

Infraestrutura de Hardware

**Juliana Regueira Basto Diniz
Abner Barros**

Caderno de Exercícios do Volume 1

Recife, 2009

Universidade Federal Rural de Pernambuco



Reitor: Prof. Valmar Corrêa de Andrade

Vice-Reitor: Prof. Reginaldo Barros

Pró-Reitor de Administração: Prof. Francisco Fernando Ramos Carvalho

Pró-Reitor de Extensão: Prof. Paulo Donizeti Siepierski

Pró-Reitor de Pesquisa e Pós-Graduação: Prof. Fernando José Freire

Pró-Reitor de Planejamento: Prof. Rinaldo Luiz Caraciolo Ferreira

Pró-Reitora de Ensino de Graduação: Profª. Maria José de Sena

Coordenação Geral de Ensino a Distância: Profª Marizete Silva Santos

Produção Gráfica e Editorial

Capa e Editoração: Allyson Vila Nova, Rafael Lira, Italo Amorim e Everton Felix

Revisão Ortográfica: Marcelo Melo

Ilustrações:

Coordenação de Produção: Marizete Silva Santos

Sumário

Apresentação	4
Exercícios Propostos do Capítulo 1	5
Exercícios Propostos do Capítulo 2	7
Exercícios Propostos do Capítulo 3	13
Conheça os Autores	17

Apresentação

Caro(a) Cursista,

Este é o seu caderno de exercícios referente aos assuntos tratados no Módulo 1 da disciplina de Infraestrutura de Hardware. Nele você encontrará uma série de exercícios elaborados de forma a explorar os assuntos apresentados, distribuídos em diversos graus de dificuldade, indo desde exercícios mais simples, que você deverá resolver sem maiores dificuldades, até exercícios mais complexos, para os quais você talvez tenha que solicitar a ajuda do Tutor da disciplina. Lembre-se, é de suma importância que você resolva integralmente todos os exercícios apresentados, pois eles refletem o conhecimento que esperamos que você tenha adquirido ao estudar os assuntos abordados neste módulo.

Ao iniciar seus estudos tenha em mente que tão importante quanto o tempo que você dedica para seus estudos é a qualidade do tempo dedicado. Procure ter um local quieto e sossegado para estudar, longe de situações que lhe causem desconcentração.

Lembre-se que existe uma equipe de Professores e Tutores prontos para responder qualquer dúvida que possa surgir, consulte-os sempre que achar necessário.

Bons estudos!

Juliana Regueira Basto Diniz

Abner Correa Barros

Professores Autores

Exercícios Propostos do Capítulo 1



Vamos Revisar?

Neste capítulo, você estudou as funções do computador e os seus componentes principais. Foram apresentados os subsistemas internos ao computador: Central de Processamento ou subsistema de processamento, *Subsistema de Memória*, *Subsistema de Entrada e Saída (E/S)* e *Subsistema de Interconexão*.

Aprendemos que a CPU possui o papel de controlar as operações do computador e processar dados. O *Subsistema de Memória* armazena dados e o *Subsistema de Entrada e Saída (E/S)* transfere dados entre o computador e o ambiente externo. Por fim, o *Subsistema de Interconexão*, responsabiliza-se pela comunicação entre a CPU, memória principal e dispositivos de E/S.

Em seguida, você pôde observar a evolução histórica dos computadores, partindo do primeiro computador eletrônico até a miniaturização dos computadores e da computação ubíqua. Aprendemos que esse conceito baseia-se na computação a toda hora e em todo o lugar, utilizando uma grande diversidade de dispositivos.



Aprenda Praticando

1. De acordo com o que você leu no capítulo 1, quais as funções básicas realizadas pelo computador?
2. Baseando-se naquilo que você leu, quais as diferenças fundamentais entre a organização e a arquitetura de computadores?
3. Quais os componentes internos a CPU? Qual a finalidade de cada um deles?
4. Considerando a evolução dos computadores, explique o que você entende por computação ubíqua.

5. Faça um paralelo entre os computadores da terceira e quarta geração.

Exercícios Propostos do Capítulo 2



Vamos Revisar?

No Capítulo 2, você aprendeu sobre as Bases Numéricas e a Notação Proposicional. Aprendeu também que as duas bases numéricas mais importantes são:

- » A **Base Decimal**, formada por dez símbolos $\{0,1,2,3,4,5,6,7,8,9\}$, utilizada por nós, seres humanos.
- » A **Base Binária**, formada apenas pelos símbolos 0 e 1, utilizada pela Unidade Lógica e Aritmética dos Computadores.

Além destas bases, você conheceu também as bases Octal e Hexadecimal que, como você deve estar lembrado, foram criadas com o intuito de facilitar a visualização e a operação por nós seres humanos de valores expressos na base binária.

Neste Capítulo você também pode aprender diversas técnicas e métodos para a conversão de valores expressos em diferentes bases numéricas, e que apesar de todos os valores inteiros poderem ser expressos em qualquer das bases numéricas estudadas, nem todo valor decimal pode ser livremente convertido entre estas bases, uma vez que alguns destes tornam-se dízimas periódicas quando passados de uma base para outra.

Por fim você aprendeu que apesar das operações aritméticas serem integralmente preservadas nas quatro bases numéricas apresentadas, você deve ter uma atenção especial com o valor no qual ocorre o “vai um” nestas operações, principalmente quando muda de uma base numérica para outra.



Aprenda Praticando

Notação Posicional

1. Preencha a tabela a seguir com os valores que estão faltando:

Decimal	Hexadecimal	Octal	Binário (8 bits)
0	0	0	00000000
1		1	
2			
3	3		
4		4	
5			00000101
6	6		
7		7	
8		10	
9			00001001
10	A		
11			
12		12	
13			
14			
15	F		
16	10	20	
17			
18			
19			
20	14		
21			
22			
23			
24	18	30	00011000

2. Verifique quanto vale cada um dos números a seguir na base decimal utilizando a Notação Posicional. Faça conforme os itens já resolvidos.

$$a) 1232 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0$$

$$= 1000 + 200 + 30 + 2 = 1232$$

$$b) 1453 =$$

$$c) 789 =$$

$$d) 125_8 = 1 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 = 64 + 16 + 5 = 85$$

$$e) 1453_8 =$$

$$f) 143_8 =$$

$$g) 53_8 =$$

$$h) 101101_2 = 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 = 32 + 8 + 4 + 1 = 45$$

$$i) 11101101_2 =$$

$$j) 00110101_2 =$$

$$k) 10101010_2 =$$

$$l) AB3F_{16} = A \cdot 16^3 + B \cdot 16^2 + 3 \cdot 16^1 + F \cdot 16^0$$

$$= 40960 + 2816 + 48 + 15 = 43839$$

$$m) FF3_{16} =$$

$$n) ABC_{16} =$$

$$o) AAA_{16} =$$

$$p) 1210_{16} =$$

3. Verifique quanto vale cada um dos números fracionários a seguir na base decimal utilizando a Notação Posicional. Faça conforme os itens já resolvidos.

$$a) 10,325 = 1 \cdot 10^1 + 3 \cdot 10^{-1} + 2 \cdot 10^{-2} + 5 \cdot 10^{-3}$$

$$= 10 + 0,3 + 0,02 + 0,005 = 10,325$$

$$b) 1,25 =$$

$$c) 123,385 =$$

$$d) 11,111_2 = 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$= 2 + 1 + 0,5 + 0,25 + 0,125 = 3,875$$

e) $100,01_2 =$

f) $1010,101_2 =$

g) $1100,001_2 =$

Conversões entre Bases Numéricas

1. Converta os números a seguir da base decimal para a base binária utilizando os métodos de divisão sucessivas e conversão direta.

a) $12 =$

b) $45 =$

c) $121 =$

d) $87 =$

e) $135 =$

f) $17 =$

g) $76 =$

2. Preencha a tabela a seguir convertendo os números dados da base decimal para a base binária nas notações de complemento a dois e magnitude e sinal utilizando palavras de 8 bits.

Decimal	Magnitude e Sinal	Complemento a dois
0	0	00000000
-1	10000001	11111111
1		
-35		
47		
-55		
-83		
-27		
27		

3. Preencha a tabela a seguir convertendo os números para as bases que forem necessárias

Decimal	Octal	Hexadecimal	Binário (8bits)
	252	AA	10101010
			10000110
	35		
232			
			10010010
	238		
			10010011
135			
		35	
	43		
	12		
121			
			00110101
		19	

Operações Aritméticas

1. Efetue as operações aritméticas a seguir na base numérica indicada. Após efetuar cada operação, converta o resultado obtido para a base decimal e verifique se está correto.

a) Base binária, 8 bits, notação de complemento a dois

- » $23 + 45$
- » $34 - 83$
- » $67 - 27$
- » $13 + 17 + 10$

b) Base Hexadecimal

- » $25 + 32$
- » $121 + 7$
- » $81 + 48$
- » $39 + 75$

c) Base Octal

» $87 + 7$

» $121 + 232$

» $53 + 31$

» $35 + 21$

Exercícios Propostos do Capítulo 3



Vamos Revisar?

No capítulo 3, você aprendeu sobre Lógica Digital e Álgebra de Boole, ou Álgebra Booleana. Descobriu que a Lógica Digital está baseada na Álgebra de Boole, a qual possui três Funções Lógicas Essenciais, as funções And, Or e Not e que a partir destas foram definidas mais três Funções Lógicas Básicas, as funções Nand, Nor, e Xor. Descobriu também que as portas lógicas são circuitos eletrônicos que implementam as funções lógicas, e que cada porta lógica possui um símbolo esquemático, uma equação booleana e uma Tabela Verdade associada.

Neste capítulo você aprendeu também uma metodologia de projeto que permite construir Circuitos Lógicos Combinacionais utilizando portas lógicas. Nesta metodologia a funcionalidade esperada do circuito é primeiramente expressa na forma de uma Tabela Verdade a partir da qual pode-se deduzir a equação booleana geral do circuito. E que, uma vez tendo-se deduzido a equação booleana geral do circuito, pode-se facilmente substituir cada operador booleano pela porta lógica que o representa tendo por fim o circuito lógico combinacional que atende aos requisitos do projeto.



Aprenda Praticando

1. Implemente a Tabela Verdade, defina as Equações Booleanas Parciais e a Equação Booleana Geral e o Circuito Elétrico de um circuito de controle para um sistema de alarme com a seguinte funcionalidade:

“O alarme soará sempre que for recebido um sinal de falha juntamente com um sinal de parada ou um sinal de alerta”.

Resolução: Seguindo a metodologia proposta na seção 3.3 do

nosso Volume 1, primeiramente devemos analisar o enunciado do problema a fim de identificar possíveis redundâncias que, se removidas, facilitem a resolução do nosso problema. Por ser um problema extremamente simples, não foi encontrada nenhuma redundância que possa ser removida. Nosso segundo passo será definir e implementar a tabela verdade que representa a funcionalidade esperada do nosso alarme. Para tanto precisamos identificar quais serão os sinais de entrada e qual será o sinal de saída do nosso circuito. Da definição do problema temos:

Como sinais de entrada:

- » SF: Sinal de Falha
- » SP: Sinal de Parada
- » SA: Sinal de Alerta

Como sinal de saída:

- » AL: Sinal de Alarme

Definidos nossos sinais de entrada e saída podemos construir nossa tabela verdade. Para tanto precisamos primeiro definir nosso padrão de sinalização, vamos considerar que todos os sinais, tanto de entrada quanto de saída, serão válidos ou ativos em 1. Precisamos definir também o tamanho da nossa tabela verdade, ou seja, quantas linhas ela terá. Uma vez que a nossa tabela verdade deve atender a todas as possibilidades de configuração dos sinais de entrada e, como temos 3 sinais de entrada, devemos construir uma tabela verdade com 8 linhas. Na Figura 1 a seguir temos como ficará a nossa tabela verdade.

SF	SP	SA	AL
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

AL = 1 Porque o Sinal de Alerta está válido (SA = 1)

AL = 1 Porque o Sinal de Falha está válido juntamente com o sinal de parada (SF = 1 e SP = 1)

AL = 1 Porque todas as condições de acionamento do Alarme são válidas

Figura 1: Definição da Tabela Verdade

Construída a nossa tabela verdade, podemos passar a definir nossas equações booleanas parciais e, a partir destas, a nossa equação booleana geral do circuito. A Figura 2 a seguir nos trás as equações booleanas parciais do circuito. Lembre-se as equações parciais correspondem sempre às condições de entrada para as quais a saída é verdadeira, ou seja, para as quais a saída é 1.

SF	SP	SA	AL	
0	0	0	0	
0	0	1	1	$\overline{SF} \cdot \overline{SP} \cdot SA$
0	1	0	0	
0	1	1	1	$\overline{SF} \cdot SP \cdot SA$
1	0	0	0	
1	0	1	1	$SF \cdot \overline{SP} \cdot SA$
1	1	0	1	$SF \cdot SP \cdot \overline{SA}$
1	1	1	1	$SF \cdot SP \cdot SA$

Figura 2: Dedução das Equações Booleanas Parciais

A partir das equações booleanas parciais podemos facilmente chegar a uma equação booleana geral do circuito, para tanto basta conectar as equações parciais encontradas entre si através do operador lógico OR, representado pelo símbolo +. A equação geral do circuito ficará da seguinte forma:

$$\overline{SF} \cdot \overline{SP} \cdot SA + \overline{SF} \cdot SP \cdot SA + SF \cdot \overline{SP} \cdot SA + \overline{SF} \cdot SP \cdot \overline{SA} + SF \cdot SP \cdot SA$$

Para concluir vamos agora implementar um circuito lógico que atenda a esta equação geral encontrada. A Figura 3 a seguir nos trás o circuito lógico desejado.

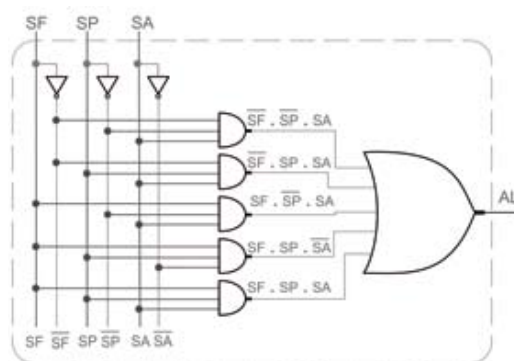


Figura 3: Circuito Lógico

2. Deduza as equações booleanas parciais e geral e Implemente um circuito lógico digital que atenda a tabela verdade da função booleana **XOR (OU Exclusivo)**.
3. Dada a tabela verdade abaixo, onde A, B e C são os sinais de entrada e F o sinal de saída, deduza as equações booleanas parciais e geral e desenhe o seu circuito lógico equivalente.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Obs: Para os exercícios apresentados a seguir, considere as entradas B2, B1 e B0 como sendo os bits da palavra binária que é aplicada à entrada do circuito lógico a ser implementado.

4. Implemente a Tabela Verdade, deduza as equações booleanas parciais e geral e desenhe o circuito lógico equivalente que atenda a seguinte função booleana:
 - » $f(B_2, B_1, B_0) = 1$, se o valor binário aplicado às entradas é par.
5. Implemente a Tabela Verdade, deduza as equações booleanas parciais e geral e desenhe o circuito lógico equivalente que atenda a seguinte função booleana:
 - » $f(B_2, B_1, B_0) = 1$, se o valor binário aplicado às entradas é primo.
6. Implemente a Tabela Verdade, deduza as equações booleanas parciais e geral e desenhe o circuito lógico equivalente que atenda a seguinte função booleana:
 - » $f(B_2, B_1, B_0) = 0$, se o valor binário aplicado às entradas é divisível por 3.

Conheça os Autores

Juliana Regueira Basto Diniz possui graduação em engenharia eletrônica pela Universidade Federal de Pernambuco, mestrado e doutorado em Ciência da Computação pela Universidade Federal de Pernambuco. Atualmente é professora da Universidade Federal Rural de Pernambuco (UFRPE), desenvolvendo trabalhos no grupo de Educação a Distância desta universidade. Seus temas de interesse em pesquisa são: Sistemas Distribuídos, Computação Ubíqua e Ensino a Distância.

Abner Corrêa Barros é mestre em Ciência da Computação com foco em Engenharia de Hardware pelo Centro de Informática da Universidade Federal de Pernambuco. Possui graduação em Ciência da Computação pela mesma universidade. Atualmente é professor da disciplina de Organização e Arquitetura de Computadores da Faculdade Maurício de Nassau e Engenheiro de Hardware da Fundação de Apoio ao Desenvolvimento da UFPE (FADE), atuando em um projeto de convênio entre o Centro de Informática da UFPE e a Petrobrás. Suas áreas de interesse e pesquisa são: Hardware Reconfigurável, Arquitetura de Cores Aritméticos e Computação de Alto Desempenho em *Field-Programmable Gate Array* (FPGA).