Hardware II,
o guia definitivo

Carlos E. Morimoto

2010

SUL EDITORES

© Carlos Eduardo Morimoto, 2010

Capa
Luciano Lourenço

Finalização da capa
Vânia Möller

Projeto gráfico e editoração
Vânia Möller

Revisão
Vânia Möller

Editor:
Luis Gomes

Dados Internacionais de Catalogação na Publicação CIP
Bibliotecária Responsável: Denise Mari de Andrade Souza - CRB 10/960

M857h

Morimoto, Carlos Eduardo
Hardware II, o guia definitivo / Carlos Eduardo Morimoto. -- Porto Alegre : Sul Editores, 2010.
1086 p.

ISBN 978-85-99593-16-5

1. Hardware. 2. Microprocessadores. 3. Informática. I. Título

CDU:004.3

Todos os direitos reservados à
Editora Meridional Ltda.

Av. Osvaldo Aranha, 440 cj. 101
Cep: 90035-190 PortoAlegre-RS
Tel: (0xx51) 3311-4082
Fax: (0xx51) 3264-4194
www.editorasulina.com.br
e-mail: sulina@editorasulina.com.br

Junho/2010

Impresso no Brazil/Printed in Brazil

SUMÁRIO

Introdução 09
Processadores 11
Uma rápida árvore genealógica 12
Memória RAM 17
A evolução dos módulos de memória 21
Memória cache 26
HD 32
Swap e cache de disco 34
Placa de vídeo 36
A placa-mãe 41
Barramentos 42
Formatos de placas 45
Chipset 49
Fonte 53
Monitores 54
Placas de som 60
Mouses e teclados 66

Capítulo 1:
Um pouco de história 71
O ENIAC 75
O transistor 76
Os primeiros chips 78
O sistema binário 80
Hardware x software 82
Década de 70: a evolução dos computadores pessoais 82
O surgimento do PC 92
O 286 96
O 386 e a era dos 32 bits 99
O Lisa e o Macintosh 101
As primeiras versões do Windows 103
A lei de Moore 108
Como são fabricados os processadores 109
O processo de litografia 110
Máscaras e steppers 114
Encapsulamento e binning 117
Sistemas embarcados e FPGAs 119
Os supercomputadores 124

Capítulo 2:
Do 486 ao Pentium D 131
A era 486 132
O Pentium e a plataforma soquete 7 138
Chipsets para placas soquete 7 139
Pentium MMX e outros processadores soquete 7 144
Pentium Pro, Pentium II e Celeron 146
Pentium III 152
Chipsets e placas para o Pentium II e III 154
K6-2 e K6-3 161
As placas super 7 164
O Athlon 166
Arquitetura 166
Cache e o bus EV6 170
Athlon Thunderbird (soquete A) 171
Athlon XP (Palomino) 177
Athlon Thoroughbred 180
Athlon Barton e o Sempron 182
Chipsets para o Athlon, Duron e Sempron 184

VIA C3 e o Crusoe 196
A era Pentium 4 198
Willamette e as memórias Rambus 199
Entendendo a arquitetura NetBurst 202
Northwood 207
Prescott 209
A introdução do Hyper-Threading 212
Smithfield, Cedar Mill e Presler 214
O soquete LGA-775 217
Pentium D e o sistema de numeração 219
Chipsets para o Pentium 4 225
Chipsets para o Pentium D 233

Capítulo 3:
A era dos 64 bits 245
Itanium e o x86-64 246
A arquitetura K8 250
Os modelos 257
Athlon 64 e FX 258
Athlon 64 X2 261
Sempron 266
Reconhecendo os processadores 271
Chipsets para o Athlon 64, X2 e Sempron 276
A plataforma Core 286
Entendendo a arquitetura 291
Core 2 Duo E6xxx (Conroe) 293
Core 2 Quad Q6xxx (Kentsfield) 298
Core 2 Duo E4xxx e Pentium E2xxx (Allendale) 300
Penryn: a segunda geração 303
Os 45 nm 305
Core 2 Duo E8xxx e E7xxx (Wolfdale) 307
Core 2 Quad Q9xxx, Q8xxx e Q7xxx (Yorkfield) 308
Pentium E5xxx e E6xxx 309
Celeron 4xx, E1xxx e E3xxx 311
Chipsets para o Core 2 Duo, Quad e Celeron 315
Quad FX e o Skulltrail 330

Capítulo 4:
Processadores e chipsets atuais 335
O K10 (Barcelona) 335
Phenom 338
Phenom X3 340
O TLB bug e o problema do Cool'n'Quiet 342
Athlon X2 7xxx 345
Phenom II 345
As versões: X4, X3 e X2 349
Athlon II X2 350
Athlon II X4 352
O Sempron de 45 nm 354
Quatro cores pelo preço de três 355
Chipsets para o Phenom e Phenom II 357
Abrindo mão das fábricas 367
Intel Core i7, Core i5 e Core i3 370
Os caches 371
Controlador de memória 372
O X58 374
Apresentando o PQI 374
Loop Stream Detector e a volta do Hyper Threading 376
Gerenciamento de energia e o Turbo Boost 378
Os modelos baseados no Bloomfield 379
Lynnfield e o Core i5 381
As linhas PCI Express e o P55 385
Clarkdale e o vídeo integrado 388
Gulftown e a era dos 6 núcleos 392

Tuban: six-core da AMD 396
O fim dos chipsets, como os conhecemos 398
A era das APUs 403
Llano, Bulldozer e Bobcat 403
Um resumo sobre os soquetes e encaixes 407
A evolução dos coolers 410
Os Intel Macs 417
Os chips PowerPC e a migração 418
EFI, TPM e o OSX 421

Capítulo 5:
Padrões de memória RAM 423
Memórias Regulares 425
Memórias FPM 426
Memórias EDO 426
Memórias SDR-SDRAM 428
Memórias DDR 429
Memórias DDR2 434
Memórias DDR3 438
A maldição dos 32 bits: o limite de 3 GB 441
Identificando módulos de memória defeituosos 447
Paridade, ECC e memórias registered 452

Capítulo 6:
Placa-mãe e barramentos 455
Componentes da placa-mãe 455
BIOS 463
Os barramentos: ISA, EISA, VLB e PCI 467
Vida e morte do AGP 475
O PCI Express 480
Como o PCI Express funciona 482
Dentro do chipset 486
AMR, CNR, ACR e HDMR: os esquecidos 486
USB 488
USB 3.0 494
IEEE 1394 (Firewire) 498
Endereços de IRQ e DMA 500
APIC 506
DMA e I/O 507

Capítulo 7:
HDs e armazenamento 509
Como os HD funcionam 509
A placa controladora 516
Os discos 519
Correção de erros e badblocks 525
Desempenho 527
NCQ 534
Cache/Buffer 535
MTBF e service life 536
As interfaces 538
IDE 539
SATA 545
RAID 547
Os modos de operação 547
As controladoras 553
A transição para os setores de 4 kbytes 556
Sistemas de arquivos 558
FAT16 e FAT32 562
NTFS 566
EXT3 569
Recuperação de dados 571

S.M.A.R.T 572
Criando uma imagem binária 574
Reparando partições 575
Recuperando a MBR e tabela de partições 578
Recuperando arquivos apagados 583
Usando o Easy Recovery 583
Usando o Photorec 586
Gigabytes e gibibytes 590
Memória Flash 592
Os formatos de cartões 596
Estado sólido: Os SSDs 604
Os acidentes evolutivos 606
Desempenho: SSD "novo" e SSD "usado" 608
Ciclos de gravação e a questão da longevidade 611
A popularização 612
HDs híbridos, aceleradores e RAM-drives 614
Phase-change memory e o futuro 616
Mídias ópticas 619

Capítulo 8:
Fontes e energia 627
Como as fontes funcionam 628
Um pouco sobre eletricidade 632
A questão da eficiência 636
O 80 PLUS 638
Calculando o consumo 643
Distribuição da capacidade 648
Single +12V rail 649
DC-DC Converter 651
Outras especificações 651
Entendendo o PFC 656
Conectores de força, cabos e exaustores 659
Filtros de linha, nobreaks e dispositivos de proteção 667
Filtros de linha 668
Estabilizadores 672
Nobreaks (UPS) 675
Monitoramento e outras funções 680
Autonomia 682
Transmissão de energia sem fios 686
Baterias 689
Chumbo Ácido 690
Ni-Cd e Ni-MH 691
Li-ion e Li-poly 693
Reparo e reciclagem 696
Zinc-air 698
Células de combustível 700
Calculando a capacidade e autonomia 703

Capítulo 9:
Montagem, manutenção e configuração do setup 705
Evitando acidentes 705
Problemas com a fonte 706
Cooler e smoke test 710
Estática (ESD) 713
Componentes defeituosos 715
Dispositivos USB 716
Softwares 716
Montagem de micros 717
Preparando o terreno 718
Conectores do painel 720
Headers USB 722
Processador 723
Pasta térmica 727

Cooler 729
Memória 734
Instalando a placa-mãe 736
HDs e DVD 737
Finalizando a montagem 742
Solucionando problemas 745
Configuração do Setup 750
Opções do Setup 750
Discos e RAID 753
Opções de Boot 758
Frequências e tensões 760
Memória 763
Componentes integrados 767
Outras opções 772
Drivers e utilitários 778
Placa-mãe 779
Chipset e 3D 779
Som, modem e outros 780
Utilitários e benchmark 782

Capítulo 10:
Dicas de overclock 785
FSB e frequência da memória 790
Underclock e undervolt 795
Overclock no Core i7 800
A questão das tensões 803
Overclock no Phenom e Phenom II 807
Extreme overclocking 811
Overclock em placas de vídeo 815
BIOS overclock 822

Capítulo 11:
Vídeo e placas 3D 827
FPS, V-Sync, Triple Buffering e tearing 829
Recursos básicos 833
O chipset 833
Clock da GPU 834
Fill rate 835
Shaders e stream processors 836
TMUs e ROPs 839
Memória: DDR2, GDDR2, GDDR3 e GDDR5 840
Antialiasing e Anisotropic Filtering 845
Entendendo o SLI 849
CrossFire e CrossFire X 854
Hybrid CrossFire e Hybrid SLI 857
TurboCache e HyperMemory 859
As APIs: DirectX e OpenGL 861
O mundo da física: Physics, Physx e Havok 867
Os primórdios: placas 3D do século passado 870
Da GeForce à Kyro II: A era DirectX 7 875
GeForce 3, Radeon 8500 e a era DirectX 8 880
DirectX 9.0: o elo perdido 885
R300 e a Radeon 9700 885
R420 e a Radeon X800 887
NV30 e a GeForce FX 5xxx 892 890
DirectX 9.0c e o Shader Model 3.0 892
NV40 e a GeForce 6 893
G70 e a GeForce 7 896
R520 e a Radeon X1800 899
A era moderna (DirectX 10, 10.1 e 11) 902
O G80 e as GeForce 8xxx 903
O G92 907
A série 9000 909

O GT200 e a série 2xx 911
R600 e RV670 (séries HD 2xxx e HD 3xxx) 917
RV770 e a série HD48xx 919
RV740, RV730 e RV710 925
Cypress, Juniper e Redwood (série HD 58xx) 928
OpenCL, CUDA e Brook+ 933
Manutenção 936
PCs antigos para jogos? 937
Conectores: VGA x DVI 943
DisplayPort 948
Características dos Monitores LCD 950

Capítulo 12:
Notebooks, netbooks e portáteis 957
Categorias 960
Fabricantes e drivers 963
Processadores 965
Pentium M 966
Core Duo e Core 2 Duo 968
Celeron M 972
Processadores ULV e CULV 975
Core i7 e i5 mobile 978
Mobile Athlon 64 e Mobile Sempron 980
Turion 64 984
Turion X2 986
Via Nano 990
Placas 3D para notebooks 993
Placas da ATI 996
Placas da nVidia 1001
nVidia Optimus 1005
Placas 3D externas 1008
PC Card, Express Card, Mini-PCI e Express-Mini 1012
A saga dos netbooks 1015
O Atom 1022
Silverthorne x Diamondville 1028
O Pine Trail 1031
Athlon Neo 1033
Os Smartbooks 1034
Os chips ARM e o resto do mundo 1037
ARM7, ARM9 e ARM11 1039
Cortex A8 1040
Cortex A9 MPCore 1043
Cortex A5 1046
Sistemas e softwares 1047
A resposta da Intel 1047
Dicas de manutenção 1049
Desmontando 1050
Tela 1062
Teclado 1068
Localizando defeitos 1073

Referências 1086

Introdução:

Nos primórdios da informática, nas décadas de 50, 60 e 70, vários fabricantes disputavam o mercado. Cada um desenvolvia seus próprios computadores, que eram incompatíveis entre si, criando uma situação na qual tanto o hardware quanto os softwares de cada arquitetura não funcionavam nas outras. Isso causava uma ineficiência generalizada, pois cada fabricante tinha que desenvolver tudo, da placa-mãe ao sistema operacional.

No começo dos anos 80, os fabricantes começaram a se especializar. Surgiu então a plataforma PC, uma arquitetura aberta que permite o uso de periféricos de diversos fabricantes e de diferentes sistemas operacionais. Graças à abertura, você pode montar seus próprios micros (ou comprar PCs montados de vários fabricantes) e escolher qual sistema operacional usar, entre as diferentes versões do Windows, distribuições Linux, ou mesmo sistemas alternativos, como o Haiku ou o Syllable.

Tradicionalmente, a principal concorrente da plataforma era a Apple, que utilizava uma arquitetura fechada (baseada em processadores PowerPC), controlando a plataforma. Naturalmente, a maior parte da produção era terceirizada, com várias empresas desenvolvendo aplicativos e acessórios. Entretanto, a necessidade de manter tudo centralizado fazia com que a Apple precisasse desenvolver muita coisa por conta própria, incluindo o desenvolvimento de placas, gabinetes, fontes e até mesmo teclados, sem falar no sistema operacional e muitos dos aplicativos. Isso fazia com que o custo dos Macs acabasse sendo muito mais alto que o dos PCs.

Power Mac de 1999, baseado no PowerPC G4

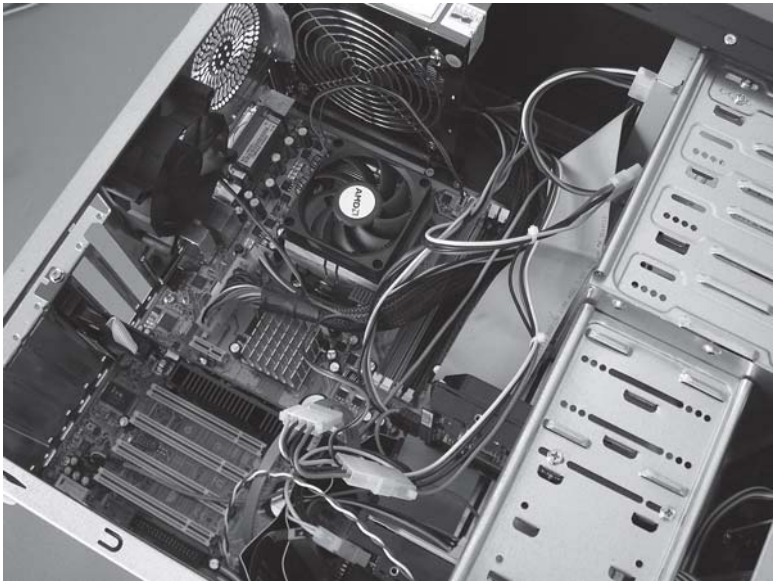
No início da década de 80, o mercado de micros PC era muito menor e muitos achavam que o modelo da Apple poderia prevalecer, mas não foi o que aconteceu. Dentro da história da informática temos inúmeros episódios que mostram que os padrões abertos quase sempre prevalecem. Um ambiente em que existam várias empresas concorrendo entre si favorece o desenvolvimento dos produtos, o que cria uma demanda maior e, graças à economia de escala, permite preços mais baixos.

No final, a Apple acabou sendo obrigada a dar o braço a torcer, adotando o uso de processadores Intel e mantendo apenas o desenvolvimento do sistema operacional e alguns componentes-chave. Hoje em dia, a única real diferença no hardware de um Mac e de um PC é o chip TPM, que é usado para dificultar o uso do Mac OS X em computadores que não são fabricados pela Apple.

Como os micros PC possuem uma arquitetura aberta, diversos fabricantes diferentes podem participar, desenvolvendo seus próprios componentes baseados em padrões já definidos. Temos então uma lista enorme de componentes compatíveis entre si, o que permite escolher as melhores opções entre diversas marcas e modelos.

Qualquer novo fabricante, com uma placa-mãe mais barata ou uma GPU mais rápida, por exemplo, pode entrar no mercado, é apenas uma questão de criar a demanda necessária. A concorrência faz com que os fabricantes sejam obrigados a trabalhar com uma margem de lucro relativamente baixa, ganhando com base no volume de peças vendidas, o que é muito bom para nós consumidores.

Começando do básico, qualquer PC é composto pelos mesmos componentes: placa-mãe, processador, cooler memória, HD, placa de vídeo, gabinete, fonte, monitor e outros periféricos (teclado, mouse, etc.):



Essa mesma divisão básica se aplica também a notebooks e netbooks (que tem termos de hardware estão cada vez mais parecidos com os desktops) e

também a outros aparelhos eletrônicos, como smartphones e tablets. A principal diferença é que neles os componentes são integrados numa única placa de circuito (muitas vezes no mesmo chip) e são utilizados chips de memória flash no lugar do HD.

Antigamente, a placa-mãe funcionava apenas como um ponto central, contendo os slots e barramentos usados pelos demais componentes. Além do processador e dos módulos de memória, era necessário comprar uma placa de vídeo, placa de som, modem, rede, etc. Cada componente era uma placa separada.

Com a integração dos componentes, a placa-mãe passou a incluir cada vez mais componentes, dando origem às placas "tudo onboard" que utilizamos atualmente (algumas placas mini-ITX destinadas a media-centers já vêm até com o processador e chips de memória!). Isso permitiu que os preços dos PCs caíssem assustadoramente, já que com menos componentes, o custo de fabricação é bem menor.

Embora componentes onboard (ou componentes integrados, que seria o termo mais correto) tenham uma certa má fama, eles são os grandes responsáveis pela queda de preço dos equipamentos em relação ao que tínhamos há uma ou duas décadas atrás. Se ainda utilizássemos placas separadas para cada componente (como na época do 486), os PCs não seriam apenas mais caros, mas também consumiriam mais energia e seriam mais propensos a problemas.

Para quem quer mais desempenho ou recursos, é sempre possível instalar placas adicionais, substituindo os componentes onboard. Um bom exemplo são as placas 3D dedicadas, que oferecem um desempenho brutalmente superior ao dos chipsets de vídeo integrados. Feitas as apresentações, vamos a uma rodada de detalhes básicos sobre os componentes:

Processadores

O processador é sempre o componente mais enfatizado em qualquer PC. Ao comprar um desktop ou notebook, quase sempre a primeira informação que se verifica é o modelo e/ou clock do processador. Além de ser o encarregado de processar a maior parte das informações, ele é o componente onde são usadas as tecnologias de fabricação mais recentes.

Existem no mundo apenas três empresas com tecnologia para fabricar processadores competitivos para micros PC: a Intel, a AMD e a VIA. Antigamente tínhamos outros fabricantes, como a IDT (que fabricou o IDT C6, concorrendo com o Pentium 1), a Texas Instruments (que fabricou chips 386 e 486), a Cyrix (que foi comprada pela VIA), a Transmeta (fabricante do Crusoé) e até mesmo a IBM. Entretanto, com o passar do tempo todas foram empurradas pra fora do mercado, deixando apenas a Intel e a AMD brigando pela supremacia e uma pequena VIA lutando para sobreviver.

Athlon X2 e Pentium D

Mais do que em qualquer outro componente, os preços dos processadores variam brutalmente de acordo com o modelo. Temos desde processadores de baixo custo, como os diferentes modelos do Sempron e do Celeron, que chegam a ser vendidos por menos de 40 dólares nos EUA, até processadores high-end, como os modelos mais caros do Core i7, que chegam a custar US$ 999.

O principal motivo de tamanha disparidade é a necessidade dos fabricantes de adaptarem seus produtos a diferentes faixas de mercado, que vão desde os PCs de baixo custo, que são vendidos por menos de 800 reais, até estações de trabalho ou PCs para jogos que chegam a custar mais de 5 mil. Muda o número de núcleos, a quantidade de cache e o clock dos processadores, mas a arquitetura usada continua quase sempre a mesma. Em muitos casos, os processadores de baixo custo são apenas versões castradas de chips mais rápidos, com parte dos componentes desativados, uma estratégia usada tanto pela Intel quanto pela AMD.

Com tantos processadores disponíveis no mercado, entender as diferenças entre os diferentes modelos dentro de cada geração pode parecer impossível, mas na verdade não é tão difícil assim, já que os modelos são ramificações de algumas poucas arquiteturas. Vamos começar com um resumo rápido:

Uma rápida árvore genealógica

Dentro do mundo PC, tudo começou com o 8088, lançado pela Intel em 1979 e usado no primeiro PC, lançado pela IBM em 1981. Depois veio o 286, lançado em 1982, e o 386, lançado em 1985.

O 386 pode ser considerado o primeiro processador moderno, pois foi o primeiro a incluir o conjunto de instruções x86 básico, usado até os dias de hoje. O 486 (que ainda faz parte das lembranças de muita gente que comprou seu primeiro computador durante a década de 90) foi lançado em 1989, mas ainda era comum encontrar micros baseados nele à venda até por volta de 1997.

Depois entramos na era atual, inaugurada pelo Pentium, que foi lançado em 1993 mas demorou alguns anos para se popularizar e substituir os 486. Em 1996 foi lançado o Pentium MMX, que deu um último fôlego à plataforma. Depois, em 1997, veio o Pentium II, que usava um encaixe diferente e por isso era incompatível com as placas-mãe antigas. A AMD soube aproveitar a oportunidade, desenvolvendo o K6-2, um chip com uma arquitetura similar ao Pentium II, mas que era compatível com as placas soquete 7 antigas.

A partir daí as coisas passaram a acontecer mais rápido. Em 1999 foi lançado o Pentium III e em 2000 o Pentium 4, que trouxe uma arquitetura bem diferente dos chips anteriores, otimizada para permitir o lançamento de processadores que trabalham a frequências mais altas.

Pentium 4 em versão soquete 478

O último Pentium III trabalhava a 1.0 GHz, enquanto o Pentium 4 atingiu rapidamente os 2.0 GHz, 3 GHz e em seguida 3.5 GHz. O problema é que o Pentium 4 possuía um desempenho por ciclo de clock inferior a outros processadores, o que fazia com que a alta frequência de operação servisse apenas para equilibrar as coisas. A primeira versão do Pentium 4 operava a 1.3 GHz e, mesmo assim, perdia para o Pentium III de 1.0 GHz em diversas aplicações.

Quanto mais alta a frequência do processador, mais energia ele consome e, consequentemente, mais calor é dissipado por ele (como diz a primeira lei da termodinâmica, "nada se perde, tudo se transforma"). O calor não era um prolema na época do Pentium 1, quando os processadores usavam apenas 10 ou 15 watts, mas é um dos grandes limitantes hoje em dia, quando muitos processadores rompem a marca dos 150 watts.

Não é incomum que processadores domésticos sejam capazes de operar ao dobro da frequência nominal quando refrigerados com nitrogênio ou hélio líquido (o recorde para o Phenom II de 45 nm, por exemplo, é de 6.5 GHz), que eliminam o problema da temperatura (veja mais detalhes no tópico sobre Extreme Overclocking do capítulo de overclock). Entretanto, ao usar um cooler regular, a temperatura se torna um limitante muito antes.

Quando as possibilidades de aumento de clock do Pentium 4 se esgotaram, a Intel lançou o Pentium D, uma versão dual-core do Pentium 4. Inicialmente os Pentium D eram caros, mas com o lançamento do Core 2 Duo eles caíram de preço e passaram a ser usados até mesmo em micros de baixo custo. Os Pentium D eram vendidos sob um sistema de numeração e não sob a frequência real de clock. O Pentium D 820, por exemplo, opera a 2.8 GHz, enquanto o 840 opera a 3.2 GHz.

Em 2003 a Intel lançou o Pentium M, um chip derivado da antiga arquitetura do Pentium III, que consome pouca energia, esquenta pouco e mesmo assim oferece um excelente desempenho. Um Pentium M de 1.4 GHz chegava a superar um Pentium 4 de 2.6 GHz em diversas aplicações.

O Pentium M foi desenvolvido originalmente para ser usado em notebooks, mas se mostrou tão eficiente que acabou sendo usado como base para o desenvolvimento da plataforma Core, usada nos processadores Core 2 Duo e Core 2 Quad. O Pentium 4 acabou se revelando um beco sem saída, descontinuado e condenado ao esquecimento.

Paralelamente a todos esses processadores, temos o Celeron, uma versão mais barata, mas com um desempenho um pouco inferior, por ter menos cache ou outras limitações. Na verdade, o Celeron não é uma família separada de chips, mas apenas um nome comercial usado nas versões mais baratas (com metade ou um quarto do cache) de vários processadores Intel. Existem Celerons baseados no Pentium II, Pentium III, Pentium 4, Pentium M e até mesmo o Celeron E1xx, que é uma versão com menos cache do Core 2 Duo.

Para efeito de comparação, entre os chips antigos e os atuais, um 486 tinha cerca de 1.2 milhões de transistores e chegou a 133 MHz, enquanto o Pentium MMX tinha 4.3 milhões e chegou a 233 MHz. Um Pentium 4 (Prescott) tem 125 milhões e chegou aos 3.8 GHz, que foi por muitos anos a frequência de clock mais alta usada por um processador x86.

O transistor é a unidade básica do processador, capaz de processar um bit de cada vez. Mais transistores permitem que o processador processe mais instruções de cada vez enquanto a frequência de operação determina quantos ciclos de processamento são executados por segundo. O uso de mais transistores permite que o processador inclua mais componentes (mais núcleos, unidades de execução, cache, etc.) e execute mais processamento por ciclo, enquanto a frequência de operação determina quantos ciclos de processamento são executados por segundo. É possível melhorar o desempenho dos processadores tanto aumentando o número de transistores quanto aumentando a frequência, mas como ambas as abordagens possuem seus limites, os fabricantes são obrigados a encontrar a melhor combinação entre as duas coisas.

Continuando, temos os processadores da AMD. Ela começou produzindo clones dos processadores 386 e 486, muito similares aos da Intel, porém mais baratos. Quando a Intel lançou o Pentium, que exigia o uso de novas placas, a AMD lançou o “5x86”, um 486 de 133 MHz, que foi bastante popular, servindo como uma opção barata de upgrade.

Embora o “5x86” e o clock de 133 MHz dessem a entender que se tratava de um processador com um desempenho similar a um Pentium 133, o desempenho era muito inferior, mal concorrendo com um Pentium 66. Este foi o primeiro de uma série de exemplos, tanto do lado da AMD, quanto do lado da Intel, em que existiu uma diferença gritante entre o desempenho de dois processadores do mesmo clock. Embora seja um item importante, a frequência de operação não é um indicador direto do desempenho do processador.

Uma analogia poderia ser feita em relação aos motores de carro. Os motores de 1.6 litros do final da década de 70, usados nas Brasílias e nos Fuscas, tinham 44 cavalos de potência, enquanto os motores 1.0 atuais chegam a mais de 70 cavalos. Além da capacidade cúbica, existem muitos outros fatores, como a eficiência do sistema de injeção de ar e combustível, taxa de

compressão, refrigeração, etc. o mesmo se aplica aos processadores, onde temos o número de unidades de execução, o índice de acerto do circuito de branch prediction, o tamanho e a eficiência dos caches e assim por diante.

Depois do 5x68 a AMD lançou o K5, um processador tecnicamente superior ao Pentium, mas que era caro e não era capaz de atingir frequências de operação competitivas. Ele foi seguido pelo K6 e mais tarde pelo K6-2, que fez muito sucesso, servindo como uma opção de processador de baixo custo e, ao mesmo tempo, como uma opção de upgrade para quem tinha um Pentium ou Pentium MMX.

Esta era do K6-2 foi uma época sombria da informática, não pelo processador em si (que excluindo o desempenho em jogos, tinha um bom custo-benefício), mas pelas placas-mãe baratas que inundaram o mercado. Aproveitando o preço acessível do processador, os fabricantes passaram a desenvolver placas cada vez mais baratas (e de qualidade cada vez pior) para vender mais, oferecendo PCs de baixo custo. A época foi marcada por aberrações. Um certo fabricante chegou a lançar uma família de placas sem cache L2, cujos capacitores estufavam, levando ao fim da vida útil da placa, depois de apenas um ano de uso.

As coisas voltaram aos trilhos com o Athlon, que se tornou a arquitetura de maior sucesso da AMD. A primeira versão usava um formato de cartucho (slot A) similar ao Pentium II, mas incompatível com as placas para ele. Ele foi sucedido pelo Athlon Thunderbird, que passou a usar o formato de soquete utilizado (com atualizações) até os dias de hoje.

Athlon XP, para placas soquete A

Competindo com o Celeron, a AMD produziu o Duron, um processador de baixo custo, idêntico ao Athlon, mas com menos cache. Em 2005 o Athlon foi descontinuado e o cargo foi herdado pelo Sempron, uma versão aperfeiçoada do Duron (com mais cache e capaz de atingir frequências mais altas), que passou a ser vendido segundo um índice de desempenho (que comparava o desempenho com o do Pentium 4) e não mais segundo o clock real.

Por volta de 2000, surgiram as primeiras notícias do “SledgeHammer”, um processador de 64 bits, que foi finalmente lançado em versão doméstica na forma do Athlon 64, que passou a ser o topo de linha da AMD. Apesar das mudanças internas, o Athlon 64 continuou sendo compatível com os

programas de 32 bits, da mesma forma que os processadores atuais são capazes de rodar softwares da época do 386, muito embora tenham incorporado diversos novos recursos.

Na prática, o fato de ser um processador de 64 bits não tornou o Athlon 64 muito mais rápido, mesmo em aplicativos otimizados (os ganhos de desempenho surgem mais devido ao controlador de memória integrado e aos novos registradores). A principal vantagem dos processadores de 64 bits foi derrubar uma limitação inerente a todos os processadores de 32 bits, que são capazes de acessar apenas 4 GB de memória RAM, um limite que está se tornando uma limitação grave em cada vez mais situações.

O Athlon 64 deu origem ao Athlon X2, o primeiro processador dual-core da AMD, onde temos dois processadores Athlon 64 no mesmo encapsulamento (dividindo a carga de processamento) e também as várias versões do Sempron para placas soquete 754, AM2 e AM3, que se tornaram bastante populares em PCs de baixo custo.

Com o avanço do Core 2 Duo, a AMD se apressou em atualizar a arquitetura do Athlon 64, incluindo algumas melhorias na arquitetura e (mais importante) suporte ao uso de 4 núcleos e cache L3 compartilhado. Surgiu então o Phenom, que foi o primeiro processador quad-core doméstico da AMD.

O Phenom foi produzido em diversas variações, incluindo versões com três núcleos (o Phenom X3) e também versões com parte do cache desativado. Elas surgiram da necessidade de maximizar o volume de produção, transformando os processadores com defeitos localizados em versões de baixo custo. Essa mesma filosofia deu origem também à série Athlon X2 7xxx, que consistiu em versões low-end (baratas, porém com desempenho mais baixo) do Phenom, com dois dos núcleos desativados.

As versões iniciais do Phenom foram produzidas usando uma técnica de 65 nanômetros (similar à usada pela Intel para produzir a versão inicial do Core 2 Duo), o que limitou o tamanho do cache L3 a apenas 2 MB. Em 2009 a AMD migrou suas fábricas para a técnica de 45 nm, dando origem ao Phenom II que trouxe 6 MB de cache L3, resultando em um ganho considerável de desempenho.

Assim como no caso do Phenom original, o Phenom II é vendido em várias versões, com 4 núcleos, 3 núcleos e até mesmo dois núcleos ativos, com o objetivo de aproveitar as unidades com pequenos defeitos.

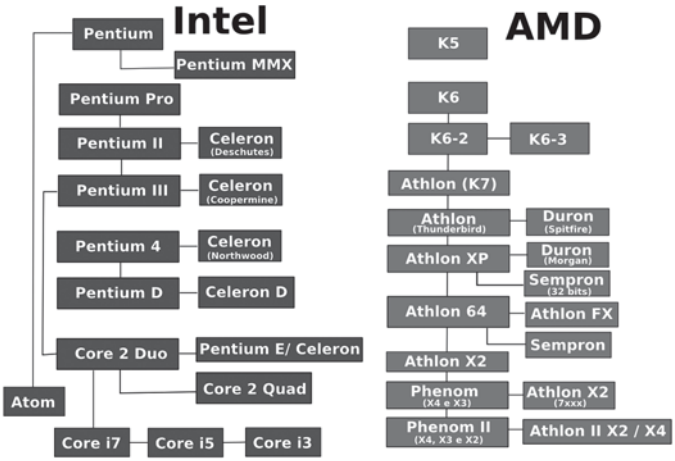
A nova técnica de fabricação deu origem também a duas novas séries de processadores de baixo custo, o Athlon II X2 e o Athlon II X4, destinadas a substituir os últimos modelos do Athlon X2. Como os nomes sugerem, o Athlon II X2 possui dois núcleos e o Athlon II X4 possui quatro. A arquitetura continua sendo a mesma do Phenom II, mas eles excluem o cache L3, removido para cortar custos.

A última rodada foi iniciada com o lançamento do Core i7 e do Core i5, que inauguraram a nova geração de processadores da Intel, realimentando o ciclo de lançamentos. Uma das mudanças introduzidas por eles foi o Turbo Boost, um sistema de gerenciamento de clock que aumenta a frequência do processador quando apenas alguns dos núcleos está sendo utilizado, funcionando como uma espécie de sistema de overclock automático. O Turbo

Boost tornou o desempenho dos processadores muito mais variável, já que passou a depender também da temperatura de operação do processador e outros fatores.

Longe do mercado de chips de alto desempenho, temos também o Intel Atom, que serve como uma opção de processador barato e de baixo consumo, destinado a netbooks e desktops ultra-compactos. Diferente de outros processadores modernos, o Atom é um descendente do Pentium 1, que utiliza uma arquitetura muito mais simples, baseada no processamento sequencial de instruções.

A seguir temos uma tabela rápida de referência, com os principais processadores da Intel e os equivalentes da AMD, que estudaremos ao longo do livro:



Memória RAM

Depois do processador, temos a memória RAM, que funciona como uma espécie de mesa de trabalho, armazenando arquivos e aplicativos em uso. A quantidade de memória RAM disponível tem um grande efeito sobre o desempenho, já que sem memória RAM suficiente o sistema é obrigado a usar memória swap, que é muito mais lenta. Na maioria das situações, ter uma quantidade suficiente de memória RAM instalada é mais importante que o desempenho do processador.

A memória RAM é um componente essencial não apenas nos PCs, mas em qualquer tipo de computador. Por mais que se tenha espaço disponível para armazenamento, na forma de HDs ou memória Flash, é sempre necessário ter uma certa quantidade de memória RAM; e, naturalmente, quanto mais melhor.

Graças ao uso da memória swap, é possível rodar a maioria dos sistemas operacionais modernos com quantidades relativamente pequenas de memória. No caso do Linux, é possível inicializar uma instalação enxuta (em modo texto, com pouca coisa além do kernel e o interpretador de comandos) com apenas 4 MB de memória. O problema é que com pouca memória o

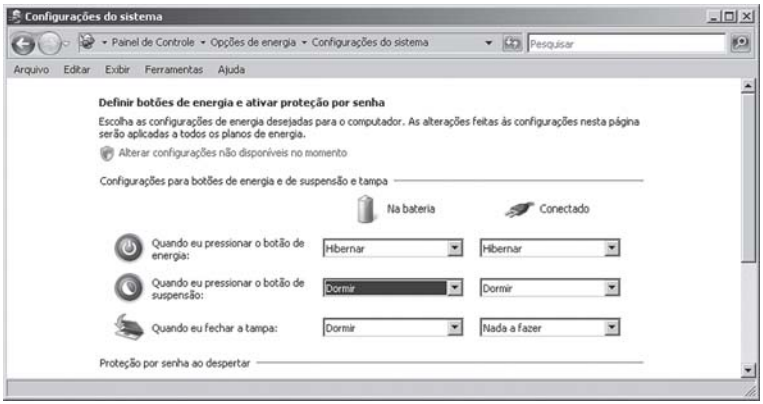
sistema fica extremamente lento (como qualquer um que já tentou rodar o Windows 7 ou uma distribuição Linux recente em um PC com menos de 256 MB de memória pode atestar... :)

A sigla "RAM" vem de "Random Access Memory", ou "memória de acesso aleatório", indicando a principal característica da memória RAM, que é o fato de permitir o acesso direto a qualquer um dos endereços disponíveis, de forma bastante rápida.

Ao carregar um programa, ele é lido no HD (ou outra mídia de armazenamento) e é transferido para a memória RAM, para só então ser executado pelo processador. A memória RAM oferece tempos de acesso brutalmente mais baixos que o HD e trabalha com taxas de transferência muito mais altas, mas possui a desvantagem de perder os dados armazenados quando o micro é desligado; daí a necessidade de salvar os arquivos periodicamente.

É também por causa disso que o processo de boot é refeito cada vez que você liga o micro. Durante o boot, o sistema operacional, drivers, bibliotecas e aplicativos são novamente copiados para a memória, junto com suas configurações e preferências.

A única forma de evitar repetir o demorado processo de boot é manter a memória RAM ativa, ou salvar seu conteúdo no HD, recuperando-o no próximo boot. Essas são as estratégias usadas pelas opções de suspender e hibernar, disponíveis tanto no Windows quanto na maioria das distribuições Linux.



Ao suspender (dormir), a maioria dos componentes do sistema são desligados, incluindo o HD, a placa de vídeo e a maior parte dos componentes da placa-mãe. Mesmo o processador entra em um estágio de baixo consumo, onde a maior parte dos componentes internos são desativados e o clock é reduzido. Praticamente, os únicos componentes que continuam realmente ativos são os módulos de memória. Graças a isso o PC acaba consumindo (geralmente) menos de 20 watts de energia e pode voltar ao estágio original muito rapidamente.

Ao hibernar, o conteúdo da memória RAM é copiado para uma área reservada do HD e o micro é desligado. Ao ligar novamente, o conteúdo da memória é restaurado e temos o sistema de volta, sem precisar passar pelo processo normal de boot. O problema da hibernação é que a restauração demora muito

mais tempo, já que é necessário ler 512 MB, 1 GB ou mesmo 4 GB de dados (equivalentes à quantidade de memória RAM instalada) a partir do HD, o que muitas vezes demora mais do que um boot completo.

Além dos diferentes tipos de memória RAM, existem também outras tecnologias de memórias de acesso aleatório, como as SRAM e, mais recentemente, as Phase-change. Temos ainda as onipresentes memórias Flash (que veremos em detalhes no capítulo sobre HDs), que concorrem com os HDs como mídia de armazenamento.

O tipo mais comum de memória RAM, aquela que compramos na forma de módulos e instalamos na placa-mãe, é chamada de **DRAM**, ou "dynamic RAM". A memória DRAM passou a ser usada apenas a partir do final da década de 70, substituindo os chips de memória SRAM, que eram muito mais caros. Com o passar do tempo, as memórias DRAM viraram o padrão, de forma que geralmente dizemos apenas "memória RAM" e não "memória DRAM".

Dentro de um chip de memória DRAM, cada bit é formado pelo conjunto de um transistor e um capacitor. O transistor controla a passagem da corrente elétrica, enquanto o capacitor a armazena por um curto período. Quando o capacitor contém um impulso elétrico, temos um bit 1 e quando ele está descarregado temos um bit 0.

Quando falo em "capacitor", tenha em mente que não estamos falando em nada similar aos capacitores eletrolíticos da placa-mãe. Os "capacitores" usados nos chips de memória são extremamente pequenos e simples (basicamente dois pequenos blocos de metal ligados ao transistor), que conservam o impulso elétrico por apenas uma fração de segundo.

Para evitar a perda dos dados, a placa-mãe inclui um circuito de refresh, que é responsável por regravar o conteúdo da memória várias vezes por segundo (a cada 64 milissegundos ou menos), algo similar ao que temos em uma TV ou monitor CRT, onde o canhão de elétrons do monitor precisa atualizar a imagem várias vezes por segundo para evitar que as células de fósforo percam seu brilho.

O processo de refresh atrapalha duplamente, pois consome energia (que acaba sendo transformada em calor, contribuindo para o aquecimento do micro) e torna o acesso à memória mais lento. Apesar disso, não existe muito o que fazer, pois a única solução seria passar a usar memória SRAM, que é absurdamente mais cara.

Mesmo utilizando um único transistor por bit, os módulos de memória RAM são formados por um número assustador deles, muito mais que os processadores e outros componentes. Um módulo de memória de 4 GB, por exemplo, é formado geralmente por 16 chips de 2 gigabits cada um (2 gigabits = 256 megabytes). Com isso, cada chip possui mais de 2 bilhões de pares de transistores e capacitores e o módulo inteiro acumula mais de 32 bilhões de conjuntos.

Apesar dessa brutal quantidade de transistores, os chips de memória são relativamente simples de se produzir, já que basta repetir a mesma estrutura indefinidamente. É muito diferente de um processador, que além de ser muito mais complexo, precisa ser capaz de operar a frequências muito mais altas.

Com a evolução das técnicas de fabricação, os módulos de memória foram ficando cada vez mais baratos com o passar das décadas. Na época dos micros 486, chegava-se a pagar 40 dólares por megabyte de memória, valor que hoje em dia compra um módulo de 1 GB ou mais.

O problema é que os requisitos dos sistemas operacionais e aplicativos também aumentaram, quase que na mesma proporção. Enquanto o MS-DOS rodava bem com 2 ou 4 MB de memória, o Windows 95 já precisava de pelo menos 16 MB. O Windows XP (assim como a maioria das distribuições Linux atuais) não roda bem com menos de 512 MB, enquanto no Windows 7 o ideal é usar 1 GB ou mais.

Naturalmente, estes são apenas os requisitos básicos para rodar aplicativos leves. Muitos jogos e aplicativos pesados precisam de 2 GB de memória para rodar bem e o hábito de manter vários aplicativos abertos simultaneamente faz com que o valor suba rapidamente.

Salvo raras exceções, a capacidade dos módulos de memória é duplicada regularmente, conforme os fabricantes introduzem novas técnicas de fabricação. Por bizarro que possa parecer, os primeiros chips de memória (produzidos no final da década de 60) tinham apenas 64 bits (ou seja, meros 8 bytes), que com o tempo se transformaram em 128 bits, 256, 512, 1024 e assim por diante, até chegar aos chips de 2 gigabits ou mais usados atualmente.

Conforme módulos de maior capacidade são introduzidos, os módulos de capacidade mais baixa são retirados de produção, já que não faz sentido manter a produção de chips de baixa capacidade, cujo custo por megabyte é mais alto. É por isso que módulos de memória de tecnologias obsoletas (como os módulos de memória SDR usados até poucos anos atrás) são tão caros e difíceis de achar.

Os micros 486 usavam memórias FPM, que foram substituídas pelas memórias EDO na época do Pentium. O lançamento do Pentium II e do K6-2 marcou a migração para as memórias SDRAM, que podem ser consideradas o início da era moderna.

Existem 4 tipos de memória SDRAM: as SDR (uma transferência por ciclo), as DDR (duas transferências por ciclo), as DDR2 (4 transferências por ciclo) e as DDR3 (8 transferências por ciclo).

Os módulos de memória SDR são usados em micros antigos: Pentium II, Pentium III e os primeiros Athlons e Durons soquete A. As memórias DDR passaram a ser usadas a partir do Athlon Thunderbird e do Pentium 4, enquanto as DDR2 foram adotadas a partir das placas soquete 775 (no caso da Intel) e a partir dos modelos Athlon 64 para placas AM2, no caso da AMD.

Mais recentemente, temos assistido a uma nova migração, com a introdução dos módulos de memória DDR3, que foram adotados a partir dos processadores Core i7 e i5 no caso da Intel e a partir dos Phenom II destinados a placas AM3, no caso da AMD.

Infelizmente, cada nova tecnologia de memória é incompatível com a antiga, o que elimina a possibilidade de aproveitar os módulos antigos ao fazer upgrade. Entretanto, a queda nos preços das memórias tem reduzido esse problema. Antigamente se gastava 500 reais ou mais para ter uma quantidade

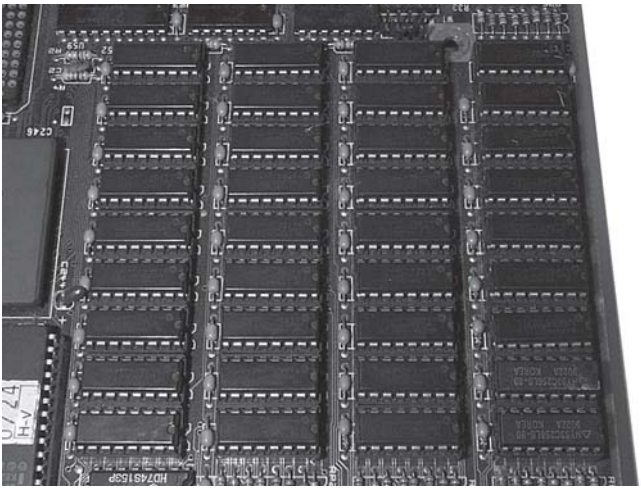
aceitável de memória no PC, enquanto hoje em dia pode-se passar muito bem com apenas um ou dois módulos de 80 reais cada.

Algumas placas (geralmente modelos de baixo custo) possuem dois tipos de soquete, permitindo usar módulos SDR e DDR, DDR e DDR2 ou DDR2 e DDR3 de acordo com a conveniência, mas sem misturar os dois tipos. Elas são relativamente comuns durante os períodos de transição, quando uma tecnologia de memória é substituída por outra e podem ser uma opção em alguns casos.

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.

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.

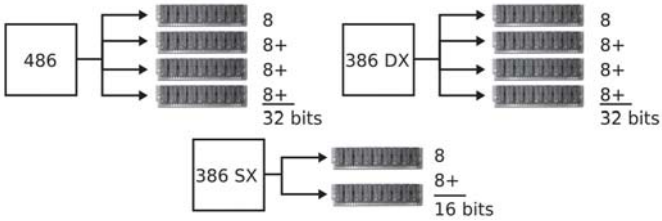
Módulo SIMM de 30 vias

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.

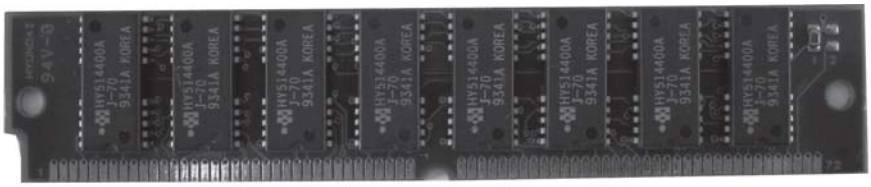
Os processadores 386 e 486 utilizavam um barramento de 32 bits para o acesso à memória, o que tornava necessário combinar 4 módulos de 30 vias para formar um banco de memória. Os 4 módulos eram então acessados pelo processador como se fossem um só. Era preciso usar os módulos em quartetos: 4 módulos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



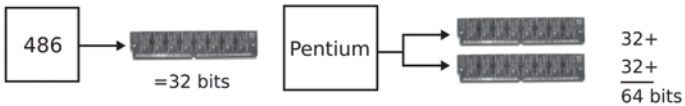
Apesar de serem muito mais práticos do que manipular diretamente os chips DIP, os módulos SIMM de 30 vias ainda eram bastante inconvenientes, já que era preciso usar 4 módulos idênticos para formar cada banco de memória. Eles foram desenvolvidos pensando mais na questão da simplicidade e economia de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



Módulo SIMM de 72 vias

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 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.

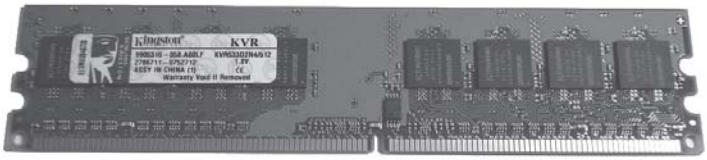
Existem três formatos de memória DIMM. Os mais antigos são os módulos de memória **SDR**, de 168 vias, que substituíram os antigos módulos de memória EDO, mas logo deram lugar às tecnologias mais recentes. Em seguida, temos os módulos de memória **DDR**, que possuem 184 contatos; os módulos **DDR2**,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

Módulo DIMM SDR (em cima) e módulo DDR

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:



Módulo DIMM DDR2

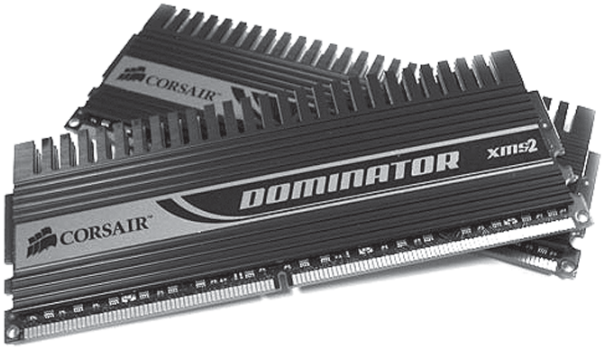
Isso é necessário, pois além das mudanças na forma de acesso, os módulos DDR2 utilizam tensão de 1.8V, enquanto os módulos DDR usam 2.5V. Se fosse possível instalar um módulo DDR2 em uma placa antiga, a maior tensão queimaria o módulo rapidamente.

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:

Módulo DDR3

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. 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.



Módulos DDR2 com dissipadores

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:

Chips BGA de memória

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

Módulo SODIMM DDR2

Memória cache

Apesar de toda a evolução, a memória RAM continua sendo muito mais lenta que o processador. O principal motivo disso é que a memória depende do processo de carga e descarga do capacitor onde é armazenado o impulso

elétrico, uma operação cuja velocidade está mais ligada às leis da física do que à técnica de fabricação.

Com o passar do tempo, diversos truques foram usados para aumentar a velocidade efetiva dos módulos de memória, incluindo o uso de múltiplas transferências por ciclo, pré-ativação de células que serão usadas nas leituras seguintes e assim por diante. Entretanto, apesar de todos os esforços, os processadores continuam a evoluir mais rápido e a diferença tende apenas a aumentar.

Se o desempenho do processador fosse atrelado ao desempenho da memória RAM, os PCs teriam estagnado na época do 486, já que não faria sentido desenvolver processadores mais rápidos apenas para que eles passassem a esperar mais e mais ciclos pelas leituras na memória. A solução veio com a introdução da memória cache, que serve como um reservatório temporário de dados com grande possibilidade de serem usados pelo processador, reduzindo a percentagem de vezes em que ele precisa buscar informações diretamente na memória.

Mesmo sendo muito pequeno em relação à memória, o cache acaba fazendo uma enorme diferença devido à maneira como os processadores trabalham. Diferente dos chipsets das placas 3D e de outros dispositivos que manipulam grandes volumes de dados, realizando operações relativamente simples, os processadores manipulam volumes de dados relativamente pequenos, executando operações complexas. Em resumo, o processador é como um matemático, que lê uma equação e fica algum tempo trabalhando nela antes de escrever o resultado. Com isso, mesmo um cache pequeno é capaz de melhorar o desempenho de maneira considerável.

Diferente de um simples buffer (como os usados em gravadores de CD para evitar que você perca a mídia por interrupções na gravação), onde os dados entram e saem na mesma ordem, o cache é um dispositivo bem mais inteligente. Além das células de memória, ele inclui um controlador que monitora o trabalho do processador, coletando blocos de informações que são frequentemente acessados e antecipando sempre que possível a leitura de dados que serão necessários nos ciclos seguintes.

Em um exemplo tosco, você pode imaginar uma lanchonete onde 10 dos lanches respondem por 90% dos pedidos. Em vez de esperarem que os clientes peçam, para só então começar a preparar os pedidos, os atendentes poderiam começar a preparar os lanches mais solicitados com antecedência (estilo McDonald's) para que os clientes recebam os pedidos mais rapidamente. Nesse caso, o tempo de preparo continua o mesmo, mas a espera para os clientes se torna muito menor.

A diferença fundamental entre a memória cache e a memória RAM é o tipo de célula usado. A memória cache é formada por células de memória SRAM, que são tipicamente formadas por conjuntos de 6 transistores, onde 4 deles formam a estrutura que mantém a carga e os outros dois controlam o acesso para leitura e gravação. Se você pudesse olhar um chip de memória SRAM com um microscópio de elétrons, veria uma estrutura similar a essa:

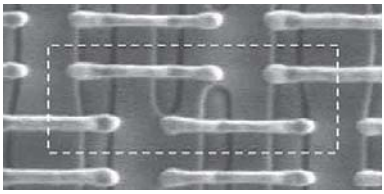
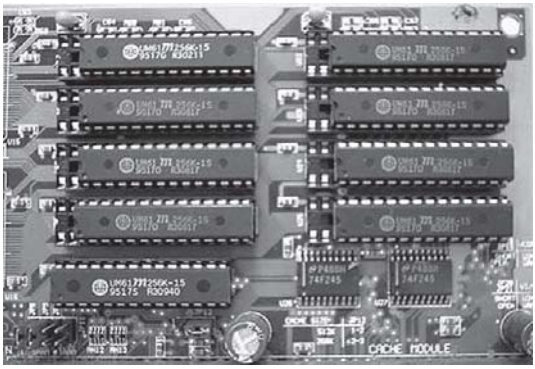


Ilustração da Intel mostrando uma célula de memória SRAM com 6 transistores

As células de memória SRAM são muito mais rápidas que as de memória RAM, mas são em compensação também muito mais caras, já que são necessários 6 transistores para cada bit de dados e mais um grande número de trilhas e circuitos adicionais. Em teoria, seria possível criar PCs que utilizassem apenas memória SRAM em vez de memória RAM, mas o custo seria proibitivo. Em vez disso, são usados pequenos blocos de cache, que graças a todas as otimizações acabam oferecendo 99% do ganho a 1% do custo.

O cache começou a ser usado na época do 386, onde ele era opcional e fazia parte da placa-mãe. Ao lançar o 486, a Intel integrou um cache de 8 KB diretamente ao processador, que embora muito pequeno, era extremamente rápido, já que operava na mesma frequência que ele e oferecia baixos tempos de latência. O cache incluído no processador passou então a ser chamado de cache L1 (nível 1) e o cache na placa-mãe passou a ser chamado de cache L2 (ou cache secundário).



Chips de cache L2 em uma placa soquete 5 antiga

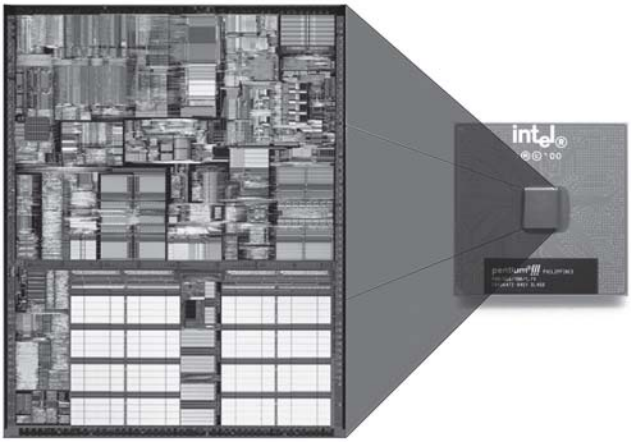
Sempre que precisa de novas informações, o processador checa primeiro as informações disponíveis no cache L1. Caso não encontre o que precisa, ele verifica em seguida o cache L2 e por último a memória. Sempre que o processador encontra o que precisa nos caches temos um “cache hit” e sempre que precisa recorrer à memória temos um “cache miss”. Quanto maior a percentagem de cache hits, melhor é o desempenho.

O cache na placa-mãe continuou a ser usado até a época das placas soquete 7, mas ele foi se tornando cada vez menos eficiente conforme os processadores passaram a usar multiplicadores de clock mais altos. O motivo é simples: instalado na placa-mãe, o cache L2 opera sempre na mesma frequência que ela (66 ou 100 MHz na época), enquanto o cache L1 opera na mesma frequência do processador.

Com a introdução das memórias SDRAM e mais tarde das DDR, a diferença de desempenho entre a memória e o cache passou a ser relativamente pequena, tornando os ganhos de desempenho cada vez menores. Isso levou a Intel a incorporar o cache L2 diretamente no processador a partir do Pentium Pro, abandonando o uso de cache na placa-mãe.

Inicialmente o cache L2 era um chip separado, que dividia o encapsulamento com o processador, mas a partir da segunda geração do Celeron (e do Pentium III Coppermine) ele passou a ser integrado diretamente ao processador, o que reduziu os tempos de acesso e também os custos.

Esta é uma foto do núcleo de um Pentium III Coppermine com seus 256 KB de cache L2 integrado, que são representados pelos 16 retângulos na parte inferior do processador. Você pode notar que o cache L2 ocupa uma área significativa do núcleo do processador, o que explica o fato de terem sido usados apenas 256 KB:



Daí em diante, o cache L2 integrado foi adotado em todos os processadores, do Athlon Thunderbird ao Core 2 Quad. Existem diferenças entre os caches usados pela Intel e a AMD (a Intel usa um cache inclusivo, enquanto a AMD usa um cache exclusivo, entre outras diferenças), mas em ambos os casos os papéis dos cache L1 e L2 são bem similares.

O cache L1 é sempre muito pequeno (de 32 a 128 KB) e oferece tempos de acesso muito baixos, equivalentes a apenas 3 ou 4 ciclos (o que em um processador de 3.0 GHz equivale a apenas 1 ou 1.33 nanossegundos). Entretanto, todo esse desempenho tem um custo, que é a necessidade de usar células com mais transistores, controladores mais sofisticados e mais trilhas de acesso, o que torna o cache L1 muito caro em termos de transistores usados.

O cache L2 por sua vez é baseado no uso de células mais lentas, com controladores mais simples e menos linhas de dados. Isso permite que o cache L2 seja sempre muito maior (de 256 KB a 2 MB); mas, em compensação, ele trabalha com tempos de acesso mais altos, de tipicamente 10 a 15 ciclos.

Embora possa soar estranho à primeira vista, essa relação é a que oferece o melhor custo-benefício na maioria dos casos, já que o bom desempenho do

cache L1 permite que o processador tenha acesso rápido aos dados na maioria das situações e o grande cache L2 serve como uma segunda parada para os casos em que ele não encontra o que precisa no L1.

Os processadores atuais usam controladores de cache bastante avançados, o que permite que os caches trabalhem com percentagens de acerto surpreendentemente boas considerando o tamanho. Tipicamente, o cache L1 responde por 80% dos acessos, o cache L2 responde por mais 18 ou 19% e a memória RAM responde pelos 1 ou 2% restantes. À primeira vista, pode parecer que não vale à pena sacrificar um espaço tão grande no processador para adicionar um grande cache L2 que responde por menos de 20% dos acessos, mas ao fazer as contas podemos ver que ele é bem importante.

Tomando como exemplo um processador onde o cache L1 trabalha com tempos de acesso de 3 ciclos, o cache L2 trabalha com 15 ciclos e a memória RAM com 140 ciclos e os caches respondem por respectivamente 80% e 19% dos acessos, teríamos a seguinte relação depois de um milhão de acessos:

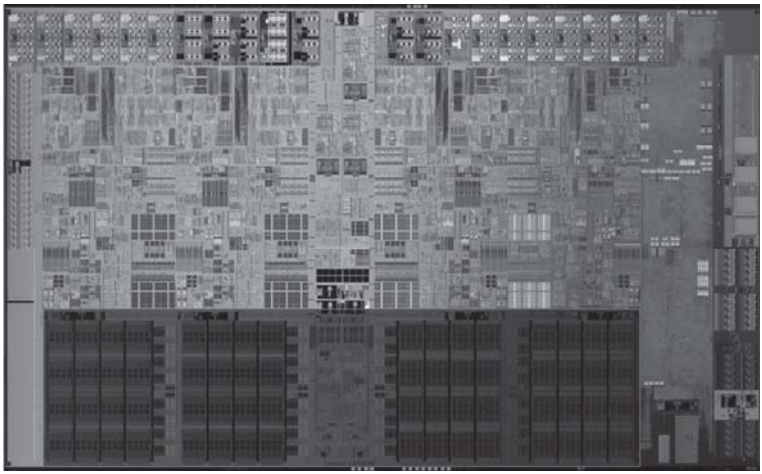
Cache L1 (80%): 2.400.000 ciclos
Cache L2 (19%): 2.850.000 ciclos
Memória (1%): 1.400.000 ciclos
Total: 6.650.000 ciclos

Você pode notar que mesmo respondendo por uma pequena parcela dos acessos, a memória RAM é responsável por um volume desproporcionalmente grande de ciclos de espera. Um aumento de apenas 1% na percentagem de acessos à memória causaria uma verdadeira tragédia, elevando o total do exemplo para mais de 8 milhões de ciclos.

É justamente por isso que processadores com caches maiores ou com controladores de memória integrados (latência mais baixa) oferecem muitas vezes ganhos de desempenho de 10% ou mais em relação aos antecessores. Da mesma maneira, um cache L1 maior ou mais rápido pode fazer uma grande diferença, mas apenas se o aumento não for às custas de uma redução no cache L2, já que pouco adianta melhorar o desempenho do cache L1 em uma ponta, se o processador vai perder bem mais tempo acessando à memória na outra.

A divisão tradicional entre cache L1 e cache L2 funcionou bem durante a fase dos processadores single-core e dual-core. Entretanto, com a introdução dos processadores quad-core passou a fazer mais sentido usar caches L1 e L2 menores e incluir um terceiro nível de cache. Com isso, temos quatro pequenos blocos de cache L1 e L2 (um para cada núcleo) e um grande cache L3 compartilhado entre todos.

Um bom exemplo é o Core i7 de 45 nm, que usa 64 KB de cache L1 e 256 KB de cache L2 por núcleo e usa um grande cache L3 de 8 MB compartilhado entre todos. Dentro do processador, ele corresponde à área sombreada no diagrama a seguir, novamente uma área considerável:



O cache L3 compartilhado no Core i7

Em PCs antigos os caches se limitavam a armazenar as últimas informações acessadas, guardando cópias de dados usados pelo processador e descartando as informações mais antigas ou acessadas com menos frequência. Os caches atuais são bem mais eficientes, incorporando algoritmos bem mais elaborados e sistemas de prefetch, que monitoram o fluxo de instruções e carregam antecipadamente dados que serão necessários nos ciclos seguintes. Desde o Pentium, o cache é também capaz de acelerar as operações de gravação, permitindo que o processador grave os dados diretamente no cache, deixando que o controlador se encarregue de gravá-los na memória posteriormente.

Outra curiosidade é que os primeiros processadores usavam caches unificados, que não faziam distinção entre dados e instruções, tratando ambos com o mesmo nível de prioridade. A partir do Pentium, o cache L1 passou a ser dividido em dois blocos independentes, um para dados e outro para instruções. Essa divisão permite que o controlador de cache use o espaço de forma mais eficiente e melhora a velocidade de acesso, já que os dois blocos passam a se comportar como dois caches independentes, permitindo que o processador leia dados e instruções simultaneamente.

Além dos caches, os processadores incluem também um conjunto de circuitos de TLB (Translation lookaside buffer), o componente que armazena endereços de memória, convertendo os endereços lógicos usados pelos aplicativos em execução nos endereços físicos nos chips de memória. O TLB é um circuito bem mais simples que os caches e é posicionado entre o cache L2 (ou L3) e a memória RAM.

Cada aplicativo (ou mais especificamente cada processo em execução) acha que tem à disposição um bloco contínuo de endereços de memória, enquanto na verdade está utilizando endereços espalhados por vários chips ou mesmo módulos de memória diferentes (ou até memória swap em alguns casos). Com isso, sempre que o processador precisa ler informações diretamente na memória RAM, é necessário primeiro converter os endereços usados pelo aplicativo nos endereços físicos da memória onde eles estão armazenados, verificando a entrada correspondente no TLB.

Sem o TLB, o processador precisaria fazer uma longa busca sequencial, pesquisando uma a uma as páginas de endereços da memória até encontrar os endereços correspondentes (um processo extremamente demorado), antes mesmo de iniciar o acesso propriamente dito.

Diferente dos caches, o TLB funciona como um buffer, que simplesmente armazena endereços em uso. Ele é um daqueles recursos que todos tomam como certo e que só recebe atenção quando algo dá errado (como no infame TLB Bug, que afetou as versões iniciais do Phenom, prejudicando o desempenho).

HD

Apesar de toda a sua importância, a memória RAM funciona apenas como uma mesa de trabalho, cujo conteúdo é descartado ao desligar o PC. Isso nos leva ao HD, que serve como uma unidade de armazenamento permanente para dados e programas.

O termo "HD" vem de "Hard Disk", ou disco rígido (ele é também chamado de "disco duro" no português de Portugal) e surgiu como uma maneira simples de diferenciá-los dos discos flexíveis encontrados nos disquetes.

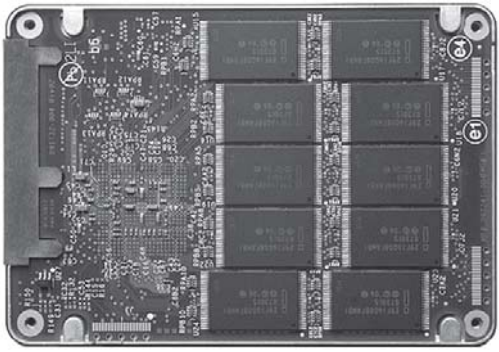
O HD armazena os dados em discos magnéticos que mantêm a gravação por vários anos. Os discos giram a uma grande velocidade e um conjunto de cabeças de leitura, instaladas em um braço móvel, faz o trabalho de gravar ou acessar os dados em qualquer posição nos discos. Junto com o CD-ROM, o HD é um dos poucos componentes mecânicos ainda usados nos micros atuais e, justamente por isso, é o que normalmente dura menos tempo (em média de três a cinco anos de uso contínuo) e o que inspira mais cuidados.

Na verdade, os discos magnéticos dos HDs são selados, pois a superfície magnética onde são armazenados os dados é extremamente fina e sensível. Qualquer grão de poeira que chegasse aos discos poderia causar danos à superfície, devido à enorme velocidade de rotação dos discos. Fotos em que

o HD aparece aberto são apenas ilustrativas, no mundo real ele é apenas uma caixa fechada sem tanta graça.

Apesar disso, é importante notar que os HDs não são fechados hermeticamente, muito menos a vácuo, como muitos pensam. Um pequeno filtro permite que o ar entre e saia, fazendo com que a pressão interna seja sempre igual à do ambiente. O ar é essencial para o funcionamento do HD, já que ele é necessário para criar o "colchão de ar" que evita que as cabeças de leitura toquem os discos.

A principal concorrente dos HDs é a memória Flash, que também serve como unidade de armazenamento permanente, funcionando basicamente como um HD sem partes móveis. Além de ser encontrada em pendrives e cartões de memória, a memória Flash é usada também nos SSDs, que lentamente estão roubando o espaço dos HDs em PCs de alto desempenho e também em notebooks:



Chips de memória Flash em em SSD

Os SSDs são instalados em portas SATA e formatados exatamente da mesma maneira que um HD magnético. A maioria dos modelos oferece um desempenho bem superior ao de um HD, com taxas de transferência mais altas e tempos de acesso mais baixos. O grande problema é o custo por megabyte, que é muito mais alto.

Isso faz com que os SSDs sejam mais adequados como unidades de boot para a instalação do sistema e dos aplicativos e não para armazenamento de grandes arquivos. Em vez de vender as calças para comprar um SSD de 320 GB, você pode comprar um SSD de 80 GB e mais um HD magnético de 1 TB por muito menos.

Além da questão da capacidade, a grande diferença entre os SSDs e os pendrives e cartões de memória é o desempenho. Os SSDs usam controladores bem mais complexos e utilizam vários chips de memória Flash, que são acessados em paralelo, multiplicando o desempenho.

Os pendrives e cartões utilizam controladores muito mais simples, que são quase sempre combinados com um único chip de memória Flash, resultando em um conjunto muito mais lento e pouco adequado ao grande volume de requisições de leitura a que um HD ou SSD é sujeitado no dia a dia. Nas placas atuais, é perfeitamente possível usar um pendrive para a instalação do sistema (basta configurar a placa-mãe para dar boot através da porta USB), mas o desempenho é muito ruim.

Concluindo, temos a questão dos live-CDs. Tradicionalmente, o sistema operacional era sempre instalado no HD antes de poder ser usado. Enquanto está trabalhando, o sistema precisa frequentemente modificar arquivos e configurações, o que seria impossível em um CD-ROM ou DVD, já que os dados gravados nele não podem ser alterados.

Isso mudou com o aparecimento do Knoppix outras distribuições Linux que rodam diretamente a partir do CD-ROM. Neste caso, um conjunto de modificações "enganam" o sistema, fazendo com que ele use a maior parte dos arquivos (os que não precisam ser alterados) a partir do CD-ROM, e salve os arquivos que forem alterados em um ramdisk, criado usando parte da memória RAM.

Isto tem algumas limitações: o desempenho é mais baixo e as configurações são perdidas ao desligar, já que tudo é armazenado na memória RAM. Pensando nisso, muitas distribuições permitem salvar as configurações em um pendrive ou em uma partição do HD, criando uma espécie de sistema híbrido, que roda a partir do CD-ROM, mas salva os arquivos em uma unidade de armazenamento permanente. Outra possibilidade é instalar diretamente o sistema em um pendrive, o que (na maioria dos casos) resulta em um desempenho bem melhor, graças aos tempos de acesso mais baixos.

Swap e cache de disco

Bem antigamente, nos anos 80 (época dos primeiros PCs), você só podia rodar programas que coubessem na memória RAM disponível. Naquela época, a memória RAM era absurdamente mais cara e o mais comum era usar apenas 256 ou 512 KB, milhares de vezes menos do que usamos hoje em dia. Os mais abonados tinham dinheiro para comprar um megabyte inteiro, mas nada além disso.

Se você quisesse rodar um programa com mais de 256 KB, tinha que comprar mais memória, não tinha conversa. Sem outra escolha, os programadores se esforçavam para deixar seus programas o mais compactos possíveis para que eles rodassem nos micros com menos memória.

A partir do 386 os PCs ganharam suporte ao uso de memória virtual, um recurso que permite usar espaço do HD para simular a presença de mais memória RAM, amenizando o problema. A memória virtual pode ser armazenada em um arquivo especialmente formatado no HD, ou em uma partição dedicada (como no caso do Linux) e a eficiência com que ela é usada varia bastante de acordo com o sistema operacional, mas ela permite que o sistema continue funcionando, mesmo com pouca memória disponível.

A ideia básica é o uso do arquivo ou partição do HD como uma área de troca (ou swap) onde o sistema pode arquivar páginas de memória que não estão sendo usadas, liberando memória física. Quando alguma das páginas é necessária, o sistema a copia de volta para a memória (possivelmente copiando outras de volta para o HD) e assim continuamente, até que você feche alguns aplicativos e o sistema volte a ter memória disponível.

O grande problema com o uso de swap é que o HD é muito mais lento que a memória RAM. Enquanto uma sequencia de 4 leituras em um módulo de

memória DDR2-800 demora cerca de 35 bilionésimos de segundo, um acesso a um setor qualquer do HD demora pelo menos 10 milésimos. A taxa de transferência nominal do mesmo módulo de memória é de 6.4 GB/s, enquanto que mesmo um HD rápido, de 7200 RPM, tem dificuldades para superar a marca de 80 MB/s, mesmo quando lendo setores sequenciais. Ou seja, a memória RAM possui nesse caso um tempo de acesso quase 300.000 vezes menor e uma taxa de transferência contínua quase 100 vezes maior que o HD.

Se lembrarmos que a memória RAM já é muito mais lenta que o processador (justamente por isso temos os caches L1, L2 e L3), fica fácil perceber o quanto o uso de memória swap por falta de memória RAM física pode prejudicar o desempenho do sistema.

A fórmula é simples: quanto menos memória RAM, mais swap é usado e mais lento o sistema fica. Sempre que um aplicativo precisa de dados armazenados no swap, o sistema precisa primeiro copiar a página de volta para a memória RAM, para que só então o processador possa acessá-lo e continuar o processamento. É por isso que micros antigos podem ficar tão lentos, chegando ao ponto de demorarem minutos para chavear entre dois aplicativos.

Hoje em dia, o uso de swap não é mais um problema tão grande pois a queda nos preços da memória fez com que a maioria dos micros novos passassem a vir com pelo menos 1 GB. Entretanto, até pouco tempo atrás era muito comum ver PCs novos com apenas 256 ou até mesmo 128 MB de memória (dos quais ainda eram subtraídos a memória usada pelo chipset de vídeo onboard), que se arrastavam para executar até mesmo as tarefas mais básicas. Nesses casos, de nada adiantava um processador mais rápido.

É fácil monitorar o uso de swap. No Windows XP ou Vista basta pressionar Ctrl+Alt+Del e acessar o gerenciador de tarefas, enquanto no Linux você pode usar o comando "free" ou um aplicativo de gerenciamento, como o ksysguard.

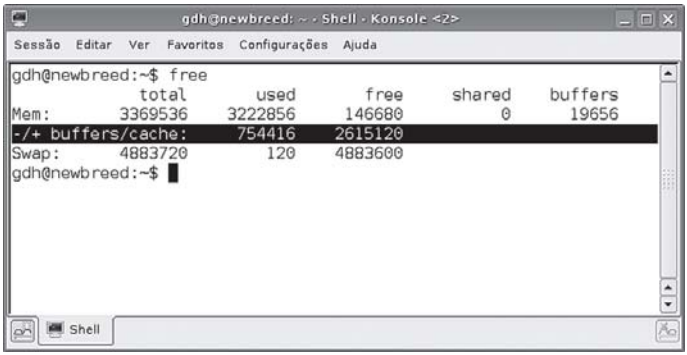
No caso do Windows Vista é possível usar um pendrive como memória adicional, através do ReadyBoost. Neste caso entretanto, o pendrive é usado como uma extensão da memória swap e não como um substituto da memória RAM. Como o pendrive oferece tempos de acesso mais baixos, ele acaba sendo mais eficiente que o HD nessa tarefa, muito embora a taxa de leitura seja geralmente mais baixa.

Esse recurso pode ajudar em micros com pouca memória RAM e também reduzir o tempo de carregamento dos programas. É uma opção para casos em que você já tem o pendrive e procura um uso para ele, mas não espere milagres. Em se tratando de memória, não existe o que inventar: ou você procura um sistema operacional e programas mais leves, ou compra mais memória. Não dá para ficar em cima do muro.

Por outro lado, quando você tem instalado mais memória do que o sistema realmente precisa, é feito o inverso. Ao invés de copiar arquivos da memória para o HD, arquivos do HD contendo os programas, arquivos e bibliotecas que já foram anteriormente abertos é que são copiados para a memória, fazendo com que o acesso a eles passe a ser instantâneo. Os programas e arquivos passam a ser abertos de forma gritantemente mais rápida, como se você tivesse um HD muito mais rápido do que realmente é.

Esse recurso é chamado de **cache de disco** e (sobretudo no Linux) é gerenciado de forma automática pelo sistema, usando a memória disponível. Naturalmente, o cache de disco é descartado imediatamente quando a memória precisa ser usada para outras coisas. Ele é apenas uma forma de aproveitar o excedente de memória, sem causar nenhum efeito desagradável.

Você pode acompanhar o uso de cache de disco usando o comando free. Ao executá-lo em um PC que já está ligado há algum tempo, você vai notar que quase toda a memória aparece como ocupada. No screenshot a seguir, temos um PC com 3 GB, onde ele reporta apenas 146 MB disponíveis:



Como pode imaginar, quase tudo está sendo usado pelo cache de disco. A memória realmente ocupada aparece na segunda linha (-/+ buffers/cache). No exemplo temos apenas 754 MB de memória em uso, o que significa que o sistema está usando mais de 2 GB de cache de disco, o que resulta em um sistema muito mais responsível, onde aplicativos já usados carregam muito rápido.

Ironicamente, a forma mais eficiente de melhorar o desempenho do HD, na maioria das aplicações, é instalar mais memória, fazendo com que uma quantidade maior de arquivos possa ser armazenada no cache de disco. É por isso que servidores de arquivos, servidores proxy e servidores de banco de dados costumam usar muita memória RAM, em muitos casos 16 GB ou mais.

Concluindo, outra forma de melhorar o desempenho do HD (sem precisar recorrer a um SSD) é usar **RAID**, onde dois ou quatro HDs passam a ser acessados como se fossem um só, multiplicando a velocidade de leitura e gravação. Esse tipo de RAID, usado para melhorar o desempenho, é chamado de RAID 0. Existe ainda o RAID 1, onde são usados dois HDs, mas o segundo é uma cópia exata do primeiro, que garante que os dados não sejam perdidos no caso de algum problema mecânico em qualquer um dos dois. O RAID tem se tornado um recurso relativamente popular, já que atualmente a maioria das placas-mãe já vêm com controladoras RAID onboard.

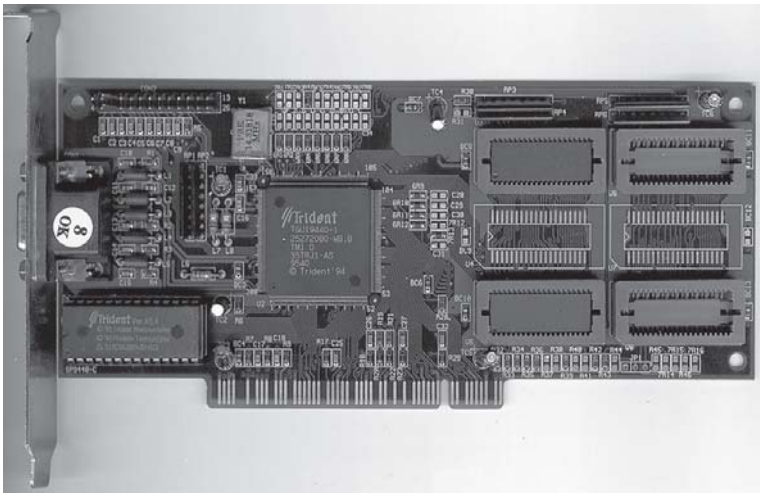
Placa de vídeo

Depois do processador, memória e HD, a placa de vídeo é provavelmente o componente mais importante do PC. Originalmente, as placas de vídeo eram dispositivos simples, que se limitavam a mostrar o conteúdo da memória de

vídeo no monitor. A memória de vídeo continha um simples bitmap da imagem atual, atualizada pelo processador, e o RAMDAC (um conversor digital-analógico que faz parte da placa de vídeo) lia a imagem periodicamente e a enviava ao monitor.

A resolução máxima suportada pela placa de vídeo era limitada pela quantidade de memória de vídeo. Na época, memória era um artigo caro, de forma que as placas vinham com apenas 1 ou 2 MB. As placas com 1 MB permitiam usar no máximo 800x600 com 16 bits de cor, ou 1024x768 com 256 cores, limitadas ao que cabia na memória de vídeo.

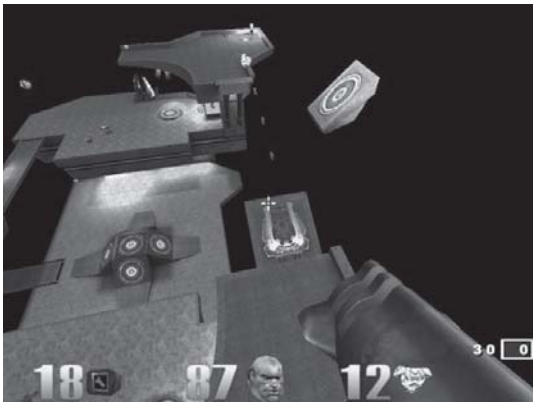
Esta da foto a seguir é uma Trident 9440, uma placa de vídeo muito comum no início dos anos 90. Uma curiosidade é que ela foi uma das poucas placas de vídeo "atualizáveis" da história. Ela vinha com apenas dois chips de memória, totalizando 1 MB, mas era possível instalar mais dois, completando 2 MB. Hoje em dia, atualizar a memória da placa de vídeo é impossível, já que as placas utilizam módulos BGA, que podem ser instalados apenas em fábrica.



Trident 9440

Em seguida, as placas passaram a suportar recursos de aceleração, que permitem fazer coisas como mover janelas ou processar arquivos de vídeo de forma a aliviar o processador principal. Esses recursos melhoram bastante a velocidade de atualização da tela (em 2D), tornando o sistema bem mais responsivo.

Finalmente, as placas deram o passo final, passando a suportar recursos 3D. Imagens em três dimensões são formadas por polígonos, formas geométricas como triângulos e retângulos em diversos formatos. Qualquer objeto em um jogo 3D é formado por um grande número destes polígonos. Cada polígono tem sua posição na imagem, um tamanho e cor específicos. O "processador" incluído na placa, responsável por todas estas funções é chamado de GPU (Graphics Processing Unit, ou Unidade de Processamento Gráfico).



Quase todo o processamento da imagem em games 3D é feito pela placa 3D

Para tornar a imagem mais real, são também aplicadas texturas sobre os polígonos. Uma textura nada mais é do que uma imagem 2D comum, aplicada sobre um conjunto de polígonos. O uso de texturas permite que um muro realmente tenha o aspecto de um muro de pedras, por exemplo, já que podemos usar a imagem de um muro real sobre os polígonos. Quanto maior o número de polígonos usados e melhor a qualidade das texturas aplicadas sobre eles, melhor será a qualidade final da imagem. Este demo da nVidia mostra um exemplo de aplicação de texturas sobre uma estrutura de polígonos:

Polígonos e imagem finalizada

O processo de criação de uma imagem tridimensional é dividido em três etapas, chamadas de desenho, geometria e renderização. Na primeira etapa, é criada uma descrição dos objetos que compõem a imagem, ou seja: quais polígonos fazem parte da imagem, qual é a forma e tamanho de cada um, qual é a posição de cada polígono na imagem, quais serão as cores usadas e, finalmente, quais texturas e quais efeitos 3D serão aplicados. Depois de feito o "projeto" entramos na fase de geometria, onde a imagem é efetivamente criada e armazenada na memória da placa 3D.

Ao final da etapa de geometria, todos os elementos que compõem a imagem estão prontos. O problema é que eles estão armazenados na memória da placa de vídeo na forma de um conjunto de operações matemáticas, coordenadas e texturas, que ainda precisam ser transformadas na imagem que será exibida no monitor. É nessa fase que chegamos à parte mais complexa e demorada do trabalho, que é a renderização da imagem.

Essa última etapa consiste em transformar as informações armazenadas na memória em uma imagem bidimensional que será mostrada no monitor. O processo de renderização é muito mais complicado do que parece; é necessário determinar (a partir do ponto de vista do espectador) quais polígonos estão visíveis, aplicar os efeitos de iluminação adequados, etc.

Apesar de o processador também ser capaz de criar imagens tridimensionais (o que é feito ao usar algoritmos de renderização que trabalham via software), ele não é muito rápido na tarefa. A placa 3D por sua vez é capaz de gerar imagens e aplicar efeitos a uma grande velocidade, o que torna possível rodar jogos 3D complexos com um bom FPS, com a placa 3D se encarregando do processamento 3D e o processador cuidando das demais tarefas.

A grande diferença é que o processador utiliza uma arquitetura serializada, otimizada para o processamento de sequências de instruções (como as usadas por aplicativos) e não para o brutal volume de cálculos necessários para renderizar uma imagem 3D. As placas 3D por sua vez utilizam uma arquitetura paralela, com vários processadores simples trabalhando em conjunto.

Em um exemplo simplista, o processador seria um matemático capaz de executar cálculos complexos, porém um de cada vez, enquanto a placa 3D seria um batalhão de alunos de primeiro grau trabalhando em conjunto, que conseguem resolver apenas cálculos simples, porém a uma grande velocidade.

Com a evolução das placas 3D, os jogos passaram a utilizar gráficos cada vez mais elaborados, explorando os recursos das placas recentes. Isso criou um círculo vicioso, que faz com que você precise de uma placa razoavelmente recente para jogar qualquer jogo atual.

As placas 3D atuais são praticamente um computador à parte. Além da quantidade generosa de memória RAM, acessada através de um barramento muito mais rápido que a memória do sistema, o chipset de vídeo é bem mais complexo e absurdamente mais rápido que o processador principal no processamento de gráficos. O chipset AMD Cypress que equipa a Radeon 5870, por exemplo, possui 2.15 bilhões de transistores, bem mais do que qualquer processador da geração atual.

As placas 3D dedicadas também incluem uma quantidade generosa de memória de vídeo (1 GB ou mais nos modelos mais recentes), acessada através de um barramento muito rápido. A GPU (o chipset da placa) é também muito poderosa, de forma que as duas coisas se combinam para oferecer um desempenho monstruoso em 3D.

Com a introdução do PCI Express, surgiu também a possibilidade de instalar duas, três ou até mesmo quatro placas, ligadas em SLI (no caso das placas nVidia) ou CrossFire (no caso das placas AMD/ATI), uma solução cara mas que permite obter um desempenho bem superior ao oferecido por apenas uma placa.