

Software Básico

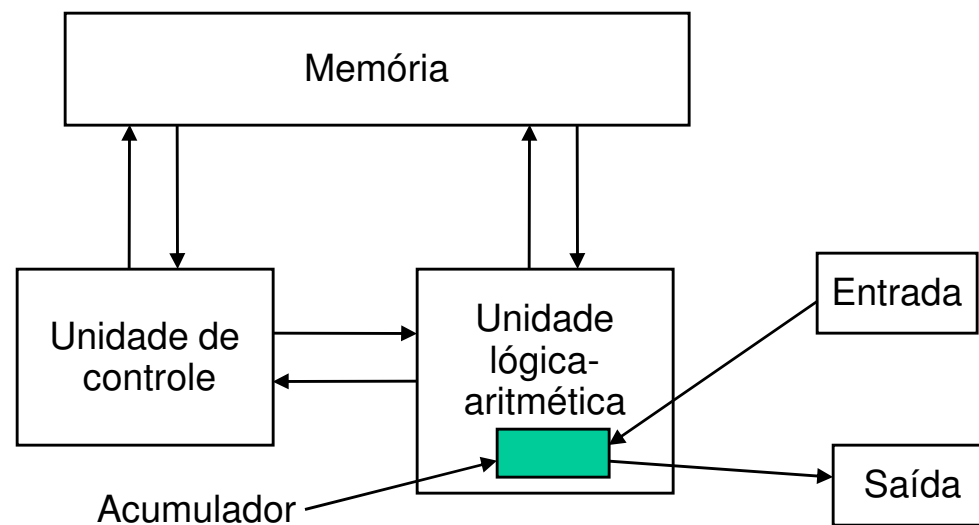
Introdução à
organização de computadores

Roteiro

- Organização da CPU
- Paralelismo (+cap.8)
- Memória (primária e secundária)
- Dispositivos de entrada e saída
- Representação de dados (+ap.A,B)

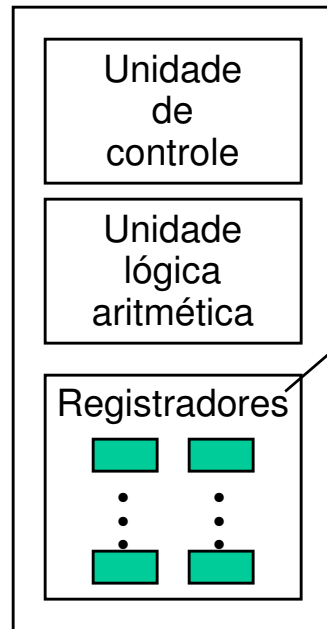
Arquitetura de von Neumann

- Arquitetura dos computadores modernos
- Programa armazenado em memória
- Unidades de controle e aritmética



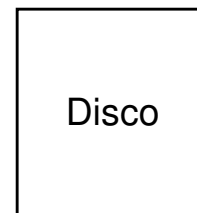
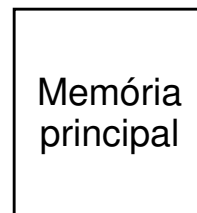
Organização de um computador simples

Unidade central de processamento
(CPU/UCP)



Registradores importantes:
PC: Program Counter
IR: Instruction Register

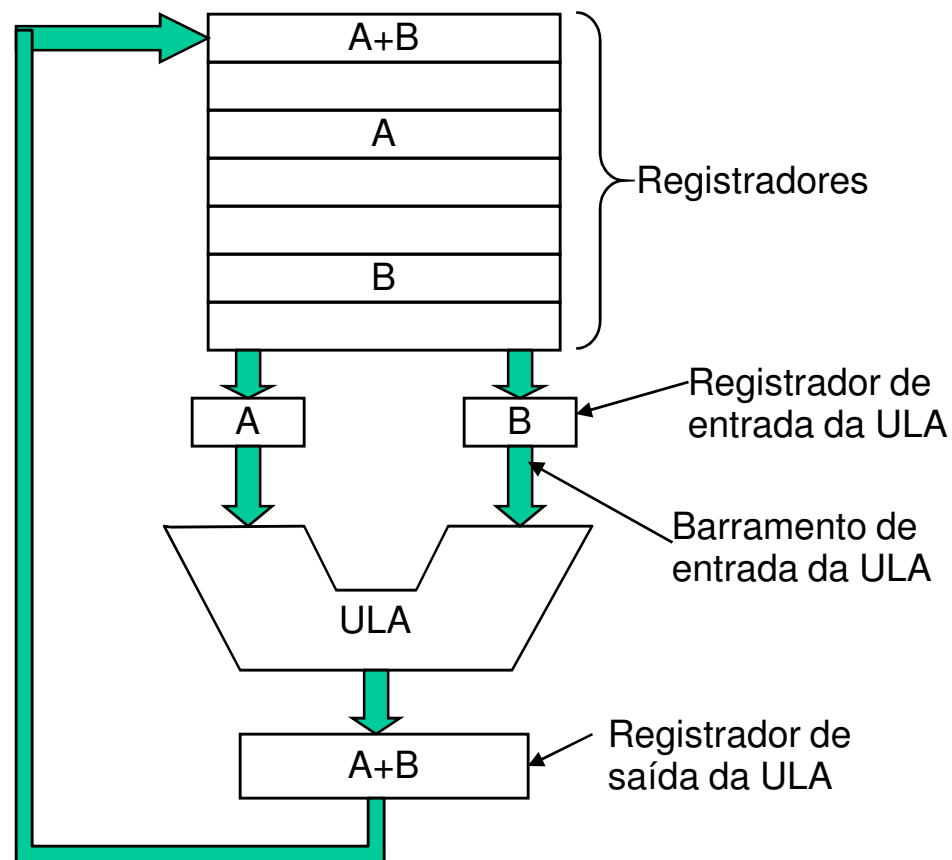
Dispositivos de E/S



Barramento

Unidade lógica-aritmética

- Vários registradores (normalmente 32)
- Operações de registrador a registrador
- 2 registradores de entrada e 1 de saída da ULA (nem todos projetos)
- ULA: dois tipos de instruções
 - Registrador-memória
 - Registrador-registrador
- Velocidade do processador limitada pela velocidade do *data path* (*ciclo caminho dos dados*)



Ciclo de execução de instruções

- Busca instrução da memória para o IR
- Atualiza o contador de programa (PC)
- Determina o tipo da instrução
- Se a instrução usa palavra na memória:
 - Determina onde palavra está
 - Busca palavra pra registrador
- Se a instrução usa dados em registradores:
 - determina a posição do dado
 - busca o dado para a unidade de execução
- Executa a instrução
- Reinicia o ciclo para a próxima instrução

Microprograma: ciclo em software

- Instruções da máquina são buscadas da memória e interpretadas
 - Subdivide as instruções em pequenas etapas
 - Apelo principalmente se conjunto de instruções da máquina for grande e complexo
- A máquina real pode ser bem mais simples, pois executa apenas os passos do interpretador
- Uso de microprogramas permitiu o desenvolvimento de processadores sofisticados com *hardware* simples
- Mas interpretação limita o desempenho

Vantagens da microprogramação (< '79)

- Permite criação de instruções de máquina sofisticadas, mais semelhantes às necessidades do programador (DEC VAX)
- Libera os arquitetos de limitações do *hardware* em máquinas baratas
 - Famílias de máquinas compatíveis: arquitetura única
- Baixo custo
- Motorola 68000 x Zilog Z8000: ambos grande conjunto de instruções
 - O Motorola era micro-programado e foi um grande sucesso na sua época
 - O Zilog fracassou, em parte pelo tempo no desenvolvimento do projeto, de execução direta

Problemas da microprogramação (> '79)

- Com o aumento da velocidade do hardware (ROM x RAM)
 - microprogramas ficaram relativamente mais lentos (busca de micro-instruções depende da memória)
- O processador não precisa ser semelhante à linguagem de alto nível:
 - para isso existem compiladores!
- Processadores simples podem ser mais rápidos se têm poucas instruções
 - Simplifica ciclo de caminho dos dados

Máquinas RISC x CISC

- RISC (Patterson / Hennessy):
 - poucas instruções, muito rápidas
 - instruções de tamanho fixo
 - apenas duas instruções para acesso à memória: *load* e *store*
- CISC:
 - instruções variadas
 - várias formas de acesso à memória
 - instruções complexas e variáveis

Máquinas RISC x CISC

- Durante anos, CPUs RISC conseguiram manter-se à frente das contemporâneas CISC
 - RISC: Sun Sparc, HP-PA, DEC Alpha
 - CISC: Intel 486, Pentium, Pentium II
- Com o passar do tempo, máquinas CISC passaram a explorar os princípios RISC
 - Pentium IV: CISC de coração RISC
 - instruções simples executadas diretamente
 - instruções complexas são decompostas (microprograma)

Princípios das máquinas atuais

(princípios de projeto RISC)

- Todas instruções *comuns* devem ser executadas diretamente pelo hardware
 - Instruções complexas podem ser interpretadas em termos das instruções simples
- Mais importante que a duração de uma instrução é a taxa de execução das instruções
 - Ex: iniciar o maior # possível de instruções /seg
 - Paralelismo / pipelining , execução fora de ordem
- Instruções devem ser facilmente decodificadas pelo processador
 - Instruções regulares, comprimento fixo, pequeno # campos, pequeno # de formatos diferentes
- Apenas *load* e *store* acessam a memória, as outras, registradores
- Grande número de registradores (mínimo 32)

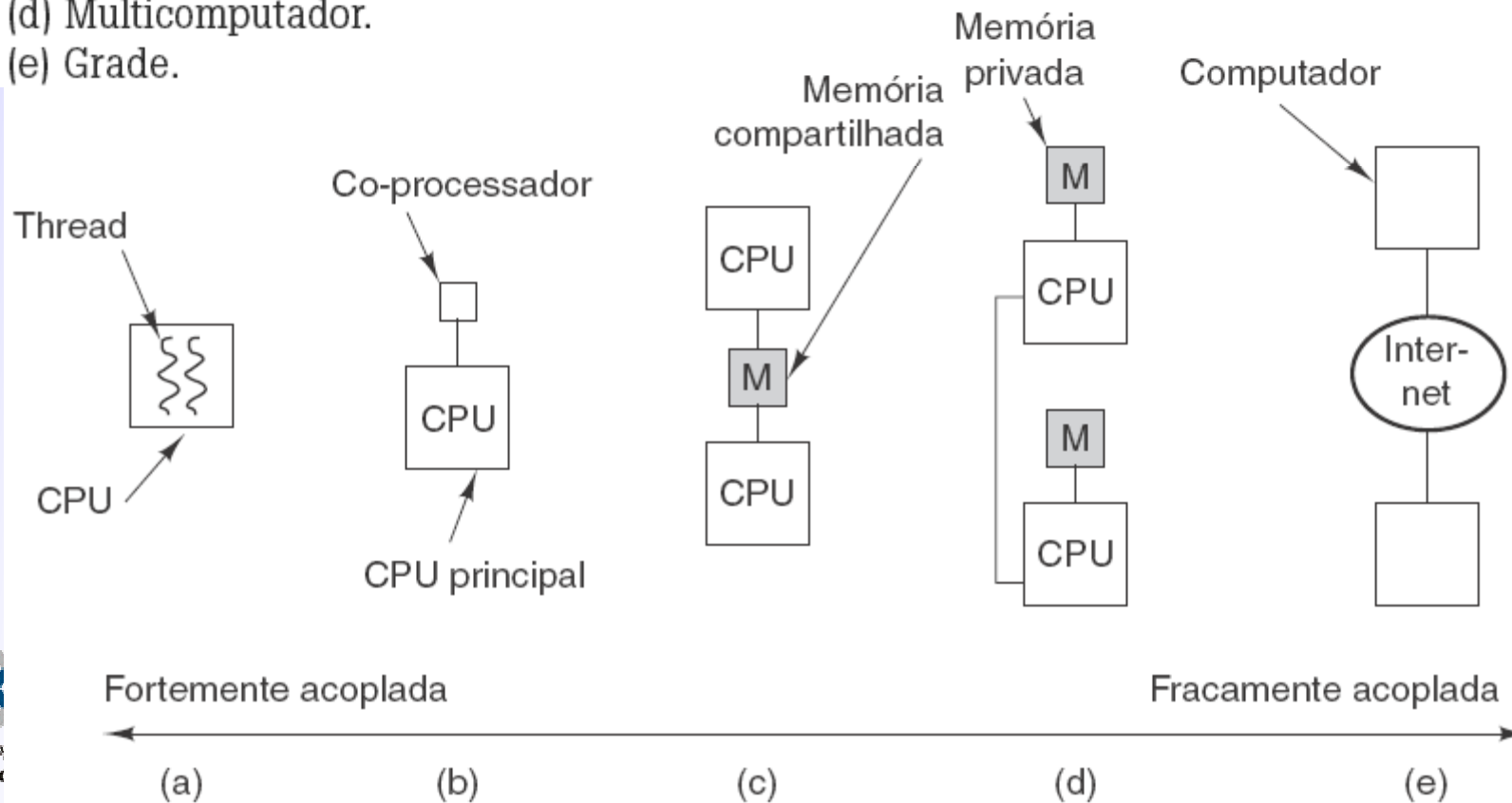
Como tornar o processador mais rápido?

- Aumentar a frequência do clock
 - Limitações físicas: capacitâncias, dissipação
- Aumentar o número de operações em paralelo
 - Se há espaço, por que não colocar mais hardware para fazer mais coisas?

Inclui partes do cap. 8

Arquiteturas de computadores paralelos

- (a) Paralelismo no chip.
- (b) Co-processador.
- (c) Multiprocessador.
- (d) Multicomputador.
- (e) Grade.



Paralelismo no processador

- Aumentar a velocidade dos processadores só permite um certo ganho
- Para ganhos ainda maiores, os processadores devem ser capazes de executar várias operações em paralelo
- Isso pode ser obtido de diversas formas

Paralelismo no processador

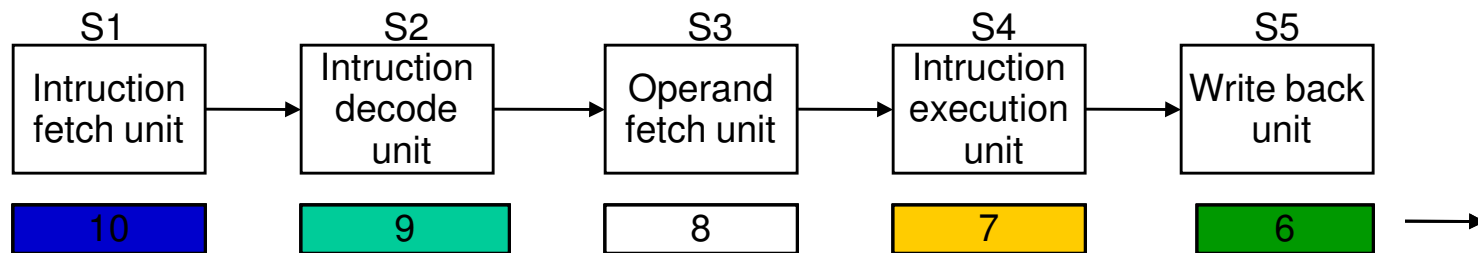
- Paralelismo a nível de instruções:
 - executar instruções concorrentemente
 - Pipelining
 - Arquiteturas super-escalares
 - VLIW
- Paralelismo a nível de *threads*:
 - executar diferentes seqüências de instruções concorrentemente
 - *Multi-threading* (diferentes granularidades)

Paralelismo a nível de instruções

- A nível de instruções, o processamento de cada instrução pode ser dividido e atribuído a módulos diferentes
 - Buffer de busca antecipada (IBM Stretch, 1959): Instrução podia ser executada imediatamente uma vez no buffer sem esperar leitura da memória (gargalo)
- Se os módulos são isolados, cada um pode se dedicar a uma instrução na seqüência do programa
- Cuidado: instruções podem depender uma das outras
 - Leitura e escrita de registradores
 - Desvios condicionais

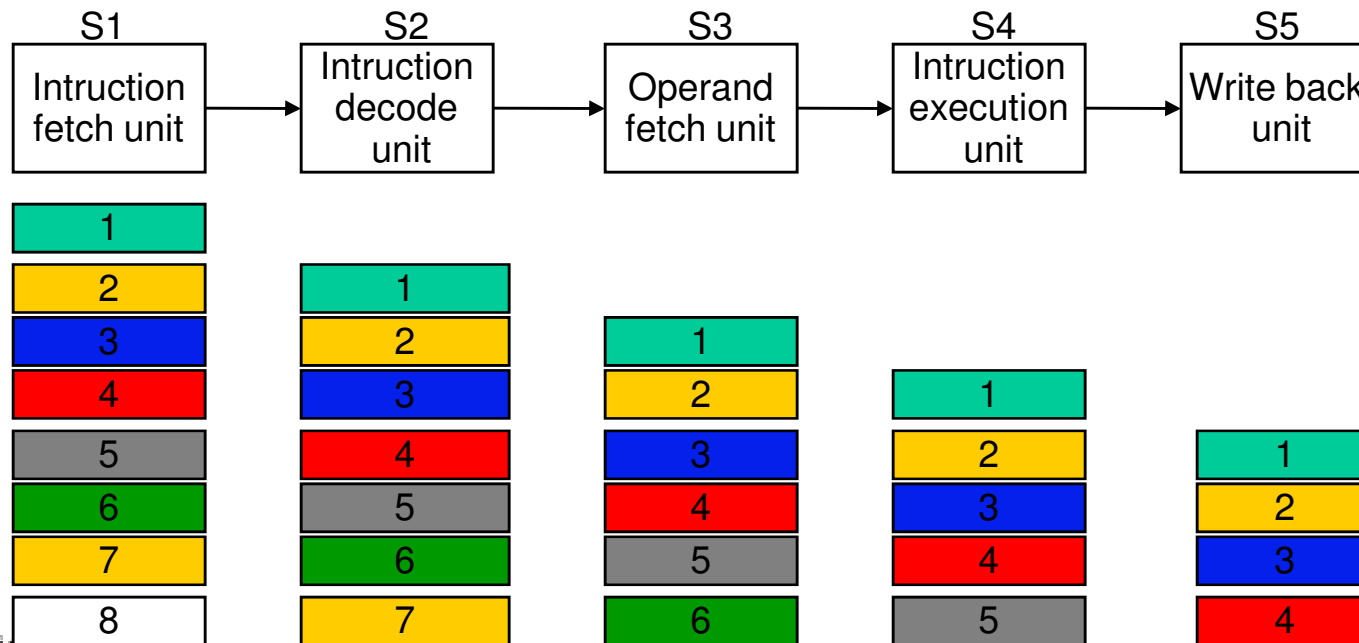
Pipeline

- Dividir o ciclo de execução em partes e definir uma “linha de montagem”



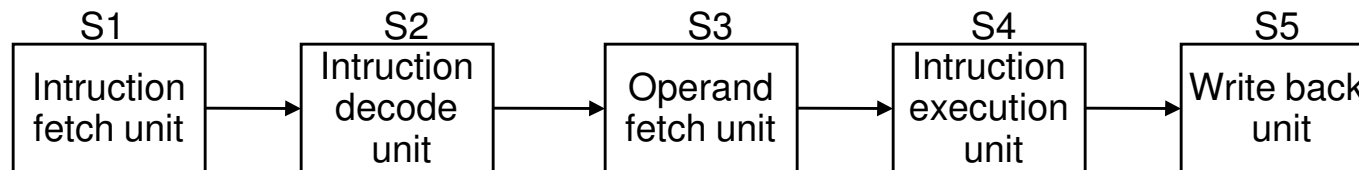
Pipeline

- Dividir o ciclo de execução em partes e definir uma “linha de montagem”



Pipeline

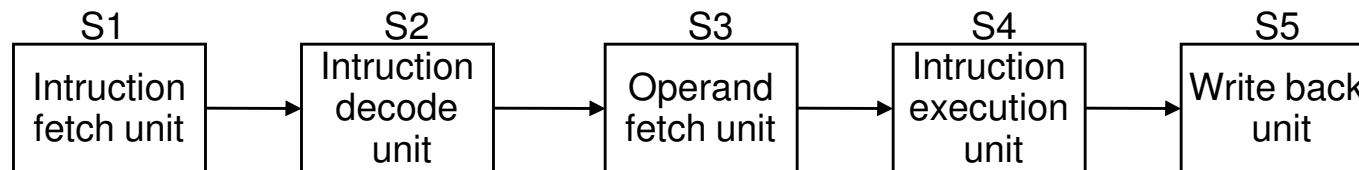
- Dividir o ciclo de execução em partes e definir uma “linha de montagem”



Compromisso: latência X taxa de processamento

Travamento do pipeline (stalls)

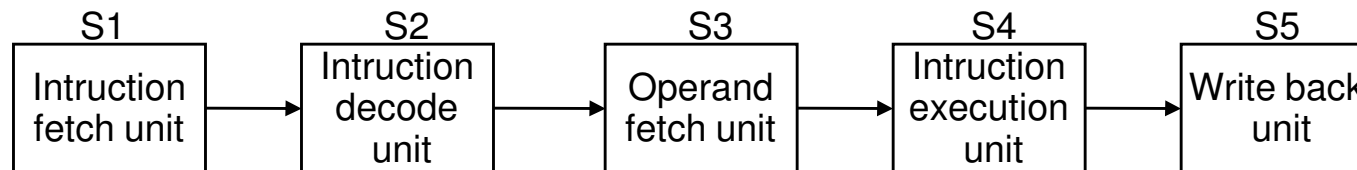
- Se uma instrução depende do resultado de uma outra, anterior, elas podem não poder ser executadas em seqüência



- Por exemplo:
 - Somar registradores R1 e R2
 - Desviar se valor resultante for zero

Travamento do pipeline (stalls)

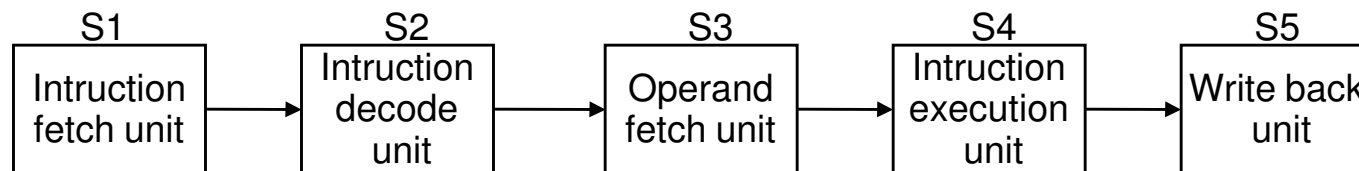
- Por exemplo:
 - Somar registradores R1 e R2
 - Desviar se valor resultante for zero



- Somente ao fim de S4 será possível saber qual o valor resultante da soma
- Se o desvio for tomado, as instruções em S1 e S2 precisam ser descartadas!

Travamento do pipeline (stalls)

- Por exemplo:
 - Leitura da memória para um registrador
 - Uso do registrador em qualquer operação



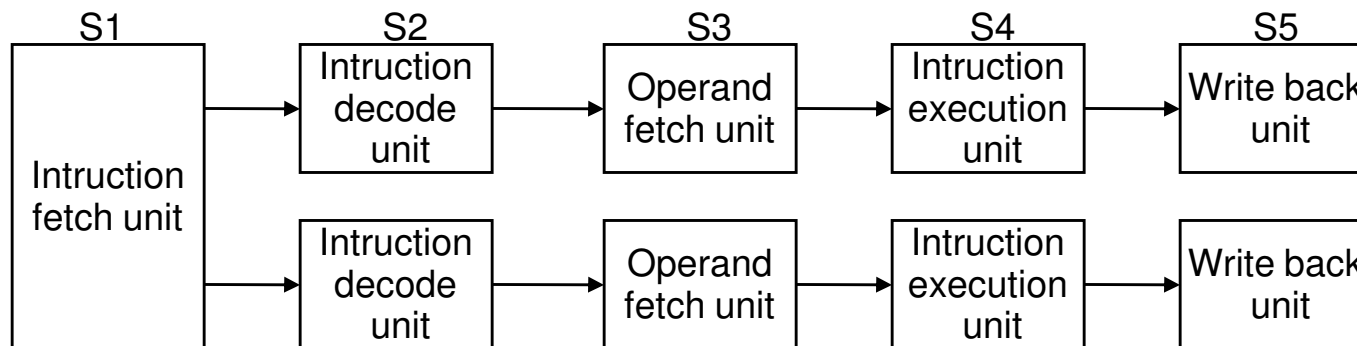
- Acessos à memória (mesmo cache) tendem a levar mais que o tempo de um estágio do pipe
- Se a instrução seguinte tentar usar o registrador, não verá o valor correto

Outras formas de paralelismo

- Devido aos problemas de travamento, há um limite até onde um pipeline pode aumentar a taxa de execução de instruções
- Depois disso, o que fazer?

Se uma pipeline já ajuda...

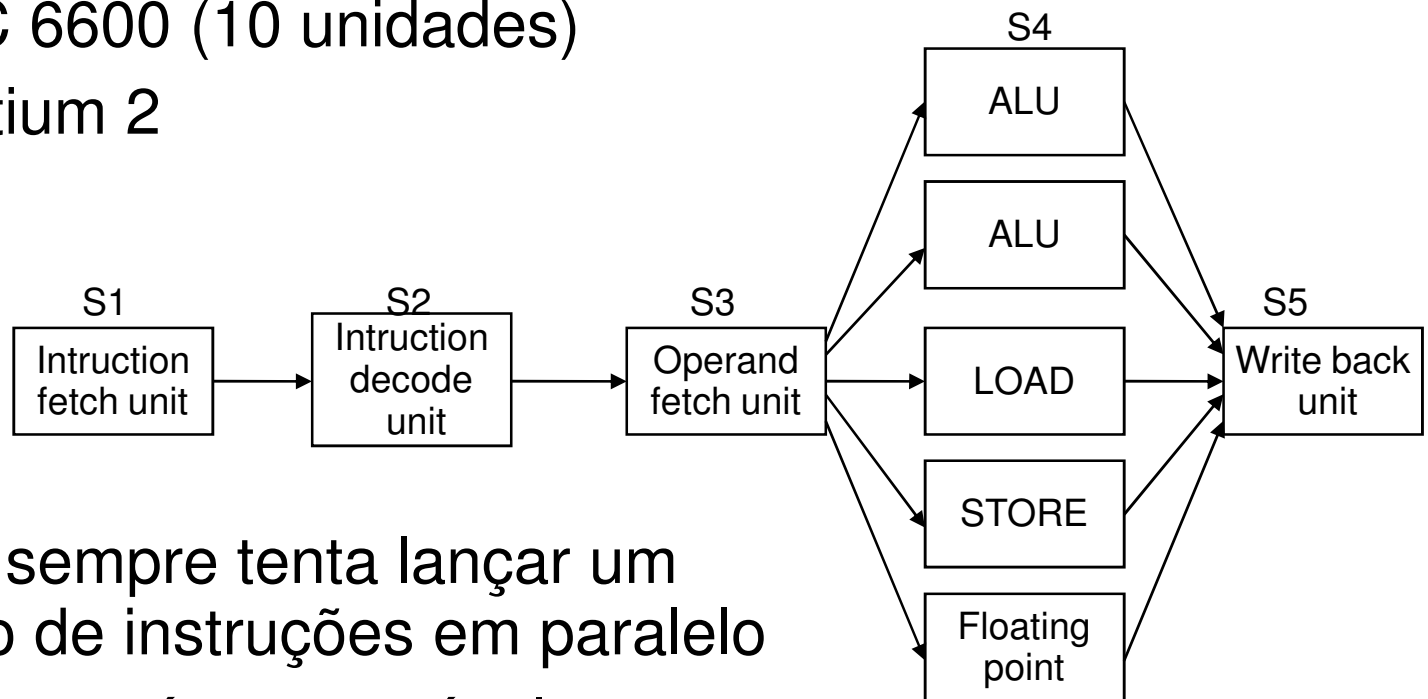
Múltiplos pipelines: Pentium (pipeline u x pipeline v)



- Nesse caso é preciso determinar se as instruções podem ser executadas em paralelo inicialmente (stalls, dependências)
 - S4 pode dominar os tempos
 - Os demais estágios são muito mais simples

Se a execução toma mais tempo...

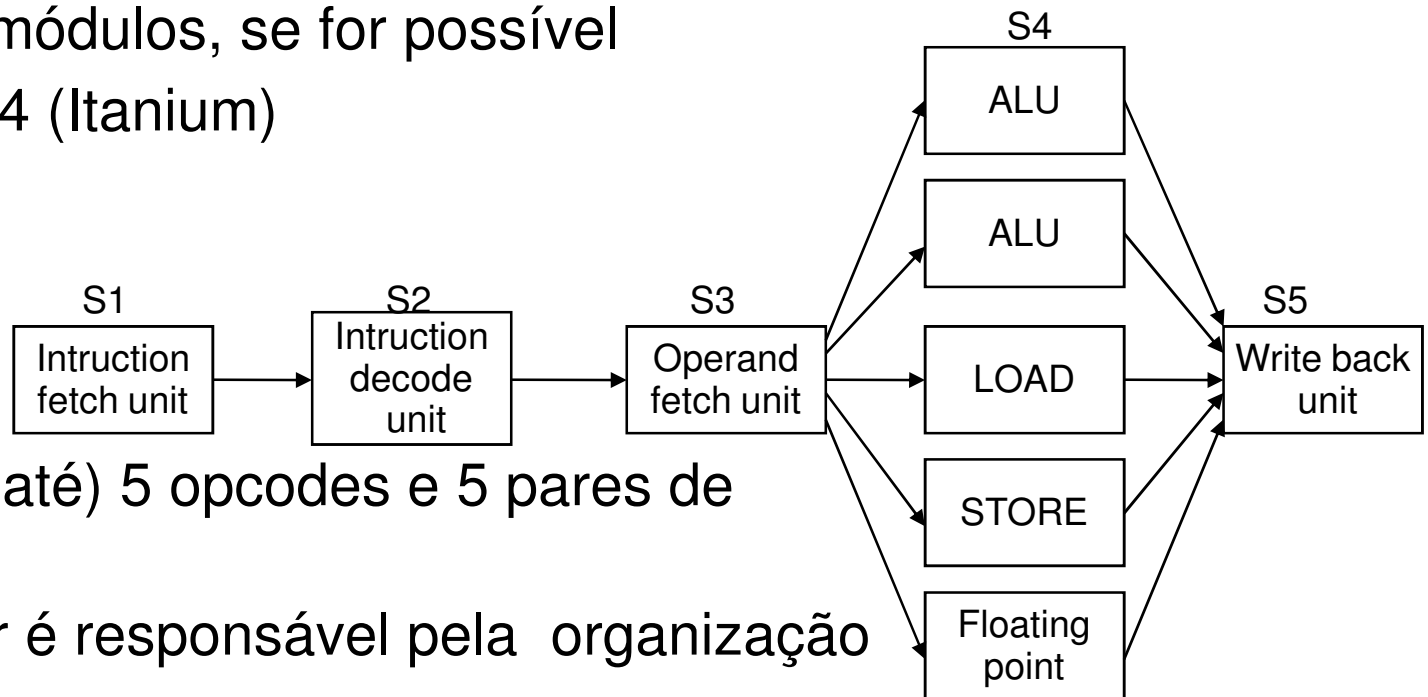
- Super-escalar: emite várias instruções se possível
 - CDC 6600 (10 unidades)
 - Pentium 2



- CPU sempre tenta lançar um grupo de instruções em paralelo
- Hardware é responsável por detectar conflitos e resolvê-los

Se a execução toma mais tempo...

- *Very Long Instruction Word (VLIW)*: cada instrução usa vários módulos, se for possível
 - IA 64 (Itanium)



- Instrução: (até) 5 opcodes e 5 pares de operandos.
- Compilador é responsável pela organização do código



Hardware não precisa se preocupar com conflitos

Paralelismo a nível de threads

- Misses nos caches de nível 1 e 2: atraso grande (pipeline pára / stall)
- Instruções têm interdependências claras
- Apenas um certo nível de paralelismo é viável na seqüência de execução de um programa
 - Dependências entre instruções
- Para ganhos maiores é preciso usar vários fluxos de execução ao mesmo tempo
 - Se um ciclo fica bloqueado, outro pode executar

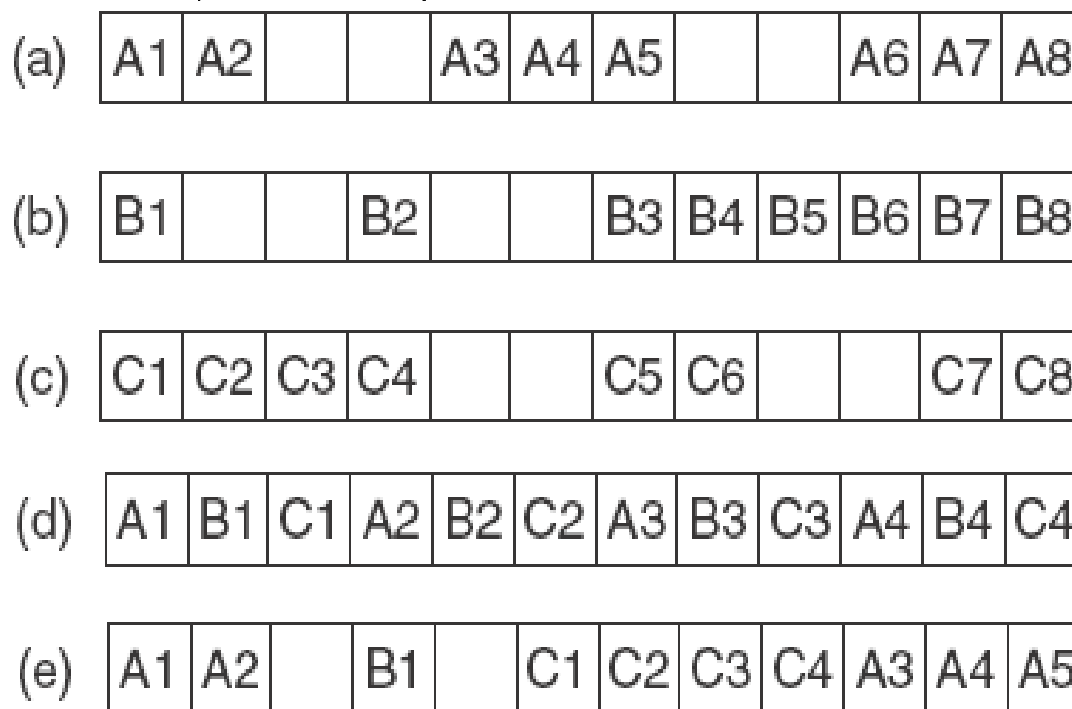
Multi-threading (Intel HyperThreading)

- CPUs passam a ter ciência de diversas seqüências de instruções para executar
 - *Multi-threading grão fino:*
 - *Cada estágio do pipeline executa uma thread*
 - *Multi-threading grão grosso:*
 - *Pipeline muda de thread somente a cada stall*
 - *Multi-threading simultâneo (em super-escalares):*
 - *Cada stall pode ser identificado isoladamente*

Multi-threading: um pipeline

(a)–(c) Três threads. Os retângulos vazios indicam que o thread parou esperando por memória.
 (d) Multithreading de granulação fina.
 (e) Multithreading de granulação grossa.

Miss no cache 1:
 stall por 2 ciclos



Ciclo →

Multi-threading: pipeline super-escalar

(a)–(c) Três threads. Os retângulos vazios indicam que o thread parou esperando por memória.

(a)

A1	A2			A3	A4	A5			A6	A7	A8
----	----	--	--	----	----	----	--	--	----	----	----

(b)

B1			B2			B3	B4	B5	B6	B7	B8
----	--	--	----	--	--	----	----	----	----	----	----

(c)

C1	C2	C3	C4			C5	C6			C7	C8
----	----	----	----	--	--	----	----	--	--	----	----

Ciclo →

A1	B1	C1	A3	B2	C3	A5	B3	C5	A6	B5	C7
A2		C2	A4		C4		B4	C6	A7	B6	C8

Ciclo →

(a) Multithreading de granulação fina

A1	B1	C1	C3	A3	A5	B2	C5	A6	A8	B3	B5
A2		C2	C4	A4			C6	A7		B4	B6

Ciclo →

Multithreading de granulação grossa

A1	B1	C2	C4	A4	B2	C6	A7	B3	B5	B7	C7
A2	C1	C3	A3	A5	C5	A6	A8	B4	B6	B8	C8

Ciclo →

Multithreading simultâneo

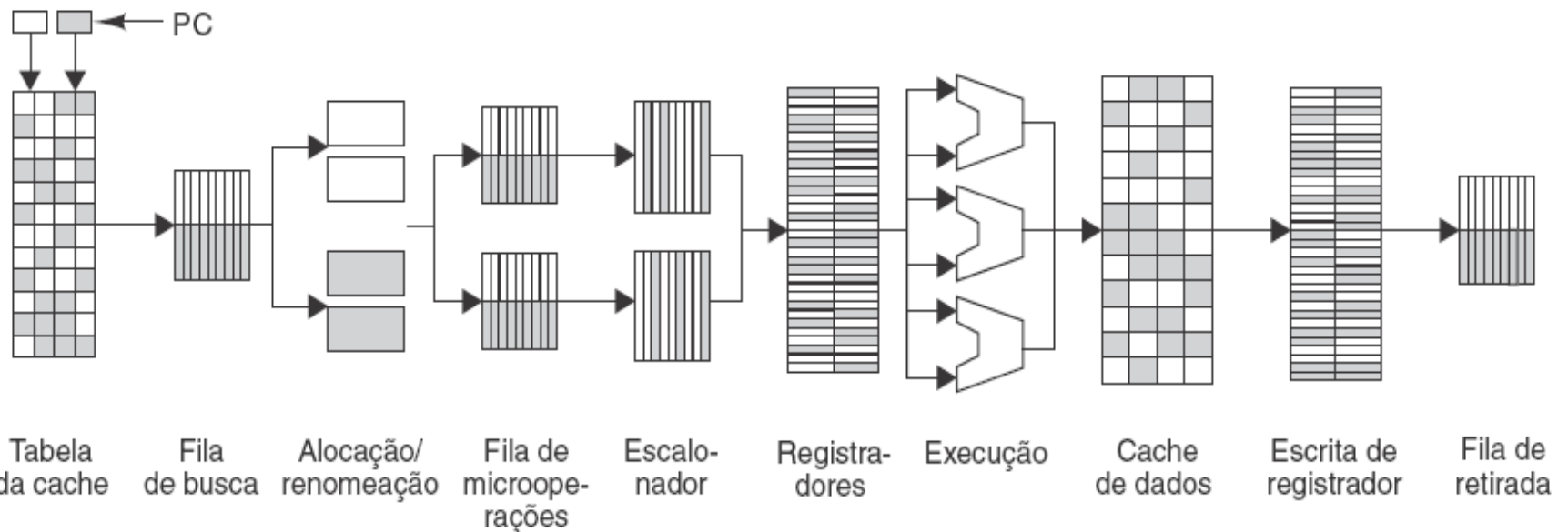
Pentium 4: como aumentar desempenho?

1. Aumentar velocidade do relógio?
 - Mais consumo de energia ; mais dissipação de calor
2. Colocar duas CPUs em um chip?
 - Duplicar CPU inteira: ~ duplica área, aumenta custo
 - Compartilhar cache: cai desempenho
3. Adicionar mais unidades funcionais?
 - Manter equilíbrio é difícil: precisa garantir que consegue encher todas as unidades ao mesmo tempo
4. Aumentar comprimento do pipeline?
 - Dependências, desvios, latência de acesso a memória
 - reduzir tempo por estágio: reduzir clock: mais energia
5. **Multithreading**

Hyperthreading no Pentium 4

Compartilhamento de recursos entre threads na microarquitetura NetBurst Pentium 4.

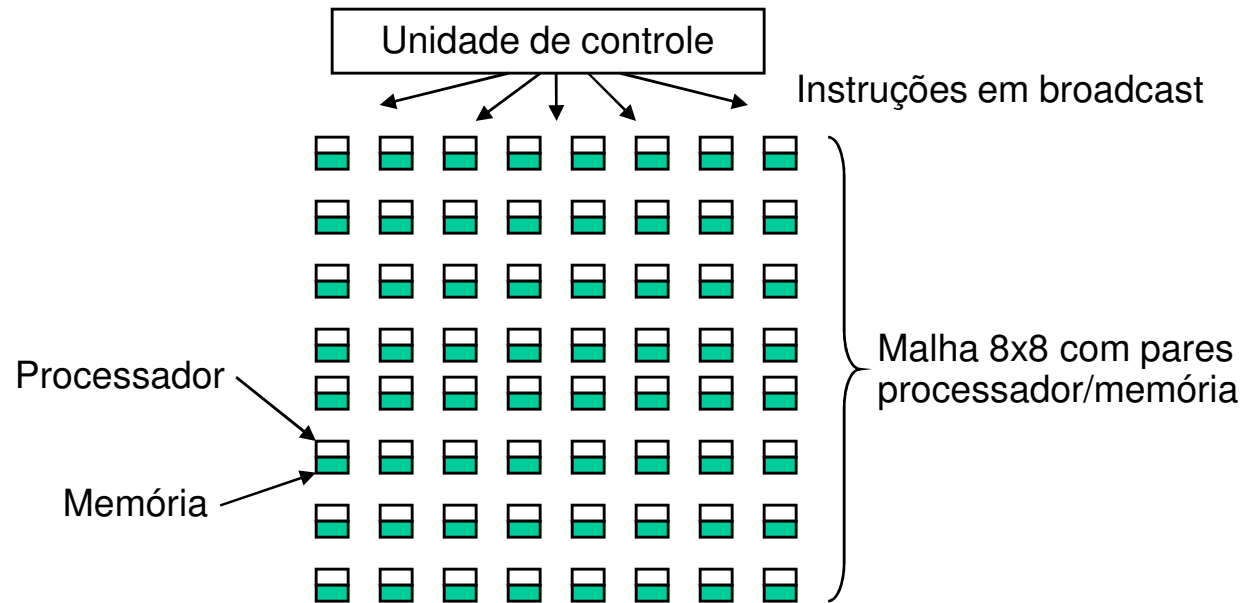
Para o sistema operacional é como se existissem 2 CPUs (virtuais) compartilhando cache e memória principal
Gerenciamento de compartilhamento de recursos



Paralelismo a nível de processador

- Dependências entre seqüências em um pipeline exigem hardware para resolvê-las
- Outro uso para esse hardware é a criação de outras CPUs completas independentes
- Pipeline/super-escalar: ganho fator de 5 ou 10 (max)
 - Ganhos maiores: múltiplas CPUq
- Interação entre CPUs
 - Memória compartilhada: multi-processador
 - Através de mensagens: multi-computador

Computadores matriciais (ILLIAC IV)



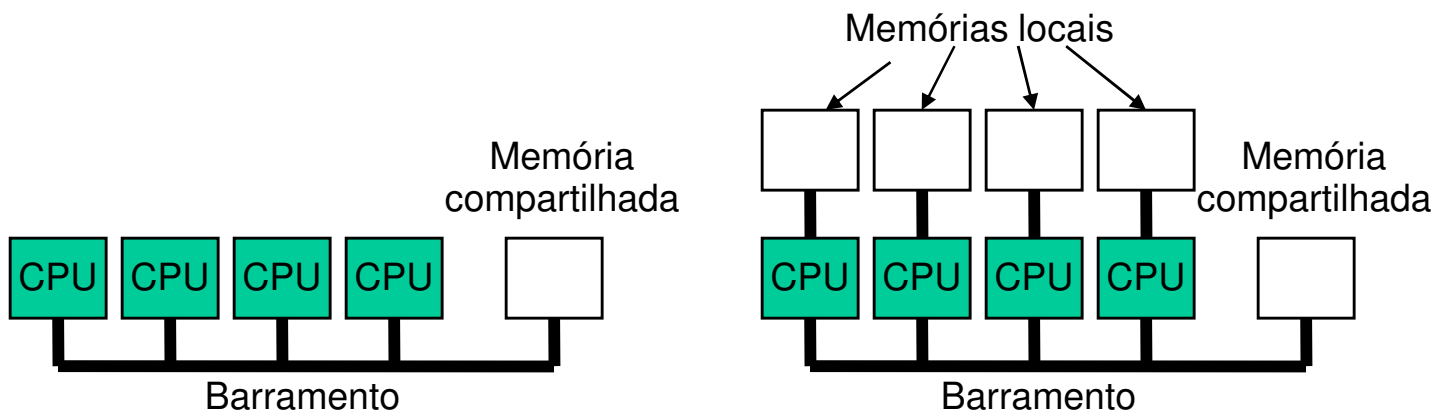
- Uma unidade de controle
- Uma matriz de unidades de execução
- Memória associada à execução

Computadores vetoriais (Cray 1)

- Todas as operações são executadas e uma única unidade de execução
 - Alto grau de paralelismo (pipeline)
 - Registrador vetorial
- Extremamente eficientes para processamento numérico (matricial)
- Unidades vetoriais podem ser acopladas a computadores convencionais

Multi-processadores

- Várias **CPUs** independentes em uma mesma máquina (mesmo chip ou não)
- Memória totalmente compartilhada ou parcialmente distribuída
- Ligação normalmente por barramento



Multi-computadores

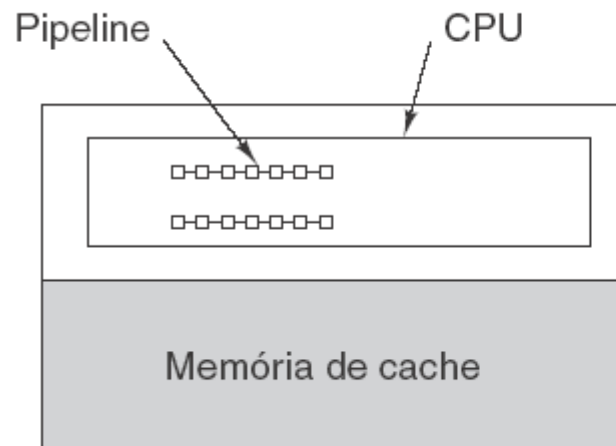
- Unidades independentes com uma rede de interconexão rápida
- Maior grau de paralelismo possível
 - Evita gargalo da memória compartilhada (barramento)
- Várias formas comerciais
 - Sistemas completos (Intel Paragon, Intel Red)
 - Sistemas pré-integrados (IBM SP)
 - Integração local (*clusters* de estações)
 - Integração fraca (sistemas distribuídos em rede)

Múltiplos processadores em um único chip

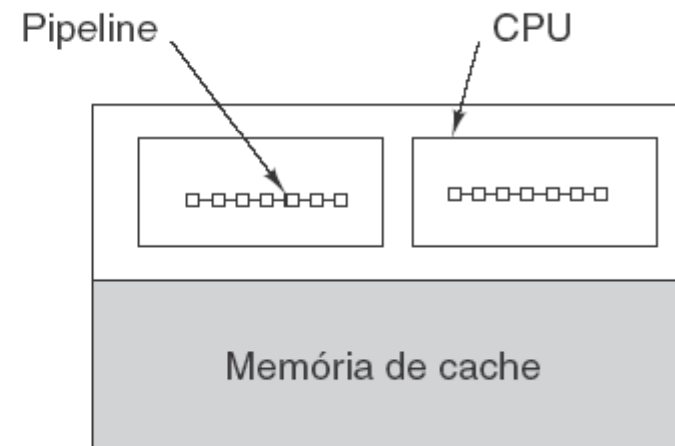
Multiprocessadores com um único chip.

(a) Chip com pipeline dual.

(b) Chip com dois núcleos.



(a)

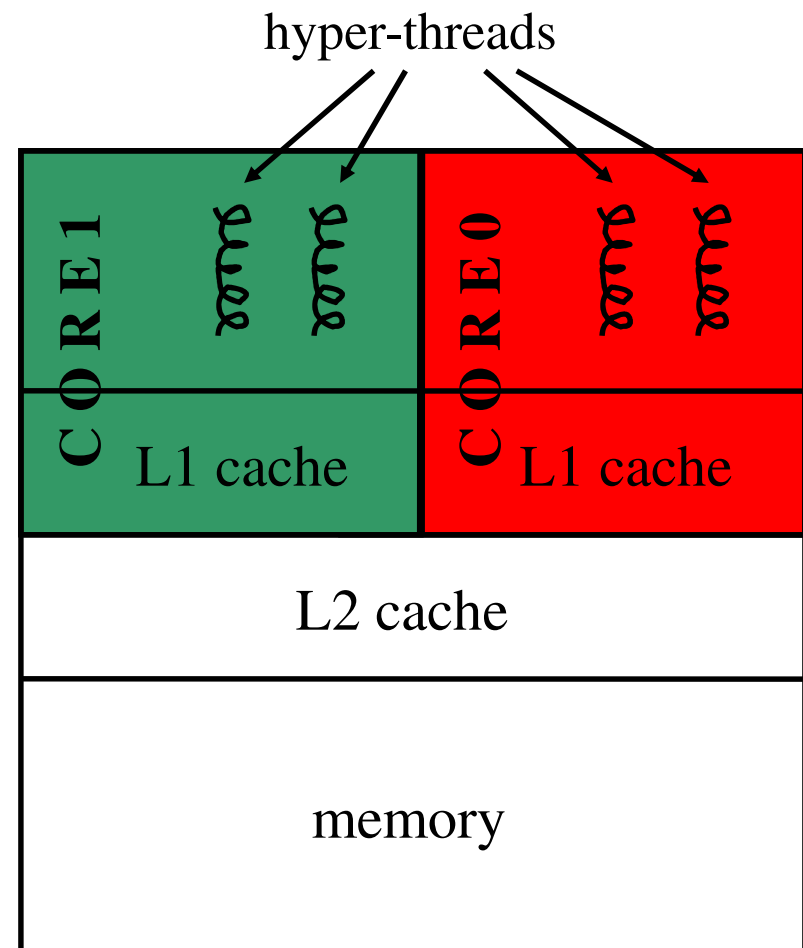
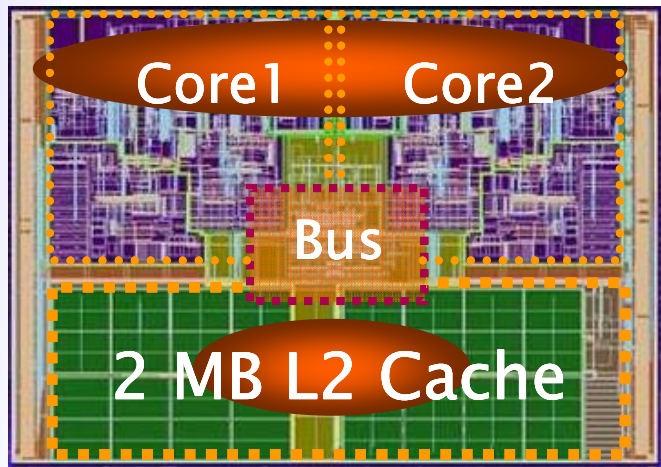


(b)

- Também são possíveis multi-computadores em um chip
- Rede de interconexão extremamente rápida

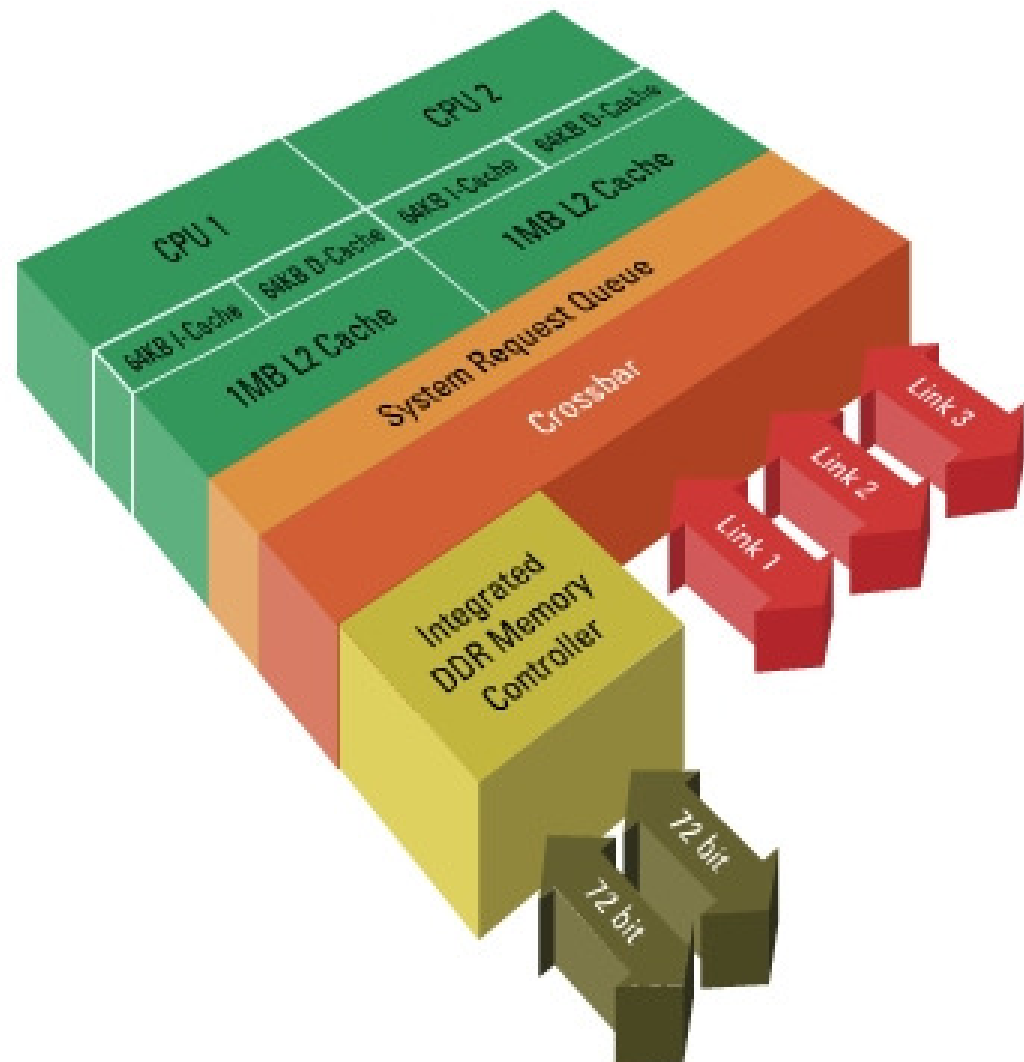
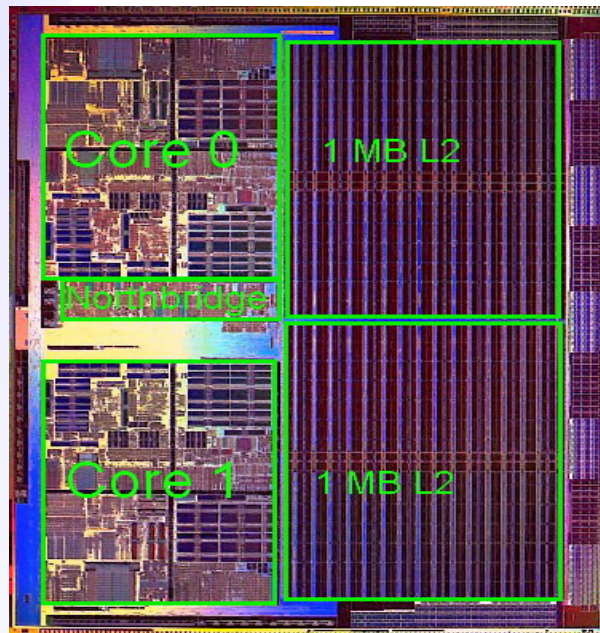
Múltiplos processadores em um único chip

- Intel Core2



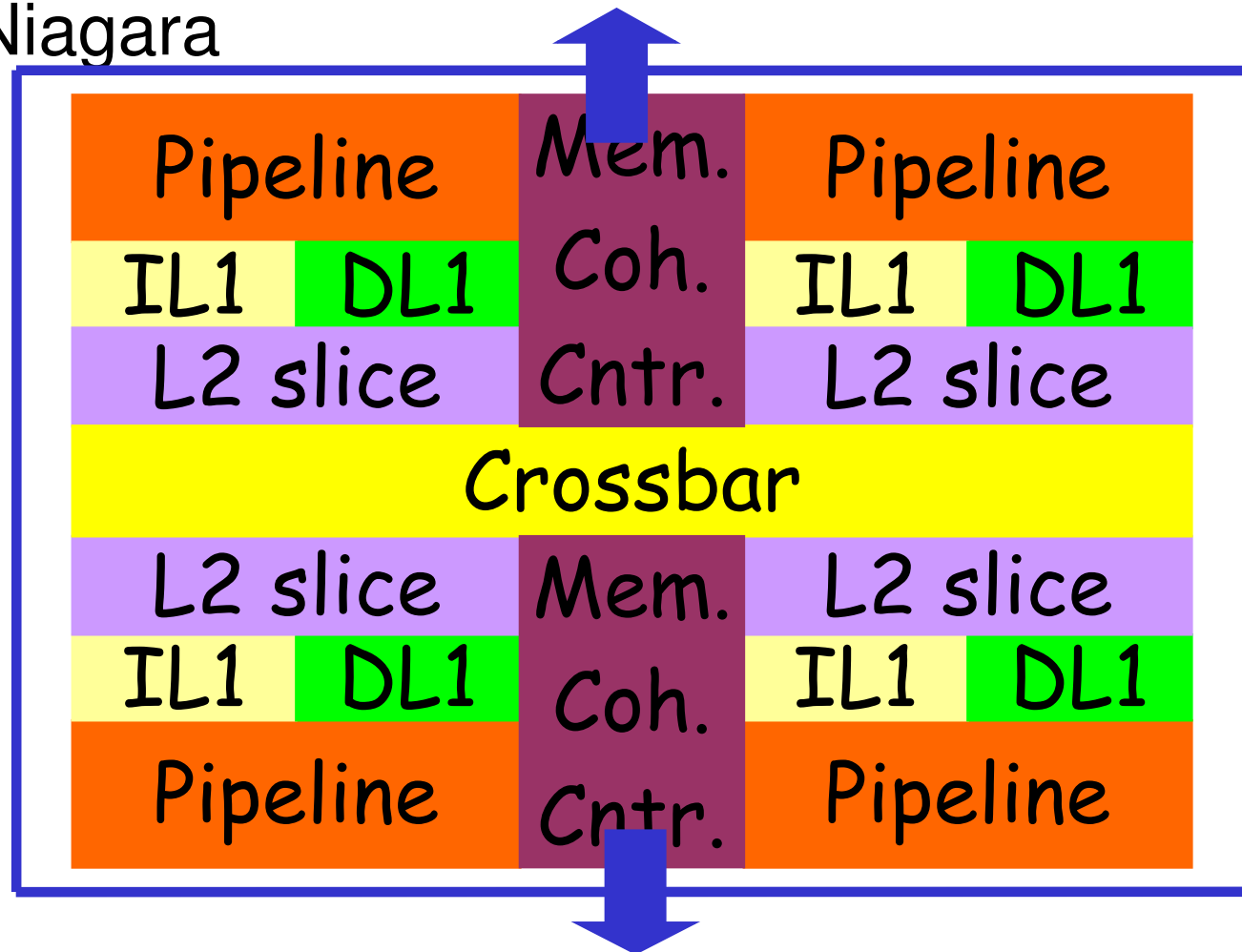
Múltiplos processadores em um único chip

- AMD Opteron



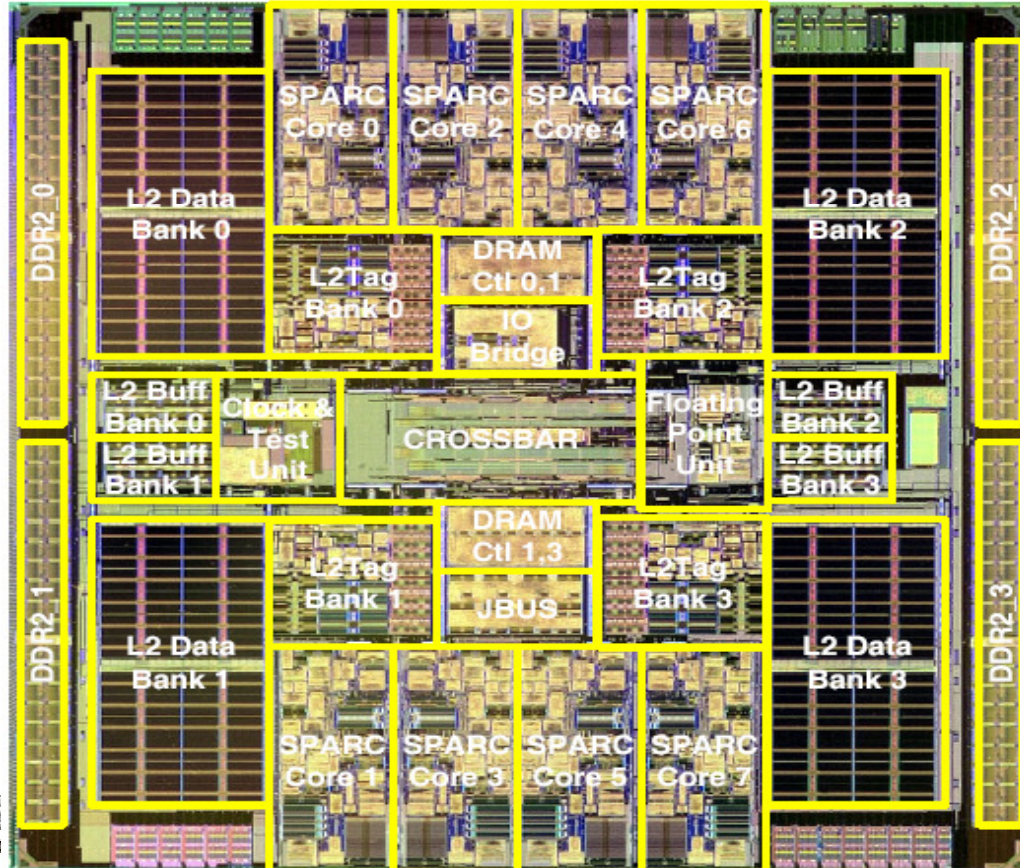
Múltiplos processadores em um único chip

- Sun Niagara



Múltiplos processadores em um único chip

- Sun Niagara



Features:

- Eight 64b Multithreaded SPARC Cores
- Shared 3MB L2 Cache
- 16KB ICache per Core
- 8KB DCache per Core
- Four 144b DDR-2 DRAM Interfaces (400 MTs)
- 3.2GB/s JBUS I/O
- Crypto: Public Key (RSA)
- Extensive RAS

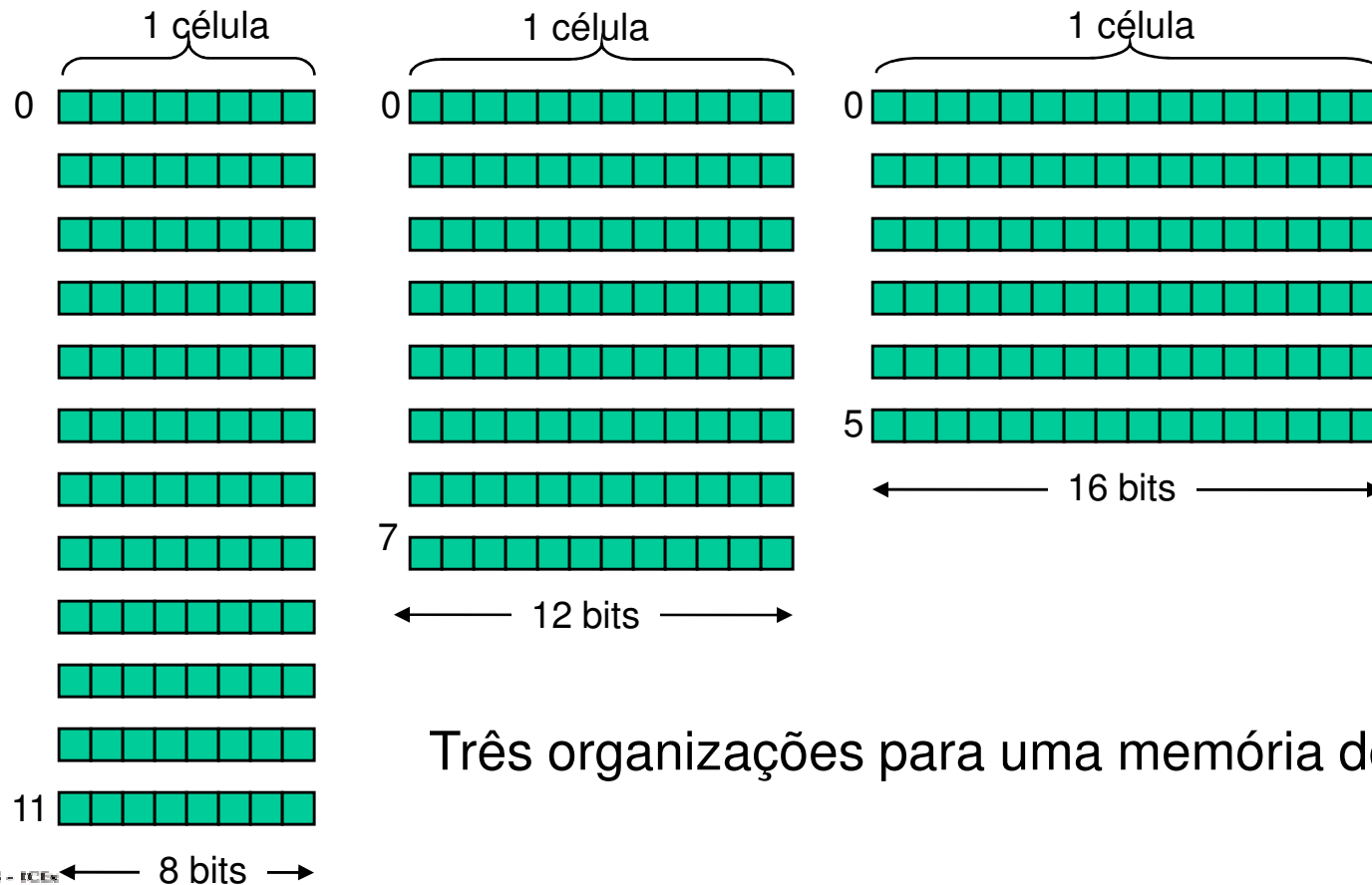
Technology:

- 90nm CMOS Process
- 9LM Copper Interconnect
- Power: 63 Watts @ 1.2GHz
- Die Size: 378mm²
- 279M Transistors
- Package: Flip-chip ceramic LGA (1933 pins)

Memória

- Unidade mínima: bits (0's e 1's)
- Unidade endereçável: células
- Cada célula armazena uma palavra (valor em binário)
- Endereços: identificação de cada palavra
- Usualmente células são bytes
- Capacidade da memória depende:
 - do número de células (espaço de endereços)
 - do tamanho de cada célula (largura da palavra)

Possíveis organizações de memória



Três organizações para uma memória de 96 bits

Possíveis organizações de memória

- Tamanho da palavra para alguns computadores comerciais históricos

Computador	Bits/célula
Burroughs B1700	1
IBM PC	8
DEC PDP-8	12
IBM 1130	16
DEC PDP-15	18
XDS 940	24
Electrologica X8	27
XDS Sigma 9	32
Honeywell 6180	36
CDC 3600	48
CDC Cyber	60

Ordem de bytes

- Ao representar valores maiores que um byte (palavras) computadores devem escolher a ordem em que os bytes são colocados na memória
- *Strings* (vetores em geral) não sofrem inversão

0003

0002

0001

0000

Memória

?

Dados (p.ex., inteiro)

Byte3

Byte0



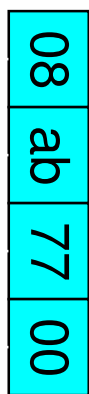
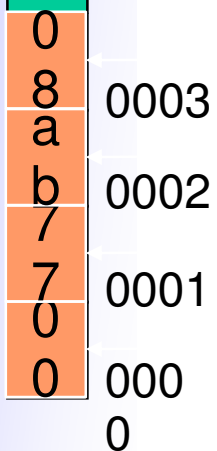
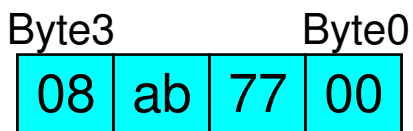
Bit 31

Bit 0



Ordem de bytes

- Considere-se o número 0x08ab7700:

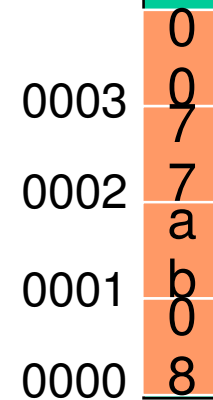
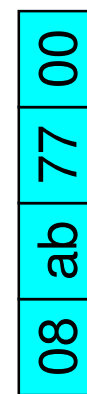


Bit 31

Bit 0

Little endian
(Intel)

Big endian
(Internet)



Detecção e correção de erros

- Lei de Murphy aplicada a redes de dados:

Erros acontecem!!!

- **Detecção** de erros: bits de paridade
- Correção: códigos de *hamming*
 - *informação é suficiente para a recuperação*

Bits de paridade

- Tornam sempre par (ímpar) o número de 1s em um byte
 - Simples, permite detecção de erros individuais
 - Normalmente implementado em hardware
- Paridade par: 01011010 -> 01011010 **0**
- Paridade ímpar: 01011010 -> 01011010 **1**

Códigos corretores de erros

- Vários bits de paridade são acrescentados segundo regras especiais
- Essa redundância extra é tal que certos erros podem ser corrigidos
- Correção de bit único
 - m bits de dados, r bits de redundância
 - n = m+r bits por palavra

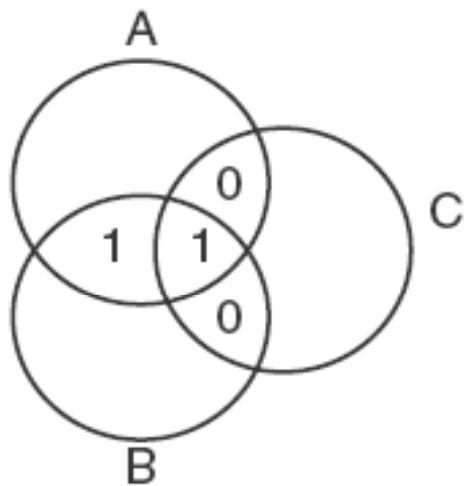
$$2^m(n+1) \leq 2^n \rightarrow (m+r+1) \leq 2^r$$

Códigos corretores de erros

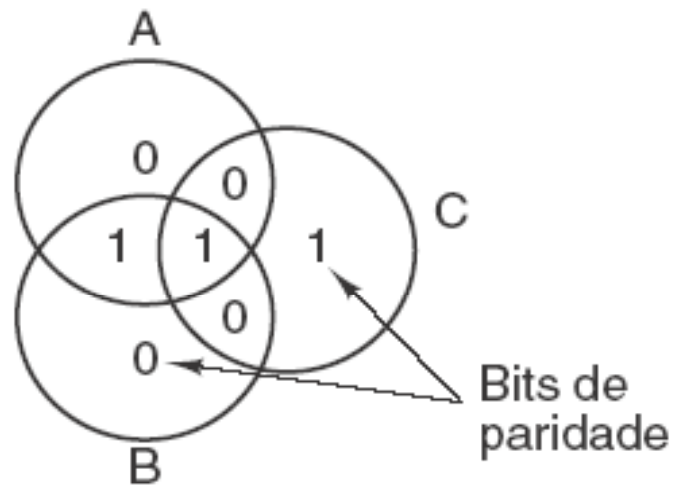
Tamanho da palavra	Bits de verificação	Tamanho total	Sobrecarga percentual
8	4	12	50
16	5	21	31
32	6	38	19
64	7	71	11
128	8	136	6
256	9	265	4
512	10	522	2

Número de bits de verificação para corrigir um único erro.

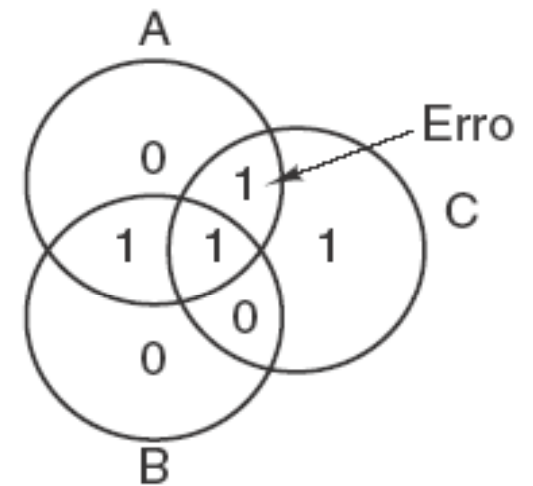
Códigos corretores de erros



(a) Codificação de 1100



(b) Paridade par adicionada



(c) Erro em AC

Código de Hamming

- Numerar bits da palavra resultante (c/ redund.)
- Posições = 2^x são bits de paridade
- Demais são espaços para bits originais
- Bit b controlado por bits de paridade correspondentes a cada componente de k

bit b controlado por bits b_1, b_2, \dots, b_j tal que

$$b = b_1 + b_2 + \dots + b_j$$

Código de Hamming (exemplo)

- Código de Hamming para 1101001

	1	2	3	4	5	6	7	8	9	10	11
	<u>a</u>	<u>b</u>	1	<u>c</u>	1	0	1	<u>d</u>	0	0	1

- a (bit₁) = bit₃ + bit₅ + bit₇ + bit₉ + bit₁₁ = 0
- b (bit₂) = bit₃ + bit₆ + bit₇ + bit₁₀ + bit₁₁ = 1
- c (bit₄) = bit₅ + bit₆ + bit₇ = 0
- d (bit₈) = bit₉ + bit₁₀ + bit₁₁ = 1
- Hamming = 0 1 1 0 1 0 1 1 0 0 1

Código de Hamming

- Correção de erros:
 - Calcula todos os bits de paridade (XOR)
 - Se todos estiverem corretos : palavra sem erro
 - Caso contrário some os bits de paridade com erro
 - O bit equivalente à soma está errado.

Código de Hamming (exemplo)

- Suponha que receba:

0 1 1 0 1 1 1 1 0 0 1

1. Calcula todos os bits de paridade (XOR)

bit 1: $0 + 1 + 1 + 1 + 0 + 1 = 0$: OK

bit 2: $1 + 1 + 1 + 1 + 0 + 1 = 1$: erro

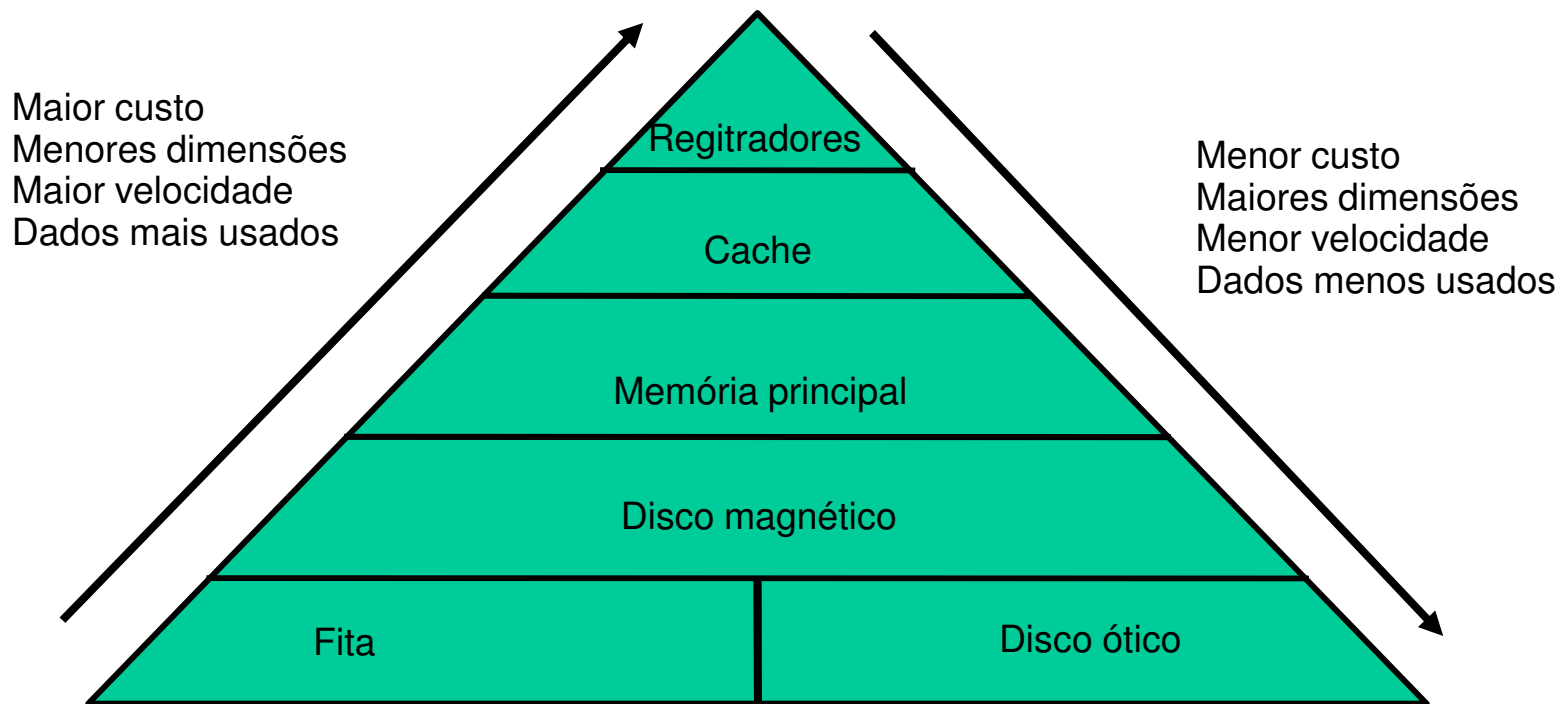
bit 4: $0 + 1 + 1 + 1 = 1$: erro

bit 8: $1 + 0 + 0 + 1 = 0$: OK

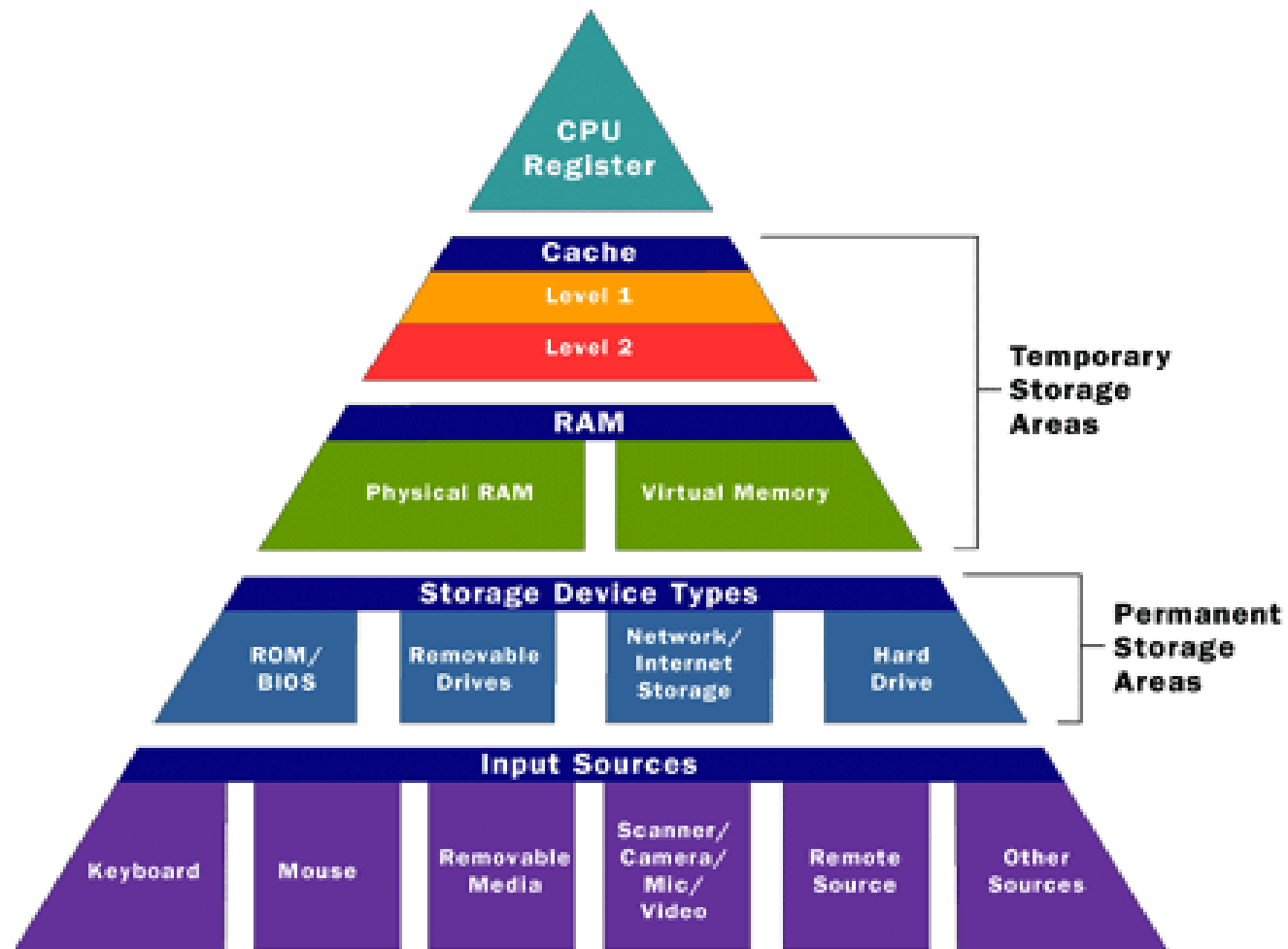
2. Some bits de paridade que deram erro:

$2 + 4 = 6$: bit 6 está com erro: alterar para 0

Hierarquia de memória



Hierarquia de memória



Memória cache

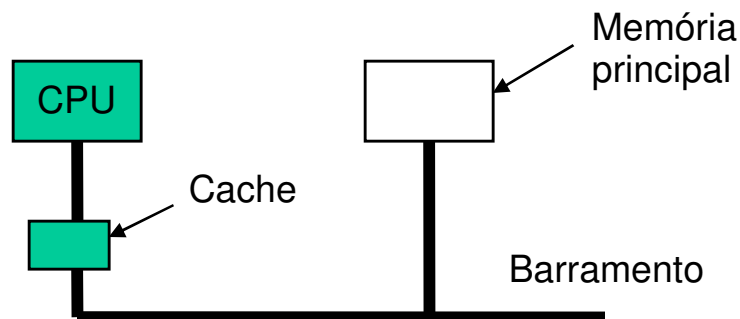
- Processadores hoje são muito mais rápidos que a memória convencional
- Acessos à memória levam várias instruções para se completar
- Se uma instrução precisa do dado lido por outra anterior que ainda não completou:
 - *pipeline* trava até que o dado esteja disponível
 - operação usa valor errado (compilador deve se certificar de que isso não ocorra)

Memória cache

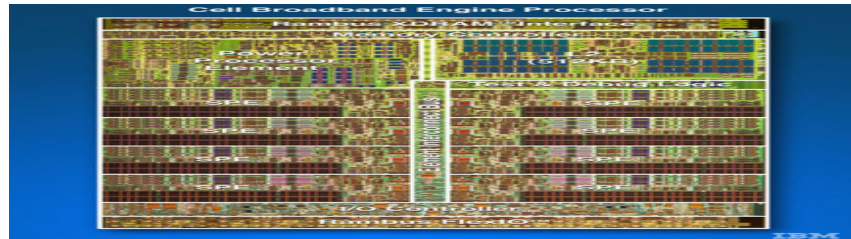
- Tempo de acesso = $c + (1-h) m$
c = custo de acesso a cache, m = tempo acesso à memória
h = cache hit ratio
- Alguns sistemas iniciam referência à memória e busca no cache em paralelo
 - Controle extra pra cancelar acesso à memória em caso de hit
- Caches unificadas X Caches divididas (instrução e dados)
 - Unificadas: bem + simples
 - Divididas: + difundida pela tendência a CPUs com pipeline

Memória cache

- Princípio da localidade:
 - nem toda a memória é utilizada em cada instante
 - acessos à memória tendem a se manter próximos a acessos recentes
- Memória principal (barata) mais lenta
- Memória rápida (pequena) junto à CPU



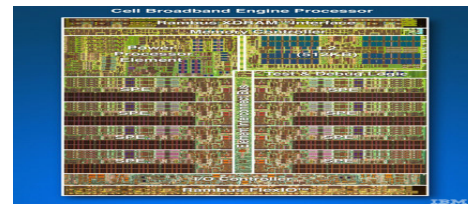
Empacotamento e tipos de memória



single in-line memory module (SIMM)



dual in-line memory module (DIMM)



small outline dual in-line memory module (SODIMM)

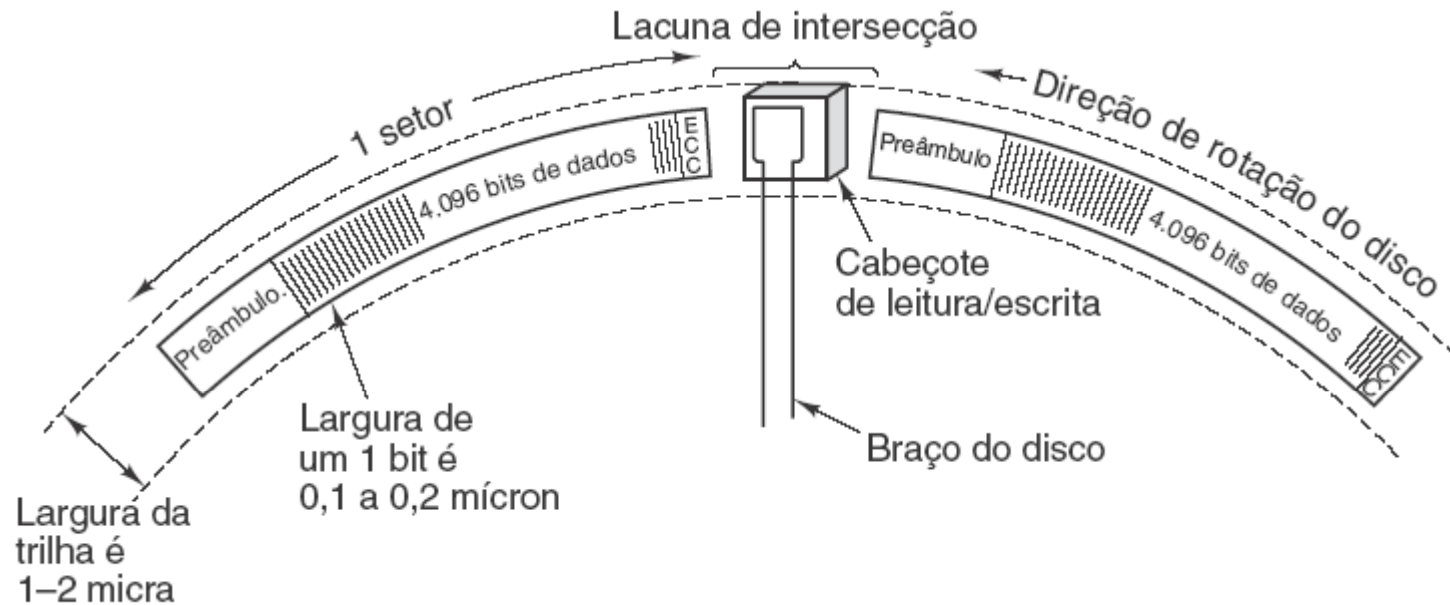
Princípio da localidade pode ser estendido

- Nem toda a memória de um programa precisa ficar em memória principal
 - Memória virtual
- Nem toda a informação precisa estar em disco o tempo todo
 - Sistemas de armazenamento terciário

Discos magnéticos

- Cabeças de gravação sobre meio magnético
- Trilhas não tem marca física (superfície magnetizada)
- Velocidade dos discos cria camada de ar entre superfície e as cabeças
- Tempo de acesso:
 - *seek* - mover da cabeça para a trilha certa ~5-10ms
 - latência rotacional: esperar pelo setor correto ~3-6ms
 - transferir os dados: ~20 us por setor de 512B (40 MB/s)

Discos magnéticos

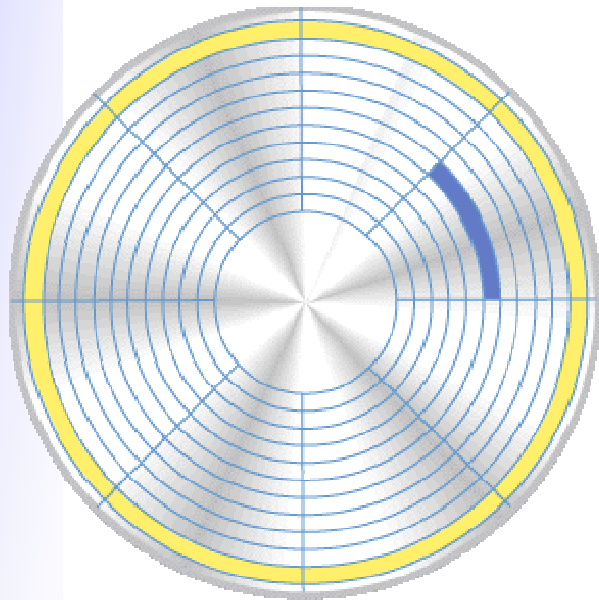


- Discos *Winchester*: alusão à numeração dos primeiros discos IBM (30MB armazenagem fixa e 30MB removível)
- Preâmbulo permite sincronização
- ECC – código pra corrigir múltiplos erros
- Capacidade após formatação ~15% menor

Discos magnéticos

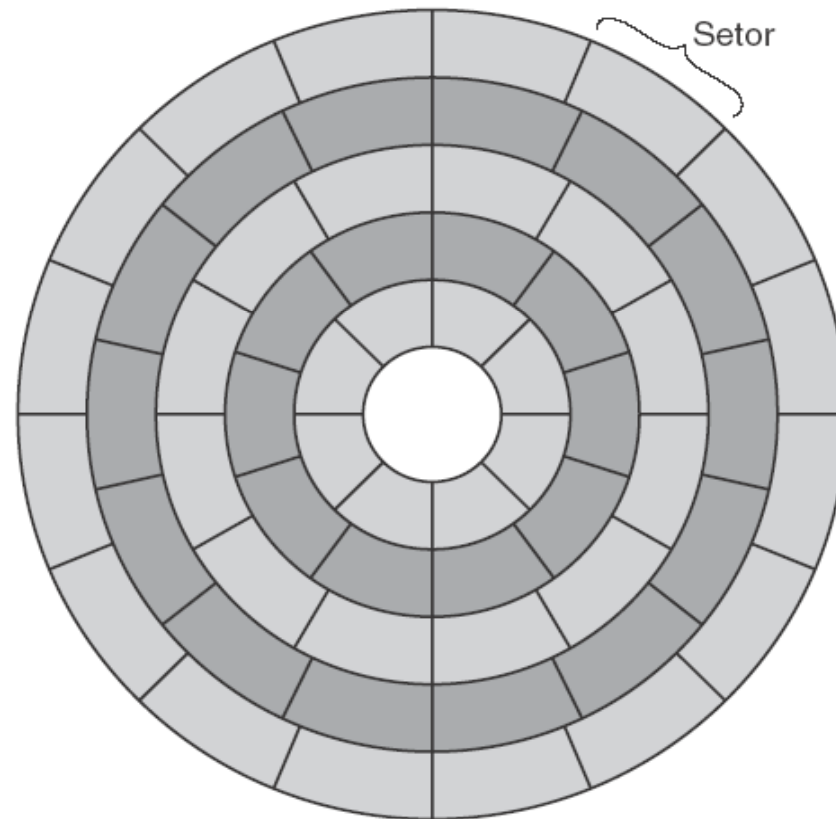
- Distância linear aumenta nas trilhas externas:
- Discos antigos tinham razão #setores/trilha fixa
- Discos modernos separam disco em zonas com razões diferentes
 - Número de setores/trilha aumenta para zonas mais externas
 - Todos os setores são do mesmo tamanho
 - Identificação de localização no disco mais complexa
 - Aumenta a capacidade

Discos Magnéticos



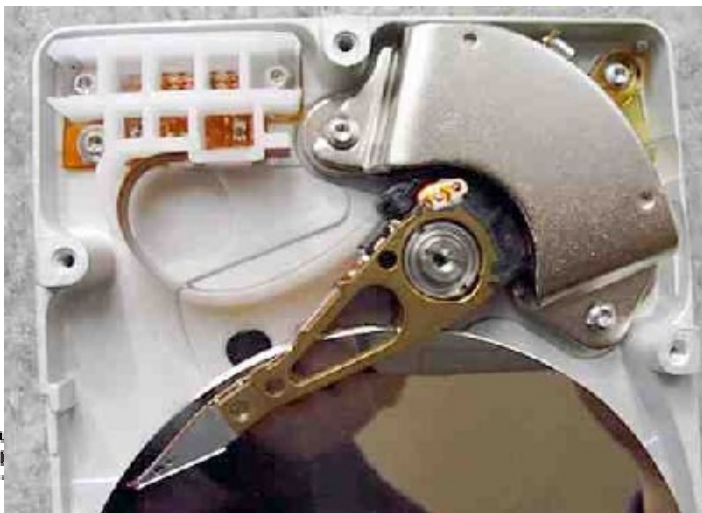
©2000 How Stuff Works

Um disco com uma só zona,
cada trilha com oito setores.



Um disco com cinco zonas.
Cada zona tem muitas trilhas.

Discos magnéticos



Controladores de discos

- Discos antigos eram controlados diretamente pelo processador que determinava # cilindro, # cabeça, # setor
- Tecnologias mais novas dotam o disco de inteligência para mapear endereços lógicos em posições físicas
 - Realizar comandos READ, WRITE, FORMAT
 - detectar e corrigir erros ,etc
 - Cache de setores, rempaear setores ruins.
- Processador/S.O. podem ignorar detalhes dos discos se o controlador é poderoso

Discos IDE

- Interface baseada na BIOS IBM PC XT
 - Herdou endereçamento setores (4 bits/cabeçote, 6 bits/setor e 10 bits/cilindro)
- Controlador junto ao dispositivo (Integrated Drive Electronics)
- Barramento EIDE: 28 bits por setor (até 128 GB)
- Barramento ATA (*PC/AT Attachment*)
 - ATAPI-6: 48 bits /setor ($2^{48} * 2^9 = 128$ PB), 100MB/s
- Ainda usa convenções da BIOS (PC XT)
- Barramento simples entre processador e controladores de disco
- Custo reduzido, desempenho idem

Discos SATA

- Nova geração do barramento ATA (serial)
 - 1 bit por vez a $\geq 150\text{MB/s}$
- Interface mais simples
- Sinalização diferente: consome menos energia
- Controladores mais poderosos



• Maior velocidade de transferência

Discos SCSI – Small Computer System Interface

- Barramento padrão de alto desempenho
 - Disco default para servidores e estações de trabalho
- Controladores complexos liberam o processador da maior parte do acesso
- Permitem a operação de vários dispositivos no mesmo barramento, ao mesmo tempo (arbitragem)
- SCSI-1: 8 b, 5 MHz, 5 MB/s
- Wide Ultra2 SCSI: 16 b, 40 MHz, 80 MB/s

Discos SCSI

Nome	Bits de dados	Frequência do barramento (MHz)	MB/s
SCSI-1	8	5	5
Fast SCSI	8	10	10
Wide Fast SCSI	16	10	20
Ultra SCSI	8	20	20
Wide Ultra SCSI	16	20	40
Ultra2 SCSI	8	40	40
Wide Ultra2 SCSI	16	40	80
Ultra3 SCSI	8	80	80
Wide Ultra3 SCSI	16	80	160
Ultra4 SCSI	8	160	160
Wide Ultra4 SCSI	16	160	320

Alguns dos possíveis parâmetros SCSI.

SCSI-I (8bits): 50 fios: 25 terra, 8 dados, 1 paridade, 9 controle, 7 reservados para uso futuro

RAID (Patterson et al., 1988)

- Redundant Array of Inexpensive Disks
 - Paralelismo: + desempenho, + confiabilidade
 - Indústria redefiniu o 'I' para Independent
 - Distribuir os dados ao longo de vários discos, em “tiras” de k setores cada (*stripes*)
 - Acessos a arquivos grandes podem ser feitos em paralelo aos vários discos
 - Processador comanda o controlador RAID
 - Controlador cuida dos detalhes internos
- RAID = caixa preta para software

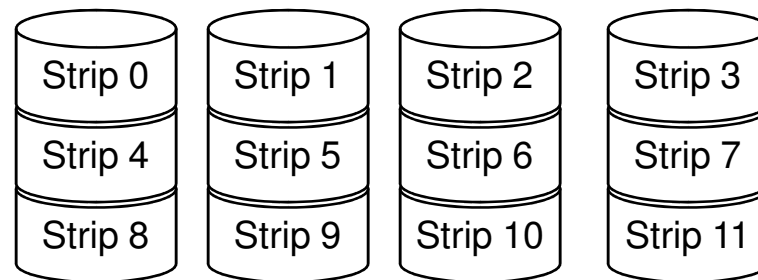
RAID

(Redundant Array of Inexpensive Disks)

- Problema: muitos discos->falhas mais frequentes
- Bons sistemas exigem redundância entre os discos para garantir funcionamento mesmo durante a falha de uma unidade
- Várias organizações possíveis
 - fatias do tamanho de setores ou bits
 - replicação total, bit paridade ou hamming
 - disco isolado para paridade ou paridade deslocada ao longo dos discos

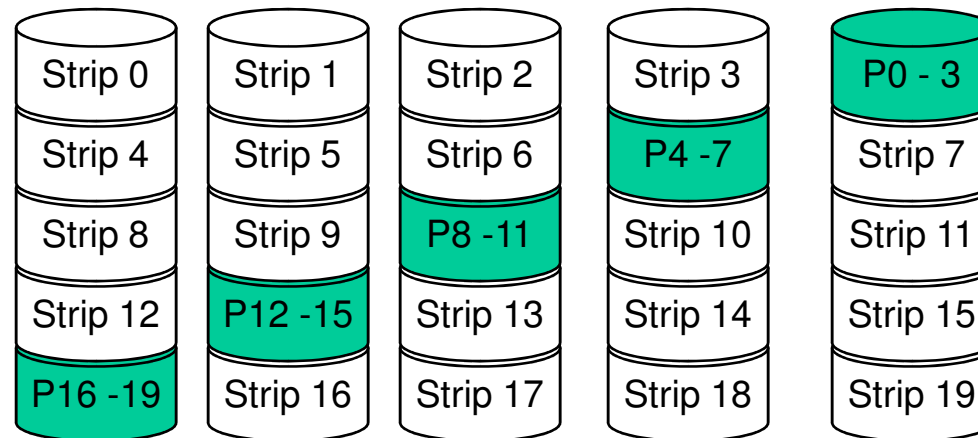
RAID 0 e 5

Distribuição sem redundância



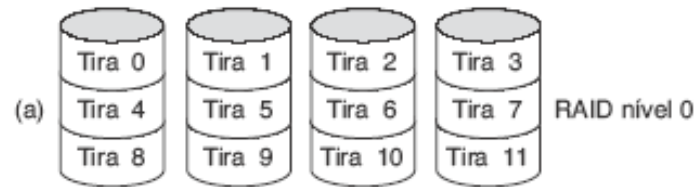
RAID Nível 0

Distribuição com adição de código de correção de erros espalhado pelos vários discos



RAID Nível 5

RAID



Melhor para requisições grandes:
paralelismo

Confiabilidade pode ser pior que 1
disco único

Paralelismo na leitura
Confiabilidade boa
Custo alto



Granulação + fina
Alguns dos bits para
correção (Hamming)
Rotações dos drives tem
que ser sincronizada
Controlador tem que fazer verificação de
Hamming



RAID níveis 0 a 2.

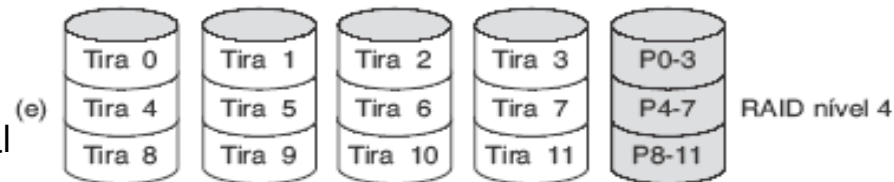
Os discos de backup e de paridade estão sombreados.

RAID

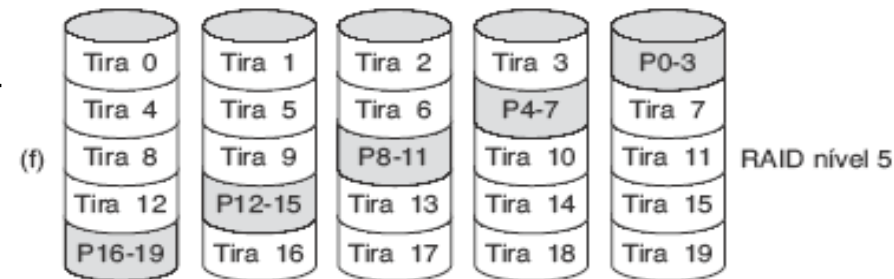
Usa bit paridade
consegue corrigir no caso
de falha no drive
Cada leitura envolve
todos os discos



Ruim para escrita pequena
Drive de paridade: gargalo
potencial



Distribui carga de paridade por
todos os discos



Falha de drive: reconstrução
complexa

RAID níveis 3 a 5.

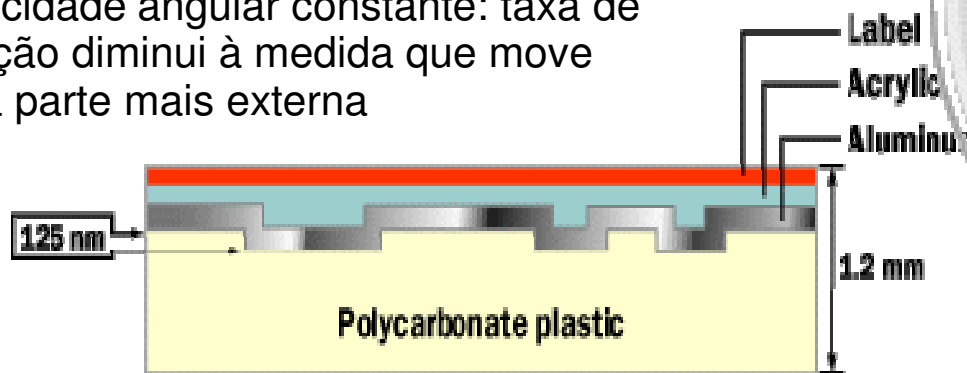
Os discos de backup e de paridade estão sombreados.

CD-ROM

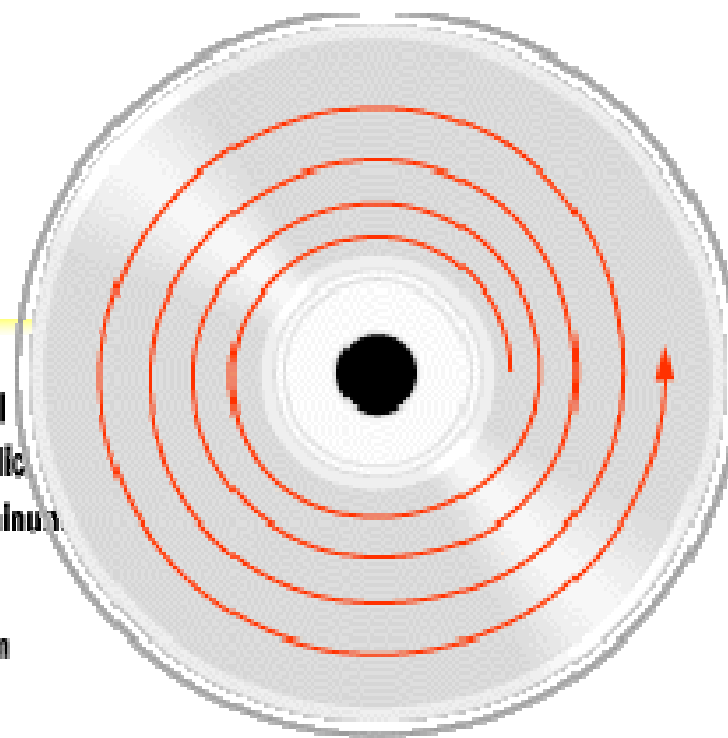
- Padrão emprestado da indústria fonográfica
 - Custo reduzido, complexidade extra
- Bits em espiral contínua
 - Velocidade angular variável
- 0s e 1s como áreas refletivas ou escuras
 - Noção de setores, apesar da espiral
 - Cada byte é escrito como 14 bits (redundância)

CD-ROM

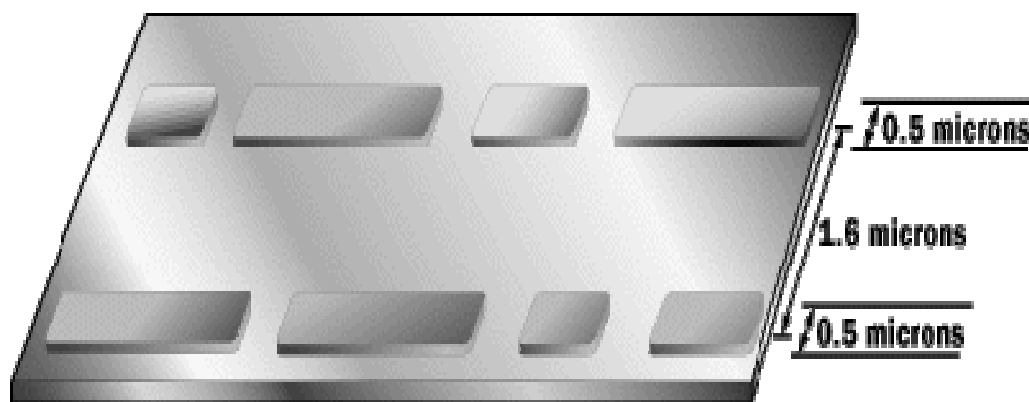
Velocidade angular constante: taxa de rotação diminui à medida que move para parte mais externa



©2000 How Stuff Works



©2000 How Stuff Works



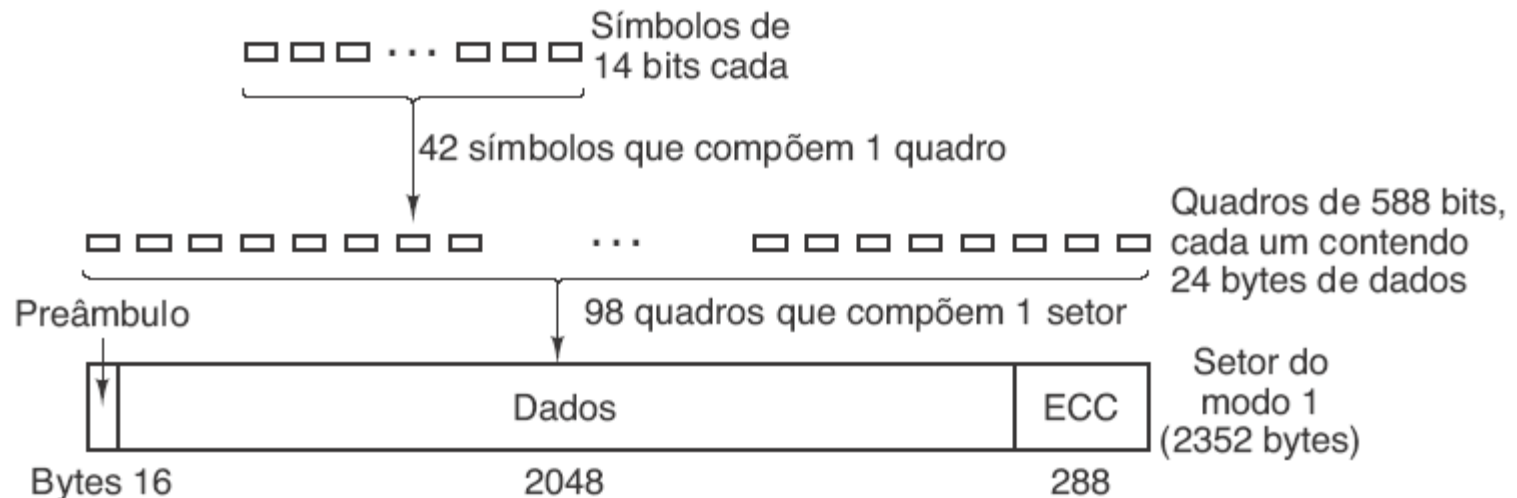
©2000 How Stuff Works

CD-ROM

Formatação de dados : redundância nos vários níveis para correção de erros
(erros de bit único, erro de rajadas e erros residuais)

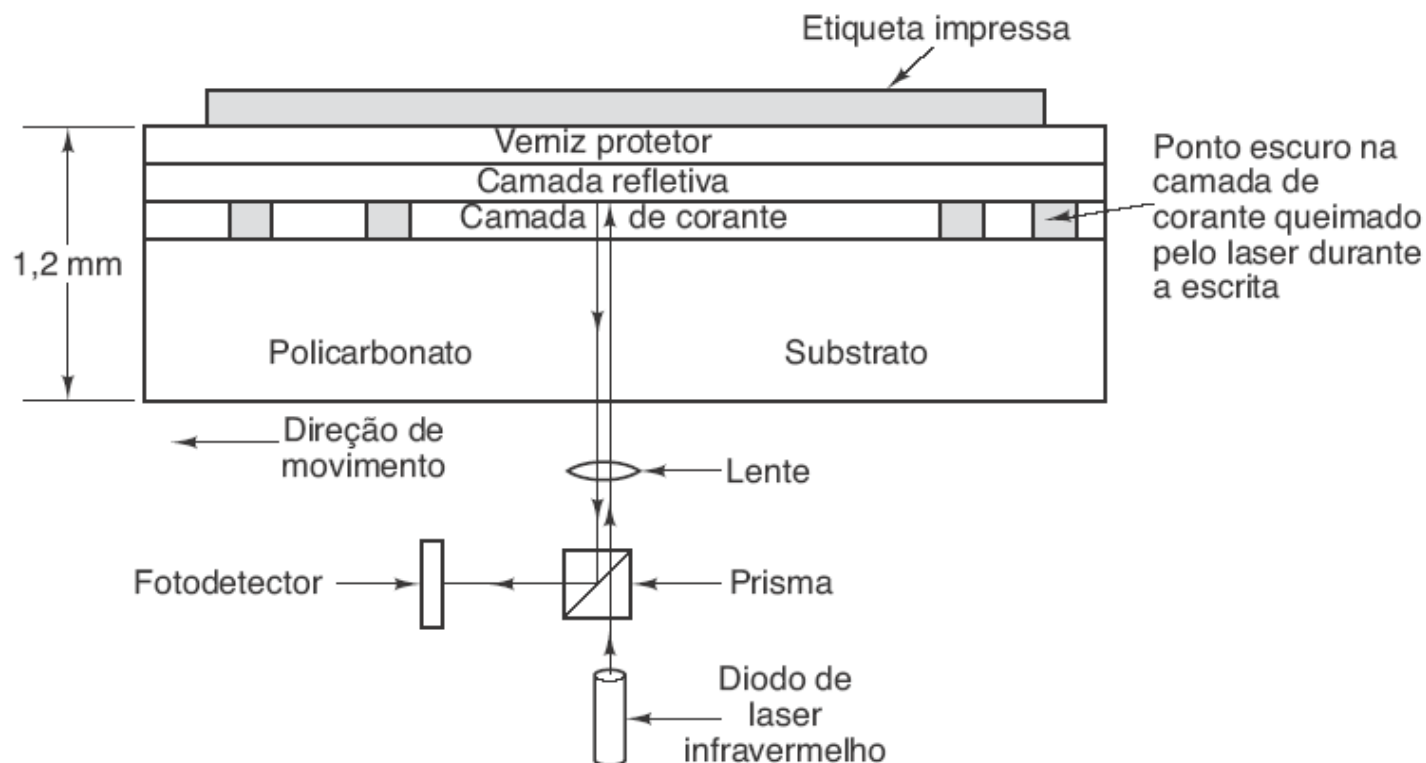
Preâmbulo: marcação de início de setor + número de setor
(ajuda na localização do setor que é muito mais difícil)

1 bit para modo: com ou sem ECC (áudio, vídeo)



CD-R(ecordable)

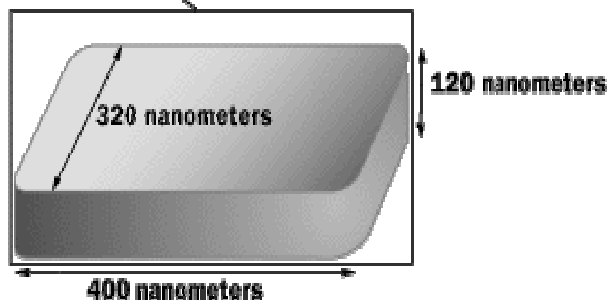
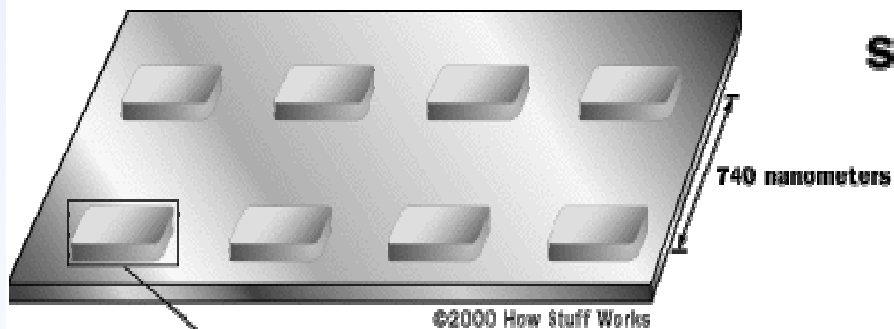
- Laser de alta potência escurece áreas no disco na gravação (depressões são simuladas)
- CD-RW: usa liga de prata (ao invés de corante) com dois estados (cristalino e amorfo) que podem ser controlados por laser



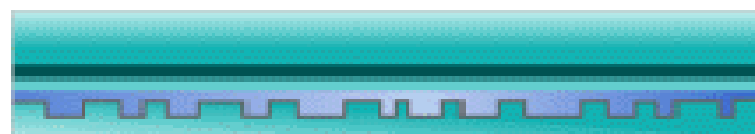
DVD

- Laser mais “fino”
 - 0,65 microns x 0,78 microns (CD)
- Espiral mais apertada
 - 0,74 microns entre trilhas x 1,6 microns (CD)
- Depressões menores
 - 0,4 microns x 0,8 microns (CD)
- Múltiplas camadas
 - 4.7 GB a 17 GB x 650 MB (CD)

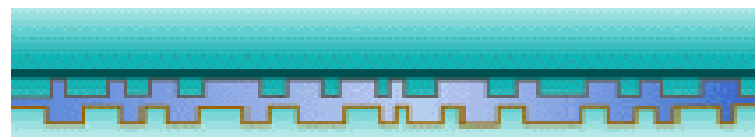
DVD



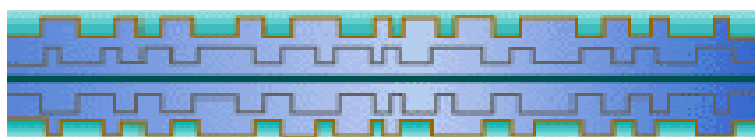
Single-sided, single layer (4.7GB)



Single-sided, double layer (8.5GB)



Double-sided, double layer (17GB)

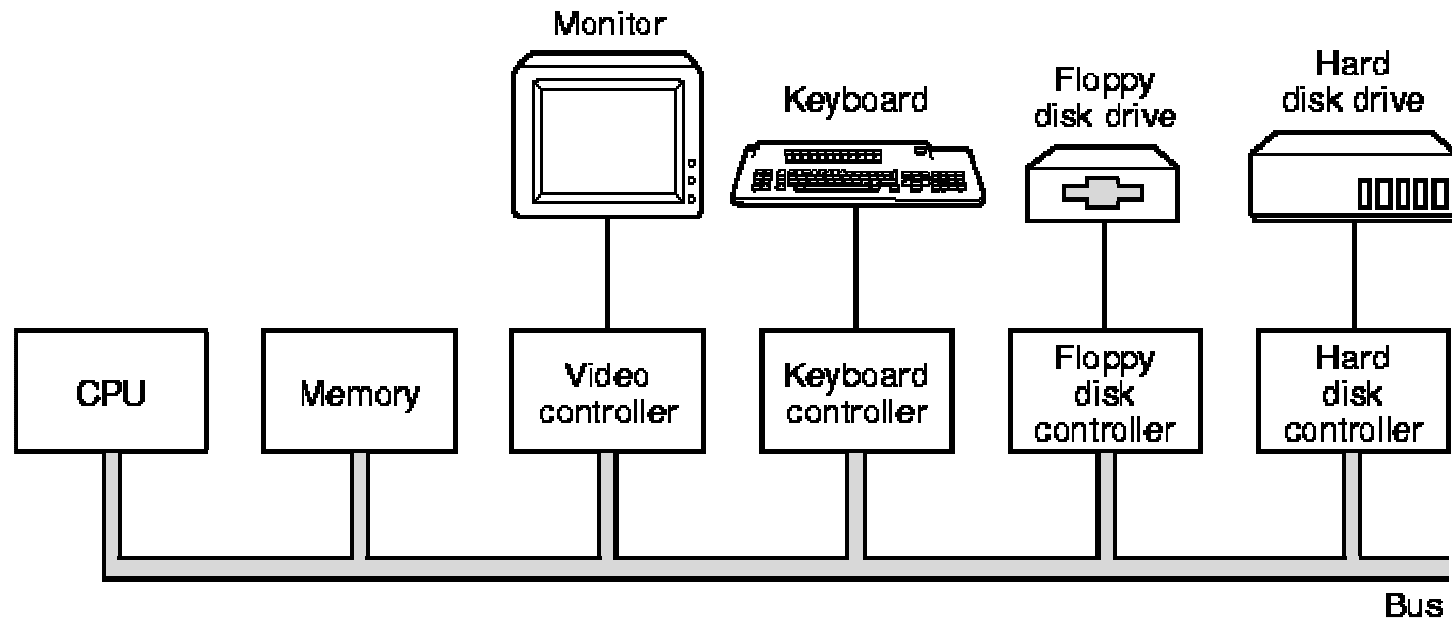


©2000 How Stuff Works

Blu-Ray

- Sucessor do DVD
 - Ganhou a batalha pelo mercado contra o HD-DVD
 - Decisão da Warner-Bros de só usar Blu-Ray
 - PlayStation 3
- Laser azul/violeta (DVD: vermelho)
 - Foco + preciso, depressões menores
- Maior densidade: 25 GB por camada/face

Dispositivos de entrada e saída



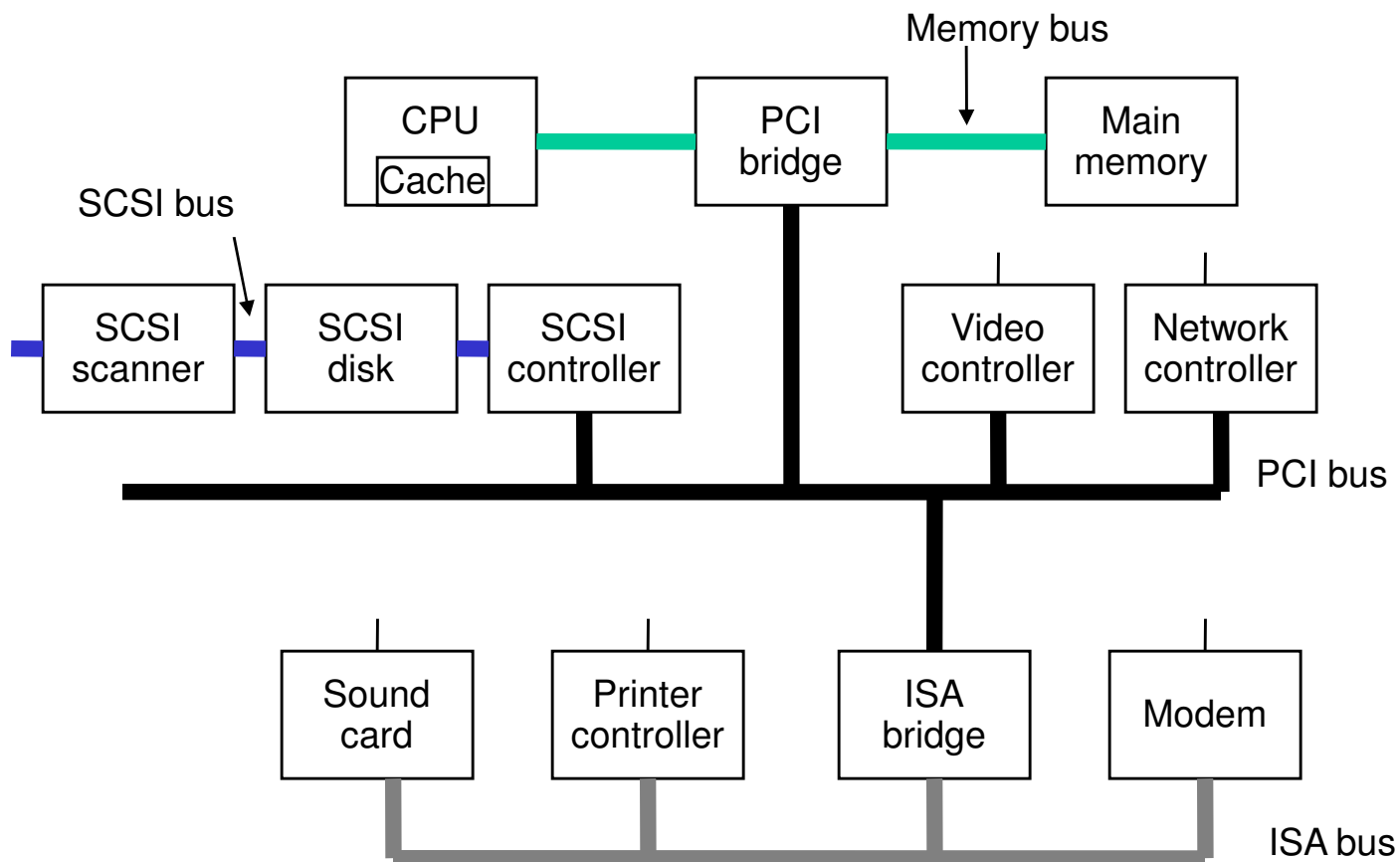
DMA (Direct Memory Access): controlador lê/escreve dados da/na memória diretamente (sem CPU), gerando interrupção ao final

Árbitro de barramento (chip): preferência a E/S sobre CPU (geralmente)

Barramentos de interconexão

- Diferentes custos/velocidades:
 - ISA/EISA (Industry Standard Architecture): muito lento
 - Microchannel (IBM)
 - PCI (Peripheral Component Interconnect): muito popular
 - AGP (Advanced Graphics Peripheral)
 - Aumenta taxa de transferência de *placas de vídeo*
 - USB (Universal Serial Bus)

Barramentos de interconexão



USB (Universal Serial Bus)

- Barramento externo com características de rede de interconexão
- Cada dispositivo tem identificador próprio
- Comunicação se dá através de um protocolo
- Diferentes conectores (USB-A, USB-B, mini)
- 480 Mbps, 5 m (USB2.0)



Firewire (IEEE 1394)



© 2005 HowStuffWorks

- Interconexão para dispositivos externos que necessitem de alta velocidade
 - Discos externos
 - Câmeras digitais,
 - Sistemas profissionais de áudio
- Como USB, basicamente uma rede local
- 800 Mbps, 100 m (v.2)

Memória Flash (NAND Flash)



- EEPROM : Electrical Erasable Programmable Read-Only Memory
 - Mas estrutura orientada a blocos (p.ex: 16 KB)
- Acesso serial à informação
- Mais semelhante a um disco (*Flash disk*)

Teclados

- Contatos identificados por processador no teclado e enviados como coordenada de uma matriz para o computador
- Códigos diferentes para tecla pressionada e liberada
- Cada código gera uma interrupção
 - Tratador da interrupção lê registrador de hardware no controlador de teclado para pegar o número da tecla (1 a 102)
 - Quando tecla é solta: outra interrupção
- Combinação de teclas por software

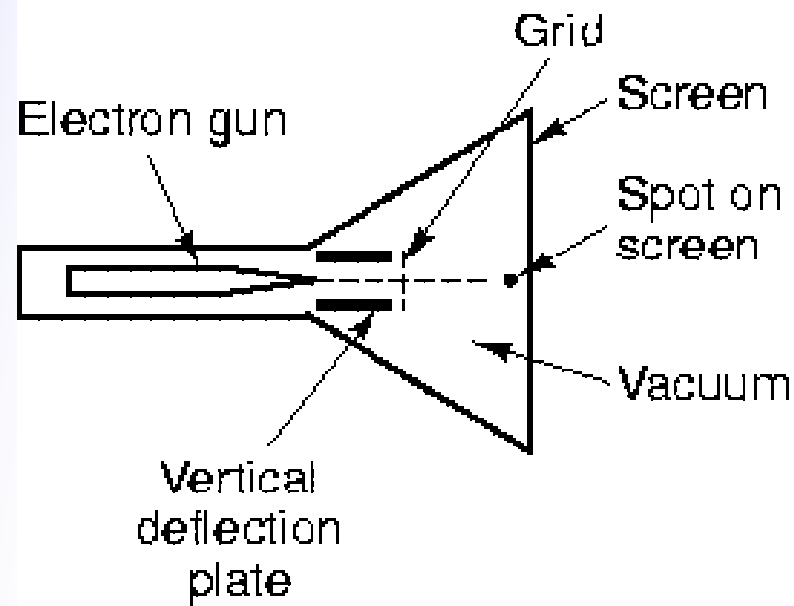
Teclados

Controlador mantém mapa de caracteres em ROM local

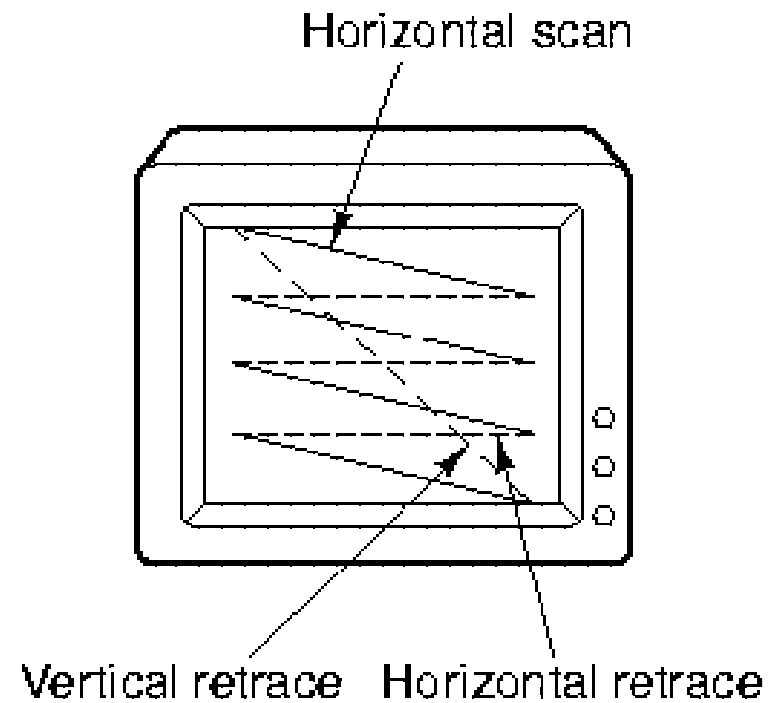


Matriz de teclas

Monitores de vídeo CRT

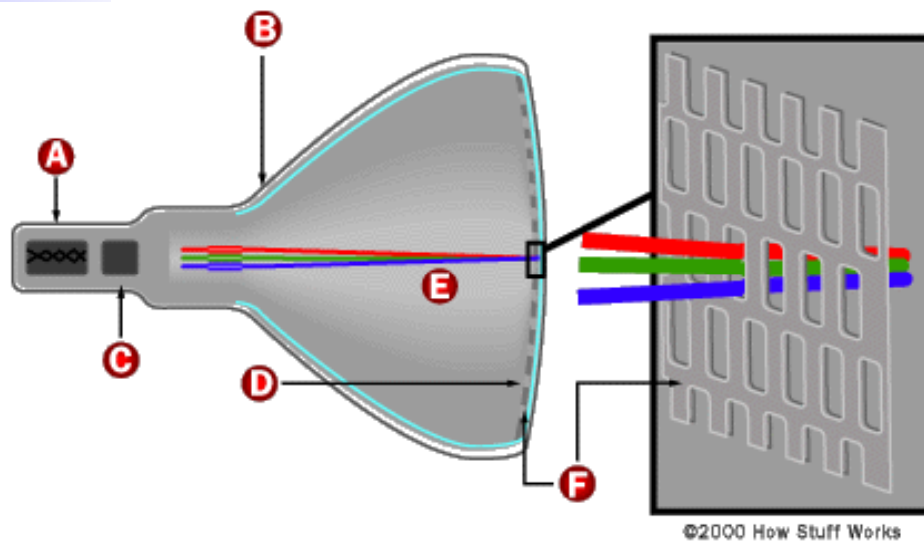


(a)



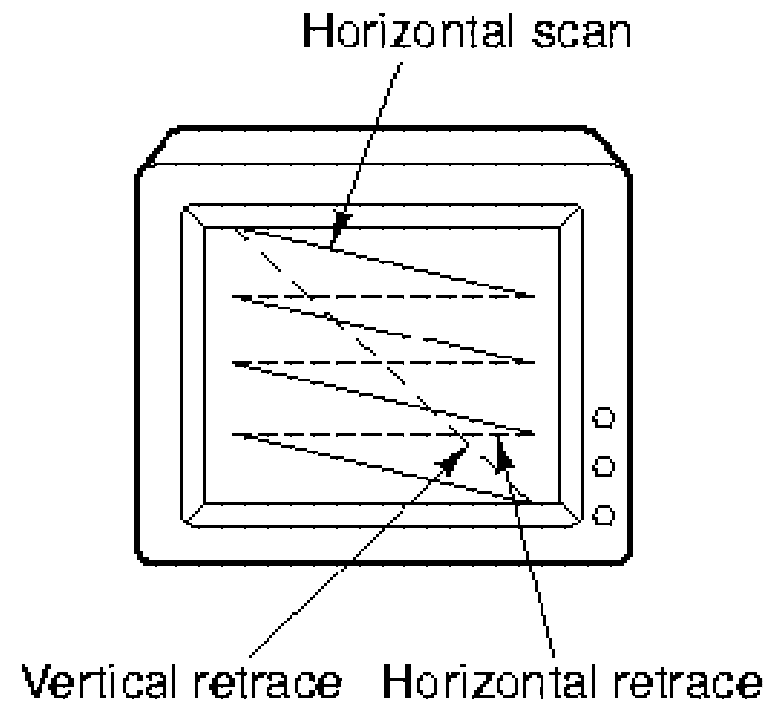
(b)

Monitores de vídeo CRT



- A** Cathode
- B** Conductive coating
- C** Anode
- D** Phosphor-coated screen
- E** Electron beams
- F** Shadow mask

(a)



(b)

Monitores de vídeo CRT

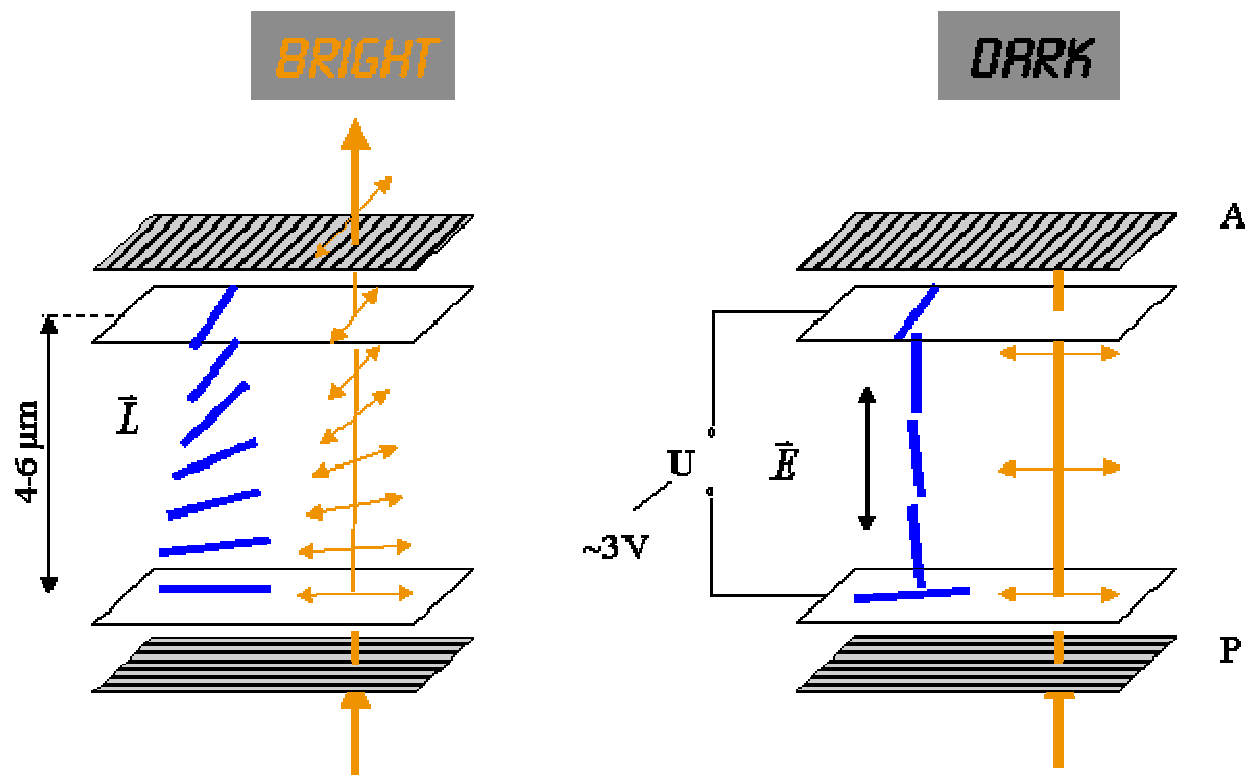
- Monitores modernos são “rasterizados”
- Geração de cores obtida com três canhões independentes que atingem regiões diferentes da tela com fósforo de cores diferentes

Monitores de cristal líquido

- Malhas de condutores horizontais em um lado da tela e verticais no outro
- Sinais são aplicados varrendo cada conector
- Cruzamentos de linhas energizadas polarizam o cristal líquido

Monitores de cristal líquido

Twisted Nematic effect (Schadt-Helfrich effect)



Monitores de cristal líquido

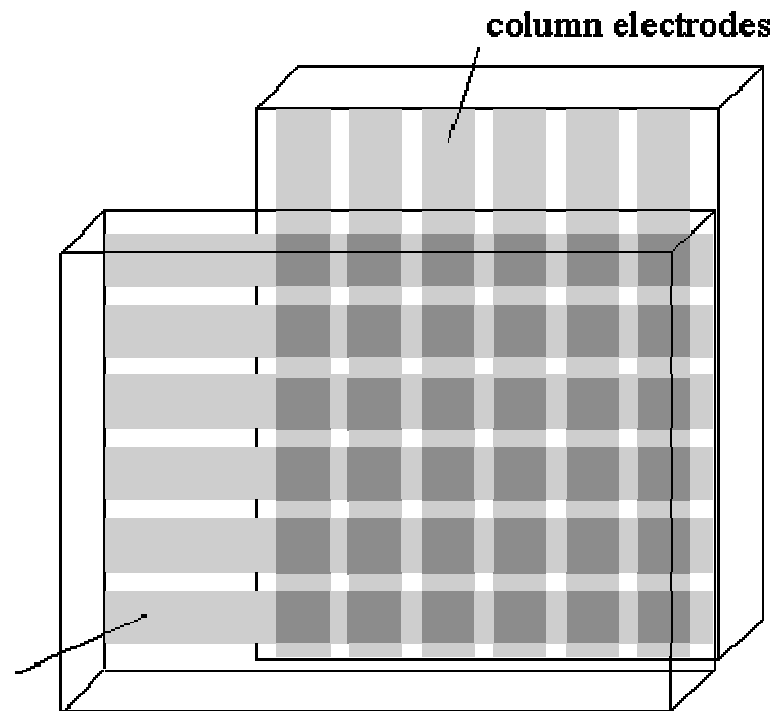
passive matrix displays

reduction in numbers
of connections:

rows \times columns



rows + columns

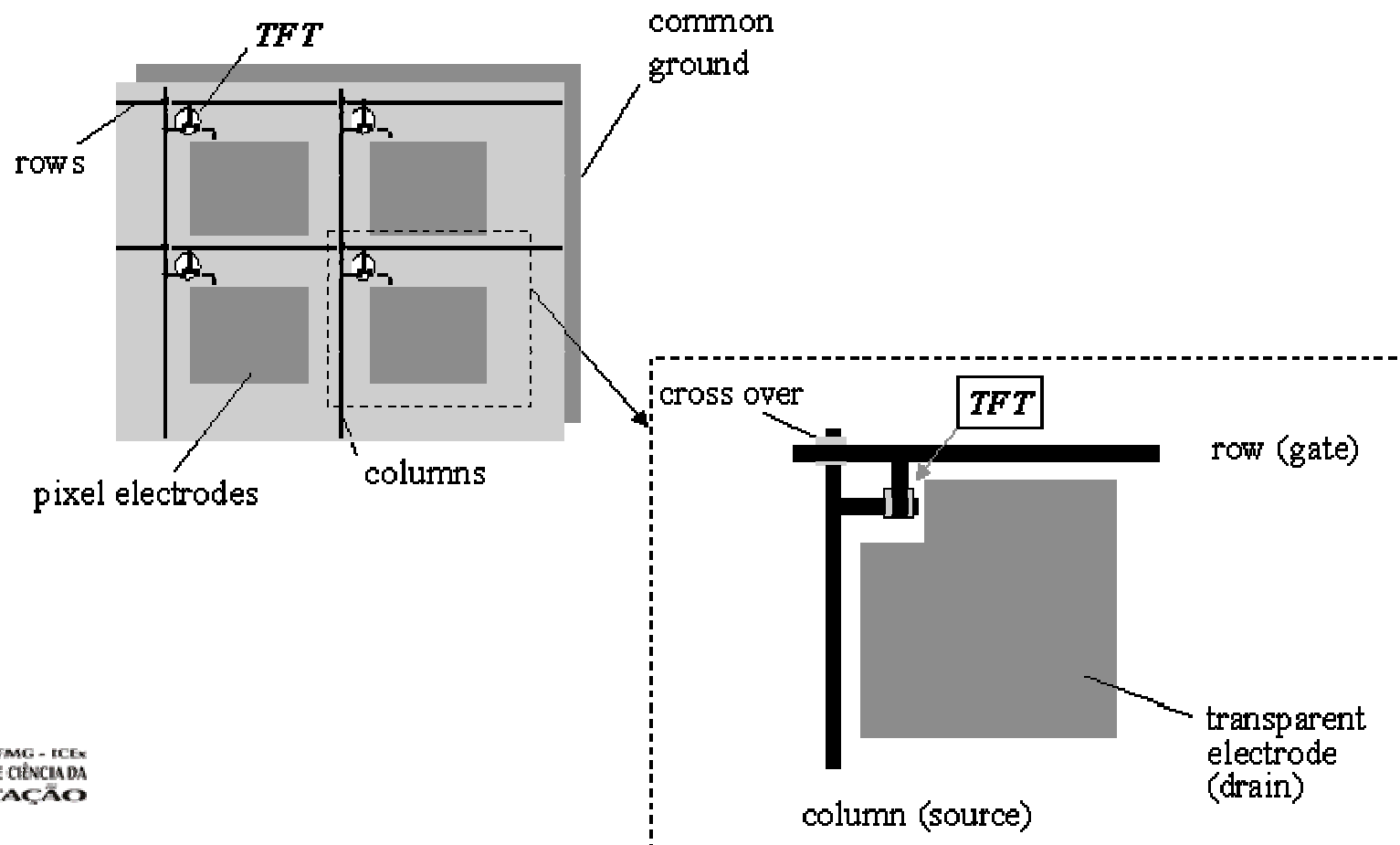


Monitores TFT (thin film transistors)

- Ao invés da malha de condutores, cada ponto da tela contém um transistor
- Um lado tem a base e o coletor do transistor
- O outro contém o emissor
- Chaveando transistores controla-se melhor o campo a tela
- Mesmas opções de iluminação do LCD passivo

Monitores TFT (thin film transistors)

Thin-Film-Transistor or active matrix displays



Mouse

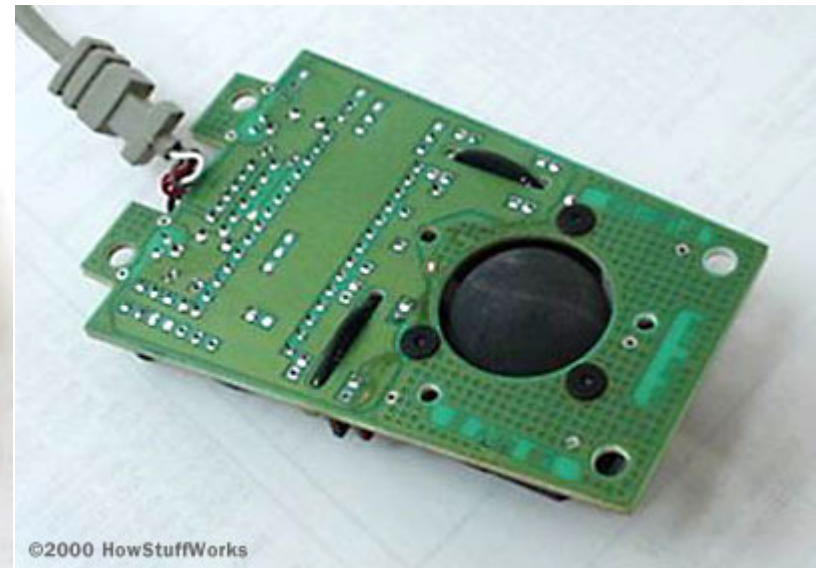
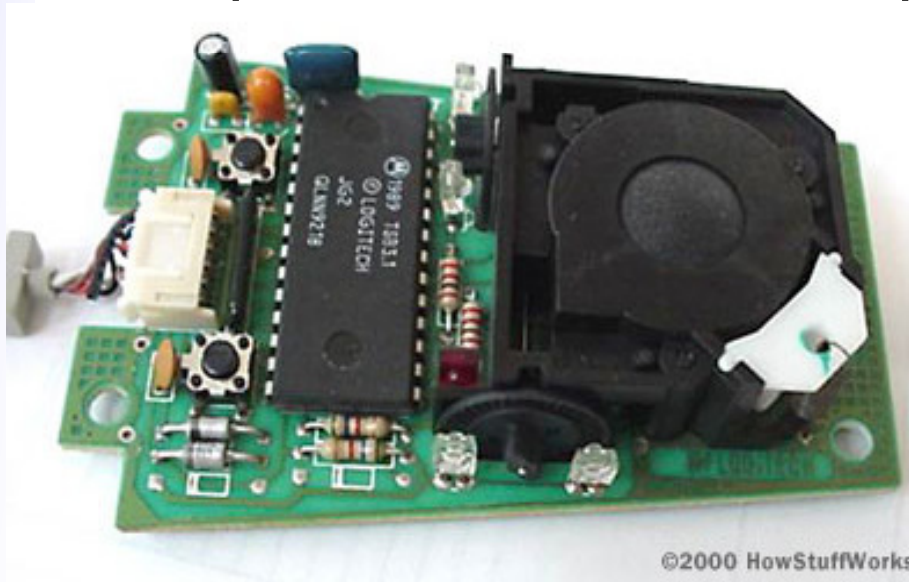
- Mede movimentos sobre uma superfície bi-dimensional (mickeys)
- A cada movimento em uma distância mínima (e.g., 0.01 polegada), envia para o computador 3 bytes:
 - Deslocamento no eixo x
 - Deslocamento no eixo y
 - Estado das teclas do mouse

Mouse

- Dois tipos principais hoje:
 - óptico-mecânico
 - esfera gira dois eixos perpendiculares
 - disco perfurado acoplado a cada eixo
 - par LED/foto-diodo detecta giro dos discos
 - óptico (antigos exigiam pad com padrão especial)
 - LED projeta padrão sobre superfície
 - Grade retangular de linhas espaçadas
 - foto-detector percebe os cruzamentos de linhas pelas alterações na quantidade de luz refletida pelo LED

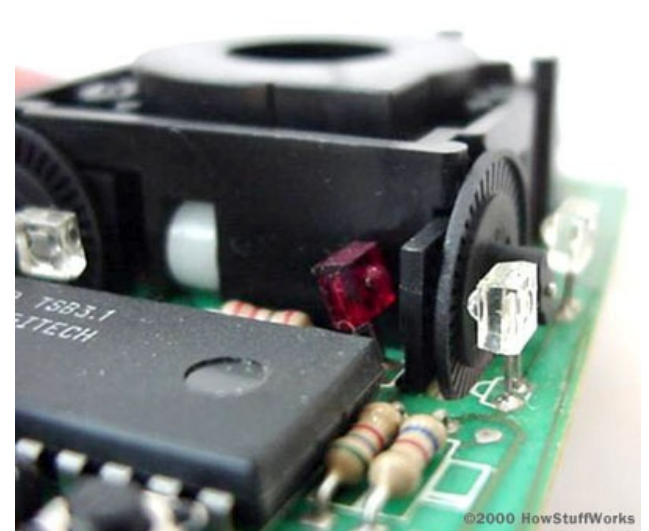
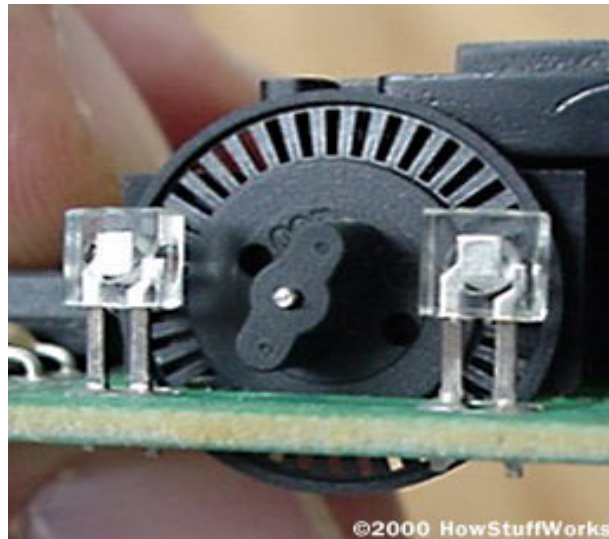
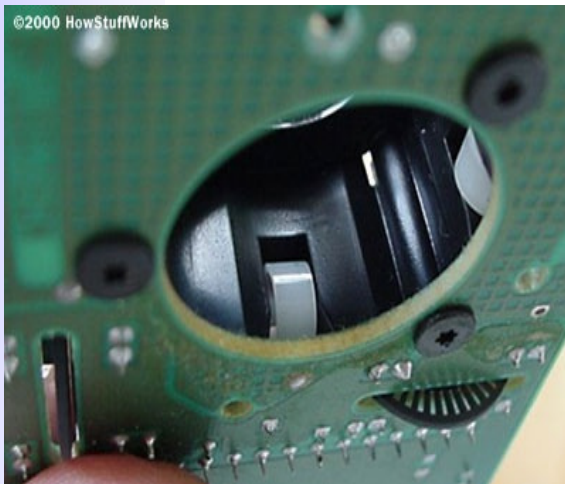
Mouse

- Conversão de movimentos em deslocamentos em duas dimensões
- Optico-mecânico ou óptico



Mouse

- Movim. da bola faz girar discos perfurados
- Pares LED/foto-diodo captam pulsos de luz
- Direção: diferença entre dois pares



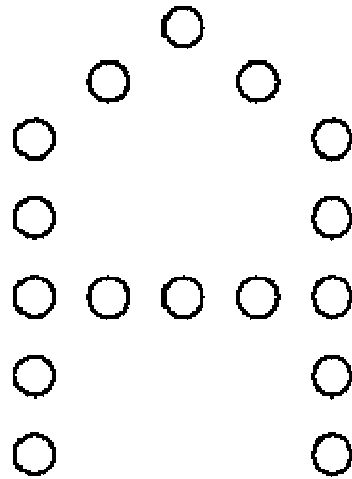
Mouse óptico

- LED vermelho com sensor óptico (câmera)
- Câmera captura milhares de imagens/s
- Hardware de proc. de sinais (DSP) detecta movimento de padrões nas imagens

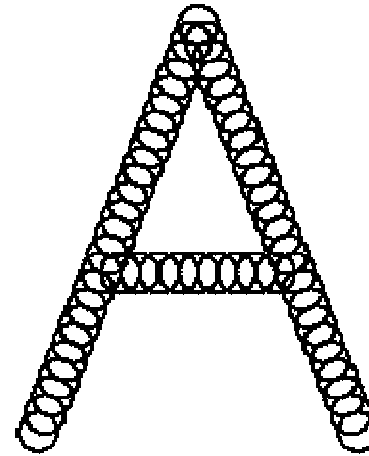


Impressoras

- Mesmo princípio comanda impressoras laser, jato de tinta e matriciais:
- Pontos sobrepostos lançados linha-a-linha

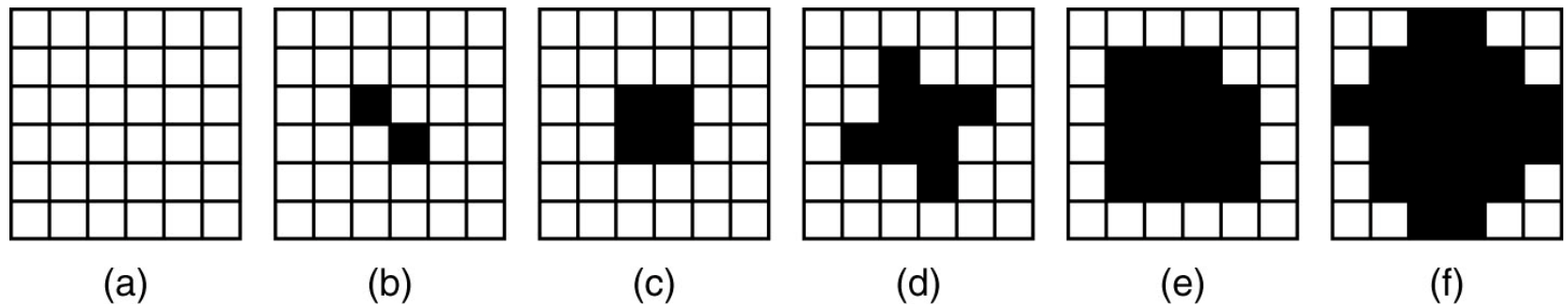


(a)



(b)

Impressoras



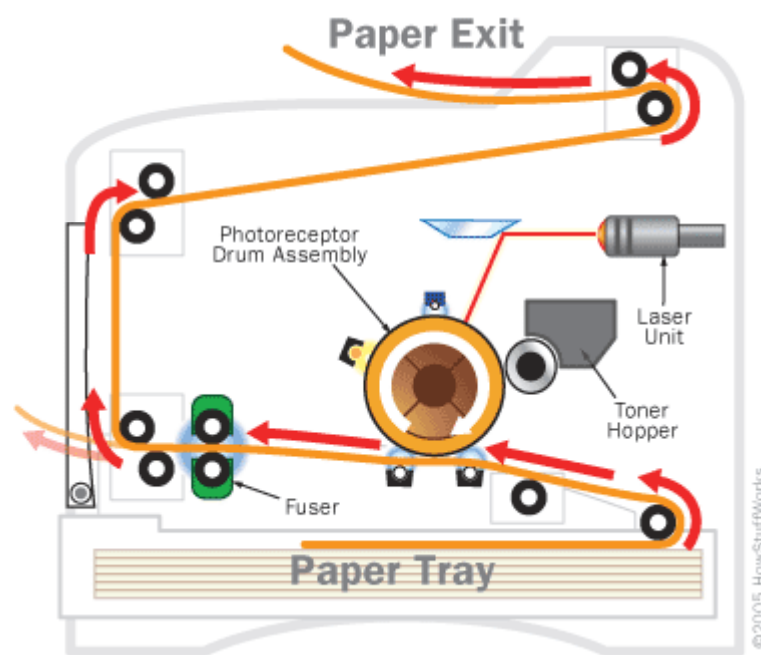
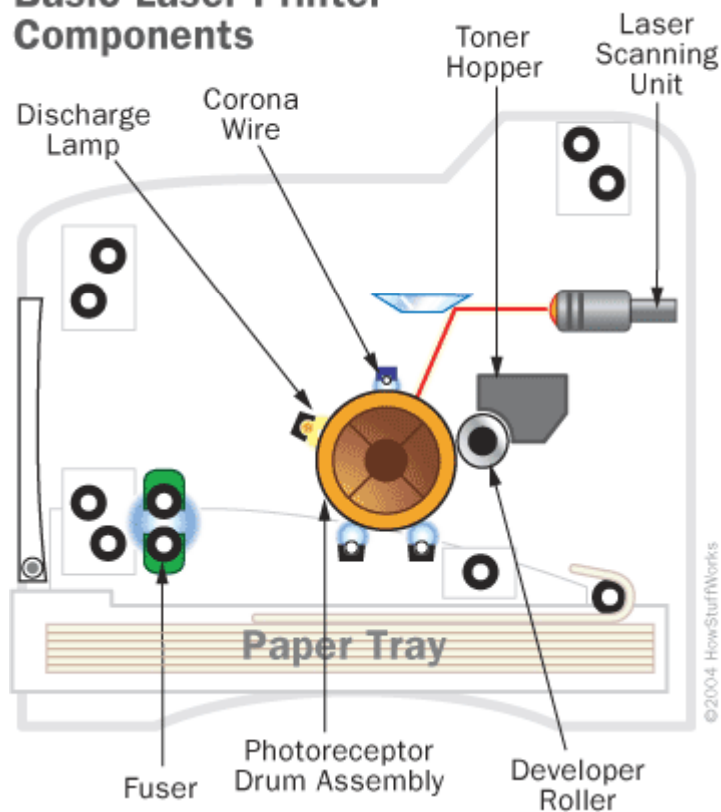
Pontos de meio-tom para várias faixas de escala de cinza.

(a) 0 – 6. (b) 14 – 20. (c) 28 – 34.

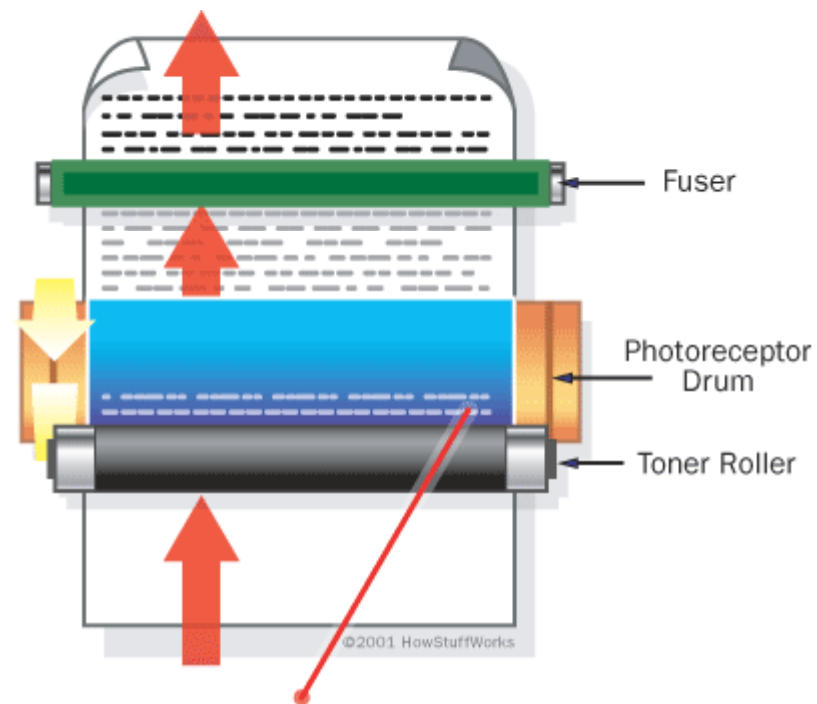
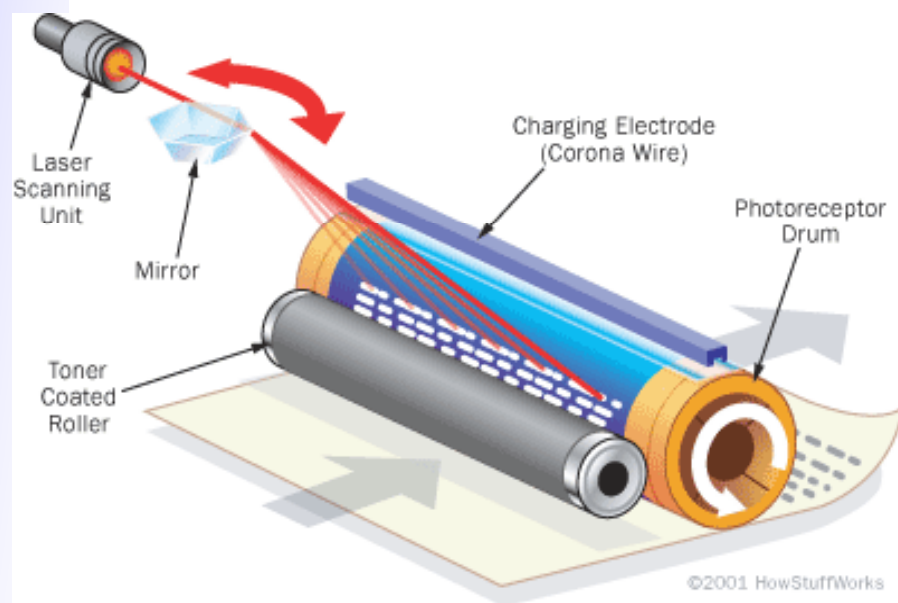
(d) 56 – 62. (e) 105 – 111. (f) 161 – 167.

Impressora laser

Basic Laser Printer Components



Impressora laser



Interfaces de rede local

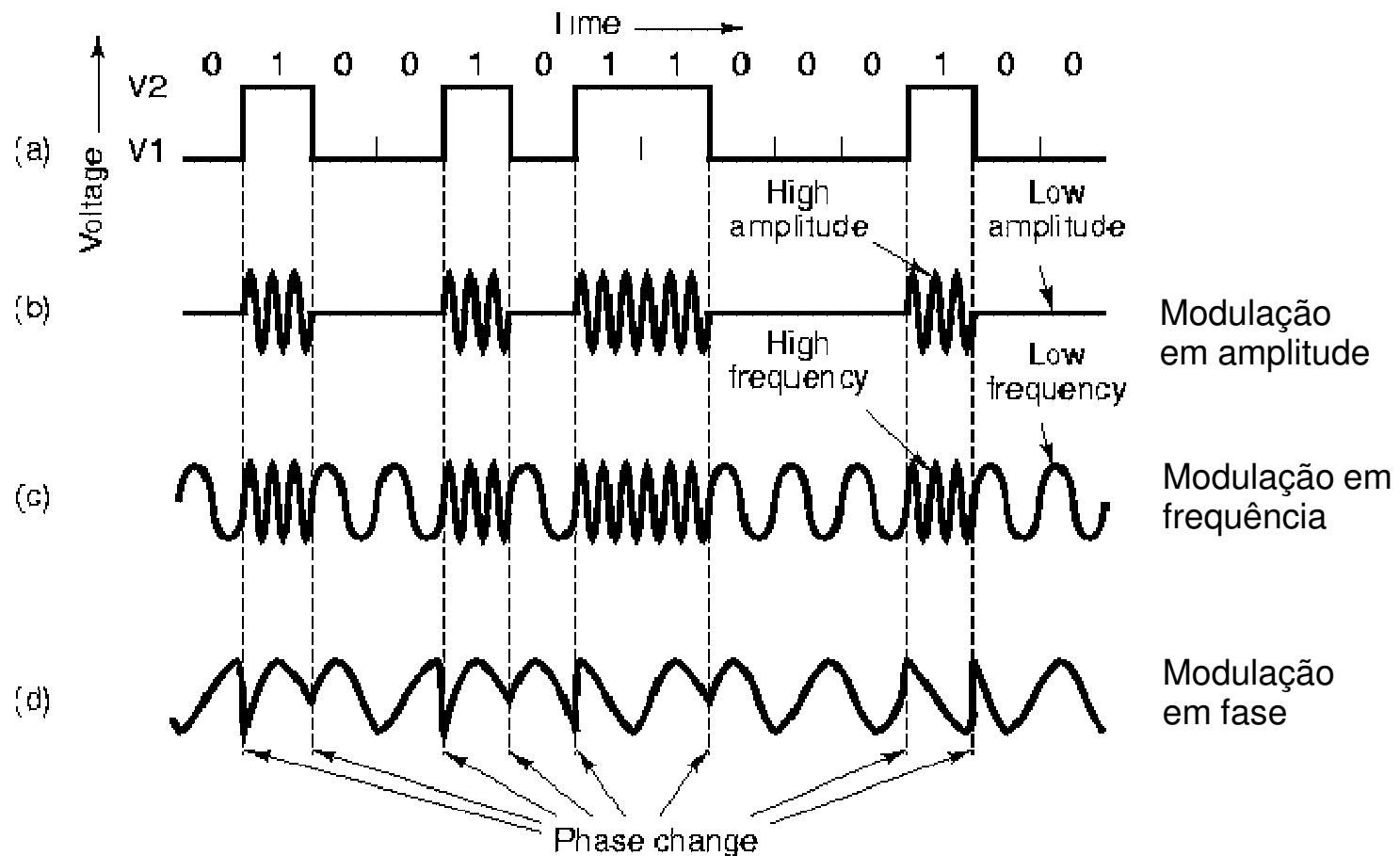
- Diversas tecnologias
 - Cabeadas:
 - Ethernet (100 Mbps, 1 Gbps, 10 Gbps)
 - Infiniband (10 Gbps ou mais)
 - Sem fio:
 - Ethernet sem fio (WiFi) (2 a 53 Mbps ou mais)
 - Bluetooth

... mais detalhes: Redes de Computadores

Modems: modulação digital

- Computadores se comunicam por sinais digitais (tensão contínua)
- Linhas telefônicas foram desenvolvidas para enviar sinais analógicos alternados
- Frequências entre 30 e 330 kHz
- Sinais digitais devem ser codificados para usar as linhas telefônicas
 - Variação de frequência, fase e/ou amplitude

Modems: modulação digital, meio analógico



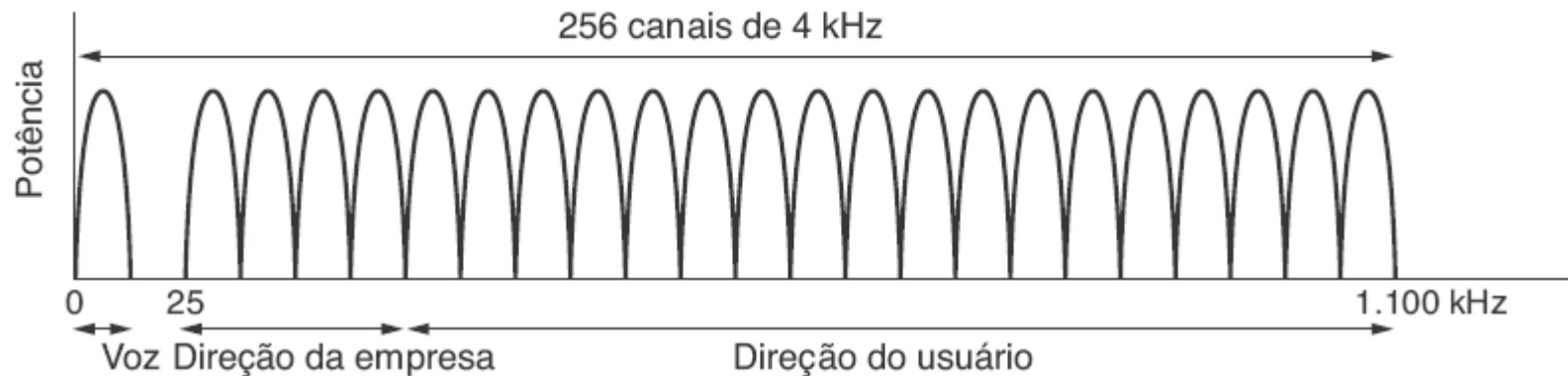
xDSL

Digital Subscriber Line

(linha digital do assinante)

- Oferece novas tecnologias de dados sobre a rede telefônica
- Remove filtros de linha que limitavam frequência e conseqüentemente banda
 - mudam o equipamento na central telefônica
- Sinais de mais alta frequência carregam dados, movidos para rede digital

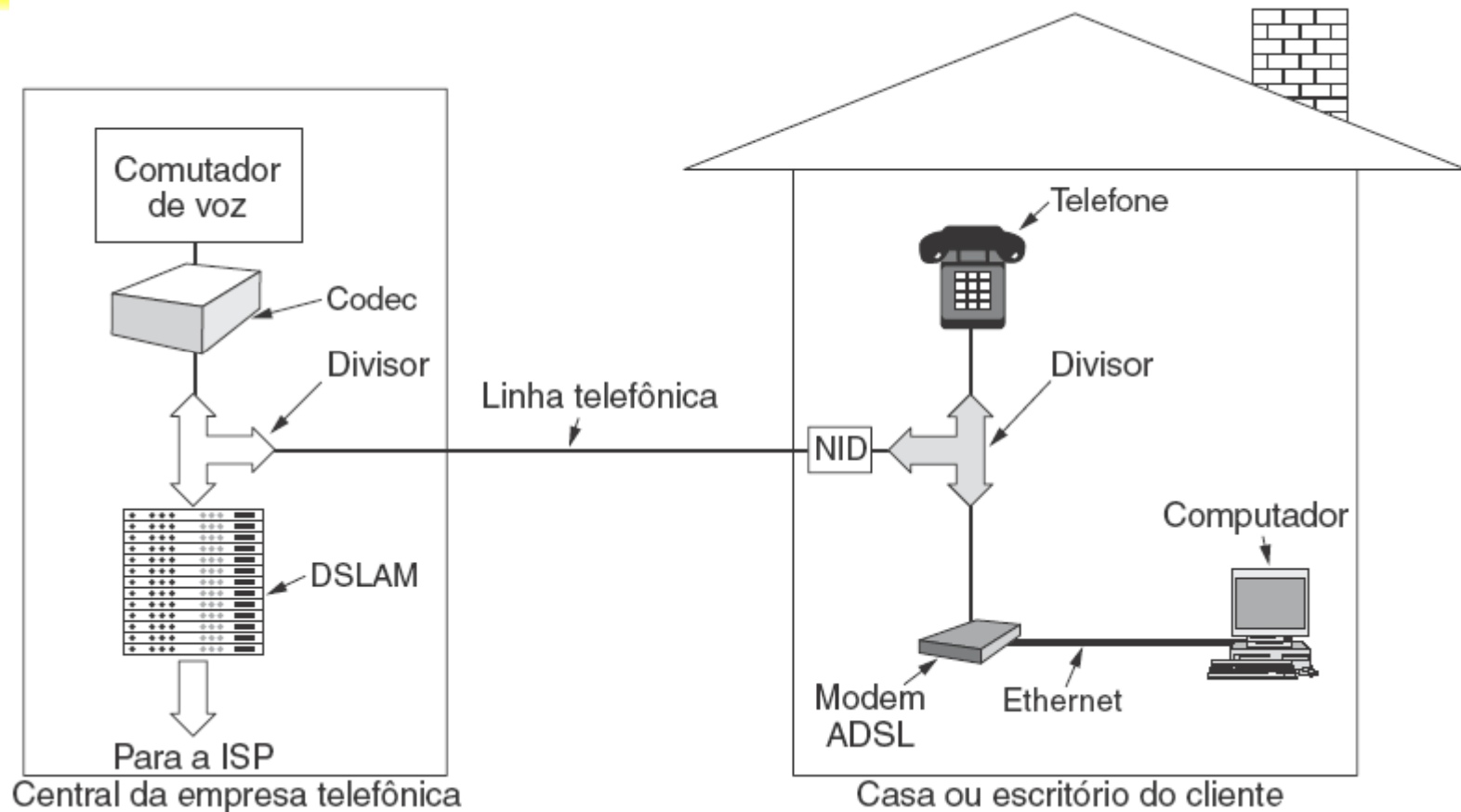
xDSL



Operação do ADSL.

Divisão comum: 32 canais na direção da empresa (upload)
resto na direção do usuário (download)

Configuração típica de equipamento ADSL.



Divisor: filtro analógico que separa faixa de voz das de dados

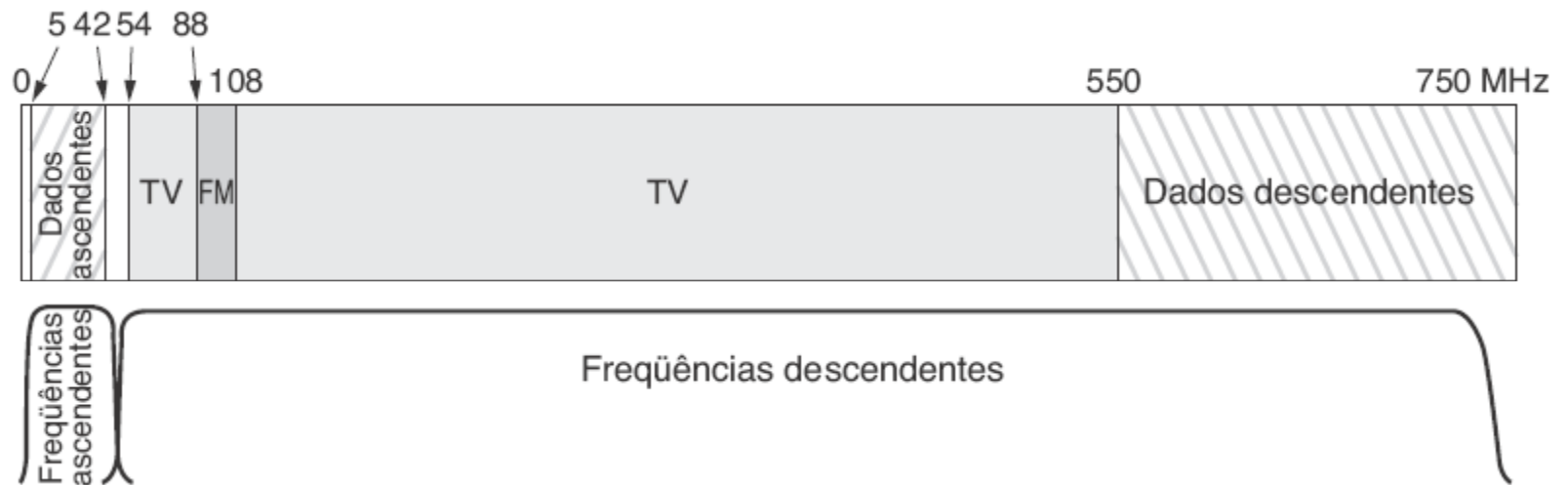
DSLAM – Digital Subscriber Line Access Multiplexer: equivale ao modem

NID = Network Interface Device

Acesso pela rede de TV a cabo

- Redes modernas de cabo
 - combinam distribuição por fibra e coaxial
 - previsão de canais *upstream* e *downstream*
- Utilização da rede de banda larga para transmissão de dados
- Utiliza alguns canais de 6 MHz do meio
 - Taxas de 50 Mbps possíveis
- Pode a princípio incluir tráfego de voz

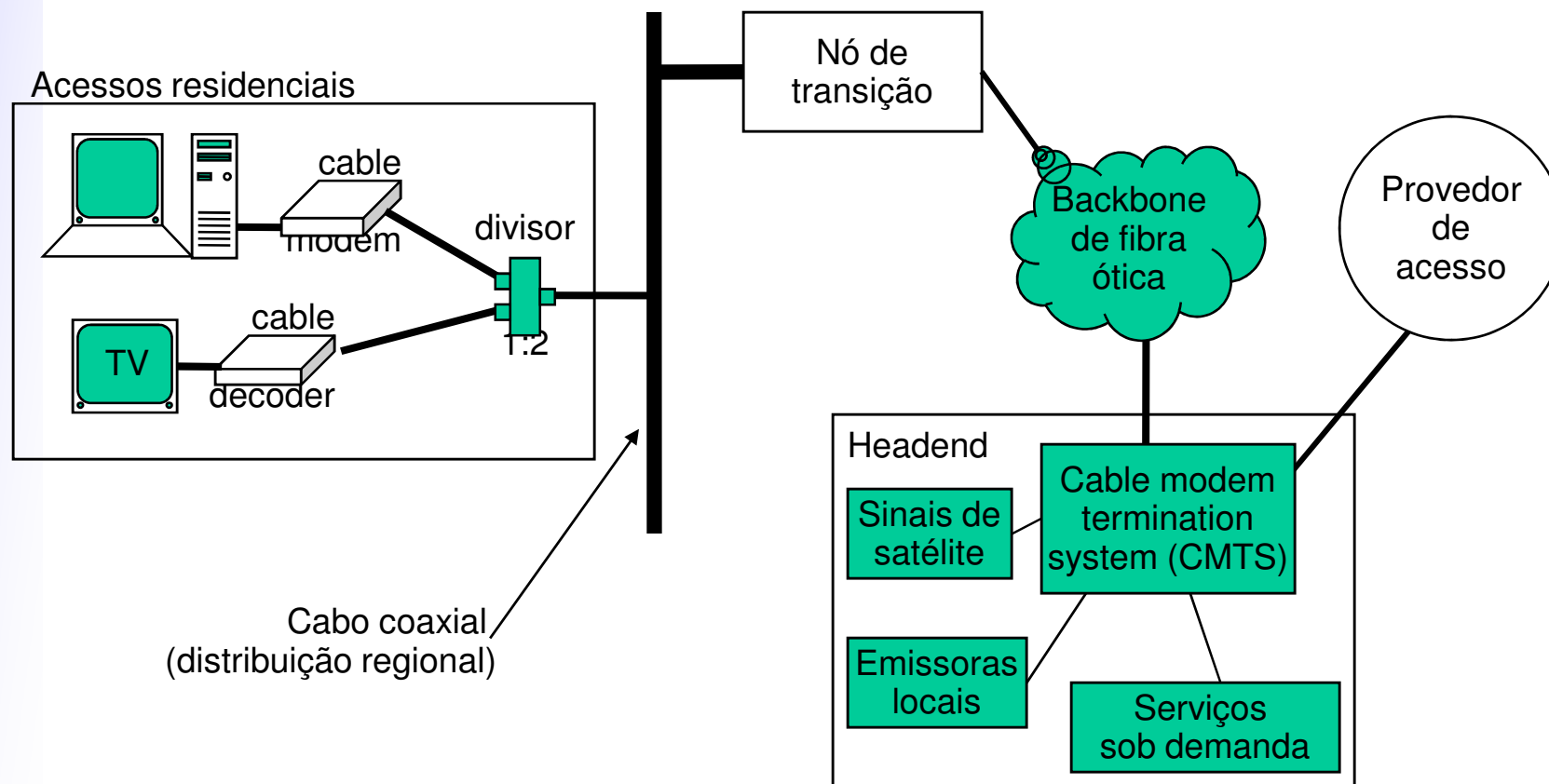
Acesso pela rede de TV a cabo



Alocação de freqüência em um sistema de TV a cabo usado para acesso à Internet.

Acesso pela rede de TV a cabo

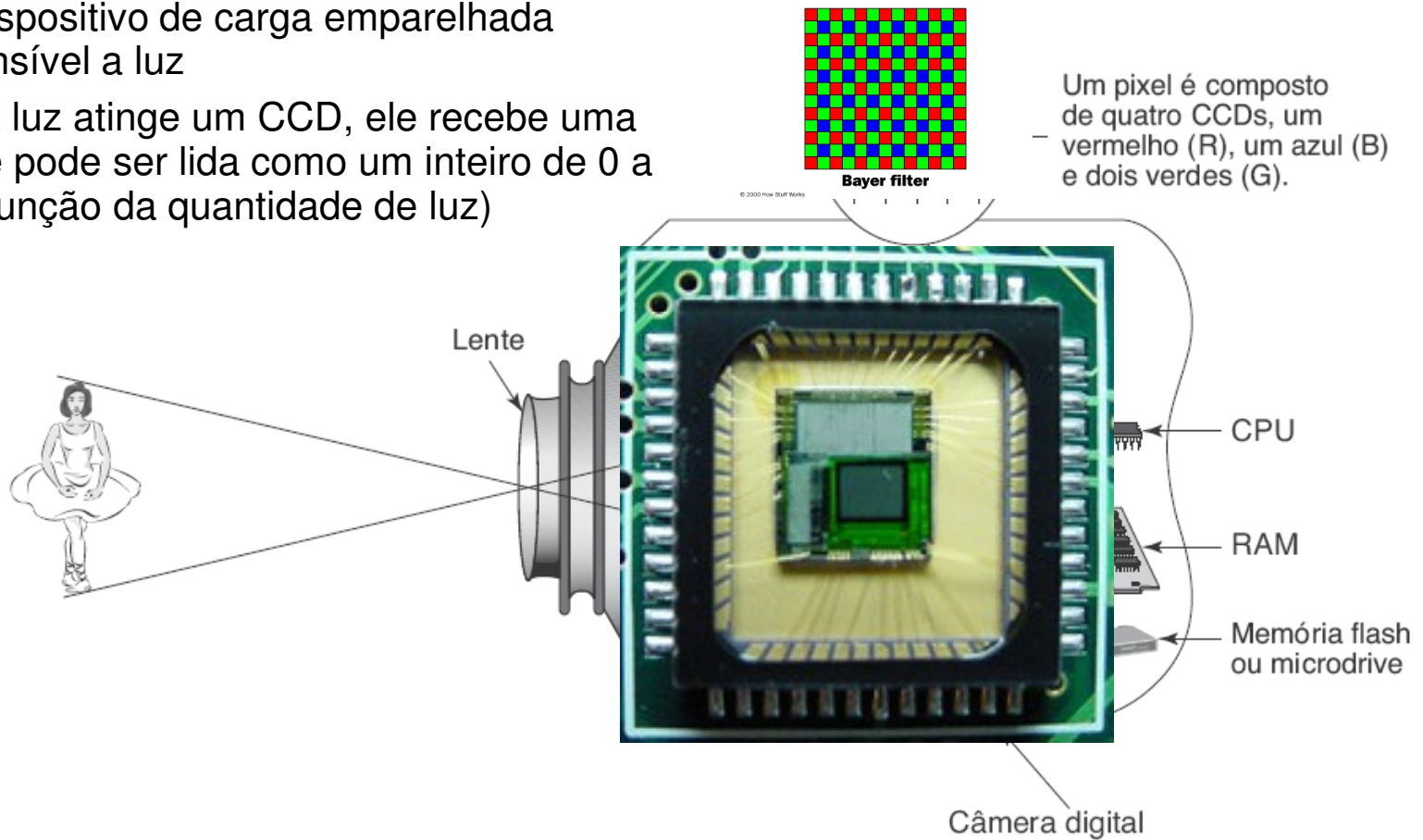
Rede híbrida fibra/coaxial



Câmeras digitais

CCD = Dispositivo de carga emparelhada sensível a luz

Quando a luz atinge um CCD, ele recebe uma carga que pode ser lida como um inteiro de 0 a 255 (em função da quantidade de luz)



Representação de dados

Inclui partes do cap. 2 e apêndices A e B

Código de Caracteres

Conjunto de caracteres usado por um computador

Cada caracter é mapeado em um número

Conjunto de Caracteres ASCII (8 bits)

Hex	Nome	Significado	Hex	Nome	Significado
0	NUL	Null	10	DLE	Data Link Escape
1	SOH	Start Of Heading	11	DC1	Device Control 1
2	STX	Start Of Text	12	DC2	Device Control 2
3	ETX	End Of Text	13	DC3	Device Control 3
4	EOT	End Of Transmission	14	DC4	Device Control 4
5	ENQ	Enquiry	15	NAK	Negative Acknowledgement
6	ACK	ACKnowledgement	16	SYN	SYNchronous idle
7	BEL	BELI	17	ETB	End of Transmission Block
8	BS	BackSpace	18	CAN	CANcel
9	HT	Horizontal Tab	19	EM	End of Medium
A	LF	Line Feed	1A	SUB	SUBstitute
B	VT	Vertical Tab	1B	ESC	ESCape
C	FF	Form Feed	1C	FS	File Separator
D	CR	Carriage Return	1D	GS	Group Separator
E	SO	Shift Out	1E	RS	Record Separator
F	SI	Shift In	1F	US	Unit Separator

Conjunto de Caracteres ASCII (8 bits)

Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.
20	(Space)	30	0	40	@	50	P	60	'	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	\$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	W	67	g	77	w
28	(38	8	48	H	58	X	68	h	78	x
29)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	
2D	-	3D	=	4D	M	5D]	6D	m	7D	}
2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

Códigos de caracteres: UNICODE (16 bits)

- Conjunto de caracteres “universal”
- Previsão para diferentes alfabetos
- Códigos baseados em dois bytes

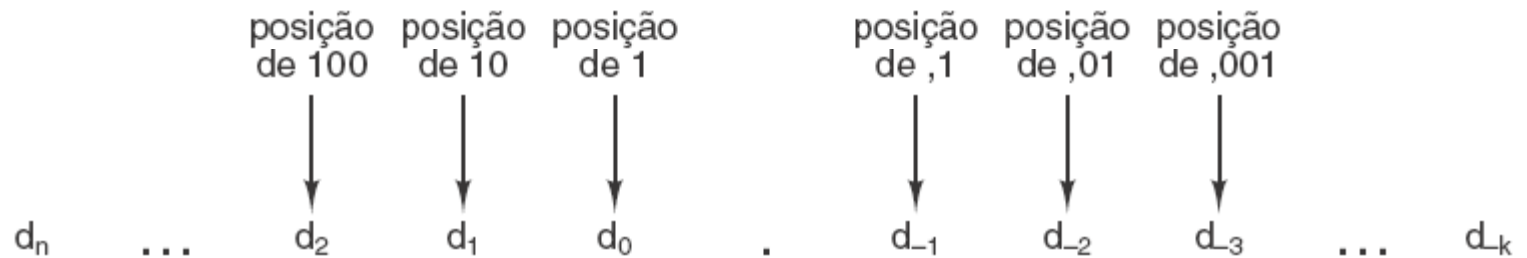
Representação de dados em memória

- Representação de inteiros
- Representação de ponto flutuante

Inclui partes do cap. 2 e apêndices A e B

Sistemas de números raiz ou números-base

A forma geral de um número decimal.



$$\text{Número} = \sum_{i=-k}^n d_i \times 10^i$$

Sistemas de números raiz ou números-base

O número 2001 em sistema binário, octal e hexadecimal.

Binário	1	1	1	1	1	0	1	0	0	0	1
	1×2^{10}	$+ 1 \times 2^9$	$+ 1 \times 2^8$	$+ 1 \times 2^7$	$+ 1 \times 2^6$	$+ 0 \times 2^5$	$+ 1 \times 2^4$	$+ 0 \times 2^3$	$+ 0 \times 2^2$	$+ 0 \times 2^1$	$+ 1 \times 2^0$
	1024	+ 512	+ 256	+ 128	+ 64	+ 0	+ 16	+ 0	+ 0	+ 0	+ 1
Octal	3	7	2	1							
	3×8^3	$+ 7 \times 8^2$	$+ 2 \times 8^1$	$+ 1 \times 8^0$							
	1536	+ 448	+ 16	+ 1							
Decimal	2	0	0	1							
	2×10^3	$+ 0 \times 10^2$	$+ 0 \times 10^1$	$+ 1 \times 10^0$							
	2000	+ 0	+ 0	+ 1							
Hexadecimal	7	D	1	.							
	7×16^2	$+ 13 \times 16^1$	$+ 1 \times 16^0$								
	1792	+ 208	+ 1								

Conversão de uma base para a outra

Números decimais e seus equivalentes binário, octais e hexadecimais.

Decimal	Binária	Octal	Hex
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D

Conversão de uma base para a outra

Números decimais e seus equivalentes binário, octais e hexadecimais.

Decimal	Binária	Octal	Hex
14	1110	16	E
15	1111	17	F
16	10000	20	10
20	10100	24	14
30	11110	36	1E
40	101000	50	28
50	110010	62	32
60	111100	74	3C
70	1000110	106	46
80	1010000	120	50
90	1011010	132	5A
100	11001000	144	64
1000	1111101000	1750	3E8
2989	101110101101	5665	BAD

Conversão de uma base para a outra

Exemplos de conversão octal
para binário e hexadecimal
para binário

Exemplo 1

Hexadecimal

1 9 4 8 . B 6
0001 1001 0100 1000 . 1011 0110 0
1 4 5 1 0 . 5 5 4

Binário

Octal

Exemplo 2

Hexadecimal

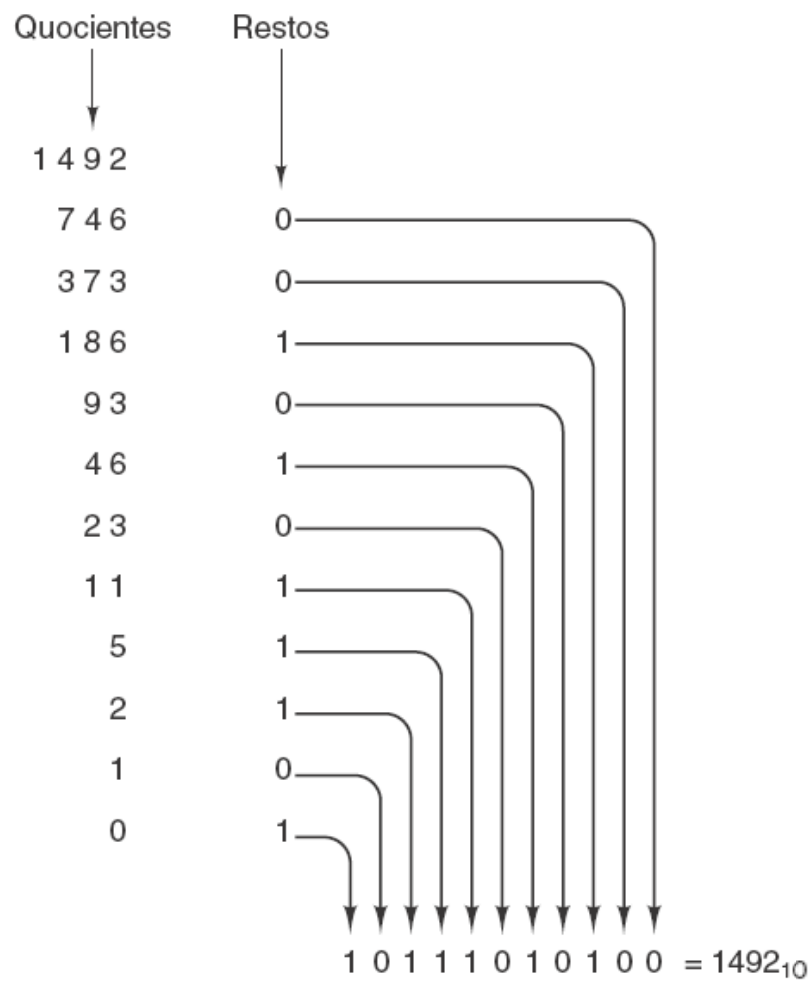
7 B A 3 . B C 4
0111 1011 1010 0011 . 1011 1100 0100
7 5 6 4 3 . 5 7 0 4

Binário

Octal

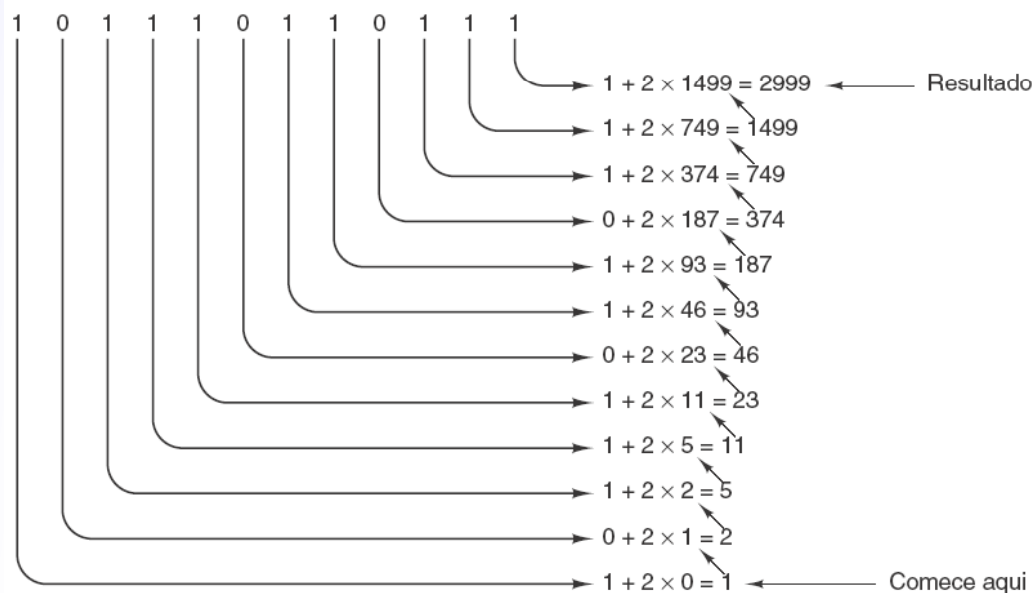
Conversão de uma base para a outra

Conversão do número decimal 1492 para binário por divisões sucessivas por 2, começando no topo e prosseguindo para baixo. Por exemplo, 93 dividido por 2 dá um quociente de 46 e um resto de 1, escrito na linha abaixo dele.



Conversão de uma base para a outra

Conversão do número binário 101110110111 para decimal por sucessivas multiplicações por 2, iniciando embaixo. Cada linha é formada multiplicando-se por 2 a que está abaixo dela e somando o bit correspondente. Por exemplo, 749 é duas vezes 374 mais o bit 1 na mesma linha que 749.



Representando números negativos

- Magnitude com sinal
- Complemento de um
- Complemento de dois
- Excesso $2^m - 1$

Números binários negativos

Números negativos de 8 bits em quatro sistemas.

N decimal	N binário	-N magnitude com sinal	-N complemento de 1	-N complemento de 2	-N excesso 128
1	00000001	10000001	11111110	11111111	01111111
2	00000010	10000010	11111101	11111110	01111110
3	00000011	10000011	11111100	11111101	01111101
4	00000100	10000100	11111011	11111100	01111100
5	00000101	10000101	11111010	11111011	01111011
6	00000110	10000110	11111001	11111010	01111010
7	00000111	10000111	11111000	11111001	01111001
8	00001000	10001000	11110111	11111000	01111000
9	00001001	10001001	11110110	11110111	01110111
10	00001010	10001010	11110101	11110110	01110110
20	00010100	10010100	11101011	11101100	01101100
30	00011110	10011110	11100001	11100010	01100010
40	00101000	10101000	11010111	11011000	01011000
50	00110010	10110010	11001101	11001110	01001110
60	00111100	10111100	11000011	11000100	01000100

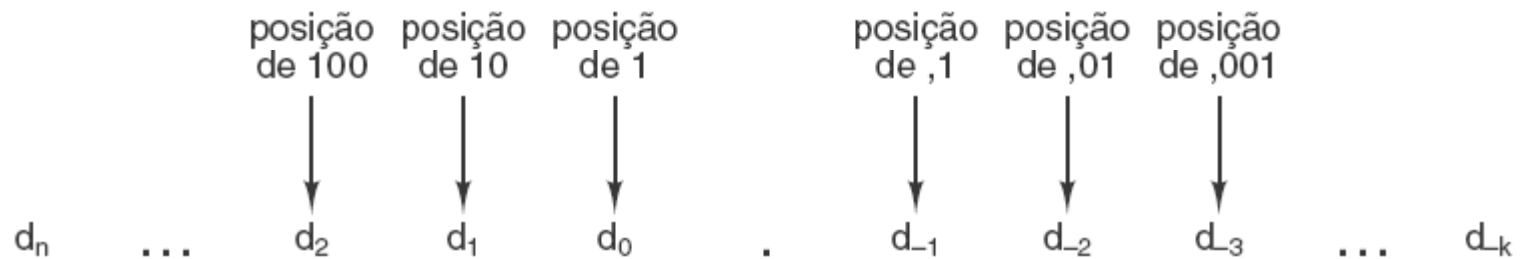
Números binários negativos

Números negativos de 8 bits em quatro sistemas. *(Continuação)*

N decimal	N binário	-N magnitude com sinal	-N complemento de 1	-N complemento de 2	-N excesso 128
70	01000110	11000110	10111001	10111010	00111010
80	01010000	11010000	10101111	10110000	00110000
90	01011010	11011010	10100101	10100110	00100110
100	01100100	11100100	10011011	10011100	00011100
127	01111111	11111111	10000000	10000001	10000001
128	Não existente	Não existente	Não existente	10000000	00000000

Sistemas de números raiz ou números-base

A forma geral de um número decimal.



$$\text{Número} = \sum_{i=-k}^n d_i \times 10^i$$

Princípios do ponto flutuante

- Deve separar faixa de precisão
- Usar notação científica $n = f \times 10^e$
 - f é a fração ou mantissa
 - e é o expoente (um número positivo ou negativo)
 - faixa determinada pelo # de dígitos/bits do expoente
 - precisão determinada pelo # de dígitos/bits na fração

- Exemplos

$$3,14 = 0,314 \times 10^1 = 3,14 \times 10^0$$

$$0,000001 = 0,1 \times 10^{-5} = 1,0 \times 10^{-6}$$

$$1941 = 0,1941 \times 10^4 = 1,941 \times 10^3$$

Princípios de ponto flutuante

Sete regiões da linha dos números reais

Números negativos grandes menores que $-0,999 \times 10^{99}$.

Números negativos entre $-0,999 \times 10^{99}$ and $-0,100 \times 10^{-99}$.

Números negativos pequenos maiores que $-0,100 \times 10^{-99}$.

Zero.

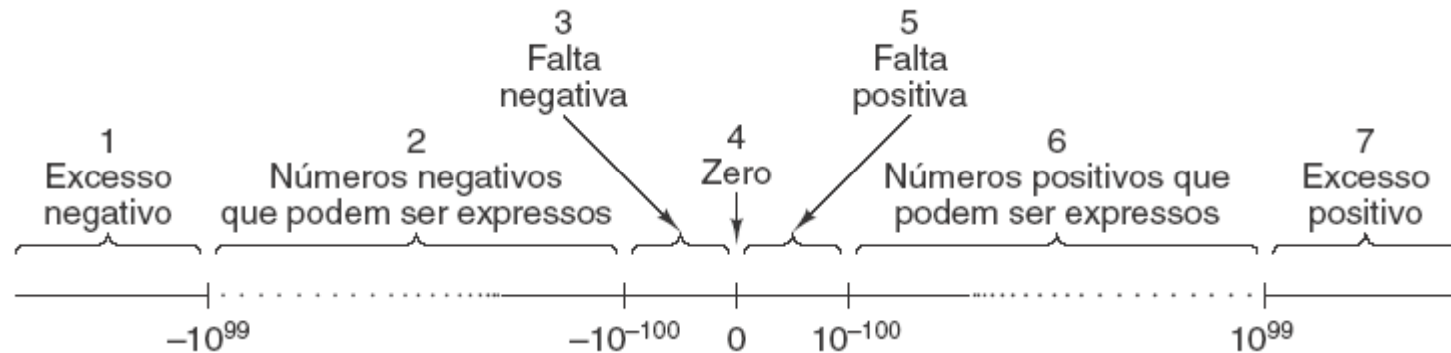
Números positivos pequenos menores que $0,100 \times 10^{-99}$.

Números positivos entre $0,100 \times 10^{-99}$ e $0,999 \times 10^{99}$.

Números positivos grandes maiores que $0,999 \times 10^{99}$.

Princípios de ponto flutuante

Sete regiões da linha dos números reais



Erro de excesso, erro de falta

Arredondamento: números de ponto flutuante não formam um contínuo

Padrão de ponto flutuante IEEE 754

Exemplos de números de ponto flutuante normalizados.

Exemplo 2: Exponenciação com base 16

Não-normalizado: $0 \underbrace{1000101}_{\text{Expoente}} . \overbrace{0000}^{16^{-1}} \overbrace{0000}^{16^{-2}} \overbrace{0001}^{16^{-3}} \overbrace{1011}^{16^{-4}} = 16^5 (1 \times 16^{-3} + B \times 16^{-4}) = 432$

Sinal Expoente + em excesso é $69 - 64 = 5$

Mantissa é $1 \times 16^{-3} + B \times 16^{-4}$

Para normalizar, desloque a mantissa 2 dígitos hexadecimais para a esquerda e subtraia 2 do expoente.

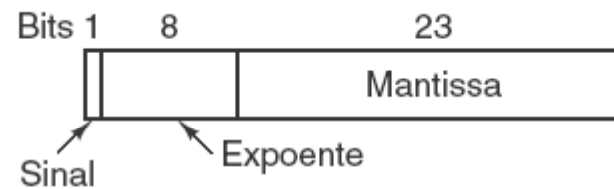
Normalizado: $0 \underbrace{1000011}_{\text{Expoente}} . \overbrace{0001}^{16^{-1}} \overbrace{1011}^{16^{-2}} \overbrace{0000}^{16^{-3}} \overbrace{0000}^{16^{-4}} = 16^3 (1 \times 16^{-1} + B \times 16^{-2}) = 432$

Sinal Expoente + em excesso é $64 \text{ é } 67 - 64 = 3$

Mantissa é $1 \times 16^{-1} + B \times 16^{-2}$

Padrão de ponto flutuante IEEE 754

Formatos de ponto flutuante padrão IEEE. (a) Precisão simples. (b) Precisão dupla.



(a)



(b)

Padrão de ponto flutuante IEEE 754

Características dos números de ponto flutuante padrão IEEE.

Item	Precisão simples	Precisão dupla
Bits de sinal	1	1
Bits de expoente	8	11
Bits na mantissa	23	52
Bits, total	32	64
Sistema de expoente	Excesso 127	Excesso 1023
Faixa de expoente	-126 a +127	-1022 a +1023
Menor número normalizado	2^{-126}	2^{-1022}
Maior número normalizado	aprox. 2^{128}	aprox. 2^{1024}
Faixa decimal	aprox. 10^{-38} a 10^{38}	aprox. 10^{-308} a 10^{308}
Menor número desnormalizado	aprox. 10^{-45}	aprox. 10^{-324}

Padrão de ponto flutuante IEEE 754

Tipos numéricos padrão IEEE.

Normalizado	±	$0 < \text{Exp} < \text{Max}$	Qualquer padrão de bits
Desnormalizado	±	0	Qualquer padrão de bits não-zero
Zero	±	0	0
Infinito	±	1 1 1...1	0
Não é um número	±	1 1 1...1	Qualquer padrão de bits não-zero

↙ Bit de sinal

Menor número normalizado: 1 no expoente e 0 na mantissa: 2^{-126}

Menor número desnormalizado: 0 no expoente e 0000..01 na mantissa

$$: 2^{-126} 2^{-23} = 2^{-149}$$

