

Unidade III

5 MEMÓRIAS

A memória é a estrutura do computador na qual são armazenados os programas e os dados e, conforme já discurremos brevemente, existem diversos tipos de memória no computador organizados de maneira hierárquica. Quanto mais próxima a memória estiver do nível superior mais rápida, porém mais cara. O nível superior diz respeito às memórias do processador, primeiramente os registradores (mais próximos da CPU) e, posteriormente, a memória *cache*. Já estudamos registradores no tópico sobre processadores; veremos a seguir a memória principal e *cache* também primárias, em seguida a memória secundária (figura 36).

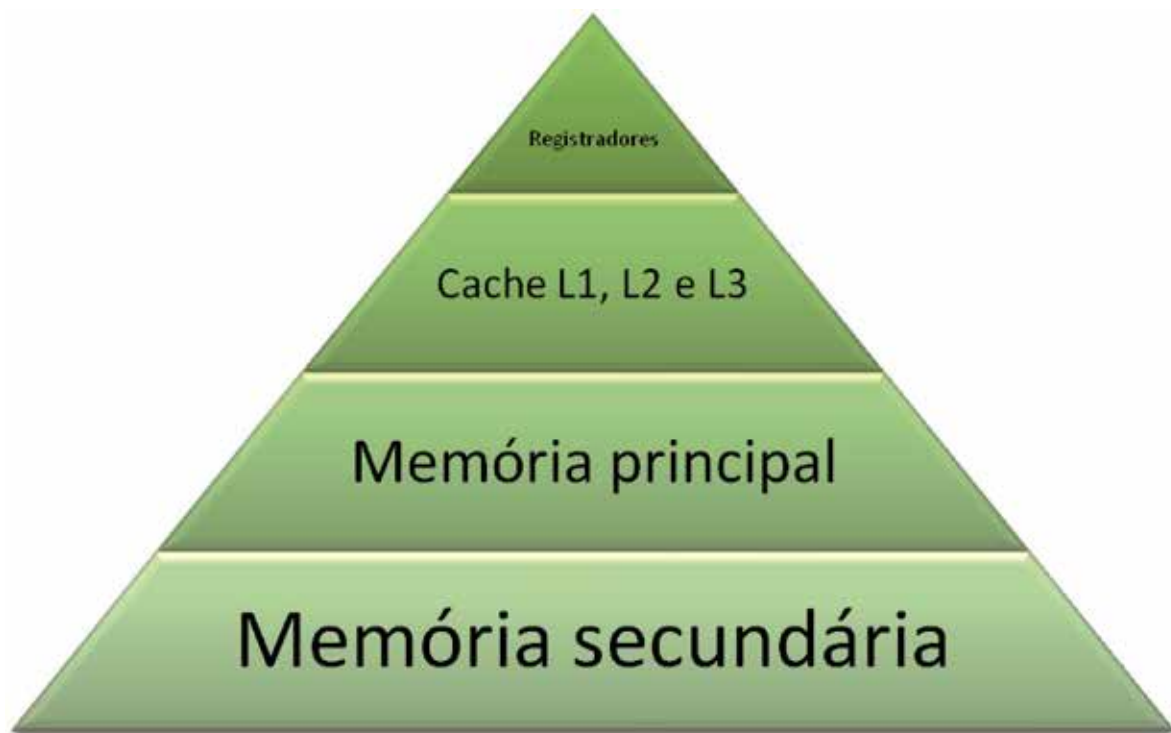


Figura 36 - Hierarquia de memória

5.1 Memória primária

Memórias primárias são memórias que o **processador** pode endereçar diretamente. Sua função é conter a informação necessária para o processador num determinado momento, por exemplo os programas em execução. Nesta categoria, se inserem a **memória principal**, **cache** e **registradores**.

5.2 Memória principal

A memória principal também é conhecida por DRAM (*Dynamic Random Access Memory*), memória dinâmica de acesso randômico (figuras 37 e 38).

Costumeiramente, a memória principal é volátil, ou seja, mantém os dados enquanto o computador estiver energizado. A memória principal é utilizada para armazenar os programas em **execução** e os dados em **uso** por esses programas. Realiza comunicação constante com o processador.

!!! Observação

A memória principal é a próxima a ser considerada no nível hierárquico, só perdendo para registradores e *cache* por ser um dispositivo externo à CPU e estar mais distante desta.



Figura 37 - Modelo de memória RAM de microcomputador (com capa protetora)

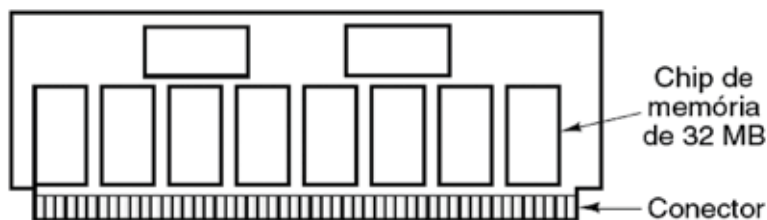


Figura 38 - Modelo de chip de memória de 32 MB

Fonte: TANENBAUM, 2007a.

Existem diversos tipos de memórias dinâmicas. Entre elas podemos citar:

- **FPM** (*Fast-Page Mode*): uma das primeiras tecnologias de memória RAM. Nessas memórias, o primeiro acesso de leitura tinha um tempo de acesso maior que as leituras seguintes. Isso ocorria porque eram realizadas quatro leituras seguidas ao invés de apenas uma por vez.

- **DRAM EDO** (*Extended Data Output*): permite que uma referência de memória seja iniciada mesmo antes que uma anterior esteja terminada, a DRAM EDO é assíncrona, ou seja, não utiliza o *clock* para transmissão dos dados.
- **SDRAM** (*Synchronous DRAM*): utiliza o *clock* como referência e mistura a memória dinâmica e estática. Essa memória realiza uma operação a cada ciclo de *clock*.
- **Rambus DRAM**: recebeu esse nome, pois foi criada pela empresa Rambus Inc. e chegou ao mercado com o apoio da Intel. Ela trabalha apenas com 16 *bits* por vez e em uma frequência muito maior, 400 MHz, e com duas operações por ciclo de *clock*. Essa memória não chegou a ter grande aceitação no mercado, pois tinha muito aquecimento e um custo elevado.
- **DDR SDRAM** (*Double Data Rate SDRAM*): consegue tratar o dobro de informações por ciclo de *clock*, quando comparado à SDRAM.
- **DDR2 SDRAM**: é uma evolução da memória DDR SDRAM, que passa a trabalhar com quatro operações por ciclo de *clock*, ou seja, o dobro da DDR SDRAM.
- **DDR3 SDRAM**: também é uma evolução, só que da DDR2 SDRAM, na qual novamente é dobrada a quantidade de operações por ciclo de *clock*, ou seja, passa a ter oito operações por ciclo de *clock*.

Mas, o que contém a memória? Qual sua estrutura?

As memórias consistem de células, cada qual com seu endereço. As células contêm os *bits* ou dígito binário, que são a unidade básica de memória. Um *bit* armazena apenas os valores 0 ou 1 e é o bloco básico de informações.

Os binários 0 e 1 representam variações diferentes na tensão elétrica dos circuitos eletrônicos do computador. Nossos programas, em linguagem de alto nível (com caracteres), funcionam no baixo nível com combinações de 0 e 1.

Mais adiante estudaremos a conversão binários/caracteres.

Um conjunto de *bits* forma um agrupamento. Um dos agrupamentos mais comuns é o *byte*, que contém 8 *bits* de comprimento. Como o *bit* é um valor binário (0 e 1), os múltiplos são potências de 2, deste modo 1KB (*kilobyte*) não é exatamente 1.000 *bytes* (10^3), mas 1.024 *bytes* (2^{10}); 1 MB (*megabyte*) contém 1.048.576 e não 1.000.000 de *bytes*; 1 GB (*gigabyte*) equivale a 1.073.741.824 de *bytes* e 1 TB (*terabyte*) equivale a 1.099.511.627.776 *bytes*.

As figuras 39, 40 e 41 representam uma célula (endereço) de 64 *bits* ou 8 *bytes*, cada quadradinho é 1 *bit* e cada fileira 1 *byte*:

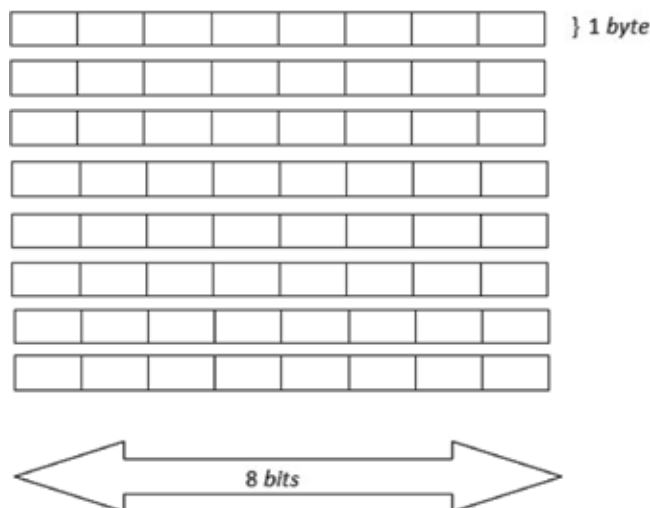


Figura 39 - Modelo de células de endereço de 8 bits

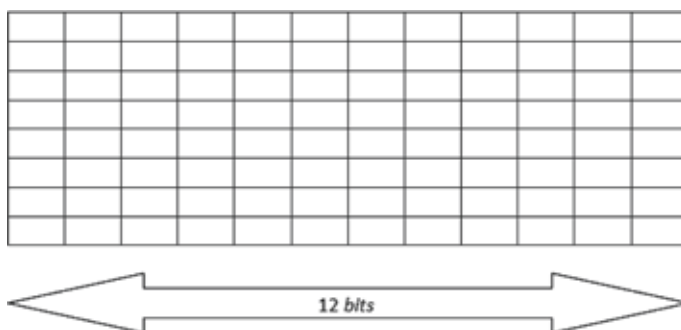


Figura 40 - Modelo de células de endereço de 12 bits

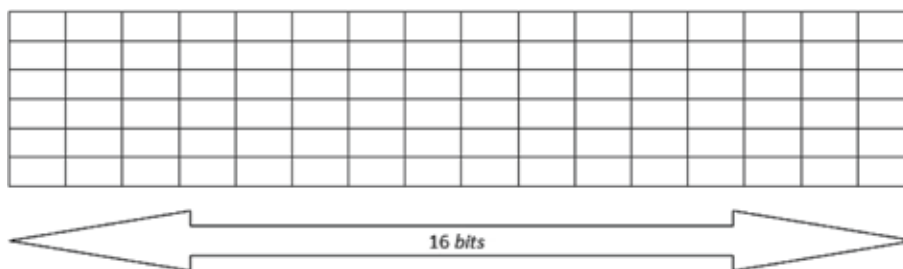


Figura 41 - Modelo de células de endereço de 16 bits



Lembrete

A célula é a menor unidade endereçável de memória.

Outro conceito importante é a **palavra**. Uma palavra é o tamanho da unidade de dados que é transferida da memória para o processador. Quando falamos que um computador é de 32 bits ou 64 bits, estamos nos referindo ao tamanho da palavra desse processador. Ou seja, um computador de 32

bits consegue transmitir 4 *bytes* entre a memória e o processador, enquanto um computador de 64 *bits* transmite 8 *bytes*. Esse conceito é importante porque, se o processador recebe um valor de 8 *bytes* da memória, todos os seus registradores têm que ter no mínimo esse comprimento. Isso implica também a quantidade de memória que um computador pode suportar. Se o registrador dedicado a tratar endereços de memória tem 1 *bit* de comprimento, o computador só suportará dois endereços ou duas células (0 e 1). Se esse registrador tiver 32 *bits*, a quantidade de células endereçáveis será de 4.294.967.295. Em um computador com células de 1 *byte*, isso equivale a 4 GB de memória. Se esse registrador tiver 64 *bits*, a quantidade de endereços possíveis sobe para 18.446.744.073.709.551.615, ou aproximadamente 18,45 *exabytes*, uma quantidade de memória impensável nos padrões tecnológicos atuais.

5.3 Memória cache

Normalmente as CPUs trabalham mais rapidamente do que a memória principal, sendo assim, existe um descompasso, ou seja, a CPU pode ficar ociosa, comprometendo o desempenho do sistema.

Os projetistas poderiam desenvolver memórias mais rápidas, entretanto, outro fator que torna a comunicação com a memória principal mais lenta é o fato do tráfego de dados entre os dispositivos terem de passar pelo barramento. A solução é inserir memórias de alto desempenho no *chip* da CPU. São as *caches*, atualmente três: L1, L2 e L3. A *cache* é conhecida por SRAM (*Static Random Access Memory*), memória estática de acesso randômico.

Quanto mais próxima da CPU mais rápida, conforme esquema representado pela figura 42:



Figura 42 - Representação de velocidade de acesso



Lembrete

A localização **lógica** da *cache* é entre a CPU e a memória principal. Normalmente, ela se encontra integrada fisicamente ao processador.

A ideia básica é a seguinte: as palavras de memória usadas com mais frequência são mantidas na *cache*, desse modo o processador busca a informação primeiramente nas *caches*, somente se não encontrar em nenhuma delas é que irá buscar o dado na memória primária.

Em outras palavras, a *cache* é um bloco de memória para o armazenamento temporário de dados (volátil) que possuem uma grande probabilidade de serem utilizados novamente. Tem por finalidade aumentar o desempenho do processador e geralmente vem acoplada a este.



Lembrete

Conforme vimos, existem as memórias RAMs estáticas e dinâmicas, ambas voláteis, de modo que as estáticas recebem o nome de SRAM, e as dinâmicas de DRAM.

As SRAMs são fabricadas com a utilização de circuitos semelhantes ao *flip-flop*, nos quais são retidas as informações. Esse tipo de memória guarda as informações enquanto houver alimentação de energia elétrica, só mudando as informações armazenadas quando o programa executar alguma alteração. As SRAMs são extremamente rápidas e por isso são utilizadas como memória *cache*.

As DRAMs usam transistores e capacitores. Quando o capacitor está carregado, significa que tem informação (1) e quando está descarregado, não há informação (0). Porém, o mesmo precisa ter um processo denominado de recarga chamado *refresh*, para evitar que os dados que estejam na memória sejam perdidos. Então, a memória DRAM exige uma complexidade maior na fabricação, que é compensada com a grande capacidade que consegue armazenar, por isso as DRAMs são utilizadas como memória principal.

A seguir, um quadro comparativo da SRAM e DRAM demonstrada na tabela 1:

	SRAM	DRAM
Fabricação	Flip-Flop	Células de transistores e capacitores
Velocidade de gravação	Alta velocidade	Baixa velocidade
Armazenamento	Baixa capacidade	Alta capacidade
Onde é utilizado	Cache	Memória principal

Tabela 1 - Quadro comparativo da SRAM e DRAM

5.4 Outras memórias

Além das memórias RAM, as quais se podem gravar, ler e apagar, existe a ROM (*Read Only Memory*) ou Memória Somente Leitura, na qual o conteúdo normalmente não deve ser modificado e nem apagado. A ROM fica localizada na placa-mãe do computador. Quando se fabrica a memória ROM, são gravadas todas as informações por meio de material próprio, informações relativas à BIOS (*Basic Input Output System*), sistema básico de entrada e saída, em outras palavras, suporte básico de acesso ao computador que oferece ao processador suporte para comunicação com demais dispositivos, POST (*Power On SelfTest*) autoteste, programa que realiza varredura do sistema computacional quando inicializamos a máquina (reconhece dispositivos) e *Setup* que mantém configurações relativas à configuração do sistema e calendário. Assim como a memória RAM, existem alguns tipos de memória ROM:

- **PROM** (*Programmable ROM*): é similar à ROM, porém pode ser programada após a fabricação.

- **EPROM** (*Erasable PROM*): além de poder ser programada, pode ser apagada e depois reprogramada. Isso ocorre com a emissão de luz ultravioleta, na qual as informações precisam ser apagadas por completo para depois ser realizada a reprogramação.
- **EEPROM** (*Electrically-Erasable PROM*): é uma evolução da EPROM, na qual a funcionalidade é similar, mudando no modo de exclusão de dados e gravação. A exclusão de dados, ao invés de luz ultravioleta, se utiliza de pulsos elétricos, e na gravação não é necessário retirar o *chip* do computador, como era necessário na EPROM.
- **EAROM** (*Electrically-Alterable PROM*): é similar a EEPROM, porém não há necessidade da gravação e alteração de todas as informações de uma única vez, podendo ser aos poucos, por isso essa memória é geralmente utilizada para reescrita parcial de informações.
- **Flash**: essa memória é um tipo de EEPROM, porém a gravação e alteração são muito mais rápidas; sem contar que tem uma maior durabilidade e consegue armazenar mais informações.

5.5 Memória secundária

A memória secundária é a memória para armazenamento permanente dos dados, ou seja, o tipo de memória que serve para a recuperação de informações gravadas em uma sessão de trabalho e que não irão se perder quando do desligamento do sistema computacional.

É o tipo mais lento de memória, porém possibilita o armazenamento de um grande volume de dados, sendo também a mais econômica em relação às demais na hierarquia de memória.

5.6 Discos magnéticos

O disco magnético (figura 43), também conhecido por disco rígido ou HD (*Hard Disk*), é um tipo de memória secundária que integra o sistema computacional, composto de pratos de alumínio com revestimento magnetizável. O processo de leitura e gravação ocorre por intermédio de agulhas que magnetizam a superfície por meio de uma corrente elétrica com polaridade positiva ou negativa. Dependendo da polaridade, as partículas magnéticas agrupam-se à direita ou à esquerda, podendo simbolizar um *bit* 0 ou um *bit* 1.



Figura 43 - Disco magnético (disco rígido)

Este exemplo possui quatro pratos com quatro cabeças de leitura e gravação em apenas um braço que realiza movimentos do centro para a periferia do disco e vice-versa.

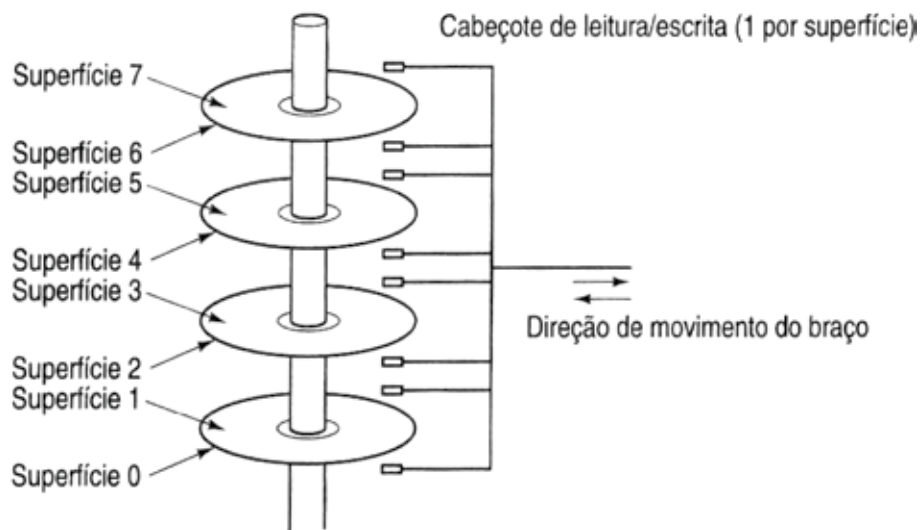


Figura 44 - Disco magnético com quatro pratos

Fonte: TANENBAUM, 2007a.

À medida que o prato gira, uma sequência circular de *bits* é lida ou escrita, o que compõem as trilhas do disco (colorido na figura 45).



Observação

Cada uma das trilhas é dividida em um número de setores normalmente fixados em 512 bytes de dados.

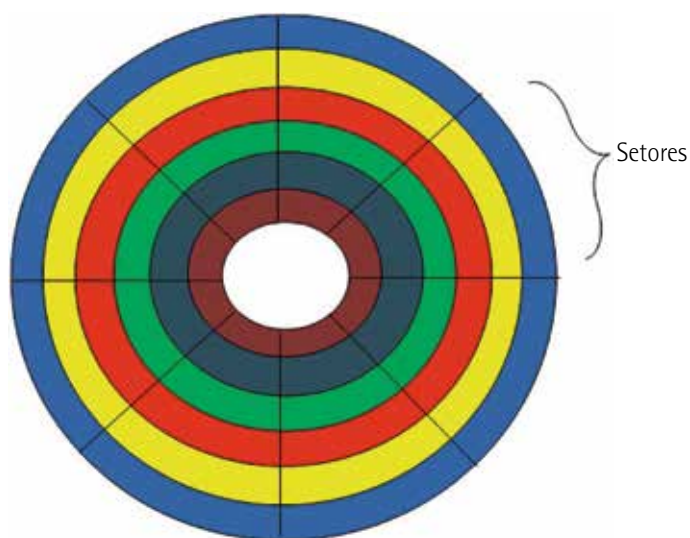


Figura 45 - Disco rígido

Os discos rígidos participam do mecanismo de *Swapping*, técnica utilizada pelo sistema operacional para "extensão" da memória principal, quando esta não é suficiente para execução de programas de alta prioridade.

O computador pode funcionar sem disco rígido, desde que ele possa reconhecer e utilizar um sistema operacional alocado em outra região, um disco removível (CD), por exemplo.



Saiba mais

<ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ea869_03/turma_U/cap2_partell.pdf>

5.7 Sistema RAID

O desempenho da CPU vem aumentando com o passar do tempo, entretanto, o mesmo não acontece com os discos magnéticos, ou seja, a lacuna entre o desempenho do processador e do disco rígido ficou maior com o passar do tempo. Tanenbaum cita Patterson *et al* (1988) como pioneiros na solução para minimizar esta lacuna. Eles definiram formas de organizar os discos, o que deu origem ao sistema RAID (*Redundant Array of Inexpensive Drives*), arranjo redundante de discos econômicos ou pela redefinição da indústria (*Redundant Array of Independent Disks*), arranjo redundante de discos independentes.



Observação

A ideia é aplicar paralelismo aos discos, ou seja, ter dois ou mais discos trabalhando simultaneamente para um mesmo fim.

Objetivos básicos:

- Acelerar o carregamento de dados do disco, por meio de uma técnica chamada divisão de dados.
- Tornar o sistema de disco mais seguro, por meio de uma técnica chamada espelhamento.

O RAID possui diversas combinações de divisão e espelhamento de dados originando diversos níveis, sendo mais comuns o RAID 0, RAID 1, RAID 2, RAID 3, RAID 4 e RAID 5, conforme a figura 46. As áreas sombreadas representam cópias de segurança dos dados:

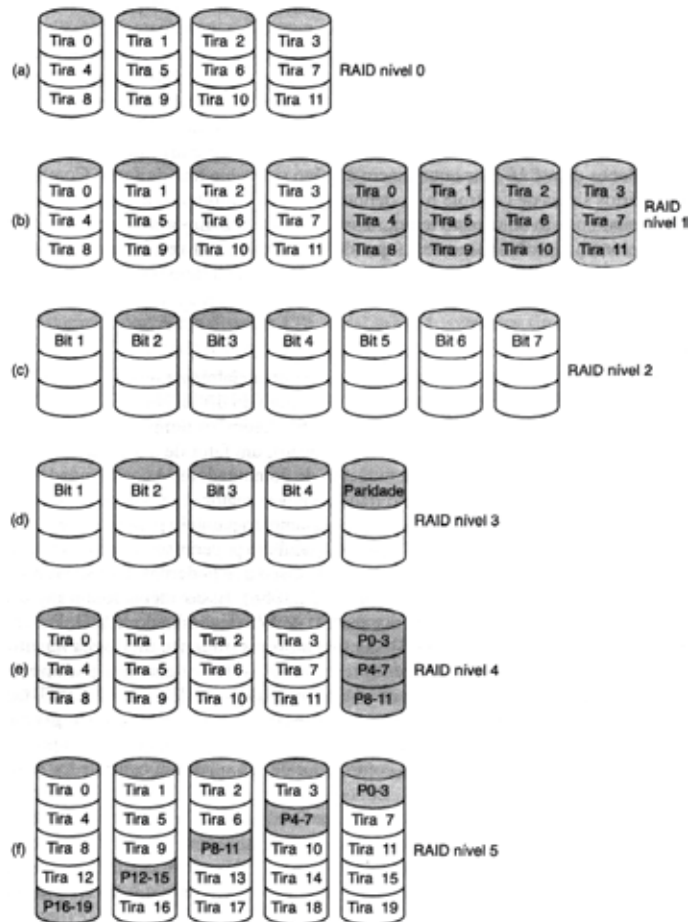


Figura 46 - RAID níveis 0 a 5

Fonte: TANENBAUM, 2007a.

CD e DVD

Além dos discos rígidos que são integrados ao computador, temos os discos removíveis, tais como CDs e DVDs, que são utilizados para armazenagem de dados e são considerados memória secundária. Os discos são lidos e gravados por meio de um dispositivo denominado **drive de dispositivo**.



Lembrete

Drive de dispositivo designa o dispositivo em si, por exemplo, *drive* de CD, *drive* de DVD. Não confundir com *driver*, *software* do dispositivo.

O CD substituiu o disco de vinil na década de 1980, primeiramente para a gravação de áudios e depois para a gravação de programas. Posteriormente, surgiram os CDs graváveis CD-R (*Compact Disc Recordable*) que só podiam ser gravados uma única vez e os regraváveis CD-RW (*Compact Disc Rewritable*). Atualmente, temos o DVD (*Digital Versatile Disk*) disco versátil digital (figura 47) que utiliza o mesmo desenho dos CDs, entretanto, possui maior capacidade de armazenamento.

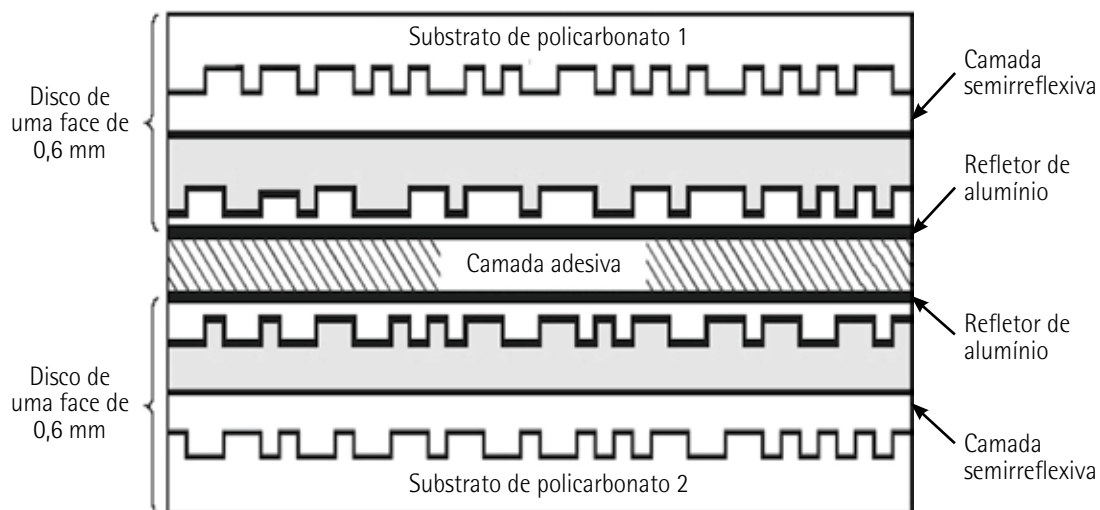


Figura 47 - DVD de dupla face e dupla camada

Fonte: TANENBAUM, 2007a.

No geral, os discos apresentam em sua superfície elementos químicos que, em contato com diferentes feixes de luz (*laser*), modificam-se, formando depressões ou planos (ver figura acima). Resumidamente, as depressões e os planos representam os 0 e 1.

Tanenbaum (2007a) elucida que os *drives* de CD-RW usam *laser* de três potências diferentes. Em alta potência, o *laser* funde a liga fazendo-a passar do estado cristalino de alta refletividade para o estado amorfo de baixa refletividade para representar uma depressão. Em potência média, a liga se funde e volta novamente ao seu estado natural cristalino para se tornar novamente um plano. Em baixa potência, o estado do material é sondado (para leitura), mas não ocorre transição de fase.



Lembrete

Atualmente, existem HDs externos ou removíveis.

O processo de leitura/gravação nos discos rígidos é magnético, nos CDs e DVDs é ótico.

Atualmente temos o *pen drive* que é um dispositivo removível para armazenamento de dados, portanto, também considerado uma memória secundária.

5.8 Entrada e saída

Nos capítulos anteriores explicamos sobre processador e memórias. Nos computadores pessoais, processador e memória principal se conectam por meio de barramentos da placa-mãe. Memórias

secundárias, tais como discos rígidos e *drives* de CD/DVD, não são encaixados à placa-mãe como a memória principal e o processador, mas conectados a esta normalmente por meio de cabos específicos.

Por sua vez, a placa-mãe é encaixada em um gabinete com fonte de alimentação e portas de comunicação. Este conjunto, mais os equipamentos de entrada e saída, compõem o sistema computacional.



Observação

Os equipamentos de entrada e saída são conhecidos como periféricos. Dentre os principais, temos: teclado, *mouse*, monitor e impressora. Estes dispositivos também são conectados ao restante do sistema computacional por meio de barramentos e portas de comunicação.

Cada dispositivo de E/S consiste de duas partes, conforme esclarece Tanenbaum (2007a): uma parte que compõe o dispositivo em si e outra que compõe o **controlador**.

A função principal do controlador é controlar seu dispositivo de E/S, garantindo o acesso ao barramento.

Alguns controladores acompanham a placa-mãe (*Chipset*), ao passo que outros precisam ser conectados a esta.

Chipset é um *chip* responsável pelo controle de diversos dispositivos de entrada e saída, além da comunicação do processador com a memória RAM, acesso a discos, entre outros. Geralmente é dividido em ponte norte e ponte sul.

Quando um programa necessita de dados de um CD, por exemplo, ele envia um comando para o controlador do *drive* de CD, o qual solicitará a leitura do CD e responderá ao controlador, que por sua vez escreverá a informação na memória principal, por meio do barramento. Neste exemplo, as informações passam do dispositivo direto para a memória, sem intervenção da CPU, portanto temos a informação em ciclo de E/S, mecanismo este conhecido por acesso direto à memória, ou DMA (*Direct Memory Access*).



Lembrete

Quando não há qualquer E/S em curso, a CPU pode ficar com todos os ciclos do barramento (ciclo de processador) para referenciar memória. Do contrário, a prioridade normalmente é de E/S, haja vista a possível perda de dados. Imagine o usuário inserindo informações no teclado, por exemplo.

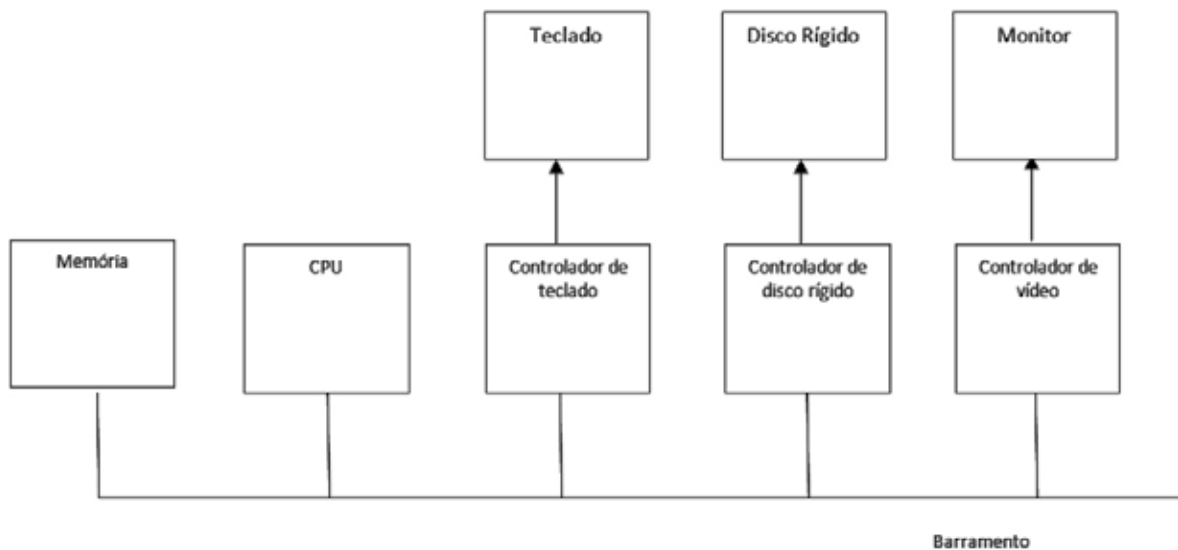


Figura 48 – Exemplo de barramento.

Fonte: TANENBAUM, 2007a.

O assunto barramentos será mais bem elaborado na Unidade III.

5.8.1 Teclado

Existem hoje no mercado diversos tipos de teclados, como os mecânicos, os eletromagnéticos, entre outros.

Este importante acessório é constituído basicamente de uma placa de circuito impresso abaixo de uma espécie de borracha e das teclas, na parte superior, que quando pressionadas fecham o circuito em determinado ponto, "gerando a informação".

Observação

Em computadores pessoais, quando uma tecla é pressionada, uma interrupção é gerada pelo sistema operacional, que lê as informações do controlador – o número de tecla (1 a 102) que foi pressionada.

Veja o código de caracteres ao final desta unidade.



Figura 49 – Exemplo de teclado.

5.8.2 Mouse

Em função da criação da interface gráfica, a utilização do *mouse* se tornou necessária, de modo que se tornou o meio mais comum de se apontar aquilo que se deseja na interface.

Atualmente, em substituição ao *mouse* com esferas, o tipo mais utilizado é o ótico, o qual possui funcionamento aparentemente simples. De acordo com Tanenbaum (2007a), o *mouse* ótico possui um diodo emissor de luz (LED) e um fotodetector e deve ser utilizado sobre uma almofada plástica com linhas de grade próximas. Quando a luz incide sobre a almofada, o fotodetector percebe o cruzamento entre linhas pela oscilação da luz refletida. Assim, um mecanismo interno do *mouse* conta as linhas cruzadas em cada direção.



Figura 50 – Exemplo de *mouse*

5.8.3 Monitor

Os monitores atuais são compostos de LCD (*Liquid Crystal Display*), ou seja, monitor de cristal líquido, em substituição aos antigos monitores de tubo CRT (*Cathode Ray Tube*, ou tubo de raios catódicos).



Figura 51 – Exemplo de monitor de CRT



Figura 52 – Monitor de LCD.

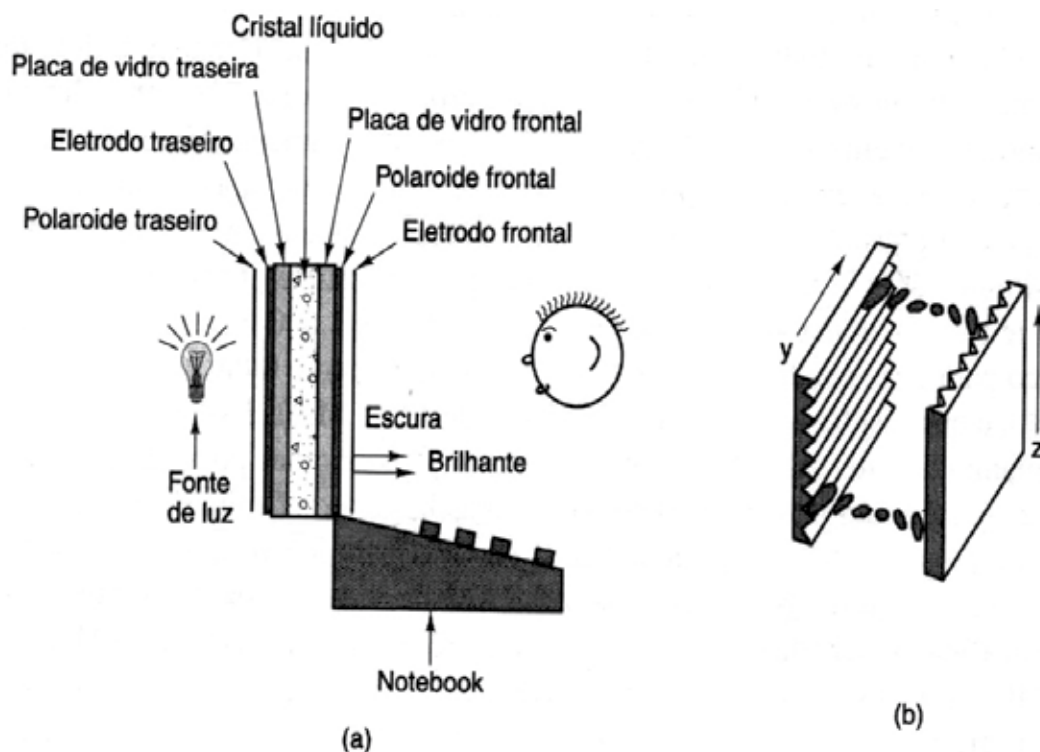


Figura 53 – Imagem de monitor de LCD.

Fonte: TANENBAUM, 2007a.

Cristais líquidos são moléculas orgânicas que se comportam como líquido e como cristal. Foram empregados pela primeira vez em visores de relógios e calculadoras. Resumidamente, a tecnologia empregada consiste em manter o cristal líquido entre placas de vidro e eletrodos, usando um campo elétrico aplicado que promove o alinhamento molecular dos cristais. Diferentes partes da tela recebem diferentes tensões elétricas, processo este que controla a imagem a ser apresentada.

5.8.4 Impressora

Existem diversos tipos de impressoras: monocromáticas ou coloridas, algumas mais lentas, outras mais rápidas, com baixa ou pouca qualidade de impressão, de uso pessoal ou para fins comerciais. As mais conhecidas são:

- **Impressora matricial:** impressora de impacto, cuja cabeça de impressão é composta por agulhas que, ao colidirem com uma fita impregnada com tinta, imprimem um ponto por agulha. É a partir desta combinação que os caracteres são criados. Monocromática: esta impressora utiliza formulários contínuos com carbono e é normalmente voltada para emissão de notas fiscais, pedidos, entre outros documentos. Apesar de antiga, ainda é utilizada por ser econômica.



Figura 54 – Impressora matricial

- **Impressora jato de tinta:** utiliza um cartucho que borrifa a tinta no papel por meio de minúsculos orifícios. Monocromáticas ou coloridas – sendo estas últimas as mais comuns –, é o tipo de impressora mais utilizada em escritórios e em ambiente doméstico.



Figura 55 – Impressora jato de tinta colorida

- **Impressora *laser*:** monocromáticas ou coloridas, são rápidas e com ótima qualidade de impressão. Funcionam de modo semelhante ao das fotocopadoras, com a utilização de toner, o qual consiste de um pó eletrostaticamente sensível que é impresso no papel por meio de fusão.



Figura 56 – Impressora a *laser* P&B

Na impressora a *laser* está presente um tambor fotossensível que recebe feixes de luz por meio de um *laser*. O feixe de luz é modelado para produzir um padrão de pontos escuros e claros. Ao girar, o tambor entra em contato com o toner, que será fixado apenas nos pontos que não receberam luz. Quando o tambor passa sobre a folha de papel, o pó é transferido e, posteriormente, o papel passa por um aquecedor chamado de fusor, fixando o toner na página, conforme a figura 57.

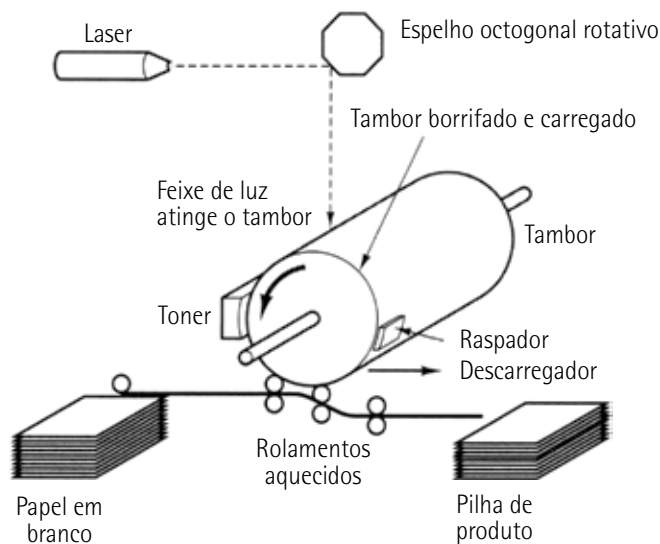


Figura 57 – Impressora a *laser*

Fonte: TANENBAUM, 2007a.

- **Plotter:** destinada à impressão de desenhos em grandes dimensões e com qualidade, tais como projetos de engenharia, plantas arquitetônicas e mapas cartográficos.

As *plotters* são especializadas em desenho vetorial, sendo comuns em estúdios de arquitetura.



Figura 58 – Plotter

5.9 Outros dispositivos de E/S

Além dos dispositivos mencionados, podemos citar outros, como: microfones, alto-falantes, *scanners*, *modems*, câmeras digitais, *webcams*, entre muitos outros equipamentos que surgem constantemente com o avanço tecnológico, tornando, portanto, impraticável a explanação de todos. Dessa maneira, detalharemos apenas os principais.



Saiba mais

<http://ci.ufpel.edu.br/treinamento/apostilas/nocoes_de_informatica/computador.pdf>

5.10 Código de caracteres

Como os computadores somente conseguem lidar com valores binários (0 e 1), ao lidar com caracteres e números, até mesmo os números precisam ser codificados. Assim, para permitir que os computadores consigam lidar com demais informações que não o sistema binário, a solução é a criação de *tabelas de conversão*.



Observação

Tabelas de conversão são chamadas de conjuntos de caracteres. Um dos padrões mais conhecidos é o ASCII (*American Standard Code for Information Interchange*), ou Código Padrão Americano para Intercâmbio de Informações.

O ASCII usa sete *bits* para descrever os caracteres, o que permite 128 combinações, conforme as tabelas 2 e 3.

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

Tabela 2 – Caracteres de 0 a 31

Fonte: TANENBAUM, 2007a.

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

Tabela 3 – Caracteres de 32 a 127

Fonte: TANENBAUM, 2007a.

O padrão ASCII é bastante limitado. Ele contém apenas os caracteres da língua inglesa, de modo que não contempla os acentos necessários para diversos idiomas e não abrange alfabetos diferentes, como o russo, o árabe e o chinês. Por esse motivo, foi criado um outro padrão, chamado UNICODE, que usa 16 *bits* para descrever cada caractere, o que permite até 65.536 caracteres diferentes, cobrindo cerca de 75 alfabetos. Linguistas estimam que seriam necessários cerca de 200.000 caracteres para suportar todos os idiomas falados atualmente.

6 SISTEMA DE NUMERAÇÃO

Será que o computador lê da mesma forma que nós, seres humanos, lemos?

Quando é necessário realizar uma conta, ou indicar um endereço de memória etc., nós lemos em sistema decimal, ou seja, $37 + 21 = 58$, pois é uma maneira mais fácil de leitura e memorização para as pessoas, sem falar que é o sistema de numeração padronizado no mundo inteiro. Esse sistema utiliza 10 números para a construção de todos os demais, esses números são:

0 1 2 3 4 5 6 7 8 9

O sistema decimal também é denominado de base 10, porém há também outros sistemas de numeração, como o sistema de base 2, base 8 e base 16. O sistema de base 2, ou binário, é formado somente por dois números que compõem todos os demais.

0 1

Já o sistema de base 8, sistema octal, possui oito números que formam os demais.

0 1 2 3 4 5 6 7

E por fim, o sistema de base 16, ou hexadecimal, possui 16 dígitos que resultam em todos os outros, sendo 10 números e 6 letras.

0 1 2 3 4 5 6 7 8 9 A B C D E F

Mas, passar para a base 10, independente do número?

Por exemplo, o número 2396 é decimal, então cada posição é multiplicada por 10, sendo que este é elevado ao valor da posição, começando de zero à direita.

2×10^3	→	2×1000	= 2000
3×10^2	→	3×100	= 300
9×10^1	→	9×10	= 90
6×10^0	→	6×1	= 6
Soma			= 2396

Mas, se tivermos o número 110010, sendo esse um número binário, é realizado o mesmo cálculo, porém, em vez de 10, troca-se por 2.

$$\begin{array}{lll} 1 \times 2^5 & \rightarrow & 1 \times 32 = 32 \\ 1 \times 2^4 & \rightarrow & 1 \times 16 = 16 \\ 0 \times 2^3 & \rightarrow & 0 \times 8 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ 0 \times 2^2 & \rightarrow & 0 \times 4 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ 1 \times 2^1 & \rightarrow & 1 \times 2 = 2 \\ 0 \times 2^0 & \rightarrow & 0 \times 1 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ \text{Soma} & \rightarrow & 32 + 16 + 2 = 50 \end{array}$$

Ou seja, o número binário 110010 é o mesmo que 50 em decimal.

Exemplos:

1110100 (binário) para decimal

$$\begin{array}{lll} 1 \times 2^6 & \rightarrow & 1 \times 64 = 64 \\ 1 \times 2^5 & \rightarrow & 1 \times 32 = 32 \\ 1 \times 2^4 & \rightarrow & 1 \times 16 = 16 \\ 0 \times 2^3 & \rightarrow & 0 \times 8 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ 1 \times 2^2 & \rightarrow & 1 \times 4 = 4 \\ 0 \times 2^1 & \rightarrow & 0 \times 2 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ 0 \times 2^0 & \rightarrow & 0 \times 1 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ \text{Soma} & \rightarrow & 64 + 32 + 16 + 4 = 116 \end{array}$$

1010101011 (binário) para decimal

$$\begin{array}{lll} 1 \times 2^9 & \rightarrow & 1 \times 512 = 512 \\ 0 \times 2^8 & \rightarrow & 0 \times 256 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ 1 \times 2^7 & \rightarrow & 1 \times 128 = 128 \\ 0 \times 2^6 & \rightarrow & 0 \times 64 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ 1 \times 2^5 & \rightarrow & 1 \times 32 = 32 \\ 0 \times 2^4 & \rightarrow & 0 \times 16 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ 1 \times 2^3 & \rightarrow & 1 \times 8 = 8 \\ 0 \times 2^2 & \rightarrow & 0 \times 4 = 0 \text{ (cálculo desnecessário, pois resulta em zero)} \\ 1 \times 2^1 & \rightarrow & 1 \times 2 = 2 \\ 1 \times 2^0 & \rightarrow & 1 \times 1 = 1 \\ \text{Soma} & \rightarrow & 512 + 128 + 32 + 8 + 2 + 1 = 683 \end{array}$$



Observação

Então, podemos converter o número que estiver em outra base para decimal? Sim, basta multiplicar pelo número da base elevado à posição, conforme foi realizado anteriormente.

O número 657 em octal para decimal:

$$6 \times 8^2 \rightarrow 6 \times 64 = 384$$

$$5 \times 8^1 \rightarrow 5 \times 8 = 40$$

$$7 \times 8^0 \rightarrow 7 \times 1 = 7$$

Somando, $384 + 40 + 7 = 431$

Então, o número octal 657 é 431 em decimal.

Exemplos:

7543 (octal) para decimal

$$7 \times 8^3 \rightarrow 7 \times 512 = 3584$$

$$5 \times 8^2 \rightarrow 5 \times 64 = 320$$

$$4 \times 8^1 \rightarrow 4 \times 8 = 32$$

$$3 \times 8^0 \rightarrow 3 \times 1 = 3$$

Somando: $3584 + 320 + 32 + 3 = 3939$

Então, o número octal 7543 é 3939 em decimal.

50126 (octal) para decimal

$$5 \times 8^4 \rightarrow 5 \times 4096 = 20480$$

$$0 \times 8^3 \rightarrow 0 \times 512 = 0$$

$$1 \times 8^2 \rightarrow 1 \times 64 = 64$$

$$2 \times 8^1 \rightarrow 2 \times 8 = 16$$

$$6 \times 8^0 \rightarrow 6 \times 1 = 6$$

Somando: $20480 + 0 + 64 + 16 + 6 = 20566$

Então, o número octal 50126 é 20566 em decimal.

11042 (octal) para decimal

$$\begin{array}{lll} 1 \times 8^4 & \rightarrow & 1 \times 4096 = 4096 \\ 1 \times 8^3 & \rightarrow & 1 \times 512 = 512 \\ 0 \times 8^2 & \rightarrow & 0 \times 64 = 0 \\ 4 \times 8^1 & \rightarrow & 4 \times 8 = 32 \\ 2 \times 8^0 & \rightarrow & 2 \times 1 = 2 \end{array}$$

Somando: $4096 + 512 + 32 + 2 = 4642$

Então, o número octal 11042 é 4642 em decimal.

Qual é o número, em decimal, de 45DA7 em hexadecimal?

$$\begin{array}{lll} 4 \times 16^4 & \rightarrow & 4 \times 65536 = 262144 \\ 5 \times 16^3 & \rightarrow & 5 \times 4096 = 20480 \\ 13 \times 16^2 & \rightarrow & 13 \times 256 = 3328 \\ 10 \times 16^1 & \rightarrow & 10 \times 16 = 160 \\ 7 \times 16^0 & \rightarrow & 7 \times 1 = 7 \end{array}$$

Somando: $262144 + 20480 + 3328 + 160 + 7 = 286119$ em decimal

Exemplos:

4E21D (hexadecimal) para decimal

$$\begin{array}{lll} 4 \times 16^4 & \rightarrow & 4 \times 65536 = 262144 \\ 14 \times 16^3 & \rightarrow & 14 \times 4096 = 57344 \\ 2 \times 16^2 & \rightarrow & 2 \times 256 = 512 \\ 1 \times 16^1 & \rightarrow & 1 \times 16 = 16 \\ 13 \times 16^0 & \rightarrow & 13 \times 1 = 13 \end{array}$$

Somando: $262144 + 57344 + 512 + 16 + 13 = 320029$

2F3B4 (hexadecimal) para decimal

$$\begin{array}{lll} 2 \times 16^4 & \rightarrow & 2 \times 65536 = 131072 \\ 15 \times 16^3 & \rightarrow & 15 \times 4096 = 61440 \\ 3 \times 16^2 & \rightarrow & 3 \times 256 = 768 \\ 11 \times 16^1 & \rightarrow & 11 \times 16 = 176 \\ 4 \times 16^0 & \rightarrow & 4 \times 1 = 4 \end{array}$$

Somando: $131072 + 61440 + 768 + 176 + 4 = 193460$

Vimos a conversão de números binários, octal e hexadecimal para decimal. Agora, será apresentada a conversão para o número binário.

6.1 Conversão para binário

Para converter um número decimal em binário, primeiro pega-se o número decimal e divide-se por 2 até o quociente ser 0. Então, retiram-se os restos na ordem inversa.

Exemplos: número decimal, 568

$$\begin{array}{r}
 568 \overline{) 2} \\
 0 \quad 284 \overline{) 2} \\
 \quad 0 \quad 142 \overline{) 2} \\
 \quad \quad 0 \quad 71 \overline{) 2} \\
 \quad \quad \quad 1 \quad 35 \overline{) 2} \\
 \quad \quad \quad \quad 1 \quad 17 \overline{) 2} \\
 \quad \quad \quad \quad \quad 1 \quad 8 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad 0 \quad 4 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad 0 \quad 2 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad 0 \quad 1 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad \quad 1 \quad 0
 \end{array}$$

Para pegar o número binário, são retirados de baixo para cima todos os restos, ficando: 1000111000

$$\begin{array}{r}
 431 \overline{) 2} \\
 1 \quad 215 \overline{) 2} \\
 \quad 1 \quad 107 \overline{) 2} \\
 \quad \quad 1 \quad 53 \overline{) 2} \\
 \quad \quad \quad 1 \quad 26 \overline{) 2} \\
 \quad \quad \quad \quad 0 \quad 13 \overline{) 2} \\
 \quad \quad \quad \quad \quad 1 \quad 6 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad 0 \quad 3 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad 1 \quad 1 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad 1 \quad 0
 \end{array}$$

Para pegar o número binário, são retirados de baixo para cima todos os restos, ficando: 110101111

$$\begin{array}{r}
 652 \overline{) 2} \\
 0 \quad 326 \overline{) 2} \\
 \quad 0 \quad 163 \overline{) 2} \\
 \quad \quad 1 \quad 81 \overline{) 2} \\
 \quad \quad \quad 1 \quad 40 \overline{) 2} \\
 \quad \quad \quad \quad 0 \quad 20 \overline{) 2} \\
 \quad \quad \quad \quad \quad 0 \quad 10 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad 0 \quad 5 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad 1 \quad 2 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad 0 \quad 1 \overline{) 2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad \quad 1 \quad 0
 \end{array}$$

Para pegar o número binário, são retirados de baixo para cima todos os restos, ficando: 1010001100

6.1.1 Octal para binário

O número octal é separado dígito a dígito. Exemplo: 745 em octal.

7
4
5

Cada dígito do número octal corresponderá a 3 dígitos em binário, sabendo que o 1º. dígito binário é multiplicado por 1, o 2º. dígito por 2, e o 3º. dígito por 4, então soma-se, sendo que o resultado máximo será 7, pois, sendo binário, o maior número será 111.

4	2	1
3º. dígito	2º. dígito	1º. dígito

Multiplicador	4	2	1
Dígito	1	1	1

Resultado:

$1 \times 4 = 4$
 $1 \times 2 = 2$
 $1 \times 1 = 1$

Somando: $4 + 2 + 1 = 7$

$7 = 111$

Agora, aplica-se o mesmo cálculo para chegar em 4 e depois 5.

$4 = 100$
 $5 = 101$

Colocando em ordem, o número binário ficará: 111100101

Exemplos:

623 (octal) para binário

$6 = 110$
 $2 = 010$
 $3 = 011$

Colocando em ordem, o número binário ficará: 110010011.

140 (octal) para binário

1 = 001

4 = 100

0 = 000

Colocando em ordem, o número binário ficará: 001100000.

476 (octal) para binário

4 = 100

7 = 111

6 = 110

Colocando em ordem, o número binário ficará: 100111110.

6.1.2 Hexadecimal para binário

É similar ao octal, porém, em vez de 3 dígitos binários para cada dígito hexadecimal, teremos 4 dígitos binários.

8	4	2	1
4º. dígito	3º. dígito	2º. dígito	1º. dígito

Número hexadecimal: A57E

A = 1010

5 = 0101

7 = 0111

E = 1110

Ficando: 101001010111110

Exemplos:

A57E (hexadecimal) para binário

D = 1101

3 = 0011

6 = 0110

2 = 0010

Ficando: 1101001101100010

6FBC8 (hexadecimal) para binário

6 = 0110

F = 1111

B = 1011

C = 1100

8 = 1000

Ficando: 01101111101111001000

A seguir, veremos a conversão para o número octal.

Para converter de decimal para octal, faremos da seguinte forma: tomemos o número decimal 454, então o dividamos por 8.

$$\begin{array}{r}
 454 \overline{) 8} \\
 \underline{6} 56 \overline{) 8} \\
 0 7 \overline{) 8} \\
 7 0
 \end{array}$$

Então, pegam-se os restos 7 0 6 e forma-se o número 706, que é o número em octal.

Exemplos:

2967 (decimal) para octal

$$\begin{array}{r}
 2967 \overline{) 8} \\
 7 370 \overline{) 8} \\
 2 46 \overline{) 8} \\
 6 5 \overline{) 8} \\
 5 0
 \end{array}$$

Resultado em octal: 5627

9153846720 (decimal) para octal

$$\begin{array}{r}
 9153846720 \overline{) 8} \\
 0 1144230840 \overline{) 8} \\
 0 143028855 \overline{) 8} \\
 7 17878606 \overline{) 8} \\
 6 2234825 \overline{) 8}
 \end{array}$$

Continuando

$$\begin{array}{r}
 6 \ 2234825 \mid \underline{8} \\
 1 \ 279353 \mid \underline{8} \\
 1 \ 34919 \mid \underline{8} \\
 7 \ 4364 \mid \underline{8} \\
 4 \ 545 \mid \underline{8} \\
 1 \ 68 \mid \underline{8} \\
 4 \ 8 \mid \underline{8} \\
 0 \ 1 \mid \underline{8} \\
 1 \ 0
 \end{array}$$

Resultado em octal: 104147116700

6.1.3 Binário para octal

Para transformar um número binário para octal, é simples: basta pegar o número binário e dividi-lo em grupos de 3 dígitos. Exemplo:

1011001
1 011 001

Agora, fazer a conta:

$$\begin{array}{l}
 1 \\
 1 \times 2^0 \rightarrow 1 \times 1 = 1
 \end{array}$$

$$\begin{array}{l}
 011 \\
 0 \times 2^2 \rightarrow 0 \times 4 = 0 \\
 1 \times 2^1 \rightarrow 1 \times 2 = 2 \\
 1 \times 2^0 \rightarrow 1 \times 1 = 1 \\
 \text{Somando} = 3
 \end{array}$$

$$\begin{array}{l}
 001 \\
 0 \times 2^2 \rightarrow 0 \times 4 = 0 \\
 0 \times 2^1 \rightarrow 0 \times 2 = 0 \\
 1 \times 2^0 \rightarrow 1 \times 1 = 1 \\
 \text{Somando} = 1
 \end{array}$$

Então, temos 131 em octal.

Exemplos:

1101010101 (binário) para octal

Dividindo em grupos de 3 números: 1 101 010 101

Agora, fazer a conta:

$$\begin{array}{r} 1 \\ 1 \times 2^0 \end{array} \rightarrow 1 \times 1 = 1$$

$$\begin{array}{r} 101 \\ 1 \times 2^2 \\ 0 \times 2^1 \\ 1 \times 2^0 \end{array} \rightarrow \begin{array}{r} 1 \times 4 = 4 \\ 0 \times 2 = 0 \\ 1 \times 1 = 1 \end{array}$$

Somando esse grupo = 5

$$\begin{array}{r} 010 \\ 0 \times 2^2 \\ 1 \times 2^1 \\ 0 \times 2^0 \end{array} \rightarrow \begin{array}{r} 0 \times 4 = 0 \\ 1 \times 2 = 2 \\ 0 \times 1 = 0 \end{array}$$

Somando esse grupo = 2

$$\begin{array}{r} 101 \\ 1 \times 2^2 \\ 0 \times 2^1 \\ 1 \times 2^0 \end{array} \rightarrow \begin{array}{r} 1 \times 4 = 4 \\ 0 \times 2 = 0 \\ 1 \times 1 = 1 \end{array}$$

Somando esse grupo = 5

Então, temos 1525 em octal.

010000111100110 (binário) para octal

Dividindo em grupos de 3 números: 010 000 111 100 110

Agora, fazer a conta:

$$\begin{array}{r} 010 \\ 0 \times 2^2 \\ 1 \times 2^1 \\ 0 \times 2^0 \end{array} \rightarrow \begin{array}{r} 0 \times 4 = 0 \\ 1 \times 2 = 2 \\ 0 \times 1 = 0 \end{array}$$

Somando esse grupo = 2

$$\begin{array}{r} 000 \\ 0 \times 2^2 \\ 0 \times 2^1 \\ 0 \times 2^0 \end{array} \rightarrow \begin{array}{r} 0 \times 4 = 0 \\ 0 \times 2 = 0 \\ 0 \times 1 = 0 \end{array}$$

Somando esse grupo = 0

111

$$1 \times 2^2 \rightarrow 1 \times 4 = 4$$

$$1 \times 2^1 \rightarrow 1 \times 2 = 2$$

$$1 \times 2^0 \rightarrow 1 \times 1 = 1$$

Somando esse grupo = 7

100

$$1 \times 2^2 \rightarrow 1 \times 4 = 4$$

$$0 \times 2^1 \rightarrow 0 \times 2 = 0$$

$$0 \times 2^0 \rightarrow 0 \times 1 = 0$$

Somando esse grupo = 4

110

$$1 \times 2^2 \rightarrow 1 \times 4 = 4$$

$$1 \times 2^1 \rightarrow 1 \times 2 = 2$$

$$0 \times 2^0 \rightarrow 0 \times 1 = 0$$

Somando esse grupo = 6

Então, temos 20746 em octal.

Hexadecimal para octal: transforma-se para binário e depois para octal.

6.1.4 Conversão para hexadecimal

Decimal para hexadecimal

Tomemos o número decimal 454, então o dividamos por 16.

$$\begin{array}{r} 454 \overline{) 16} \\ 6 \quad 28 \overline{) 16} \\ 12 \quad 1 \overline{) 16} \\ 1 \quad 0 \end{array}$$

Então, pegam-se os restos de baixo para cima 1 12 6 e forma-se o número hexadecimal, porém lembrando:

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

Então, o número hexadecimal fica 1C6.

Exemplos:

5782 (decimal) para hexadecimal

$$\begin{array}{r}
 5782 \overline{) 16} \\
 \underline{6 \quad 361} \overline{) 16} \\
 \quad \underline{9 \quad 22} \overline{) 16} \\
 \quad \quad \underline{6 \quad 1} \overline{) 16} \\
 \quad \quad \quad \underline{1 \quad 0}
 \end{array}$$

Resultado: 1696

894672 (decimal) para hexadecimal

$$\begin{array}{r}
 894672 \overline{) 16} \\
 \underline{0 \quad 55917} \overline{) 16} \\
 \quad \underline{13 \quad 3494} \overline{) 16} \\
 \quad \quad \underline{6 \quad 218} \overline{) 16} \\
 \quad \quad \quad \underline{10 \quad 13} \overline{) 16} \\
 \quad \quad \quad \quad \underline{13 \quad 0}
 \end{array}$$

Resultado: DA6D0

6.1.5 Binário para hexadecimal

Para transformar um número binário para octal é simples: basta pegar o número binário e dividi-lo em grupos de 4 dígitos. Exemplo:

1011101
101 1101

Agora, fazer a conta:

$$\begin{array}{lcl}
 101 & & \\
 1 \times 2^2 & \rightarrow & 1 \times 4 = 4 \\
 0 \times 2^1 & \rightarrow & 0 \times 2 = 0 \\
 1 \times 2^0 & \rightarrow & 1 \times 1 = 1 \\
 \text{Somando} & = & 5
 \end{array}$$

$$\begin{array}{lcl}
 1101 & & \\
 1 \times 2^3 & \rightarrow & 1 \times 8 = 8 \\
 1 \times 2^2 & \rightarrow & 1 \times 4 = 4 \\
 0 \times 2^1 & \rightarrow & 0 \times 2 = 0 \\
 1 \times 2^0 & \rightarrow & 1 \times 1 = 1 \\
 \text{Somando} & = & 13
 \end{array}$$

Então, temos 5D em hexadecimal.

Exemplos:

10001011010101 (binário) para hexadecimal

Separando: 10 0010 1101 0101

Agora, fazer a conta:

10

$$1 \times 2^1 \rightarrow 1 \times 2 = 2$$

$$0 \times 2^0 \rightarrow 0 \times 1 = 0$$

Somando = 2

0010

$$0 \times 2^3 \rightarrow 0 \times 8 = 0$$

$$0 \times 2^2 \rightarrow 0 \times 4 = 0$$

$$1 \times 2^1 \rightarrow 1 \times 2 = 2$$

$$0 \times 2^0 \rightarrow 0 \times 1 = 0$$

Somando = 2

1101

$$1 \times 2^3 \rightarrow 1 \times 8 = 8$$

$$1 \times 2^2 \rightarrow 1 \times 4 = 4$$

$$0 \times 2^1 \rightarrow 0 \times 2 = 0$$

$$1 \times 2^0 \rightarrow 1 \times 1 = 1$$

Somando = 13

0101

$$0 \times 2^3 \rightarrow 0 \times 8 = 0$$

$$1 \times 2^2 \rightarrow 1 \times 4 = 4$$

$$0 \times 2^1 \rightarrow 0 \times 2 = 0$$

$$1 \times 2^0 \rightarrow 1 \times 1 = 1$$

Somando = 5

Então, temos 22D5 em hexadecimal.

11100111101111001010 (binário) para hexadecimal

Separando: 1110 0111 1011 1100 1010

Agora, fazer a conta:

1110

$$1 \times 2^3 \rightarrow 1 \times 8 = 8$$

$$1 \times 2^2 \rightarrow 1 \times 4 = 4$$

$$1 \times 2^1 \rightarrow 1 \times 2 = 2$$

$$0 \times 2^0 \rightarrow 0 \times 1 = 0$$

Somando = 14

0111

$$0 \times 2^3 \rightarrow 0 \times 8 = 0$$

$$1 \times 2^2 \rightarrow 1 \times 4 = 4$$

$$1 \times 2^1 \rightarrow 1 \times 2 = 2$$

$$1 \times 2^0 \rightarrow 1 \times 1 = 1$$

Somando = 7

1011

$$1 \times 2^3 \rightarrow 1 \times 8 = 8$$

$$0 \times 2^2 \rightarrow 1 \times 4 = 0$$

$$1 \times 2^1 \rightarrow 0 \times 2 = 2$$

$$1 \times 2^0 \rightarrow 1 \times 1 = 1$$

Somando = 11

1100

$$1 \times 2^3 \rightarrow 1 \times 8 = 8$$

$$1 \times 2^2 \rightarrow 1 \times 4 = 4$$

$$0 \times 2^1 \rightarrow 0 \times 2 = 0$$

$$0 \times 2^0 \rightarrow 0 \times 1 = 0$$

Somando = 12

1010

$$1 \times 2^3 \rightarrow 1 \times 8 = 8$$

$$0 \times 2^2 \rightarrow 0 \times 4 = 0$$

$$1 \times 2^1 \rightarrow 1 \times 2 = 2$$

$$0 \times 2^0 \rightarrow 0 \times 1 = 0$$

Somando = 10

Então, temos E7BCA em hexadecimal.

6.1.6 Octal para hexadecimal

Transforma-se para binário e depois em hexadecimal.

Hexadecimal	Decimal	Octal	Binário
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111

Tabela 4 - Quadro de conversão

6.2 Cálculo binário

O cálculo binário tem algumas regras, que são:

6.2.1 Soma binário

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ e vai } 1$$

Exemplos:

a)

$$\begin{array}{r} 1101 \\ +110 \\ \hline 10011 \end{array}$$

Esse cálculo é realizado no mesmo padrão do decimal: da direita para a esquerda, soma-se 1 com 0, resulta em 1; depois 0 com 1, resulta em 1; depois 1 com 1, resulta em 0 e sobe 1.



Lembrete

Esse *bit* 1 que sobe somará com o 1 à esquerda do primeiro número, o que resultará 0 e sobrá 1.

b)

$$\begin{array}{r} 1011 \\ +100 \\ \hline 1111 \end{array}$$

c)

$$\begin{array}{r} 1111 \\ +111 \\ \hline 10110 \end{array}$$

d)

$$\begin{array}{r} 1100 \\ +101 \\ \hline 10001 \end{array}$$

e)

$$\begin{array}{r} 1010 \\ +111 \\ \hline 10001 \end{array}$$

f)

$$\begin{array}{r} 1110 \\ +101 \\ \hline 10011 \end{array}$$

6.2.2 Subtração binário

$$0 - 0 = 0$$

$$0 - 1 = 1, \text{ e vai } 1$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Exemplos:

a)

$$\begin{array}{r} 1101 \\ -110 \\ \hline 0111 \end{array}$$

Aqui, o primeiro cálculo foi 1-0, resultando em 1; após 0 -1, esse resultará em 1 e sobe 1. Então, teremos 1-1-1. Aqui, vamos "quebrar" a conta: façamos 1-1, o que resulta em 0, e agora esse 0 -1 (1 que foi deixado de lado), o que resulta em 1 e sobe 1. Por último, temos 1-1, sendo 0 o resultado.

b)

$$\begin{array}{r} 1111 \\ -110 \\ \hline 1001 \end{array}$$

c)

$$\begin{array}{r} 1000 \\ -101 \\ \hline 0011 \end{array}$$

d)

$$\begin{array}{r} 1100 \\ -101 \\ \hline 0111 \end{array}$$

e)

$$\begin{array}{r} 1101 \\ -111 \\ \hline 0110 \end{array}$$

f)

$$\begin{array}{r} 1011 \\ -111 \\ \hline 0100 \end{array}$$



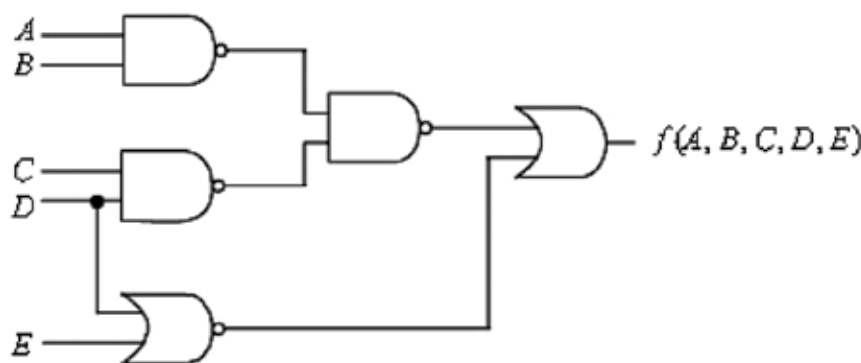
Resumo

Aprendemos como as memórias funcionam. O que é cálculo binário e como funciona a lógica matemática do computador. Enfim, como funciona o sistema de numeração do computador. Como fazemos conversões entre sistemas de numerações distintos. Entendemos o funcionamento lógico-matemático do computador.



Exercícios

Questão 1. (ENADE 2008). No circuito acima, que possui cinco entradas – A, B, C, D e E – e uma saída $f(A, B, C, D, E)$, qual opção apresenta uma expressão lógica equivalente à função $f(A, B, C, D, E)$?



Figura

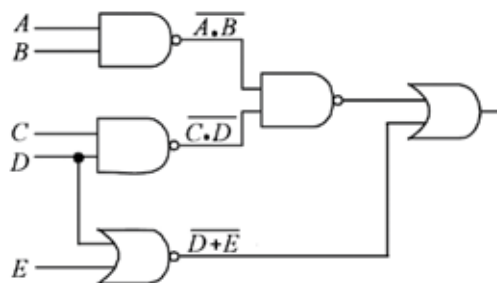
- A) $\overline{A.B} + \overline{C.D} + D.E$
- B) $(A + B).(C + D) + D.E$
- C) $\overline{A.B} + \overline{C.D} + D + E$
- D) $A.B + C.D + D + E$
- E) $A.B + C.D + \overline{D.E}$

Alternativa correta: E

Análise das alternativas

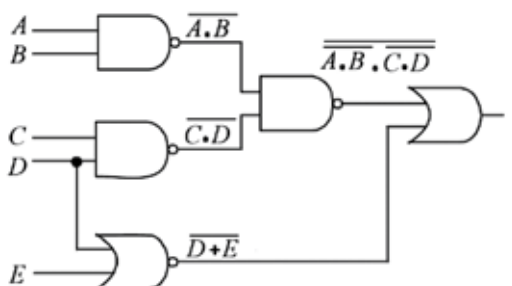
Justificativa. Para resolver a questão vamos utilizar um modelo gráfico onde os sinais de entrada (A, B, C, D e E), vão se propagar em cada nível lógico através de suas portas lógicas.

Nível 1



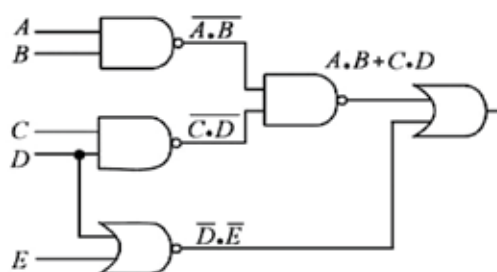
Figura

Nível 2



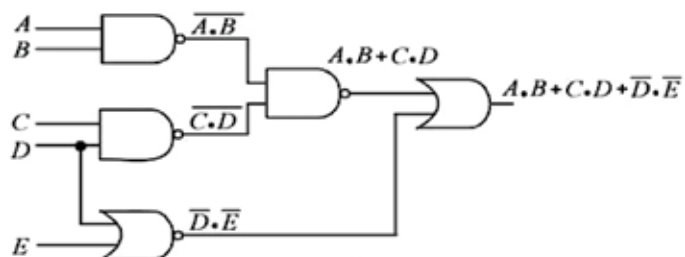
Figura

Nível 3



Figura

Nível 4



Figura

Questão 2. Durante a evolução do homem, uma série de símbolos foram sendo incorporados ao seu cotidiano. Um numeral pode ser representado por um ou mais símbolos para representar um número. Os números, assim como as palavras, se diferenciam das coisas a que se referem. Os símbolos "10", "dez" e "X" são numerais diferentes, que representam o mesmo número. Será que o computador lê da mesma forma com que nós seres humanos lemos? O ser humano utiliza o sistema decimal, que é composto pelos números 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ou seja, é composto por dez números do zero ao nove, que podem ser combinados para a construção de todos os decimais. O sistema decimal, também é denominado de base 10. O sistema binário utiliza apenas os números 0 e 1, sendo denominado de base 2. O sistema de base 8 ou sistema octal, possuem oito números 0, 1, 2, 3, 4, 5, 6, 7. O sistema de base 16, ou hexadecimal, possui 16 dígitos que resultam em todos os outros, sendo 10 números e 6 letras, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Convertendo o número decimal 60 para binário, octal e hexadecimal, qual a alternativa certa.

- A) 111100, 47 , C3
B) 100110, 07, C3
C) 111100, 70, 30
D) 111001, 71, 39
E) 111100, 74, 3C

Resolução desta questão na Plataforma.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.