**Eletrônica digital**

**Autoria: Danyelle Garcia Guedes e Jessica Laisa Silva**

#### APRESENTAÇÃO DO LIVRO

A sociedade atual caminha para um grau cada vez mais intenso no que diz respeito à dependência das tecnologias digitais para o desenvolvimento de atividades que executa rotineiramente. A tecnologia se desenvolve cada vez mais e com rapidez. Quase toda a totalidade desses produtos desenvolvidos faz uso da eletrônica embarcada, ou seja, de sistemas digitais embutidos, como computadores para a automação das ações desempenhadas por esse dispositivo. A eletrônica digital é a área de conhecimento que conceitua as tecnologias para os sistemas embarcados. A obra é fundamentada na construção de diversos dispositivos atuantes no tratamento e processamento das informações. Uma das fases importantes de um projeto de eletrônica digital consiste na implementação propriamente dita das funções eletrônicas nos dispositivos eletrônicos.

Neste livro, serão introduzidos elementos importantes para essa área. Serão apresentados desde os tipos de sinais, as funções lógicas à classificação de contadores. O estudante entenderá como se fundamenta a tecnologia da multiplexação, aplicando os multiplexadores. E compreenderá os conceitos relacionados aos dispositivos conversores A/D e D/A, bem como a sua precisão de emprego. Também, será apresentado como são desenvolvidos algoritmos simples em sistemas digitais programáveis e a lógica por trás do processamento de sinais digitais, bem como soluções computacionais, fundamentando dispositivos microcontroladores e microprocessadores, e a elaboração de programas para dispositivos FPGA por meio da programação VHDL.

#### SUMÁRIO

**Unidade 1**

Definição dos tipos de sinais na eletrônica

Sistema de numeração para circuitos digitais

Funções e portas lógicas empregadas em circuitos digitais

Fundamentos da lógica combinacional

**Unidade 2**

Códigos e Simplificação de Expressões de Circuitos Digitais

Fundamentos dos Codificadores e Decodificadores

Aspectos Essenciais dos Circuitos Aritméticos

Fundamentos dos Biestáveis Lógicos

**Unidade 3**

Dispositivos Contadores em Eletrônica Digital

Dispositivos Registradores em Eletrônica Digital

Dispositivos Multiplexadores em Eletrônica Digital

Dispositivos Conversores em Eletrônica Digital

**Unidade 4**

Aspectos Essenciais da Lógica Programável e Software

Fundamentação do Processamento de Sinais

Soluções Computacionais em Eletrônica Digital

Elementos Básicos de Programação VHDL

# **UNIDADE 01 - OBJETIVOS**

Olá. Seja muito bem-vindo à **Unidade 1**. Nosso objetivo é auxiliá-lo no desenvolvimento das seguintes competências profissionais até o término desta etapa de estudos:

1. Discernir a respeito dos fundamentos técnicos dos sinais digitais e analógicos, conhecendo os principais níveis, variáveis e conectivos lógicos.

2. Entender os sistemas de numeração binário e hexadecimal e calcular operações de adição, subtração, multiplicação e divisão de números binários e hexadecimais.

3. Aplicar as funções e portas lógicas em circuitos digitais, destacando as expressões booleanas e determinando expressões e circuitos lógicos.

4. Realizar análise lógica combinacional, destacando as propriedades universais NAND e NOR em circuitos digitais.

# **UNIDADE 01 - Capítulo 01**

# **Definição dos tipos de sinais na eletrônica**

**Sinais na eletrônica <T2>**

O funcionamento dos circuitos eletrônicos se baseia fundamentalmente nos tipos de sinais com os quais operam. Nesse sentido, os circuitos eletrônicos podem ser classificados conforme os tipos de sinais de operação. Assim, eles podem ser categorizados como circuitos analógicos e circuitos digitais.

Enquanto os circuitos analógicos são fundamentados pelo uso de sinais analógicos, os circuitos digitais operam por meio da transmissão de sinais digitais. Mas, afinal, o que é um sinal?

Em síntese, o sinal consiste em algo que transmite informação, a qual pode ser a respeito do estado ou do comportamento de algum sistema físico (OPPENHEIM; SCHAFER, 2013). Em termos matemáticos, representa-se um sinal por meio de uma função de uma ou mais variáveis independentes, que, em muitas ocasiões, consiste em uma variável que representa o tempo.

EXEMPLO: A fala pode ser um fenômeno representado por um sinal, no qual tal função, matematicamente, é expressa por meio da pressão acústica em termos de tempo.

Em geral, é conveniente tomar a variável independente de tal função como sendo o tempo, mas ela pode representar outras variáveis com base na aplicação.

Os sinais podem se apresentar por meio de um comportamento caracteristicamente contínuo ou, ainda, podem apresentar descontinuidades, ou seja, alterações abruptas de sua forma. Em outras palavras, os sinais podem ser contínuos ou descontínuos em função de sua variável independente.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

Para diferenciar um sinal contínuo de um sinal discreto, utilizam-se os colchetes na função do sinal discreto e os parênteses na função do sinal contínuo.

x(n) → sinal contínuo

x[n] → sinal discreto

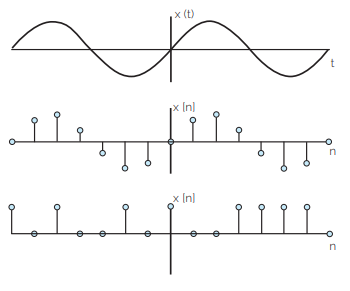
## FIM DA SEÇÃO **<IMPORTANTE>**

Um sinal pode ser determinado por meio de três categorias, a saber:

• Contínuo no tempo e na amplitude.

• Discreto no tempo e contínuo na amplitude.

• Discreto no tempo e na amplitude.



<Dia: extrair imagem do PDF. Pág 16>

<Leg: Três tipos de sinal.>

Os sinais digitais, ou sinais discretos no tempo e na amplitude, são configurados por meio de sequenciamentos de valores, que são identificados em instantes de tempo periódicos, e podem ser determinados por meio de valores inteiros e finitos. Por outro lado, os sinais analógicos, ou de tempo e amplitude contínuos, possuem estados determinados em qualquer instante de tempo. Assim, podem ser determinados pela identificação de qualquer valor numérico real.

Em sua generalidade, as grandezas que se apresentam na natureza encontram-se na forma de sinal analógico, ou seja, a maior parte das informações que são transmitidas no mundo real são funções que apresentam comportamento contínuo quanto à sua variação.

Enquanto os circuitos eletrônicos digitais operam com grandezas com comportamento discreto, os circuitos eletrônicos analógicos operam com grandezas que variam continuamente no tempo.

Mesmo as tecnologias da eletrônica analógica estando em menor uso na atualidade, o desenvolvimento das tecnologias digitais é fundamentado nos sinais analógicos. O conhecimento das duas áreas é primordial para os profissionais que atuam no desenvolvimento tecnológico do setor. Como mencionado, os sinais do mundo real são, em maioria, representados por sinais analógicos. Para a criação de dispositivos digitais e para a operação dos circuitos digitais, é necessário condicionar esses sinais analógicos para que consigam ser adequadamente empregados nos dispositivos digitais.

De acordo com o Floyd (2007), a maioria daquilo que se pode medir quantitativamente na natureza se encontra na forma analógica. Para Tocci, Widmer e Moss (2017), o alcance das capacidades da tecnologia digital, tendo como base a operação com sinais de entrada e saída analógicos em quatro passos, deve ter a seguinte sequência.

• Primeiro, deve-se implementar a conversão da variável física em sinal elétrico, analógico.

• Em seguida, deve-se converter o sinal elétrico em digital.

• Processa-se a informação digital.

• Por fim, é feita a conversão das saídas digitais em analógicas, que é o modelo demandado no mundo real.

Na natureza, fenômenos como temperatura, umidade, luminosidade, pressão atmosférica, pressão sonora, distância, velocidade e som variam de forma analógica. Assim, seu comportamento precisa ser condicionado às máquinas digitais.

EXEMPLO: Em uma medida do comportamento instantâneo da temperatura teremos um sinal contínuo, isto é, analógico. Entretanto, se a leitura for realizada apenas a cada hora, os valores da temperatura obtidos terão um comportamento discreto, mais especificamente, 24 valores de temperatura (Flouyd, 2007).

## INÍCIO DA SEÇÃO **<VOCÊ SABIA?>**

Os sistemas que trabalham com sinais podem ser contextualizados da seguinte forma geral: um dispositivo emissor codifica a informação em uma forma de sinal; esse sinal, por sua vez, é transmitido por um sistema de comunicação; e, por fim, é recebido por um componente receptor que atua na descodificação e interpretação da informação contida no sinal.

## FIM DA SEÇÃO **<VOCÊ SABIA?>**

Estabelecendo uma relação entre os sinais digitais e analógicos, é possível determinar algumas vantagens dos sinais digitais, elencadas a seguir.

• O projeto dos circuitos digitais é mais fácil de implementar. Estes se caracterizam por serem mais confiáveis e flexíveis e apresentarem maior robustez e, consequentemente, a partir das propriedades anteriores, um menor custo final em relação aos circuitos analógicos.

• Além disso, os sinais digitais são menos suscetíveis à interferência de ruídos (alterações indesejadas na tensão ou corrente são irrelevantes, tendo em vista que a sua entrada ou saída de sinal é medida em termos de seu estado lógico). Assim, variações de tensão, corrente, frequência ou fase não distorcem o comportamento discreto do sinal.

• O sinal digital se propaga com maior velocidade, pois apresenta menor quantidade de componentes e seu valor físico é menor.

**Sinais digitais e eletrônica digital <T2>**

Os circuitos e sistemas que fazem parte dos dispositivos da eletrônica digital são constituídos apenas em dois estados possíveis. Tais estados são representados por dois níveis de tensão: alto e baixo. Esses níveis podem ser representados, ainda, por níveis de corrente, níveis de bits ou níveis de ressaltos.

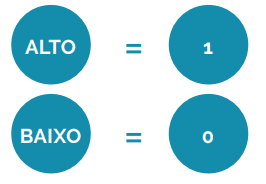
## INÍCIO DA SEÇÃO **<IMPORTANTE>**

A combinação desses estados é identificada, em sistemas digitais, como códigos que são utilizados para representar caracteres alfabéticos, símbolos, números e outras informações.

## FIM DA SEÇÃO **<IMPORTANTE>**

Para o sistema de numeração que apresenta apenas dois estados, tem-se a denominação de sistema binário. E, nesse caso, os dois níveis são representados pelos dígitos 1 e 0.

Um dígito binário é conhecido como bit (binary digit = dígito binário), assim, cada um dos dígitos 1 e 0 do sistema binário é identificado como um bit. O bit 1 é usado para representar um estado de maior tensão, alto, e o bit 0 para representar um estado de menor tensão, baixo. A afirmação anterior é uma representação definida como lógica positiva, a qual será adotada neste estudo.



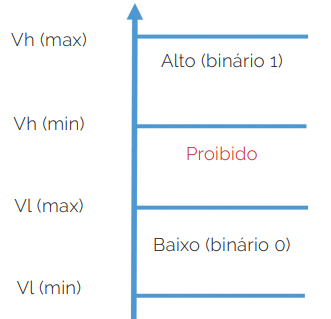
<Dia: recriar>

<Leg: Representação da lógica positiva.>

Caso, em dado sistema, o nível 0 seja utilizado como alto, e o nível 1, como baixo, estará se representando a lógica negativa. Nesse sentido, tem-se que os códigos são formados por meio de grupos de bits e são desenvolvidos para representar números, letras, símbolos, instruções ou qualquer outro tipo de grupo necessário à aplicação do sistema.

É importante estar atento ao fato de que, na prática, não há sobreposições para faixas aceitáveis dos níveis alto e baixo, ou seja, em um circuito digital, o nível de tensão alto pode estar representando qualquer tensão entre um valor mínimo e um valor máximo especificados. Para um nível baixo, este pode ser qualquer valor de tensão entre um valor mínimo e um máximo especificados (FLOYD, 2009).

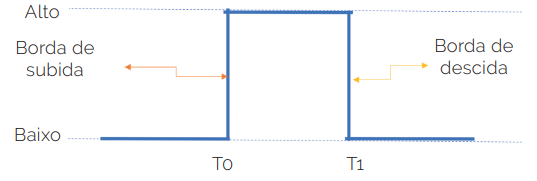
Considere as faixas de níveis representados na imagem a seguir. Para o nível alto, a tensão Vh(max) é o valor máximo de tensão e Vh(min) é o valor mínimo de tensão. No caso do nível baixo, Vl(max) é o valor máximo de tensão e Vl(min) é o valor mínimo de tensão para esse nível. Qualquer valor entre Vh(min) e Vl(max) é considerado inaceitável em uma operação considerada adequada. Dessa forma, valores nessa faixa são valores proibidos, pois podem ser considerados tanto altos quanto baixos, ou seja, a correta interpretação fica comprometida.



< Dia: extrair imagem do PDF. Pág 24 >

<Leg: Faixas de níveis lógicos de tensão.>

Sabendo disso, tem-se que as formas das ondas de um sinal digital são constituídas por níveis de tensão que comutam entre os estados alto e baixo. Observe a seguir que, para a geração de um único pulso positivo, apenas a tensão ou corrente passa pelo nível baixo para o nível alto e depois volta para o nível baixo.



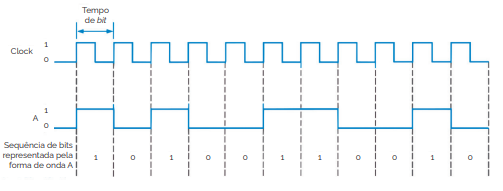
< Dia: extrair imagem do PDF. Pág 24 >

<Leg: Forma de pulso positivo ideal.>

Assim, a forma da onda digital é composta de uma série de pulsos. E cada pulso é formado por bordas de subida, que se inicia no instante T0, e por uma borda de descida, que ocorre no instante T1.

No caso da Figura Anterior, considera-se o pulso como sendo ideal, pois tanto a borda de descida como a de subida comutam em um tempo zero de forma instantânea, mas, na realidade, tais transições não ocorrem instantaneamente, e sim buscam se aproximar desse ideal. Em geral, as formas de ondas digitais são compostas de uma série de pulsos (que podem ser chamados de trem de pulsos) e classificadas como periódicas ou não periódicas.

Com base nessa configuração dos pulsos da onda digital, é possível definir o tempo de bit, que nada mais é que o intervalo de tempo que o bit ocupa na sequência de bits da forma da onda. Nos sistemas, tem-se que as formas de onda são sincronizadas com uma forma de onda de temporização de referência que se chama clock. O clock é definido como uma onda periódica que tem intervalos de pulsos igual ao tempo de um bit. A figura a seguir representa essa definição.



< Dia: extrair imagem do PDF. Pág 26 >

<Leg: Forma de onda de clock sincronizada com uma forma de onda que representa uma sequência de bits.>

O gráfico que apresenta a relação atual do tempo de duas ou mais formas de onda é definido como diagrama de temporização. Os dados são definidos como um grupo de bits que carregam alguma informação. No caso dos dados binários, estes se apresentam como formas de onda digitais.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Você deve ter aprendido que um sinal nada mais é que uma função que carrega consigo uma informação quando a ela está associado um sentido físico. Em termos matemáticos, representa-se um sinal por meio de uma função de uma ou mais variáveis independentes, que, em muitas ocasiões, consiste em uma variável que representa o tempo. Os sinais podem se apresentar por meio de um comportamento caracteristicamente contínuo ou, ainda, podem apresentar descontinuidades, ou seja, alterações abruptas da sua forma. Em outras palavras, os sinais podem ser contínuos ou descontínuos em função de sua variável independente. Os sinais digitais, ou sinais discretos no tempo e na amplitude, são configurados por meio de sequenciamentos de valores que são identificados em instantes de tempo periódicos e podem ser determinados por meio de valores inteiros e finitos.

# **UNIDADE 01 - Capítulo 02**

# **Sistema de numeração para circuitos digitais**

**Sistema decimal <T2>**

Na tecnologia digital, existem muitos sistemas de numeração disponíveis para uso; os mais habituais, no entanto, são os sistemas decimal, binário, octal e hexadecimal.

O sistema decimal é mais próximo da realidade geral da sociedade; trata-se de uma ferramenta presente na vida de todas as pessoas, mesmo aquelas que não trabalham com tecnologia digital, tendo em vista que é o sistema adotado para a representação de quantidades diversas na sociedade. O sistema decimal se caracteriza por possuir 10 dígitos fundamentais, de 0 a 9. Por isso, o sistema decimal é também chamado de sistema de base 10.

O sistema decimal se caracteriza por apresentar valor posicional associado a cada um dos seus dígitos. Isso quer dizer que, em função da posição ocupada por um dígito dentro de um número, é associado um valor para o respectivo dígito. Dessa maneira, a posição do dígito informa o valor relativo que ele representa. Logo, tem-se que a posição de cada um dos dígitos dentro de um número decimal indica a magnitude que a quantidade irá representar, e esta é dada por meio de pesos associados. No caso, os pesos são dados por meio de potências de 10. Assim, em geral, para um número com N posições, é possível realizar a contagem de 10N números diferentes, iniciando-se pelo zero.

**Sistema binário <T2>**

Para os sistemas digitais, o sistema de numeração mais adequado é o sistema binário, tendo em vista que este facilita o projeto do equipamento eletrônico ao permitir operar em apenas dois níveis de tensão. No caso decimal, seriam necessários 10 níveis de tensão diferentes.

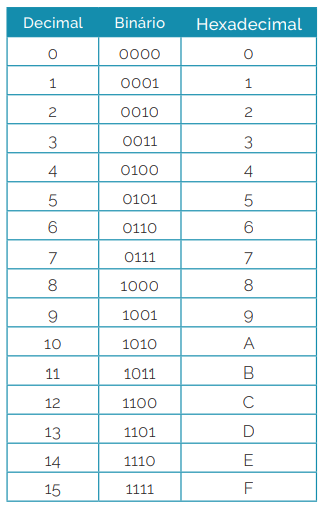
O sistema binário é também conhecido como sistema numérico de base 2. Nele existem apenas dois símbolos numéricos possíveis, o 0 e o 1. De maneira similar ao sistema decimal, o sistema binário apresenta valor posicional. Assim, para cada dígito binário do número, há um valor próprio ou peso, o qual é expresso em potências de 2.

Cada dígito binário é identificado como bit, para o sistema binário. No caso do exemplo anterior, este apresenta dois bits na porção inteira do número e três bits na porção fracionária. Pode-se identificar o bit mais significativo (MSB – most significant bit) como sendo o de maior peso, ou seja, o da esquerda, e o bit menos significativo (LSB – least significant bit), aquele que apresenta menor peso no número, o mais à direita.

Para o sistema binário, fazendo uso de um valor N de bits ou posições, tem-se que podem ser feitas 2N contagens de números diferentes. A última contagem será sempre com todos os bits do número igual a 1, o que equivalerá ao valor 2N-1. Por exemplo: para um número de quatro bits, tem-se 24 =16 contagens e 11112 =24 -1=1510.

**Sistema hexadecimal <T2>**

O sistema hexadecimal é composto de 16 caracteres, assim ele é de base 16. Uma importante particularidade do sistema hexadecimal é que este é formado por caracteres numéricos e alfabéticos, conforme apresentado na tabela a seguir.



<Dia: recriar>

<Leg: Equivalência entre os sistemas numéricos.

COTAS: Decimal Binário Hexadecimal

0 0000 0

1 0001 1

2 0010 2

3 0011 3

4 0100 4

5 0101 5

6 0110 6

7 0111 7

8 1000 8

9 1001 9

10 1010 A

11 1011 B

12 1100 C

13 1101 D

14 1110 E

15 1111 F>

Em geral, a maior parte dos sistemas digitais operam processando dados binários em grupos múltiplos de quatro bits. Isso torna o sistema hexadecimal mais simplificado, tendo em vista que cada dígito em hexadecimal corresponde a quatro dígitos de um número no sistema binário.

**Sistema octal <T2>**

Por fim, tem-se o sistema de numeração octal, o qual é representado por uma base 8. Nesse sentido, esse sistema utiliza oito caracteres para representação de seus números, de 0 a 7.

De maneira similar ao sistema hexadecimal, um número no sistema octal representa um número de três dígitos no sistema binário. Tanto o sistema octal como o hexadecimal são representações compactas de números binários grandes. O sistema octal, porém, foi substituído pelo sistema hexadecimal.

**Conversões entre os sistemas numéricos <T2>**

**Conversão de binários em decimais <T3>**

Um número maior que o conjunto-base no sistema decimal é determinado a partir da identificação do elemento-base correspondente associado ao valor da posição que este ocupa no número em questão.

O método dos pesos consiste em uma simples forma de realizar tais conversões. Um exemplo dessa representação do decimal é dado a seguir. Para o número decimal 81, são utilizados os elementos 8 e 1 do conjunto-base do sistema decimal. Atribuindo o valor posicional que cada elemento apresenta, tendo em vista a posição que ocupa no número, tem-se que o 8 representa a presença de 8 grupos de uma dezena, e o 1, de 1 grupo de unidade, ou seja:

8 × 101 + 1 × 100

8 × 10 + 1 × 1

81

Pensando de maneira similar, tem-se que, para o sistema binário, pode-se representar qualquer quantidade por meio do correto manuseio do conjunto de base desse sistema.

Da mesma forma, para a identificação de um número, deve-se somar cada algarismo com a base do sistema elevada ao valor da posição que o elemento ocupa no número. Nesse caso, a base é 2.

Para determinar, por exemplo, o valor, em decimal, dos seguintes valores: 111 e 1010.

• Para 111:

1 - ocupa a posição 2; 1 - ocupa a posição 1; 1 - ocupa a posição 0.

Fazendo os cálculos, obtém-se:

1 × 22 + 1 × 21 + 1 × 20

1 × 4 + 1 × 2 + 1 × 1

4 + 2 + 1

7

• Para 1010:

1 - ocupa a posição 3; 0 - ocupa a posição 2; 1 - ocupa a posição 1; 0 - ocupa a posição 0.

Fazendo os cálculos, obtém-se:

1 × 23 + 0 × 22 + 1 × 21 + 0 × 20

1 × 8 + 0 × 4 + 1 × 2 + 0 × 1

8 + 0 + 2 + 0

10

Outra maneira de converter um binário em decimal é por meio do método Double-dabble. Nesse modelo, o passo a passo apresentado a seguir deve ser respeitado.

• Identifica-se o bit 1 que está mais à esquerda no número binário.

• Em seguida, dobra-se esse elemento e é feita a soma dele ao bit que está à sua direita.

• O valor é anotado sob o próximo bit.

• Os passos 2 e 3 são repetidos até que o número binário seja finalizado.

EXEMPLO:

Para o número binário 111:

Multiplicando o valor mais à esquerda por 2:

1 × 2 = 2

Resultado é somado ao bit adjacente:

2 + 1 = 3

Resultado é multiplicado por 2:

3 × 2 = 6

Resultado é somado ao bit adjacente:

6 + 1 = 7

Número finalizado.

Para o número binário 1011:

1 × 2 = 2

2 + 0 = 2

2 × 2 = 4

4 + 1 = 5

5 × 2 = 10

10 + 1 = 11

**Conversão de sistema decimal em binário <T3>**

Há dois meios viáveis para a conversão de decimais inteiros em binários. No primeiro método, pode-se implementar o processo inverso do método de mudança de binário para decimal.

No caso, o número decimal é representado pela soma de potências de 2. Devem-se determinar os pesos, que, uma vez somados, resultariam no decimal equivalente.

910 = 8 + 1 = 23 + 0 + 0 + 20

1 0 0 12

No caminho demonstrado anteriormente, a ideia principal do mecanismo se baseia na segmentação do valor em um somatório no qual os elementos da soma são potências de 2, por exemplo: 16=24 . As potências que forem possíveis de ser identificadas são correspondentes ao bit 1 e as que não forem, por sua vez, são correspondentes ao bit 0.

Observe a conversão do valor a seguir.

7510 = 64 + 8 + 2 + 1 = 26 + 0+ 0 + 23 + 0+ 21 + 20

1 0 0 1 0 1 12

**Conversão entre sistemas hexadecimal e binário <T3>**

Para converter um binário em hexadecimal, realiza-se a operação direta. O número binário é separado em grupos de quatro bits e, a partir do bit mais à direita, deve-se substituir pelo hexadecimal equivalente.

EXEMPLO:

i) 1100101001012

1100; 1010; 0101

C; A; 5

Para a conversão do hexadecimal em binário, basta fazer o processo inverso. Para determinar o correspondente decimal de um número hexadecimal, primeiro se converte o hexadecimal em binário e, em seguida, converte-se o binário em decimal.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Você deve ter aprendido que na tecnologia digital existem muitos sistemas de numeração disponíveis para uso. Os mais habituais, no entanto, são os sistemas decimal, binário, octal e hexadecimal. O sistema decimal se caracteriza por apresentar valor posicional associado a cada um dos seus dígitos. Isso quer dizer que, em função da posição ocupada por um dígito dentro de um número, é associado um valor para o respectivo dígito. Dessa maneira, a posição do dígito informa o valor relativo que ele representa. O sistema binário é também conhecido como sistema numérico de base 2. Nele existem apenas dois símbolos numéricos possíveis, o 0 e o 1. O sistema hexadecimal é composto de 16 caracteres. Assim, ele é de base 16. Já o sistema de numeração octal é representado por uma base 8. Diversas operações podem ser desenvolvidas para a adequada conversão de um sistema numérico em outro.

# **UNIDADE 01 - Capítulo 03**

# **Funções e portas lógicas empregadas em circuitos digitais**

**Funções e portas lógicas <T2>**

A implementação da lógica necessária para o funcionamento dos circuitos digitais se dá por meio do desenvolvimento de funções lógicas. Para a representação das funções lógicas, são utilizadas portas lógicas, que consistem em circuitos eletrônicos que desenvolvem operações definidas com os elementos de entrada para a geração de saídas.

Todas as portas lógicas possuem uma representação simbólica específica para facilitar a construção dos projetos digitais. Essas representações seguem a lógica dos conectivos lógicos (E, OU e Negação).

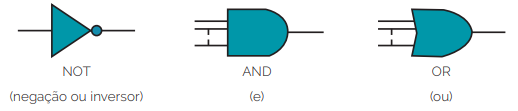
## INÍCIO DA SEÇÃO **<IMPORTANTE>**

Cabe enfatizar que a simbologia lógica das portas lógicas consiste em elementos padronizados de acordo com a Norma 91-1984 da ANSI/IEEE, a qual é adotada de forma convencional pela indústria privada e militar para documentação interna, bem como é o padrão adotado na literatura geral

## FIM DA SEÇÃO **<IMPORTANTE>**

Com base, pois, nos conectivos lógicos (E, OU e negação), as funções lógicas fundamentais executam três operações lógicas conhecidas como: AND (E), OR (OU) e NOT (negação ou inversor).

Em geral, as funções lógicas implementam operações específicas entre os elementos inseridos em suas entradas e geram uma saída. A figura a seguir apresenta a representação simbólica para as portas lógicas NOT, AND e OR.



< Dia: extrair imagem do PDF. Pág 41 >

<Leg: Operações lógicas básicas e seus símbolos equivalentes.>

Ao lado esquerdo são inseridos os dados de entrada e ao lado direito representa-se a saída produzida após a execução da operação lógica da porta nos dados de entrada.

**Tabelas-verdade <T3>**

Construir uma tabela-verdade consiste em uma técnica que serve para a descrição de como a saída de um circuito lógico irá se comportar, uma vez que ela é dependente dos estados lógicos da entrada.

Na tabela-verdade, relacionam-se todos os estados lógicos possíveis para as variáveis de entrada e as respectivas respostas após a aplicação da operação lógica referente à porta lógica em questão. As tabelas-verdade relacionam a quantidade de possibilidades de saída com as variações das entradas. O número de combinações possíveis é igual a 2N para uma tabela com N entradas.

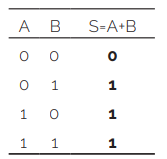
**Porta OU/OR <T3>**

A porta lógica OU/OR representa a primeira operação lógica mais básica. Ela é demonstrada pela expressão:

S = A + B, lê-se: S é igual a A ou B.

O sinal + não representa a operação de adição, representa a operação lógica OR/OU.

Em geral, a porta OR pode ter uma ou mais entradas e produz uma única saída. Se a saída for ALTA, ou 1, quer dizer que uma ou mais entradas apresentaram estado ALTO. Se, porém, uma saída for BAIXA, ou 0, quer dizer que todas as entradas tiveram nível BAIXO, ou 0. Mas, se pelo menos uma das entradas apresentar nível ALTO, ou 1, sendo que as demais apresentam nível BAIXO, mesmo assim a saídas terá nível ALTO. Dessa forma, é possível construir a tabela-verdade para essa porta tendo como base a sua expressão.



<Dia: recriar>

<Leg: Tabela-verdade porta OR com duas variáveis.

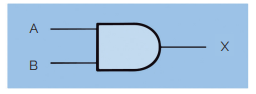
COTAS: A B S=A+B 0 0 0 0 1 1 1 0 1 1 1 1>

**Porta E ou AND <T3>**

A porta lógica E/AND representa a segunda operação lógica mais básica. Ela é demonstrada pela expressão:

S=A.B, lê-se: S é igual a A e B.

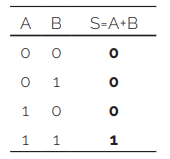
O sinal (.) representa a operação de multiplicação lógica entre variáveis que são inseridas na entrada da porta lógica AND/E.



< Dia: extrair imagem do PDF. Pág 43 >

<Leg: – Formato característico da porta AND.>

Na operação da porta AND, a saída será ALTO se todas as entradas também apresentarem nível ALTO. Para a ocorrência de qualquer entrada com nível BAIXO, a saída será imediatamente BAIXO. Dessa maneira, a principal finalidade da porta AND é determinar se as condições verdadeiras são simultâneas, indicadas pelo estado ALTO das entradas, produzindo um estado ALTO na saída. Dessa forma, é possível construir a tabela-verdade para essa porta também tendo como base a sua expressão.



<Dia: recriar>

<Leg: Tabela-verdade porta AND com duas variáveis.

COTAS: A B S=A+B 0 0 0 0 1 0 1 0 0 1 1 1>

**Porta Not/Inversor <T3>**

A operação NOT, de maneira divergente às operações AND e OR, possibilita realizar operações com uma única variável de entrada. Um exemplo disso pode ser ilustrado tomando-se a variável A e submetendo-a à operação NOT. Nesse caso, o resultado x será expresso da seguinte forma.

x = , onde lê-se: x é igual a A negado ou z é igual ao inverso de A. Ou seja, o valor de z será o oposto do valor lógico atribuído a A.



< Dia: extrair imagem do PDF. Pág 45>

<Leg: Operação de inversão.>

Assim, a operação do inversor sobre uma entrada indica que um nível lógico será alterado para uma saída com valor oposto. Dessa maneira, caso o estado de entrada seja ALTO, com a aplicação do inversor, a saída ficará no estado BAIXO e, se o estado de entrada for BAIXO, com a aplicação do inversor, a saída ficará no estado ALTO.

Em resumo, a operação NOT resulta em saídas de nível oposto ao nível da entrada.

**Operações e expressões booleanas <T2>**

De maneira geral, todos os tipos de circuitos lógicos podem ser demonstrados por meio do uso das três operações booleanas básicas apresentadas (AND, OR e NOT). Em uma mesma expressão, é possível encontrar, de acordo com o circuito lógico projetado, a associação dessas operações booleanas em diferentes configurações e é possível que exista confusão no momento da determinação de qual expressão deve ser realizada primeiro.

EXEMPLO: Considere que S=A.B+C.

Nesse caso, poderia ser desenvolvida primeiro a operação AND e depois a operação OR ou poderia ser desenvolvida primeiro a operação OR e depois a operação AND. A fim de que erros não sejam cometidos, deve-se ter em mente que as operações AND são realizadas primeiro e depois as operações OR, isso se não houver o uso de parênteses, operador, indicando a preferência dos termos a serem realizados primeiro.

**Analisando saídas para circuitos lógicos <T2>**

Com base na expressão booleana para uma saída em um circuito, é possível identificar o estado lógico da saída para qualquer que seja o conjunto de estados lógicos presentes em sua entrada.

Algumas regras podem ser utilizadas para avaliar a expressão booleana.

• Devem ser, inicialmente, implementadas todas as inversões dos termos simples presentes na expressão.

• As operações em parênteses são as próximas a serem desenvolvidas. • Deve-se, seguidamente, realizar todas as operações AND antes das operações OR que não estiverem entre parênteses.

• Caso a expressão tenha uma barra, deve-se desenvolver sua operação e depois realizar a inversão do resultado obtido.

EXEMPLO: Observe a expressão booleana a seguir e o respectivo desenvolvimento da saída do circuito, sabendo que A=0, B=0, C=1, D=1 e E=1.

É possível, ainda, realizar essa mesma análise utilizando a tabela- -verdade. Na verdade, esse método apresenta algumas vantagens:

- Possibilita a análise de uma porta ou combinação lógica de uma só vez.

– É de fácil conferência.

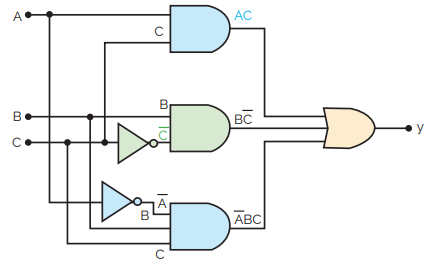
– Facilita a identificação de erros lógicos.

Nesse caso, deve-se construir a tabela-verdade com todas as entradas e com as operações, separadamente, de acordo com a ordem de prioridade.

**Desenvolvendo circuitos lógicos a partir de expressões booleanas <T2>**

A partir de uma expressão booleana que define um circuito, é possível construir diretamente o diagrama do circuito lógico. Por exemplo, para uma expressão S=A+B, é possível logo identificar que será necessária uma porta OR de duas entradas. E, dessa mesma forma, pode-se pensar para circuitos de maior complexidade.

Observe o exemplo de circuito apresentado na figura a seguir.



< Dia: extrair imagem do PDF. Pág 49 >

<Leg: Diagrama lógico.>

É possível encontrar o valor da expressão y final identificando-se todas as operações lógicas implementadas.

Observe que o circuito é composto de três entradas: A, B e C.

Na primeira porta lógica, a azul, é feita a operação AND com as entradas A e C.

Na segunda porta lógica, a verde, é feita a operação AND com a entrada B e com o inverso da entrada C.

Na terceira porta lógica, a roxa, é feita a operação AND com as entradas inversas de A, B e com a entrada C.

Por fim, as saídas das portas AND azul, verde e roxa são entradas para a porta OR, a amarela, que resulta na saída y.

A saída y é então identificada pela seguinte expressão lógica:

AC + B + BC = y

Esse mesmo processo pode ser utilizado em outros tipos de circuitos. No entanto, existem técnicas mais eficientes que simplificam esse processo e facilitam a identificação de expressões mais complexas.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Para a representação das funções lógicas, são utilizadas portas lógicas, que consistem em circuitos eletrônicos que desenvolvem operações definidas com os elementos de entrada para a geração de saídas. Construir uma tabela-verdade consiste em uma técnica que serve para a descrição de como a saída de um circuito lógico irá se comportar, de maneira que ela é dependente dos estados lógicos da entrada. A operação OR/OU resulta em saídas 1 sempre que qualquer uma de suas entradas for 1. A operação AND/E resulta em saídas 1 sempre que todas as entradas forem 1. A operação NOT resulta em saídas de nível oposto ao nível da entrada. Em uma mesma expressão, pode-se encontrar, de acordo com circuito lógico projetado, a associação dessas operações booleanas em diferentes configurações e é possível que exista confusão no momento da determinação de qual expressão deve ser realizada primeiro. A partir da expressão booleana para uma saída em um circuito, identifica-se o estado lógico da saída para qualquer que seja o conjunto de estados lógicos presentes na entrada do circuito. A partir de uma expressão booleana que define um circuito, pode- -se construir diretamente o diagrama do circuito lógico

# **UNIDADE 01 - Capítulo 04**

# **Fundamentos da lógica combinacional**

**Formas básicas da lógica combinacional <T2>**

As operações lógicas desenvolvidas a partir dos sistemas digitais se dão por meio de cálculos matemáticos. A álgebra booleana consiste no fundamento matemático mais apropriado e é o mecanismo que viabiliza as operações lógicas necessárias com o sistema de numeração binário. Essa álgebra foi desenvolvida por Georg Boole, em 1854. A álgebra booleana é atualmente a matemática dos sistemas digitais.

**Algumas propriedades da álgebra booleana <T2>**

Assim como as demais regras matemáticas, na álgebra de Boole existem algumas propriedades básicas que são fundamentais para a implementação de operações. Elas servem de base para o desenvolvimento das operações lógicas.

**Lei cumulativa**  **<T3>**

• Cumulativa da adição

Na operação de adição realizada entre duas ou mais variáveis, tem-se que a ordem dos fatores da soma resultará em um mesmo valor:

A+B=B+A

A cumulativa de adição é análoga à operação OR com circuitos lógicos.

• Cumulativa da multiplicação

No caso da multiplicação, tem-se que a ordem dos fatores não altera o valor do produto final, ou seja:

AB = BA

A cumulativa da multiplicação é equivalente à operação lógica da porta AND.

**Lei associativa <T3>**

Independentemente da ordem em que se apresentem os fatores de uma operação, o resultado final tenderá sempre a permanecer o mesmo.

• Associativa da adição

Realizando a adição com três variáveis para exemplificar a lei em questão, tem-se que não há distinção no valor do resultado final caso se inicie a soma entre os dois primeiros ou os dois últimos elementos antes de fazer o somatório final.

Na operação OR, nesse contexto, quando se opera com mais de duas variáveis, o resultado será o mesmo, independentemente da forma que se opte por realizar a soma das variáveis.

A + (B + C) = (A + B) + C

• Associativa da multiplicação

De forma similar à adição, ocorre a operação associativa da multiplicação. Independentemente das variáveis que se opte por iniciar a operação na expressão matemática, o resultado do produto final será invariável.

A(BC) = (AB)C

**Lei distributiva <T3>**

Por fim, tem-se a propriedade distributiva. Nesse caso, quando se rescreve a operação de multiplicação em função de dois elementos, obtém-se um produto final que não sofre alteração.

A(B + C) = AB + AC

Assim, expressões booleanas definem as operações de um circuito, e os diagramas de circuito podem ser construídos a partir de tais expressões, basta seguir todas as propriedades e encontrar as formas simplificadas das expressões.

## INÍCIO DA SEÇÃO **<SAIBA MAIS>**

Uma importante maneira de simplificar expressões booleanas se dá por meio do diagrama de Karnaugh. Um mapa de Karnaugh provê um método sistemático para simplificação de expressões Booleanas e, se usado adequadamente, produz a expressão de soma-de-produtos ou de produto-de-somas mais simples possível, conhecida como expressão mínima. (TOCCI; WIDMER; MOSS, 2007, p. 20)

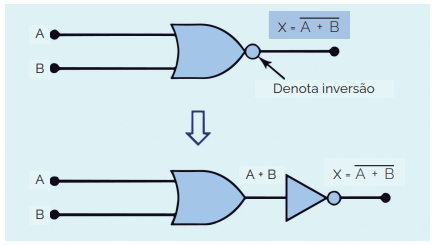
## INÍCIO DA SEÇÃO **<SAIBA MAIS>**

**Portas NAND e NOR <T2>**

Além das portas lógicas básicas AND, OR e NOT, é possível trabalhar com circuitos lógicos fazendo uso de duas outras importantes portas lógicas, também muito utilizadas. As portas NAND e NOR, ademais, são portas lógicas adicionais de grande uso.

É possível fazer uso das portas AND e da porta OR associadas diretamente com o inversor, e tal associação resulta em uma porta já definida como NAND e NOR, respectivamente.

No caso da porta NOR com duas entradas lógicas, a operação implementada é a mesma da porta OR, mas com um inversor, isso resulta na seguinte operação base: x =

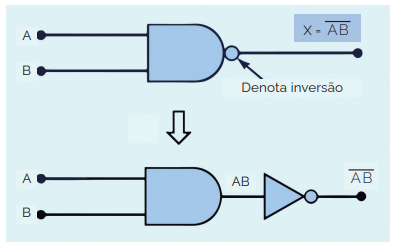


< Dia: extrair imagem do PDF. Pág 61 >

<Leg: Porta NOR.>

No caso da porta NAND, é implementada uma operação semelhante a AND, mas com uso de um inversor. A expressão fica da seguinte forma:

x =



<Dia: extrair imagem do PDF. Pág 61 >

<Leg: Porta NAND.>

## INÍCIO DA SEÇÃO **<SAIBA MAIS>**

Existe, ainda, a porta lógica identificada por EX-OR ou porta OR-exclusiva, a qual é formada a partir da combinação de portas lógicas: duas portas AND, uma porta OR, e dois inversores. Outra porta lógica passível de uso é a EX-NOR, dada pela inversão à porta EX-OR.

## FIM DA SEÇÃO **<SAIBA MAIS>**

A partir de tais funções lógicas, é possível produzir diversos circuitos lógicos combinacionais para várias aplicações. E tal associação das portas lógicas ajuda no desenvolvimento de diversos problemas lógicos.

A partir das diferentes portas lógicas, é possível construir diversas combinações simples e complexas de circuitos. Essas combinações possibilitam a construção de operações de circuitos somadores, subtratores, codificadores, decodificadores e outros que são aplicados na produção de sistemas digitais diversos. Assim, um circuito combinacional consiste em um circuito que tem uma saída dependente das combinações entre os níveis de entrada do circuito. Os níveis lógicos de saídas são sempre dependentes das combinações dos níveis lógicos de entrada.

Os circuitos lógicos combinacionais podem ser utilizados na resolução de problemas em que respostas são associadas a situações específicas inseridas por meio dos níveis de entrada lógicos.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Você deve ter aprendido que, para a representação das funções lógicas, são utilizadas portas lógicas e que as operações lógicas desenvolvidas a partir dos sistemas digitais se dão por meio de cálculos matemáticos. A álgebra booleana consiste no fundamento matemático mais apropriado e é o mecanismo que viabiliza as operações lógicas necessárias com o sistema de numeração binário. A álgebra de Boole apresenta regras cumulativas, associativas e distributivas que devem ser seguidas. Diversos métodos de simplificação podem ser implementados em expressões booleanas, entre elas pelo mapa de Karnaugh, pelas tabelas-verdade etc. Os circuitos lógicos combinacionais podem ser utilizados na resolução de problemas em que respostas são associadas a situações específicas inseridas por meio dos níveis de entrada lógico.

# **UNIDADE 02 - Capítulo 01**

# **Códigos e Simplificação de Expressões de Circuitos Digitais**

**Boole e DeMorgan para simplificação de funções <T2>**

É por meio dos fundamentos da álgebra booleana que são implementadas as operações matemáticas com sistemas digitais. As operações booleanas são utilizadas relacionando os componentes das portas lógicas NOT, AND, OR, NAND e NOR, por exemplo. Por meio da álgebra de Boole, definiram-se algumas leis e regras de operações descritas resumidamente por meio das expressões a seguir.

Primeiro, perceba que se define como variável a simbologia empregada na representação de uma grandeza lógica.

Exemplo;

Observe as seguintes relações: A=0 ou B=0 ou A=1 ou B=1. Nesses casos, A e B representam as variáveis.

Em um pensamento semelhante, pode-se definir como complemento o inverso da variável. A leitura do complemento de uma variável B, por exemplo, é B negado ou B barrado.

Exemplo;

Observe as seguintes relações: Se A=0, o complemento de A é dado por . Assim, representa o complemento de A. Nesse caso, lê-se A negado ou A barrado. Por fim, pode-se definir uma literal como sendo a variável ou o complemento dessa variável. É possível relacionar literais lógicos mediante diversas regras fundamentais, como as operações de adição, multiplicação, comutação, complementação, distribuição e associação. Tais regras podem ser identificadas pelas expressões gerais apresentadas a seguir, que podem ser empregadas em operações booleanas como mecanismos facilitadores da execução de processos de simplificação de expressões.

Na adição, tem-se as seguintes relações possíveis:

A + 0 = A A + 1 = 1 0 + 0 = 0 0 + 1 = 1

1 + 0 = 1 1 + 1 = 1 A + A = A A + = A

Na multiplicação, tem-se as seguintes relações possíveis:

A . 0 = 0 A . 1 = A 1 . 0 = 0 1 . 1 = 1

0 . 0 = 0 0 . 1 = 0 A . A = A A . = 0

Na comutatividade, tem-se as seguintes relações possíveis:

A + B = B + A

A . B = B . A

Na complementação, tem-se as seguintes relações possíveis:

A = 0 → =1

A = 1 → =0

A =

Na associatividade, tem-se as seguintes relações possíveis:

A + (B + C) = (A + B) + C = A + B + C

A . (B . C) = (A . B) . C = A . B . C

Na distributividade, tem-se as seguintes relações possíveis:

A . (B + C) = A . B + A . C

Ainda relacionando expressões importantes para a implementação do processo de simplificação de expressões booleanas, é possível a aplicação de dois teoremas importantes originados pelo matemático DeMorgan, que ficaram conhecidos como teoremas de DeMorgan. Os teoremas de DeMorgan são originários da verificação da equivalência existente entre as portas lógicas NAND e OR negativa e da equivalência entre as portas NOR e AND negativa.

Um dos teoremas de DeMorgan afirma que “o complemento de um produto de variáveis é igual à soma dos complementos das variáveis” (FLOYD, 2007, p. 207). Ou seja, quando duas ou mais variáveis são submetidas a uma operação lógica do tipo AND, o valor dos complementos é equivalente ao valor fornecido em uma operação lógica do tipo OR entre os complementos das variáveis individuais.

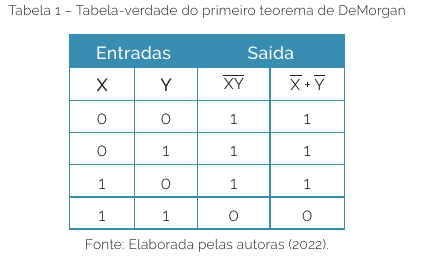
Em termos de expressão, tem-se a seguinte relação:

( 1° Teorema de DeMorgan)

O outro teorema de DeMorgan afirma que “o complemento de uma soma de variáveis é igual ao produto dos complementos das variáveis” (FLOYD, 2007, p. 208). Ou seja, quando duas ou mais variáveis são submetidas a uma operação lógica do tipo OR, o valor dos complementos é equivalente ao valor fornecido em uma operação lógica do tipo AND entre os complementos das variáveis individuais. Em termos de expressão, tem-se a seguinte relação:

(2° Teorema de DeMorgan)

Observe a seguir as representações em termos de portas lógicas das equivalências propostas por meio dos teoremas de DeMorgan. Observe, ainda, as seguintes tabelas-verdade para as respectivas equivalências de DeMorgan.

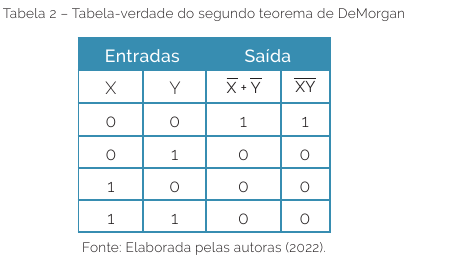


<Dia: recriar tabela da p. 72>

<Leg: Tabela-verdade do primeiro teorema de DeMorgan.>

<COTAS:

Entradas X 0 0 1 Y 0 1 Saída 1 1 0 1 1 1 0 1 1 1 0 Fonte: Elaborada pelas autoras (2022).>



<Dia: recriar tabela da p. 72 >

<Leg: Tabela 2 – Tabela-verdade do segundo teorema de DeMorgan>

<Cotas: Entradas X 0 0 1 Y 0 1 Saída 1 0 0 1 1 0 0 1 0 0 0 Fonte: Elaborada pelas autoras (2022). >

Em termos gerais, a simplificação, torna o custo associado menor, tendo em vista que simplifica a implementação e a fabricação desses circuitos. Em termos de operação, a simplificação potencializa a capacidade de processamento dos dados, reduz o consumo e os atrasos de propagação de sinais.

**Códigos de caracteres <T2>**

A eletrônica digital e os sistemas digitais operam com uso de codificadores e decodificadores. Isso se dá pelo fato de que o sistema de representação para tratar dados em sistemas digitais não é o mesmo que o mundo real opera.

Todas as informações enviadas do mundo real para os circuitos digitais se configuram inicialmente como sinais de formatos analógicos. Assim, todos os sinais originados com as informações do mundo real precisam ser transformados em arranjos que possam ser processados em sistemas digitais.

Da mesma forma, todas as informações transferidas e processadas precisam ser convertidas em configurações compreensíveis pelo homem. Os códigos são todos sistemas de representação que não são compreendidos diretamente por um ser humano.

Mediante códigos são desenvolvidas as representações dos dados nas máquinas digitais, e a respectiva padronização é necessária para que todas as máquinas consigam se comunicar e transferir dados e informações de maneira correta.

Os sistemas de representação mais utilizados para relacionar os dados digitais com os dados do mundo real são o sistema decimal, hexadecimal, binário e octal. Qualquer dado numérico representado no sistema decimal pode ser transformado em um equivalente em binário. Dessa forma, os 0s e 1s do sistema binário podem ser codificados em elementos decimais, por exemplo.

No mundo real, os sinais são analógicos, o que significa que a todo instante os sistemas digitais desenvolvem procedimentos de conversão de base, de forma a possibilitar que as informações do mundo real sejam operadas nos dispositivos digitais.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

Os processos de conversão e os relacionamentos dos códigos podem se caracterizar, a depender do tipo e do tamanho da informação, como operações exaustivas, demoradas e complexas.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Segundo Tanenbaum (2013), um conjunto de caracteres mínimo é formado idealmente por 26 letras maiúsculas, por 26 letras minúsculas, pelos algarismos decimais (0 a 9) e por caracteres especiais (espaço, vírgula, entre outros). No caso das operações de transferência de caracteres em sistemas computacionais, há a atribuição de valores ou números associados a cada um desses caracteres.

O mapeamento das informações com base em números inteiros é dado como código de caracteres. Para computadores que estabelecem comunicação, precisam usar o mesmo código. A fim de viabilizar a comunicação entre computadores, foram criados códigos padrão. A seguir serão detalhados os códigos BCD, Gray e ASCII.

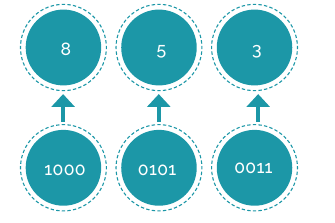
**Sistema BCD <T2>**

O código ou sistema BCD (Binary Coded Decimal), também identificado como decimal codificado em binário, compreende uma forma de codificação de números decimais em códigos binários. O código BCD possui 10 grupos de códigos, e os dígitos decimais são codificados por um mecanismo de combinação entre 4 bits binários, ocorrendo a codificação dígito a dígito.

Segundo Tocci, Widmer e Moss (2007), caso cada dígito de um decimal seja representado por um binário, obtém-se um código decimal codificado em binário (BCD). Os dígitos decimais podem possuir o valor 9 como máximo. Assim, faz-se necessário o uso de 4 bits para codificação de um dígito em binário.

Exemplo:

Observe o decimal 85310 .



<Dia: recriar>

<Leg: Representação de decimais em dígitos binários.

Fonte: Elaborada pelas autoras (2022).>

Em BCD, os dígitos decimais 853 correspondem aos seguintes grupos de 4 bits: 1000 0101 0011, respectivamente. Cada dígito decimal é codificado em binário puro, sendo 4 bits por dígito. Em códigos BCD, são aproveitados 10 grupos, não sendo usados os números 1010, 1011, 1100, 1101, 1110, e 1111.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

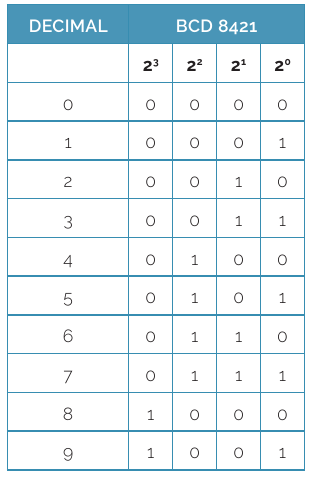
Em BCD, os números 1010, 1011, 1100, 1101, 1110 e 1111 são considerados bits proibidos.

## FIM DA SEÇÃO **<IMPORTANTE>**

Entre os códigos BCD de 4 bits disponíveis, o mais utilizado e conhecido é o 8421. Mas há diversos outros, como 7421, 5211 e 2421.

**BDC 8421 <T2>**

O termo 8421 se refere ao valor posicional do elemento em um número binário. Ou seja, descrevem, respectivamente, 23=8, 22=4, 21=2, 20=1.



<Dia: recriar>

<Leg: Concepção do código 8421.>

<COTAS: DECIMAL BCD 8421 23 22 21 20

0 0 0 0 0

1 0 0 0 1

2 0 0 1 0

3 0 0 1 1

4 0 1 0 0

5 0 1 0 1

6 0 1 1 0

7 0 1 1 1

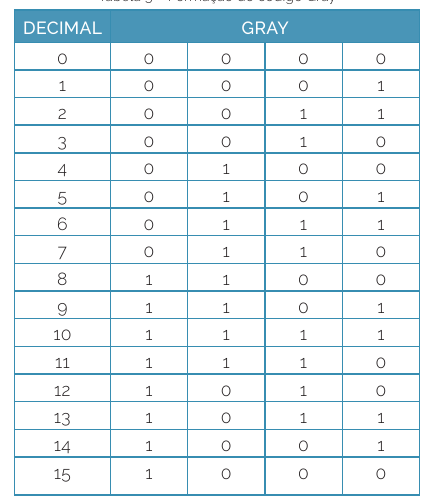
8 1 0 0 0

9 1 0 0 1>

Nos casos dos códigos BCD 7421, o BCD 5211 e o BCD 2421, a conversão para decimal é dada de forma similar ao BCD 8421. As formações dos códigos são apresentadas na tabela subsequente.

**Código Gray <T2>**

Os sistemas digitais executam operações rapidamente, respondendo às alterações que se processam nas entradas digitais. À medida que se detecta alguma variação nas entradas, o sistema responde rapidamente. Uma grande variedade de condições de entrada acarreta um comportamento mais complexo do sistema. A frenética mudança que pode ocorrer em bits de uma sequência de contagem binária pode acarretar situações de equivocada interpretação das entradas. Pensando na minimização desses riscos, o código Gray foi desenvolvido. No código Gray, há a variação de apenas um bit de um número para outro.



<Dia: recriar>

<Leg: Formação do código Gray.>

<COTAS: DECIMAL GRAY

0 0 0 0 0

1 0 0 0 1

2 0 0 1 1

3 0 0 1 0

4 0 1 0 0

5 0 1 0 1

6 0 1 1 1

7 0 1 1 0

8 1 1 0 0

9 1 1 0 1

10 1 1 1 1

11 1 1 1 0

12 1 0 1 0

13 1 0 1 1

14 1 0 0 1

15 1 0 0 0>

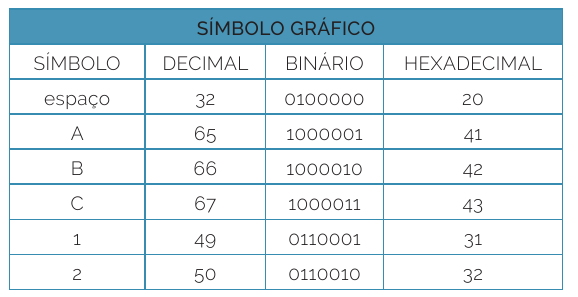
Nesse caso, a conversão se inicia no bit mais significativo (MSB) do binário e este se torna o Gray mais significativo (MSB). Em seguida, esse binário MSB é comparado com o próximo bit binário; se forem iguais, o Gray seguinte é 0, mas, se forem diferentes, o Gray seguinte é 1.

**ASCII <T2>**

O Código Padrão Americano de Troca de Informações, ou ASCII, (*American Standard Code for Information Interchange*) é alfanumérico.

O teclado de um computador tem um microprocessador dedicado que constantemente escaneia (“lê”) o circuito do teclado para detectar quando uma tecla foi pressionada e liberada. Uma única varredura é gerada pelo software do computador representando aquela tecla em particular. O código de varredura é então convertido em código alfanumérico (ASCII) para ser usado pelo computador. (FLOYD, 2007, p. 106)

O ASCII possui 128 caracteres e símbolos representados por um código de 8 bits, sendo o mais significativo igual a 0. Vai de 00 até 7F em hexadecimal. Os 32 primeiros componentes são não gráficos e servem para operações de controle (exemplo: nulo, escape etc.). Já os outros componentes são símbolos gráficos (exemplo: letras maiusculas e minúsculas, números decimais, símbolos, sinais de pontuação etc.). O ASCII destina-se à transferência de informações alfanuméricas e armazenamento interno de dados digitados por um usuário.



<Dia: recriar>

<LEG: Equivalentes entre ASCII, decimais, binários e hexadecimais.>

<COTAS: SÍMBOLO DECIMAL BINÁRIO HEXADECIMAL

espaço 32 0100000 20

A 65 1000001 41

B 66 1000010 42

C 67 1000011 43

1 49 0110001 31

2 50 0110010 32>

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Todas as formas possíveis de simplificação podem ser utilizadas para tornar os circuitos lógicos configurações mais simples. Tabelas-verdade, mapa de Karnaugh, álgebra de Boole, teoremas de DeMorgan são meios para a simplificação. No que diz respeito às operações booleanas, é possível relacionar literais lógicos por meio de diversas regras fundamentais, como operações de adição, multiplicação, comutação, complementação, distribuição e associação. Os teoremas de DeMorgan são originários da verificação da equivalência existente entre as portas lógicas NAND e OR negativa e da equivalência entre as portas NOR e AND negativa. A eletrônica digital e os sistemas digitais operam com uso de codificadores e decodificadores. Isso ocorre porque o sistema de representação para tratar dados em sistemas digitais não é o mesmo que o mundo real opera. Os códigos são todos os sistemas de representação que não são compreendidos diretamente por um ser humano. Para viabilizar a comunicação entre computadores, foram criados códigos padrão, os códigos BCD, Gray e ASCII.

# **UNIDADE 02 - Capítulo 02**

# **Fundamentos dos Codificadores e Decodificadores**

**Informações e equipamentos digitais <T2>**

Os dispositivos desenvolvidos para operar com sinais digitais são conhecidos como sistemas digitais. Estes, por sua vez, são desenvolvidos para processar dados em formato de bits. Assim, 0s e 1s são utilizados para transferência e processamento de dados e informações nesses dispositivos. Para possibilitar a comunicação dos dados do mundo real com os sistemas digitais, a informação precisa ser convertida em formatos entendíveis pelos sistemas digitais. Isto é, para que os sistemas digitais transmitam informações interpretáveis pelo ser humano, é necessário que os dados sejam convertidos no formato mais usual da rotina humana.

Os sistemas computacionais constituem-se fundamentalmente por processadores, memória e dispositivos de entrada e saída de dados, que, interconectados, configuram os computadores digitais. Por meio dos dispositivos de entrada e saída, as informações são inseridas no computador ou transmitidas para o usuário.

O teclado de um computador é um exemplo de dispositivo de entrada de informações, pelo qual o usuário pode inserir dados no sistema. O funcionamento geral de um teclado pode ser entendido da seguinte forma: imagine que o usuário esteja inserindo números; estas são as entradas inseridas. A entrada do teclado é dada em valor decimal, tais informações são codificadas para uma linguagem própria do instrumento digital, que, por sua vez, opera em binário. O valor da informação em binário é decodificado e apresentado no componente de saída do dispositivo em decimal, que, por sua vez, é compreendida pelo usuário.

A lógica combinacional fundamenta-se no fato de que dois ou mais estados de entrada em um circuito digital definem um ou mais estados de saída. Os codificadores e decodificadores são os circuitos lógicos combinacionais.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

No desenvolvimento de projetos de eletrônica digital, os codificadores e os decodificadores possuem um papel importante, sendo empregados no processo de conversão dos dados de um formulário para outro.

## FIM DA SEÇÃO **<IMPORTANTE>**

Codificadores e decodificares são utilizados com frequência em sistemas de comunicação, como telecomunicações, redes e transferência de dados, em que a informação é levada de uma extremidade para a outra. Ao final do agente receptor, os dados codificados são coletados do código e então processados para exibição.

O código binário de N dígitos pode ser usado para armazenar 2N elementos distintos de informação codificada. É para isso que codificadores e decodificadores são usados. Os codificadores convertem 2N linhas de entrada em um código de N bits, e os decodificadores decodificam os N bits em 2N linhas.

**Codificadores <T2>**

Um codificador é um dispositivo que converte o sinal de dados ativo em um formato de mensagem codificada, podendo ser entendido também como um dispositivo que converte o sinal analógico em sinais digitais.

Quando um sinal de entrada é aplicado a um codificador, os circuitos lógicos envolvidos nele convertem essa entrada específica em saída binária codificada. Ou seja, o codificador é um dispositivo ou um transdutor que converte as informações de um formato para outro formato.

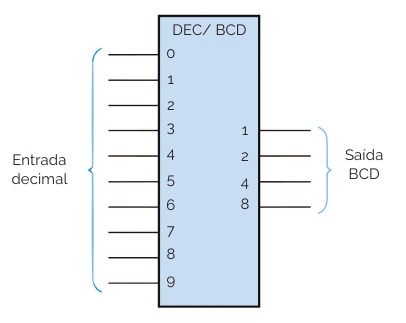
De acordo com Floyd (2007):

Um codificador é um circuito lógico que realiza essencialmente a função “inversa” do decodificador. Um codificador aceita um nível ativo em uma de suas entradas representando um dígito, tal como um dígito decimal ou octal, e o converte em uma saída codificada, tal como binário ou BCD. Codificadores também podem ser implementados para codificar vários símbolos e caracteres alfabéticos. O processo de conversão de símbolos familiares ou números em um formato codificado é denominado de codificação. (FLOYD, 2007, p. 340)

O codificador compreende um circuito combinacional que converte dados e informações binárias na forma de 2N linhas de entrada em N linhas de saída, que representam N código de bits para a entrada.

**Codificador de decimal para BCD <T3>**

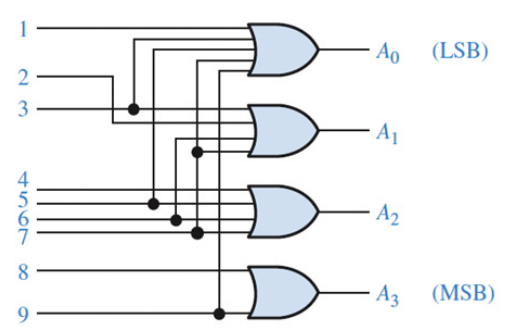
O codificador de decimal para BCD possui 10 entradas, sendo que cada uma é para cada dígito decimal. Além disso, ele apresenta quatro saídas que correspondem ao código BCD.



<Dia: extrair imagem do PDF. PG. 88>

<LEG: Símbolo lógico para um codificador de decimal para BCD.>

A implementação do circuito lógico para a realização da codificação de cada dígito decimal para BCD pode ser feita com expressões lógicas. É necessário apenas realizar a operação OR com as linhas da entrada dos decimais mais adequados. O resultado da lógica do codificador é dado segundo a figura subsequente.



< Dia: extrair imagem do PDF. pg. 90>

<Leg: Diagrama lógico básico de um codificador de decimal para BCD.>

Tendo em vista que as saídas BCD são todas de nível BAIXO, quando não houver entradas ALTAS para uma entrada de dígito 0 não é necessária. Assim, quando um nível ALTO ocorre em alguma linha de entrada de dígito decimal, os níveis apropriados aparecem nas quatro linhas de saída BCD.

**Decodificadores <T2>**

A codificação se baseia na transferência da informação de um tipo de código para outro. A decodificação, por outro lado, compreende o processo inverso ao da codificação. O decodificador atua na conversão da linguagem de máquina para linguagem ou código de outro tipo, que possa ser entendido pelo usuário.

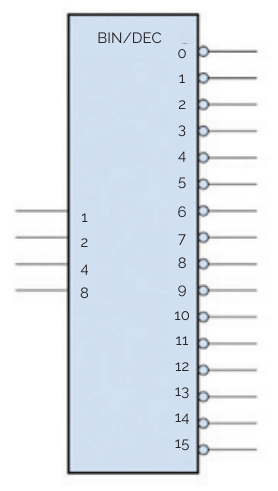
Se retomarmos o exemplo do usuário digitando números no teclado de um sistema digital de um computador, verificamos que o papel da decodificação é representado pelo display, que apresenta o número que foi digitado. Assim como os codificadores, o decodificador consiste em um circuito combinacional, mas com funcionamento inverso. O decodificador é um circuito usado para transformar o código em um conjunto de sinais.

Um decodificador gera o sinal original como saída do sinal de entrada codificado e converte N linhas de entrada em 2N linhas de saída. Uma porta AND pode ser usada como o elemento básico de decodificação, tendo em vista que produz uma saída alta somente quando todas as entradas são altas. A entrada de um decodificador é composta de código BCD e, na saída, constam as 10 linhas referentes aos decimais.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

A regra geral diz: são N linhas de entrada para 2N linhas de saída. Mas há decodificadores que apresentam valor de saída inferior a 2N.

## FIM DA SEÇÃO **<IMPORTANTE>**



< Dia: extrair imagem do PDF. PG. 92>

<LEG: Símbolo lógico para um decodificador de BCD para decimal.>

A decodificação de todas as combinações possíveis de quatro bits demanda o uso de 16 portas de decodificação, o que corresponde a um decodificador de 4 linhas para 16 linhas. É possível visualizar que, caso seja necessária uma saída ativa em nível BAIXO para cada número decodificado, o decodificador final completo se dará por meio do uso de portas NAND e inversores. Caso se deseje produzir saídas ativas em nível ALTO, pode-se usar portas AND.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Você deve ter aprendido que, para possibilitar a comunicação dos dados do mundo real com os sistemas digitais, a informação precisa ser convertida em formatos compreensíveis pelos sistemas digitais. Como se sabe, a lógica combinacional fundamenta-se no fato de que dois ou mais estados de entrada em um circuito digital definem um ou mais estados de saída. Os codificadores e decodificadores são os circuitos lógicos combinacionais. Um codificador é um dispositivo que converte o sinal de dados ativo em um formato de mensagem codificada ou pode ser entendido também como um dispositivo que converte o sinal analógico em sinais digitais. Assim como os codificadores, o decodificador consiste em um circuito combinacional, mas com funcionamento inverso. O decodificador é um circuito usado para transformar o código em um conjunto de sinais.

# **UNIDADE 02 - Capítulo 03**

# **Aspectos Essenciais dos Circuitos Aritméticos**

**Circuitos aritméticos <T2>**

Um circuito aritmético consiste em um conjunto de portas com um conjunto separado de entradas para cada número que deve ser processado. As portas são conectadas de forma a realizar uma ação aritmética, e as saídas do circuito da porta são os dígitos do resultado (adição, subtração, multiplicação ou divisão).

Dessa forma, os circuitos aritméticos são capazes de realizar sete operações aritméticas diferentes, fazendo uso de apenas um único circuito composto. Ele usa um somador completo para realizar essas operações.

**Somadores básicos <T2>**

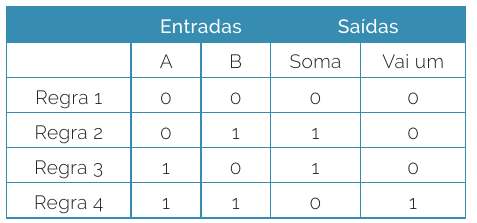
Os circuitos somadores são amplamente utilizados em computadores e calculadoras por executarem operações de adição com dois números binários por vez, sendo que cada número binário pode apresentar vários dígitos.

Os somadores são importantes em computadores e também em outros tipos de sistemas digitais nos quais dados numéricos são processados. Uma compreensão da operação básica de um somador é fundamental no estudo de sistemas digitais. (FLOYD, 2007, p. 314)

Os tipos de circuitos somadores mais básicos são o meio-somador e o somador-completo.

**Meio-somador <T3>**

O meio-somador, ou HA (Half-Adder), consiste em um tipo de circuito lógico combinacional formado pelas portas XOR e AND. Caracteriza-se por apresentar duas entradas e duas saídas. As saídas são formadas por um componente que representa a soma Σ; e por um componente carry out, que significa “vai um”. O circuito meio-somador executa a operação de somas com as suas duas entradas, os dois bits apenas.



<Dia: recriar>

<Leg: Meio-somador.>

<COTAS: Entradas Saídas

A B Soma Vai um

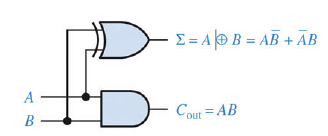
Regra 1 0 0 0 0

Regra 2 0 1 1 0

Regra 3 1 0 1 0

Regra 4 1 1 0 1 >

Conforme Floyd (2007, p. 314), “o meio-somador aceita dois dígitos binários em suas entradas e produz dois dígitos binários em suas saídas, um bit de soma e um *bit* de *carry*”. O *carry* é dado por uma operação AND entre as entradas.

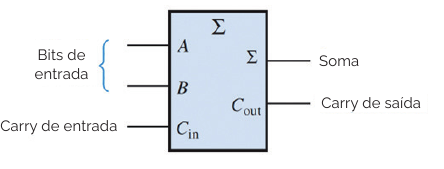


< Dia: extrair imagem do PDF. PG. 97>

<Leg: Diagrama lógico do meio-somador.>

**Somador-completo <T3>**

O meio-somador é um circuito que busca facilitar o processo de adição entre dois números binários em um algarismo. No entanto, compreende um circuito que não é suficiente no processo de soma de números binários que possuem mais de um algarismo, pois não permite o transporte do *carry* para a outra posição na coluna. Para suprir essa falta, utiliza-se o somador-completo, ou FA (full-adder). Segundo Floyd (2007, p. 315), “o somador-completo aceita dois bits de entrada e um carry de entrada, e gera uma saída de soma e um *carry* de saída. Um somador-completo tem um *carry* de entrada, enquanto um meio-somador não tem”.



< Dia: extrair imagem do PDF. P. 98>

<Leg: Símbolo lógico do somador-completo.>



<Dia: recriar>

<Leg: Somador-completo.>

<COTAS: Entradas Saídas

A B C entrada Soma Vai um

0 0 0 0 0

0 1 1 0 1

0 0 0 0 1

0 1 1 1 0

1 0 0 0 1

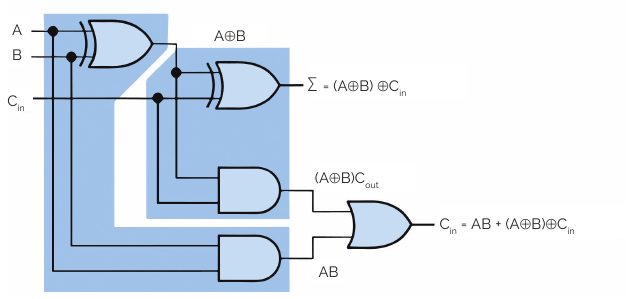
1 1 1 1 0

1 0 0 1 0

1 1 1 1 1>

No circuito somador-completo, dois bits são somados na entrada com o carry de entrada. O meio-somador gera a soma de bit por meio da operação XOR. Assim, para o carry na entrada do somador-completo, é necessário desenvolver a operação XOR. Dessa forma, tem-se a seguinte expressão para soma no somador-completo:

Para executar a função de soma no somador-completo, devem ser usadas duas portas XOR, cada uma com duas entradas, sendo a primeira responsável por gerar o termo A⊕B e a segunda por gerar uma saída com o *carry* de entrada e com a resposta da saída da primeira operação.



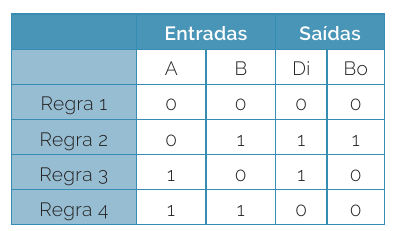
< Dia: extrair imagem do PDF. p. 99>

<Leg: Circuito lógico completo para um somador-completo.>

Como resposta a esse circuito, tem-se, no *carry* de saída, o nível 1 para quando as duas entradas da primeira porta XOR forem iguais a 1 ou para quando as duas entradas na segunda porta forem 1. Assim, o *carry* de saída do somador-completo é dado pela operação AND com as entradas da primeira porta e pela operação AND com o *carry* de entrada. Os dois termos gerados passam pela operação OR. Observe a expressão a seguir:

**Meio-subtrator <T3>**

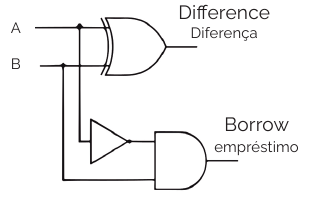
O meio-subtrator, ou HS (*Half-Subtrator*), consiste em um tipo de circuito lógico combinacional que executa operação de subtração entre dois algarismos binários. Caracteriza-se por apresentar duas entradas e duas saídas. As saídas são formadas por um componente que representa a diferença: Di; e por um componente de empréstimo, *borrow*: Bo. O circuito meio-subtrator executa a operação de subtração com as suas duas entradas, os dois bits apenas.



<Dia: recriar>

<Leg: Meio-subtrator.>

Perceba que o *borrow* saída apresentará nível 1 quando somente A apresentar valor 0 e B apresentar valor 1. O *borrow* é dado por uma operação AND entre as entradas. A saída da subtração, a diferença, só apresenta nível 1 quando as variáveis de entrada são diferentes. Dessa forma, pode ser expressa por uma operação XOR.



< Dia: extrair imagem do PDF. p. 101>

<Leg: Diagrama lógico do meio-subtrator.>

**Subtrator-completo <T3>**

Um subtrator completo, ou FS (*full-subtractor*), é um circuito combinacional que realiza subtração envolvendo três bits, ou seja, A (minuendo), B (subtraendo) e Bin (emprestado). Esse circuito aceita três entradas: A (minuendo), B (subtraendo) e um Bin (bit emprestado). Além disso, produz duas saídas: D (diferença) e Bout (retirada).



<Dia: recriar>

<Leg: Subtrator-completo.>

<COTAS: Entradas Saídas

A B Bin Di Bo

0 0 0 0 0

0 1 1 1 1

0 0 0 1 1

0 1 1 0 1

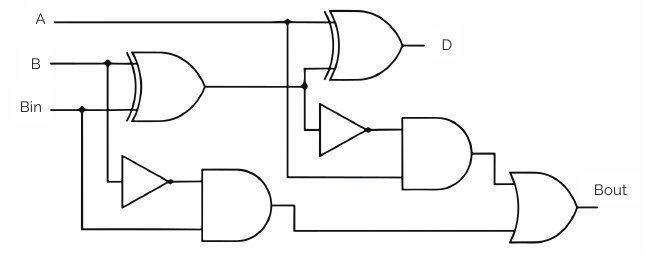
1 0 0 1 0

1 1 1 0 0

1 0 0 0 0

1 1 1 1 1>

No circuito subtrator-completo, dois bits são submetidos a uma operação XOR entre eles na entrada e com o *borrow* de entrada. Para o *borrow* de saída, implementam-se operações AND entre A e B, entre A e Bin e entre B e Bin e, em seguida, efetuam-se uma operação OR e uma operação XOR.



< Dia: extrair imagem do PDF. p. 103 >

<Leg: Circuito lógico completo para um subtrator-completo.>

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Os circuitos aritméticos compreendem circuitos capazes de realizar sete operações aritméticas diferentes, fazendo uso de apenas um único circuito composto. Os circuitos somadores são amplamente utilizados em computadores e calculadoras, tendo em vista que executam operações de adição com dois números binários por vez, sendo que cada número binário pode apresentar vários dígitos. O meio-somador, ou HA (Half-Adder), consiste em um tipo de circuito lógico combinacional formado pelas portas XOR e AND. Um somador-completo tem um *carry* de entrada, enquanto que um meio-somador não tem. O meio-subtrator, ou HS (Half-Subtrator), consiste em um tipo de circuito lógico combinacional que executa operação de subtração entre dois algarismos binários. Um subtrator completo, ou FS (full-subtractor), é um circuito combinacional que realiza subtração envolvendo três bits, ou seja, A (minuendo), B (subtraendo) e Bin (emprestado).

# **UNIDADE 02 - Capítulo 04**

# **Fundamentos dos Biestáveis Lógicos**

**Circuitos sequenciais <T2>**

Os circuitos sequenciais consistem em circuitos lógicos combinacionais compostos de variáveis de entrada (X), portas lógicas e variável de saída (Z). O circuito combinacional produz uma saída com base apenas na variável de entrada. Já no caso de um circuito sequencial, é produzida uma saída com base na entrada atual inserida no circuito e nas variáveis de entrada anteriores.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

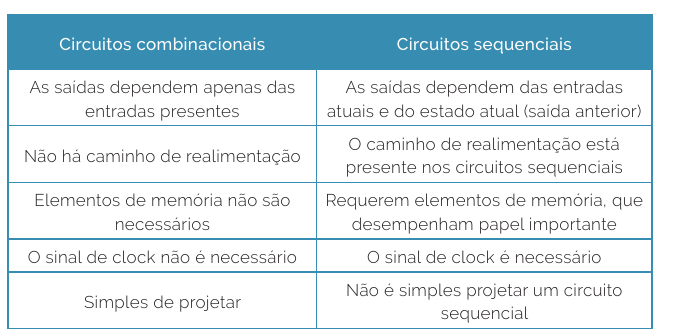
O circuito combinacional não usa nenhuma memória. Assim, o estado anterior de entrada não tem nenhum efeito sobre o estado atual do circuito.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Nesse modelo de circuito, as entradas internas são uma função de estados de saída anteriores. As entradas secundárias se comportam como variáveis de estado produzidas pelos elementos de armazenamento, enquanto as saídas secundárias são excitações para os elementos de armazenamento. Um circuito sequencial não precisa sempre conter um circuito combinacional. Assim, o circuito sequencial pode conter apenas o elemento de memória.

• Circuito sequencial assíncrono: Não fazem uso de um sinal de *clock*. Porém, usam os pulsos das entradas. Compreendem circuitos mais rápidos que os sequenciais síncronos, tendo em vista a ocorrência do pulso de *clock* e que eles modificam seu estado imediatamente com a mudança do sinal de entrada. Esses circuitos são empregados para operações em que a velocidade de operação é importante e independente do pulso de *clock* interno.

• Circuito sequencial síncrono: Os circuitos síncronos, por sua vez, utilizam o sinal de clock e entradas de nível. O pulso de saída possui o mesmo tempo de duração do pulso de *clock* para os circuitos sequenciais com *clock*. Eles aguardam a chegada do próximo pulso de *clock* para realizar a próxima operação. Desse modo, esses circuitos são um pouco mais lentos em comparação aos assíncronos.



<Dia: recriar>

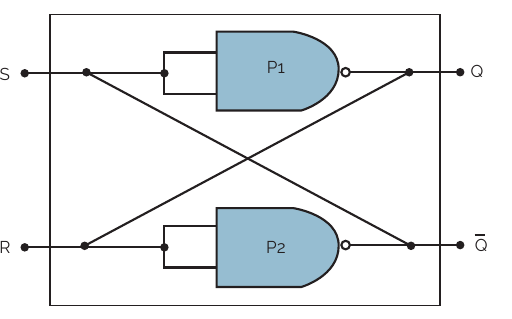
<LEG: Diferenças entre os circuitos combinacionais e circuitos sequenciais.>

<COTAS: Circuitos combinacionais As saídas dependem apenas das entradas presentes Não há caminho de realimentação Elementos de memória não são necessários Circuitos sequenciais As saídas dependem das entradas atuais e do estado atual (saída anterior) O caminho de realimentação está presente nos circuitos sequenciais Requerem elementos de memória, que desempenham papel importante O sinal de clock não é necessário Simples de projetar O sinal de clock é necessário Não é simples projetar um circuito sequencial>

**Biestáveis lógicos <T2>**

Os circuitos sequenciais se caracterizam por apresentarem saídas com um sinal, para dado instante, influenciado por eventos que ocorreram anteriormente. Esse fundamento pressupõe que o circuito deve apresentar a propriedade de memória, ou seja, deve ser capaz de “lembrar” o seu estado passado, para que possa modificar o seu passo à medida que recebe novos comandos.

Um circuito biestável possui dois estados estáveis e pode ser usado para armazenar informações de estado. Ele pode armazenar 1 bit de informação binária. O circuito pode ser feito para mudar de estado por sinais aplicados a uma ou mais entradas de controle e terá uma ou duas saídas. Dispositivos biestáveis simples podem ser construídos usando nada mais do que portas NAND ou NOR cruzadas.



< Dia: extrair imagem do PDF. P. 109>

<LEG: Biestáveis lógicos.>

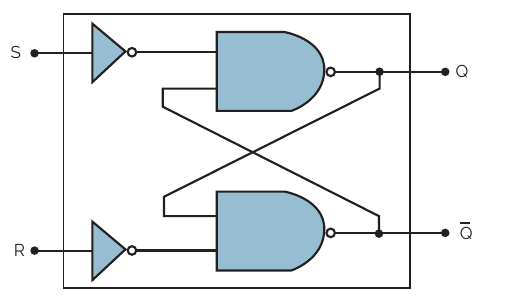
O *latch* é um tipo de dispositivo de armazenamento temporário que tem dois estados estáveis (biestável) e é normalmente colocado numa categoria separada dos *flip-flops*. Os *latches* são similares aos *flip-flops* porque eles são dispositivos biestáveis que podem permanecer em um dos dois estados estáveis usando uma configuração de realimentação, na qual as saídas são conectadas de volta às entradas opostas. A principal diferença entre os *latches* e os *flip-flops* é o método usado para a mudança de estado deles. (FLOYD, 2007, p. 388)

O termo *latch* é associado aos biestáveis que não têm terminais de *clock.*

Os *flip-flops* são dispositivos biestáveis síncronos, também conhecidos como multivibradores biestáveis. Nesse caso, o termo síncrono significa que a saída muda de estado apenas no momento especificado pela entrada de disparo denominada de clock (CLK), a qual é indicada como uma entrada de controle (C); ou seja, as mudanças na saída ocorrem em sincronismo com o *clock*. (FLOYD, 2007, p. 394)

**Tipo *Reset-Set* (RS) <T3>**

O circuito *flip-flop* apresenta a limitação de ter valor máximo de entrada que a porta lógica é capaz de suportar sem entrar em colapso por sobrecarga de corrente. Uma resolução para essa limitação foi encontrada no uso das portas lógicas não como inversores, mas como portas NAND.



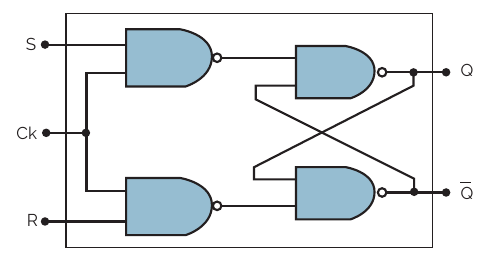
< Dia: extrair imagem do PDF. P. 110>

<LEG: Flip-flop RS básico.>

É possível descrever 0 ou 1 nas células de memória pelas entradas R e S sem que problemas *fan-out* ocorram.

**Síncrono <T3>**

Existem situações em que o circuito sequencial demanda a sincronização dos biestáveis por meio de pulsos de comando ou pulsos de relógio. Por exemplo, um circuito *flip-flop* pode ser convertido em um biestável do tipo síncrono. Esse processo é feito pela substituição dos inversores por portas do tipo NAND.

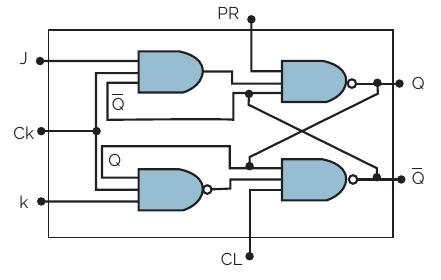


<Dia: extrair imagem do PDF. p. 111 >

<LEG: *Flip-flop* RS com entrada *clock.*>

Esse circuito exibe o problema de ambiguidade para entradas em estado 1 ao mesmo tempo, tendo em vista que resulta em saída indeterminada.

O *flip-flop* JK acaba com a indeterminação dos biestáveis RS. O *flip-flop* T age como interruptor do tipo “pera”, por meio do qual o estado é alterado a cada pulso de relógio. O *flip-flop* D forma uma unidade de retardo, em que a entrada é copiada para a saída, mas é retardada pelo tempo do relógio de pulso

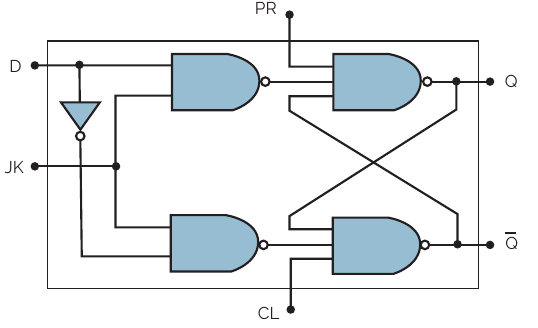


< Dia: extrair imagem do PDF. p. 113 >

<LEG: *Flip-flop* JK.>

No circuito do JK, PR e CL descrevem as funções *preset* e *clear*, as quais possibilitam o ajuste das condições de funcionamento inicial, pois possuem prioridade sobre as outras entradas. Caso PR e CL sejam ativadas simultaneamente, a saída será indeterminada.

Para o flip-flop tipo D, o circuito é obtido alterando-se o flip-flop RS síncrono ou o JK, por meio da introdução de um inversor interligando às entradas JK ou RS. Esse procedimento elimina o problema da ambiguidade para S=R ou J=K.



< Dia: extrair imagem do PDF. p. 114>

<Leg: *Flip-flop* D.>

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Os circuitos sequenciais consistem em circuitos lógicos combinacionais compostos de variáveis de entrada (X), portas lógicas e variável de saída (Z). O circuito combinacional não usa nenhuma memória. Nesse caso, o estado anterior de entrada não tem nenhum efeito sobre o estado atual do circuito. Em eletrônica digital, entende-se que dispositivos capazes de realizar operação correspondente à chave de comutação são conhecidos como biestáveis lógicos, ou *flip-flops*. Os *flip-flops* são dispositivos biestáveis síncronos, também conhecidos como multivibradores biestáveis. Nesse caso, o termo “síncrono” significa que a saída muda de estado apenas no momento especificado pela entrada de disparo denominada de *clock* (CLK), a qual é indicada como uma entrada de controle (C); ou seja, as mudanças na saída ocorrem em sincronismo com o *clock*.

# **UNIDADE 03 - Objetivos**

Olá. Seja muito bem-vindo à Unidade 3. Nosso objetivo é auxiliá-lo no desenvolvimento das seguintes competências profissionais até o término desta etapa de estudos:

1. Classificar e aplicar os tipos de contadores: assíncronos, síncronos, anel, cascata etc., em circuitos digitais.

2. Utilizar registradores de deslocamento em circuitos digitais.

3. Entender a tecnologia da multiplexação, aplicando multiplexadores em circuitos digitais.

4. Aplicar dispositivos conversores A/D e D/A em circuitos digitais.

# **UNIDADE 03 - Capítulo 01**

# **Dispositivos Contadores em Eletrônica Digital**

**Contadores <T2>**

Na eletrônica digital, os contadores consistem em dispositivos que armazenam e exibem a quantidade de vezes que certo evento ou processo se desenvolve com base em dados temporais acompanhados por um relógio.

Os contadores são circuitos sequenciais e baseiam-se na contagem de pulsos. Assim, pode-se identificar um contador como uma versão mais ampla dos *flip-flops* (FF), pois possui aplicação de sinais de *clock*. Mas é importante estar atento ao fato de que existem diversos tipos de contadores.

Os circuitos sequenciais apresentam, para dado instante, saídas com um sinal que sofre influência de eventos que aconteceram antes. Nesse caso, esses circuitos possuem a propriedade de memória, sendo capazes de “lembrar” um estado ocorrido, mas modificando o seu passo à medida que novos comandos são recebidos. Assim, a memória é o elemento central desse tipo de circuito (IDOETA; CAPUANO, 1993).

Em geral, a função de um contador consiste em incrementar ou decrementar um valor inicial. O valor do contador representa o número de pulsos de *clock* recebidos na entrada de *clock*. Quando os pulsos de clock são contados de forma crescente, é chamado de contador. No contador descendente, o valor da contagem é decrementado em um na chegada de cada pulso de *clock*.

Os contadores podem ser classificados com base na maneira como recebem os pulsos de *clock*, ou seja, se o pino com *clock* é comum ou não aos demais *flip-flops* presentes no sistema, assim as categorias podem ser: contadores síncronos e contadores assíncronos. Os contadores assíncronos, ou contadores ondulantes (*ripple counters*), são caracterizados pelo primeiro *flip-flop* receber o *clock* por um pulso de *clock* externo e por cada *flip-flop* posterior receber o *clock* por meio das saídas do *flip-flop* que o anteceder. No caso dos contadores síncronos, a entrada de *clock* é conectada a todos os *flip-flops,* assim todos recebem o *clock* ao mesmo tempo. Assim, pode-se identificar, nessas categorias, contadores com base no tipo de sequência, o número de estados ou o número de *flip-flops*.

**Contadores assíncronos <T3>**

Assíncrono é uma denominação para eventos estabelecidos em uma relação temporal fixada entre si, ocorrendo, assim, em tempos diferentes. Como já definido, os contadores assíncronos são compostos de uma série de *flip-flops*, nos quais a saída de cada um é conectada à entrada de *clock* do próximo *flip-flop* de ordem superior. Os *flip-flops* no contador assíncrono são acionados individualmente, ou seja, não são sincronizados.

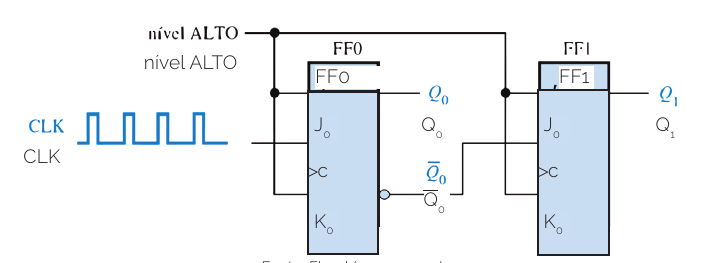
## INÍCIO DA SEÇÃO **<IMPORTANTE>**

Nos contadores assíncronos, a entrada de *clock* sempre é conectada no *flip-flop* menos significativo.

## FIM DA SEÇÃO **<IMPORTANTE>**

Os contadores assíncronos normalmente são de fácil projeção, são constituídos por *flip-flops* que não alteram seu estado ao mesmo tempo, tendo em vista que não possuem o mesmo pulso de *clock*. No caso, o pulso de *clock* é dado como entrada somente no primeiro FF, o qual se caracteriza como menos significativo. Para as outras entradas de *clock*, nos demais blocos, a alimentação se dá pelo FF que as antecede. A grande limitação dos contadores assíncronos consiste na velocidade de operação.

**Contador de 2 bits <T3>**



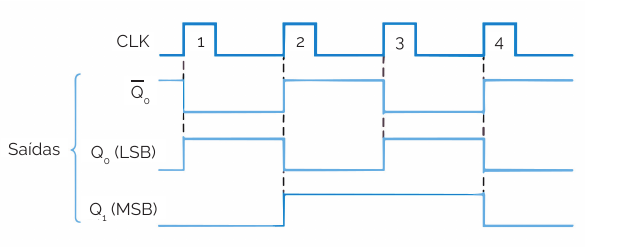
< Dia: extrair imagem do PDF. pg. 123>

<Leg: Contador assíncrono de 2 bits.>

Analisando o contador assíncrono de 2 bits, pode-se observar que o sistema é composto de dois *flip-flop*, além disso, um *clock*, identificado pela sigla CLK, é aplicado na entrada *clock* C apresentando para o primeiro *flip-flop*, identificado como FF0, o qual é o bit menos significativo do sistema.

Um atraso de propagação acontece de forma inerente para um *flip-flop*, isso quer dizer que a transição do pulso de *clock* de entrada e a transição da saída Q0 do FF0 não ocorrem, nunca, ao mesmo tempo, ou seja, FF0 e FF1 não são disparados ao mesmo tempo. Esse fator caracteriza o contador como assíncrono.

No diagrama de temporização, são apresentadas as formas de onda para as saídas dos dois flip-flops, FF0 e FF1, à medida que pulsos de clock são atribuídos ao sistema do contador. Nesse caso, as transições das saídas Q0 e Q1 são apresentadas como fenômenos simultâneos, mas isso foi implementado a título de simplificar a explicação do processo, dessa maneira, é importante estar atento ao fato de que, na realidade, há um breve atraso entre CLK e Q0 e entre as saídas Q0 e Q1 .



< Dia: extrair imagem do PDF. p. 124>

<Leg: Diagrama de temporização para contador assíncrono de 2 bits.>

**Atrasos de propagação <T3>**

Os atrasos de propagação caracterizam os contadores assíncronos, tendo em vista que o efeito do pulso de entrada só é sentido pelo *flip-flop* menos significativo, não atingindo o próximo FF. Da mesma forma, ocorre entre os demais FF. Por conta dos atrasos na propagação, o pulso de clock aplicado na entrada do *flip-flop* sofre o efeito de ondulação por meio do contador até que alcance o último FF.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

De acordo com Floyd (2007), o atraso cumulativo máximo num contador tem que ser menor que o período da forma de onda do *clock*.

## FIM DA SEÇÃO **<IMPORTANTE>**

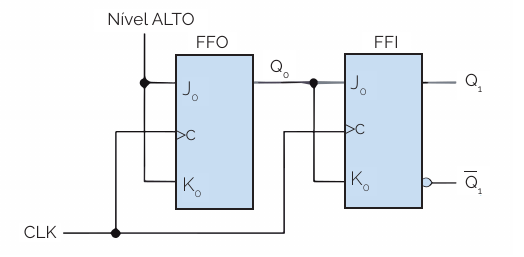
Tais atrasos ocorrem de forma cumulativa no contador assíncrono e esse fato caracteriza a principal desvantagem desse dispositivo para diversas aplicações, tendo em vista que limita a taxa, na qual o contador é capaz de receber pulsos de *clock*, acarretando problemas de decodificação.

**Contadores síncronos <T3>**

Os contadores síncronos possuem um relacionamento de tempo fixo entre os eventos que ocorrem entre os *flip-flops*, ou seja, nesse caso os pulsos de *clock* ocorrem ao mesmo tempo para todos os *flip-flops*, tendo em vista que o *clock* é conectado a todos os *flip-flops* por meio de uma mesma linha.

Assim, os contadores síncronos eliminam os atrasos visualizados nos contadores assíncronos, porque os *flip-flops* são controlados pelo mesmo pulso de *clock*.

**Contador binário síncrono de 2 bits <T3>**



< Dia: extrair imagem do PDF. P. 127>

<LEG: Contador síncrono de 2 bits.>

Analisando o contador síncrono de 2 bits, é possível visualizar que é composto de dois *flip-flops.* O pulso de *clock*, CLK, é conectado diretamente aos dois *flip-flops* por meio de uma linha comum. O desenvolvimento operacional de um contador síncrono de 2 bits pode se iniciar no estado binário zero, com os dois FF resetados. À medida que ocorre o primeiro pulso de *clock*, há a subida da borda positiva ocorrendo a comutação de FF0 e o Q0 mude de estado passando para nível alto.

No caso do outro *flip-flop*, FF1, as suas entradas estão também inicialmente em nível baixo, tendo em vista que Q0 é a saída, as entradas J1 e K1 estão conectadas em nível baixo, pois ainda não atingiu nível alto (devido ao atraso de propagação da borda de disparo do pulso de *clock*). Assim, quando é aplicada a borda de subida do primeiro pulso de clock J = 0 e K = 0. Esse fenômeno consiste em uma condição de repouso, dessa maneira FF1 não altera de estado.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Os contadores consistem em dispositivos que armazenam, e por vezes exibem, a quantidade de vezes que certo evento ou processo se desenvolve com base em dados temporais acompanhados por um relógio. O valor do contador representa o número de pulsos de *clock* recebidos na entrada de *clock*. Quando os pulsos de *clock* são contados de forma crescente, é chamado de contador. No contador descendente, o valor da contagem é decrementado na chegada de cada pulso de *clock*. Os contadores podem ser classificados com base na maneira como recebem os pulsos de *clock*, ou seja, se o pino com *clock* é comum ou não aos demais *flip-flops* presentes no sistema, assim as categorias podem ser: contadores síncronos e contadores assíncronos. Assíncrono é uma denominação para eventos que não estabelecem uma relação temporal fixada entre si, ocorrendo, assim, em tempos diferentes. Os contadores síncronos possuem um relacionamento de tempo fixo entre os eventos ocorridos entre os *flip-flops*, ou seja, nesse caso, os pulsos de *clock* ocorrem ao mesmo tempo para todos os *flip-flops*, tendo em vista que o *clock* é conectado a todos os *flip-flops* por meio de uma mesma linha.

# **UNIDADE 03 - Capítulo 02**

# **Dispositivos Registradores em Eletrônica Digital**

**Registradores de deslocamento <T2>**

Registradores, assim como contadores digitais, são dispositivos construídos a partir de flip-flops e atuam, principalmente, nos processos de armazenamento de dados digitais, nesse caso, porém, não há sequenciamento de estados internos como uma propriedade (com exceção de aplicações bem específicas). Assim, os registradores são dispositivos que fazem uso de *flip-flops*, circuitos capazes de armazenar ou registrar bits, com o principal objetivo de poder guardar informações binárias.

Os registradores de deslocamento (RD), ou *shift register*, consistem em dispositivos síncronos em que os dados podem apresentar entradas seriais ou paralelas e permanecem armazenados até sua saída, também serial ou paralela. O fluxo serial dos dados de um registrador é geralmente de deslocamento (*shifting*), daí essa denominação, pois os dados ficam deslocamento para a direita ou para a esquerda. Em geral, os registradores de deslocamento apresentam o deslocamento de seus dados de 1 bit para a direita ou para a esquerda com base na aplicação do pulso de *clock*.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

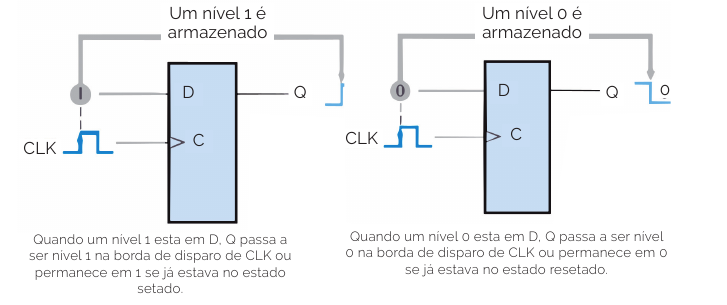
É importante estar atento ao fato de que, caso os dados seriais sejam realimentados para a entrada serial do mesmo registrador, tem-se que a tal operação é dado o nome de rotação de dados. No caso da entrada paralela de dados, é carga de registrador.

## FIM DA SEÇÃO **<IMPORTANTE>**

Em resumo, um registrador consiste em um circuito digital que apresenta duas funcionalidades básicas, são elas:

• Armazenamento de dados.

• Movimentação de dados.



< Dia: extrair imagem do PDF. PG. 132>

<LEG: *Flip-flop* como armazenador.>

Os *flip-flops* tipo D descrevem o conceito de armazenamento do nível, tanto para 1 como para 0. Perceba que, no primeiro *flip-flop* (à esquerda), o nível 1 é aplicado na entrada de dados e, à medida que ocorre o pulso de *clock*, esse nível é armazenado por meio do set do *flip-flop*. Assim, na remoção do nível, o *flip-flop* fica “setado” e o nível fica armazenado.

Da mesma maneira ocorre para o nível 0, porém, nesse caso, é armazenado por meio do *reset* do sistema. É possível indicar um registrador pela quantidade de bit que pode reter ou armazenar, isso é caracterizado como a capacidade de armazenamento do registrador. Cada um dos *flip-flops*, ou estágios, compreende um registrador de deslocamento e cada um representa um bit de capacidade de armazenamento, assim a quantidade de estágios ou FF indica sua capacidade de armazenamento.

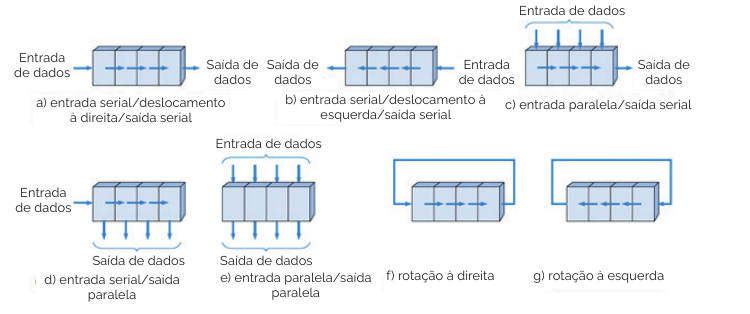
Além disso, uma outra propriedade se relaciona à capacidade de deslocamento de um registrador. Essa capacidade possibilita a movimentação dos dados de um estágio para outro em um registrador ou para fora dele por meio de aplicações de pulsos de *clock*. Neste sentido, é possível categorizar os registradores de deslocamento em:

• RD com entrada série e saída série (ES/EE).

• RD com entrada série e saída paralela (ES/EP).

• RD com entrada paralela e saída série (EP/EE).

• RD universais.



< Dia: extrair imagem do PDF. PG. 133>

<LEG: Movimentos básicos de dados em registradores de deslocamento (são usados quatro bits como ilustração. Os bits se movem na direção das setas)>

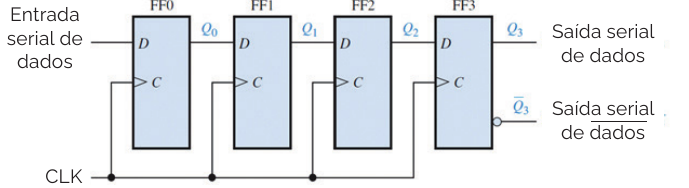
Esses tipos de registradores consistem em circuitos integrados padronizados com o passar dos anos. Assim, o fluxo de dados é caracterizado por meio de chips especificados, que podem ser organizados em diversos arranjos dando forma a registradores de tamanhos diferentes.

**Registradores de deslocamento com entrada e saída seriais**  **<T3>**

Os registradores de deslocamento que possuem entrada e saída do tipo serial são aqueles que admitem dados em série, ou seja, dados apresentando um bit por vez em uma linha de dados apenas. Nesse caso, é gerada uma saída de informação composta de dados armazenados em forma serial. Nesse caso, os dados entram por um único *flip-flop*, são deslocados e disponibilizados por meio de uma única saída.

De acordo com Tocci, Widmer e Moss (2017):

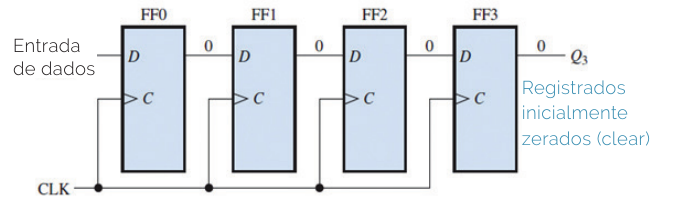
Um registrador de deslocamento de entrada serial/saída serial terá de ser carregado um bit por vez a cada pulso de *clock* ao longo do conjunto de flip-flops até a outra extremidade do registrador. Com *clock* continuo, os dados sairão do registrador um bit de cada vez na mesma ordem em que foram carregados. (TOCCI; WIDMER; MOSS, 2017, p. 380)



< Dia: extrair imagem do PDF. P. 134>

< LEG: RD com entrada e saída seriais.>

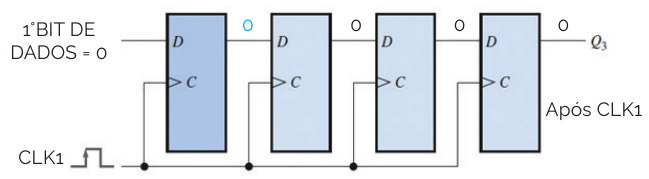
Perceba como se dá a disposição dos *flip-flops* para registrador de deslocamento com entrada e saída seriais. Nesse caso, é exibido um RD com 4 bits, daí o uso de 4 estágios do tipo D. Com quatro flip-flops, o registrador exibe a capacidade de armazenar até 4 bits de dados. Para ilustrar o comportamento do RD com entrada e saída seriais, imagine a aplicação da entrada de quatro bits: 1010 no registrador, sendo iniciado o processo pelo bit mais à direita, ou seja 0.



< Dia: extrair imagem do PDF. P. 135>

<LEG: RD com entrada e saída seriais em estado zerado.>

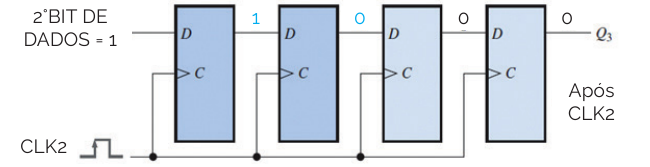
O estado inicial do registrador é o estado zerados. Observe agora o sistema com a aplicação da entrada, iniciando pelo primeiro bit, o nível 0, na linha de entrada, o qual faz com que D seja igual a 0 para FF0. Nesse caso, quando o primeiro pulso de *clock* é aplicado, é feito o *reset* em FF0 e o nível 0 é armazenado.



< Dia: extrair imagem do PDF. P. 135>

<LEG: RD com entrada e saída seriais para 1° bit de dados.>

Em seguida, é inserido o segundo bit de nível 1 na entrada de dados. Isso faz com que D seja agora igual a 1 para FF0 e D seja igual a 0 para FF1, tendo em vista que a entrada de FF1 está conectada à saída de FF0. Dessa maneira, à medida que ocorre o segundo pulso de *clock*, os dados são deslocados, FF0 é setado e o nível, que antes estava em FF0, vai para FF1.



< Dia: extrair imagem do PDF. P. 136>

<LEG: RD com entrada e saída seriais para 2° bit de dados.>

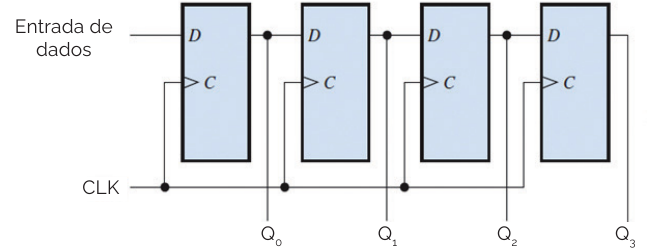
Da mesma maneira acontece para o terceiro e o quarto bit. No caso do último nível, a entrada serial é finalizada no registrador e os dados são armazenados por certo tempo até que os *flip-flops* sejam realimentados.

**Registradores de deslocamento com entrada serial e saída paralela <T3>**

Registradores de Deslocamento com Entrada Serial e Saída Paralela No caso de um RD com entrada serial e saída paralela, a entrada dos bits no registrador é dada em série, iniciando pelo bit mais à direita, mas a saída é dada em paralelo.

De acordo com Haupt e Dachi (2018):

No registrador de deslocamento com entrada série e saída paralela (ES/SP), os dados são introduzidos também por um únicos flip-flop, são deslocados, mas estarão disponíveis na saída de cada flip-flop. Assim, todas as saídas desse registrador estarão disponíveis simultaneamente. (HAUPT; DACHI, 2018, p. 121)



< Dia: extrair imagem do PDF. P. 137>

<LEG: RD com entrada seriais e saída paralela.>

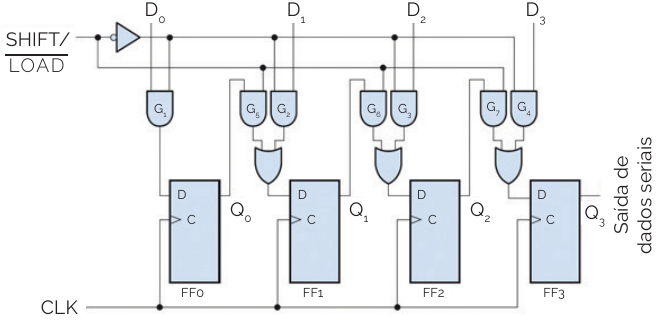
A saída de cada *flip-flop* fica disponível, assim, uma vez armazenada a informação, cada bit se apresenta em uma linha própria de saída e todos os bits são oferecidos de forma simultânea.

**Registradores de deslocamento com entrada paralela e saída serial <T3>**

No caso de um RD com entrada paralela e saída serial, a entrada dos bits no registrador é dada em paralelo, em que os bits são inseridos simultaneamente em cada um dos *flip-flops* em linhas paralelas ao invés de bit a bit, como no caso da entrada em série, e a saída é dada em série, uma vez que os dados estão armazenados no registrador.

De acordo com Haupt e Dachi (2018):

Em um registrador do tipo com entrada paralela e saída em série (EP/SS), todos os *flip-flops* são carregados simultaneamente, isto é, os dados são introduzidos ao mesmo tempo em todos os *flip-flops*. A saída, no entanto, se dará por meio de um único *flip-flop*. Internamente, o carreamento é efetuado por meio de entradas *preset* e *clear* de cada *flip-flop*. (HAUPT; DACHI, 2018, p. 123)



< Dia: extrair imagem do PDF. P. 138>

<LEG: Diagrama lógico de RD com entrada paralela e saída serial.>

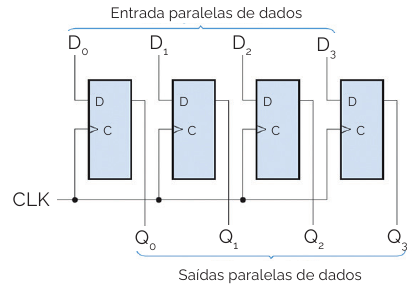
De acordo com Haupt e Dachi (2018), a entrada SHIFT/LOAD controla o modo de operação do registrador, assim, se a sua entrada estiver em estado 1, estará habilitada a entrada paralela de dados. Assim, a entrada SHIFT/LOAD permite a carga dos quatro bits carregados no exemplo da figura anterior. Se, porém, SHIFT/LOAD for de estado 0, baixo, as portas G1 e G4 estarão habilitadas, possibilitando que cada bit seja aplicado à entrada D do respectivo *flip-flop*.

No caso, à medida que o pulso de *clock* ocorrer, o set será aplicado aos FF com D=1 e o *reset* será aplicado aos FF com D=0, armazenando os 4 bits ao mesmo tempo. Se SHIFT/LOAD for de estado 1, alto, as portas G1 e G4 estarão desabilitadas, e as portas G5 a G7 habilitadas, possibilitando o deslocamento dos bits à direita para um próximo estágio.

**Registrador de deslocamento com entrada paralela e saída paralela <T3>**

De acordo com Tocci, Widmer e Moss (2017):

Um grupo de flip-flops que armazenam múltiplos bits ao mesmo tempo e no quais todos os bits do valor binários armazenado estão diretamente disponíveis é conhecido como registrador de deslocamento de entrada paralela/ saída serial. (TOCCI; WIDMER; MOSS, 2017, p. 379)



< Dia: extrair imagem do PDF. p. 139>

<Leg: RD com entrada paralela e saída paralela>

O RD com entrada e saída paralelas é caracterizado pela entrada de todos os bits simultaneamente e imediatamente após o aparecimento de todos os bits nas saídas.

**Registradores universais <T3>**

No registrador universal, é possível escolher os tipos de entradas e saídas assumidas durante o uso, ou seja, se serão desenvolvidas em série ou em paralelo.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Você deve ter aprendido que os registradores são dispositivos usados em flip-flops, circuitos que têm a capacidade de armazenamento ou registro de bits, com o principal objetivo de poder armazenar informações binárias. Os registradores de deslocamento apresentam o deslocamento de seus dados de 1 bit para a direita ou para a esquerda, com base na aplicação do pulso de *clock*. Em resumo, um registrador consiste em um circuito digital que apresenta duas funcionalidades básicas, são elas: armazenamento de dados e movimentação de dados. Além disso, há uma outra propriedade relacionada à capacidade de deslocamento de um registrador. Essa capacidade possibilita a movimentação dos dados de um estágio para outro em um registrador ou para fora dele por meio de aplicações de pulsos de clock. Nesse sentido, é possível categorizar os registradores de deslocamento em: RD com entrada série e saída série (ES/EE), RD com entrada série e saída paralela (ES/EP), RD com entrada paralela e saída série (EP/EE), RD universais.

# **UNIDADE 03 - Capítulo 03**

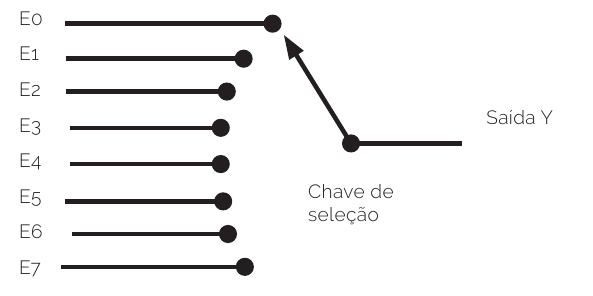
# **Dispositivos Multiplexadores Eletrônica Digital**

**Multiplexadores <T2>**

Você já deve ter observado que sistemas de som disponibilizam a opção de o usuário selecionar a forma como irá ouvir algum tipo de áudio, por exemplo: pela entrada MP3, MP4, pelo sintonizador de televisão, pelo sintonizador de rádio ou pelo sistema de áudio DVD.

Para a escolha do modo de uso do som, o sistema apresenta um chaveamento que possibilita tal operação. Por meio da chave, pode-se escolher um dos tipos de sinal eletrônico enviado para o amplificador de potência e para os alto-falantes do sistema.

Perceba que o cenário descrito anteriormente representa o dispositivo da eletrônica digital conhecido como multiplexador (MUX). Em síntese, define-se um multiplexador como um dispositivo por meio do qual é possível escolher entre uma diversidade de sinais e transferir tais informações para uma dada saída.



< Dia: extrair imagem do PDF. P. 142>

<LEG: MUX.>

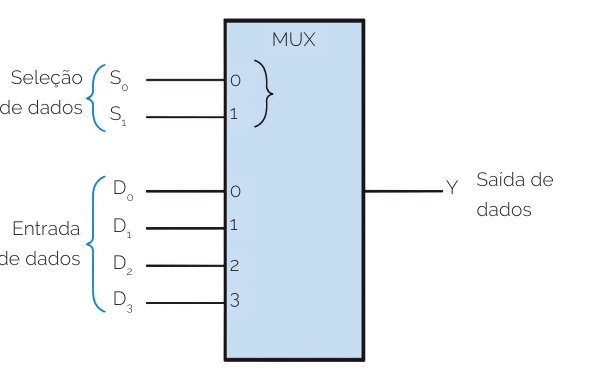
Define-se um multiplexador atua como um dispositivo seletor de dados, ou seja, é um circuito lógico receptor de variados tipos de sinais de entradas digitais, e que seleciona um deles, em certo instante de tempo, para que seja transferido para uma saída.

Para diversos tipos de aplicações, um subsistema digital precisa receber os sinais provenientes das diversas fontes, daí a necessidade do uso do sistema de comutação, ou seja, de chaveamento, que seleciona, em dado instante, qualquer uma das fontes de sinais.

De acordo com Floyd (2007):

Um multiplexador (MUX) é um dispositivo que permite que informações digitais de diversas fontes sejam encaminhadas para uma única linha para serem transmitidas nessa linha para um destino comum. Um multiplexador básico tem várias linhas de entrada de dados e uma única linha de saída. Ele também possui entradas de seleção de dados, as quais permitem que os dados digitais de quaisquer entradas sejam comutados para a linha de saída. Os multiplexadores também são conhecidos como seletores de dados. (FLOYD, 2007, p. 347)

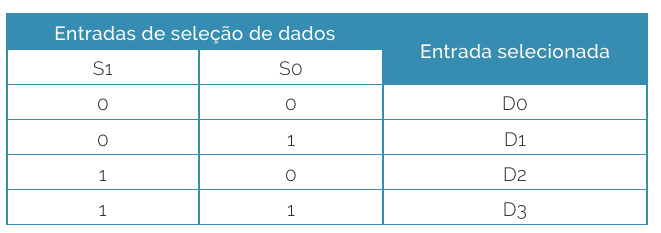
Para um MUX de 4 bits os dados passam das variadas linhas na entrada para uma única linha na saída. No caso, há duas linhas de seleção, tendo em vista que com dois bits de seleção, qualquer uma das quatro entradas pode ser escolhida.



< Dia: extrair imagem do PDF. p. 144>

<Leg: Símbolo lógico de um MUX de 4 bits.>

Assim, um binário 0 é aplicado nas linhas de seleção, ficando S1=0 e S0=0, o dado de entrada D0 irá aparecer na saída de dados. Caso um binário 1 seja aplicado nas linhas de seleção, ficando S1=0 e S0=1, o dado de entrada D1 irá aparecer na saída de dados. Caso um binário 2 seja aplicado nas linhas de seleção, ficando S1=1 e S0=0, o dado de entrada D2 irá aparecer na saída de dados. Caso um binário 3 seja aplicado nas linhas de seleção, ficando S1=1 e S0=1, o dado de entrada D3 irá aparecer na saída de dados.



<DIA: recriar>

<Leg: Seleção de dados no MUX de 2 bits.>

<COTAS: Entradas de seleção de dados

S1 S0 Entrada selecionada

0 0 D0

0 1 D1

1 0 D2

1 1 D3

>

Imaginando essa operação em termos de circuitos lógicos, para a realização da operação de multiplexação, a saída de dados é o estado da entrada que foi escolhida. Com base nessa ideia é possível representar a operação para a saída em termos de entradas de dados e entradas de seleção:

• A saída será D0 se S1 = 0 e S0=0 → D0 .

• A saída será D1 se S1 = 0 e S0 = 1 → D1 S0 .

• A saída será D2 se S1 = 1 e S0 = 0 → D2 S1 .

• A saída será D3 se S1 = 1 e S0 = 1 → D3 S1 S0 .

Assim, implementando-se uma operação OR com tais saídas, tem-se a seguinte expressão:

Para a execução de tal expressão, são necessárias quatro portas AND com três entradas cada, uma porta OR com quatro entradas e o uso de dois inversores.

**Aplicações de multiplexadores <T3>**

Os circuitos multiplexadores podem ser utilizados em diversos tipos de aplicações de sistemas digitais. Tais aplicações incluem desde usos em processos de seleção de dados, roteamento, sequenciamento de operações, conversões série-paralelo, até geração de formas de ondas e de funções lógicas.

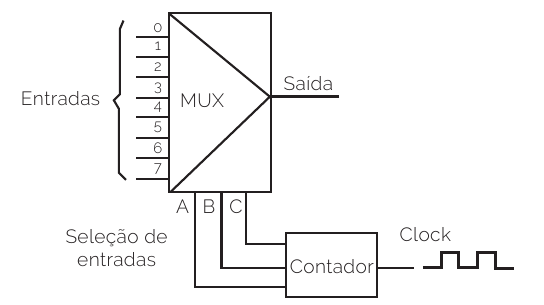
## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

As três principais aplicações dos multiplexadores elencadas são: comutação aleatória de entradas, serialização de sinais digitais, geração de funções booleanas.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

O processo de comutação aleatória de entrada é empregado quando se tem vários sensores e um único indicador. Nesse modelo, cada sensor corresponde a um código binário. As entradas de seleção têm um código aplicado e o indicador exibe a leitura correspondente a tal ponto escolhido.

A multiplexação dos sinais, nesse caso, é dada por meio da seleção sequencial das entradas do multiplexador. É realizada a varredura sequencial de todas as entradas e um contador é adaptado nos terminais de seleção.



< Dia: extrair imagem do PDF. p. 147>

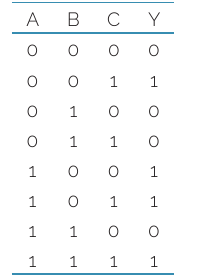
<Leg: Circuito de serialização de um MUX.>

No caso de serialização de um MUX, a saída do contador vai variar de 000 até 111, em que ocorrerão intercalações no decorrer do tempo das informações aplicadas nas entradas.

Assim, é possível transmitir-se oitos distintos sinais por meio de uma única linha de transmissão. A velocidade de varredura pode ser alta o suficiente para simular uma transmissão ocorrida ao mesmo tempo. No caso, a recuperação dos sinais originais demanda o uso de um demultiplexador, o qual deve operar ao mesmo tempo que o multiplexador.

Além dessas, uma outra aplicação para o multiplexador diz respeito à geração de funções booleanas. Nesta situação, o multiplex se torna um bloco funcional universal e, nesse modelo, ele facilita a implementação das funções lógicas, suprimindo a necessidade de realização de simplificações complexas e trabalhosas e minimizando os circuitos.

De acordo com Floyd (2007, p. 353), quando o dispositivo substituir portas discretas, pode frequentemente diminuir bastante o número de CIs e tornar muito fáceis as alterações de projetos. O uso de um MUX de oito entradas, por exemplo, permite a implementação de qualquer tipo de expressão booleana de três variáveis, 23=8.



<Dia: recriar>

<Leg: Tabela-verdade para exemplo de função booleana.>

<COTAS:

A B C Y

0 0 0 0

0 0 1 1

0 1 0 0

0 1 1 0

1 0 0 1

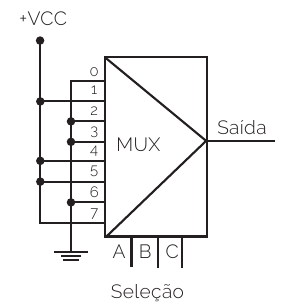
1 0 1 1

1 1 0 0

1 1 1 1>

A partir da tabela-verdade, que Y é nível 1, somente quando as combinações das variáveis de entrada são as seguintes: 001, 011, 101 e 110. Assim, para todas as demais combinações, Y assume o nível 0.

Para que tal função seja implementada com o seletor de dados definido, a entrada de dados selecionada para cada uma das combinações precisa estar conectada ao nível alto (5 V). Todas as outras entradas de dados devem ser conectadas ao nível baixo.



< Dia: extrair imagem do PDF. p. 149>

<Leg: Circuito de um MUX para implementar funções booleanas.>

A implementação de tal função por meio de portas lógicas demandaria uso de quatro portas AND de três entradas, uma porta OR de quatro entradas e três inversores, a menos que essa expressão possa ser simplificada.

O exemplo do circuito de um MUX para implementar funções booleanas representa a forma como um seletor de dados de oito entradas pode ser empregado para se comportar como um gerador de função lógica para três variáveis. É importante, porém, perceber que esse dispositivo também pode ser empregado na geração de função lógica de quatro variáveis por meio do uso de um dos bits (A0) em conjunto com as entradas de dados (FLOYD, 2007).

Como se sabe é necessária uma tabela-verdade de quatro variáveis para 16 combinações das variáveis de entrada. No caso do uso de um seletor de dados de oito bits, cada entrada é selecionada duas vezes: a primeira vez quando A0 é nível 0 e a segunda vez, quando A0 é nível 1.

Com base em tais informações, é possível utilizar as regras (sendo Y a saída e A0 o bit menos significativo):

•Caso Y=0 nas duas vezes em que uma dada entrada for selecionada por uma certa combinação de variáveis de entrada, A3A2A1, conecte essa entrada de dados em GND (0).

• Caso Y=1 nas duas vezes em que uma dada entrada for selecionada por uma certa combinação de variáveis de entrada, A3A2A1, conecte essa entrada de dados em +V (1).

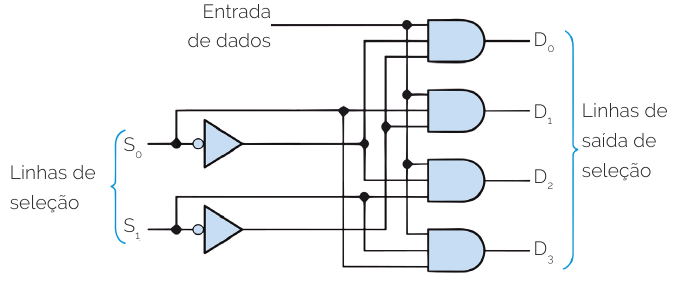
• Caso Y sendo diferente nas duas vezes em que uma dada entrada de dados for selecionada por uma certa combinação de variáveis de entrada, A3A2A1, e se Y=A0, conecte essa entrada de dados em A0.

• Caso Y sendo diferente nas duas vezes em que uma dada entrada de dados for selecionada por uma certa combinação de variáveis de entrada, A3A2A1, e se Y = A0, conecte essa entrada de dados em A0.

**Demultiplexador <T3>**

De acordo com Floyd (2007):

Um demultiplexador (DEMUX) basicamente inverte a função da multiplexação. Ele recebe informações digitais a partir de uma linha e as distribui para um determinado número de linhas de saída. Por essa razão, o demultiplexador também é conhecido como distribuidor de dados. ...os decodificadores também podem ser usados como demultiplexadores. (FLOYD, 2018, p. 356)



< Dia: extrair imagem do PDF. p. 150>

<Leg: Demultiplexador de 1 para 4 linhas.>

Assim, os demultiplexadores são dispositivos realizadores de operações inversas aos multiplexadores. Por vezes, podem ser utilizados em conjunto. O demultiplexador consiste em um circuito combinacional redistribuidor das diversas linhas dos sinais digitais multiplexados. Ele possui as entradas de endereços, uma entrada de sinais, um terminal habilitador e as saídas. Ele realiza a conversão série/paralelo.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Um multiplexador atua como um dispositivo seletor de dados, ou seja, é um circuito lógico receptor de variados tipos de sinais de entrada, digitais, e seleciona um deles, em certo instante de tempo, para que este seja transferido para uma saída. Para diversos tipos de aplicações, um subsistema digital precisa receber os sinais provenientes das diversas fontes, daí a necessidade do uso do sistema de comutação, ou seja, de chaveamento, que seleciona, em dado instante, qualquer uma das fontes de sinais. Os circuitos multiplexadores utilizados em diversos tipos de aplicações de sistemas digitais. Tais aplicações incluem desde usos em processos de seleção de dados, roteamento, sequenciamento de operações, conversões série-paralelo até geração de formas de ondas e de funções lógicas. O demultiplexador consiste em um circuito combinacional redistribuidor das diversas linhas dos sinais digitais multiplexados.

# **UNIDADE 03 - Capítulo 04**

# **Dispositivos Conversores em Eletrônica Digital**

**Quantidade digital *versus* quantidade analógica <T2>**

Os fenômenos ocorridos na natureza, quando tratados em termos de sinais, constituem sistemas analógicos. Os sistemas digitais, por outro lado, operam com sinais digitais.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

Uma quantidade digital apresenta um valor caracterizado entre duas formas possíveis: 0 ou 1, alto ou baixo, verdadeiro ou falso, ligado ou desligado.

## FIM DA SEÇÃO **<IMPORTANTE>**

Em termos práticos, as quantidades digitais, como a tensão, consistem em valores inseridos em faixas e definem-se, de acordo com Haupt e Dachi (2018), em valores inseridos dentro de determinada faixa. Ainda, apresentam o mesmo valor digital.

Para a lógica TTL, por exemplo, tem-se que:

• 0 a 0,8 V = 0 lógico

• 2 a 5 V = 1 lógico

Ou seja, para qualquer valor que esteja inserido na faixa entre 0 a 0,8 V, atribui-se o valor digital 0 e, da mesma forma, para qualquer valor que esteja inserido no intervalo de 2 a 5 V, atribui-se o valor lógico 1.

Dessa forma, os valores exatos para a tensão em circuitos digitais não são tão relevantes, tendo em vista que esses circuitos respondem da mesma forma para todos os valores de tensão inseridos no intervalo.

As quantidades analógicas, por sua vez, caracterizam-se por assumir qualquer valor ao longo de uma faixa contínua de valores e, nesse caso, os valores exatos são relevantes. Cada valor em uma quantidade analógica tem associado a si um valor diferente.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

Na natureza, existem diversas quantidades mensuráveis, por exemplo: temperatura, pressão, volume, velocidade, intensidade, luminosa, som, posição, fluxo, entre outros.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Como a grande maioria das informações coletadas no mundo real estão no formato analógico, para que estas possam ser processadas em dispositivos digitais, é necessário tratamento a fim de que sejam colocadas em formato digital.

As informações coletadas a partir dos sistemas reais, pois, constam em um estado analógico, ou seja, contínuo, necessitando, assim, de tratamento para que possam ser processados em dispositivos digitais; desse modo, a informação é convertida de analógica para digital. Já no caso em que a informação retorne para o formato utilizado nos processos reais, deve-se realizar o processo de conversão inverso, no qual a informação digital é convertida em analógica.

A diferença exibida na interface entre os formatos digital, dos computadores, e o formato analógico, dos processos reais, precisa ser arrumada; para tanto a fundamentação dessa interface é realizada por meio do uso de alguns importantes componentes:

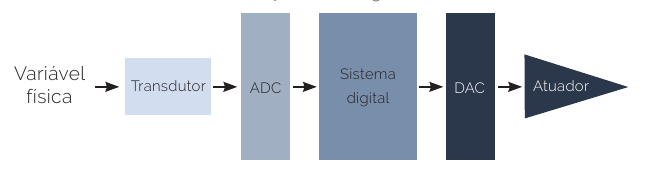
• Sensores ou transdutor: são empregados no processo de medição de todas as variáveis contínuas ou discretas.

• Atuadores: são empregados no acionamento de parâmetros contínuos ou discretos.

• Conversores de sinais analógicos em digitais: atuam no processo de conversão da informação proveniente dos processos reais para o formato digital.

• Conversores de sinais digitais em analógicos: atuam no processo de conversão da informação proveniente do formato digital para o formato dos processos reais.

• Dispositivos de entrada e saída de dados discretos: estabelecem a forma de entrada e saída das informações.



< Dia: extrair imagem do PDF. P. 157>

<LEG: Conversores analógico-digital e digital-analógico usados na interface das informações reais e digitais.>

De acordo com Haupt e Dachi (2018), a amplitude de um sinal amostrado pode ser quantizada em 2n níveis, sendo que cada n representa o número de bits que são utilizados para a representação de uma amostra no conversor analógico-digital.

**Conversores de sinal analógico em digital <T2>**

Os conversores de sinais analógicos em digitais compreendem um dos formatos mais simples de conversão. Nesse tipo de dispositivos, o processo de conversão pode ser dividido em etapas apresentado a seguir:

• Primeiramente, os dispositivos de medição, sensores e transdutores, geram o sinal analógico.

• É feito o acondicionamento do sinal analógico de maneira que ele assume a forma mais apropriada. É executada a filtragem do sinal para remoção de ruídos e é arranjada a conversão de uma forma de sinal em outra.

• No dispositivo de multiplexação, é efetivado o compartilhamento do tempo do conversor entre os canais de entrada.

• O sinal é, então, amplificado tornando-se compatível com a faixa do conversor.

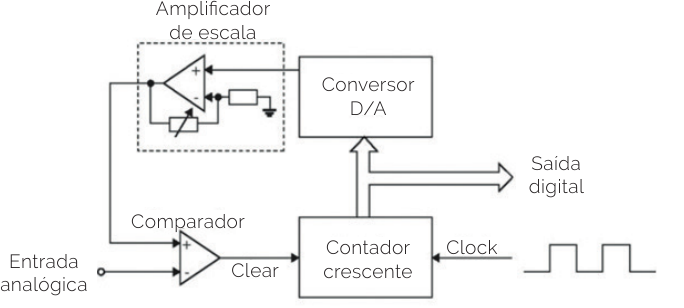
• O conversor converte o sinal de analógico para digital em três fases:

i. Amostragem: o sinal contínuo é convertido em uma série de sinais analógicos discretos em intervalos periódicos.

ii. Quantização: é atribuído cada sinal analógico discreto aos números finitos de níveis de amplitude definidos.

iii. Codificação: os níveis de amplitude discretos são convertidos em códigos digitais.

Os principais conversores são os de contagem ascendente e o de rastreamento.



< Dia: extrair imagem do PDF. P. 159>

<LEG: Conversor de contagem ascendente.>

O conversor de contagem ascendente composto de um amplificador de escala, o qual disponibiliza a variação da saída analógica do conversor D/A. O início do processo, com o contador resetado, é dado pela seguinte sequência:

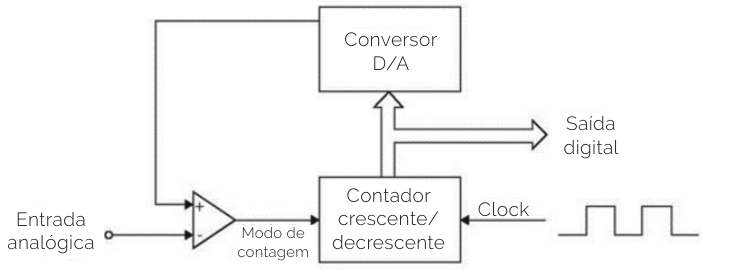
• O contador inicia, após o reset, a contagem ascendente na mesma frequência que o sinal de clock.

• As saídas atuam como entrada digital para o conversor D/A, a saída analógica cresce com o aumento da contagem.

• A saída do conversor D/A é comparada com a entrada analógica. O comparador fornece o nível lógico 0 quando a tensão presente na entrada positiva é menor ou igual a tensão da entrada negativa. Na ação inversa, a saída do comparador é 1.

• Enquanto a entrada analógica é maior que a saída do conversor D/A, a saída em nível lógico igual a 0 possibilita a contagem ascendente. Ao contrário, a contagem é inibida.

Esse conversor apresenta a desvantagem de que, para cada ciclo, o contador deve ser resetado, perdendo velocidade de conversão. Outra opção é o conversor A/D por rastreamento.



< Dia: extrair imagem do PDF. p. 160>

<Leg: Conversor A/D por rastreamento.>

A diferença desse conversor para o anterior reside no fato de que ele opera com contador crescente/decrescente. O início do processo de conversão, nesse caso, dá-se também com o contador resetado. Enquanto a tensão na entrada positiva for menor que na negativa, o contador é incrementado durante o clock. Quando a tensão na entrada positiva é maior que na negativa, o modo de contagem é revertido.

**Parâmetros de conversores A/D <T3>**

Existem diversos conversores A/D na forma de circuitos integrados. Para a escolha do tipo mais adequado, devem ser analisadas as características, as principais são: resolução, erro de quantização e ruído de quantização. A resolução consiste como uma função dos números de bits de saída, assim, quanto maior o número de bits na entrada, melhor será a resolução do conversor. O erro de quantização se associa à diferença entre o valor analógico e o digital para o qual este foi convertido. Nesse processo de conversão, é possível perder precisão por conta da resolução, ou seja, erro de quantização. O ruído de quantização diz respeito ao fato de que o processo de conversão pode inserir ruídos, indesejados, sua atenuação pode se dar por filtros digitais.

**Conversores de sinal digital em analógico <T3>**

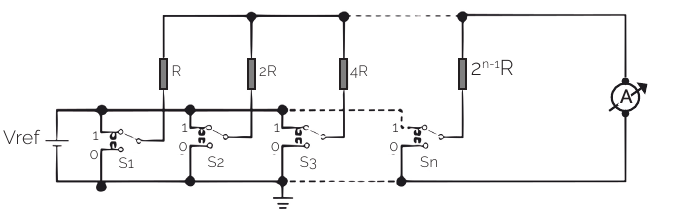
Equivalentemente ao funcionamento do conversor de sinal analógico em digital, o conversor de sinal digital em analógico atua na conversão de sinal, que sai da forma digital para analógica.

A conversão digital-analógico ocorre em duas etapas:

• Decodificação, por meio da qual a saída digital é transformada em valores analógicos de momentos discretos de tempo.

• Exploração de dados, por meio da qual cada valor consecutivo é convertido em sinal contínuo para ativação do atuador analógico ao longo do intervalo de amostragem.

Esse caso se baseia no processo de: um dado binário ou BCD é transformado em tensão ou corrente com valor equivalente ao valor digital.



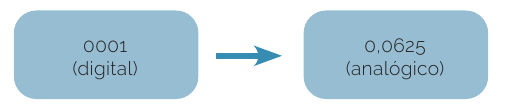
< Dia: extrair imagem do PDF. p. 162 >

<Leg: Diagrama de análise do conversor D/A.>

Considerando que cada chave presente no diagrama anterior é tida como sendo 1 bit, as duas posições relativas a determinada chave consistem em dois estados possíveis para o bit correspondente a essa chave.

Por exemplo, imagine um conversor de 4 bits conhecido como somador (R, 2R, 4R, 8R). Neste, quando todas as chaves assumirem a posição 0, tem-se que não é relatada a presença de corrente circulando ao longo do circuito; assim, no amperímetro não será exibido nenhum valor. A partir disso, a chave S4 está fechada, n é igual a 4, devido ao valor posicional, assim tem-se 00012, como sendo o valor binário associado e a corrente, que irá circular no circuito, será dada pela expressão a seguir.

Assim, tem-se a seguinte conversão:



< Dia: extrair imagem do PDF. p. 162>

<Leg: Resultado da conversão anterior.>

**Parâmetros de conversores D/A <T3>**

Existem diversos conversores D/A na forma de circuitos integrados. Para a escolha do tipo mais adequado, devem ser analisadas as características, as principais são: resolução e valor de fundo de escala. A resolução de um conversor é um valor associado ao número de bits presentes no conversor, quanto maior o número de bits na entrada, melhor será a resolução do conversor. Já o valor do fundo de escala diz respeito ao número máximo de saída analógica para o qual o conversor é capaz de fornecer.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Os fenômenos ocorridos na natureza, quando tratados em termos de sinais, constituem sistemas analógicos. Os sistemas digitais, por outro lado, operam com sinais digitais. As informações coletadas a partir dos sistemas reais constam em um estado analógico, ou seja, contínuo, necessitando, assim, de tratamento para que possam ser processadas em dispositivos digitais; a informação é convertida de analógica em digital. Já no caso necessário que a informação retorne para o formato utilizado nos processos reais, deve-se realizar o processo de conversão inverso, no qual a informação digital é convertida em analógica. A diferença exibida na interface entre os formatos digital, dos computadores, e o formato analógico, dos processos reais, precisa ser arrumada, para tanto a fundamentação dessa interface é realizada por meio do uso de alguns importantes componentes: sensores ou transdutor, atuadores, conversores de sinais analógicos em digitais, conversores de sinais digitais em analógicos e dispositivos de entrada e saída de dados discretos. Os elementos conversores ADC e DAC atuam como interfaces entre os dispositivos digitais e os processos do mundo analógico.

## **UNIDADE 4 - OBJETIVOS**

Olá. Seja muito bem-vindo à Unidade 4. Nosso objetivo é auxiliá-lo no desenvolvimento das seguintes competências profissionais até o término desta etapa de estudos:

1. Desenvolver algoritmos simples em sistemas digitais programáveis.

2. Entender a lógica por trás do processamento de sinais digitais.

3. Discernir a respeito de soluções computacionais, fundamentando dispositivos microcontroladores e microprocessadores.

4. Elaborar programas para dispositivos FPGA por meio da programação VHDL.

# **UNIDADE 04 - Capítulo 01**

# **Aspectos Essenciais da Lógica Programável e Software**

**Controladores lógicos programáveis <T2>**

O CLP, ou Controlador Lógico Programável, do inglês *Programmable Logic Controller,* compreende um dispositivo de estado sólido empregado para realização de operações de armazenamento de instruções e rotinas as quais direcionam o processo de gerenciamento e controle de um processo industrial. Tal dispositivo atua por meio da realização de operações lógicas, aritméticas, manipulando dados, aplicando funções e blocos operacionais, para implementação de procedimentos de contagem, sequenciamento lógico e temporização. Além de tudo, os CLPs podem operar em rede, estabelecendo comunicação com sistemas (SILVA, 2018).

Na pirâmide de automação industrial, os CLPs configuram o segundo nível, representando os dispositivos fundamentais nos processos automatizados desenvolvidos em uma planta industrial de processamento, uma vez que possibilitam o controle mecanizado, robotizado ou automatizado do processo.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

A pirâmide de automação classifica as diferentes camadas de TI das plantas de produção automatizadas industriais. Cada camada ou nível tem as próprias tarefas e infraestrutura de TI dentro das plantas de produção. Em geral, a pirâmide de automação é dividida em seis camadas.

## FIM DA SEÇÃO **<IMPORTANTE>**

Em geral, um CLP configura-se por uma unidade de módulos de entrada, outra de módulos de saída e outra de um processador, ou unidade de processamento central (CPU), além da fonte de alimentação.

Na entrada, o controlador recebe os sinais elétricos originários de máquinas e/ou processos industriais, nesse caso, o CLP é conectado diretamente aos sinais dos transdutores e atuadores. Na saída, é realizada a recepção dos sinais processados dentro do CLP.

Em seguida, eles são enviados como resposta para uso nas máquinas e nos processos respectivos, que convertem os sinais da unidade de processamento em sinal digital ou analógicos de forma a controlar os dispositivos de saída.

A unidade de processamento do CLP consiste no centro de operações lógicas em que ocorre o processamento necessário dos dados de acordo com a programação inserida pelo programador da indústria. Nesse caso, nessa unidade, encontra-se a programação da lógica ladder, o tipo de linguagem de programação do CLP.

Por fim, tem-se a fonte de alimentação atuante na conversão da corrente elétrica, de corrente alternada (CA) de entrada para corrente contínua (CC), energizando assim o sistema.

Assim, para executar todas as operações necessárias a um tipo de processo industrial específico ou a uma demanda de uma máquina, como operações lógicas e aritméticas, de temporização etc., o controlador é munido de uma memória, por meio da qual armazena as informações.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

Na memória, os dados são organizados em forma de lista de palavras, são os comandos e procedimentos de controle que devem ser seguidos. Assim, é por meio da orientação armazenada na memória, indicada por meio de sinais digitais e/ou analógicos, que máquinas e processos controlados operam.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

É possível descrever a atividade de um CLP em algumas etapas apresentadas a seguir:

• Ocorre a energização do sistema por meio da sua conexão à rede elétrica.

• É implementada a limpeza das memórias de entrada e saída e, assim, a redefinição do armazenamento do sistema.

• As operações de escrita e leitura na memória são testadas.

• A capacidade de execução do programa usuário é testada.

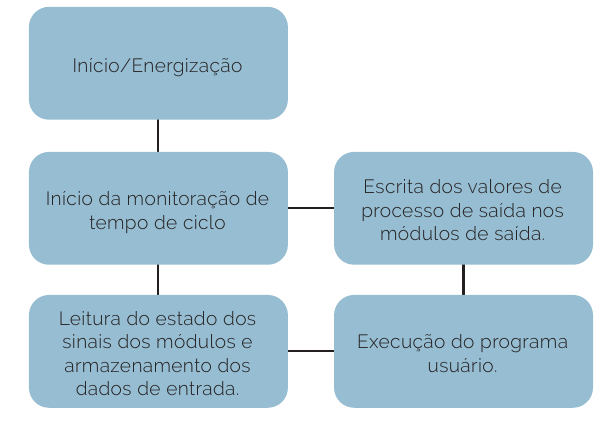
• É realizada a varredura.

Por meio da varredura, é feita a leitura dos estados de todos os componentes de entrada e a avaliação dos respectivos acionamentos. A unidade de entrada lê os estados dos dispositivos de entrada, como interruptores, e fornece informações ao processador. O processador processa as informações de acordo com a instrução da programação armazenada e os valores são guardados na memória. O programa é executado. É realizada a atualização das saídas tanto na memória como na saída física do sistema. Cada etapa da varredura ocorre em certo tempo, o qual varia de acordo com:

• Quantidade de entradas.

• Comprimento da lógica/loops no programa.

• Quantidade de saídas.



< Dia: extrair imagem do PDF. p. 173>

<Leg: Ciclo de varredura de um CLP.>

No final de cada ciclo de varredura, o CLP guarda os resultados obtidos e os reutiliza durante a execução do processamento do programa.

Segundo Silva (2018), o controle industrial se desenvolve em tempo real. Toda e qualquer alteração nos sinais de entrada leva à imediata ação no sinal de saída associado. Os controles programáveis são controladores com tecnologia embasada em tecnologias computacionais, assim compreendem dispositivos digitais. Para tais, o controle é estabelecido por meio do uso de softwares, cuja execução determina o desenvolvimento dos processos.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

Em geral, todo resultado fornecido pela lógica programável de um dispositivo é dependente da associação dos elementos de hardware e software em uma configuração de unidade funcional.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Geralmente, cada tipo de dispositivo possui suporte em termos de software de seu fabricante. Tais pacotes são comumente conhecidos como projetos auxiliados por computador ou CAD (*computer aided design*).

O processo de programação, nesse tipo de aplicação, demanda uso de um computador, onde é inserido o software de programação, o software propriamente dito, o dispositivo lógico, no caso aqui exemplificado como um CLP, e um sistema para conexão entre o dispositivo programável e o computador, de maneira que seja possível estabelecer um caminho para a inserção dos dados programados no dispositivo destino.

Para a inserção dos comandos lógicos representados no circuito lógico empregado no processamento em uso do hardware, é possível escolher entre duas formas diferentes de fazer esse processo: por meio da inserção via esquemática ou inserção via texto. No caso da inserção via texto, o uso convencional de linguagens como VHDL e verilog. No caso da via esquemática, a possibilidade de uso simbólico ou gráfico dos elementos lógicos do circuito, por exemplo, uso de portas lógicas.

Para simplificar o circuito lógico inserido no programa, faz-se uso da inserção por meio de blocos funcionais os quais simplificam circuitos mais complexos. No caso, os circuitos são inseridos em segmentos salvos como blocos e então conectados como circuito completo de forma hierárquica.

Uma vez inserido o circuito lógico, um compilador é usado para controlar as ferramentas de CAD, processando o esquema do circuito e produzindo uma implementação para o dispositivo destino. Em seguida, é feita a simulação funcional para garantir o funcionamento do programa inserido antes da inserção dele no hardware.

É realizada, então, a verificação da operação do circuito lógico, se ela estiver correta, após isso, o compilador ultrapassa diversas fases de forma automática preparando o projeto para o dispositivo de destino. Após essa síntese, o compilador implementa o projeto, mapeando e possibilitando o encaixe de acordo com a arquitetura e arranjo de pinos.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

No caso da interface de um software de um CLP, a figura a seguir exemplifica tal interface. Para atingir a fase da implementação do fluxo do projeto, o software tem que admitir os dispositivos específicos e ter informações detalhadas dos pinos.

A simulação de temporização constata que o circuito opera na frequência do projeto e não são visualizados atrasos de propagação ou outros problemas de temporização que afetem a operação global.

## FIM DA SEÇÃO **<IMPORTANTE>**

Após as simulações funcional e de temporização e, assim, após a verificação do funcionamento adequado do projeto é iniciada sequência de download. Nesse caso, uma sequência de bits é gerada para representação do projeto final, esta é enviada para o dispositivo destino e configurada automaticamente. Após isso, o projeto é testado no hardware.

Em geral, a linguagem utilizada na construção de programas para controladores programáveis é regulamentada pela IEC 61131-3, a qual unifica toda a sintaxe e semântica para este dispositivo digital. As linguagens identificadas, nesse sentido, são: lista de instruções, texto estruturado, diagrama de blocos funcionais, diagrama ladder e função de gráfico sequencial.

No caso, é possível dividir essas linguagens de acordo com a forma de representação de cada um, tendo em vista que se caracterizam como textuais e gráficas. Sendo as textuais a lista de instruções e a de texto estruturado, e as demais sendo as gráficas, no caso diagrama de blocos funcionais e diagrama ladder.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

CLP, ou Controlador Lógico Programável, do inglês *Programmable Logic Controller* compreende um dispositivo de estado sólido empregado para realização de operações de armazenamento de instruções e rotinas, as quais direcionam o processo de gerenciamento e controle de um processo industrial. Assim, para executar todas as operações necessárias a um tipo de processo industrial específico ou a uma demanda de uma máquina, como operações lógicas e aritméticas, de temporização etc., o controlador é munido de uma memória, por meio da qual armazena as informações. Os controles programáveis são controladores que têm tecnologia embasada em tecnologias computacionais, assim compreendem dispositivos digitais. Para tais, o controle é estabelecido por meio do uso de softwares, cuja execução determina o desenvolvimento dos processos.

# **UNIDADE 04 - Capítulo 02**

# **Fundamentação do Processamento de Sinais**

**Processamento de sinais digitais <T2>**

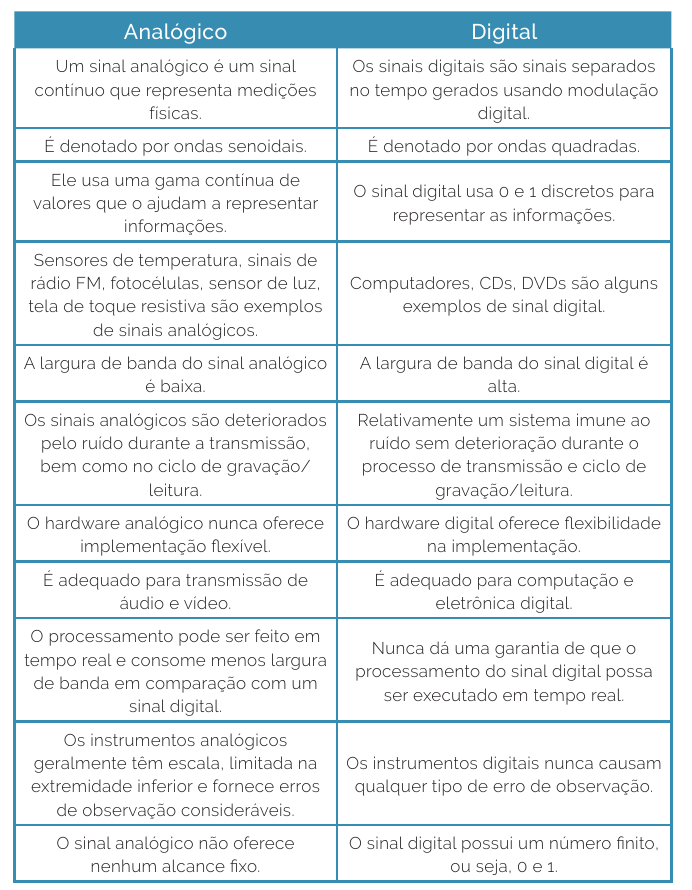
O Processamento Digital de Sinais é um importante ramo da engenharia de Eletrônica e Telecomunicações, o qual lida com a improvisação da confiabilidade e precisão da comunicação digital, aplicando múltiplas técnicas. Assim, consiste em uma importante tecnologia empregada em diversas aplicações como equipamentos automotivos, industriais, instrumentações, médicos, militares etc.

O ramo do processamento de sinais digitais atua na incorporação de conceitos matemáticos de programação e de hardwares atuantes no processamento de sinais analógicos. Alguns exemplos de incorporação do processamento de sinais digitais compreendem processos de melhoramento de imagens, atividades de compressão de dados, para possibilitar a transmissão e armazenamento, processos de reconhecimento de voz, com redução de ruídos etc.

Configuram-se sistemas digitais como originário da associação de dispositivos manipuladores da informação lógica ou dos dados e informações físicas por meio de formatos digitais.

Assim, um sinal digital é empregado na representação de dados por meio de sequência de valores intervalados em qualquer momento, os quais podem se apresentar como número fixado de valores, ou seja, representa um número real em uma faixa constante de valores.

Já no caso de sistemas analógicos, são compostos de dispositivos que manejam valores físicos em formato analógico. O sinal desse tipo se caracteriza por ser contínuo, apresentando uma quantidade variável no tempo, ou seja, é uma variável com base no tempo. Geralmente, compreendem sinais do tipo que funcionam com valores físicos e fenômenos naturais como temperatura, frequência, pressão, peso, iluminação, etc.



<Dia: recriar>

<Leg: Diferenças entre sinais analógicos e digitais.>

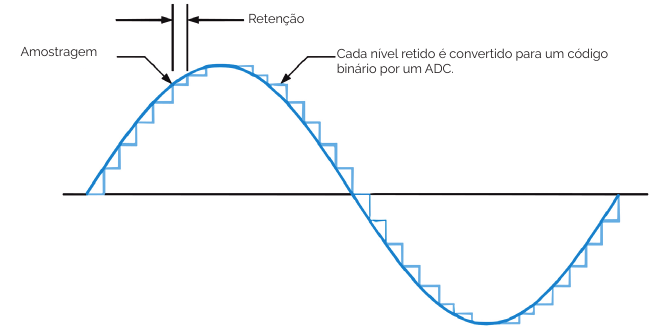
<Cotas: Analógico Digital Um sinal analógico é um sinal contínuo que representa medições físicas. Os sinais digitais são sinais separados no tempo gerados usando modulação digital. É denotado por ondas senoidais. É denotado por ondas quadradas. Ele usa uma gama contínua de valores que o ajudam a representar informações. O sinal digital usa 0 e 1 discretos para representar as informações. Sensores de temperatura, sinais de rádio FM, fotocélulas, sensor de luz, tela de toque resistiva são exemplos de sinais analógicos. Computadores, CDs, DVDs são alguns exemplos de sinal digital. A largura de banda do sinal analógico é baixa. A largura de banda do sinal digital é alta. Os sinais analógicos são deteriorados pelo ruído durante a transmissão, bem como no ciclo de gravação/ leitura. Relativamente um sistema imune ao ruído sem deterioração durante o processo de transmissão e ciclo de gravação/leitura. O hardware analógico nunca oferece implementação flexível. O hardware digital oferece flexibilidade na implementação. É adequado para transmissão de áudio e vídeo. É adequado para computação e eletrônica digital. O processamento pode ser feito em tempo real e consome menos largura de banda em comparação com um sinal digital. Nunca dá uma garantia de que o processamento do sinal digital possa ser executado em tempo real. Os instrumentos analógicos geralmente têm escala, limitada na extremidade inferior e fornece erros de observação consideráveis. Os instrumentos digitais nunca causam qualquer tipo de erro de observação. O sinal analógico não oferece nenhum alcance fixo. O sinal digital possui um número finito, ou seja, 0 e 1.>

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

Os dispositivos que operam com processamento de sinais digitais atuam na tradução dos sinais analógicos, os quais se caracterizam por variar continuamente, em sinais digitais, caracterizada em uma série de níveis discretos.

## FIM DA SEÇÃO **<IMPORTANTE>**

Séries de níveis analógicos configuram-se por uma variação de sinal com formato de escada. Assim, um sinal analógico original é modificado para se comportar como frações discretas convertendo-se a onda, caracterizada primordialmente como senoidal.



< Dia: extrair imagem do PDF. p. 182>

<Leg: Sinal analógico original (senoidal) e a sua aproximação na forma de “escada”.>

Observe na representação da figura anterior que a transformação do sinal analógico, original, em sinal digital ocorre por meio de uma aproximação, a qual é representada pela escadaria, e esse processo se dá por meio de um circuito de amostragem e retenção, também identificados.

A escada é a forma de aproximação dos dados e compreende uma forma quantizada em códigos binários, sendo cada um dos degraus discretos componentes da escada transformados a partir de processo de conversão analógica-digital (A/D). Dessa maneira, nesse caso, a conversão ocorre com uso de um conversor analógico-digital (ADC).

Após a conversão do sinal analógico, na forma codificada em binário, é destinado para um aplicado processador de sinais digitais (DSP). Tal processador é capaz de desenvolver variadas operações nos dados que recebe, por exemplo: remoção de ruídos e interferências, expansão da amplitude de certas frequências e redução de outras, codificação de dados e detecção e correção de erros.

O DSP, dessa forma, compreende um tipo especial de microprocessador, tendo em vista que processa dados em tempo real; nesse sentido, seu emprego é direcionado para sistemas processadores de dados digitalmente que representam sinais analógicos. Um DSP, de maneira similar a um microprocessador convencional, apresenta uma unidade central de processamento (CPU) e unidades de memória e diversas funções de interface.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

O processador de sinais digitais (DSP) é o elemento central em um sistema de processamento de sinais digitais. Esses componentes são programados em linguagem *assembly* ou linguagem C.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Por meio de processadores de sinais digitais, é possível realizar a limpeza de sons gravados, implementar processos de remoção de características indesejadas como ecos de linhas de comunicações, melhorar a resolução de imagens em sistemas importantes, como equipamentos de tomografia computadorizados, facilitando processos de diagnóstico médico, entre diversas outras operações complementares.

Posteriormente ao processamento do sinal pelo DSP, é passível de ser convertido para a sua forma original, ou seja, de digital para analógico, mas numa versão bastante aperfeiçoada. Tal processo é implementado por meio de um conversor digital-analógico (DAC).

Os DSPs constituem um tipo especializado de microprocessador, porém apresenta algumas particularidades importantes. Os microprocessadores típicos são construídos com objetivo de possuir funções de propósitos gerais, operando com uso de variados pacotes de software.

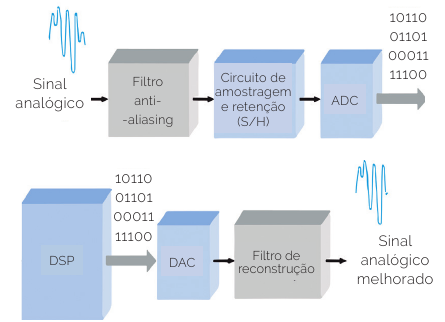
No caso dos DSPs, categoria de microprocessadores que apresentam aplicações específicas, consistem em processadores numéricos de alta velocidade, com capacidade de trabalhar em tempo real no processamento de informações na mesma proporção em que são geradas, tudo isso fazendo uso de algoritmos específicos.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

No caso do conversor analógico-digital (ADC), precisa receber as amostras do sinal analógico de entrada numa frequência satisfatória para capturar todas as variações importantes de amplitude do sinal; nesse caso, o processador DSP precisa sustentar a sincronia entre a taxa de amostragem do conversor de maneira que sejam feitos os cálculos em velocidade equivalente ao recebimento das amostras.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Após o processamento dos dados digitais pelo DSP, eles seguem para um conversor digital-analógico (DAC) em que são convertidos novamente para o formato analógico.



< Dia: extrair imagem do PDF. p. 184>

<Leg: Diagrama em bloco básico de um sistema de processamento de sinais digitais.>

O sinal passa primeiro pelo filtro anti-aliasing e pelo circuito de amostragem e retenção. No caso, a amostragem consiste no processo no qual se adquire um número satisfatório de valores discretos em pontos da forma de onda, que determinam a forma da onda. Assim, quanto maior o número de amostras obtidas, maior o índice de precisão da forma de onda definida. O processo de amostragem converte o sinal analógico em uma série de impulsos. Cada pulso obtido é dado por uma amplitude do sinal em certo instante do tempo.

Uma vez demandada a amostragem de um sinal, é importante estar atento a alguns requisitos os quais viabilizam a alta precisão do sinal representado. Em geral, todos os sinais analógicos possuem um espectro de componentes harmônicas, ondas senoidais com diferentes frequências e amplitudes. Uma vez as harmônicas somadas, o resultado é o sinal original.

Antes de ser amostrado, o sinal precisa passar por um filtro passa-baixos, ou filtro anti-aliasing, de maneira que as frequências harmônicas acima de certo valor, de acordo com a frequência de Nyquist, sejam eliminadas.

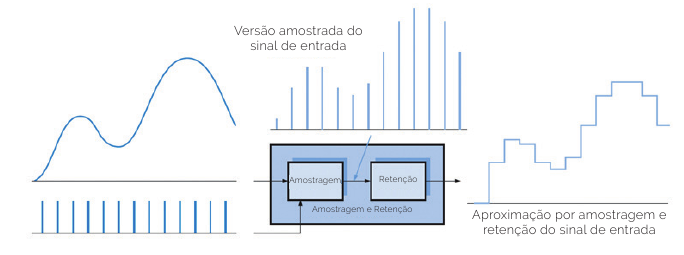
Caso existam componentes de frequência no sinal analógico, excedendo a frequência de Nyquist, ocorrerá uma condição aliasing (falseamento ou sobreposição), a qual é indesejada. O filtro anti-aliasing deve limitar o espectro de frequência do sinal analógico para uma dada frequência de amostragem.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

Evita-se o erro de aliasing, o filtro precisa eliminar todas as frequências acima da frequência mínima no espectro de amostragem ou aumentar a frequência de amostragem, valor que é limitado pelo desempenho do conversor analógico-digital (ADC).

## FIM DA SEÇÃO **<IMPORTANTE>**

Após a filtragem do sinal e da operação amostragem, o nível amostrado precisa ser mantido constante até a ocorrência da amostra seguinte, dessa maneira, é fornecido tempo para o ADC conseguir processar o valor amostrado. A amostragem e retenção resultam na forma de onda em escada com configuração aproximada a formada analógica de entrada.



< Dia: extrair imagem do PDF. P. 187>

<LEG: Operação de amostragem e retenção.>

O processo de conversão de analógico em digital consiste no processo de conversão da saída do circuito de amostragem e retenção em uma série de códigos binários, os quais remetem à amplitude do sinal de entrada analógico em cada instante amostrado.

A partir da amostragem e retenção, é mantido constante o sinal de entrada analógico entre os pulsos de amostragem, dessa forma, a conversão analógico-digital pode ser realizada por meio do uso de um valor constante ao invés de um sinal analógico variante ao longo do intervalo de conversão, correspondente ao tempo entre os pulsos de amostragem (FLOYD, 2007).

Dando seguimento ao processamento do sinal, a quantização como sendo o processo de conversão de um valor analógico em um código. Ao longo da quantização, o ADC converte os valores amostrados do sinal analógico em código binário. A precisão da representação cresce na medida em que mais bits são usados para representar um valor amostrado. Em seguida, tem-se a conversão dos sinais binários em analógicos.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Você deve ter aprendido que o ramo do processamento de sinais digitais atua na incorporação de conceitos matemáticos de programação e de hardwares atuantes no processamento de sinais analógicos. Configuram-se sistemas digitais como originário da associação de dispositivos manipuladores da informação lógica ou dados e informações físicas por meio de formatos digitais. Os dispositivos que operam com processamento de sinais digitais atuam na tradução dos sinais analógicos, os quais se caracterizam por variar continuamente, em sinais digitais, que se caracterizam em uma série de níveis discretos. O DSP compreende um tipo especial de microprocessador, tendo em vista que processa dados em tempo real; nesse sentido, seu emprego é direcionado para sistemas processadores de dados digitalmente representando sinais analógicos. Um DSP, de maneira similar a um microprocessador convencional, apresenta uma unidade central de processamento (CPU) e unidades de memória e diversas funções de interface.

# **UNIDADE 04 - Capítulo 03**

# **Soluções Computacionais em Eletrônica Digital**

**Microcontroladores e microprocessadores <T2>**

Foi a partir do desenvolvimento e da produção dos semicondutores, em meados dos anos 1970, que os dispositivos, os quais usam a tecnologia da eletrônica digital, puderam se desenvolver e expandir em passos cada mais significativos. A produção do primeiro dispositivo transistor possibilitou o início dessa nova era da tecnologia digital.

Transistores nada mais são que um dispositivo eletrônico construído com uso de material semicondutor aplicado no processo tanto de condução como de isolamento elétrico. Atuam como dispositivos de chaveamento e amplificação para controle e regulação de fluxos de sinais eletrônicos em um circuito.

Antes da criação dos transistores, eram utilizadas válvulas termoiônicas, as quais controlavam o fluxo de corrente elétrica em um circuito por meio do controle do fluxo em alto vácuo estabelecido entre eletrodos, arranjando dentro de recipiente completamente fechado.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

As válvulas termoiônicas exibiram um comportamento importante e possibilitavam o funcionamento dos primeiros computadores, no entanto, seu princípio era com base na ideia de que, ao aquecer um elemento em uma atmosfera de vácuo, era possível induzir a emissão de elétrons, criando assim uma corrente elétrica entre os eletrodos presentes no arranjo. Porém, esses componentes demandavam muito espaço físico ocupado, tendo em vista que sua estrutura era bastante robusta, além disso, visualizava-se grande perda de energia e muita produção de calor. Daí os transistores se tornaram uma opção mais eficiente.

## FIM DA SEÇÃO **<IMPORTANTE>**



<DIA: inserir imagem INOVA\_21\_2025\_F001>  
<Leg: Válvulas termiônicas.>

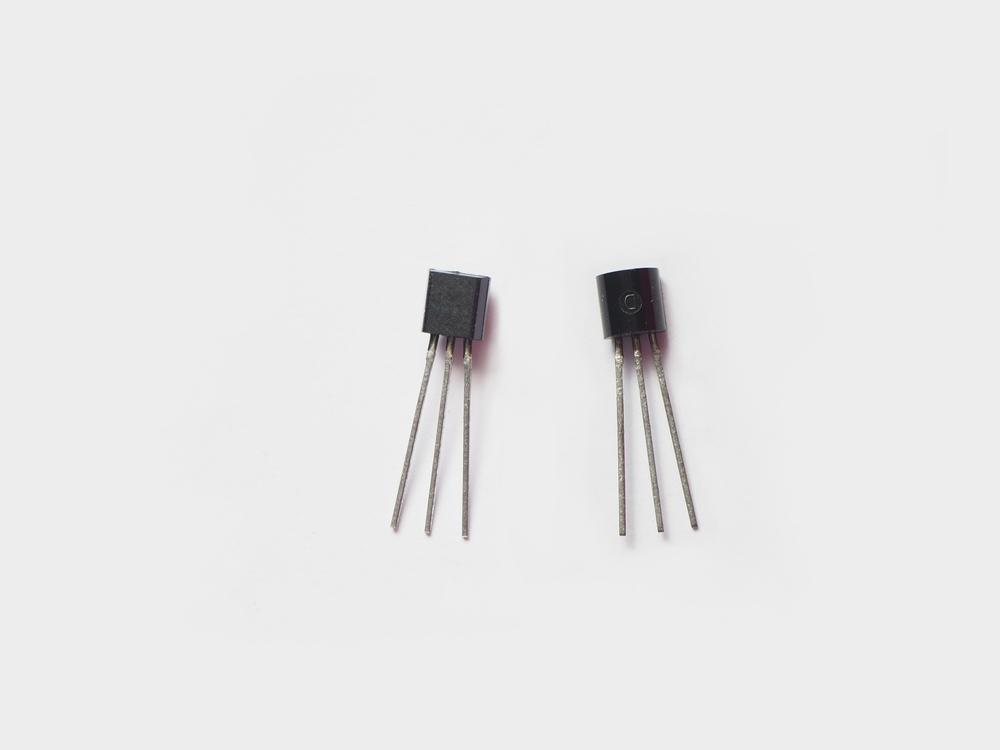
Os transistores são identificados como um componente fundamental para processadores, atrelado a esse fato que a evolução dos processadores se deu por conta da evolução dos transistores. Dessa maneira, como efeito em cascata, foi possível, e ainda é, visualizar a evolução dos computadores digitais e todos os demais dispositivos que operam com uso da eletrônica digital.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

As válvulas termoiônicas nasceram no século XX e sua criação possibilitou a produção dos primeiros computadores. O ENIAC (Electronic Numerical Integrator Analyser and Computer) foi o primeiro computador digital eletrônico e fazia uso de válvulas termoiônicas, cerca de dezoito mil válvulas termoiônicas, assim esse computador conseguia processar cerca de 5000 operações de adição, 357 operações de multiplicação e 38 divisões a cada segundo.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Os transistores possibilitaram a ágil execução de operações matemáticas, com custo menor e com menores desvantagens. A evolução dos transistores se deu, em geral, pelo estudo de diversos tipos de elementos que atuassem como semicondutores e pela miniaturização dessas estruturas.



<DIA: inserir imagem INOVA\_21\_2025\_F002>

<Leg: Transistores.>

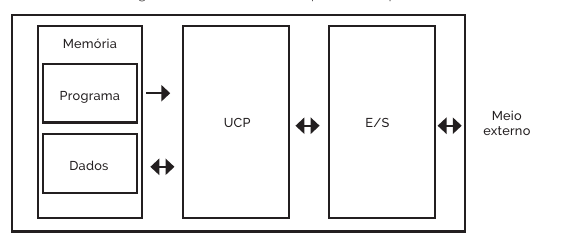
Os primeiros semicondutores empregados foram feitos de germânio, depois adotou-se o silício. A definição da necessidade progressiva de tentar aumentar a quantidade de transistores em um mesmo circuito integrado, mantendo o tamanho físico dos seus microchips, anunciada e decretada pelo fundador da empresa Intel, o empresário Gordon Moore, serviu como fenômeno impulsionador para a busca de evoluções dos circuitos integrados e para o desenvolvimento de tecnologias de miniaturização dos transistores e de sua acoplação em massa em um mesmo microchip. Esse fenômeno, por sua vez, impactou intensamente o desenvolvimento dos processadores.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

A ideia difundida por Moore se tornou bastante conhecida e é fundamental até a contemporaneidade como a Lei de Moore.

## FIM DA SEÇÃO **<IMPORTANTE>**

Como é de conhecimento comum, os computadores, em sua essência, são dispositivos com a capacidade de realização de cálculos matemáticos. A arquitetura de um computador é formada basicamente por três blocos: o bloco da unidade central de processamento (CPU), o bloco da unidade de armazenamento de dados e softwares, ou somente unidade de memória, e o bloco da unidade de entrada e saída.



<DIA: extrair do PDF. p. 192>  
<Leg: Sistema computacional padrão.>

Os processadores, por sua vez, compreendem um componente central dos computadores, eles são a unidade central de processamento (CPU).

A unidade central de processamento compreende o microprocessador do sistema computacional. Seu funcionamento pode ser comparado ao do cérebro humano, assim poderia ser entendido como o cérebro do sistema computacional, é o local onde são executadas todas as atividades inteligentes.

Essa função dos processadores só é possível por conta do conjunto de dispositivos arranjados de maneira a operar com processos lógicos, com processos de tomada de decisão e desenvolvimento de ações de demandadas na medida em que uma operação é executada.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

No que diz respeito à tomada de decisão, esse processo é desenvolvido sob instruções designadas por meio da execução de programas de acordo com o projeto pelo programador.

Assim, as diversas configurações de sistemas computacionais podem ser construídas a partir do uso de microcontroladores e microprocessadores.

## FIM DA SEÇÃO **<IMPORTANTE>**

Os microprocessadores são um formato especializado de circuito integrado destinado ao processamento mais robusto de dados e capaz de executar diversos programas ao mesmo tempo, trabalhando com uma grande extensão de dados.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

No mercado atual, há uma grande variedade de tipos de microprocessadores, os quais variam na capacidade de processamento disponibilizada; assim podem ser escolhidos de forma mais adequada ao tipo de aplicação ao qual será empregado (aplicações em computadores pessoais versus aplicação em supercomputadores industriais).

## FIM DA SEÇÃO **<IMPORTANTE>**

Os microcontroladores, por sua vez, compreendem circuitos integrados que também têm propriedades similares aos microprocessadores. Em sua arquitetura interna, é possível encontrar uma unidade de processamento, uma unidade de memória e unidades de entrada e saída, mas, nesse caso, uma memória (RAM e ROM) e alguns elementos periféricos, como interfaces seriais, PWM, contadores, temporizadores etc.

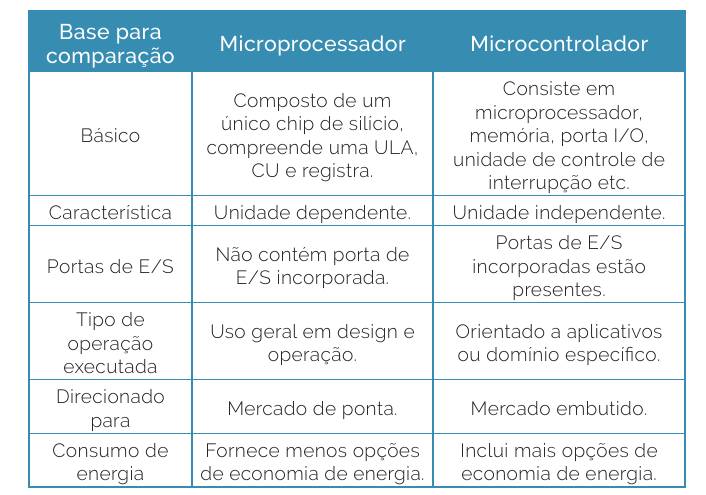
## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

Os sistemas embarcados (embedded system) são sistemas embutidos e compreendem um conjunto entre hardware e software, em que os microcontroladores são projetados para a execução de operações, tarefas e funções desenvolvidas pelo sistema abrangente do dispositivo, seja mecânico ou elétrico.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Assim, podem-se diferenciar microprocessadores e microcontroladores em alguns aspectos: enquanto os microcontroladores são construídos para compor sistemas embarcados, os microprocessadores são desenvolvidos para operações em computadores pessoais e outras aplicações gerais.

A grosso modo, essa é uma importante diferença, mas, em termos de estrutura configuracional do sistema, essa diferenciação pode ser mais complexa, sabendo que, na mesma proporção em que foi ficando mais fácil produzir um circuito integrado com custo mais baixo e com maiores aplicações, no caso dos microcontroladores, também visualizou-se a evolução da densidade e a complexidade do circuito integrado, tornando--se dois elementos grosseiramente muito parecidos.



<DIA: recriar>

<LEG: Comparação entre MCU e MPU.>

<COTAS: Base para comparação Básico Característica Portas de E/S Tipo de operação executada Direcionado para Microprocessador Composto de um único chip de silício, compreende uma ULA, CU e registra. Unidade dependente. Não contém porta de E/S incorporada. Microcontrolador Consiste em microprocessador, memória, porta I/O, unidade de controle de interrupção etc. Unidade independente. Portas de E/S incorporadas estão presentes. Uso geral em design e operação. Mercado de ponta. Consumo de energia Fornece menos opções de economia de energia. Orientado a aplicativos ou domínio específico. Mercado embutido. Inclui mais opções de economia de energia.>

Além disso, algumas importantes diferenças entre esses dois dispositivos podem ser relacionadas a partir da avaliação de custo relativo, velocidade de processamento e consumo energético; assim, quanto a:

• Custo

O valor correspondente à adoção de microcontroladores tendência para valores mais baixos, isso porque os microprocessadores são produzidos para emprego em dispositivos mais robustos e de maior valor associado, os quais irão atuar no controle de periféricos externos para estimular o desempenho, desenvolvem atividades mais complexas. Os microcontroladores possuem apenas a capacidade de execução de uma função pré-determinada.

• Velocidade

A velocidade de processamento é outro importante fator comparativo dos dois dispositivos.

• Consumo de energia

Por fim, tem-se o consumo de energia. Em microcontroladores, há baixo consumo de energia, eles normalmente são usados com baterias que podem segurar a atividade por longo tempo.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

Você deve ter aprendido que a produção do primeiro dispositivo transistor possibilitou o início dessa nova era da tecnologia digital. Os transistores são um dispositivo eletrônico construído com uso de material semicondutor aplicado no processo, tanto de condução como de isolamento elétrico. Atua como dispositivo de chaveamento e amplificação para controle e regulação de fluxos de sinais eletrônicos em um circuito. Os transistores são identificados como um componente fundamental para processadores. A arquitetura de um computador é formada basicamente por três blocos: o bloco da unidade central de processamento (CPU), o bloco da unidade de armazenamento de dados e softwares, ou, somente, unidade de memória, e o bloco de unidades de entrada e saída. Os processadores, por sua vez, compreendem um componente central dos computadores e são a unidade central de processamento (CPU). Assim, as diversas configurações de sistemas computacionais podem ser construídas a partir do uso de microcontroladores e microprocessadores.

# **UNIDADE 04 - Capítulo 04**

# **Elementos Básicos de Programação VHDL**

**FPGA e programação VHDL <T2>**

A prototipação de qualquer tipo de circuito digital, podendo ser um simples circuito até um complexo circuito composto de circuitos sequenciais, pode ser desenvolvida no dispositivo programável FPGA, ou *Field* (*Programmable Gate Array*), ou arranjo de portas programáveis por ação de campo.

Como se sabe, foi a partir dos avanços tecnológicos e pelo desenvolvimento das tecnologias de fabricação que os circuitos integrados tornaram-se mais potentes. Os transistores, por exemplo, são componentes importantes em todo esse processo e sua miniaturização possibilitou o avanço dos circuitos integrados.

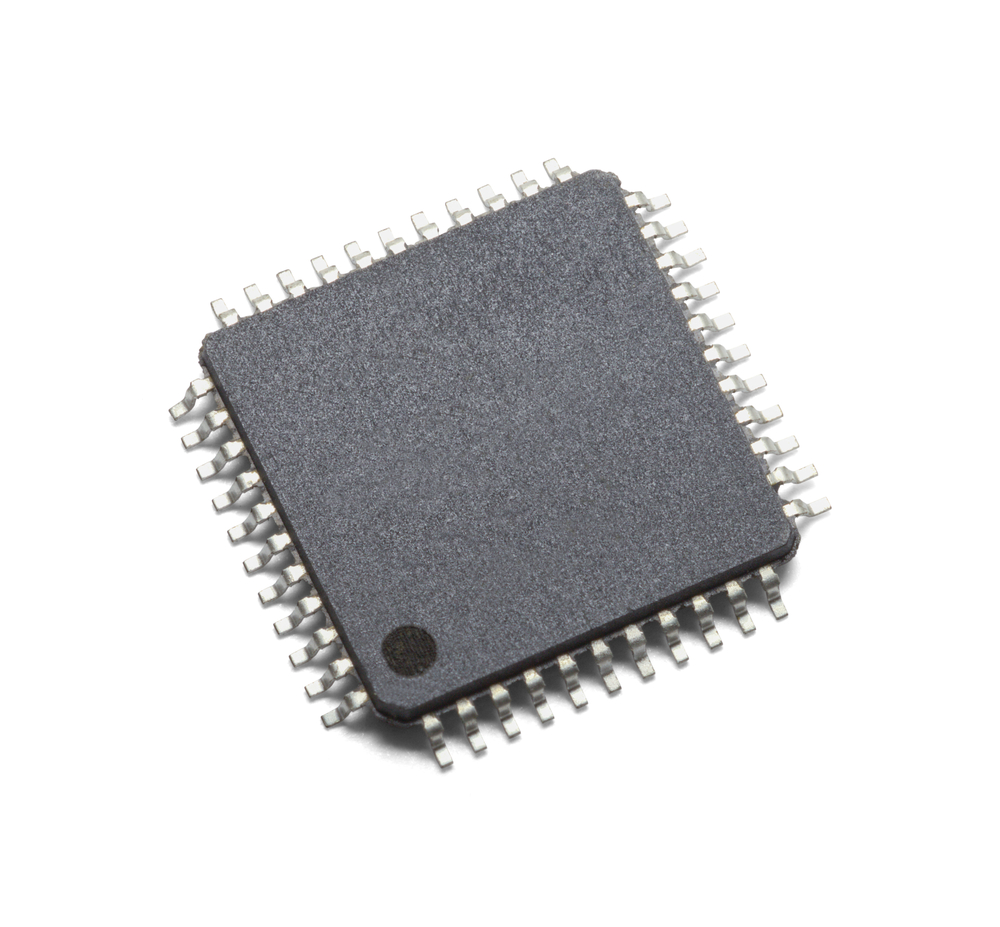
## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

Atualmente, os circuitos integrados são desenvolvidos com mais de 1 bilhão de transistores em apenas uma placa de circuito integrado.

## FIM DA SEÇÃO **<VOCÊ SABIA>**

Em geral, o mercado industrial atual de equipamentos eletrônicos é dominado pelos circuitos integrados que possuem aplicação específica, os conhecidos ASICs, ou *Aplication Specific Integrated Circuits*. Tais circuitos são produzidos para apresentarem funcionalidades ficadas, ou seja, executam uma mesma operação repetidas vezes de acordo com a aplicação do dispositivo eletrônico do qual fazem parte. Esse cenário incentivou um tipo de circuito integrado, capaz de reproduzir qualquer modelo de função lógica com base na modificação interna, fosse idealizado. Daí surgiram os PLDs, ou *Programmable Logic Devices*, ou ainda dispositivos lógico programáveis. Assim, existem diversos tipos de lógica programável, desde dispositivos de baixa capacidade até os de mais alta densidade que podem atuar em substituição de centenas de dispositivos de funções fixas.

Os dispositivos lógicos programáveis simples são os SPLD e esses foram os primeiros a serem criados. Atualmente, o SPLD é encontrado em aplicações de pequena escala. Ele é capaz, no caso um SPLD apenas, de substituir até 10 circuitos integrados que possuam função fixa, em função do tipo de funções e do tipo de SPLD.



<DIA: inserir imagem INOVA\_21\_2025\_F003>

<Leg: Encapsulamento de SPLDs.>

É possível encontrar o SPLD em duas categorias: o PAL e o GAL. O dispositivo PAL (*programmable array of logic*), só pode ser programado apenas uma vez, é composto de um arranjo de portas AND, programável, e um arranjo de portas OR, fixo. O dispositivo GAL (*generic array logic*), por sua vez, compreende um dispositivo PAL, mas pode ser reprogramado diversas vezes. Nesse caso, ele é composto basicamente de um arranjo de portas AND, programável, e um arranjo de portas OR, fixo, mas com saídas programáveis.

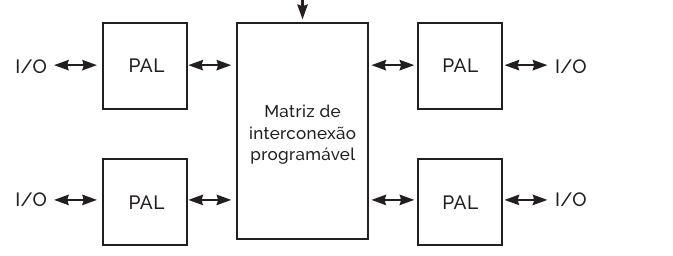
O outro tipo de PLD é o CPLD ou dispositivo de Lógica Programável Complexo. Tendo como base a evolução tecnológica e a expansão da capacidade de implementação de um chip por um circuito, ou seja, a densidade de chip, as indústrias que produziam os SPLDs desenvolveram os CLPDS, modelos mais complexos. Esse dispositivo é composto de múltiplos SPLDs, além disso ele é capaz de substituir vários circuitos integrados de funções fixas.

## INÍCIO DA SEÇÃO **<IMPORTANTE>**

Para o diagrama em bloco básico de um CPLD composto de quatro blocos de arranjo lógico (LABs), *logic array blocks*, e um arranjo de interconexões programáveis (PIA), *programmable interconection array,* como apresentado na f igura a seguir, a depender do tipo de CPLD específico, há a possibilidade de existir de 2 a 64 LABs.

## FIM DA SEÇÃO **<IMPORTANTE>**

Ou seja, o CPLD é formado por um arranjo de matrizes de lógica programável conectadas umas às outras por meio de uma matriz de interconexão programável. Compreende uma ampla rede de portas lógicas.



<DIA: extrair do PDF. P. 206>

<LEG: Estrutura básica de um CPLD.>

É comum visualizar o uso de CPLDs na implementação de quaisquer tipos de funções lógicas, como decodificadores, codificadores, multiplexadores, demultiplexadores e somadores. Eles possuem uma ampla variedade de configurações. Apresentam o encapsulamento com uma extensão de 44 a 160 pinos.

Por volta da década de 1980, foi desenvolvido então um novo modelo de PLD: o FGPA. O FGPA ou arranjo de Portas Programáveis por Campo consiste em um modelo mais robusto de PLD com densidade consideravelmente superior ao do CPLD. Muitas vezes é possível sobrepor as aplicações do FGPA e dos CLPDs.

Em geral, os FPGAs com grandes capacidades são passíveis de apresentar milhares de blocos lógicos, memória e outros elementos. A forma mais comum de encapsulamento é o BGA (*ball-grid array*), que pode apresentar mais de 1.000 pinos de entradas e saídas.

A montagem virtual de qualquer modelo de circuito digital pode ser implementada em um FGPA, respeitando-se as limitações de blocos lógicos e de frequência de operação do dispositivo. O FGPA pode ser reprogramado em campo, ou seja, diretamente no cliente, isso é importante, pois diminui os custos e há redução de tempo de resposta para correções de erros ou substituição de programações desnecessárias.

A montagem do circuito digital com esse dispositivo precisa da linguagem de descrição de hardware, ou seja, a linguagem HDL, ou hardware description language, uma linguagem que representa as noções primordiais do hardware, sua sintaxe e semântica, elementos possibilitadores de fluxo de dados e a temporização dos circuitos, além do manuseio dos fios de conexão, a construção das hierarquias etc. Diversas são as linguagens de descrição: Verilog, VHDL, AHDL, SDL etc.

## INÍCIO DA SEÇÃO **<VOCÊ SABIA>**

A linguagem VHDL foi criada pelo departamento de defesa dos Estados Unidos, por volta da década de 1980, como forma de padronizar a descrição de documentos VHSIC, *very high speed integrated circuit*. Em 1987, essa linguagem tornou-se pública pela norma 1076 no IEEE, *Institute of Electrical and Electronic Engineers.*

## FIM DA SEÇÃO **<VOCÊ SABIA>**

A VHDL é uma linguagem de programação que é usada para descrever, simular e criar hardware como circuitos digitais (ICS).

O HDL é usado principalmente para descobrir as falhas no projeto antes de implementá-lo no hardware. A principal vantagem dos HDLs é que eles fornecem recursos de modelagem flexíveis e podem expressar projetos grandes e complexos (maior que 107 portas).

VHDL significa Circuito de Integração de Alta Velocidade HDL (*Hardware Description Language*). É uma linguagem de descrição de hardware padrão do IEEE usada para descrever e simular o comportamento de circuitos digitais complexos. VHDL suporta os seguintes recursos:

• Metodologias de design e suas características.

• Atividades sequenciais e simultâneas.

• Troca de projeto.

• Estandardização.

VHDL suporta os seguintes recursos:

• Metodologias de design e suas características.

• Atividades sequenciais e simultâneas.

• Troca de projeto.

• Estandardização.

## INÍCIO DA SEÇÃO **<RESUMINDO>**

A prototipação de qualquer tipo de circuito digital, podendo ser um simples circuito até um complexo circuito composto de circuitos sequenciais, pode ser desenvolvida no dispositivo programável FPGA, ou Field – Programmable Gate Array, ou arranjo de portas programáveis por ação de campo. O FGPA ou arranjo de Portas Programáveis por Campo consiste em um modelo mais robusto de PLD com densidade consideravelmente superior ao do CPLD. Muitas vezes é possível sobrepor as aplicações do FGPA e dos CPLDs. O FGPA pode ser reprogramado em campo, ou seja, diretamente no cliente, isso é importante, pois diminui os custos e há redução de tempo de resposta para correções de erros ou substituição de programações desnecessárias. A VHDL é uma linguagem de programação usada para descrever, simular e criar hardware como circuitos digitais (ICS). A linguagem VHDL consiste em uma linguagem do nível de abstração de registradores e possibilita que os circuitos digitais sejam descritos estruturalmente e comportamentalmente.

# **REFERÊNCIAS**

FLOYD, T. L. Sistemas digitais: fundamentos e aplicações. 9. ed. Porto Alegre: Bookman, 2007.

HAUPT, A.; DACHI, E. Eletrônica digital. São Paulo: Blucher, 2018.

IDOETA, I.; CAPUANO, F. Elementos de eletrônica digital. São Paulo: Saraiva Educação SA, 1993.

OPPENHEIM, A. V.; SCHAFER, R. W. Processamento em tempo discreto de sinais. 3. ed. São Paulo: Pearson, 2013.

SILVA, A. Introdução às linguagens de Programação para CLP. São Paulo: Blucher, 2018.

TANENBAUM, A. S. Organização Estruturada de Computadores. 6. ed. São Paulo: Pearson Universidades, 2013.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. Sistemas Digitais: princípios e aplicações. 12. ed. São Paulo: Pearson, 2017.

WILLIAMS, L. Analog vs Digital: What is the Difference Between Analog and Digital? Guru99, 2022. Disponível em: https://www.guru99.com/ analog-vs-digital.html. Acesso em: 18 mar 2022.