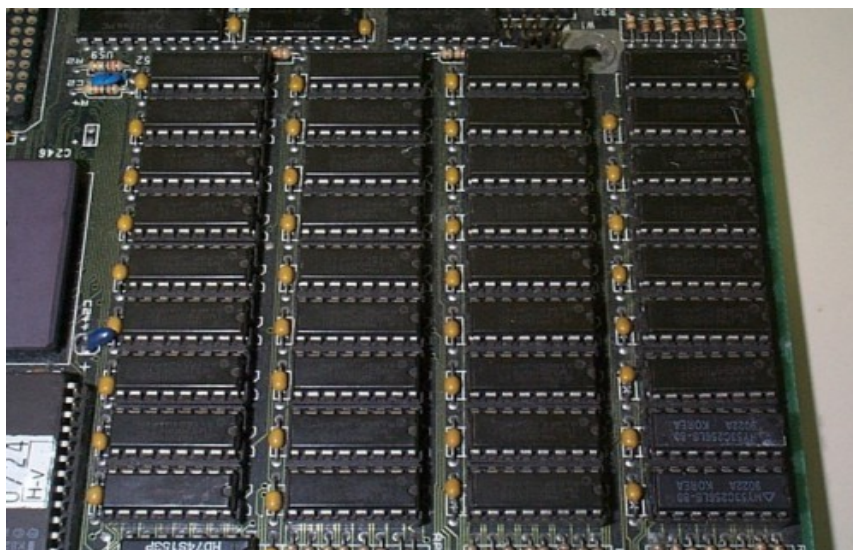
Não existe um caminho de volta, ligando cada endereço de volta ao controlador de memória. Em vez disso, é usado um barramento comum, compartilhado por todos os endereços do módulo. O controlador de memória sabe que os dados que está recebendo são os armazenados no endereço X, pois ele se "lembra" que acabou de acessá-lo.

A Evolução dos Módulos de Memória

Nos micros XT, 286 e nos primeiros 386, ainda não eram utilizados módulos de memória. Em vez disso, os chips de memória eram instalados diretamente na placa-mãe, encaixados individualmente em colunas de soquetes (ou soldados), onde cada coluna formava um banco de memória.

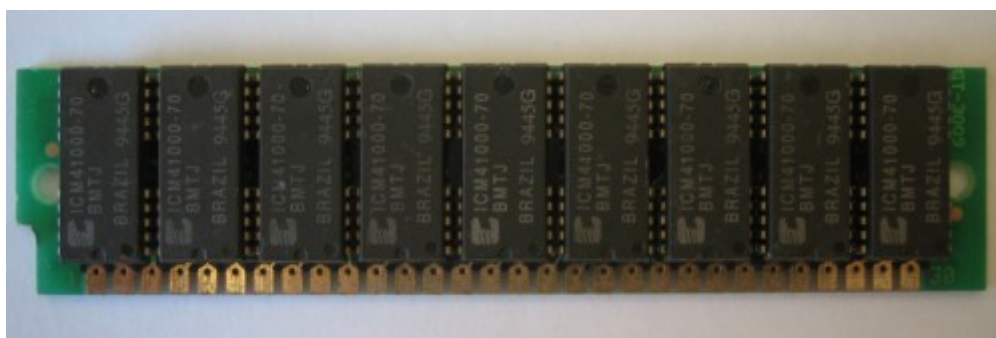
MEMÓRIAS

Esse era um sistema antiquado e que trazia várias desvantagens, por dificultar upgrades de memória ou a substituição de módulos com defeito. Imagine você fazendo um upgrade de memória em uma placa como esta:



Não é só você que não achou muito atraente a ideia de ficar catando chips de memória um a um. Foi questão de tempo até que alguém aparecesse com uma alternativa mais prática, capaz de tornar a instalação fácil até mesmo para usuários inexperientes.

Os módulos de memória são pequenas placas de circuito onde os chips DIP são soldados, facilitando o manuseio e a instalação. Os primeiros módulos de memória criados são chamados de módulos SIMM, sigla que significa "Single In Line Memory Module", justamente porque existe uma única via de contatos, com 30 vias. Apesar de existirem contatos também na parte de trás do módulo, eles servem apenas como uma extensão dos contatos frontais, de forma a aumentar a área de contato com o soquete. Examinando o módulo, você verá um pequeno orifício em cada contato, que serve justamente para unificar os dois lados.

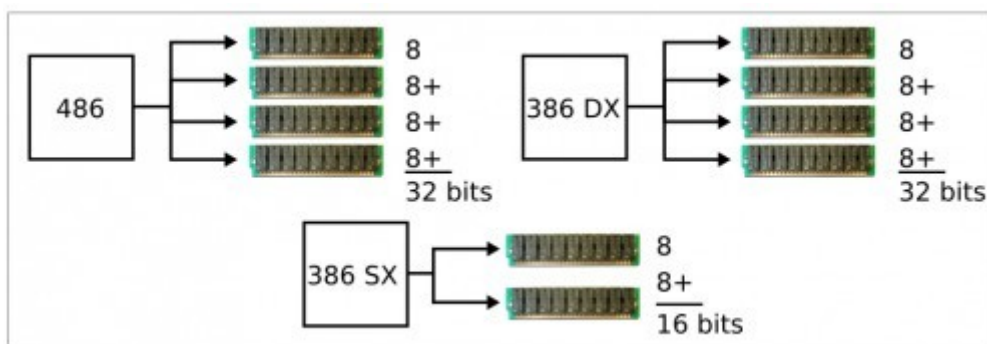


MEMÓRIAS

Os módulos de 30 vias possuíam sempre 8 ou 9 chips de memória. Cada chip fornecia um único bit de dados em cada transferência, de forma que 8 deles formavam um módulo capaz de transferir 8 bits por ciclo. No caso dos módulos com 9 chips, o último era destinado a armazenar os bits de paridade, que melhoravam a confiabilidade, permitindo identificar erros. Hoje em dia os módulos de memória são mais confiáveis, de forma que a paridade não é mais usada. No lugar dela, temos o ECC, um sistema mais avançado, usado em módulos de memória destinados a servidores. Os módulos de 30 vias foram utilizados em micros 386 e 486 e foram fabricados em várias capacidades. Os mais comuns foram os módulos de 1 MB, mas era possível encontrar também módulos de 512 KB, 2 MB e 4 MB. Existiram também módulos de 8 e 16 MB, mas eles eram muito raros devido ao custo.

Os processadores 386 e 486 utilizavam um barramento de 32 bits para o acesso à memória, o que tornava necessário combinar 4 módulos de 30 vias para formar um banco de memória. Os 4 módulos eram então acessados pelo processador como se fossem um só. Era preciso usar os módulos em quartetos: 4 módulos ou 8 módulos, mas nunca um número quebrado.

A exceção ficava por conta dos micros equipados com processadores 386SX, onde são necessários apenas 2 módulos, já que o 386SX acessa a memória usando palavras de 16 bits:



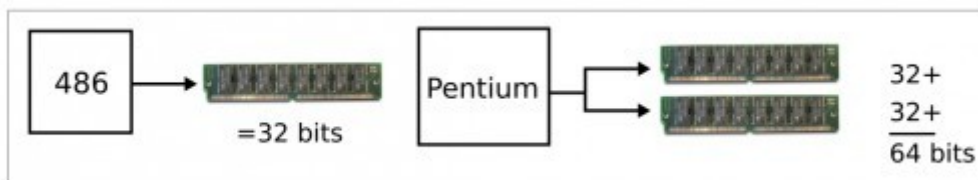
Apesar de serem muito mais práticos do que manipular diretamente os chips DIP, os módulos SIMM de 30 vias ainda eram bastante inconvenientes, já que era preciso usar 4 módulos idênticos para formar cada banco de memória. Eles foram desenvolvidos pensando mais na questão da simplicidade e economia de custos do que na praticidade.

Para solucionar o problema, os fabricantes criaram um novo tipo de módulo de memória SIMM de 32 bits, que possui 72 vias. Os módulos de 72 vias substituíram rapidamente os antigos nas placas para 486 e se tornaram o padrão nos micros Pentium, sendo em seguida substituídos pelos módulos de 168 vias.

MEMÓRIAS



Em vez de quatro módulos, é preciso apenas um módulo SIMM de 72 vias para formar cada banco de memória nos micros 486. Como o Pentium acessa a memória usando palavras de 64 bits, são necessários 2 módulos em cada banco. É por isso que nos micros Pentium 1 precisamos sempre usar os módulos de memória em pares:



O acesso de 64 bits à memória foi introduzido para permitir que o processador conseguisse acessar grandes quantidades de dados mais rapidamente. O processador é tão mais rápido que a memória RAM, que depois de esperar vários ciclos para poder acessá-la, o melhor a fazer é pegar a maior quantidade de dados possível e guardar tudo no cache. Naturalmente os dados serão processados em blocos de 32 bits, mas a poupança ajuda bastante.

Dentro de um banco, todos os módulos são acessados ao mesmo tempo, como se fossem um só, por isso era sempre recomendável usar dois módulos iguais. Ao usar quatro módulos, o importante era que cada par fosse composto por dois módulos iguais. Não existia problema em usar dois pares de módulos diferentes, como ao usar dois de 16 MB e mais dois de 8 MB para totalizar 48 MB, por exemplo.

Uma curiosidade é que algumas placas-mãe para Pentium podem trabalhar com apenas um módulo de 72 vias. Nesse caso, a placa engana o processador, fazendo dois acessos de 32 bits consecutivos, entregando os dados de uma só vez para o processador. Apesar de funcionar, esse esquema reduz bastante a velocidade do micro, pois a taxa de transferência ao ler dados a partir da memória é efetivamente reduzida à metade.

Finalmente, temos os módulos **DIMM**, usados atualmente. Ao contrário dos módulos SIMM de 30 e 72 vias, os módulos DIMM possuem contatos em ambos os lados do módulo, o que justifica seu nome, "Double In Line Memory Module" ou "módulo de memória com dupla linha de contato". Todos os módulos DIMM são módulos de 64 bits, o que eliminou a necessidade de usar 2 ou 4

MEMÓRIAS

módulos para formar um banco de memória. Muitas placas-mãe oferecem a opção de usar dois módulos (acessados simultaneamente) para melhorar a velocidade de acesso.

Esse recurso é chamado de **dual-channel** e melhora consideravelmente o desempenho, sobretudo nas placas-mãe com vídeo onboard, onde a placa de vídeo disputa o acesso à memória RAM com o processador principal. De qualquer forma, mesmo nas placas dual-channel, usar os módulos em pares é opcional; você pode perfeitamente usar um único módulo, mas neste caso o suporte à dual-channel fica desativado.

Existem três formatos de memória DIMM. Os mais antigos são os módulos de memória **SDR**, de 168 vias, que substituíram os antigos módulos de memória EDO, mas logo deram lugar às tecnologias mais recentes. Em seguida, temos os módulos de memória **DDR**, que possuem 184 contatos; os módulos **DDR2**, que possuem 240; e, os módulos **DDR3**, que também possuem 240 contatos, mas utilizam tensões e sinalizações diferentes.

Apesar do maior número de contatos, os módulos DDR, DDR2 e DDR3 são exatamente do mesmo tamanho que os módulos SDR de 168 vias, por isso foram introduzidas mudanças na posição dos chanfros de encaixe, de forma que você não consiga encaixar os módulos em placas incompatíveis.

Os módulos SDR possuem dois chanfros, enquanto os DDR possuem apenas um chanfro, que ainda por cima é colocado em uma posição diferente:



Os módulos DDR2 também utilizam um único chanfro, mas ele está posicionado mais próximo do canto do módulo em relação ao usado nos módulos DDR, de forma que é novamente impossível encaixar um módulo DDR2 numa placa antiga:



Isso é necessário, pois além das mudanças na forma de acesso, os módulos DDR2 utilizam tensão de 1.8V, enquanto os módulos DDR usam 2.5V. Se fosse possível instalar um módulo DDR2 em uma placa antiga, a maior tensão queimaria o módulo rapidamente.

Outra característica que torna os módulos DDR2 diferentes é a presença de um terminador resistivo dentro de cada chip de memória. O terminador é necessário para "fechar o circuito", evitando que os sinais elétricos retornem na forma de interferência ao chegarem ao final do barramento. Nos módulos DDR os terminadores são instalados na placa-mãe, o que torna a terminação menos eficiente. Como os módulos DDR2 operam a frequências muito mais altas, a presença do terminador dentro dos próprios chips se tornou uma necessidade, já que torna o sinal mais estável e livre de ruídos.

Em seguida temos os módulos DDR3. Como comentei, eles utilizam os mesmos 240 contatos dos módulos DDR2 e mantém o mesmo formato. A única diferença visível (fora etiquetas e códigos de identificação) é a mudança na posição do chanfro, que passou a ser posicionado mais próximo do canto do módulo. O chanfro serve justamente para impedir que os módulos sejam encaixados em placas incompatíveis:



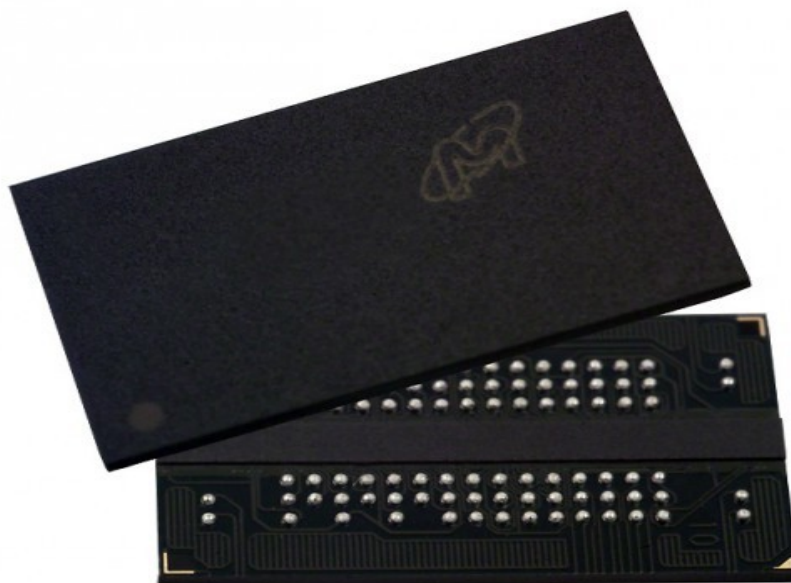
Como os módulos DDR2 e DDR3 trabalham a frequências mais altas, o uso de dissipadores se tornou mais comum. Eles não são realmente necessários, mas a melhor dissipação do calor permite que o módulo trabalhe a frequências mais altas, por isso eles se tornaram norma nos módulos de alto desempenho e, principalmente, nos módulos "premium", destinados a overclock.

MEMÓRIAS

Alguns fabricantes chegam a utilizar heat-pipes ou a oferecer coolers ativos, que podem ser instalados sobre os módulos, mas nesse caso o efeito é mais estético do que prático.



Outra diferença é que os chips DDR2 e DDR3 utilizam o encapsulamento BGA (Ball Grid Array), no lugar do encapsulamento TSOP (Thin Small-Outline Package), usado nos chips SDR e DDR. A grande diferença é que no BGA os pontos de solda são posicionados diretamente na parte inferior dos chips, em vez de serem usadas as "perninhas" laterais. Isso reduz a distância que o sinal elétrico precisa percorrer, além de reduzir o nível de interferências, permitindo que os módulos sejam capazes de operar a frequências mais altas. Esta imagem ilustrativa da Micron mostra bem como os chips se parecem, com a face visível e os pontos de solda:



Concluindo, existem também os módulos SODIMM (Small Outline DIMM), destinados a notebooks. Eles são basicamente versões miniaturizadas dos módulos destinados a desktops, que

MEMÓRIAS

utilizam os mesmos tipos de chips de memória.

Os módulos SODIMM SDR possuem 144 pinos, os módulos SODIMM DDR e DDR2 possuem 200 pinos e os módulos SODIMM DDR3 possuem 204 pinos. Nos módulos SDR o chanfro fica próximo ao centro do módulo, enquanto nos DDR e DDR2 ele fica à esquerda.

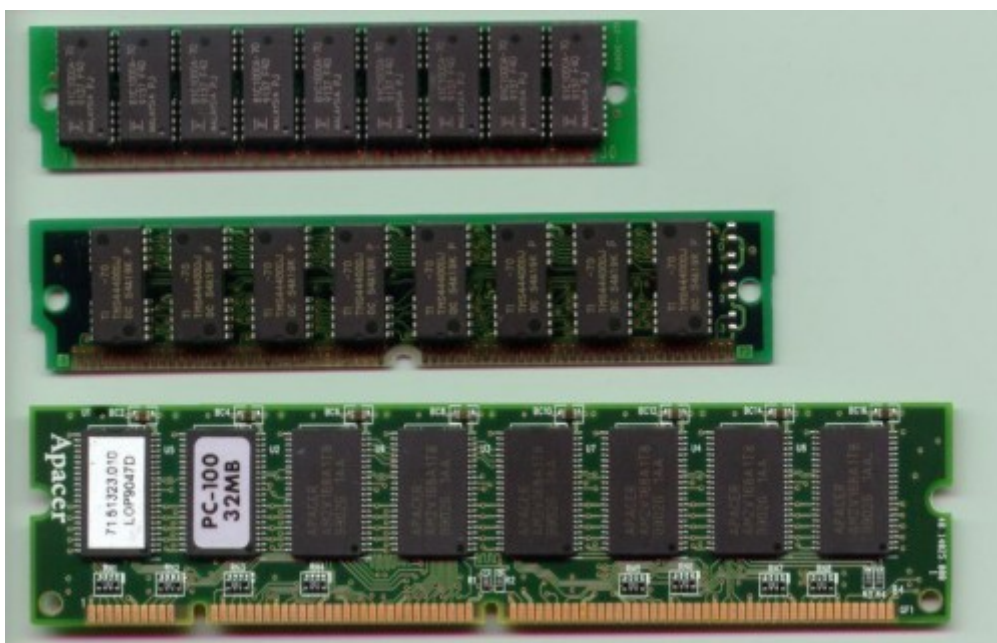
Assim como nos módulos para desktops, existe uma pequena diferença no posicionamento do chanfro entre os módulos DDR e DDR2, que impede o encaixe incorreto, já que ambos são incompatíveis:



A Questão do Acesso de Dados

Antigamente (na época dos módulos SIMM de 30 vias usados nos micros 386 e 486), cada chip de memória oferecia a leitura de um bit de cada vez. Os módulos de 30 vias eram compostos por 8 chips de memória (com exceção dos módulos com paridade, que usavam 9 chips), o que resultava na leitura de 8 bits por ciclo. Apesar disso, o processador lia 32 bits de dados a cada ciclo, de forma que era necessário usar os módulos em quartetos.

Do ponto de vista do processador, não existia divisão, os chips eram acessados como se fossem um só. O processador não via 32 endereços separados, em 32 chips diferentes, mas sim um único endereço, contendo 32 bits.



Nos módulos DIMM atuais são geralmente usados 8 chips de 8 bits cada um, formando os 64 bits fornecidos ao processador. Existem ainda módulos com 16 chips de 4 bits cada, ou ainda, módulos com 4 chips de 16 bits (comuns em notebooks). Do ponto de vista do processador, não faz diferença, desde que somados, os chips totalizem 64 bits.

Imagine que o controlador de memória envia sequências com 4, 8 ou 16 pares de endereços RAS e CAS e recebe de volta o mesmo número de leituras de 64 bits. Mesmo em casos em que o processador precisa de apenas alguns poucos bytes, contendo uma instrução ou bloco de dados, ele precisa ler todo o bloco de 64 bits adjacente, mesmo que seja para descartar os demais.

No caso dos chipsets e processadores com controladores de memória dual-channel, continuamos tendo acessos de 64 bits, a única diferença é que agora o controlador de memória é capaz de acessar dois endereços diferentes (cada um em um módulo de memória) a cada ciclo de

MEMÓRIAS

clock, ao invés de apenas um. Isso permite transferir o dobro de dados por ciclo, fazendo com que o processador precise esperar menos tempo ao transferir grandes quantidades de dados.

Na verdade, nos PCs contemporâneos, praticamente qualquer dispositivo pode acessar a memória diretamente através do barramento PCI Express, PCI (ou AGP no caso de micros mais antigos) e até mesmo a partir das portas SATA, IDE e USB. Naturalmente, todos os acessos são coordenados pelo processador, mas como a memória é uma só, temos situações onde o processador precisa esperar para acessar a memória, porque ela está sendo acessada por outro dispositivo.

Existem várias formas de melhorar o desempenho da memória RAM. A primeira é aumentar o número de bits lidos por ciclo, tornando o barramento mais largo, como o aumento de 32 para 64 bits introduzida pelo Pentium 1, que continua até os dias de hoje. O problema em usar um barramento mais largo é que o maior número de trilhas necessárias, tanto na placa-mãe quanto nos próprios módulos de memória, aumentam a complexidade e o custo de produção.

A segunda é acessar dois ou mais módulos de memória simultaneamente, como nas placas e processadores com controladores de memória dual-channel ou triple-channel. O problema é que nesse caso precisamos de dois módulos, além de circuitos e trilhas adicionais na placa-mãe e pinos adicionais no soquete do processador.

A terceira é criar módulos de memória mais rápidos, como no caso das memórias DDR2 e DDR3. Essa questão da velocidade pode ser dividida em dois quesitos complementares: o número de ciclos por segundo e a latência, que é o tempo que a primeira operação numa série de operações de leitura ou escrita demora para ser concluída. O tempo de latência poderia ser comparado ao tempo de acesso de um HD, enquanto o número de ciclos poderia ser comparado ao clock do processador.

É aqui que entram as diferentes tecnologias de memórias que foram introduzidas ao longo das últimas décadas, começando pelas memórias regulares, usadas nos XTs e 286, que evoluíram para as memórias FPM, usadas em PCs 386 e 486, em seguida para as memórias EDO, usadas nos últimos micros 486s e nos Pentium. Estas três primeiras tecnologias foram então substituídas pelas memórias SDR-SDRAM, seguidas pelas memórias DDR e pelas DDR2 e DDR3 usadas atualmente. Tudo começou com as memórias regulares, que são o tipo mais primitivo de memória RAM. Nelas, o acesso é feito da forma tradicional, enviando o endereço RAS, depois o CAS e aguardando a leitura dos dados para cada ciclo de leitura.

Isso funcionava bem nos micros XT e 286, onde o clock do processador era muito baixo e a memória RAM podia funcionar de forma sincronizada com ele. Em um 286 de 8 MHz, eram usados chips com tempo de acesso de 125 ns (nanossegundos) e em um de 12 MHz eram usados chips de 83 ns.

MEMÓRIAS

O problema era que a partir daí as memórias da época atingiram seu limite e passou a ser necessário fazer com que a memória trabalhasse de forma assíncrona, onde o processador trabalha a uma frequência mais alta que a memória RAM.

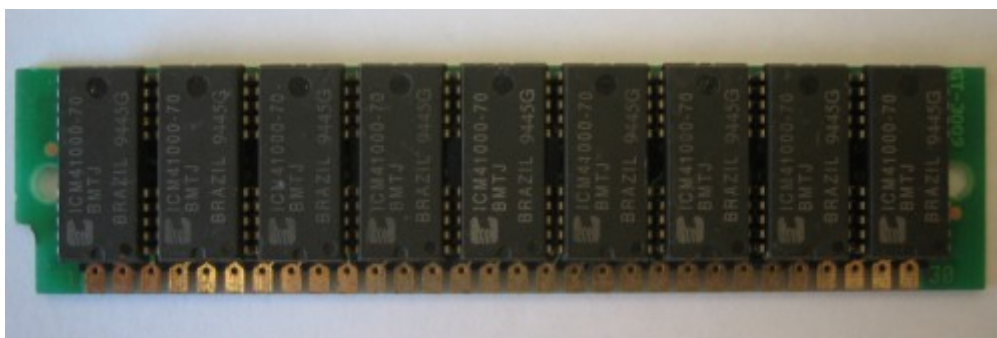
A partir do 386, a diferença passou a ser muito grande, o que levou à introdução da memória cache e ao início da corrida em busca de módulos de memória mais rápidos.

Memórias FPM

A primeira melhora significativa na arquitetura das memórias veio com o FPM (Fast-Page Mode, ou "modo de paginação rápida"). A ideia é que, ao ler um bloco de instruções ou arquivo gravado na memória, os dados estão quase sempre gravados sequencialmente. Não é preciso então enviar o endereço RAS e CAS para cada bit a ser lido, mas simplesmente enviar o endereço RAS (linha) uma vez e em seguida enviar uma sequência de até 4 endereços CAS (coluna), realizando uma série rápida de 4 leituras.

O primeiro ciclo de leitura continua tomando o mesmo tempo, mas as 3 leituras seguintes passam a ser bem mais rápidas. Graças a essa pequena otimização, as memórias FPM conseguem ser até 30% mais rápidas que as memórias regulares, sem grandes alterações nos chips de memória ou na técnica de fabricação. O burst (sequência de acessos rápidos) de 4 leituras pode ser prolongado para 8, ou até mesmo 16 leituras consecutivas, desde que sejam lidos dados gravados em endereços adjacentes, da mesma linha.

As memórias FPM foram utilizadas em micros 386, 486 e nos primeiros micros Pentium, na forma de módulos SIMM de 30 ou 72 vias, com tempos de acesso de 80, 70 ou 60 ns, sendo as de 70 ns as mais comuns.



Instaladas em uma placa-mãe soquete 7, que trabalhe com bus de 66 MHz, os intervalos de espera de memórias FPM podem ser de até 6-3-3-3, o que significa que o processador terá de esperar cinco ciclos da placa-mãe para a memória efetuar a primeira leitura de dados e somente mais 3 ciclos para cada leitura subsequente. Os tempos de espera das memórias podiam ser configurados no setup, através da opção "Memory Timing" ou similar, onde ficavam disponíveis

opções como "slow", "normal" e "fast", que substituem os valores numéricos. No caso das placas para 486, que operavam a clocks mais baixos (30, 33 ou 40 MHz), os tempos de espera podiam ser configurados com valores mais baixos, como 4-3-3-3 ou 3-2-2-2, já que, com menos ciclos por segundo, é natural que os tempos de resposta dos módulos correspondam a um número menor de ciclos da placa-mãe.

Memórias EDO

As memórias EDO (Extended Data Output) foram introduzidas a partir de 1994 e trouxeram mais uma melhoria significativa no modo de acesso a dados. Nas memórias FPM, uma leitura não pode ser iniciada antes que a anterior termine, mesmo dentro do burst de 4 leituras dentro da mesma linha. O controlador precisa esperar que os dados referentes à leitura anterior cheguem, antes de poder ativar o endereço CAS seguinte.

Nas memórias EDO, o controlador faz a leitura enviando o endereço RAS (como de costume) e em seguida enviando os 4 endereços CAS em uma frequência predefinida, sem precisar esperar que o acesso anterior termine. Os sinais chegam às células de memória na sequência em que foram enviados e, depois de um pequeno espaço de tempo, o controlador recebe de volta as 4 leituras.

O resultado acaba sendo exatamente o mesmo, mas passa a ser feito de forma mais rápida. Usadas em uma placa soquete 7, operando a 66 MHz, as memórias EDO são capazes de trabalhar com tempos de acesso de apenas 6-2-2-2, ou mesmo 5-2-2-2 (nos módulos de 60 ns). Nos bursts de 8 ou mais leituras, o ganho acaba sendo ainda maior, com o módulo FPM realizando a leitura dos 8 endereços em 27 ciclos (6-3-3-3-3-3-3-3) e o EDO em 20 (6-2-2-2-2-2-2-2). Veja que o ganho é maior em leituras de vários endereços consecutivos, por isso alguns aplicativos se beneficiavam mais do que outros.

Os módulos de memória EDO foram produzidos em versões com tempos de acesso de 70, 60 e 50 ns, com predominância dos módulos de 60 ns. Eles foram usados predominantemente na forma de módulos de 72 vias, usados nos micros 486 e Pentium fabricados a partir de 1995.



Existiram ainda alguns módulos DIMM de 168 vias com memória EDO. Eles foram bastante

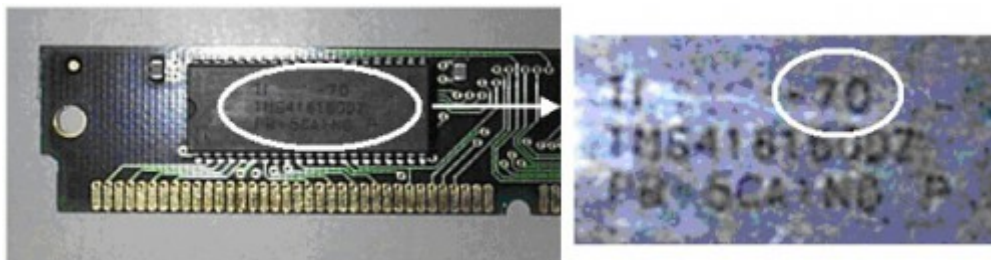
MEMÓRIAS

raros, pois foram logo substituídos pelos módulos de memória SDRAM.

As melhorias na arquitetura das memórias EDO tornaram-nas incompatíveis com placas-mãe equipadas com chipsets antigos. Basicamente, apenas as placas para processadores Pentium e algumas placas-mãe para 486 com slots PCI (as mais recentes) aceitam trabalhar com memórias EDO. Existem também placas para 486 "tolerantes" que funcionam com memórias EDO, apesar de não serem capazes de tirar proveito do modo de acesso mais rápido, e finalmente, as placas incompatíveis, que nem chegam a inicializar caso sejam instaladas memórias EDO.

Todos os módulos de 30 vias são de memórias FPM, enquanto (com exceção de alguns módulos antigos) todos os de 168 vias são de memórias SDRAM. A confusão existe apenas nos módulos de 72 vias, que podem ser tanto de memórias EDO quanto de memórias FPM. Para saber quem é quem, basta verificar o tempo de acesso. Todo módulo de memória traz seus dados estampados nos chips, na forma de alguns códigos; o tempo de acesso é indicado no final da primeira linha. Se ela terminar com -7, -70, ou apenas 7, ou 70, o módulo possui tempo de acesso de 70 ns. Se por outro lado a primeira linha terminar com -6, -60, 6 ou 60 o módulo é de 60 ns.

Como quase todos os módulos de 70 ns são de memórias FPM, e quase todos os módulos de memórias EDO são de 60 ns, você pode usar esse método para determinar com 95% de certeza o tipo de memória usada:



No final da era Pentium, as memórias EDO enfrentaram a concorrência das memórias BEDO (Burst EDO), que utilizavam um sistema de pipeline para permitir acessos mais rápidos que as EDO. Em um Bus de 66 MHz, as memórias BEDO eram capazes de funcionar com temporização de 5-1-1-1, quase 30% mais rápido que as memórias EDO convencionais.

No papel as memórias BEDO eram interessantes, mas elas nunca foram usadas em grande escala. A tecnologia era propriedade da Micron, que ansiava por cobrar royalties dos demais fabricantes, caso ela fosse adotada em grande escala. Os fabricantes de memória trabalham com margens de lucro incrivelmente apertadas, de forma que a palavra "royalties" gera calafrios. Ao invés de caírem na armadilha da Micron, eles se apressaram em adotar as memórias SDRAM, que além de serem um padrão aberto, eram tecnicamente superiores.

Memórias SDR-SDRAM

Tanto as memórias FPM quanto as memórias EDO são assíncronas, o que significa que elas trabalham em seu próprio ritmo, independentemente dos ciclos da placa-mãe. Isso explica porque memórias FPM que foram projetadas para funcionar em placas para processadores 386 ou 486 funcionam sem problemas em placas soquete 7, que trabalham a 66 MHz. Na verdade, a memória continua trabalhando na mesma velocidade, o que muda são os tempos de espera que passam a ser mais altos. Assim, em vez de responder a cada 2 ciclos da placa-mãe, elas podem passar a responder a cada 3 ou 4 ciclos, por exemplo.

As memórias SDRAM (Synchronous Dynamic RAM) por sua vez, são capazes de trabalhar sincronizadas com os ciclos da placa-mãe, sem tempos de espera. Isso significa que a temporização das memórias SDRAM é sempre de uma leitura por ciclo. Independentemente da frequência de barramento utilizada, os tempos de acesso serão sempre de 6-1-1-1, ou mesmo 5-1-1-1.

Veja que o primeiro acesso continua tomando vários ciclos, pois nele é necessário realizar o acesso padrão, ativando a linha (RAS) e depois a coluna (CAS). Apenas a partir do segundo acesso é que as otimizações entram em ação e a memória consegue realizar uma leitura por ciclo, até o final da leitura.

O burst de leitura pode ser de 2, 4 ou 8 endereços e existe também o modo "full page" (uma novidade introduzida nos módulos SDRAM), onde o controlador pode especificar um número qualquer de endereços a serem lidos sequencialmente, até um máximo de 512. Ou seja, em situações ideais, pode ser possível realizar a leitura de 256 setores em 260 ciclos! Só para efeito de comparação, se fossem usadas memórias regulares, com tempos de acesso similares, a mesma tarefa tomaria pelo menos 1280 ciclos.

Outra característica que ajuda as memórias SDRAM a serem mais rápidas que as EDO e FPM é a divisão dos módulos de memória em vários bancos. Um módulo DIMM pode ser formado por 2, 4, ou mesmo 8 bancos de memória, cada um englobando parte dos endereços disponíveis. Apenas um dos bancos pode ser acessado de cada vez, mas o controlador de memória pode aproveitar o tempo de ociosidade para fazer algumas operações nos demais, como executar os ciclos de refresh e também a pré-carga dos bancos que serão acessados em seguida. Nos módulos EDO e FPM, todas essas operações precisam ser feitas entre os ciclos de leitura, o que toma tempo e reduz a frequência das operações de leitura.

A partir das memórias SDRAM, tornou-se desnecessário falar em tempos de acesso, já que a memória trabalha de forma sincronizada em relação aos ciclos da placa-mãe. As memórias passaram então a ser rotuladas de acordo com a frequência em que são capazes de operar. No caso

MEMÓRIAS

das memórias SDR-SDRAM temos as memórias PC-66, PC-100 e PC-133, no caso das DDR-SDRAM temos as DDR-200, DDR-266, DDR-333, DDR-400 (e assim por diante), enquanto nas DDR2 temos as DDR2-533, DDR2-666, DDR2-800, DDR2-1066, que foram sucedidas pelos padrões de memórias DDR3.

Um módulo de memória PC-133 deve ser capaz de operar a 133 MHz, fornecendo 133 milhões de leituras (teóricas) por segundo. Entretanto, essa velocidade é atingida apenas quando o módulo realiza um burst de várias leituras. O primeiro acesso continua levando 5, 6 ou mesmo 7 ciclos da placa-mãe, como nas memórias antigas.

Ou seja, o fato de ser um módulo PC-100 não indica que ele possua um tempo de acesso de 10 ns ou menos (nem mesmo os módulos DDR3 atuais atingem essa marca). Pelo contrário, a maioria dos módulos PC-100 trabalhavam com tempos de acesso de 40 ns. Entretanto, graças a todas as otimizações que vimos, as leituras podiam ser paralelizadas, de forma que, no final, o módulo suporta bursts de leitura onde, depois de um lento ciclo inicial, ele consegue realmente entregar 64 bits de dados a cada 10 ns.

Independentemente da frequência de operação, temos também os módulos CL2 e CL3, onde o "CL" é abreviação de "CAS latency", ou seja, o tempo de latência relacionado ao envio do valor CAS, durante o primeiro acesso de cada burst.

Em módulos CL2, o envio do valor CAS toma 2 ciclos, enquanto nos CL3 toma 3 ciclos. A eles, somamos um ciclo inicial e mais dois ciclos relacionados ao envio do valor RAS, totalizando 5 (nos módulos CL2) ou 6 (nos CL3) ciclos para o acesso inicial. Entretanto, a diferença prática acaba sendo pequena, pois os acessos seguintes demoram sempre apenas um ciclo.

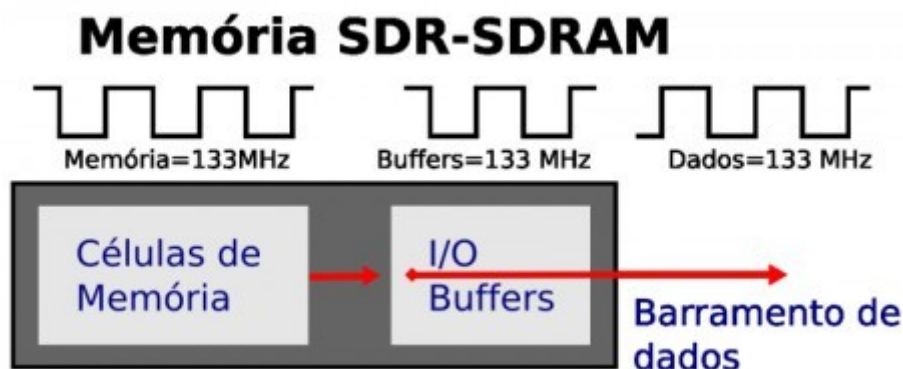
Um módulo CL2 realizaria um burst de 8 leituras em 12 ciclos (5-1-1-1-1-1-1-1), enquanto o CL3 demoraria 13 ciclos (6-1-1-1-1-1-1-1). Ou seja, embora os módulos CL2 tenham sido celebrados e sejam alvo de um grande esforço de marketing por parte dos fabricantes, a diferença de performance era muito pequena para justificar a diferença de preço.

Veja que das memórias regulares, até as SDRAM, foi possível multiplicar a velocidade de transferência sem fazer alterações fundamentais nas células, que continuam seguindo o mesmo projeto básico, com um transistor e um capacitor para cada bit armazenado. Desde a década de 80, as reduções nos tempos de acesso foram apenas incrementais, acompanhando as melhorias nas técnicas de fabricação. O que realmente evoluiu com o passar do tempo foram os circuitos em torno dos módulos, que otimizaram o processo de leitura, extraindo mais e mais performance.

Chegamos então às memórias DDR, DDR2 e DDR3 usadas atualmente, que levam este processo crescente de otimização a um novo nível, permitindo que o módulo de memória realize várias transferências por ciclo, multiplicando a frequência efetiva.

Memórias DDR

Apesar das otimizações, os módulos de memória SDR-SDRAM continuam realizando apenas uma transferência por ciclo, da forma mais simples possível. Depois de decorrido o longo ciclo inicial, as células de memória entregam uma leitura de dados por ciclo, que passa pelos buffers de saída e é despachada através do barramento de dados. Todos os componentes trabalham na mesma frequência:

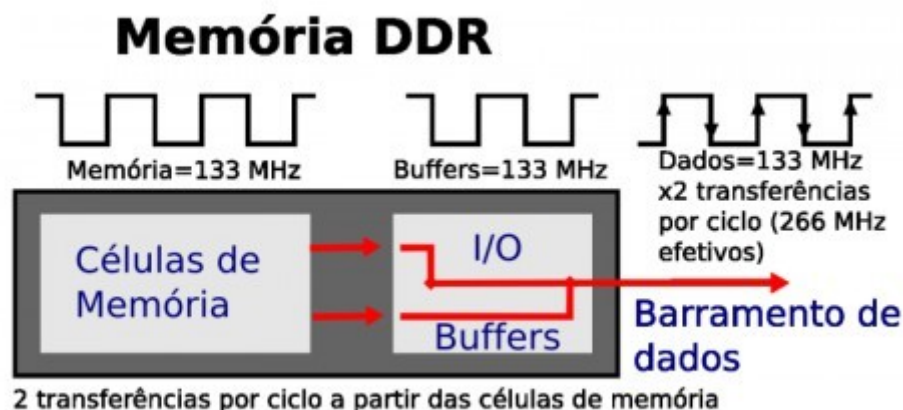


As memórias DDR implementam um novo truque, que as torna capazes de realizarem duas transferências por ciclo e serem quase duas vezes mais rápidas que as memórias SDRAM regulares, mesmo mantendo a mesma frequência de operação e a mesma tecnologia básica. Vem daí o termo "DDR", que significa "Double Data Rate", ou frequência dupla de transferência de dados.

Com o lançamento das memórias DDR, as SDRAM passaram a ser chamadas de "SDR" (Single Data Rate) ou SDR-SDRAM, reforçando a diferença entre as duas tecnologias. Embora tanto as memórias DDR quanto as DDR2 e DDR3 continuem sendo memórias SDRAM, o termo saiu de moda, já que é muito mais fácil dizer "DDR" do que "DDR-SDRAM".

De volta à tecnologia, os chips de memória DDR incluem circuitos adicionais, que permitem gerar comandos de acesso e receber os dados referentes às leituras duas vezes por ciclo de clock, executando uma operação no início do ciclo e outra no final. Como são utilizadas as mesmas trilhas para realizar ambas as transferências, não foi necessário fazer grandes modificações nem nos módulos de memória, nem nas placas-mãe.

Apesar disso, as células de memória propriamente ditas continuam operando na mesma frequência. Em um módulo DDR-266, por exemplo, elas operam a apenas 133 MHz, da mesma forma que num módulo SDR PC-133. O pulo do gato é fazer com que cada um dos dois comandos de leitura (ou gravação) sejam enviados para um endereço diferente, na mesma linha. As duas leituras são enviadas através do barramento de dados na forma de duas transferências separadas, uma realizada no início e a outra no final do ciclo de clock:



O maior problema é que o ciclo inicial continua demorando o mesmo tempo que nas memórias SDRAM, de forma que o ganho aparece apenas em leituras de vários setores consecutivos e a taxa de transferência nunca chega realmente a dobrar, variando bastante de acordo com o tipo de aplicativo usado.

A temporização para um burst de 8 leituras, usando memórias DDR, seria $5 - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2}$ (8.5 ciclos) ao invés de $5 - 1 - 1 - 1 - 1 - 1 - 1 - 1$ (12 ciclos) como num módulo SDR. A diferença é menor em bursts menores, de apenas duas ou quatro leituras.

Apesar disso, as memórias DDR acabaram sendo um excelente negócio, pois tornaram possível obter ganhos perceptíveis de performance sem um aumento considerável no custo. Justamente por isso elas se popularizaram rapidamente, substituindo as memórias SDR em um espaço de menos de um ano.

Os módulos DDR podem ser vendidos tanto segundo sua frequência de operação quanto segundo sua taxa de transferência.

DDR-200 (100 MHz) = PC1600

DDR-266 (133 MHz) = PC2100

DDR-333 (166 MHz) = PC2700

DDR-400 (200 MHz) = PC3200

DDR-466 (233 MHz) = PC3700

DDR-500 (250 MHz) = PC4000

Assim como no caso dos módulos SDR, existem módulos de memória DDR CL2 e CL3, sendo que nos CL2 o tempo do acesso inicial é reduzido em um ciclo, resultando em um pequeno ganho de desempenho. Como as DDR realizam duas operações por ciclo, surgiram também os módulos CL2.5, que ficam no meio do caminho.

MEMÓRIAS

As especificações dos módulos indicam a frequência máxima para a qual seu funcionamento foi comprovado. Nada impede que você use o módulo a uma frequência mais baixa que o especificado; você pode usar um módulo DDR-400 em uma placa-mãe configurada para trabalhar a 133 MHz, por exemplo, mas nesse caso não existe ganho de desempenho com relação a um módulo DDR-266, com exceção de pequenas diferenças relacionadas ao valor CAS ou à temporização dos dois módulos.

Quase sempre, é possível também usar o módulo a frequências um pouco mais altas que o especificado, fazendo overclock. O módulo DDR-400 poderia funcionar então a 215 MHz, por exemplo. Fazer overclock sem aumentar a tensão da memória não traz perigo para os módulos (mesmo a longo prazo), porém você também não tem garantia de estabilidade. Normalmente os módulos CL2 ou CL2.5 suportam melhor os overlocks, já que o controlador tem mais margem para aumentar a temporização dos módulos para compensar o aumento na frequência.

Ao misturar dois módulos de especificações diferentes, é necessário nivelar por baixo, usando a frequência suportada pelo módulo mais lento. Justamente por isso, nem sempre é conveniente aproveitar os módulos antigos ao fazer upgrade de memória, pois você acaba subutilizando os novos módulos, obrigando-os a acompanharem as temporizações dos módulos antigos. Continuando, quase todos os módulos de memória SDRAM ou DDR possuem um chip de identificação chamado de "**SPD**" (Serial Presence Detect), que armazena os códigos de identificação do módulo, detalhes sobre a frequência, tempos de acesso, CAS latency e outras especificações. Estas informações são exibidas por programas de identificação, como o CPU-Z e o Sandra. No Linux, você pode ler as informações gravadas no chip usando o script "decode-dimms.pl" (você pode encontrá-lo usando o comando "locate"), que faz parte do pacote "lm-sensors". Ele retorna uma longa lista de informações sobre cada um dos módulos instalados na máquina, como neste exemplo:

Memory Serial Presence Detect Decoder

By Philip Edelbrock, Christian Zuckschwerdt, Burkart Lingner,

Jean Delvare and others

Version 2.10.1

Decoding EEPROM: /sys/bus/i2c/drivers/eeprom/0-0050

Guessing DIMM is in bank 1

---== SPD EEPROM Information ==---

EEPROM Checksum of bytes 0-62 OK (0x8C)

MEMÓRIAS

of bytes written to SDRAM EEPROM 128

Total number of bytes in EEPROM 256

*Fundamental Memory type **DDR SDRAM***

SPD Revision 0.0

---=== Memory Characteristics ===---

Maximum module speed 400MHz (PC3200)

Size 512 MB

tCL-tRCD-tRP-tRAS 3-3-3-8

Supported CAS Latencies 3, 2.5, 2

Supported CS Latencies 0

Supported WE Latencies 1

Minimum Cycle Time (CAS 3) 5 ns

Maximum Access Time (CAS 3) 0.65 ns

Minimum Cycle Time (CAS 2.5) 6 ns

Maximum Access Time (CAS 2.5) 0.7 ns

Minimum Cycle Time (CAS 2) 7.5 ns

Maximum Access Time (CAS 2) 0.75 ns

---=== Manufacturing Information ===---

Manufacturer Kingston

Manufacturing Location Code 0x04

Part Number K

Manufacturing Date 0x001E

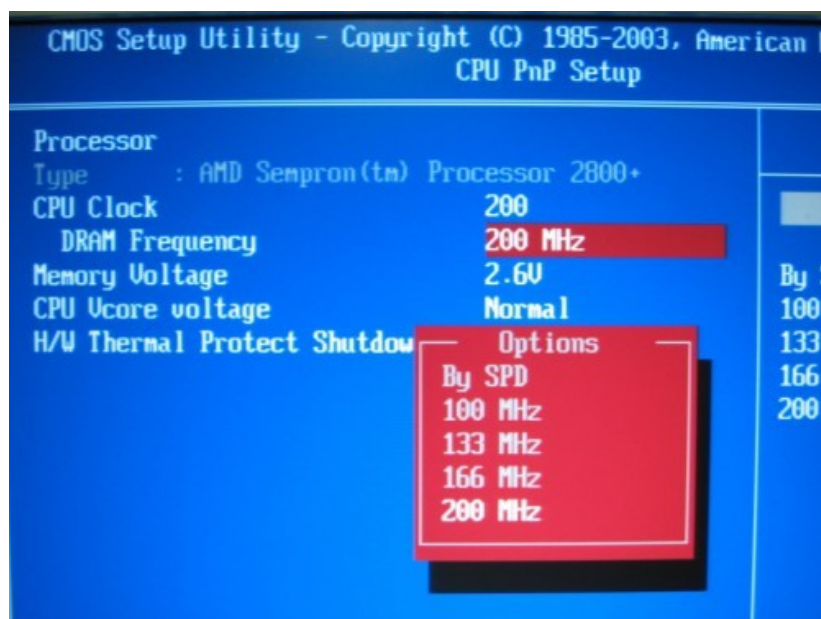
Assembly Serial Number 0x6B376D48

Pelas informações, podemos ver que se trata de um módulo DDR-400 (PC3200) de 512 MB da Kingston. Veja que o módulo suporta o uso de CAS 3, 2.5 ou 2, mas em seguida é especificado que o tempo mínimo de acesso usando CAS 3 são 5 ns e usando CAS 2 são 7.5 ns. Ou seja, o módulo só é capaz de usar CAS 2 em frequências mais baixas. Ao operar a 200 MHz, sua frequência nominal, ele passa automaticamente a usar CAS 3. Apesar das especificações serem um pouco confusas, elas indicam que na verdade tenho em mãos um módulo CL3. O SPD é um pequeno chip de memória EEPROM, com apenas 128 ou 256 bytes, que pode ser localizado facilmente no módulo:



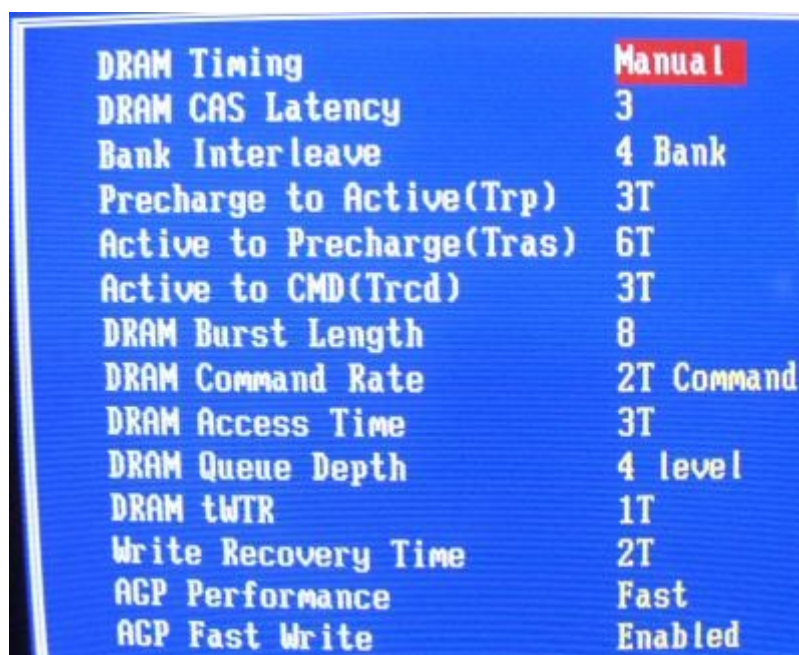
Graças a ele, a placa-mãe pode utilizar automaticamente as configurações recomendadas para o módulo, facilitando a configuração. De qualquer maneira, você pode também desativar a configuração automática (By SPD) e especificar sua própria configuração através do Setup.

A maioria das placas atuais permite que a memória opere de forma assíncrona com o clock da placa-mãe, permitindo que a placa-mãe opere a 166 MHz, enquanto a memória opera a 200 ou 233 MHz, por exemplo. Ao usar um módulo antigo, também é possível fazer o contrário, mantendo a placa-mãe a 200 MHz, mas configurando a memória para operar a 133 MHz, por exemplo. Basta localizar a opção "DRAM Frequency", "Memclock Value" ou similar:



Na maioria das placas é possível ajustar manualmente o valor CAS. Isso pode ser útil ao fazer overclock, pois um módulo DDR-400, pode não conseguir trabalhar estavelmente a 233 MHz (por exemplo), mantendo o CAS em 2 tempos, mas pode funcionar perfeitamente se o tempo for aumentado para 3 tempos. O inverso também é possível. Um módulo DDR-400 CAS 3 poderia vir a trabalhar estavelmente com CAS 2 se a frequência fosse reduzida para 166 MHz, por exemplo, oferecendo uma boa flexibilidade para quando você tem tempo disponível e quer chegar ao melhor desempenho possível.

Muitas placas vão mais longe, oferecendo conjuntos completos de ajustes, como neste segundo screenshot:



Brincar com a frequência e tempos de acesso da memória não oferece riscos para o equipamento. No máximo você pode precisar limpar o setup, para que o micro volte a inicializar depois de tentar usar uma configuração não suportada pelos módulos.

O maior risco está em aumentar a tensão usada pelos módulos (Memory Voltage). Está comprovado que pequenos aumentos na tensão aumentam a possibilidade dos módulos trabalharem estavelmente a frequências mais altas, sobretudo nos módulos DDR2, que dissipam mais calor. O problema é que isso também pode abreviar a vida útil dos módulos, por isso nem sempre é uma boa ideia. Aumentos de até 5 a 8% estão dentro do limite de tolerância dos circuitos e não oferecem grandes riscos. Você pode usar 2.65V em um módulo DDR ou 1.9V em um módulo DDR2, por exemplo, mas aumentos mais agressivos resultam quase sempre em uma redução significativa da vida útil dos módulos. Muitas placas oferecem a opção de aumentar as tensões dos módulos em 30% ou mais, o que pode danificá-los em poucas horas.

SEGUNDA PARTE

Introdução

Seguindo a tendência inaugurada pelas memórias DDR, as DDR2 novamente duplicaram a taxa de transferência, realizando agora 4 transferências por ciclo. Novamente, as células de memória continuam trabalhando na mesma frequência anterior e o acesso inicial continua demorando aproximadamente o mesmo tempo. Entretanto, as demais operações dentro do burst passaram a ser realizadas em apenas um quarto de ciclo de clock. Usando memórias DDR2, um burst de 8 leituras demoraria apenas 6.75 ciclos de clock ($5 - \frac{1}{4} - \frac{1}{4} - \frac{1}{4} - \frac{1}{4} - \frac{1}{4} - \frac{1}{4} - \frac{1}{4}$), contra 8.5 ciclos nas DDR e 12 nas SDR.

Como você pode imaginar, a diferença é maior em aplicativos que precisam manipular grandes blocos de dados e menor em aplicativos que leem pequenos blocos de dados espalhados. Em nenhuma situação prática a transferência chega realmente a dobrar; dizer que as "DDR2 são duas vezes mais rápidas" é apenas uma figura de linguagem.

Em 2005, quando os primeiros módulos DDR2-533 chegaram ao mercado, eles rapidamente ganharam a fama de "lentos", pois eram comparados a módulos DDR-400 ou DDR-466, que já estavam entrincheirados.

Embora um módulo DDR2 ganhe de um DDR da mesma frequência em todos os quesitos

MEMÓRIAS

(um DDR2-800 contra um DDR-400, por exemplo), o mesmo não acontece se comparamos módulos de frequências diferentes. Um módulo DDR2-533 opera a apenas 133 MHz, por isso acaba realmente perdendo para um DDR-400 (200 MHz) em muitas aplicações, já que o ganho ao realizar 4 operações por ciclo acaba não sendo suficiente para compensar a diferença na frequência de operação das células de memória. Vale lembrar que um módulo DDR2-533 trabalha com tempos de latência similares a um módulo DDR-266.

Realizar bursts de leituras rápidas pode não ser a forma mais perfeita de criar memórias mais rápidas (devido ao lento ciclo inicial), mas é sem dúvida a mais simples e barata. A frequência de operação das memórias aumenta de forma gradual, conforme são melhoradas as técnicas de produção. Realizar mais leituras por ciclo de clock é a única forma simples de melhorar exponencialmente a taxa de transferência dos módulos. A dificuldade em criar chips de memória capazes de operar a frequências (reais) mais altas é similar à que temos no caso dos processadores. Não é possível criar um processador capaz de operar ao dobro do clock de uma hora para a outra, mas é possível criar um processador dual-core ou quad-core, por exemplo. No caso das memórias é mais simples, pois você pode ler vários endereços simultaneamente (ou quase), fazendo apenas mudanças nos circuitos

Memórias DDR2

Dependendo da fonte, você pode ler tanto que as memórias DDR2 operam ao dobro da frequência das DDR quanto que elas realizam quatro transferências por ciclo em vez de duas. Nenhuma das duas explicações estão erradas, mas ambas são incompletas.

Como disse, as células de memória continuam trabalhando na mesma frequência das memórias SDR e DDR, mas os buffers de entrada e saída, responsáveis por ler os dados, passaram a operar ao dobro da frequência. É justamente esta frequência que é "vista" pelo restante do sistema, de forma que a maioria dos aplicativos de diagnóstico mostram a frequência dobrada usada pelos circuitos de entrada e não a frequência real das células de memória.

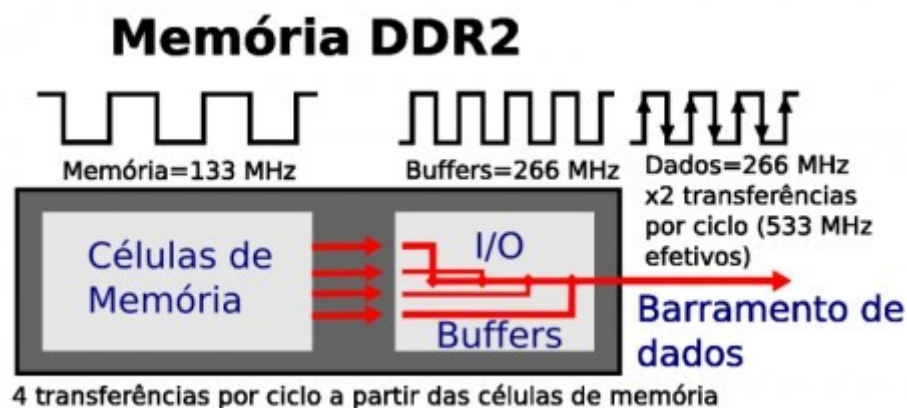
Devido a esta ambiguidade, não é errado dizer que os módulos DDR2 operam ao dobro da frequência dos DDR (os buffers e outros circuitos de apoio realmente operam), nem que são realizadas 4 leituras por ciclo (já que as células de memória continuam operando à mesma frequência).

Ao realizar uma leitura, o controlador de memória gera quatro sinais distintos, que ativam a leitura de quatro endereços adjacentes (recurso batizado de "4-bit prefetch"). As quatro leituras são feitas simultaneamente e os dados são entregues ao buffer, que se encarrega de despachá-los através

MEMÓRIAS

do barramento principal.

Presumindo que o módulo DDR2 do exemplo operasse a 133 MHz, teríamos as células de memória ainda operando na mesma frequência, mas agora entregando 4 leituras de setores sequenciais por ciclo. Os buffers e o barramento de dados operam agora a 266 MHz, de forma que as 4 leituras podem ser enviadas em 2 ciclos, com duas transferências por ciclo. Os dois ciclos do barramento são realizados no mesmo espaço de tempo que apenas um ciclo das células de memória:



Como vimos, as células de memória podem ser grosseiramente comparadas a uma planilha eletrônica, com inúmeras linhas e colunas. Não existe uma grande dificuldade em ler vários endereços diferentes simultaneamente, desde que o fabricante consiga desenvolver os circuitos de controle necessários. Graças a isso, o desenvolvimento das memórias tem sido focado em realizar mais leituras por ciclo, combinado com aumentos graduais nas frequências de operação.

Quando as memórias DIMM surgiram, ainda na época do Pentium II, os módulos mais rápidos operavam a 100 MHz (os famosos módulos PC-100). Atualmente temos chips de memória de até 333 MHz que, combinados com as 4 leituras por ciclo do padrão DDR2, resultam em módulos com transferência teórica de até 10.6 GB/s:

DDR2-533 (133 MHz) = PC2-4200

DDR2-667 (166 MHz) = PC2-5300

DDR2-800 (200 MHz) = PC2-6400

DDR2-933 (233 MHz) = PC2-7500

DDR2-1066 (266 MHz) = PC2-8500

DDR2-1200 (300 MHz) = PC2-9600 (extra-oficial)

DDR2-1333 (333 MHz) = PC2-10600 (extra-oficial)

O CAS latency dos módulos DDR2 é medido em termos de ciclos do circuito controlador,

MEMÓRIAS

por isso são normalmente o dobro do que nos módulos DDR. É como em "duas unidades" ou "quatro metades", no final dá no mesmo. Um módulo DDR2-800 com CAS latency 4 possui o mesmo tempo de acesso que um DDR-400 com CAS latency 2.

Normalmente, as especificações das memórias DDR2 incluem não apenas o CAS latency (**tCL**), mas também o RAS to CAS delay (**tRCD**), Row Precharge Time (**tRP**) e RAS Activate to Charge (**tRAS**). Estes mesmos valores podem ser encontrados nas especificações de módulos DDR e SDR, mas com as memórias DDR2 os fabricantes passaram a divulgá-los de forma mais aberta, usando qualquer redução nos valores para diferenciar seus módulos dos concorrentes. Temos então módulos DDR2-800 "4-4-4-12" ou "5-5-5-15", por exemplo.

O primeiro número é o CAS latency, que já conhecemos. O seguinte é o RAS to CAS delay, que é o tempo que o controlador precisa esperar entre o envio do endereço RAS e o CAS. Para realizar uma leitura, o controlador envia o sinal RAS, espera o tempo referente ao RAS to CAS delay, envia o sinal CAS, aguarda o número de ciclos referente a ele e então finalmente tem a leitura. Em um módulo DDR2 4-4-4-12, tanto o tCL quanto o tRCD demoram 4 ciclos, de forma que o acesso inicial demoraria um total de 8 ciclos. Em um módulo 5-5-5-15, o tempo subiria para 10 ciclos.

É importante lembrar (mais uma vez :) que aqui estamos falando de ciclos dos circuitos de acesso, que trabalham ao dobro da frequência. Os 8 ciclos de um módulo DDR2 equivalem ao mesmo espaço de tempo consumido por 4 ciclos de um módulo DDR ou SDR.

Junto com o ciclo inicial, o controlador pode realizar um burst de mais 7 leituras (totalizando 8). Cada uma destas leituras adicionais consome o equivalente a meio ciclo do controlador (ou um quarto de ciclo das células de memória). Caso ele precise de mais dados dentro da mesma linha, ele repete o envio do sinal CAS e realiza um novo burst de leituras.

Note que o controlador só precisa enviar o sinal RAS ao mudar a linha ativa, de forma que o tRCD só entra na conta no primeiro acesso. Para os seguintes, temos apenas o tempo referente ao tCL. Caso o controlador precise realizar 24 leituras (dentro da mesma linha), num módulo DDR2 4-4-4-12, teríamos 11.5 ciclos ($8+3.5$) para as 8 primeiras leituras e mais 15 ciclos ($4+3.5+4+3.5$) para as 16 leituras subsequentes. É por causa dessa peculiaridade que os módulos DDR e DDR2 não possuem mais o "full-page mode" suportado pelas memórias SDRAM; ele deixou de ser necessário.

O Row Precharge Time (tRP) entra em ação quando o controlador precisa alternar entre diferentes linhas. Cada linha inclui 512 endereços de memória, o equivalente a 4 KB de dados. As linhas são divididas em 4 ou 8 páginas, de forma que um módulo DDR2 de 1 GB teria 8 páginas de 32.768 linhas, ou 4 páginas de 65.536 linhas. Com tantas linhas e páginas diferentes, não é difícil imaginar que o chaveamento entre elas é muito frequente. Quando falo em linhas e páginas, tenha

MEMÓRIAS

em mente que essa é apenas a forma como o controlador de memória "enxerga" o módulo. Fisicamente, mesmo os bits de uma mesma linha estão espalhados pelos vários chips do módulo. Antes de poder acessar uma determinada linha, o controlador de memória precisa carregá-la (precharge). Isso consiste em recarregar os capacitores dentro das células de memória, facilitando a leitura dos dados. O Row Precharge Time (tRP) é justamente o tempo necessário para fazer o carregamento, necessário antes de chavear para outra linha, seja no mesmo banco, seja num banco diferente.

Sendo assim, mesmo que seja necessário ler um único setor, a leitura demorará (em um módulo 4-4-4-12) 4 ciclos para o tRP, 4 ciclos para o tRCD, 4 ciclos para o tCL, totalizando 12 ciclos. Estes 12 ciclos são justamente o tempo referente ao RAS Activate to Charge (tRAS), que é o tempo mínimo para realizar uma leitura completa. O tRAS é sempre proporcional aos três primeiros valores, pois é justamente a soma dos três. É por isso que ele sempre é mais alto em módulos com CAS latency mais alto.

É possível reduzir o tRAS utilizando um recurso chamado Additive Latency, onde o comando para iniciar o precharge do banco seguinte pode ser enviado antes que a leitura atual termine. Isso faz com que o tempo total da leitura seguinte seja reduzido em 1 ou até mesmo 2 ciclos. Esse é o caso dos módulos 5-4-4-11 ou 4-4-4-11, por exemplo. Em outros casos é necessário um ciclo adicional para fechar o banco, que aumenta o tRAS em vez de diminuir. De qualquer forma, o tRAS é, entre os quatro, o parâmetro que menos influi no desempenho, pois só faz alguma diferença real quando o sistema precisa realizar séries de acessos rápidos, a linhas diferentes.

Bem, esta última parte exigiu uma explicação mais complexa que o habitual. Como você pode ver, os tempos de acesso dos módulos DDR2 é um pouco mais complexo do que pode parecer à primeira vista.

Entretanto, o mais importante dos 4 valores continua sendo o primeiro (o bom e velho CAS latency, ou tCL), já que o tRCD e o tRP são quase sempre iguais a ele e o tRAS é a soma dos três. Ou seja, se o CAS latency é mais baixo, automaticamente os demais valores também são.

O processo de transição: Com relação ao suporte por parte dos processadores e chipsets, o lançamento das memórias DDR2 teve um impacto diferente para a Intel e a AMD. Para a Intel, a migração para as memórias DDR2 foi mais simples, já que na plataforma Core o controlador de memória é incluído no chipset, de forma que aderir a uma nova tecnologia demanda apenas modificações nos chipsets e placas.

A Intel oferece suporte a memórias DDR2 em seus chipsets desde o i915P, lançado em 2004. Inicialmente, os chipsets ofereciam tanto suporte a memórias DDR quanto DDR2, de forma que ficava a cargo do fabricante escolher qual padrão utilizar. Existem inclusive placas híbridas, que

suportam ambos os padrões, como a ECS 915P-A, que possuem dois slots de cada tipo, permitindo que você escolha qual padrão utilizar. A partir de um certo ponto, entretanto, as memórias DDR2 caíram de preço e quase todas as placas soquete 775 passaram a vir com suporte exclusivo a memórias DDR2.

Para a AMD, a mudança foi mais tortuosa, já que o Athlon 64 e derivados utilizam um controlador de memória embutido diretamente no processador, desenvolvido de forma a minimizar os tempos de acesso.

Por um lado isto é bom, pois oferece um ganho real de desempenho, mas por outro é ruim, pois qualquer mudança no tipo de memória usado demanda mudanças no processador e no soquete usado. Foi justamente isso que aconteceu quando a AMD decidiu fazer a migração das memórias DDR para as DDR2. Além das mudanças internas no processador e controlador de memória, o soquete 754 foi substituído pelo soquete 939 e em seguida pelo AM2, quebrando a compatibilidade com as placas antigas.

Com a adoção por parte da AMD, a procura (e consequentemente a produção) das memórias DDR2 aumentou bastante, fazendo com que os preços passassem a cair rapidamente. A partir do final de 2006, os preços dos módulos de memória DDR2 (nos EUA) caíram a ponto de passarem a ser mais baratos que os módulos DDR regulares. Como sempre, as mudanças chegam ao Brasil com alguns meses de atraso, mas a partir do início de 2007 as memórias DDR2 passaram a ser encontradas por preços inferiores às DDR por aqui também.

Outra questão importante é a popularização de módulos DDR2-1066 e também de módulos de baixa latência, com temporização de 4-4-3-11, ou mesmo 3-3-3-9. Ao contrário da geração inicial de módulos DDR2, que ofereciam mais banda, mas em compensação trabalhavam com tempos de acesso muito mais altos, esta segunda geração de módulos DDR2 é indiscutivelmente mais rápida. O ganho prático em utilizar memórias DDR2 não é tão grande assim, já que o barramento com a memória é apenas um dos fatores que determina o desempenho do PC. Entretanto, a partir do momento em que a nova tecnologia passa a ser mais barata que a anterior, qualquer ganho, por menor que seja, é muito bem-vindo.

Memórias DDR3

Podemos dizer que as memórias DDR2 atingiram seu pico evolutivo nos módulos DDR2-1066, que é o último padrão reconhecido pelo JEDEC. Naturalmente, é possível encontrar módulos mais rápidos no mercado, como os módulos DDR2-1200 "SLI-Ready" marqueteados pela nVidia e os módulos DDR2-1333 (produzidos em pequenos volumes por vários fabricantes), mas eles são

MEMÓRIAS

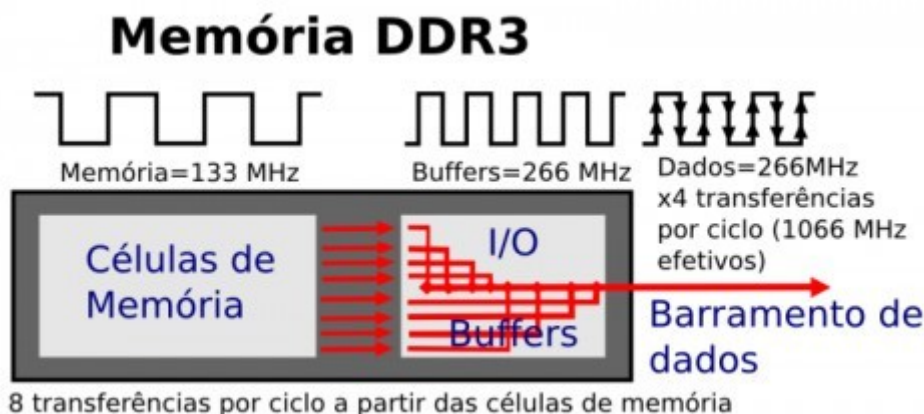
consideravelmente mais caros e o ganho de desempenho na prática é pequeno. Oficialmente, eles são apenas módulos DDR2-1066 overclocados, já que não existem padrões para módulos DDR2-1200 e DDR2-1333 certificados pelo JEDEC.

Considerando que em um módulo DDR2-1066 as células de memória operam a nada menos que 266 MHz (uma evolução expressiva em relação aos módulos PC-100 e PC-133 do começo do milênio, onde as células operavam a apenas 100 ou 133 MHz), não é de se estranhar que os fabricantes tenham enfrentado dificuldades a partir daí.

A solução veio com as memórias DDR3, que mais uma vez duplicaram a frequência efetiva dos módulos, realizando agora 8 transferências por ciclo de clock, contra as 4 transferências do DDR2.

A grande sacada é que o aumento na frequência é (novamente) obtido através do acesso simultâneo a endereços adjacentes e não através do aumento da frequência real das células de memória, o que permitiu estender o uso das células de 133 a 266 MHz por mais uma geração.

Em um módulo DDR3-1066, por exemplo, as células de memória operam a apenas 133 MHz, com os buffers de dados operando a 266 MHz e realizando 4 transferências por ciclo, resultando em uma frequência efetiva de 1066 MHz. Em módulos DDR3 com células operando a 266 MHz, a frequência efetiva é de impressionantes 2133 MHz e com a maturação da tecnologia parece ser apenas questão de tempo até que tenhamos módulos ainda mais rápidos no mercado.



Inicialmente, os módulos DDR3 foram lançados em versão DDR3-1066 (133 MHz x 8) e DDR3-1333 (166 MHz x 8), seguidos pelo padrão DDR3-1600 (200 MHz x 8). Os três padrões são também chamados de (respectivamente) PC3-8500, PC3-10667 e PC3-12800, nesse caso dando ênfase à taxa de transferência teórica:

DDR3-1066 (133MHz) = PC3-8500

DDR3-1333 (166 MHz) = PC3-10667

DDR3-1666 (200 MHz) = PC3-12800

Apesar do aumento no número de transferências por ciclo, os buffers de dados continuam

trabalhando a apenas o dobro da frequência das células de memória. Ou seja, a frequência interna (das células de memória) de um módulo DDR3-1600 é de 200 MHz e a frequência externa (dos buffers de dados) é de 400 MHz. As células de memória realizam 8 transferências por ciclo de clock (em vez de 4, como nas DDR2) e os buffers de dados (que operam ao dobro da frequência) realizam 4 transferências por ciclo de clock, em vez de apenas duas, como nos módulos DDR2.

Se as mudanças parassem por aí, os módulos DDR3 não ofereceriam ganhos muito grandes na prática, pois o tempo de latência inicial continuaria sendo o mesmo que nos módulos DDR2 (já que não houve mudança na frequência das células de memória). Se um módulo DDR3 operasse com tempos de acesso 10-10-10-30, os ganhos seriam pequenos em relação a um DDR2 5-5-5-15, já que só haveria ganho nos acessos subsequentes.

Para evitar isso, os módulos DDR3 incluem um sistema integrado de calibragem do sinal, que melhora de forma considerável a estabilidade dos sinais, possibilitando o uso de tempos de latência mais baixos, sem que a estabilidade seja comprometida.

Os módulos DDR3 utilizam também 8 bancos em vez de 4, o que ajuda a reduzir o tempo de latência em módulos de grande capacidade. Eles também trouxeram uma nova redução na tensão usada, que caiu para apenas 1.5V, ao invés dos 1.8V usados pelas memórias DDR2. A redução na tensão faz com que o consumo elétrico dos módulos caia proporcionalmente, o que os torna mais atrativos para os fabricantes de notebooks.

Somadas todas essas melhorias, os tempos de acesso "reais" dos módulos foram sensivelmente reduzidos. Em vez de trabalharem com tempos de acesso 10-10-10-30, a geração inicial de módulos DDR3 é capaz de trabalhar com temporização 9-9-9-24, ou mesmo 7-7-7-15.

O primeiro chipset a incluir suporte às memórias DDR3 foi o Intel P35, lançado em 2007. Em vez de cometer o mesmo erro que cometeu ao lançar o Pentium 4, quando tentou forçar o uso das memórias Rambus, a Intel adotou uma postura conservadora, equipando o P35 com suporte simultâneo a memórias DDR3 e DDR2 e deixando que os fabricantes de placas escolhessem qual das duas tecnologias utilizar.

Como era de se imaginar, todas as placas mainstream e de baixo custo passaram a suportar exclusivamente memórias DDR2 (que eram muito mais baratas), com as memórias DDR3 ficando relegadas ao mercado de alto desempenho. Isso continuou ao longo de 2008 e 2009, com a mesma fórmula sendo repetida no chipset P45 e nos lançamentos subsequentes.

Isso fez com que a procura pelos módulos DDR3 continuasse fraca e os preços se mantivessem altos. Para complicar, os primeiros módulos DDR3 não ofereciam um ganho de desempenho tangível em relação aos DDR2 na plataforma soquete 775. Embora a frequência efetiva fosse mais baixa, os módulos DDR2 trabalhavam com tempos de acesso mais baixos, o que fazia

MEMÓRIAS

com que a competição fosse acirrada, com os DDR2 se saindo melhor em muitas situações.

Isso levou os fabricantes de memória a apostarem na produção de módulos de alto desempenho, em uma corrida armamentista que deu origem a módulos overclocados, capazes de trabalhar a 2000 MHz ou mais (frequência efetiva), mas que em compensação utilizavam tensões de até 2.0V, muito acima dos 1.5V recomendados. Embora vendidos em pequenas quantidades, estes módulos lideravam os benchmarks e por isso recebiam uma atenção desproporcional.

Um bom exemplo dessa época insana são estes módulos DDR3-2000 (9-9-9-24) da G.Skill, que utilizam tensão de 1.9V e são equipados com um cooler ativo (ligado ao módulo através de um heat-pipe) para manter a temperatura de operação em níveis aceitáveis:



Eles não eram apenas caros (o kit com dois módulos de 2 GB custava nada menos que US\$ 300 na época de lançamento), mas a tensão fazia com que a vida útil fosse reduzida, com muitos módulos apresentando defeitos prematuros depois de alguns meses de uso, um problema que atingiu também módulos de outros fabricantes.

A Intel resolveu colocar ordem no galinheiro com o lançamento do Core i7, limitando o suporte oficial aos módulos DDR3-1066 e DDR3-1333 e advertindo que o uso de tensões superiores a 1.65 poderia danificar o controlador de memória integrado ao processador.

Isso obrigou os fabricantes a se concentrarem na fabricação de módulos de baixa frequência e baixa latência, em vez de continuarem a investir no simples aumento das frequências. Outra mudança positiva foi que os módulos voltaram a utilizar tensões "normais", variando de 1.5V nos módulos value, a 1.65V nos módulos de alto desempenho, dentro dos valores recomendados pela Intel para

uso em conjunto com o i7.

A AMD adotou uma postura similar à da Intel durante a fase de transição, lançando a plataforma AM3 (com o uso de memórias DDR3) porém equipando os processadores Phenom II com controladores híbridos, com suporte simultâneo a memórias DDR2 e DDR3. Isso permitiu que os processadores continuassem compatíveis com as placas AM2+, permitindo que você decidisse entre usar memórias DDR2 ou DDR3 ao escolher a placa-mãe.

As memórias DDR2 demoraram quase 3 anos para se popularizarem desde a introdução do chipset i915P, em 2004, ultrapassando as vendas das memórias DDR antigas apenas a partir de 2007.

Apesar das diferenças na implementação, as memórias DDR3 acabaram seguindo um caminho similar, com os módulos inicialmente custando muito mais caro e caindo a um nível próximo dos preços dos módulos DDR2 apenas a partir do início de 2010, novamente quase três anos depois do lançamento das primeiras placas e chipsets.

A maldição dos 32 bits: o limite de 3 GB

Assim como no caso dos HDs, existiram diversos limites com relação ao endereçamento da memória RAM durante a história dos micros PCs, causados por limitações do sistema operacional, limitações do chipset e limitações dos processadores. Mesmo os processadores e sistemas operacionais de 64 bits possuem limites com relação à quantidade máxima de memória que pode ser endereçada, embora muito mais elevados. As versões originais do Athlon 64 são capazes de endereçar até 1 terabyte de memória RAM (independentemente da placa-mãe usada) e este limite pode ser expandido no futuro. O grande problema são os processadores e os sistemas operacionais de 32 bits.

Tudo começou com o limite de 640 KB de memória do processador 8088, usado nos primeiros PCs. Na verdade, o 8088 utilizava um total de 20 bits para o endereçamento da memória, divididos em 16 bits "nativos", que permitiam endereçar páginas de 64 KB e mais 4 bits adicionais, que permitiam endereçar 16 páginas, totalizando 1 MB.

Por decisão da equipe de desenvolvimento, apenas os primeiros 640 KB (chamados de memória convencional) ficavam disponíveis para uso do sistema operacional e aplicativos. Os 384 KB restantes (chamados de memória estendida) eram usados para endereçar a memória presente em outros dispositivos, como o BIOS da placa-mãe e a memória da placa de vídeo.

Na época dos primeiros PCs, a memória RAM era muito cara, de forma que não era comum

MEMÓRIAS

o uso de mais do que 256 KB, mas, mesmo que fosse utilizado 1 MB completo, apenas os primeiros 640 KB ficariam disponíveis. Os 384 KB adicionais podiam ser utilizados para fazer cache do BIOS e da memória da placa de vídeo (o que melhorava o desempenho), mas não para uso geral.

Para manter compatibilidade com os aplicativos antigos, mesmo os processadores atuais são capazes de operar em modo real, onde simulam o funcionamento de um 8088, acessando apenas os primeiros 640 KB de memória. É dentro deste limite que rodam o MS-DOS e aplicativos antigos.

Quando o Windows, Linux ou qualquer sistema operacional é carregado, o processador passa para o modo protegido, onde é capaz de acessar toda a memória disponível. Desde o 386, todos os processadores utilizam 32 bits para o endereçamento da memória, o que permite endereçar até 4 GB. Esse limite é chamado de **VAS** (Virtual Address Space) e indica justamente o total de memória que o sistema é capaz de endereçar, incluindo não apenas a memória RAM, mas também a memória da placa de vídeo e outros dispositivos. O problema reside justamente aí.

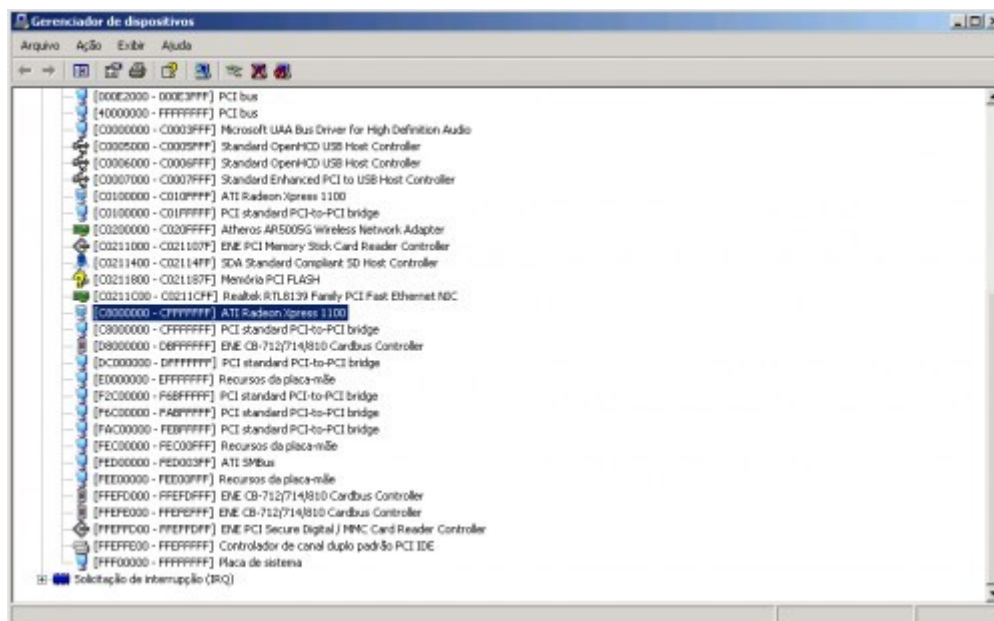
Imagine que você resolvesse montar um PC com 4 GB de memória RAM e duas placas de vídeo com 1 GB de memória RAM em CrossFire ou SLI. Como a memória das placas de vídeo consome espaço do Virtual Address Space, o sistema seria capaz de acessar apenas os primeiros 2 GB da memória (um pouco menos na prática, já que mais alguns blocos serão reservados a outros dispositivos), de forma que usar 4 GB acabaria sendo um desperdício de dinheiro.

Isso ocorre não apenas ao utilizar um processador de 32 bits, mas também ao usar um processador de 64 bits em conjunto com um sistema operacional de 32 bits, incluindo o Windows XP e as versões de 32 bits do Vista e Windows 7.

De uma forma geral, não é recomendável utilizar mais do que **3 GB** ao usar um sistema operacional de 32 bits, pois é justamente a área entre os 3 e 4 GB do Virtual Address Space que é utilizada pelo processador para endereçar a memória de dispositivos diversos. Assim como dois corpos não podem ocupar o mesmo espaço ao mesmo tempo, dois dispositivos não podem compartilhar a mesma área de endereços, o que faz com que porções da memória RAM que invadem áreas reservadas a outros dispositivos simplesmente não sejam vistas pelo sistema.

Você pode verificar as áreas de memória reservadas através do gerenciador de dispositivos do Windows, usando a opção Exibir > Recursos por tipo > Memória:

MEMÓRIAS



Os endereços fornecidos aqui estão em hexa, mas você pode usar a própria calculadora do Windows para convertê-los em números decimais. No caso, por exemplo, tenho a placa de vídeo utilizando os endereços C8000000 a CFFFFFFF, que correspondem aos bytes de 3.355.443.200 a 3.489.660.927 e uma série de dispositivos a partir do "Recursos da placa-mãe" que ocupam praticamente todos os endereços do byte 3.758.096.384 ao 4.294.967.295 (que corresponde justamente ao final da área de 4 GB endereçada pelo sistema). Se instalasse uma placa 3D offboard, com 256 MB, ela ocuparia os bytes de 3.221.225.472 a 3.489.660.927, que correspondem justamente aos 256 MB da placa.

Se você se contentar em utilizar uma placa de vídeo low-end (com pouca memória de vídeo), pode ficar com até 3.5 GB de endereços disponíveis, dependendo do sistema operacional usado.

No caso das versões de 32 bits do Vista, por exemplo, o limite máximo (segundo o <http://support.microsoft.com/kb/929605>) é de 3.12 GB. Ou seja, na maioria dos casos, não vale a pena pagar por mais 4 GB de memória, já que apenas uma pequena parte do último GB seria utilizada. Melhor se contentar com 3 GB.

Mesmo ao utilizar um processador de 64 bits, combinado com um sistema operacional de 64 bits, um grande volume de endereços entre os 3 e 4 GB de memória continuam reservados, de forma a manter compatibilidade com os programas de 32 bits, fazendo com que, novamente, pelo menos 512 MB entre os 3 e 4 GB da memória não sejam usados. A vantagem nesse caso é que você pode instalar mais de 4 GB de memória. Com 8 GB, por exemplo, você ficaria com de 7 a 7.5 GB utilizáveis.

MEMÓRIAS

Algumas placas de 64 bits oferecem a opção "Memory Hole Remapping" (ou similar) no setup. Ao ativar esta opção, as áreas de memória reservadas aos dispositivos é movida para uma área mais alta do Virtual Address Space (que nos processadores de 64 bits é muito maior), liberando a maior parte da memória antes inacessível por causa deles.

Outras placas suportam o Memory Hoisting (da palavra "hoist", que significa levantar ou suspender), um recurso que segue outro princípio, deixando os endereços dos dispositivos onde estão e remapeando os trechos de memória subscritos por eles para áreas mais altas do espectro de endereços, permitindo que o sistema tenha acesso a elas.

Uma observação é que estas duas opções causam muitos problemas de compatibilidade com aplicativos, por isso você só deve utilizá-las caso realmente faça questão de ter acesso à toda a memória instalada.

Além das limitações relacionadas ao Virtual Address Space, existem outros limites de memória, impostos pelo chipset ou por limitações do sistema operacional usado.

A maioria dos chipsets antigos, para placas soquete 7, por exemplo, eram capazes de endereçar apenas 128 ou 256 MB de memória RAM. O chipset Intel i815, usado em uma grande parte das placas para processadores Pentium III e Celeron era capaz de endereçar apenas 512 MB, enquanto o 915GL (para o Pentium 4) era capaz de endereçar 2 GB. Até pouco tempo atrás, o suporte a 4 GB ou mais de memória estava restrito a alguns chipsets destinados a servidores.

O Windows 95/98/SE é capaz de endereçar até 512 MB de memória RAM. Programas como o cacheman permitem utilizar estas versões antigas do Windows em PCs com mais de 512 MB de RAM, mas o desempenho acaba sendo pior do que com apenas 512.

O Windows XP e as versões de 32 bits do Vista são capazes de endereçar 4 GB, de forma que o limite fica por conta do Virtual Address Space (os pouco mais de 3 GB). O maior problema são as versões "populares" do Windows, onde o limite é artificialmente reduzido, de forma a restringir o uso do sistema aos PCs de baixo custo. O XP Starter possui um limite de 256 MB (1 GB na versão para netbooks), enquanto o Vista Starter está limitado a 1 GB.

Nas versões de 64 bits do Vista, os limites são estendidos, mas ainda existem limitações. O Vista Home Basic está limitado a 8 GB, o Home Premium a 16 GB e as demais versões (Business, Enterprise e Ultimate) a 128 GB.

A memória swap não entra na conta, pois ela é acessada através de um componente especializado incluído no processador, chamado de **MMU**(Memory Management Unit) que permite justamente que dados armazenados na memória RAM física sejam movidos para a memória swap (que é na verdade um arquivo ou partição no HD), conforme necessário.

Graças ao MMU é possível complementar os 3 ou 3.5 GB de memória RAM física com uma

MEMÓRIAS

quantidade indefinida de memória swap, limitada apenas ao máximo suportado pelo sistema operacional. A principal questão é que o uso da memória swap é muito limitado em um PC atual, já que ela é muito lenta.

Uma dúvida comum é com relação ao uso do **PAE** (Physical Address Extension), uma extensão para processadores de 32 bits, presente desde o Pentium Pro, que adiciona 4 bits ao endereçamento da memória, permitindo que o processador seja capaz de acessar até 16 páginas de 4 GB cada, totalizando 64 GB. Cada programa continua restrito a um máximo de 4 GB, mas o sistema pode alocar diferentes páginas para aplicativos diferentes, utilizando assim toda a memória disponível.

O PAE já foi muito usado em servidores, mas foi logo substituído pelo uso de processadores e sistemas operacionais de 64 bits. Ele é suportado pelo Windows Server 2003 e pelo Windows 2000, onde pode ser ativado através da opção `"/PAE"` no arquivo `boot.ini`, mas não é uma solução viável para quem precisa usar mais do que 4 GB de memória em um desktop, devido a uma série de problemas relacionados aos drivers.

O PAE apresenta endereços de 64 bits aos drivers, o que torna necessário que eles sejam modificados para utilizar o sistema. Como o uso do PAE sempre foi limitado a servidores, muitos dispositivos não possuem drivers compatíveis e nem sempre é fácil encontrar os disponíveis.

Existem ainda problemas de compatibilidade com diversos aplicativos. Atualmente, é muito mais simples migrar para as versões de 64 bits do Vista (ou mesmo para o XP de 64 bits) do que passar a utilizar o PAE, de forma que é melhor não perder tempo com ele no caso dos desktops.

Outro problema fundamental do PAE é a questão do desempenho. O processador continua sendo capaz de acessar apenas 4 GB de memória por vez e precisa chavear continuamente entre as páginas disponíveis. Além de demorar um certo tempo, cada chaveamento faz com que os dados armazenados nos caches precisem ser completamente substituídos, o que prejudica de forma perceptível o desempenho. Ou seja, tentar usar o PAE em um desktop para acessar mais memória e assim melhorar o desempenho é simplesmente contra produtivo. :)

Uma observação importante é que todas as versões do Windows XP, a partir do SP2 tem o PAE desativado, justamente para evitar problemas de compatibilidade. Nele, a opção `"/PAE"` no `boot.ini` simplesmente não faz nada.

No caso do Linux, o modo de acesso à memória é escolhido durante a compilação do Kernel. Ao utilizar um processador de 32 bits, o Linux oferece suporte nativo a até 4 GB de memória usando o modo normal de operação do processador e a até 64 GB usando o PAE. Ou seja, ele simplesmente acompanha o suporte disponível no hardware, sem nenhuma limitação adicional. Para melhorar o desempenho do sistema em máquinas antigas, que utilizam 1 GB de memória ou

MEMÓRIAS

menos, existe uma terceira opção, onde o Kernel endereça apenas 1 GB de memória, sendo que 896 MB ficam disponíveis para os aplicativos e o restante é reservado para uso do Kernel. Neste modo de operação, o comando "free" vai reportar que existem apenas 896 MB de memória disponível, mesmo que você possua 1 GB ou mais.

É possível escolher entre as três opções ao compilar o Kernel, na opção "Processor Type and Features > High Memory Support". Até por volta de 2005, muitas distribuições vinham com o suporte a apenas 1 GB ativado por padrão, mas atualmente a grande maioria utiliza a opção "4 GB". É possível mudar a opção ao recompilar o Kernel manualmente.

A desvantagem de ativar o suporte a 4 GB é que o sistema ficará um pouco mais lento em micros com menos de 1 GB de memória (justamente por isso existe a primeira opção). O suporte a 64 GB só pode ser ativado caso você esteja usando um processador com suporte ao PAE, além de uma placa-mãe compatível.

Naturalmente, tudo isso se aplica apenas ao usar uma distribuição com um Kernel compilado para processadores de 32 bits. Ao usar uma versão de 64 bits, o Kernel acessa toda a memória disponível, limitado apenas ao volume de memória suportado pelo chipset ou processador.

Voltando ao mundo Windows, existe mais uma limitação importante, que é o limite de **2 GB** por aplicativo. Os 3, 3.12 ou 3.5 GB de memória que um sistema de 32 bits é capaz de endereçar é repartida entre todos os programas abertos, mas no caso do Windows, cada programa sozinho não pode usar mais do que 2 GB.

Isso acontece por que, assim como o sistema operacional, os programas de 32 bits endereçam a memória utilizando endereços de 32 bits e por isso são capazes de endereçar 4 GB de memória. No Windows, metade da faixa de endereços de cada programa é reservada para uso do kernel, sobretudo para a comunicação entre o sistema operacional e o aplicativo. Com isso, cada aplicativo fica com apenas metade da sua área de endereços disponível, o que limita seu uso de memória a 2 GB.

Esta é mais uma limitação importante em alguns jogos e aplicativos pesados, que podem facilmente exceder essa marca. Pior, além de não acessarem toda a memória disponível, os aplicativos de 32 bits frequentemente travam sem aviso ao romper a barreira dos 2 GB, exibindo alguma mensagem de erro genérica.

Este limite está profundamente enraizado no sistema, de forma que é quase impossível que ele venha a ser removido em futuras versões. Ele afeta tanto quem utiliza as versões de 32 bits do Windows (com mais de 2 GB de memória RAM instalada), quanto quem roda programas de 32 bits sobre as versões de 64 bits do Windows (neste caso o limite se aplica apenas aos aplicativos de 32 bits, naturalmente).

Uma forma de amenizar o problema, caso você esteja enfrentando problemas de travamento em algum jogo ou aplicativo específico, é modificar a divisão de endereços entre o aplicativo e o kernel.

No Windows XP é possível reservar 3 GB para os aplicativos através da opção `"/3gb"` no arquivo `boot.ini`. Entretanto, usá-la causa problemas de estabilidade em um número muito grande de programas, por isso ela não é necessariamente uma boa ideia. Além disso, apenas programas que explicitamente indicam que são capazes de acessar mais de 2 GB são autorizados pelo sistema a utilizar os endereços adicionais, o que reduz brutalmente o número de programas que são realmente capazes de se beneficiar do uso da opção.

No caso do Vista, é possível ajustar os valores de forma mais flexível através da opção `"IncreaseUserVa"` do `BCDedit`. Através dele você pode reservar, por exemplo, 2.2 ou 2.4 GB para os aplicativos. Usando valores mais baixos que os 3 GB do XP, você evita a maior parte dos problemas de estabilidade. De uma forma geral, funciona bem com a alocação de até 2.6 GB para os aplicativos. A partir daí você corre o risco de ver telas azuis.

Como você pode ver, o uso das opções permite apenas minimizar o problema. A única forma de se livrar dele completamente é rodar um sistema de 64 bits e usar apenas aplicativos de 64 bits sobre ele.

Identificando módulos de memória defeituosos

Apesar da complexidade, não é muito comum que um processador venha com defeito de fábrica, geralmente eles são testados antes do encapsulamento e os que não funcionam são simplesmente descartados. O problema mais comum com relação a eles são os casos de superaquecimento, que podem ser resolvidos com a limpeza do cooler ou uma ventilação adequada. Mas, não podemos dizer o mesmo dos módulos de memória, onde os defeitos são muito mais comuns e infelizmente mais difíceis de detectar. Um agravante é que existem muitos fabricantes diferentes de memórias, todos trabalhando com margens muito baixas, por isso existe sempre a tentação de colocar no mercado módulos que não foram adequadamente testados em fábrica.

A placa-mãe também pode ser responsável por vários erros, já que ela é a encarregada de transportar os dados dos módulos de memória até o processador. Se a placa-mãe corromper os dados pelo caminho você terá travamentos e outros sintomas semelhantes aos causados por um módulo de memória com defeito. Esses erros são cruéis, pois são muitas vezes difíceis de perceber. Eles muitas vezes são confundidos com problemas do sistema operacional.

Seja qual for a causa, erros de memória podem causar muita dor de cabeça, então a melhor

MEMÓRIAS

forma de lidar com o problema é rodar um teste completo sempre que você montar um PC ou instalar um novo módulo de memória.

Existem inúmeros programas desse tipo, um dos melhores é o Memtest86, disponível no: <http://www.memtest86.com/>

A grande vantagem do memtest86 sobre outros testes de memória é que além de pequeno, gratuito e de código aberto, ele é quase automático. Você precisa apenas gravar um CD (ou disquete) e dar boot para que o teste seja iniciado automaticamente. Ele fica num loop eterno, repetindo os testes e indicando os erros que encontrar até que você se sinta satisfeito.

Existem duas opções de download. O mais prático é baixar uma imagem ISO, que pode ser usada para fazer um CD bootável. O arquivo tem apenas 64 kb compactado e 1.6 MB depois de descompactado. No site você vai encontrar também os programas para gerar o disquete de boot no Windows e Linux. Tanto faz usar o CD ou o disquete, o programa é exatamente o mesmo.

É necessário dar boot diretamente no Memtest para que ele possa testar realmente toda a memória do sistema. Se ele rodasse como um programa sobre o Linux ou Windows, não teria como acessar áreas utilizadas pelo sistema e programas, e o teste não seria confiável.

O Memtest86 realiza um total de 9 testes. Os 5 primeiros são relativamente rápidos, mas os 4 testes finais são muito mais rigorosos, capazes de encontrar erros não detectados pelos testes iniciais, mas são em contrapartida muito demorados. Os 8 testes são executados automaticamente, mas o nono (veja detalhes a seguir) precisa ser ativado manualmente, já que é muito demorado. O ideal é deixar o teste correndo em loop durante a madrugada e olhar os resultados no outro dia de manhã.

Se, por acaso, a tabela inicial do Memtest86 informar incorretamente a quantidade de memória, acesse a opção "3" (memory sizing) e, dentro dela, a opção "3" (probe). Isso fará com que o Memtest86 detecte a memória, desprezando as informações do BIOS. Na tela principal, pressione a tecla 5 para ter um sumário com todos os erros encontrados em cada teste:

```
Memtest86 v1.20 | Pass 4% #
Pentium II 3067 Mhz | Test 26% #####
L1 Cache: 8K 134MB/s | Test #3 [Moving inv, 8 bit pattern, ca
L2 Cache: 128K 314MB/s | Testing: 100K - 128M 128M
Memory : 128M 344MB/s | Pattern: 20202020
Chipset : Intel i440fx

WallTime  Cached  RsvdMem  MemMap  Cache  ECC  Test  Pass  Errors
-----
0:00:40  128M      0K  e820-Std  on  off  Std    0    0

(ESC)Reboot (c)configuration (SP)scroll_lock (CR)scroll_unlock
```

Basicamente é isto, não existe muita configuração a fazer. A alma do negócio é ter paciência e deixar ele fazer seu trabalho, se possível por tempo suficiente para realizar o teste longo.

O tempo necessário varia de acordo com o desempenho e principalmente com a quantidade de memória instalada. Dobrar a quantidade de memória dobra o tempo do teste. Usar um processador mais rápido tem pouca influência, pois o gargalo é a velocidade de acesso à memória, não o processamento.

O Memtest86 já vem pré-instalado em muitas distribuições, aparecendo como uma opção no menu de boot. Vamos entender o que faz cada um dos testes:

Teste 0 (Address test, walking ones, no cache): Simplesmente testa o acesso a todos os endereços da memória, algo semelhante com o que o BIOS faz na contagem de memória durante o boot. Ele é o teste mais rápido e serve basicamente para checar a quantidade de memória disponível.

Teste 1 (Address test, own address): Este teste é semelhante ao 0, mas adota uma estratégia diferente, checando endereços e não apenas realizando uma contagem rápida. Isso permite detectar problemas de endereçamento no módulo. Outra diferença é que este teste é feito sem utilizar os caches do processador, para evitar que ele mascare defeitos nos módulos.

Teste 2 (Moving inversions, ones&zeros): Escreve bits 1 e depois bits 0 em todos os endereços da memória. Este algoritmo também utiliza os caches L1 e L2 do processador. É um teste rápido que identifica os erros mais grosseiros, onde algumas células do módulo estão realmente queimadas.

Teste 3 (Moving inversions, 8 bit pat): É aqui que os testes para detectar erros mais complexos começam. O terceiro teste escreve sequências de dados de 8 bits, repetindo o teste 20 vezes com sequências diferentes.

Teste 4 (Moving inversions, random pattern): Este teste é similar ao teste número 3, mas dessa vez usando uma sequência randômica de acessos, que é repetida um total de 60 vezes. Parece um grande exagero, já que este teste é destinado a identificar os mesmos erros que o teste 3, mas é justamente essa "insistência" e o uso de diversas sequências diferentes de operações que torna o memtest tão confiável na detecção de erros, capaz de detectar erros raros ou transitórios, que não aparecem em outros testes.

Teste 5 (Block move, 64 moves): É um pouco mais rigoroso que o teste 4. Continua movendo dados de um endereço para outro da memória, mas agora são movidos blocos de 4 megabits de cada vez. Este teste é repetido 64 vezes.

Teste 6 (Moving inversions, 32 bit pat): Os dados gravados em cada bit de memória são lidos e movidos para o endereço adjacente. É mais ou menos uma combinação dos três testes anteriores, pois testa o endereçamento, leitura e escrita de dados. A operação é repetida 32 vezes no total, para testar todas as combinações possíveis.

Este teste detecta um tipo de erro muito comum que é a "contaminação" de endereços. Isso ocorre quando, por um defeito de fabricação, o isolamento elétrico entre duas ou mais células de memória fica muito fino, permitindo que os elétrons saltem de uma para outra. Isso faz com que ao gravar um dos bits o outro também seja gravado com o mesmo valor. Esse tipo de problema pode ser bastante intermitente, acontecendo apenas quando o segundo bit estiver com um valor zero, ou apenas esporadicamente, daí a necessidade de tantas repetições.

Teste 7 (Random number sequence): Para eliminar qualquer dúvida, são agora escritas sequências de números randômicos, preenchendo todos os endereços da memória. Os números são conferidos em pequenos blocos e o teste é repetido diversas vezes.

Teste 8 (Modulo 20, ones&zeros): Este teste é basicamente uma repetição do teste 7, mas agora utilizando um algoritmo diferente, chamado "Modulo-X", que elimina a possibilidade de qualquer defeito ter passado despercebido pelos testes anteriores por ter sido mascarado pelos caches L1 e L2 ou mesmo pelos registradores do processador. Note que tanto o teste 7 quanto o 8 são bastante demorados e servem apenas para detectar erros extremamente raros, eliminando qualquer dúvida sobre a saúde dos módulos. Eles foram criados realmente como um exercício de perfeccionismo.

Teste 9 (Bit fade test, 90 min, 2 patterns): Este é um teste final, que permite detectar erros

raros relacionados com os circuitos de refresh, ou soft-erros causados por fatores diversos, que alterem os dados armazenados.

No teste, todos os endereços são preenchidos usando uma sequência de valores predefinidos. O programa aguarda 90 minutos e verifica os dados gravados, anteriormente. Esses dados são produzidos usando sequências matemáticas, de forma que o programa só precisa repetir as mesmas operações na hora de verificar, sem precisar guardar uma cópia de tudo que gravou em algum lugar. Em seguida, a mesma sequência é gravada novamente, mas desta vez com os dígitos invertidos (o que era 1 vira 0 e o que era 0 vira 1). O programa aguarda mais 90 minutos e checka novamente. Este teste demora mais de três horas, por isso não é executado automaticamente junto com os outros 8. Pense nele como um último refúgio para os paranóicos. Para executá-lo, pressione "C", depois "1" (Test Selection) e em seguida "3" (Select Test). Na opção "Test Number [1-9]" pressione "9" e em seguida "0" (Continue).

Ao detectar um erro, a primeira providência é trocar o módulo de memória e refazer o teste. Em alguns casos o problema pode não ser no módulo, mas sim na placa-mãe. Em alguns casos mais raros pode ser até mesmo que tanto o módulo quanto a placa estejam bons, e o problema seja apenas algum tipo de incompatibilidade entre eles. Eu observei isso, por exemplo, em uma Tyan Tiger MPX que testei em 2001. A placa só ficou completamente estável com o terceiro módulo de memória que testei, sendo que os dois primeiros não tinham defeitos e passaram no teste do memtest86 depois de instalados em outros micros.

Um dos motivos era que essa placa utilizava um valor CAS de 2.5 ciclos (valor fixo), uma configuração incomum para a época. Assim como ela, muitas outras placas utilizam configurações incomuns, que podem causar incompatibilidades com memórias de algumas marcas. Jamais jogue fora um módulo com erros antes de testá-lo em outra placa-mãe diferente, pois ele pode estar bom.

Experimente também baixar a frequência de operação da memória, ou do FSB para testar o módulo em frequências mais baixas que as nominais. Muitas vezes um módulo danificado por estática ou por variações nas tensões fornecidas pela fonte, pode deixar de funcionar estavelmente na frequência máxima, mas continuar suportando frequências mais baixas.

Outra dica é limpar cuidadosamente o módulo, removendo a poeira acumulada e limpando os contatos usando uma borracha de vinil (as borrachas de escola, do tipo que não esfarela). Na falta de uma, você também pode usar uma cédula em bom estado. O papel moeda é abrasivo e realmente limpa os contatos na medida certa.

Aqui temos um módulo de 512 MB danificado por estática, flagrado pelo teste. Veja que foram identificados diversos endereços defeituosos. A lista mostra apenas os primeiros erros, mas pressionando a tecla "C" e depois "4" (error summary), é possível ver o número total. Nesse caso, o

MEMÓRIAS

módulo tinha nada menos do que 222 endereços defeituosos. Além de identificar os erros, o memtest mostra a partir de qual MB do módulo eles começam. Pelo screenshot, você pode ver que eles começam a partir do 433° MB:

Memtest-86 v3.2 | Pass 5X #
AMD Duron 1100 Mhz | Test 67x #####
L1 Cache: 128 6748MB/s | Test #3 (Moving inversions, 8 bit pattern)
L2 Cache: 64K 2988MB/s | Testing: 108K - 480M 480M
Memory : 480M 417MB/s | Pattern: 04040404
Chipset : VIA KM266

WallTime	Cached	RsvdMem	MemMap	Cache	ECC	Test	Pass	Errors	ECC Errs
0:02:19	480M	128K	e820-Std	on	off	Std	0	114	0

Ist	Pass	Failing Address	Good	Bad	Err-Bits	Count	Chan
3	0	0001b19df08 - 433.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b1adf08 - 433.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b25df08 - 434.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b2bdf08 - 434.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b2ddf08 - 434.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b36df08 - 435.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b3adf08 - 435.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b7fdf08 - 439.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b8adf08 - 440.8MB	f7f7f7f7	f7b7f7f7	00400000	1	
3	0	0001b9edf08 - 441.8MB	f7f7f7f7	f7b7f7f7	00400000	1	

(ESC)exit (c)configuration (SP)scroll_lock (CR)scroll_unlock

No caso de módulos em que os erros aparecem logo nos primeiros endereços, não existe o que fazer, pois eles farão com que o sistema trave logo no início do boot. Módulos onde os defeitos se concentram no final (como esse do exemplo) ainda podem ser usados para testes, pois o sistema acessa sempre a memória a partir do começo, deixando os últimos endereços do módulo por último. Enquanto você estiver rodando aplicativos leves e o sistema não acessar os endereços defeituosos do módulo, tudo funciona normalmente.

No Linux é possível orientar o sistema a utilizar apenas o começo do módulo, ignorando os endereços a partir da onde começam os erros. Isso é feito passando a opção "mem=384M" (onde o "384" é a quantidade de memória que deve ser usada) para o Kernel na tela de boot. Isso varia um pouco de distribuição para distribuição. No Ubuntu, por exemplo, é preciso pressionar a tecla "F6" e em seguida adicionar o "mem=384M" (sem mexer nas demais opções da linha):



No caso do Windows XP, é possível usar a opção `"/maxmem=`. Adicione a linha no arquivo `"boot.ini"`, especificando a quantidade de memória que deve ser utilizada (em MB), como em `"/maxmem=384"`. Esta alteração pode ser feita também através do `msconfig`, através da aba `"Boot.ini > Opções Avançadas"`.

Depois de concluído o boot, você pode confirmar usando o comando `"free"`, que reporta o uso de memória. Você verá que independentemente da capacidade real do módulo, o sistema usa a memória apenas até o MB especificado na opção. Essa dica permite aproveitar a parte "boa" do módulo em algum micro usado para aplicações leves, ao invés de ter que jogá-lo fora. Não se esqueça de etiquetar os módulos defeituosos (ou o PC onde eles forem instalados), indicando a partir de qual MB foram identificados endereços defeituosos. Assim você evita ter de executar o teste novamente cada vez que precisar reinstalar o sistema, ou instalar os módulos em outro micro.

Além do `memtest86` e outros softwares, também existem testadores de memória dedicados, que executam uma sequência de testes automatizados, que, além de detectar defeitos, identificam as características do módulo, como a frequência suportada, CAS latency e assim por diante. Estes testadores são caros e, para ser sincero, o teste não é tão confiável quanto a sequência do `memtest`. A vantagem é que eles permitem testar um grande número de módulos em pouco tempo, de forma prática, por isso podem ser interessantes para distribuidores e lojas.

Duas das maiores empresas nesse segmento são a <http://www.memorytest.com/> e a

<http://www.simmtester.com/>, onde você pode se informar sobre os preços e os modelos disponíveis.



Paridade, ECC e memórias registered

Por melhor que seja a qualidade, todos os tipos de memória são passíveis de erros, que podem ser causados por inúmeros fatores, desde variações na tensão da tomada que não são completamente absorvidos pela fonte de alimentação, estática, diversos tipos de interferências eletromagnéticas e, por incrível que possa parecer, até mesmo raios cósmicos (que num PC doméstico causam um soft-error a cada poucos meses), como você pode ver neste estudo da IBM: <http://www-1.ibm.com/servers/eserver/pseries/campaigns/chipkill.pdf>

Ao contrário dos "hard-errors", que são danos físicos nos módulos de memória, causados por eletricidade estática ou outros tipos de descargas, os soft-erros são erros momentâneos, onde um ou alguns poucos bits são alterados, sem que os chips de memória sejam danificados.

Eles podem causar os mais diversos efeitos colaterais, como travamentos de programas, pequenos danos em arquivos salvos e assim por diante. Num desktop eles não costumam ser catastróficos, mas podem causar efeitos sérios em sistemas que manipulam informações sensíveis, como no caso dos bancos, por exemplo, onde um soft-error poderia mudar o saldo da sua conta bancária.

Para aumentar o grau de confiabilidade dos sistemas, foram criados métodos de diagnóstico e correção de erros. Tudo começou com os sistemas de paridade, usados em muitos módulos de 30 e

MEMÓRIAS

72 vias.

A paridade é um método mais antigo, que somente é capaz de identificar alterações nos dados depositados nas memórias, sem condições de fazer qualquer tipo de correção. A paridade consiste na adição de mais um bit para cada byte de memória, que passa a ter 9 bits, tendo o último a função de diagnosticar alterações nos dados.

A operação de checagem dos dados na paridade é bem simples: são contados o número de bits "1" de cada byte. Se o número for par, o bit de paridade assume o valor "1" e caso seja ímpar, o 9º bit assume o valor "0". Quando requisitados pelo processador, os dados são checados pelo circuito de paridade que verifica se o número de bits "1" corresponde ao depositado no 9º bit.

Caso seja constatada alteração nos dados, ele envia ao processador uma mensagem de erro. Claro que esse método não é 100% eficaz, pois não é capaz de detectar a alteração de um número de bits que mantenha a paridade. Caso, por exemplo, dois bits zero retornassem alterados para bits um, o circuito de paridade não notaria a alteração nos dados. Felizmente, a possibilidade da alteração de dois ou mais bits ao mesmo tempo é remota.

Exemplo de Byte de dados	Número de Bits "1" no Byte	Bit de paridade
00000000	0	1
10110011	5	0
00100100	2	1
11111111	8	1

O uso da paridade não torna o computador mais lento, pois os circuitos responsáveis pela checagem dos dados são independentes do restante do sistema. Seu único efeito colateral é o encarecimento dos módulos de memória, que em vez de 8 ou 16 chips, passam a ter 9 ou 18, tornando-se pelo menos 12% mais caros.

Além do aumento no custo, o grande problema da paridade é que ela apenas permite identificar erros, sem corrigi-los. Isso acaba fazendo com que a utilidade prática não seja tão grande. Conforme os módulos de memória foram tornando-se mais confiáveis, os módulos com paridade entraram em desuso.

Em seguida temos o **ECC**, o sistema atual, que permite não apenas identificar, mas também corrigir erros simples. O ECC acaba sendo a solução perfeita, pois permite que um servidor continue funcionando, sem interrupções e de forma confiável, mesmo com um grande número de soft-errors, causados por fatores diversos.

MEMÓRIAS

O número de bits necessários para implementar o ECC decresce conforme aumenta a largura do barramento usado pelo módulo. Em um módulo de 32 bits (como os antigos módulos de 72 vias), são necessários 7 bits adicionais para cada 32 bits de memória, mas nos módulos DIMM de 64 bits atuais, são necessários apenas 8 bits para cada 64 bits de memória, ou seja, o mesmo que seria necessário para usar paridade.

Os módulos DIMM com ECC são fáceis de identificar, pois eles possuem 5, 9 ou 18 chips, em vez de 4, 8 ou 16. O uso de ECC é mais comum em módulos registered, que são específicos para servidores, mas também é possível encontrar alguns módulos unbuffered com ECC:



Continuando, temos os módulos **registered DIMM**, que são também fonte de dúvidas frequentes. Os módulos de memória que usamos nos PCs domésticos são chamados de unbuffered.

Eles usam um layout simples e eficiente, onde o controlador de memória tem acesso direto aos chips de memória, garantindo tempos de latência mais baixos.

Essa simplicidade tem um custo, que é uma limitação no número de chips por módulo e também no número de módulos que podem ser instalados na mesma placa-mãe. Salvo raras exceções, os módulos unbuffered possuem no máximo 16 chips de memória e é possível projetar placas-mãe com suporte para até 4 módulos.

Isso não é um problema nos desktops, onde normalmente não precisamos de mais do que 2 ou 4 GB de RAM, mas é uma grave limitação nos servidores, onde é comum o uso de mais memória. Os módulos registered incluem chips adicionais (registradores) que funcionam como uma interface adicional entre o controlador e os chips. Eles permitem que o controlador suporte um número maior de módulos de memória e também que sejam usados módulos com mais chips, permitindo a instalação de quantidades muito maiores de memória. Muitas placas para servidores incluem 8 slots de memória, e existem módulos registered com 32 ou até mesmo 48 chips (sem contar os chips adicionais no caso dos módulos com ECC). É fácil reconhecer os módulos registered, devido à presença dos chips adicionais:



A desvantagem é que o uso dos registradores retarda a transmissão dos sinais, aumentando a latência e consequentemente reduzindo o desempenho dos módulos. A maioria das placas com suporte a módulos registered não suporta módulos unbuffered, de forma que seu uso não é uma opção. Também não é possível usar módulos registered, muito menos misturá-los com módulos unbuffered nas placas para desktop que não os suportam.

O suporte a módulos registered está disponível apenas em placas-mãe destinadas a servidores e workstations, onde a possibilidade de usar mais memória supera as desvantagens. É possível encontrar tanto módulos de memória SDRAM quanto módulos DDR e DDR2 em versão registered. Por utilizarem componentes adicionais e serem produzidos em pequena quantidade, eles normalmente custam o dobro do preço dos módulos unbuffered, de forma que você só deve considerar seu uso quando realmente necessário.

Referências

MORIMOTO, C. E. **Tudo sobre a memória RAM.** Parte 1, 2010. Disponível em:

<<http://www.hardware.com.br/tutoriais/memoria-ram/>>. Acesso em: 01/09/2016.

MORIMOTO, C. E. **Tudo sobre a memória RAM.** Parte 1, 2010. Disponível em:

<<http://www.hardware.com.br/tutoriais/memoria-ram2/>>. Acesso em: 01/09/2016.