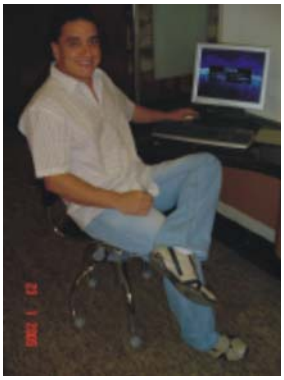
Claretiano
CENTRO UNIVERSITÁRIO

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

CURSOS DE GRADUAÇÃO – EAD

Arquitetura e Organização de Computadores – *Prof. Ms. Fernando Marco Perez Campos*



Meu nome é **Fernando Marco Perez Campos**.Sou mestre em Engenharia de Produção na linha de pesquisa de Sistemas de Informação pela Universidade Paulista (São Paulo – SP). Sou graduado em Ciência da Computação também pela Universidade Paulista. Atuo como professor nos cursos de Ciência da Computação, Engenharia e Sistemas de Informação e áreas afins, ministrando as seguintes disciplinas: Lógica de Programação; Arquitetura de Computadores; Organização de Computadores; Sistemas Digitais; Processamento de Dados; Programação de Computadores (estrutura de dados); Teoria da Linguagem de Programação; Ciência da Computação Integrada. Atuo também no mercado de trabalho na área de consultoria e assessoria em informática.

Fernando Marco Perez Campos

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

Batatais

Claretiano

2014

Versão: dez./2014

004.22 C21a

Campos, Fernando Marco Perez
Arquitetura e Organização de Computadores / Fernando Marco Perez Campos – Batatais, SP : Claretiano, 2014
219 p.

ISBN: 978-85-8377-132-6

1. Sistemas numéricos. 2. Circuitos digitais. 3. Processadores. 4. Memórias. 5. Dispositivos de entrada e saída. I. Arquitetura e Organização de Computadores.

CDD 004.22

Claretiano - Centro Universitário
Rua Dom Bosco, 466 - Bairro: Castelo – Batatais SP – CEP 14.300-000
cead@claretiano.edu.br
Fone: (16) 3660-1777 – Fax: (16) 3660-1780 – 0800 941 0006
www.claretianobt.com.br

*Fazemos parte do **Claretiano - Rede de Educação***

SUMÁRIO

Caderno de Referência de Conteúdo

Conteúdo ––

Sistemas numéricos e conversões de bases. Circuitos digitais. Processadores. Memórias. Dispositivos de entrada e saída. Componentes de fluxo de dados. Unidade Aritmética e Lógica. Unidade de Controle e Microprogramação. Arquitetura do SAP1. Conjunto de instruções e programação do SAP1. Execução de instruções no SAP1. SAP 2.

––

1. INTRODUÇÃO

Seja bem-vindo ao estudo de *Arquitetura e Organização de Computadores*. Nesta parte, chamada *Caderno de Referência de Conteúdo*, você encontrará o conteúdo das **nove unidades de estudo**.

Antes de iniciarmos este estudo, faz-se necessário responder alguns questionamentos, tais como:

Como funciona um computador? Qual é a configuração adequada para rodar determinados aplicativos? Como programar de forma mais eficiente? Quais são os componentes necessários para integrar os computadores aos outros dispositivos eletrônicos do nosso dia a dia?

Para responder essas perguntas, é necessário estudar a computação a partir dos seus conceitos fundamentais, conhecendo as partes funcionais dos circuitos digitais até chegar ao sistema como um todo.

A arquitetura dos computadores atuais é grande e complexa, porém trata-se da evolução de arquiteturas bem mais simples. Assim, a partir do momento em que essas arquiteturas simples são compreendidas, conseguimos entender também o funcionamento dos computadores atuais, pois o princípio de funcionamento é o mesmo.

A tarefa de programar um computador está diretamente relacionada à máquina, pois são as máquinas que executam os programas. Dessa forma, se você busca escrever programas eficientes, terá que entender como esses programas irão utilizar os recursos de *hardwares* disponíveis. Você sabia que os melhores programadores possuem um profundo conhecimento de *hardware*?

Outro aspecto importante relacionado aos nossos estudos é a computação ubíqua – que utiliza o conceito de computação pervasiva (presente em quase tudo o que usamos) – com a computação móvel, para tornar o computador onipresente.

A viabilidade da computação ubíqua está condicionada à possibilidade de integração de diferentes dispositivos. Para poder integrá-los, é necessário conhecê-los.

Você ainda estudará os sistemas numéricos e as conversões de base, compreenderá a implementação de sistemas computacionais no nível da lógica digital e suas implementações, conhecerá os princípios e as organizações das estruturas físicas de um computador e a relação existente entre o *hardware* e o *software*.

Além disso, terá a oportunidade de descobrir como um computador executa os programas; entender a evolução das tecnologias de *hardware* até os dias de hoje; aprender alguns conceitos relacionados aos projetos de computadores e entender como funciona a execução de um programa.

Além de todos os conceitos apresentados, é importante que você compreenda os conceitos relacionados ao *hardware* de um computador e ao funcionamento do sistema durante a execução de um programa.

Para ter um rendimento satisfatório, é importante que você realize todas as atividades e interaja com seus colegas de curso e seu tutor.

Seremos uma equipe virtual e, portanto, é imprescindível a sua colaboração. Assim, trocaremos ideias com a finalidade de agregar conhecimentos.

2. ORIENTAÇÕES PARA ESTUDO

Abordagem Geral

Neste tópico, apresenta-se uma visão geral do que será estudado. Aqui, você entrará em contato com os assuntos principais deste conteúdo de forma breve e geral e terá a oportunidade de aprofundar essas questões no estudo de cada unidade. Desse modo, essa Abordagem Geral visa fornecer-lhe o conhecimento básico necessário a partir do qual você possa construir um referencial teórico com base sólida – científica e cultural – para que, no futuro exercício de sua profissão, você a exerça com competência cognitiva, ética e responsabilidade social.

Arquitetura e Organização de Computadores está presente na maior parte das grades curriculares dos cursos de Computação e Informática, como, por exemplo, Ciência da Computação, Engenharia da Computação, Sistemas de Informação, Licenciatura em Computação, Análise e Desenvolvimento de Sistemas, Tecnologia da Informação, entre outros. A sua aplicação pode ser direta ou indireta, dependendo do curso em questão e também da área de atuação do profissional da Computação ou Informática.

A aplicação direta está relacionada às atividades que utilizam a computação como atividade-fim, ou seja, que utilizam a computação pela computação e não como uma ferramenta para solucionar problemas de outras áreas. São exemplos desses profissionais os projetistas de *hardware*, engenheiros e bacharéis em Ciência da Computação (quando desenvolvem códigos em linguagem de montagem ou *drivers* para novos dispositivos).

Já a aplicação indireta está ligada aos profissionais que utilizam os conceitos de arquitetura e organização de computadores para desenvolver melhor uma atividade que não está diretamente relacionada à disciplina. O conhecimento do funcionamento da máquina e seus princípios proporcionam, contudo, maior eficiência e eficácia ao produto ou ao processo em questão.

Como este *Caderno de Referência de Conteúdo* está focado nos cursos que utilizam a computação como uma ferramenta de desenvolvimento ou como parte de uma infraestrutura, trataremos sobre o tema com ênfase nessas aplicações.

O maior diferencial entre um profissional que possui uma formação prática e outro que possui uma formação acadêmica é justamente o domínio do seu conhecimento. Enquanto o profissional prático se limita ao conhecimento da atividade, o profissional acadêmico vai além e conhece tudo o que está ao redor da atividade, e aplica esse conhecimento para o seu benefício.

Vejamos alguns cenários em que o conhecimento destes conteúdos poderão ser de grande valor:

Um cliente que possui um estacionamento solicitou a construção de um *software* para reconhecer o número da placa de seus clientes que entram e saem do seu pátio. A partir daí, o sistema deve controlar o tempo de estacionamento utilizado e gerar uma fatura. O custo computacional (esforço do processador) para identificar os números que constam na placa de cada veículo é muito grande e, por conta disso, é necessário refletir sobre um meio de aperfeiçoamento do processador para que o programa fique mais leve. Os profissionais de desenvolvimento, em especial a equipe de análise de requisitos, poderão identificar a necessidade de dimensionar um processador com mais núcleos ou mesmo uma máquina com mais processadores para poder atender os requisitos desse *software* e, além disso, os profissionais que codificarão o programa deverão entender como programar especificamente para esses processadores. Portanto, podemos observar que os conhecimentos de como os processadores funcionam é fundamental para este cenário.

Outro exemplo é a necessidade de um programa de funcionamento para os produtos elétrico-eletrônicos que são microcontrolados. A programação desses dispositivos obedece à mesma lógica da programação dos computadores com que estamos acostumados a trabalhar; em alguns casos, inclusive, é utilizada a mesma linguagem de programação, como Basic ou C. Para programar tais dispositivos, mesmo utilizando essas linguagens, é necessário conhecer o *hardware* que rodará o programa. Assim, o programador saberá quais são os recursos de *hardware* disponíveis para utilização e, também, como habilitá-los para receber e enviar sinais. Desse modo, todas as unidades de estudo serão importantes para o conhecimento desse tipo de *hardware* e da sua correta programação.

Em outro cenário, imagine que você esteja desenvolvendo uma aplicação que precisa trocar mensagens (interagir) com dispositivos de *hardware* externo. Como exemplo, consideremos que esse dispositivo seja uma fonte de tensão e corrente e, adicionalmente, que ele possua uma interface serial. Você deverá obter informações sobre como o equipamento troca as suas mensagens, configurações de portas e interrupções para só então poder desenvolver um programa para trabalhar adequadamente com tal dispositivo.

Outra situação é pensar que algumas rotinas exigem o controle direto do *hardware*, muitas vezes passando até por cima do sistema operacional. Essas aplicações, na maior parte das vezes, necessitam do acesso ao valor de alguns registradores do processador. Uma possibilidade para resolver esse problema é programar em *assembly* (linguagem de montagem – um nível acima da linguagem de máquina). Para essa programação, é necessário um conhecimento profundo dos sistemas numéricos binários e hexadecimais; portanto, é desejável que você tenha conhecimento de sistemas digitais e também dos processadores, memórias e dispositivos de entrada e saída.

Como um conhecimento maior da arquitetura e da organização dos computadores pode proporcionar um melhor trabalho? Disponibilizando os dados e as informações necessárias para entender como a máquina funciona e também propiciando o conhecimento adequado para trabalhar de forma mais eficiente e eficaz com essas máquinas.

Os conhecimentos adquiridos neste *Caderno de Referência de Conteúdo* também são necessários para o desenvolvimento de programas com um custo computacional menor. Você já pensou nas diferenças existentes entre construir uma estrutura condicional ***if else*** ou uma estrutura equivalente utilizando apenas dois ***ifs***? Onde o processador gasta o seu tempo? Nas comparações ou nas atribuições? Quanto tempo um processador gasta para buscar e executar as instruções? Esse tempo varia de acordo com a instrução que está sendo executada ou é igual para todas elas? A *Arquitetura e Organização de Computadores* vai ajudá-lo a refletir e responder perguntas como essas.

Na primeira unidade, você terá a oportunidade de aprender sobre os sistemas numéricos, suas bases, as conversões numéricas entre números de bases diferentes, além das operações aritméticas entre eles.

A linguagem natural dos computadores é de base dois (binária). O computador é capaz de entender apenas zeros e uns, ou melhor, podemos afirmar que ele é capaz de entender a presença ou a ausência de corrente elétrica em um determinado ponto do circuito, em um determinado momento. Portanto, para facilitar o entendimento das máquinas, essas condições são traduzidas em zeros e uns.

No sistema numérico decimal, temos dez algarismos para representar qualquer quantidade. Os algarismos do sistema numérico decimal são: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Assim, os números são formados multiplicando-se cada algarismo com a base (que no caso é dez) elevada à posição desse algarismo na formação do número.

Por exemplo:

$(87632)_{10} = 8 \text{ X } 10^4 + 7 \text{ X } 10^3 + 6 \text{ X } 10^2 + 3 \text{ X } 10^1 + 2 \text{ X } 10^0$

Ou, ainda:

$8 \text{ X } 10.000 + 7 \text{ X } 1000 + 6 \text{ X } 100 + 3 \text{ X } 10 + 2 \text{ X } 1 = (87632)_{10}$

Observe que, mesmo trabalhando com números de base dez, quando escrevemos vários números de bases diferentes em um mesmo documento, é muito importante indicar a base. A indicação da base deve ser feita colocando o número entre parênteses e a base como índice do lado de fora do parêntese.

Quando trabalhamos com números no sistema hexadecimal, além da indicação citada, é muito comum encontrarmos o número fora dos parênteses seguido da letra maiúscula **H**. Por exemplo, **46H**.

Da mesma forma, podemos escrever os números de outras bases como, por exemplo, números binários (de base dois), octais (de base oito) e hexadecimais (de base dezesseis).

O sistema binário utiliza apenas os algarismos 0 e 1. Assim:

$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (11)_{10}$

$(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (9)_{10}$

$(101011)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (43)_{10}$

$(11011)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (27)_{10}$

O sistema octal utiliza os algarismos 0, 1, 2, 3, 4, 5, 6 e 7. Um erro muito comum é considerar o número oito como sendo o último número do sistema octal. Não pode ser assim, pois o zero conta como um algarismo e por isso o oito não pode entrar como um símbolo válido para os números do sistema octal. Veja:

$(167)_8 = 1 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 = (119)_{10}$

O último exemplo será de um número em hexadecimal. Os algarismos do sistema numérico hexadecimal são: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F. Os símbolos A, B, C, D, E e F valem, respectivamente: 10, 11, 12, 13, 14 e 15.

$(210)_{16} = 2 \times 16^2 + 1 \times 16^1 + 0 \times 16^0 = (528)_{10}$

$(5AF)_{16} = 5 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = (1455)_{10}$

Não será possível demonstrarmos todas as possibilidades, mas todos os sistemas numéricos obedecem sempre à mesma regra.

Utilizando as regras adotadas nos exemplos anteriores, você conseguirá converter um número de qualquer base para a base dez.

E o contrário? É possível?

Claro que sim. É possível converter um número de base dez para um número equivalente em qualquer base. Basta fazer o contrário da formulação adotada na conversão de um número de qualquer base para a base dez. Deve-se dividir o número de base dez pela base para a qual se deseja converter o número. As divisões devem ser sucessivas até que o resultado seja menor do que a base de destino. Ao terminar as divisões, o resultado deverá ser montado pegando-se o último resultado com os restos das divisões anteriores. Observe os exemplos.

Conversões da base dez para binário

$(43)_{10} = (\quad)_2$

Resolução:

43/2 = 21 com resto 1.

21/2 = 10 com resto 1.

10/2 = 5 com resto 0.

5/2 = 2 com resto 1.

2/2 = 1 com resto 0.

Deve-se montar o número partindo-se do último resultado (1) com os últimos restos no sentido da última operação para a primeira. Assim, temos o seguinte resultado: $(101011)_2$.

$(43)_{10} = (101011)_2$

Conversão da base dez para a base oito

$(73)_{10} = (\quad)_8$.

73/8 = 9 com resto 1.

9/8 = 1 com resto 1

Deve-se montar o número partindo-se do último resultado (1) com os últimos restos no sentido da última operação para a primeira. Assim, temos o seguinte resultado: $(111)_8$.

$(73)_{10} = (111)_8$

Conversão da base dez para a base dezesseis (hexadecimal)

$(73)_{10} = (\quad)_{16}$.

73/16 = 4 com resto 9.

Observe que foi possível executar apenas uma divisão. Neste caso, deve-se pegar o resultado com o resto (nessa ordem) para montar o resultado final.

$(73)_{10} = (49)_{16}$

Depois de entender os sistemas de numeração e as suas conversões, você terá a possibilidade de aprender a realizar operações aritméticas (soma e subtração) em sistemas numéricos de bases diferentes de dez.

Na unidade 2, você terá contato com os sistemas digitais e alguns exemplos de circuitos digitais. Para entender o funcionamento dos computadores digitais, é necessário entender como os dispositivos digitais funcionam e como eles se relacionam.

Todo circuito digital pode ser representado por um diagrama composto por símbolos que representam portas lógicas. As portas lógicas implementam funções booleanas que podem ser entendidas como partes de lógica baseada em sinais **ligado** ou **desligado**.

Outra forma de representação de um circuito digital é a utilização de expressões booleanas (que operam sobre variáveis binárias e a lógica clássica).

Os circuitos devem receber sinais por meio das variáveis de entrada. Esses sinais serão processados pelo circuito, que devolverá ao ambiente externo o resultado do processamento nas variáveis de saída.

Para exemplificar essa operação, imagine um problema prático: suponha que você deseja construir um circuito capaz de somar dois *bits*. Para isso, será utilizado um circuito que receberá os dois *bits* nas variáveis A e B e o resultado será disponibilizado nas variáveis S (soma) e V (vai um). Para esse exemplo, as variáveis A e B serão variáveis de entrada, enquanto as variáveis S e V serão de saída e receberão, respectivamente, o resultado da soma e o resultado do estouro (vai um).

Para construir o circuito somador, deve-se, em primeiro lugar, montar uma tabela que armazene todas as possibilidades de valores para as variáveis de entrada e os resultados das variáveis de saída para cada caso das variáveis de entrada. Essa tabela recebe o nome de **tabela verdade** e deverá ter o número de linhas igual a 2^n, em que n é igual ao número de variáveis de entrada. Para entender o motivo dessa conta, basta lembrar que, em um sistema binário, as variáveis de entrada receberão apenas dois valores: zero ou um. Assim, fazendo uma análise combinatória envolvendo todas as possibilidades para n variáveis de entrada, acabamos por obter o valor 2^n.

Veja o exemplo da tabela verdade do somador desse exemplo:

Tabela 1 Tabela verdade de um circuito somador de dois *bits*.

A	B	S	V
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Todos os valores a serem somados estão em binário; portanto, não se preocupe em entender os resultados. Durante o estudo da Unidade 1, você terá a possibilidade de aprender que, em binário, 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1 e 1 + 1 = 0 e vai um.

A partir da tabela verdade, utiliza-se uma técnica para escrever a(s) expressão(ões) booleana(s) da(s) variável(is) de saída(s).

Uma das formas de extrair a expressão booleana a partir da tabela verdade é multiplicar (operação **e** booleana) as variáveis de entrada cuja(s) saída(s) for(em) igual(is) a um, invertendo a variável de entrada (complementando) quando as mesmas forem zero. Usaremos o sinal de apóstrofe como complemento.

Neste caso, teremos:

S = A’ . B + A . B’

V = A . B.

A técnica de extração de expressão booleana apresentada é denominada **soma de produtos**, pois devem ser multiplicadas as variáveis de entrada (operação **e** ou ***and***), e, então, somadas as linhas com saídas iguais a um (operação **ou** ou ***or***). Na unidade 2, você também terá a oportunidade de aprender a técnica de produto das somas, que é exatamente o contrário da técnica soma de produtos. Nessa técnica, devem-se considerar as saídas iguais a zero, somar as variáveis de entrada (complementando as variáveis com valor igual a um) e multiplicar as linhas cujas saídas sejam iguais a zero.

As duas técnicas apresentadas são de simples compreensão, porém não produzem expressões simplificadas, ou seja, que utilizam o menor número de portas lógicas possível. Para obter expressões simplificadas, devemos aplicar algumas propriedades, teoremas ou postulados da álgebra booleana da mesma forma que fazemos com as expressões matemáticas, ou utilizar alguma ferramenta para simplificação.

Como nosso curso não tem o foco em sistemas digitais, as simplificações algébricas booleanas não serão abordadas, sendo apresentada somente a técnica de simplificação utilizando mapas de Veitch-Karnaugh. Esses mapas constituem uma ferramenta bem simples para que, a partir da tabela verdade, seja escrita a expressão booleana correspondente e, assim, os circuitos sejam implementados utilizando-se o menor número possível de portas lógicas.

Veja, na Tabela 2, um exemplo do uso da ferramenta Mapas de, aplicado a uma tabela-verdade com três variáveis de entrada.

Tabela 2 Utilização da ferramenta Mapas de Veitch-Karnaugh em uma tabela verdade.

Extração da expressão pelo método da Tabela Verdade.

linha	A	B	C	S
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	0
8	1	1	1	0

Extração da expressão utilizando a ferramenta Mapa de Veitch-Karnaugh.

		B C			
		0	1	1	0

Expressão extraída da tabela verdade	Expressão extraída do Mapa de Veitch-Karnaugh
$S = A' . B' . C' + A' . B' . C + A' . B . C' + A' . B . C.$	$S= A'$

Comentários:

Pode-se escolher pegar as saídas iguais a um ou as saídas iguais a zero. O ideal é escolher a saída com menor número de ocorrências. Se a escolha for considerar as saídas iguais a um, a operação consiste em multiplicar as variáveis de entrada da linha cuja saída seja igual a um e somar às outras multiplicações provenientes das outras linhas cujas saídas sejam iguais a um. Já se a escolha for considerar as saídas iguais a zero, a operação consiste em somar as variáveis de entrada da linha cuja saída seja igual a zero (colocar entre parênteses) e multiplicar com as outras somas provenientes das outras linhas cujas saídas sejam iguais a zero.

Para este caso, temos os seguintes termos (provenientes de cada linha com saída igual a um):

Primeira linha: $A' . B' . C'$
Segunda linha: $A' . B' . C$
Terceira linha: $A' . B . C'$
Quarta linha: $A' . B . C.$

Efetuando a soma das multiplicações, temos:

$S = A' . B' . C' + A' . B' . C + A' . B . C' + A' . B . C.$

A operação consiste em analisar o enlace de uns ou de zeros, verificando, em todo o espaço do enlace, as variáveis de entrada que mudam ou não de valor. As variáveis que mudam de valor não entram na expressão e as variáveis que não mudam entram da seguinte forma: para enlaces de uns, quando o valor da variável de entrada não mudar e for igual a um, ela entrará sem complemento; e, se o valor da variável de entrada não mudar de valor e for igual a zero, ela deverá entrar complementada. Para enlaces de zeros, quando o valor da variável de entrada não mudar e for igual a um, ela entrará complementada; e, se o valor da variável de entrada não mudar e for igual a zero, ela entrará sem o complemento (portanto, de forma oposta aos enlaces de uns). Quando os enlaces forem de uns, as variáveis de entrada que sobrarem deverão ser multiplicadas e os enlaces (quando existir mais de um) deverão ser somados. Para enlaces de zeros, as variáveis de entrada que sobrarem deverão ser somadas (inserir parênteses) e os enlaces (quando existir mais de um) deverão ser multiplicados. Para este caso, temos apenas um enlace ocupando toda a primeira linha. Assim, as variáveis B e C mudam de valor ao longo de todo o enlace da primeira linha, enquanto a variável A vale zero para todo o enlace. A expressão fica: $S = A'$. É importante notar a evidente simplificação que o mapa de Veitch-Karnaugh proporcionou. As duas expressões são equivalentes, ou seja, fazem a mesma coisa, só que a expressão proveniente do mapa possui apenas uma porta **não** (***not***) – para complementar o valor de A, enquanto a expressão extraída da tabela-verdade utiliza varias portas **e** (***and***), **ou** (***or***) e **não** (***not***).

Os circuitos podem ser construídos utilizando-se os símbolos das portas lógicas e as expressões booleanas obtidas da tabela verdade.

Depois de conhecer os circuitos digitais, desde a sua concepção até a sua implementação, você poderá conhecer os principais circuitos utilizados em conjunto com os sistemas microcontrolados e também nos computadores atuais.

Além dos circuitos digitais estarem presentes em sistemas de controle de temperatura, sistemas de alarme, controle de motores e bombas etc., eles também estão presentes nos computadores, dentro dos processadores e em muitos outros circuitos de controle.

Veja, a seguir, alguns desses circuitos e como eles funcionam.

1) **Multiplexadores**: são dispositivos que recebem sinais provenientes de vários canais de dados e transmitem esses sinais utilizando apenas um canal. Para isso, esse dispositivo compartilha o canal por meio de uma chave de seleção que determinará qual entrada estará conectada à sua única saída em um determinado instante.
2) **Demultiplexadores**: fazem o inverso dos multiplexadores. Eles recebem como entrada apenas um canal de dados e selecionam, por meio de um conjunto de chaves, qual das saídas disponíveis estará conectada a essa entrada em um determinado instante.
3) **Codificadores e decodificadores**: são circuitos responsáveis por transformar códigos de um padrão para outro. Muitas vezes, isso é necessário para poder tratar as informações corretamente em um sistema. Por exemplo, podemos codificar o sistema numérico decimal em um código binário de quatro *bits* para poder processar esses dados em binário. Para devolver o resultado ao usuário, pode-se utilizar um decodificador para transformar o resultado em um código decimal.
4) **Somadores e Subtradores**: são circuitos capazes de efetuar a soma ou subtração de *bits*. Eles operam sobre o sistema binário e podem ser associados para efetuarem operações de vários *bits* simultaneamente. Dentro dos processadores, existem muitos dispositivos como esses, compondo a unidade lógica e aritmética do *chip*.
5) ***Flip-flops***: são circuitos sequenciais, ou seja, a sua saída não depende apenas das suas entradas e sim das saídas atuais (existe uma retroalimentação dos sinais digitais). Esses dispositivos podem ser utilizados como elementos de memória, registradores, contadores etc.

Com esses conceitos, você estará apto a continuar os estudos nas próximas unidades, entendendo não apenas as partes que compõem os computadores e sim como essas partes funcionam.

Mais adiante, na Unidade 3, você terá a oportunidade de estudar os processadores.

O processador, também conhecido como CPU (Unidade Central de Processamento) é tão importante que muitas vezes fazemos referência à toda caixa do computador (gabinete) como sendo a CPU.

É nesse dispositivo que todas as instruções são executadas e que os programas conseguem converter dados de entrada em dados de saída. Podemos dizer que se trata do **cérebro** do computador.

Todo computador tem um processador. Alguns possuem até mais de um (máquinas multiprocessadas). Os processadores atuais também suportam o multiprocessamento, pois possuem vários núcleos dentro de um mesmo *chip*.

Os processadores executam os programas por meio de suas instruções. As instruções determinam os passos para que o processador busque o que tem que ser feito na memória em que o programa está armazenado, procure os dados necessários para efetuar o processamento daquela instrução, execute a instrução e guarde o resultado em uma memória de saída chamada de registrador.

Podemos dividir os processadores em três partes:

1) **Unidade de Controle**: é a parte que controla o fluxo de *bits* dentro do processador.
2) **ULA (Unidade Lógica e Aritmética)**: é o local onde os dados são processados.
3) **Registradores**: é uma "memória" que armazena apenas uma palavra de dados. Uma palavra de dados é um conjunto de *bits* que tem um significado na arquitetura do processador.

Desse modo, ao final da Unidade 3, você saberá quais são as arquiteturas de processadores existentes e também como esses processadores trabalham.

Na Unidade 4, trataremos das memórias. "Quanto você tem de memória no seu computador?". Você já deve ter ouvido muito essa pergunta, não é mesmo? Mas a resposta é sempre relativa e depende de a qual memória a pergunta faz referência.

Os computadores atuais possuem vários tipos de memórias. As memórias maiores e mais lentas normalmente são utilizadas para cópias de segurança ou armazenamento de dados que não são utilizados com frequência. Essas memórias são chamadas de memórias secundárias ou memória de massa. São exemplos dessas memórias as fitas DAT (apropriadas para cópias de segurança), CDs, DVDs e HDs (*Hard Disk* ou disco rígido). Os HDs são bem mais rápidos do que as fitas e são utilizados para armazenar o sistema operacional, os programas e os dados que são utilizados com mais frequência.

Quando um programa é executado, ele é dividido em processos e esses processos devem ser carregados em memórias mais rápidas. Essas memórias constituem a **memória principal** do computador e, embora menores por conta do custo, são muito mais eficientes e, por isso, ficam próximas ao processador. São exemplos dessas memórias: memória RAM, memória CACHE e registradores. Essas três memórias são voláteis (os dados armazenados são apagados quando a energia do computador for cortada) e a diferença entre elas é a velocidade.

As memórias RAM são mais lentas do que as memórias CACHE e as memórias CACHE são mais lentas do que os registradores (que já estão dentro do processador).

Ainda na Unidade 4, você poderá aprender sobre as tecnologias das memórias apresentadas e as suas relações. Com essas informações, você será capaz de dimensionar adequadamente os tipos e os tamanhos das memórias a serem utilizadas em um sistema, de acordo com a sua aplicação.

Na Unidade 5, você terá a possibilidade de entender os diferentes tipos de dispositivos que se conectam com os computadores e as particularidades de cada um deles. O funcionamento de uma impressora, um monitor, um *modem*, um *mouse* e muitos outros dispositivos serão explorados nessa unidade. Além de estudar esses dispositivos, você entenderá também como ocorre a comunicação entre eles.

Basicamente, as comunicações podem ser seriais ou paralelas. Cada uma dessas formas de comunicação oferece uma vantagem e também uma desvantagem.

Em uma comunicação serial, em que um *bit* é enviado atrás do outro, é intuitivo afirmar que a comunicação é menos eficiente do que em uma comunicação paralela, na qual os *bits* são enviados em paralelo (vários *bits* por vez). Na prática, essa afirmativa é falsa, pois, quando consideramos as possíveis velocidades de transmissão, deve-se considerar também o limite máximo de frequência por conta das possíveis interferências. Assim, sem essas informações, podemos cometer alguns erros baseados em ideias falsas sobre um universo extremamente nebuloso e complexo.

Essa unidade vai trabalhar essas questões com o objetivo de proporcionar a você, profissional da área de informática, a formação adequada para a escolha de dispositivos e suas interfaces de comunicação.

Nas Unidades 6, 7 e 8, estudaremos um modelo de computador muito simples, denominado SAP1. Esse computador tem finalidade apenas didática, porém utiliza a mesma base e lógica dos computadores atuais.

Na Unidade 6, será apresentada a parte física do SAP1. O SAP1 possui os seguintes módulos: contador de programa, módulo de entrada e registrador de endereço de memória, registrador de instrução, controlador-sequencializador, acumulador, somador/subtrador, registrador B, registrador de saída e *leds*. Cada um desses módulos será explicado detalhadamente no decorrer da unidade.

A Unidade 7 irá abordar o conjunto de instruções do SAP1, seus códigos e aplicações; e a Unidade 8 demonstrará como essas instruções serão executadas.

Para finalizar o estudo deste *Caderno de Referência de Conteúdo*, na Unidade 9, você terá a oportunidade de conhecer o SAP2, uma evolução do SAP1 que caminha para os conceitos utilizados nos computadores atuais.

O SAP2 possui muito mais instruções quando comparado ao SAP1, além de possuir mais registradores e já trabalhar com 16 *bits* por palavra de dados. A sua compreensão vai propiciar o entendimento do funcionamento dos computadores atuais que, embora sejam muito mais complexos, mantém os mesmos conceitos.

Ao final do *Caderno de Referência de Conteúdo* de *Arquitetura e Organização de Computadores*, você será capaz de entender como os programas são executados nos computadores e, por isso, poderá utilizar melhor as máquinas e desenvolver programas mais eficazes e eficientes.

Vale lembrar a importância do embasamento teórico que este conteúdo proporcionará para uma adequada gestão da tecnologia da informação no quesito infraestrutura.

Bons Estudos!

Glossário de Conceitos

O Glossário de Conceitos permite a você uma consulta rápida e precisa das definições conceituais, possibilitando-lhe um bom domínio dos termos técnico-científicos utilizados na área de conhecimento dos temas tratados em *Arquitetura e Organização de Computadores*. Veja, a seguir, a definição dos principais conceitos:

1) **Assembler**: compilador *assembly*.
2) **Assembly**: linguagem de montagem. Um programa escrito em *assembly* utiliza mnemônicos e números em hexadecimal.
3) **Barramento**: conjunto de condutores paralelos capazes de transmitir dados de um módulo para outro.
4) **Capacitores**: componente eletrônico que se caracteriza por acumular carga elétrica.
5) **Circuito combinacional**: circuito eletrônico digital que considera apenas as entradas atuais do circuito para gerar sinais de saída.
6) **Circuito sequencial**: circuito eletrônico digital que considera a saída atual do circuito em conjunto com as novas entradas para gerar os sinais de saída.
7) **Codificadores**: circuitos digitais capazes de transformar um tipo de padrão digital em outro tipo, como, por exemplo, um circuito para transformar um código binário em um código para conectar um *display* de sete segmentos.

8) **Compilador**: conjunto de aplicativos capazes de traduzir um programa escrito em uma linguagem qualquer para a linguagem de máquina.
9) **Complemento**: o que falta para um todo. Em sistemas de numeração, complemento é o que falta para a base.
10) **Concatenar**: estabelecer conexão entre; encadear, ligar.
11) **CPU**: unidade central de processamento ou, simplesmente, processador.
12) **Demultiplexador**: circuito digital capaz de selecionar uma de suas várias saídas para conectar à sua única entrada.
13) **Diodo**: componente eletrônico que permite a passagem de elétrons em uma única direção.
14) **Display de sete segmentos**: visor com sete *leds* em formato de linha, capaz de indicar os algarismos decimais de zero a nove.
15) **EEPROM**: memória que pode ser gravada e apagada várias vezes por meio de sinais eletrônicos.
16) **EPROM**: memória que pode ser gravada e apagada por luz ultravioleta.
17) **Expressões booleanas**: expressões que utilizam variáveis lógicas (variáveis binárias) para representar um circuito eletrônico.
18) **Flash**: tipo de memória EEPROM muito utilizado em *pen drives* e cartões de memória.
19) **Led**: diodo emissor de luz.
20) **Linguagem de montagem**: linguagem escrita utilizando zeros e uns. Utiliza OPCODES e números binários.
21) **Mapa de Veitch-Karnaugh**: ferramenta utilizada para simplificar expressões booleanas.
22) **Microcontrolador**: componente eletrônico que possui uma arquitetura semelhante à dos processadores, com a diferença de prover, adicionalmente, portas de entrada e saída, conversores AD/DA, entrada e saída serial e muitos outros recursos.
23) **Microprograma**: é o conjunto das macroinstruções de um computador, acompanhado das suas respectivas microinstruções.
24) **Mnemônico**: apelido dado aos OPCODES e são utilizados durante a programação em *assembly*.
25) **Multicomputadores**: arquitetura que possui mais de um processador com memória dedicada.
26) **Multiplexador**: circuito digital capaz de selecionar uma de suas várias entradas para conectar a sua única saída.
27) **Multiprocessadores**: arquitetura que possui mais de um processador compartilhando uma única memória.
28) **OPCODE**: código de operação de uma instrução, definido pelo fabricante do *hardware*.
29) **Palavra de dado**: conjunto de *bits* que representa uma informação, podendo o número de *bits* variar de uma arquitetura para outra.
30) **Pipeline**: método utilizado para processar mais de uma instrução por vez. Utiliza o conceito de estágios de processamento, ou seja, várias instruções podem ser executadas ao mesmo tempo, porém cada uma em um estágio diferente.
31) **PROM**: memória que pode ser gravada apenas uma vez.
32) **RAM**: memória volátil de acesso aleatório.
33) **Registrador**: memória de uma única palavra de dado.
34) **ROM**: memória somente leitura.
35) **Sistema binário**: tipo de sistema de numeração que utiliza dois símbolos para representar qualquer quantidade; os símbolos são: 0 e 1.

36) **Sistema decimal**: tipo de sistema de numeração que utiliza nove símbolos para representar qualquer quantidade; os símbolos são: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.
37) **Sistema hexadecimal**: tipo de sistema de numeração que utiliza dezesseis símbolos para representar qualquer quantidade; os símbolos são: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, e F.
38) **Sistema Octal**: tipo de sistema de numeração que utiliza oito símbolos para representar qualquer quantidade; os símbolos são: 0, 1, 2, 3, 4, 5, 6, e 7.
39) **Sistema Operacional**: conjunto de programas que tem a função de prover uma interface entre o homem e a máquina, gerenciando os arquivos e os programas conforme a necessidade de cada tarefa.
40) **Tabela Verdade**: utilizada em lógica matemática e/ou em sistemas digitais, é uma tabela que contempla todas as possibilidades de entradas de uma expressão ou circuito, juntamente com as suas respectivas saídas.
41) **Transistor**: componente eletrônico que permite ou não a passagem de corrente de um ponto para outro, dependendo da sua polarização.
42) **Volatilidade**: indica se os dados gravados em uma memória serão perdidos com a queda de energia (memórias voláteis) ou não (memórias não voláteis).

Esquema dos Conceitos-chave

Para que você tenha uma visão geral dos conceitos mais importantes deste estudo, apresentamos, a seguir (Figura 1), um Esquema dos Conceitos-chave. O mais aconselhável é que você mesmo faça o seu esquema de conceitos-chave ou até mesmo o seu mapa mental. Esse exercício é uma forma de você construir o seu conhecimento, ressignificando as informações a partir de suas próprias percepções.

É importante ressaltar que o propósito desse Esquema dos Conceitos-chave é representar, de maneira gráfica, as relações entre os conceitos por meio de palavras-chave, partindo dos mais complexos para os mais simples. Esse recurso pode auxiliar você na ordenação e na sequenciação hierarquizada dos conteúdos de ensino.

Com base na teoria de aprendizagem significativa, entende-se que, por meio da organização das ideias e dos princípios em esquemas e mapas mentais, o indivíduo pode construir o seu conhecimento de maneira mais produtiva e obter, assim, ganhos pedagógicos significativos no seu processo de ensino e aprendizagem.

Aplicado a diversas áreas do ensino e da aprendizagem escolar (tais como planejamentos de currículo, sistemas e pesquisas em Educação), o Esquema dos Conceitos-chave baseia-se, ainda, na ideia fundamental da Psicologia Cognitiva de Ausubel, que estabelece que a aprendizagem ocorre pela assimilação de novos conceitos e de proposições na estrutura cognitiva do aluno. Assim, novas ideias e informações são aprendidas, uma vez que existem pontos de ancoragem.

Tem-se de destacar que "aprendizagem" não significa, apenas, realizar acréscimos na estrutura cognitiva do aluno; é preciso, sobretudo, estabelecer modificações para que ela se configure como uma aprendizagem significativa. Para isso, é importante considerar as entradas de conhecimento e organizar bem os materiais de aprendizagem. Além disso, as novas ideias e os novos conceitos devem ser potencialmente significativos para o aluno, uma vez que, ao fixar esses conceitos nas suas já existentes estruturas cognitivas, outros serão também relembrados.

Nessa perspectiva, partindo-se do pressuposto de que é você o principal agente da construção do próprio conhecimento, por meio de sua predisposição afetiva e de suas motivações internas e externas, o Esquema dos Conceitos-chave tem por objetivo tornar significativa a sua aprendizagem, transformando o seu conhecimento sistematizado em conteúdo curricular, ou seja, estabelecendo uma relação entre aquilo que você acabou de conhecer com o que já fazia parte do seu conhecimento de mundo (adaptado do *site* disponível em: <http://penta2.ufrgs.br/edutools/mapasconceituais/utilizamapasconceituais.html>. Acesso em: 11 mar. 2010).

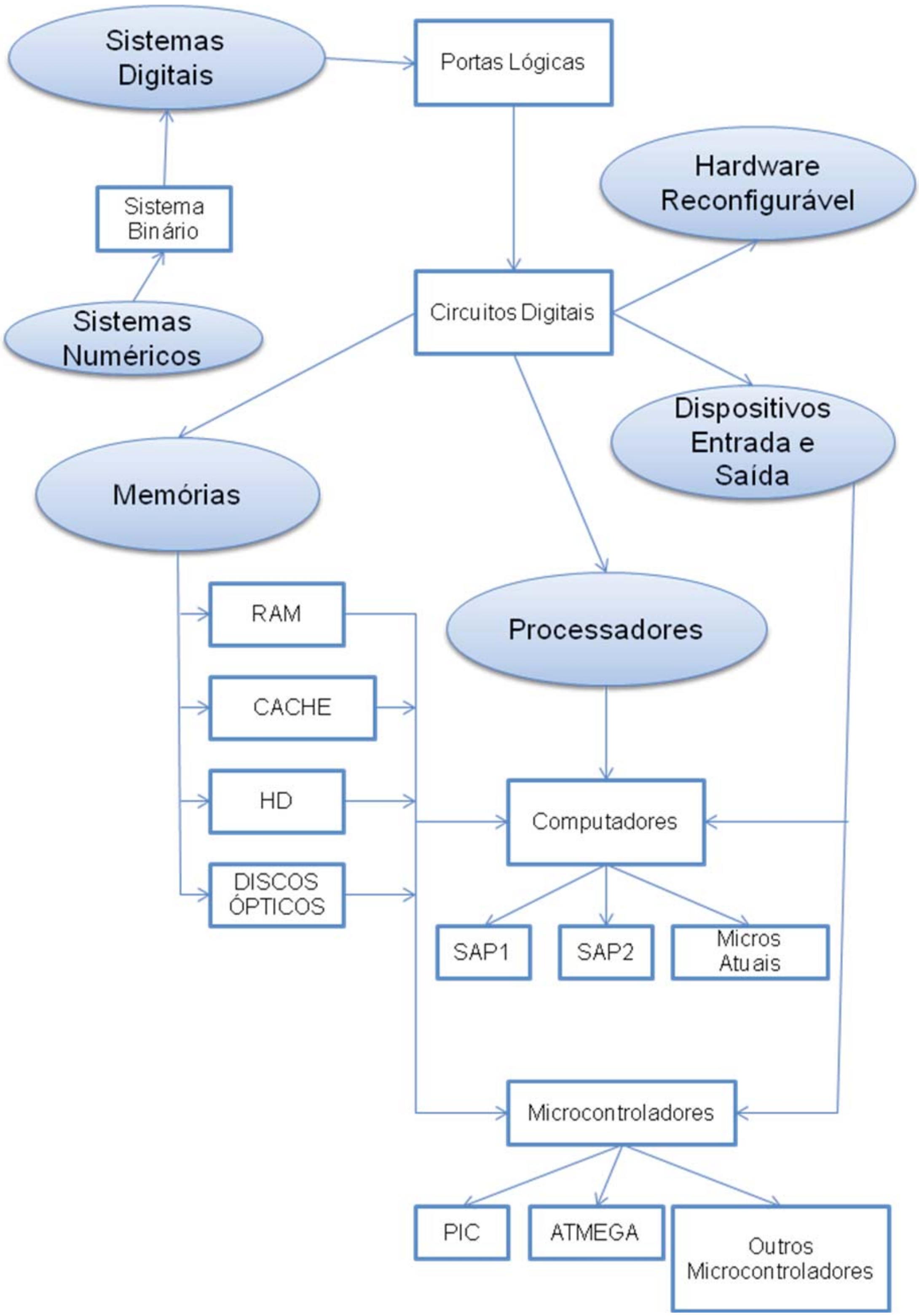


Figura 1 *Esquema dos Conceitos-chave de Arquitetura e Organização de Computadores*.

Como pode observar, esse Esquema oferece a você, como dissemos anteriormente, uma visão geral dos conceitos mais importantes deste estudo. Ao segui-lo, será possível transitar entre os principais conceitos e descobrir o caminho para construir o seu processo de ensino-aprendizagem.

O Esquema dos Conceitos-chave é mais um dos recursos de aprendizagem que vem se somar àqueles disponíveis no ambiente virtual, por meio de suas ferramentas interativas, bem como àqueles relacionados às atividades didático-pedagógicas realizadas presencialmente no polo. Lembre-se de que você, aluno EaD, deve valer-se da sua autonomia na construção de seu próprio conhecimento.

Questões Autoavaliativas

No final de cada unidade, você encontrará algumas questões autoavaliativas sobre os conteúdos ali tratados, as quais podem ser de **múltipla escolha**, **abertas objetivas** ou **abertas dissertativas**.

Responder, discutir e comentar essas questões, bem como relacioná-las com a prática do ensino de Arquitetura de Computadores pode ser uma forma de você avaliar o seu conhecimento. Assim, mediante a resolução de questões pertinentes ao assunto tratado, você estará se preparando para a avaliação final, que será dissertativa. Além disso, essa é uma maneira privilegiada de você testar seus conhecimentos e adquirir uma formação sólida para a sua prática profissional.

As **questões de múltipla escolha** são as que têm como resposta apenas uma alternativa correta. Por sua vez, entendem-se por **questões abertas objetivas** as que se referem aos conteúdos matemáticos ou àqueles que exigem uma resposta determinada, inalterada. Já as **questões abertas dissertativas** obtêm por resposta uma interpretação pessoal sobre o tema tratado; por isso, normalmente, não há nada relacionado a elas no item Gabarito. Você pode comentar suas respostas com o seu tutor ou com seus colegas de turma.

Bibliografia Básica

É fundamental que você use a Bibliografia Básica em seus estudos, mas não se prenda só a ela. Consulte, também, as bibliografias complementares.

Figuras (ilustrações, quadros...)

Neste material instrucional, as ilustrações fazem parte integrante dos conteúdos, ou seja, elas não são meramente ilustrativas, pois esquematizam e resumem conteúdos explicitados no texto. Não deixe de observar a relação dessas figuras com os conteúdos, pois relacionar aquilo que está no campo visual com o conceitual faz parte de uma boa formação intelectual.

Dicas (motivacionais)

Este estudo convida você a olhar, de forma mais apurada, a Educação como processo de emancipação do ser humano. É importante que você se atente às explicações teóricas, práticas e científicas que estão presentes nos meios de comunicação, bem como partilhe suas descobertas com seus colegas, pois, ao compartilhar com outras pessoas aquilo que você observa, permite-se descobrir algo que ainda não se conhece, aprendendo a ver e a notar o que não havia sido percebido antes. Observar é, portanto, uma capacidade que nos impele à maturidade.

Você, como aluno dos Cursos de Graduação na modalidade EaD, necessita de uma formação conceitual sólida e consistente. Para isso, você contará com a ajuda do tutor a distância, do tutor presencial e, sobretudo, da interação com seus colegas. Sugerimos, pois, que organize bem o seu tempo e realize as atividades nas datas estipuladas.

É importante, ainda, que você anote as suas reflexões em seu caderno ou no Bloco de Anotações, pois, no futuro, elas poderão ser utilizadas na elaboração de sua monografia ou de produções científicas.

Leia os livros da bibliografia indicada, para que você amplie seus horizontes teóricos. Coteje-os com o material didático, discuta a unidade com seus colegas e com o tutor e assista às videoaulas.

No final de cada unidade, você encontrará algumas questões autoavaliativas, que são importantes para a sua análise sobre os conteúdos desenvolvidos e para saber se estes foram significativos para sua formação. Indague, reflita, conteste e construa resenhas, pois esses procedimentos serão importantes para o seu amadurecimento intelectual.

Lembre-se de que o segredo do sucesso em um curso na modalidade a distância é participar, ou seja, interagir, procurando sempre cooperar e colaborar com seus colegas e tutores.

Caso precise de auxílio sobre algum assunto relacionado a este *Caderno de Referência de Conteúdo*, entre em contato com seu tutor. Ele estará pronto para ajudar você.

Sistemas Numéricos e Conversões de Bases

1

1. OBJETIVOS

- Compreender os sistemas numéricos de diversas bases numéricas.
- Relacionar a aplicação dos sistemas numéricos binários e hexadecimal com sistemas computacionais.
- Aprender a conversão entre números de diferentes bases numéricas.
- Entender os mecanismos de soma e de subtração em sistemas numéricos diferentes.
- Compreender a subtração por complemento da base em sistemas numéricos de base 2 e sistemas numéricos de base 16.

2. CONTEÚDOS

- Sistemas numéricos de base 10, base 2, base 16 e de outras bases.
- Conversões numéricas entre sistemas numéricos de bases diferentes, de base 10 para qualquer base, de qualquer base para base 10.
- Conversões numéricas entre sistemas de base 16 para base 2 e de base 2 para base 16.
- Operações aritméticas entre sistemas de numeração de bases diferentes.
- Adição direta nas bases 2 e 16.
- Subtração direta nas bases 2 e 16.
- Subtração por complemento da base nas bases 2 e 16.

3. ORIENTAÇÕES PARA O ESTUDO DA UNIDADE

Antes de iniciar o estudo desta unidade, é importante que você leia as orientações a seguir:

1) A motivação é um fator fundamental para o sucesso de qualquer atividade. Esta unidade é muito importante para o estudo de *Arquitetura e Organização de Computadores*, pois, aqui serão apresentados os conceitos fundamentais do sistema de numeração que as máquinas digitais utilizam: o sistema de numeração binário (de base 2) e a sua relação com os outros sistemas de numeração utilizados para a programação em baixo nível ou de *hardware*.
2) Atente para os detalhes, pois às vezeso que nos parece simplescausa o maior problema durante o cálculo ou conversão de números em sistemas numéricos de bases diferentes.
3) Sempre que você estiver trabalhando com sistemas numéricos de bases diferentes, deixe ao seu alcance os dígitos de cada sistema de numeração ordenados em ordem crescente para consultas durante as operações aritméticas.
4) Você verá que os algarismos do sistema numérico hexadecimal são substituídos por letras após o algarismo nove. Para fixar, escreva sobre cada algarismo o seu valor numérico correspondente. Exemplo: A=10, B=11, C=12, D=13, E=14, F=15.
5) Tenha sempre à mãoos livros indicados na bibliografia básica e complementar. Não se contente em estudar apenas por meio de uma única fonte de informação.
6) Tire suas dúvidas com o seu tutor sempre que considerar necessário, e não deixe de levar para as aulas presenciais as suas dúvidas e questionamentos.

4. INTRODUÇÃO À UNIDADE

Você sabe como é o funcionamento do sistema numérico decimal (ou sistema numérico de base 10)? Ele é o mais utilizado para quantificar valores? Por quê?

Além do sistema numérico decimal, existem outros sistemas numéricos como, por exemplo, sistema binário (ou sistema numérico de base 2), sistema hexadecimal (ou sistema numérico de base 16) e muitos outros.

Nesta unidade, você entenderá como os sistemas numéricos funcionam e o que significa o número que indica a base.

VOCÊ SABIA QUE... ———————————————————————————

Para contar as horas, é utilizado o sistema numérico de base 24, enquanto para contar os minutos o sistema numérico é de base 60?

———————————————————————————————————

5. SISTEMA NUMÉRICO DECIMAL (BASE 10)

Talvez você já tenha ouvido falar em sistema numérico decimal ou de base 10.

O sistema numérico decimal (base 10) utiliza 10 algarismos (símbolos) para representar qualquer quantidade.

O número 10 não foi adotado por acaso. Antigamente, quando o homem percebeu a necessidade de criar um meio de quantificar os objetos, sentiu a necessidade de adotar uma base para realizar essa contagem. Como temos dez dedos nas mãos, a base 10 foi adotada, pelo fato de ser um valor visivelmente mais fácil de ser compreendido.

Observe os símbolos utilizados no sistema numérico decimal.

0	1	2	3	4	5	6	7	8	9

Para representar quantidades maiores, utilizamos pesos diferentes para cada posição do algarismo.

Como a base é 10, para cada posição à esquerda, o peso vai ser 10 vezes maior do que a posição à direita.

Assim, surgiram os nomes: unidade, dezena, centena, milhar, dezena de milhar, centena de milhar, milhão etc.

5248 =	5000 +	200 +	40 +	8
=	5 X 1000 +	2 X 100 +	4 X 10 +	8 X 1
=	5 X 10^3 +	2 X 10^2 +	4 X 10^1 +	8 X 10^0

A forma de decompor um número, de acordo com o exemplo anterior, serve para qualquer sistema numérico, bastando para isso mudar a base (em negrito) e manter o expoente da base como a posição do algarismo, lembrando que a posição mais à direita é a posição zero.

A forma de decompor um número pode ser chamada de **lei de formação**.

$N = A_n * B_n + A_{n+1} * B_{n+1} + A_{n+2} * B_{n+2} + A_{n+3} * B_{n+3} + A_{n+4} * B_{n+4}...$

N = Número.

A_n = Algarismo.

B_n = Base.

n = Posição do algarismo que vale zero para a posição mais à direita do número.

Vamos recordar: os números terminados com **0**, **2**, **4**, **6** e **8** são **pares** e os terminados com **1**, **3**, **5**, **7** e **9** são **ímpares**.

Para representar o número decimal, colocamos o número entre parênteses e a base como índice.

Exemplificando:

$(2596)_{10}$

$(2837)_{10}$

$(388)_{10}$

6. SISTEMA NUMÉRICO BINÁRIO (BASE 2)

Sistema binário é aquele que utiliza somente dois algarismos para representar qualquer quantidade.

Como o sistema tem somente dois algarismos, ele recebe o nome de *bit,* que é a abreviação de dígito binário em inglês.

O computador digital tem este nome porque a base de numeração em que ele trabalha é binária, ou seja, dentro dos circuitos digitais que compõem um computador, ele só entende a presença ou a ausência de sinal. A partir daí, constrói todas as informações, efetuando todas as operações que ele é capaz de realizar.

Observe os símbolos utilizados para representar qualquer quantidade do sistema binário.

0	1

Números binários terminados em **0** são **pares** e terminados em **1** são **ímpares**.

A **lei de formação** aplica-se a todos os sistemas numéricos e, portanto, também aplica-se ao sistema binário.

Observe o exemplo para um número de 4 bits.

0101 =	$0 \text{ X } 2^3$ +	$1 \text{ X } 2^2$	$0 \text{ X } 2^1$ +	$1 \text{ X } 2^0$
=	0 +	4 +	0 +	1
=	4 + 1 = 5 no sistema decimal			

Observe, a fórmula utilizada no exemplo anterior.

$N = \ldots \mathbf{A_{n+4} * B^{n+4} + A_{n+3} * B^{n+3} + A_{n+2} * B^{n+2} + A_{n+1} * B^{n+1}} + A_n * B^n$

n = Número que indica a posição do algarismo ou da base (inicialmente a posição vale zero).

A = Algarismo.

B = Base.

N = Número convertido para a base dez.

Para representar o número binário, colocamos o número entre parênteses e a base como índice.

Exemplificando:

$(0101)_2$

$(1001)_2$

$(111101010)_2$

7. SISTEMA NUMÉRICO HEXADECIMAL (BASE 16)

Sistema numérico hexadecimal é utilizado nos projetos computacionais, sejam eles projetos de *hardware* (parte física do computador) ou projetos de *software* (parte lógica do computador).

O sistema numérico hexadecimal é muito utilizado por tratar-se de um sistema mais próximo da base 10 para a representação de quantidades. Além disso, ele é de fácil conversão para a base 2, que é o sistema numérico utilizado nos computadores digitais.

Veja, a seguir, os símbolos que compõem o sistema hexadecimal.

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

Observe que os números de 10 a 15 foram representados pelas letras (de A a F).

Números hexadecimais terminados em 0, 2, 4, 6, 8, A, C e E são pares; os números terminados em 1, 3, 5, 7, 9, B, D e F são ímpares.

2FA1 =	2 X 16^3 +	15 X 16^2	10 X 16^1 +	1 x 16^0
=	8.192 +	3.840 +	160 +	1
=	8.192 + 3.840 + 160 + 1 = 12.193 no sistema decimal			

Desse modo, é importante saber o motivo pelo qual utilizamos as letras entre **A** e **F** para representar os algarismos maiores do que 10. Pois se ao invés das letras, fossem utilizados os números, como você iria diferenciar o seguinte valor: $(11)_{16}$ do $(B)_{16}$? Os dois seriam escritos na forma $(11)_{16}$.

Para representar o número hexadecimal, utilizamos a letra H após o número ou colocamos o número entre parênteses e a base como índice.

Exemplificando:

2A96**H** ou $(2A96)_{16}$

Para trabalhar com sistemas numéricos de outras bases, basta observar a lei de formação e não se esquecer de representar a base, colocando de preferência o número entre parênteses e a base como índice.

8. CONVERSÃO ENTRE BASES

Veja como é simples converter números entre bases diferentes.

Por exemplo: uma base qualquer para base 10.

Aplica-se a **lei de formação** conforme exemplos anteriores:

$N = A_n * B_n + A_{n+1} * B_{n+1} + A_{n+2} * B_{n+2} + A_{n+3} * B_{n+3} + A_{n+4} * B_{n+4...}$

N = Número.

A_n = Algarismo.

B_n = Base.

n = Posição do algarismo que vale zero para a posição mais à direita do número.

Exemplificando:

Conversão do sistema numérico binário para o decimal.

$(1101)_2$ =	1 X 2^3 +	1 X 2^2	0 X 2^1 +	1 X 2^0
=	8 +	4 +	0 +	1
=	8 + 4 + 1 = $(13)_{10}$ no sistema decimal			

$(101101)_2$ =	1 X 2^5 +	0 X 2^4	1 X 2^3 +	1 X 2^2 +	0 X 2^1 +	1 X 2^0
=	32 +	0 +	8	4	0 +	1
=	32 + 0 + 8 + 4 + 0 + 1 = $(45)_{10}$ no sistema decimal					

Conversão do sistema numérico hexadecimal para o decimal.

$(2FA1)_{16}$ =	2 X 16^3 +	15 X 16^2	10 X 16^1	1 X 16^0
=	8.192 +	3.840 +	160 +	1
=	8.192 + 3.840 + 160 +1 = $(12.193)_{10}$ no sistema decimal			

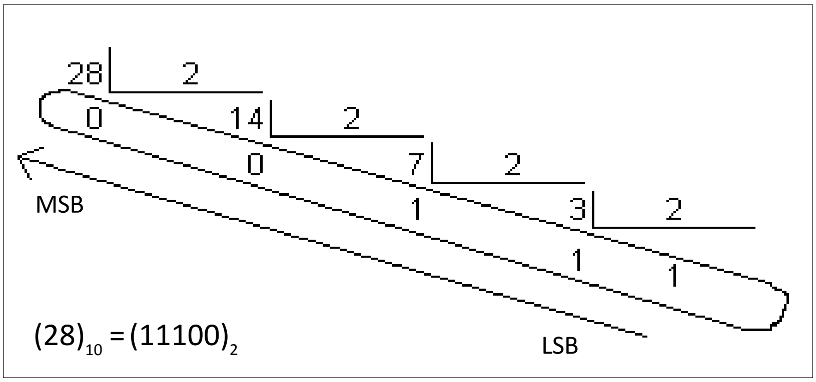
$(10A1)_{16}$ =	1 X 16^3 +	0 X 16^2	10 X 16^1	1 X 16^0
=	4.096 +	0 +	160 +	1
=	4.096 + 0 + 160 +1 = $(4.257)_{10}$ no sistema decimal			

Base 10 para qualquer base

Para converter um número de base 10 para uma base qualquer, por exemplo X, deve-se efetuar sucessivas divisões do número de base 10 pela base X, até que o quociente seja menor do que a base X. Para compor o número, pega-se o valor do último resultado e se estabelece uma relação com os restos das divisões anteriores, partindo-se da última divisão até a primeira.

Vejam os seguintes exemplos:

Conversão do número $(28)_{10}$ para o seu correspondente em binário:

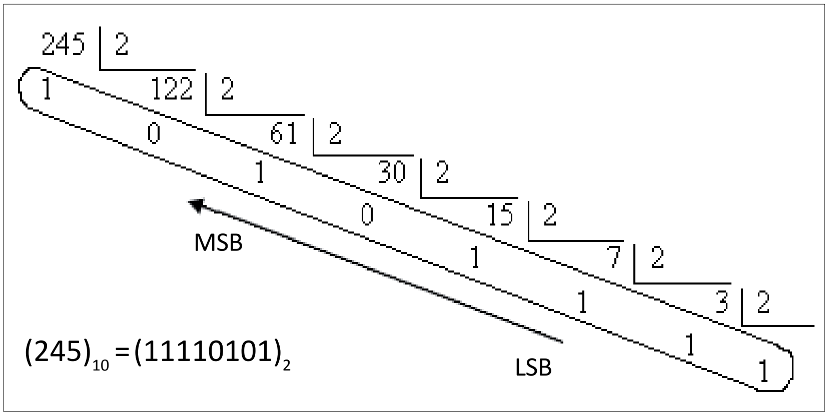


Analisando a conversão anterior, saiba que:

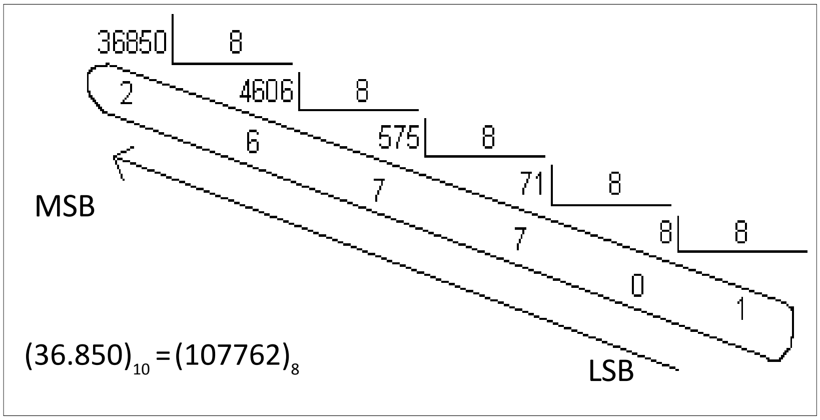
MSB indica o *bit* mais significativo (o que tem maior valor).

LSB indica o *bit* menos significativo (o que tem menor valor).

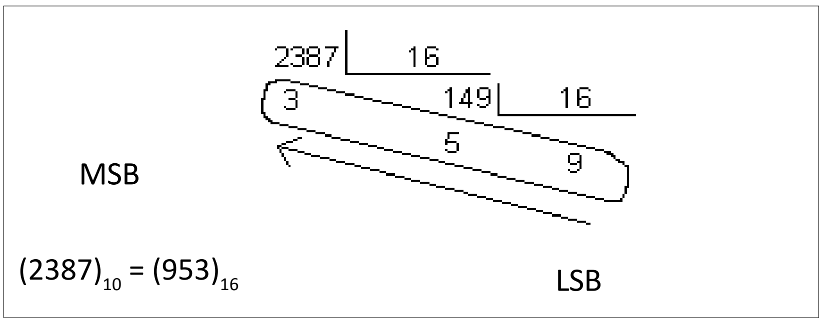
Conversão do número $(245)_{10}$ para o seu correspondente em binário:



Conversão do número $(36.850)_{10}$ para o seu correspondente na base 8:



Conversão do número $(2387)_{10}$ para o seu correspondente na base 16.



Conversão entre as bases 2 e 16

Com base nos conceitos estudados até agora, para convertermos um número de base diferente de 10 para outra base diferente de 10, teríamos primeiro que converter o número da base qualquer para a base 10 e, em seguida, para a base desejada.

A seguir, você verá como converter um número de base diferente de 10 para outra base diferente de 10, sem ter que convertê-lo primeiro para a base 10. Portanto, fique atento às explicações.

Conversão da base 2 para a base 16

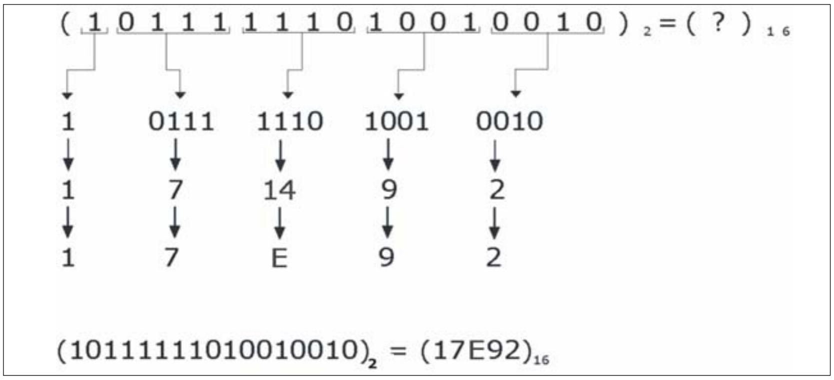
Agora, você verá como é que se faz para converter um número da base 2 para a base 16, sem passar pela base 10.

Divida o número da base 2 em grupos de 4 *bits* da direita para a esquerda e, depois, converta cada grupo de 4 *bits* diretamente para hexadecimal (da mesma forma que faria, se fosse converter para a base 10; quando o resultado for maior que 9, você deverá utilizar letras no lugar de números).

Como o maior número, que pode ser obtido com 4 *bits,* é 15, o valor do agrupamento sempre será representado por um único algarismo de base 16.

Depois, basta você **concatenar** todos os dígitos resultantes. Como? Basta apontar que a base resultante é 16.

Exemplificando:



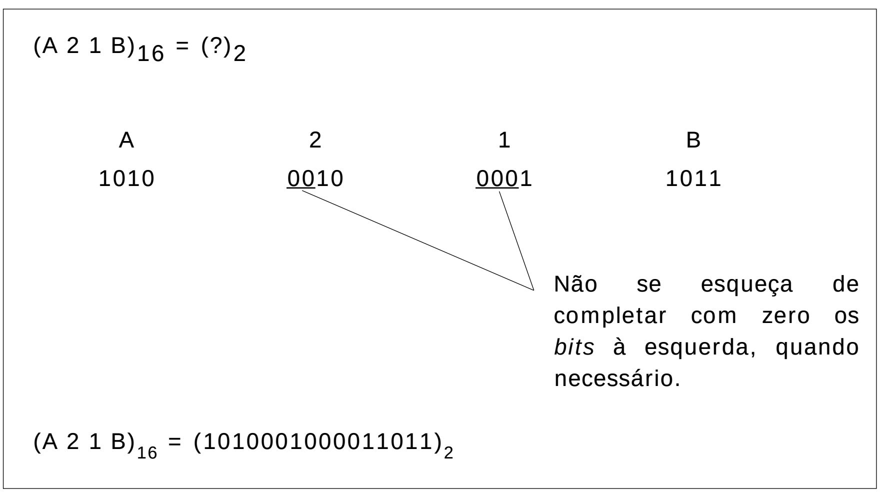
Conversão da base 16 para a base 2

O processo é o inverso do que vimos no exercício anterior.

Converte-se cada algarismo do sistema hexadecimal para a base 2, preenchendo, quando necessário, os dígitos à esquerda com zeros, a fim de obter sempre resultados de 4 *bits*.

Concatenam-se os grupos de quatro *bits.* Depois, basta indicar a base como quiser.

Exemplificando:



9. CONVERSÕES DE BASE COM PONTO FLUTUANTE

Neste tópico, você aprenderá a converter números com ponto flutuante de qualquer base para a base dez e converter números com ponto flutuante da base dez para qualquer base.

Conversões de números com ponto flutuante de qualquer base para a base dez

Para converter números de qualquer base para a base dez, devemos utilizar a lei de formação dos números,conforme já estudado nesta unidade.

Da mesma forma, podemos converter números com ponto flutuante de qualquer base para a base dez utilizando a mesma regra, porém escrevendo expoentes negativos para os algarismos que estão posicionados à direita da vírgula. Assim, temos:

$N = \mathbf{A_{n-4} * B^{n-4} + A_{n-3} * B^{n-3} + A_{n-2} * B^{n-2} + A_{n-1} * B^{n-1}} + A_n * B^n + A_{n+1} * B^{n+1} + A_{n+2} * B^{n+2} + A_{n+3} * B^{n+3} + A_{n+4} * B^{n+4} \ldots$

Onde:

n = Número que indica a posição do algarismo ou da base.

A = Algarismo.

B = Base.

N = Número convertido para a base dez.

Exemplos:

$(1011{,}11)_2 \quad = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} =$

$= 8 + 0 + 2 + 1 + 0{,}5 + 0{,}25 = (11{,}75)_{10}$

$(1001{,}01)_2 \quad = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} =$

$= 8 + 0 + 0 + 1 + 0 + 0{,}25 = (9{,}25)_{10}$

$(1010{,}111)_2 \quad = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} =$

$= 8 + 0 + 2 + 0 + 0{,}5 + 0{,}25 + 0{,}125 = (10{,}875)_{10}$

FA,39H $\quad = 15 \times 16^1 + 10 \times 16^0 + 3 \times 16^{-1} + 9 \times 16^{-2}$

$= 240 + 10 + 0{,}1875 + 0{,}11106 = (250{,}29856)_{10}$

292,ACH $\quad = 2 \times 16^2 + 9 \times 16^1 + 2 \times 16^0 + 3 \times 16^{-1} + 9 \times 16^{-2}$

$= 512 + 144 + 2 + 0{,}1875 + 0{,}03515625 = (658{,}22265625)_{10}$

Conversões de números com ponto flutuante da base dez para qualquer base

Para converter um número da base dez para outra base qualquer, deve-se inicialmente fazer a conversão da parte inteira, conforme já estudado nesta unidade, e, posteriormente, converter a parte fracionária, utilizando-se a regra das multiplicações sucessivas pela base em que se quer transformar o número. Para montar o resultado, deve-se pegar a parte inteira de cada multiplicação efetuada até alcançar um resultado cuja parte fracionária seja zero ou até o momento em que ocorra uma repetição na parte fracionária, indicando, assim, uma dízima periódica, ou mesmo uma repetição do tipo dízima não periódica.

Exemplos:

a) $(25{,}848)_{10} = ()_2$

Resolução da parte inteira do número apresentado.

$(25)_{10} = (11001)_2$

Resolução da parte fracionária do número apresentado:

0,848	0,696	0,392	0,784
X 2	X 2	X 2	X 2
1,696	1,392	0,784	1,568
1	**1**	**0**	**1**

$= (0{,}1101)_2$

Resposta:

$(11001)_2 + (0{,}1101)_2 = (11001{,}1101)_2$

b) $(54{,}624)_{10} = ()_2$

Resolução da parte inteira do número apresentado.

$(54)_{10} = (110110)_2$

Resolução da parte fracionária do número apresentado:

0,624	0,248	0,496	0,992
X 2	X 2	X 2	X 2
1,248	0,496	0,992	1,984
1	**0**	**0**	**1**

$= (0{,}1001)_2$

Resposta:

$(110110)_2 + (0{,}1001)_2 = (110110{,}1001)_2$

10. OPERAÇÕES ARITMÉTICAS EM SISTEMAS NUMÉRICOS DE BASE 2 E DE BASE 16

Você já parou para pensar nos resultados das somas numéricas que fazemos diariamente? E nas subtrações? Por que 5 mais 2 é igual a 7? E se esses valores não estivessem no sistema de numeração decimal, como é que essas operações seriam feitas?

Os computadores digitais trabalham o tempo todo no sistema numérico binário, por isso os computadores são denominados "computadores digitais". Sendo assim, não poderíamos deixar de lado a apresentação da técnica que essas maravilhosas máquinas utilizam para efetuaras operações aritméticas na base binária. Veremos, a seguir, como as operações de soma e subtração são executadas, de forma especial, no sistema de numeração binário.

Operação aritmética SOMA

Para entendermos como o computador executa uma soma, é fundamental entendermos inicialmente como podemos descrever tecnicamente o que ocorre em uma soma no sistema decimal.

Considere os algarismos decimais dispostos em ordem da seguinte forma:

0	1	2	3	4	5	6	7	8	9

Soma significa deslocamento à direita na sequência dos algarismos decimais, ou seja, 5 + 2 é igual a 7 porque, estando no 5 e executando dois deslocamentos à direita nos algarismos decimais ordenados, iremos parar no algarismo 7. Veja:

0	1	2	3	4	5	6	**7**	8	9
					->	->	^		

Vejamos agora o que acontece com a seguinte operação aritmética decimal:7 + 5 = (?)

0	1	**2**	3	4	5	6	7	8	9
->	->	^					->	->	->
← --(vai um)									

Então quer dizer que 7 + 5 = 2, certo? Claro que não. O que ocorre é que, quando não existem mais algarismos à esquerda e, mesmo assim, é necessário fazer um deslocamento, ocorre o retorno para o algarismo zero e continua-se a contar a partir daí (contando inclusive o retorno). Esse fato é conhecido como “estouro” e é o famoso “vaium”.

Agora, sim. Veja:

0	1	2	3	4	5	6	7	8	9
->	->	^					->	->	->
← -- (vai um)									

$$\begin{array}{rr} 1 & \\ & 7 \\ & 5 \quad + \\ \hline (1 & 2)_{10} \end{array}$$

Vamos analisar outros exemplos de soma direta na base 10 com mais dígitos:

1) $(737)_{10} + (475)_{10}$

0	1	2	3	4	5	6	7	8	9
->	->	^					->	->	->
← -- (vai um)									

	1		
7	3	7	
4	7	5	+
		2	

0	1	2	3	4	5	6	7	8	9
->	^		-> estouro	->	->	->	->	->	->
← ----------------------------------- (vai um)									

1	1		
7	3	7	
4	7	5	+
	1	2	

0	1	2	3	4	5	6	7	8	9
->	->	^					-> estouro	->	->
← -----------------------------------(vai um)									

1	1	1		
	7	3	7	
	4	8	5	+
1	2	1	2	

Resultado:

$(737)_{10} + (475)_{10} = (1212)_{10}$

2) $(51)_{10} + (94)_{10}$

0	1	2	3	4	5		7	8	9
				->	^				
Não teve estouro									

	5	1	
	9	4	+
		5	

0	1	2	3	4	5	6	7	8	9
->	->	->	->	^	->	->	->	->	->
← ----------------------------------- (vai um)									

1			
	5	1	
	9	4	+
1	4	5	

Resultado:

$(51)_{10} + (94)_{10} = (145)_{10}$

Agora que você já sabe como é o mecanismo da soma no sistema decimal, vamos analisar como é realizada a soma em sistemas numéricos diferentes da base dez.

O que muda de um sistema numérico para outro é a quantidade de algarismos para a representação das quantidades. O primeiro passo é escrever os algarismos na ordem crescente de forma idêntica à que já foi vista no sistema decimal e, a partir daí, contar os deslocamentos e os estouros, considerando os algarismos da base com a qualse está trabalhando.

A seguir, serão apresentados alguns exemplos de soma na base 16 (hexadecimal) e também na base 2 (binária).

Soma no sistema numérico hexadecimal (base 16):

Algarismos da base hexadecimal:

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

Veja uma operação simples em Hexadecimal:

5H + 7H = ?H.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
					>	>	>	>	>	>	>	^			
---------------------Não ocorre estouro ---------------------															

		5	
		7	+
		C	

Resposta:

5H + 7H = CH

Agora será apresentada uma operação com estouro em hexadecimal:

8H + DH = (?H)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
>	>	>	>	>	^			>	>	>	>	>	>	>	>
← --- (vai um)															

	1		
		8	
		D	+
	1	5	

Resposta:

8H + DH = 15H

Tenha sempre muito cuidado no momento de atribuir um valor numérico aos algarismos representados por letras. É muito simples, mas muitos erros acontecem por falta de atenção. Uma sugestão é deixar uma tabela sempre à mão com os valores dos algarismos da seguinte forma:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Observe, a seguir, outros exemplos em hexadecimal (base 16):

1) 476H + 1DEH = ?H

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
>	>	>	>	^		>	>	>	>	>	>	>	>	>	>
← --- (vai um)															

		1		
	4	7	6	
	1	D	E	+
			4	

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

← --- (vai um)

	1	1		
	4	7	6	
	1	D	E	+
		5	4	

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

---------------------Não ocorre estouro -----------------------

	1	1		
	4	7	6	
	1	D	E	+
	6	5	4	

Resposta:

476H + 1DEH = 654H

2) F1EH + FFH = ?H

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

← --- (vai um)

		1		
	F	1	E	
		F	F	+
			D	

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

← --- (vai um)

	1	1		
	F	1	E	
		F	F	+
		1	D	

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

← --- (vai um)

1	1	1		
	F	1	E	
		F	F	+
1	0	1	D	

Resultado:

F1EH + FFH = 101DH

Soma em binário (sistema numérico de base 2):

Algarismos do sistema numérico binário:

0	1

Somar no sistema numérico binário não é diferente de somar em outros sistemas numéricos. O problema é que, como ocorrem muitos estouros, é necessário um pouco mais de atenção para não errar.

Veja um exemplo simples de soma no sistema numérico binário:

$(0)_2 + (1)_2 = (?)_2$

0	1
>	^

			0	
			1	+
			1	

Resultado:

$(0)_2 + (1)_2 = (1)_2$

Observe agora, a apresentação de um exemplo com estouro:

$(1)_2 + (1)_2 = (?)_2$

0	1
^	>
← ---- estouro	

		1		
			0	
			1	+
		1	0	

Resultado:

$(1)_2 + (1)_2 = (10)_2$

Por esse motivo, deixo aqui uma frase bem humorada:

"Existem 10 tipos de pessoas: as que conhecem o sistema de numeração binário e as que não conhecem" (autor desconhecido).

Claro que, se fizermos uma análise um pouco mais crítica sobre a frase citada, perceberemos que se trata de um erro de português, caso o 10 esteja na base decimal, ou de notação numérica de base se os algarismos 10 estiverem na base numérica binária. Percebemos, dessa forma, a importância de indicarmos a base sempre que estivermos trabalhando com sistemas de numeração de bases diferentes.

Mesmo que o sistema padrão seja o decimal e estivermos apresentando um número nessa base, em um projeto onde os números de bases diferentes se misturam,devemos sempre fazer a indicação corretamente. Exemplo: $(34)_{10}$.

A seguir, serão apresentados outros exemplos de soma no sistema numérico binário:

1) $(101)_2 + (11)_2 = (?)_2$

2)

0	1
^	>
← ---- estouro	

		1		
	1	0	1	
	0	1	1	+
			0	

0	1
^	>
← ---- estouro	

	1	1		
	1	0	1	
	0	1	1	+
		0	0	

0	1
^	>
← ---- estouro	

1	1	1		
	1	0	1	
	0	1	1	+
1	0	0	0	

Resposta:

$(101)_2 + (11)_2 = (1000)_2$

3) $(101)_2 + (10)_2 = (?)_2$

0	1
>	^
----sem estouro ---	

	1	0	1	
	0	1	0	+
			1	

0	1
>	^
---- sem estouro ---	

	1	0	1	
	0	1	0	+
		1	1	

0	1
	^
--- sem estouro ---	

	1	0	1	
	0	1	0	+
	1	1	1	

Resposta:

$(101)_2 + (10)_2 = (111)_2$

Subtração direta em decimal e subtração direta em outras bases

A subtração direta ocorre da forma inversa à apresentada na soma, ou seja, a subtração é o deslocamento à esquerda no conjunto de algarismos dispostos em ordem crescente, que constituem a base numérica.

O "estouro" ou, ainda, o "vai um" agora passa a ser o "empresta um" da casa vizinha e,assim, devemos ficar atentos aos zeros, que serão importantes para esses casos. É necessário igualar as casas antes de iniciar uma operação de subtração em qualquer base.

A seguir, apresentaremos exemplos dos sistemas de numeração de base 10, de base hexadecimal e de base 2. Observe que todos os exemplos terão como resultado um número positivo.

Exemplos no sistema numérico de base 10 (decimal):

$(94)_{10} – (58)_{10}=$

0	1	2	3	4	5	6	7	8	9
<-	<-	<-	<-	<-		^	<-	<-	<-
--------------------vem um------------------------------- →									

	9	4	
	1		
	5	8	-
		6	

0	1	2	3	4	5	6	7	8	9
			^	<-	<-	<-	<-	<-	<-"vem um"
--------------------vem um--- →									

	9	4	
	1		
	5	8	-
	3	6	

Resultado:

$(94)_{10} – (58)_{10}= (36)_{10}$

Exemplo no sistema numérico hexadecimal (de base 16):

476H – 1DEH = ?H.

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

---------------------------vem um------------------------- →

	4	7	6	
		1		
	1	D	E	-
			8	

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
<	<	<	<	<	<	<	<- vem um		^	<	<	<	<	<	<

---------------------------vem um------------------------- →

	4	7	6	
	1	1		
	1	D	E	-
		9	8	

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		^	<	<- vem um											

---------------------Não ocorre estouro -----------------------

	4	7	6	
	1	1		
	1	D	E	-
	2	9	8	

Resposta:

476H – 1DEH = 298H.

Exemplo no sistema de numeração de base 2 (binário):

$(101)_2 - (10)_2 = (?)_2$

0	1
	^

----sem estouro ---

	1	0	1	
	0	1	0	-
			1	

0	1
<	^

-------- estouro -----à

	1	0	1	
	1			
	0	1	0	-
		1	1	

0	1
^	<
--- sem estouro ---	

	1	0	1	
	1			
	0	1	0	_
	0	1	1	

Resposta:

$(101)_2 - (10)_2 = (011)_2$

O problema aparece quando o resultado é negativo! Quando isso se dá, ocorrem infinitos estouros a partir do algarismo mais significativo (aquele que está mais à esquerda). Não devemos considerar esses estouros na formação do resultado, mas eles nos indicam que o resultado é negativo e está apresentado na forma de complemento da base.

Antes de prosseguirmos, vamos entender o que é o complemento de um número.

O complemento de um número é o que falta para a base. Por exemplo: no sistema de numeração decimal, o complemento de 4 é 6, pois é o que falta para alcançar a base, que é dez.

No sistema de numeração hexadecimal, o complemento de 7é 9, pois 7 + 9 é 16, ou seja, é o que falta para atingir a base.

Em outros sistemas de numeração, aconteceda mesma forma. No sistema de numeração binário, o complemento de 1 é 1, pois é o que falta para alcançar o 2, a base.

Mas, existe um problema. Se considerarmos complemento como sendo o que falta para a base, poderemos ter alguns estouros indesejados durante o cálculo de alguns dígitos.

Para resolver esse problema, devemos sempre considerar complemento como sendo o que falta para a base menos um, e, somente depois dos cálculos dígito a dígito, devolver o um que foi subtraído.

Assim,o complemento do número decimal 4 é o número decimal 5, e não o número decimal 6. O complemento do número hexadecimal 8 é 6 e não 7, como apresentado anteriormente, e o complemento do número binário 1 é 0 e não 1, como apresentado na definição.

Para esses casos, podemos considerar que o complemento de um número é o simétrico na apresentação dos algarismos de uma determinada base, ou seja, se você deslocou três algarismos do zero ao três, o algarismo simétrico é àquele que sofre três deslocamentos do último algarismo em direção à esquerda. Exemplo: para o sistema decimal, o simétrico de zero é nove, o simétrico de um é oito, o simétrico de dois é sete e, assim, sucessivamente Só não podemos esquecer de somar um no final da operação dígito a dígito. Observe os exemplos a seguir:

Exemplo para o sistema de numeração decimal

Algarismo:	0	1	2	3	4	5	6	7	8	9
Simétrico:	9	8	7	6	5	4	3	2	1	0

Exemplo para o sistema de numeração hexadecimal

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

Exemplo para o sistema de numeração binário

Algarismo:	0	1
Simétrico:	1	0

A seguir, confira alguns exemplos de exercícios para facilitar seu aprendizado.

Exemplos de exercícios para encontrar o complemento de um número: ––––––––––––––

Apresente o complemento dos números a seguir:

(100110)2 = (011001)2+(1)2=(011010)2

(10010)2 = (01101)2+(1)2=(01110)2

(1000101010)2=(0111010101)2+(1)2=(0111010110)2

FA25H = 05DAH + 1H = 05DBH

7836H = 87C9H + 1H = 87CAH

39840H = C67BFH + 1H = C67C0H

(345)10 = (654)10 + (1)10

(1565)10 = (8434)10+(1)10

––

Subtração com resultado negativo

Confira, a seguir, alguns exemplos comentados de subtrações com resultado negativo.

Exemplos no sistema numérico de base 10 (decimal):

$(58)_{10} - (94)_{10} = (-36)_{10}$

				Comentários
	5	8		
1				
	9	4	–	O “emprestaum” final indica que o resultado é negativo e, portanto,deverá ser complementado.
	6	4		
				Como o resultado é negativo, é necessário complementá-lo para obter o resultado final.
	3	5		
		1		
–	3	6		É importante indicar, com osinal “–”, que o resultado é negativo.

Exemplo no sistema numérico hexadecimal (base 16):

476H –FDEH = – B68H

					Comentários
	4	7	6		
1	1	1			
	F	D	E	–	O "emprestaum" final indica que o resultado é negativo e, portanto, deverá ser complementado.
	4	9	8		
					Como o resultado é negativo, é necessário complementá-lo para obter o resultado final.
	B	6	7		
			1		
–	B	6	8		É importante indicar, com o sinal "–", que o resultado é negativo.

Exemplo no sistema de numeração de base 2 (binário):

$(10)_2 - (101)_2 = (-\,011)_2$

					Comentários
	0	1	0		
1		1			
	1	0	1	–	O "emprestaum" final indica que o resultado é negativo e, portanto, deverá ser complementado.
	1	0	1		
					Como o resultado é negativo, é necessário complementá-lo para obter o resultado final.
	0	1	0		
			1	+	
–	0	1	1		É importante indicar, com o sinal "–", que o resultado é negativo.

Subtração por complemento da base

A subtração por complemento da base consiste em transformar uma subtração em uma soma.Para isso, devemos pegar o primeiro operando, somar com o complemento dígito a dígito do segundo operando e somar um.

É necessário ficar atento à operação do dígito mais significativo (última operação da soma). Agora tudo ficará invertido. Se, na soma, encontrarmos um estouro na última operação, esse estouro simplesmente irá indicar que o resultado é positivo e ele já está "pronto". Não é necessário descer o último "vai um" e nem complementar o resultado.

Por outro lado, se não houver nenhum estouro na operação do dígito mais significativo (última operação), aí, sim, temos a indicação de que o resultado é negativo e será necessário complementá-lo.Vejamos alguns exemplos:

Subtração pelo método do complemento da base com resultado positivo

<u>Exemplos no sistema numérico de base 10 (decimal):</u>

$(94)_{10} - (58)_{10} =$

Como os dois termos já possuem o mesmo número de dígitos, não é necessário igualar as casas; assim, basta fazer a soma do primeiro termo com o complemento digito a dígito do segundo termo mais um.

$(94)_{10} + (41)_{10} + (1)_{10} =$

Resolução:

1			
	9	4	
	4	1	+
		1	
	3	6	

Comentários
O estouro na operação final indica que o resultado é positivo e, portanto, já é o resultado final da operação de subtração. Observe que o estouro não desce para o resultado.
Como o resultado é positivo, não será necessário complementá-lo.

Resultado:

$(94)_{10} - (58)_{10} = (36)_{10}$

<u>Exemplo no sistema numérico hexadecimal (base 16):</u>

FDEH –476H =

Como os dois termos já possuem o mesmo número de dígitos, não é necessário igualar as casas; assim, basta fazer a soma do primeiro termo com o complemento dígito a dígito do segundo termo mais um.

FDEH + B89H + 1H =

1	1	1		
	F	D	E	
	B	8	9	+
			1	
	B	6	8	

Comentários
O estouro na operação final indica que o resultado é positivo e, portanto, já é o resultado final da operação de subtração. Observe que o estouro não desce para o resultado.
Como o resultado é positivo, não será necessário complementá-lo.

Resultado:

FDEH–476H =B68H

Exemplo no sistema de numeração de base 2 (binário):

$(101)_2 - (10)_2 =$

Para resolver este exemplo, devemos:

- igualar as casas, inserindo zeros à esquerda:

  $(101)_2 - (010)_2 =$

- fazer a soma do primeiro termo com o complemento dígito a dígito do segundo termo mais um:

  $(101)_2 + (101)_2 + (1)_2 =$

Resolução:

					Comentários
1		1			
	1	0	1		
	1	0	1	+	O estouro na operação final indica que o resultado é positivo e, portanto, já é o resultado final da operação de subtração.
			1		Observe que o estouro não desce para o resultado.
	0	1	1		
					Como o resultado é positivo, não será necessário complementá-lo.

Resultado:

$(101)_2 - (10)_2 = (011)_2$

Subtração pelo método do complemento da base com resultado negativo

Exemplos no sistema numérico de base 10 (decimal):

$(58)_{10} - (94)_{10} =$

Como os dois termos já possuem o mesmo número de dígitos, não é necessário igualar as casas; assim, basta fazer a soma do primeiro termo com o complemento dígito a dígito do segundo termo mais um.

$(58)_{10} + (05)_{10} + (1)_{10} =$

Resolução:

				Comentários
	1			
0	5	8		
	0	5	+	A falta do estouro na operação final indica que o resultado é negativo e, portanto, deverá ser complementado.
		1		
	6	4		
				Como o resultado é negativo, é necessário complementá-lo para obter o resultado final.
	3	5		
		1		
–	3	6		É importante indicar, com o sinal "–", que o resultado é negativo.

Resultado:

$(58)_{10} - (94)_{10} = (-36)_{10}$

<u>Exemplo no sistema numérico hexadecimal (base 16):</u>

476H – FDEH =

Como os dois termos já possuem o mesmo número de dígitos, não é necessário igualar as casas; assim, basta fazer a soma do primeiro termo com o complemento dígito a dígito do segundo termo mais um.

476H +021H + 1H =

					Comentários
	4	7	6		
0	0	2	1	+	A falta do estouro na operação final indica que o resultado é negativo e, portanto, deverá ser complementado.
			1		
	4	9	8		
	B	6	7		Como o resultado é negativo, é necessário complementá-lo para obter o resultado final.
			1		
–	B	6	8		É importante indicar, com o sinal "–", que o resultado é negativo.

Resultado:

476H – FDEH = – B68H

<u>Exemplo no sistema de numeração de base 2 (binário):</u>

$(10)_2 - (101)_2 =$

Para resolver este exemplo, devemos:

- igualar as casas, inserindo zeros à esquerda:

  $(010)_2 - (101)_2 =$

- somar o primeiro termo com o complemento dígito a dígito do segundo termo mais um: $(010)_2 + (010)_2 + (1)_2 =$

Resolução:

					Comentários
	1				
0	0	1	0		
	0	1	0	+	Ao contrário da subtração direta, a falta do estouro na operação final indica que o resultado é negativo e, portanto deverá ser complementado.
			1		
	1	0	1		
					Como o resultado é negativo, é necessário complementá-lo para obter o resultado final.
	0	1	0		
			1	+	É importante indicar, com o sinal "–", que o resultado é negativo.
–	0	1	1		

Resultado:

$(10)_2 - (101)_2 = (-011)_2$

Concluímos o estudo da nossa primeira unidade.Agora, sugerimos que você resolva as questões autoavaliativas a seguir e, assim, verifique o seu aprendizado.

11. QUESTÕES AUTOAVALIATIVAS

Sugerimos que você procure responder, discutir e comentar as questões a seguir que tratam da temática desenvolvida nesta unidade.

A autoavaliação pode ser uma ferramenta importante para você testar o seu desempenho. Se você encontrar dificuldades em responder a essas questões, procure revisar os conteúdos estudados para sanar as suas dúvidas. Esse é o momento ideal para que você faça uma revisão desta unidade. Lembre-se de que, na Educação a Distância, a construção do conhecimento ocorre de forma cooperativa e colaborativa; compartilhe, portanto, as suas descobertas com os seus colegas.

Confira, a seguir, as questões propostas para verificar o seu desempenho no estudo desta unidade:

1) Normalmente, trabalhamos com o sistema decimal, ou seja, é natural do ser humano trabalhar com um sistema numérico de base dez. Os computadores trabalham com sistemas binários, ou seja, sistemas de base dois, que utilizam apenas dois símbolos para representar qualquer quantidade. Usamos muitos outros sistemas de numeração, muitas vezes sem nem mesmo perceber, como é o caso das horas (base vinte e quatro) e os minutos (base sessenta). Sendo assim, explique os sistemas de numeração e cite exemplos.

2) Como é possível converter um número de qualquer base para a base dez?

3) Como é possível converter um número decimal para qualquer base?

4) Como deve ser aplicada a notação que indica a base de um número?

5) Explique o processo de soma nos sistemas de numeração decimal, hexadecimal e binário.

6) Explique o processo de subtração nos sistemas de numeração decimal, hexadecimal e binário.

7) Como se faz o complemento *bit* a *bit* de um número no sistema hexadecimal? E no sistema binário?

8) O que significa o termo "complemento de um número"?

9) Como se faz uma subtração por complemento da base?

12. CONSIDERAÇÕES

Nesta unidade,você teve a oportunidade de aprender um pouco sobre os sistemas de numeração. Inicialmente, estudamos o sistema decimal, o mais utilizado pelos seres humanos;em seguida,estudamos o sistema de numeração hexadecimal, que é aplicado principalmente na programação de baixo nível e também na configuração de computadores; e, por fim, estudamoso sistema binário (sistemas de numeração de base 2),o qual é um sistema utilizado pelos computadores digitais,ou seja, os programas em linguagem de máquina estão todos escritos utilizando zeros e uns, incluindo as instruções que são representadas pelos seus respectivos códigos de operação (*opcode*). Assim, os sistemas de numeração foram formalizados com o objetivo de capacitá-lo a trabalhar com sistemas de numeração de qualquer base.

Além disso, estudamos as conversões de valores, como mudanças da base dez para qualquer base e de qualquer base para a base dez, mudanças diretas entre o sistema de numeração binário e o sistema de numeração hexadecimal e vice-versa. Além disso, você conheceu o mecanismo de resolução de problemas aritméticos, como soma e subtração de números nas bases apresentadas, e a operação de complemento.

Agora você já está pronto para continuar os estudos na próxima unidade (sistemas digitais). Bons estudos!

13. BIBLIOGRAFIA

LOURENÇO, Antônio Carlos et al. *Circuitos digitais*.São Paulo: Érica, 1996.

STALLINGS, William. *Arquitetura e organização de computadores*: projeto para o desempenho. Tradução de Carlos Camarão de Figueiredo. 5. ed. São Paulo: Prentice-Hall, 2003.