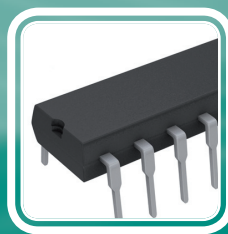


Circuitos Digitais

Rosumiro Trindade Junior

Jodelson Moreira Julião

Curso Técnico em Manutenção e
Suporte em Informática





Circuitos Digitais

Rosumiro Trindade Junior

Jodelson Moreira Julião



Manaus-Amazonas
2012

Presidência da República Federativa do Brasil
Ministério da Educação
Secretaria de Educação Profissional e Tecnológica

© Centro de Educação Tecnológica do Amazonas

Este Caderno foi elaborado em parceria entre o Centro de Educação Tecnológica do Amazonas e a Universidade Federal de Santa Catarina para a Rede e-Tec Brasil.

Equipe de Elaboração

Centro de Educação Tecnológica do Amazonas
– CETAM

Coordenação Institucional

Helder Câmara Viana/CETAM

Professor-autor

Rosumiro Trindade Junior/CETAM
Jodelson Moreira Julião/CETAM

Comissão de Acompanhamento e Validação

Universidade Federal de Santa Catarina – UFSC

Coordenação Institucional

Araci Hack Catapan/UFSC

Coordenação do Projeto

Sílvia Modesto Nassar/UFSC

Coordenação de Design Instrucional

Beatriz Helena Dal Molin/UNIOESTE e UFSC

Coordenação de Design Gráfico

Juliana Tonietto/UFSC

Design Instrucional

Gustavo Pereira Mateus/UFSC

Web Master

Rafaela Lunardi Comarella/UFSC

Web Design

Beatriz Wilges/UFSC
Mônica Nassar Machuca/UFSC

Diagramação

Bárbara Zardo/UFSC
Breno Takamine/UFSC
Liana Domeneghini Chiaradia/UFSC
Luiz Fernando Tomé/UFSC
Marília Ceriulli Hermoso/UFSC
Roberto Gava Colombo/UFSC

Revisão

Júlio César Ramos/UFSC

Projeto Gráfico

e-Tec/MEC

Catálogo na fonte elaborada pela DECTI da Biblioteca
Central da UFSC

T833c Trindade Junior, Rosumiro
Circuitos digitais / Rosumiro Trindade Júnior, Jodelson Moreira
Julião. – Manaus : Centro de Educação Tecnológica do Amazonas, 2012.
120p. : il., grafs. tabs.

Inclui bibliografia

Curso técnico em manutenção e suporte em informática

ISBN: 978-85-63576-41-5

1. Eletrônica digital. 2. Circuitos digitais. 3. Ensino a distância.
I. Julião, Jodelson Moreira. II. Título. III. Título: Curso técnico em
manutenção e suporte em informática.

CDU: 621.38.037.37

Apresentação e-Tec Brasil

Bem-vindo a Rede e-Tec Brasil!

Você faz parte de uma rede nacional de ensino, que por sua vez constitui uma das ações do Pronatec - Programa Nacional de Acesso ao Ensino Técnico e Emprego. O Pronatec, instituído pela Lei nº 12.513/2011, tem como objetivo principal expandir, interiorizar e democratizar a oferta de cursos de Educação Profissional e Tecnológica (EPT) para a população brasileira propiciando caminho de o acesso mais rápido ao emprego.

É neste âmbito que as ações da Rede e-Tec Brasil promovem a parceria entre a Secretaria de Educação Profissional e Tecnológica (SETEC) e as instâncias promotoras de ensino técnico como os Institutos Federais, as Secretarias de Educação dos Estados, as Universidades, as Escolas e Colégios Tecnológicos e o Sistema S.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade, e promover o fortalecimento da formação de jovens moradores de regiões distantes, geograficamente ou economicamente, dos grandes centros.

A Rede e-Tec Brasil leva diversos cursos técnicos a todas as regiões do país, incentivando os estudantes a concluir o ensino médio e realizar uma formação e atualização contínuas. Os cursos são ofertados pelas instituições de educação profissional e o atendimento ao estudante é realizado tanto nas sedes das instituições quanto em suas unidades remotas, os polos.

Os parceiros da Rede e-Tec Brasil acreditam em uma educação profissional qualificada – integradora do ensino médio e educação técnica, - é capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!

Desejamos sucesso na sua formação profissional!

Ministério da Educação
Dezembro de 2012

Nosso contato
etecbrasil@mec.gov.br

Indicação de ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.



Atenção: indica pontos de maior relevância no texto.



Saiba mais: oferece novas informações que enriquecem o assunto ou “curiosidades” e notícias recentes relacionadas ao tema estudado.



Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.



Mídias integradas: sempre que se desejar que os estudantes desenvolvam atividades empregando diferentes mídias: vídeos, filmes, jornais, ambiente AVEA e outras.



Atividades de aprendizagem: apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.

Sumário

Palavra dos professores-autores	9
Apresentação da disciplina	11
Projeto instrucional	13
Aula 1 – Sistemas digitais & sistemas analógicos	15
1.1 Introdução	15
1.2 Sistemas digitais & sistemas analógicos	15
1.3 Vantagens da eletrônica digital	17
1.4 Entendendo o 1 (um) e o 0 (zero)	18
1.5 Sistemas digitais	19
Aula 2 – Sistemas de numeração	23
2.1 Introdução	23
2.2 Sistema de numeração decimal	23
2.3 Sistema de numeração binário (base 2)	23
2.4 Sistema de numeração octal (base 8)	25
2.5 Sistema de numeração hexadecimal (base 16)	27
2.6 Equivalência entre o sistema hexadecimal e outros sistemas de numeração	27
2.7 Conversões hexadecimal-binário e binário-hexadecimal	28
Aula 3 - Portas lógicas e suas funções lógicas	31
3.1 Introdução	31
3.2 Funções: AND (E) , OR (OU) e NOT (NÃO)	31
3.3 Interligação das portas lógicas	36
3.4 Circuito lógico, expressão booleana e tabela-verdade	36

Aula 4 – Álgebra de Boole	41
4.1 Introdução	41
4.2 Revisando	42
4.3 Postulados (leis) booleanos	42
4.4 Propriedades booleanas	46
4.5 Leis booleanas	47
4.6 Princípio da dualidade	48
4.7 Teoremas de De Morgan	48
4.8 Exemplos de simplificação usando a álgebra booleana	49
4.9 Simplificação de expressão lógica através do Diagrama de Veitch Karnaugh (Mapa de Karnaugh)	50
4.9.2 Mapa de Karnaugh para duas variáveis	51
Aula 5 – Circuitos combinacionais	57
5.1 Introdução	57
5.2 Circuitos combinacionais de duas entradas	57
5.3 Circuitos combinacionais de três entradas	59
5.4 Circuitos aritméticos	61
5.5 Decodificador	64
5.6 Codificador	65
5.7 Multiplexadores	67
5.8 Demultiplexadores	71
Aula 6 – Flip-flops	77
6.1 Introdução	77
6.2 Conceito de memória	78
6.3 Bloco Elementar (<i>flip-flop</i>)	78
6.4 <i>Flip-flop</i> RS (<i>Reset – Set</i>) básico	79
6.5 Sinais de <i>clock</i> e <i>flip-flops</i> síncronos	81
6.6 <i>Flip-Flop</i> RS síncrono	82
6.7 <i>Flip-Flop</i> JK	83

Aula 7 – Circuitos sequenciais	87
7.1 Introdução	87
7.2 Registradores	87
7.2.5 Modo paralelo & modo serial de transmissão de um registrador	90
7.3 Contadores	95
7.4 Memórias	100
7.5 Exemplos de memórias	104
Aula 8 – Conversores	107
8.1 Conversores	107
8.2 Entendendo ondas	107
8.3 Características comuns aos conversores D/A e A/D	108
8.4 Conversor analógico/digital (A/D)	108
8.5 Conversor digital/analógico (D/A)	113
8.6 Sinais digitais	116

Palavra dos professores-autores

Caros estudantes e futuros colegas!

Temos imensa alegria de apresentar esta obra como um caminho para oportunizar a você, estudante, a possibilidade de ingressar em um dos mais fascinantes e promissores ramos das ciências exatas: a eletrônica digital. Enfatizamos a importância de sua participação em todas as atividades sugeridas, através dos *links* de acesso e figuras ilustrativas, pois, dessa forma, você estará adquirindo conhecimento, buscando informações interessantes e novas.

Esperamos que aproveitem ao máximo o tempo de estudo e o que sugerimos como conteúdo, pois todo embasamento teórico é importante para qualquer conhecimento prático e, por conseguinte, fundamental para uma boa atuação profissional. Esperamos que gostem e desejamos sucesso a todos.

Rosumiro Trindade Junior
Jodelson Moreira Julião

Apresentação da disciplina

Vivemos na era digital. Todos os equipamentos modernos, desde os mais simples aos mais avançados, e até mesmo as novidades que estão por vir, têm uma coisa em comum: são sistemas digitais e, como tal, são implementados com circuitos digitais. As portas lógicas e os *flip-flops* são as células básicas de todos os equipamentos digitais.

Este caderno, assim como a eletrônica digital, está basicamente dividido em duas partes: a lógica combinacional e a lógica sequencial; porém, iniciaremos vendo conceitos básicos, fundamentais para o entendimento das duas categorias de lógica.

Em seguida, na lógica combinacional estudaremos a implementação de circuitos lógicos a partir das portas lógicas. Na lógica sequencial veremos que a grande maioria dos circuitos digitais é pulsada, ou seja, precisa de um pulso de *clock* para sincronizar suas ações. Conheceremos como atuam alguns dos principais equipamentos de um computador, como por exemplo, as memórias.

Projeto instrucional

Disciplina: Circuitos Digitais (carga horária: 30h).

Ementa: Eletrônica digital; Sistemas digitais; Conceitos básicos de circuitos digitais; Sistemas de numeração; Funções e portas lógicas; Álgebra de Boole e simplificação de circuitos lógicos; Circuitos combinacionais; *flip-flop*, registradores e contadores; circuitos multiplex, demultiplex e memórias; aplicação prática de circuitos lógicos, conversores D/A e A/D, memórias semicondutoras e módulos de memória.

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
1. Sistemas digitais & sistemas analógicos	Conceituar sistemas digitais. Conhecer os conceitos básicos do sistema digital, como os estados lógicos 1 e 0. Conhecer os diferentes níveis de abstração dos circuitos digitais.	Texto explicativo com utilização de ilustrações e vídeo.	1
2. Sistemas de numeração	Caracterizar as diferentes bases de numeração. Representar números nas bases decimal, binária e hexadecimal. Conhecer os conceitos de dígito mais significativo e menos significativo. Efetuar a conversão entre decimal e as outras bases e vice-versa.	Texto explicativo com utilização de ilustrações e vídeo.	4
3. Portas lógicas e suas funções lógicas	Conhecer as portas lógicas básicas AND, OR, NOT, NAND, NOR, OU-EXCLUSIVO e COINCIDÊNCIA. Identificar os símbolos representativos das portas lógicas. Conhecer o funcionamento e as funções das portas lógicas básicas. Aprender a montar tabelas da verdade. Montar combinações de portas a partir da tabela-verdade e vice-versa.	Texto explicativo com utilização de ilustrações, pesquisa na internet e vídeo.	4
4. Álgebra de Boole	Conhecer os postulados e leis que regem a álgebra booleana. Fazer redução das expressões algébricas booleanas. Aprender a montar o mapa de Karnaugh. Saber desenhar circuitos a partir do mapa de Karnaugh.	Texto explicativo com utilização de ilustrações, pesquisa na internet e vídeo.	5
Continua			

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
5. Circuitos combinacionais	Caracterizar um circuito combinacional. Conhecer os circuitos codificadores/decodificadores, multiplexadores/demultiplexadores, comparadores e meio somadores/meio subtratores. Conhecer o funcionamento de cada um dos circuitos. Aprender a realizar operações de adição e subtração com números binários.	Texto explicativo com utilização de ilustrações, pesquisa na internet e vídeo.	5
6. <i>Flip-flops</i>	Distinguir circuito sequencial de circuito combinatório. Compreender o funcionamento do <i>flip-flop</i>. Conhecer diferentes tipos de <i>flip-flops</i>. Desenhar o circuito interno dos diferentes <i>flip-flops</i>.	Texto explicativo com utilização de ilustrações, pesquisa na internet e vídeo.	3
7. Circuitos sequenciais	Compreender o princípio de funcionamento de um contador. Compreender o princípio de funcionamento de um registrador de deslocamento e as suas características. Compreender a diferença entre transmissão de dados em série e a paralela.	Texto explicativo com utilização de ilustrações e pesquisa na internet.	6
8. Conversores	Definir conversores A/D e D/A. Compreender os princípios de funcionamento dos conversores A/D e D/A. Conhecer o princípio de amostragem por pulso e sua importância para o conversor A/D. Conhecer o princípio da quantização e sua importância para o conversor D/A. Fazer as representações gráficas dos conversores A/D e D/A de <i>n bits</i>.	Texto explicativo com utilização de ilustrações, e pesquisa na internet.	2
Conclusão			

Aula 1 – Sistemas digitais & sistemas analógicos

Objetivos

Conceituar sistemas digitais.

Conhecer os conceitos básicos do sistema digital, como os estados lógicos 1 e 0.

Conhecer os diferentes níveis de abstração dos circuitos digitais.

1.1 Introdução

Atualmente a eletrônica está dividida em dois segmentos que, certamente, você já deve ter ouvido falar.

- Eletrônica analógica
- Eletrônica digital

Vamos então, nesta aula, conhecer o que caracteriza um sistema digital, para que nas próximas aulas conheçamos os princípios básicos que regem a eletrônica digital, com suas unidades elementares e a álgebra booleana, que controla tais unidades de forma a ajudar a dar solução às necessidades propostas.

1.2 Sistemas digitais & sistemas analógicos

Sistemas digitais são sistemas simples e fáceis de entender. Em um conceito generalista podemos dizer que um sistema digital é todo e qualquer sistema que dê solução a um problema em pulsos. Já um sistema analógico pode dar solução ao mesmo problema de forma continuada, sem interrupções.

Por exemplo, imagine-se subindo uma rampa para vencer uma determinada altura, onde você pode ir subindo continuamente até chegar ao topo. Nesse caso podemos dizer que a solução encontrada para vencer a altura foi um sistema analógico, já que você vai subindo de forma regular e continuada. Agora se imagine subindo a mesma altura por meio de uma escada, onde você só pode subir a determinados intervalos (degraus). Nesse caso, dizemos que a solução encontrada é um sistema digital, pois você irá subir somente em intervalos contínuos.

Tomemos outro exemplo, a seguir.

Problema: utilizar um cronômetro para marcar um determinado tempo.

Para a solução desse problema, podemos utilizar dois tipos de relógios: um cronômetro que marque o tempo continuamente, sem interrupções, como o exibido na Figura 1.1, no qual podemos acompanhar o ponteiro se movimentando seguidamente com o passar do tempo, ou um que marque o tempo como o mostrado na Figura 1.2, em que só podemos acompanhar o passar do tempo em intervalos predeterminados.



Figura 1.1: Cronômetro analógico

Fonte: <http://www.getimage.com>



Figura 1.2: Cronômetro digital

Fonte: <http://www.getimage.com>

Vamos ver mais um exemplo.

Problema: utilizar um equipamento que faça medições elétricas.

Para a solução desse problema, podemos utilizar o multímetro mostrado na Figura 1.3 ou o mostrado na Figura 1.4. É fácil perceber a diferença entre os dois, já que no primeiro (multímetro analógico) podemos acompanhar a movimentação do ponteiro demonstrando a qualquer instante sua marcação. Já no segundo (multímetro digital) só podemos acompanhar os valores alterando em intervalos determinados.



Figura 1.3: Multímetro analógico

Fonte: www.eletronicadigital.com



Figura 1.4: Multímetro digital

Fonte: www.eletronicadigital.com

Há, porém, casos em que o sistema pode trabalhar com as duas formas simultaneamente; por exemplo, uma filmadora digital, que recebe como informação de entrada as imagens (entrada analógica) e grava o resultado em formatos digitais (saída e processamento digitais). Outro exemplo similar seria um gravador que um repórter usa durante uma entrevista que salve o resultado como um arquivo MP3.

Estabelecendo conceitos, podemos então afirmar a eletrônica analógica processa sinais com **funções contínuas** e a eletrônica digital processa sinais com **funções discretas**.

1.3 Vantagens da eletrônica digital

Com o uso de sistemas analógicos, o conjunto de valores representados é virtualmente infinito, o que torna o seu controle bem complexo. Já com o uso de um sistema digital, os valores representados passam a ser um número estabelecido, ou seja, finito, tornando o seu controle bem mais simples.

O sistema digital nos permite ainda usar uma técnica de numeração extremamente mais simples que a analógica, pois como usa sinais discretos, ou seja, facilmente numerados, podemos representá-los com a numeração binária (conjunto universo com apenas dois números: 1 e 0). Em contrapartida,



Assista na nossa videoteca disponível no AVEA ao vídeo Aula 01 - Sistema analógico & Sistema digital, ou acesse <http://www.metacafe.com/w/8434122> para assisti-lo

Caro estudante, após assistir ao vídeo, escreva um pequeno comentário e poste-o no fórum criado para essa finalidade. Discuta com dois outros colegas do fórum sobre o tema do vídeo.

o sistema analógico, por ter sinais contínuos, requer um sistema numérico mais complexo para representação dos seus resultados.

Assim sendo, podemos dizer que num dispositivo digital eletrônico teremos o processamento de conjuntos finitos cujos elementos se apresentam em apenas dois valores (0 ou 1). A esses conjuntos dá-se o nome de *bytes*, e aos seus elementos, o nome de *bits*.

1.4 Entendendo o 1 (um) e o 0 (zero)

Para compreendermos o que representam o 1 (um) e o 0 (zero) em sistemas digitais, vamos imaginar situações opostas umas às outras.

- Porta aberta – porta fechada.
- Luz acesa – luz apagada.
- Alto – baixo.
- Esquerda – direita.
- Norte – sul.
- Verdade – mentira.
- 1 – 0.

É exatamente isso o que o 1 (um) e o 0 (zero) representam. Um é o estado oposto de outro.

Para ficar mais claro, imagine a seguinte situação: um interruptor ligado a uma lâmpada. Quando o interruptor é acionado, a lâmpada se acende. Quando voltarmos a mover o interruptor, a lâmpada se apaga. Então, nesse caso poderíamos dizer que quando a lâmpada está acesa ela está em estado 1 (um). E quando ela está apagada, ela está em estado 0 (zero).

Ou ainda, imagine um semáforo. Quando a luz vermelha estiver acesa (estado 1) as demais luzes, amarela e verde, estarão apagadas (estado 0). Momentos depois a luz verde se acenda (passando para estado 1) fazendo com que a luz vermelha mude seu estado para 0.

Desse modo, dizemos que em circuitos digitais, quando há tensão em um determinado elemento do circuito, ele está em estado 1, e quando há ausência de tensão, ele está em 0. A Figura 1.5 mostra um circuito integrado representado um determinado instante quando há ou não tensão em alguns dos seus pinos. A resposta que esse elemento dará a esse instante dependerá do seu arranjo interno. Esse arranjo interno será mais bem compreendido nas demais aulas que teremos.



Assista na nossa videoteca disponível no AVEA ao vídeo Aula 02 - Estados lógicos 1 e 0, ou acesse <http://www.metacafe.com/w/8418147>

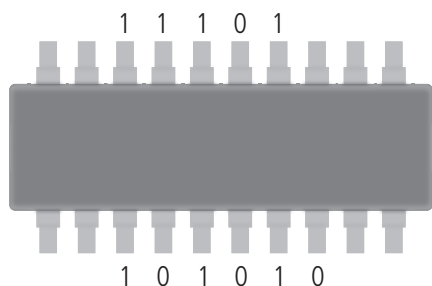


Figura 1.5: Circuito integrado representando estados em alguns pinos

Fonte: Adaptado de <http://www.digikey.com/>

1.5 Sistemas digitais

Conceitua-se um sistema eletrônico como um conjunto de elementos que são conectados de forma a compor um arranjo específico e assim concretizar uma funcionalidade específica. Por exemplo, equipamentos de som, que resultam em sinais sonoros, ou uma televisão, que resulta em sinais visuais.

Um sistema eletrônico digital pode então ser definido como um conjunto de componentes interconectados que processam informações em forma digital ou discreta (GÜNTZEL, 2001).

Há hoje uma enorme variedade de equipamentos digitais, cada um com grau maior ou menor de complexidade. Em geral, os sistemas podem ir de um, dois ou milhares de componentes. Quanto maior a complexidade do sistema, maior será o número de elementos que o compõe, tornando o seu entendimento e sua implementação mais detalhada, exigindo maiores níveis de abstração no processo de projetá-los.

O Quadro 1.1 mostra diferentes níveis de abstração para elementos que compõem um circuito digital.

Quadro 1.1: Nível de abstração	
NÍVEL	ELEMENTOS
Transistor	Transistores, resistores, capacitores.
Portas	Portas lógicas, <i>flip-flops</i> .
Registrador	Somadores, comparadores, contadores, registradores.
Processador	Processadores, controladores, ASICs.

Fonte: Güntzel (2001)

A-Z

Abstração

É o processo ou resultado de generalização por redução do conteúdo da informação de um conceito ou fenômeno observável, normalmente para reter apenas a informação que é relevante para um propósito particular. Através da abstração podemos imaginar as resultantes de determinada decisão ou ação, sem recorrer a mecanismos físicos ou mecânicos de resolução (ABSTRAÇÃO, 2012).

Conforme o Quadro 1.1 mostra, o menor nível de **abstração** é o nível de transistor, composto por transistores, resistores e capacitores, que mesmo sendo imensamente usados em sistemas analógicos podem também auxiliar na construção de sistemas digitais, de acordo com suas funcionalidades, com o objetivo de alcançar um fim desejado.

O nível seguinte, o das portas, composto por portas lógicas e *flip-flops*, assim como o nível dos registradores, composto por somadores, comparadores, contadores e registradores, será tema das nossas próximas aulas.

Resumo

Conhecemos o que diferencia um sistema analógico de um sistema digital, assim como passamos a entender a representatividade do estado 1 e do estado 0, e finalmente conhecemos os níveis de abstração de um sistema digital.

Atividades de aprendizagem

1. Relacione três problemas (situações) do seu dia a dia e suas respectivas soluções analógicas e digitais.
2. Crie cinco diferentes opostos que possam ser representados por 1 e 0.
3. Pesquise na internet e determine o valor lógico 0 ou 1, sendo 0 para falso e 1 para verdadeiro, de cada uma das seguintes proposições:
 - a) Cabe a Alexander Fleming a descoberta da penicilina. ()
 - b) Crateús é uma cidade do Mato Grosso do Sul. ()
 - c) Santos Dumont, inventor do avião, morreu de hepatite. ()
 - d) A estátua de Zeus é considerada uma das 7 Maravilhas do Mundo. ()

- e) The Beatles foi uma banda de rock originária da cidade de Reinheim (Alemanha). ()
- f) Joinville é a capital de Santa Catarina. ()
- g) Neil Armstrong foi o primeiro homem a pisar na lua em 1969. ()

Aula 2 – Sistemas de numeração

Objetivos

Caracterizar as diferentes bases de numeração.

Representar números nas bases decimal, binária e hexadecimal.

Efetuar a conversão entre a decimal e as outras bases e vice-versa.

2.1 Introdução

O homem, desde os tempos antigos, necessitou dar números às coisas que faziam parte de seu cotidiano, fossem elas animais ou objetos. Atualmente a dependência do homem em atribuir valores numéricos às coisas aumentou numa escala sem precedentes, de tal forma que a praticamente tudo é atribuído um valor. Nesta aula iremos abordar os sistemas numéricos (ou sistema de numeração) bem como suas aplicações.



Mídias integradas: Assista na nossa videoteca disponível no AVEA ao vídeo Aula 03 - Sistemas numéricos I, ou acesse <http://www.metacafe.com/w/8420369>

2.2 Sistema de numeração decimal

O mais usado e popular dos sistemas numeração é composto por dez caracteres ou dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9). Qualquer número é a combinação dos dez dígitos. Nos circuitos analógicos o processamento da informação é efetuado usando o sistema de numeração decimal.

- $255 = 200 + 50 + 5$
- $255 = 2 \times 10^2 + 5 \times 10^1 + 5 \times 10^0$
- $37,46 = 3 \times 10^1 + 7 \times 10^0 + 4 \times 10^{-1} + 6 \times 10^{-2}$

2.3 Sistema de numeração binário (base 2)

O sistema binário utiliza apenas dois caracteres (que são 0 e 1) para representar qualquer quantidade. A esses caracteres denominamos *bit* **binary digit**. O caractere mais à esquerda corresponde ao *bit* mais significativo, enquanto o caractere mais à direita corresponde ao *bit* menos significativo.

O conjunto de quatro bits é denominado **nibble**; o de oito *bits*, *byte*. Nos circuitos digitais, o processamento da informação é feito através do sistema de numeração binário.

$$1\ 1\ 0\ 1_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$1\ 1\ 0,1\ 0\ 1_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 1 \times 2^{-3}$$

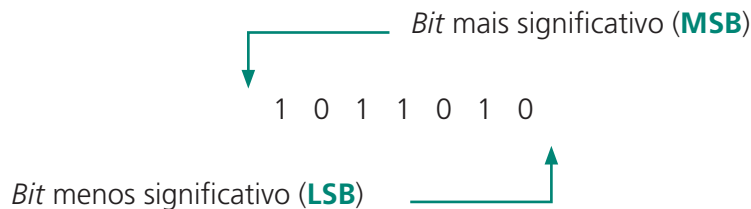
A-Z

MSB

Most Significant Bit
(bit mais significativo)

LSB

Least Significant Bit
(bit menos significativo).



2.3.1 Conversão binário-decimal



A conversão binário para decimal é feita somando-se os produtos das séries de potência de base dois.

Exemplos: $1\ 0\ 1\ 1_2$ conversão para decimal

$$\begin{aligned} \text{a)} \quad 1\ 1\ 0\ 1_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = \\ &= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = \\ &= 13 \text{ (decimal)} \end{aligned}$$



Mídias integradas: Assista na nossa videoteca disponível no AVEA ao vídeo Aula 06 – Conversão de binário em decimal, ou acesse <http://www.metacafe.com/w/8437191>

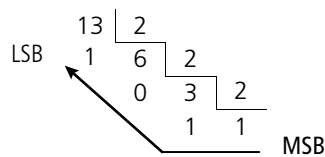
$$\begin{aligned} \text{b)} \quad 1\ 0\ 0\ 1,0\ 1_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 1 \times 8 + 0 + 0 + 1 \times 1 + 0 + 1 \times 1/4 \\ &= 8 + 1 + 0,25 \\ &= 9,25 \text{ (decimal)} \end{aligned}$$

2.3.2 Conversão decimal-binário



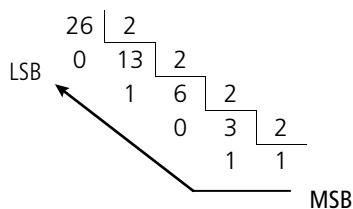
O processo consiste em efetuar sucessivas divisões por 2, pois essa é a base do sistema que queremos converter; quando efetuamos a divisão por 2, obtemos como resto os números 1 ou 0; esse resto será o número binário convertido do sistema decimal.

Exemplo 2 = 13 (decimal) → conversão para binário



Portanto : 13 (decimal) = $1\ 1\ 0\ 1_2$ (binário)

Exemplo 3 = 26 (decimal) → conversão para binário



Portanto: $26_{10} = 11010_2$ (26 decimal = 11010 binário)



Mídias integradas: Assista na nossa videoteca disponível no AVEA ao vídeo Aula 05 - Conversão de decimal para binário, ou acesse <http://www.metacafe.com/w/8434663>



Saiba mais: Para saber mais sobre conversão de números decimais fracionários para o sistema binário, acesse sistema binário (matemática) disponível em [http://pt.wikipedia.org/wiki/Sistema_bin%C3%A1rio_\(matem%C3%A1tica\)#Bin.C3.A1rios_a_decimais](http://pt.wikipedia.org/wiki/Sistema_bin%C3%A1rio_(matem%C3%A1tica)#Bin.C3.A1rios_a_decimais).

2.4 Sistema de numeração octal (base 8)

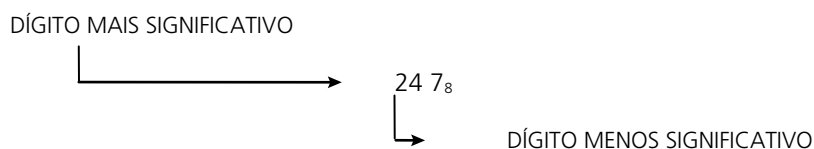
O sistema de numeração octal usa oito caracteres; com a combinação desses oitos caracteres se forma qualquer grandeza nesse sistema de numeração.

0, 1, 2, 3, 4, 5, 6 e 7

Exemplos de números octais: 2_8 , 18_8 e 257_8

2.4.1 Conversão octal-decimal e decimal-octal

Como exemplo vamos converter o número 247_8 (octal) para decimal



$$247_8 = 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 = 2 \times 64 + 4 \times 8 + 7 \times 1 = 128 + 32 + 7 = 167$$

Portanto : 247_8 (octal) = 167 (decimal)

Voltando para a base 8



2.4.2 Conversão octal-binário e binário-octal

Três dígitos binários são necessários para formar um dígito octal.

Quadro 2.1: Relação dígitos em binário e octal

binário	octal
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

Fonte: Elaborado pelos autores

Exemplo da conversão octal-binário e vice-versa.

35₈ em binário

3	5	→ Octal
0 1 1	1 0 1	→ Binário



A conversão é simples, basta separar cada dígito octal, da direita para a esquerda, lembrando que a cada dígito octal correspondem três dígitos binários.

A-Z

Microcontrolador

É um circuito integrado (*chip*) que pode ser programado via *software* (compiladores); diferencia-se do microprocessador por apresentar todos os tipos de memória interna e também outros periféricos. Todo microcontrolador é um microprocessador, pois tem a mesma arquitetura interna; podemos dizer que um microcontrolador é um computador de um só *chip*.

2.5 Sistema de numeração hexadecimal (base 16)

O sistema hexadecimal é um sistema alfanumérico cuja base é o número 16. O sistema de numeração hexadecimal usa os dez dígitos do sistema decimal e também as seis primeiras letras maiúsculas do nosso alfabeto. Apesar de não usar os dois níveis lógicos reconhecidos por qualquer máquina digital (0, 1), como os computadores, por exemplo, o sistema hexadecimal é muito utilizado na área de desenvolvimento de circuitos digitais e programação de **microcontroladores**. Podemos dizer que um número no sistema hexadecimal nada mais é que um número binário compactado; o mesmo podemos afirmar quando comparamos o sistema de numeração binário com o sistema de numeração octal.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F são os dígitos hexadecimais

Exemplos de números hexadecimais: $A3_{16}$ $5D4F_{16}$ $F0FA_{16}$



Mídias integradas: Assista na nossa videoteca disponível no AVEA ao vídeo Aula 04 - Sistemas numéricos II, ou acesse <http://www.metacafe.com/w/8434283>

2.6 Equivalência entre o sistema hexadecimal e outros sistemas de numeração

Quadro 2.2: Equivalência entre o sistema hexadecimal e outros sistemas de numeração

BINÁRIO	HEXADECIMAL	OCTAL	DECIMAL
0 0 0 0	0	0	0
0 0 0 1	1	1	1
0 0 1 0	2	2	2
0 0 1 1	3	3	3
0 1 0 0	4	4	4
0 1 0 1	5	5	5
0 1 1 0	6	6	6
0 1 1 1	7	7	7
1 0 0 0	8	10	8
1 0 0 1	9	11	9
1 0 1 0	A	12	10
1 0 1 1	B	13	11
1 1 0 0	C	14	12
1 1 0 1	D	15	13
1 1 1 0	E	16	14
1 1 1 1	F	17	15

Fonte: Elaborado pelos autores

2.7 Conversões hexadecimal-binário e binário-hexadecimal

Quatro dígitos binários são usados para representar um dígito hexadecimal.

Os *bits* são separados de quatro em quatro da direita para a esquerda para formar um único dígito hexadecimal.



Assista na nossa videoteca disponível no AVEA ao vídeo Aula 07 - Conversão de binário em hexadecimal ou acesse <http://www.metacafe.com/w/8440775> para assisti-lo.

Exemplo 1: Conversão do binário 11010011_2 para hexadecimal.

$$\begin{array}{r|l} 110 & 0011 \\ 1 & \\ \hline D & 3 \end{array}$$

$$11010011_2 = D3_{16}$$

Exemplo 2: Conversão do hexadecimal $13A_{16}$ para o binário.

$$\begin{array}{r|l|l} 1 & 3 & A \\ 0001 & 0011 & 1010 \end{array}$$

$$13A_{16} = 000100111010_2 = 100111010_2$$

Observe que para completar a coluna referente ao dígito 1 do número $13A$, foram acrescentados 000 apenas para formar o conjunto de quatro *bits* necessário; porém, ao unificarmos os dígitos 000100111010 os três primeiros zeros (à esquerda) foram dispensados do número final 100111010_2 .



Para saber mais sobre conversões de bases decimal, binário, hexadecimal, acesse http://www.joinville.ifsc.edu.br/~stefano/digital/Digital_Superior_parte1v1.pdf

Enfim, para converter do sistema de numeração decimal para os outros sistemas (binário, octal e hexadecimal), basta dividir pela base do sistema de numeração a ser convertido, aproveitando somente o resto da divisão que será o resultado da conversão. Para converter qualquer outro sistema de numeração para o sistema decimal, basta converter o número em questão em soma de potência com a base correspondente à base que se quer converter.

Resumo

Nesta aula você aprendeu a diferenciar os sistemas numéricos, conheceu os conceitos de dígito mais significativo e menos significativo e, ao final, aprendeu a realizar as conversões entre eles.

Atividades de aprendizagem

1. Converter os números decimais para o sistema binário:

a) 36_{10}

b) 87_{10}

c) 255_{10}

d) 123_{10}

2. Converter os números binários para o sistema decimal:

- a) 1111_2
- b) 100100_2
- c) 10101_2
- d) 11100_2 .

3. Converter para o sistema hexadecimal:

- a) 11011101_2
- b) 10011101_2
- c) 230_{10}
- d) 73_{10} .

4. Fazer as seguintes conversões:

- a) $3D_{16}$ para binário
- b) $F0FA_{16}$ para decimal
- c) $7C1_{16}$ para binário
- d) FF_{16} para binário.

Aula 3 - Portas lógicas e suas funções lógicas

Objetivos

Conhecer as portas lógicas básicas AND, OR, NOT, NAND, NOR, OU-EXCLUSIVO e COINCIDÊNCIA.

Identificar os símbolos representativos das portas lógicas.

Conhecer o funcionamento e as funções das portas lógicas básicas.

Aprender a montar tabelas da verdade.

Montar combinações de portas a partir da tabela-verdade e vice-versa.

3.1 Introdução

A álgebra de Boole, desenvolvida em meados do século XIX, possibilitou o desenvolvimento de todos os equipamentos digitais como computadores, telefone celular, MP3 entre outros. A implementação de circuitos digitais foi um salto significativo para a eletrônica, pois as limitações existentes no sistema analógicos foram superadas. Os sistemas digitais respondem às funções lógicas AND (E), OR (OU) e NOT (NÃO). Todos os blocos lógicos que implementam qualquer sistema digital são combinações das três funções básicas.

3.2 Funções: AND (E) , OR (OU) e NOT (NÃO)

Existem apenas dois estados para uma função lógica:

- o estado 0 (zero), o estado da não condução (ausência de tensão); e
- o estado 1 (um), o estado da condução (presença de tensão).

Os blocos lógicos ou portas lógicas podem assumir apenas dois níveis lógicos: 0 ou 1.

3.2.1 Função AND ou E

Essa função executa a multiplicação dos níveis lógicos (estados lógicos) presentes em suas entradas.

$$\text{Função E ou AND} \rightarrow S = A \cdot B$$

Observe a Figura 3.1.

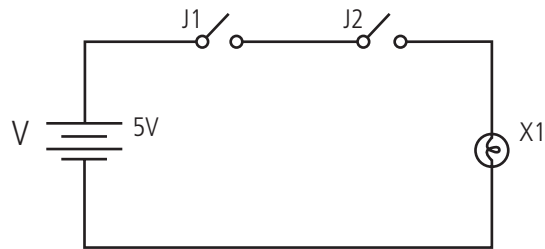


Figura 3.1: Circuito chaveamento Ew

Fonte: Elaborada pelos autores

- se J1 e J2 abertas, lâmpada X1 não acende;
- se J1 aberta e J2 fechada, lâmpada X1 não acende;
- se J1 fechada e J2 aberta, lâmpada X1 não acende; e
- se J1 e J2 fechadas, lâmpada X1 acende.

3.2.2 Porta lógica AND ou E

É o bloco lógico que leva em sua saída o nível lógico 1 apenas quando todas as suas entradas forem nível lógico 1, ou seja, executa a função lógica E.

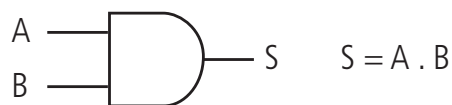


TABELA VERDADE		
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Figura 3.2: Porta lógica E

Fonte: Elaborada pelos autores

3.2.3 Função lógica OR ou OU

Essa função deixa sua saída em nível lógico alto (1) quando uma ou mais entradas estiverem em nível lógico alto e é representada pela adição dos níveis lógicos das suas entradas.

Função OU (OR) $\rightarrow S = A + B$

Observe a Figura 3.3

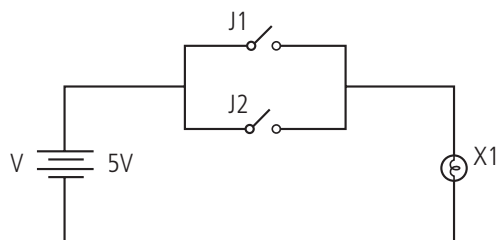


Figura 3.3: Circuito chaveamento OU

Fonte: Elaborada pelos autores

A lâmpada acenderá quando apenas uma ou todas as chaves – J1 e J2 – estiverem fechadas e somente apagará se todas as chaves – J1 e J2 – estiverem abertas.

3.2.4 Porta lógica OR ou OU

É o bloco lógico que leva em sua saída o nível lógico 1 quando apenas uma de suas entradas apresentar nível lógico 1.

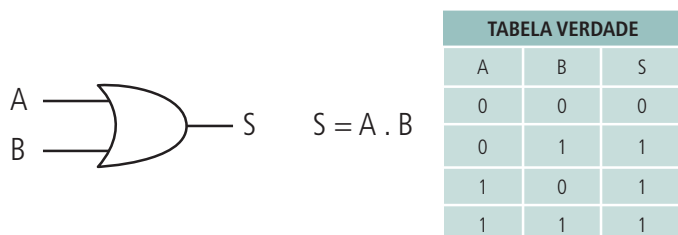


Figura 3.4: Porta lógica OU

Fonte: Elaborada pelos autores

3.2.5 Função lógica NOT ou NÃO

É a função que complementa ("inverte") o estado lógico de uma variável; ou seja, se o nível lógico de uma função é baixo (zero), a função complementa ("inverte") para nível alto (1) e vice-versa.

Função NÃO (NOT) $\rightarrow S = \bar{A}$

Observe a Figura 3.5.

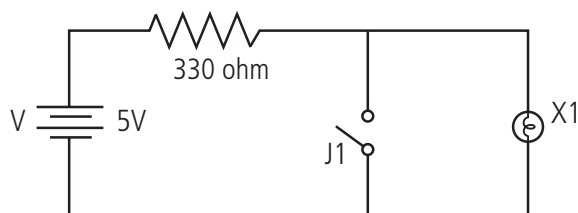


Figura 3.5: Circuito chaveamento NÃO

Fonte: Elaborada pelos autores

Enquanto a chave J1 estiver aberta, a lâmpada acenderá; fechando a chave J1, a lâmpada apagará, pois a corrente elétrica sempre irá fluir pelo caminho que oferece menor resistência elétrica, no caso o “caminho” com J1 fechada ao invés da lâmpada X1, que é uma resistência pura.

3.2.6 Porta lógica NOT ou NÃO (INVERSOR)

É o bloco lógico que inverte o estado lógico de uma variável, ou seja, quando a entrada está em nível lógico 1, teremos na saída nível lógico zero. Esta porta lógica executa a função lógica NOT (Figura 3.6)

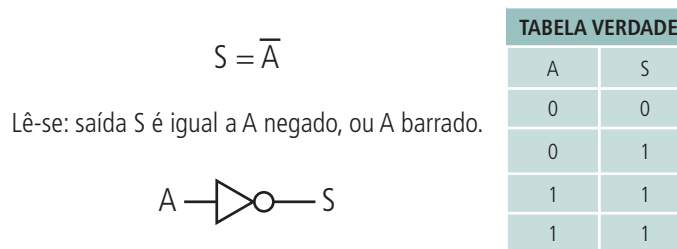


Figura 3.6: Porta lógica NÃO

Fonte: Elaborada pelos autores



A negação de um elemento também pode ser chamada de o seu complementar, ou seja, $\bar{A}$ é o complementar de A.



Para saber mais sobre as funções lógicas e portas lógicas, acesse o site <http://www.eletronicadigital.com/site/curso-eletronica-digital/8-licao2.html?start=4>

Existem outras funções lógicas que são combinações das funções básicas (E, OU, NÃO); a partir dessas funções, foram criadas as portas lógicas NAND , NOR , XOR e XNOR.

3.2.7 Porta NAND OU NE

É o bloco lógico que executa a função lógica NE. Essa porta lógica produz um 0 em sua saída apenas quando todas as entradas são 1 (Figura 3.7).

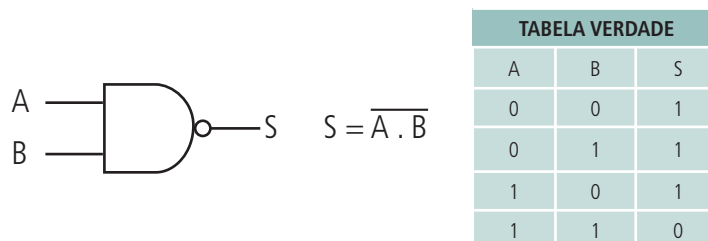


Figura 3.7: Porta lógica NAND ou NE

Fonte: Elaborada pelos autores

3.2.8 Porta lógica NOR ou NOU

É o bloco lógico que produz nível lógico zero em sua saída quando uma ou mais entradas são 1 (Figura 3.8).

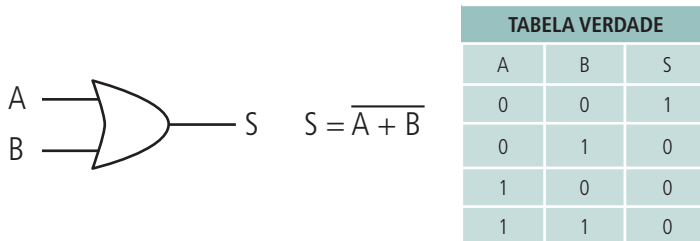


Figura 3.8: Porta lógica NOU ou NOR

Fonte: Elaborada pelos autores

3.2.9 Porta lógica OU-EXCLUSIVA (XOR)

É o bloco lógico que produz nível lógico 1 em sua saída apenas quando as entradas A e B estão em níveis lógicos diferentes (Figura 3.9).

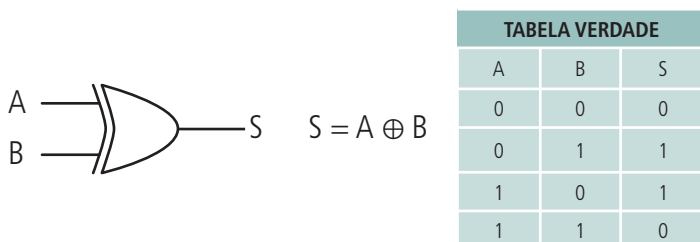


Figura 3.9: Porta Lógica XOR

Fonte: Elaborada pelos autores

3.2.10 Porta lógica COINCIDÊNCIA (XNOR)

É o bloco lógico que produz nível lógico 1 em sua saída apenas quando as entradas A e B estão em níveis lógicos iguais (Figura 3.10).

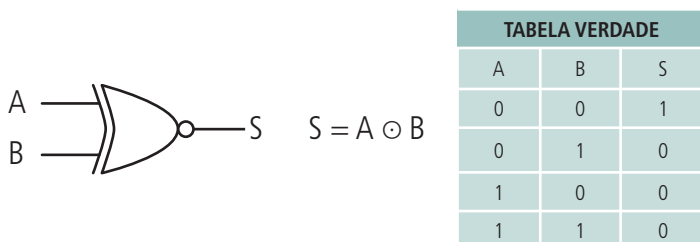


Figura 3.10: Porta lógica XNOR

Fonte: Elaborada pelos autores

As portas lógicas podem ter mais de duas entradas e irão executar suas respectivas tabelas-verdade de acordo com suas funções lógicas correspondentes. Na prática todas as portas lógicas estão sob a forma de **circuitos integrados (CI)**.

A-Z

Circuito integrado

Representado pela sigla CI ou IC, refere-se aos dispositivos utilizados em qualquer sistema digital.



Para saber mais sobre o *layout* dos circuitos integrados correspondente às portas lógicas, acesse http://www.joinville.ifsc.edu.br/~mtaques/Digital/ED_C1.pdf

3.3 Interligação das portas lógicas

As portas lógicas básicas podem apresentar mais de duas entradas; esses blocos lógicos são a base de qualquer sistema digital. Quando as portas se interligam, formam um circuito lógico que produz em sua saída uma expressão lógica (expressão booleana). A combinação das portas lógicas forma um circuito combinacional, pois a saída desses circuitos só depende das variáveis de entrada.

Exemplos de circuitos lógicos com suas respectivas expressões booleanas.

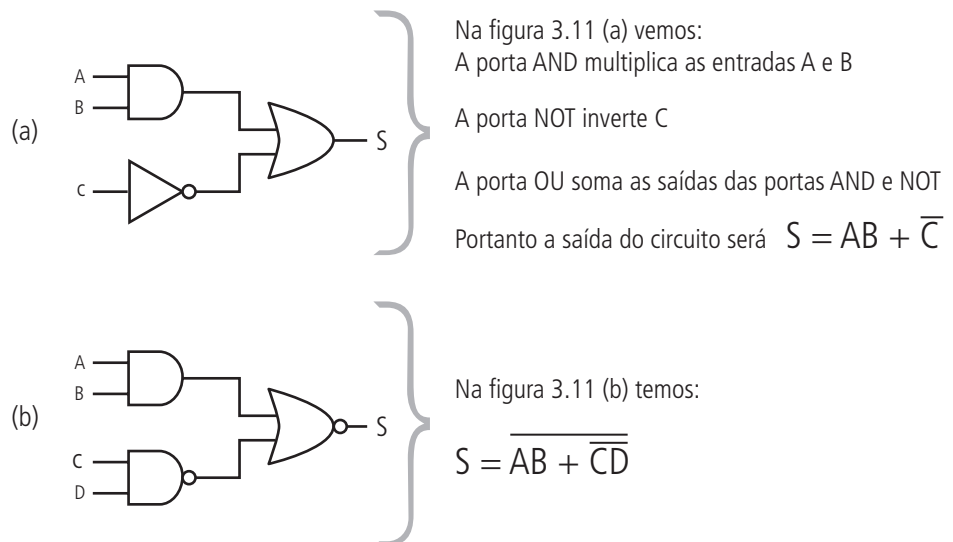


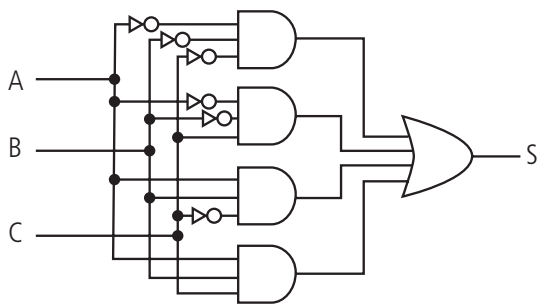
Figura 3.11: Exemplos de circuitos lógicos

Fonte: Elaborada pelos autores

3.4 Circuito lógico, expressão booleana e tabela-verdade

Com o circuito lógico podemos obter a sua expressão booleana e, a partir desta, sua respectiva tabela-verdade, que são as possíveis situações assumidas pela expressão lógica.

Exemplo resolvido 3.1: Determine a expressão booleana e construa a tabela-verdade.

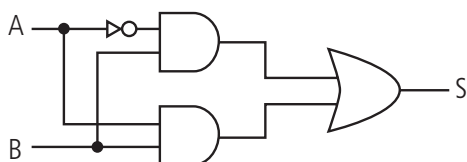


Podemos observar que a saída S é o resultado da soma de quatro portas AND, sendo que algumas entradas estão negadas.

$$S = \bar{A}.\bar{B}.\bar{C} + \bar{A}.\bar{B}.C + A.B.C + A.\bar{B}.C$$

A	B	C	S
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Exemplo resolvido 3.2: Determine a expressão booleana e construa a tabela-verdade do circuito abaixo.



$$S = \bar{A} B + A B$$

TABELA VERDADE		
A	B	S
0	0	0
0	1	1
1	0	0
1	1	1

Podemos também obter a expressão booleana da tabela-verdade e seu respectivo circuito lógico.

Método da soma de produtos

TABELA-VERDADE		
A	S	C
0	0	0
0	0	1
0	1	0

Para determinar a expressão booleana, basta verificar os resultados verdadeiros da tabela, ou seja, quando a saída apresentar nível lógico 1.

1º Resultado de $S = 1$: quando $A = 0$ e $B = 0$, implica em $\bar{A} \cdot \bar{B}$

2º Resultado de $S = 1$: quando $A = 1$ e $B = 0$, implica em $A \cdot \bar{B}$

3º Resultado de $S = 1$: quando $A = 1$ e $B = 1$, implica em $A \cdot B$

A soma dos resultados nos dá: $S = \bar{A} \cdot \bar{B} + A \cdot \bar{B} + A \cdot B$, que pode ser representado pelo circuito mostrado na Figura 3.12 a seguir.

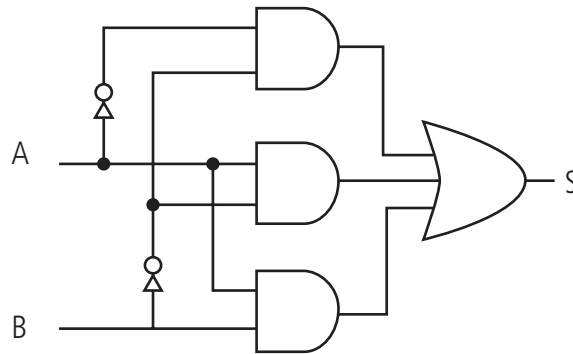


Figura 3.12: Soma de produtos

Fonte: Elaborada pelos autores



Percebe-se que quando é atribuído o valor 0 (zero) para uma variável, essa variável será complementada; e quando atribui-se 1, ela não é complementada. Somamos todos os produtos para obter a expressão lógica. Posteriormente simplificaremos nossa expressão booleana e, como consequência, será simplificado nosso circuito lógico. $S = \bar{A} \cdot \bar{B} + \bar{A} \cdot B + A \cdot B$

Exemplo resolvido 3.3: Determine a expressão booleana e o circuito lógico para a seguinte tabela-verdade.

$$S = \bar{A} \bar{B} + \bar{A} B$$

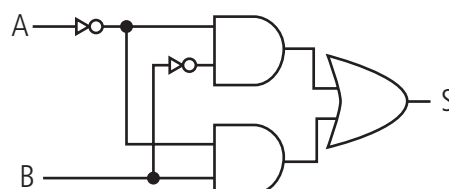


TABELA VERDADE			
A	B	S	
0	0	1	AB
0	1	1	AB
1	0	0	
1	1	0	

Resumo

Nesta aula conhecemos as portas lógicas básicas, aprendemos a identificar os símbolos representativos das portas lógicas, conhecemos o funcionamento e as funções das portas lógicas básicas; aprendemos a montar tabelas-verdade e, por último, aprendemos a montar combinações de portas a partir da tabela-verdade.

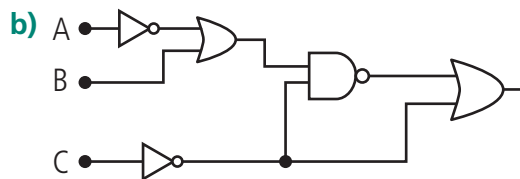
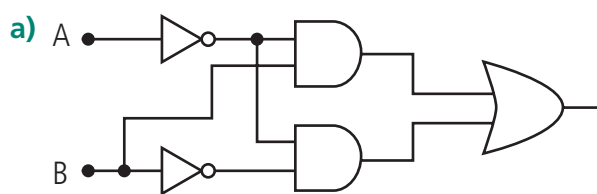
Atividades de aprendizagem

1. A tabela-verdade abaixo corresponde a qual função:

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

- a) AND (E)
- b) NAND (Não-E)
- c) OR (OU)
- d) OU-EXCLUSIVA (XOR)

2. Obter a expressão booleana dos seguintes circuitos lógicos



3. No circuito abaixo, determinar a expressão booleana e completar a tabela-verdade:

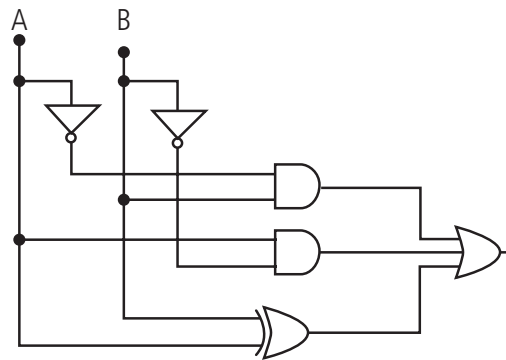


TABELA VERDADE		
A	B	S
0	0	
0	1	
1	0	
1	1	

4. Dada a tabela-verdade abaixo, determine a expressão lógica e o seu respectivo circuito lógico:

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

5. Se numa implementação lógica precisarmos usar inversores em série (duas portas não seguidas), qual o resultado obtido na saída da segunda?

Aula 4 – Álgebra de Boole

Objetivos

Conhecer os postulados e leis que regem a álgebra booleana.

Fazer redução das expressões algébricas booleanas.

Aprender a montar o mapa de Karnaugh.

Saber desenhar circuitos a partir do mapa de Karnaugh.

4.1 Introdução

George Boole, um matemático inglês, em meados do século XIX desenvolveu uma teoria completamente diferente para a época, baseada em uma série de postulados e operações simples para resolver uma infinidade de problemas.

Suas soluções foram então denominadas de álgebra de Boole e possuem a capacidade de resolver problemas práticos de controle e fabricação de produtos. Porém, à época não havia eletrônica nem máquinas suficientemente avançadas para utilizar seus princípios, os quais eram usados apenas pela filosofia.

A álgebra de Boole veio então, com o advento da eletrônica, a se tornar importante especificamente para a eletrônica digital, tornando-se uma importantíssima ferramenta em projetos de circuitos lógicos, pois através desse recurso, um circuito complexo pode ser reduzido a um mais simples, mas executando os mesmos comandos do circuito complexo original.

Veremos agora os postulados e teoremas da Álgebra de Boole, bem como o teorema de De Morgan, que são nossas ferramentas matemáticas na simplificação de expressões booleanas e, como consequência, a simplificação de circuitos lógicos digitais.



Veja a definição de teorema acessando <http://pt.wikipedia.org/wiki/Teorema>

Postulado ou axioma

É uma proposição que não pode ser demonstrada e é considerada como óbvia ou como um consenso inicial necessário para a construção ou aceitação de uma teoria (AXIOMA, 2012).

4.2 Revisando

Antes de vermos os **postulados**, leis e princípios da álgebra booleana, vamos rever o que já sabemos.

4.2.1 Adição ou união (OU)

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 1$

4.2.2 Multiplicação ou interseção (E)

- $0 \cdot 0 = 0$
- $0 \cdot 1 = 0$
- $1 \cdot 0 = 0$
- $1 \cdot 1 = 1$

4.2.3 Complemento ou negação

- $\overline{1} = 0$
- $\overline{0} = 1$

4.3 Postulados (leis) booleanos

4.3.1 Postulado 1 (Lei do elemento absorvente para a soma lógica)

A soma lógica de uma variável binária mais um 1 lógico equivale a 1 lógico, ou,

- $A + 1 = 1$

Demonstração:

- Para $A + 1$, se A igual a 0, pela ADIÇÃO, $0 + 1 = 1$
- Para $A + 1$, se A igual a 1, da mesma forma, $1 + 1 = 1$
- Desse modo, independente do valor de A (0 ou 1), $A + 1$ será sempre igual a 1.

4.3.2 Postulado 2 (Lei do elemento neutro para a soma lógica)

A soma lógica de uma variável binária mais um 0 lógico equivale ao valor da variável binária, ou,

- $A + 0 = A$

Demonstração:

- Para $A + 1$, se $A = 0$, pela ADIÇÃO, $0 + 1 = 1$
- Para $A + 1$, se $A = 1$, da mesma forma, $1 + 1 = 1$
- Desse modo, verificamos que nas duas hipóteses para o valor da variável A o resultado sempre será o valor dessa variável A .

4.3.3 Postulado 3 (Lei do elemento neutro para a multiplicação lógica)

O produto lógico de uma variável binária por um 1 lógico é igual ao valor da variável binária, ou,

- $A \cdot 1 = A$

Demonstração:

- Para $A \cdot 1$, se $A = 0$, pela MULTIPLICAÇÃO, $0 \cdot 1 = 0$
- Para $A \cdot 1$, se $A = 1$, da mesma forma, $1 \cdot 1 = 1$
- Desse modo, verificamos que nas duas hipóteses para o valor da variável A o resultado sempre será o valor dessa variável A .

4.3.4 Postulado 4 (Lei do elemento absorvente para a multiplicação lógica)

O produto lógico de uma variável binária por um 0 lógico é igual a um 0 lógico, ou,

- $A \cdot 0 = 0$

Demonstração:

- Se $A = 0$, temos que $0 \cdot 0 = 0$
- Se $A = 1$, temos que $1 \cdot 0 = 0$
- Desse modo, independente do valor de A (0 ou 1) o resultado será sempre igual a 0.

4.3.5 Postulado 5 (Lei da idempotência)

Tanto a soma quanto a multiplicação lógica de duas variáveis binárias iguais equivalem ao valor lógico dessa variável binária, ou,

- (1) $A + A = A$
- (2) $A \cdot A = A$

Demonstração (1):

- Para $A + A$, se $A = 0$, pela ADIÇÃO, $0 + 0 = 0$
- Para $A + A$, se $A = 1$, da mesma forma, $1 + 1 = 1$
- Desse modo, independente do valor de A (0 ou 1) $A + A$ será sempre igual a A .

Demonstração (2):

- Para $A \cdot A$, se $A = 0$, pela MULTIPLICAÇÃO, $0 \cdot 0 = 0$
- Para $A \cdot A$, se $A = 1$, da mesma forma, $1 \cdot 1 = 1$
- Desse modo, independente do valor de A (0 ou 1) $A \cdot A$ será sempre igual a A .

4.3.6 Postulado 6 (Lei do elemento complementaridade para a soma lógica)

A soma lógica de uma variável binária mais a negação da mesma variável binária equivale a 1 lógico, ou,

- $A + \bar{A} = 1$

$\bar{A}$ também é comumente representado por A' .



Demonstração:

- Se $A = 0$, temos que A' será 1, então $1 + 1 = 1$
- Se $A = 1$, temos que A' será 0, então $0 + 1 = 1$
- Desse modo, independente do valor de A (0 ou 1), $A + A'$ será sempre igual a 1.

4.3.7 Postulado 7 (Lei do elemento complementar para a multiplicação lógica)

O produto lógico de uma variável binária mais a negação da mesma variável binária equivale a 0 lógico, ou,

- $A \cdot \bar{A} = 0$

Demonstração:

- Se $A = 0$, temos que A' será 1, então $1 \cdot 0 = 0$
- Se $A = 1$, temos que A' será 0, então $0 \cdot 1 = 0$
- Desse modo, independente do valor de A (0 ou 1), $A' \cdot A$ será sempre igual a 0.

4.3.8 Postulado 8 (Lei da involução)

Se uma variável binária é negada duas vezes, esta não varia, ou,

- $A = A$



A também pode ser representado por A'' .

Demonstração:

- Se $A = 0$, então $(A')'$ será $(1)' = 0$
- Se $A = 1$, então $(A')'$ será $(0)' = 1 =$
- Desse modo, independente do valor de A (0 ou 1), A'' será sempre igual a A .

Este postulado é válido para qualquer número par de negações.

4.3.9 Postulado 9

Se os dois membros de uma igualdade forem negados, esta não sofre qualquer alteração, por exemplo,

- $S = A + B \rightarrow \overline{S} = \overline{(A + B)}$
- $S = A \cdot B \rightarrow \overline{S} = \overline{(A \cdot B)}$

4.4 Propriedades booleanas

4.4.1 Propriedade comutativa

- Para a adição $\rightarrow A + B = B + A$
- Para a multiplicação $\rightarrow A \cdot B = B \cdot A$

4.4.2 Propriedade associativa

- Para a adição $\rightarrow A + (B + C) = (A + B) + C$
- Para a multiplicação $\rightarrow A \cdot (B \cdot C) = (A \cdot B) \cdot C$

4.4.3 Propriedade distributiva

- $A \cdot (B + C) = A \cdot B + A \cdot C$
- $(A \cdot B) + C = (A + C) \cdot (B + C)$

4.5 Leis booleanas

4.5.1 Lei da absorção

- $A + A \cdot B = A$

Demonstração:

Evidenciando a variável A, temos, $A + AB = A \cdot (1 + B)$

Porém, pelo postulado 2 temos que $1 + B = 1$, então: $A + AB = A \cdot 1 = A$

- $A \cdot (A + B) = A$

Demonstração:

$$A \cdot (A + B) = (A \cdot A) + (A \cdot B) \rightarrow \text{Pelo postulado 5: } A \cdot A = A$$

$$= A + (A \cdot B) \rightarrow \text{Evidenciando } A \text{ teremos}$$

$$= A (1 + B) \rightarrow \text{Pelo postulado 1: } (1 + B) = 1$$

$$= A (1) = A$$

4.5.2 Lei da dualidade

- $A \cdot B + A \cdot C = (A + B) \cdot (A + C)$

4.5.3 Outras leis importantes

- $A + \bar{A} \cdot B = A + B$
- $(A + \bar{B}) \cdot B = A \cdot B$
- $A \cdot B + A \cdot \bar{B} = A$
- $(A + B) \cdot (A + \bar{B}) = A$
- $A \cdot (A + B) = A$
- $A \cdot (\bar{A} + B) = A \cdot B$
- $A \cdot B + \bar{A} \cdot C = (A + C) \cdot (\bar{A} + B)$

4.6 Princípio da dualidade

Há um princípio da álgebra booleana conhecido como princípio da dualidade que afirma que:

- Para uma expressão booleana qualquer, se trocarmos as operações “E” e as operações “OU” entre si, assim como os valores “0” e “1” entre si, obteremos uma expressão igualmente válida.

Observe exemplos da aplicação do princípio:

- $A + 0 = A \rightarrow A \cdot 1 = A$
- $A + 1 = 1 \rightarrow A \cdot 0 = 0$
- $A + A = A \rightarrow A \cdot A = A$
- $A + \bar{A} = 1 \rightarrow A \cdot \bar{A} = 0$

4.7 Teoremas de De Morgan

Augustus De Morgan, um matemático inglês contemporâneo de Boole, que chegou inclusive a conhecê-lo, propôs dois teoremas baseados no princípio de que é possível mudar o operador sem alterar a expressão, e que hoje representam uma parte significativa na álgebra lógica, ou álgebra booleana.

4.7.1 Primeira lei

A primeira lei de De Morgan afirma que “o complementar da interseção de dois ou mais conjuntos é igual à união dos complementares dos conjuntos iniciais”.

Essa lei aplicada à álgebra booleana pode ser interpretada como: “o complementar de uma operação E é igual à operação OU dos seus complementares.” Ou seja:

- $\overline{A \cdot B} = \overline{A} + \overline{B}$

4.7.2 Segunda lei

A segunda lei de De Morgan afirma que “o complementar da união de dois ou mais conjuntos é igual à interseção dos complementares dos conjuntos iniciais”.

Essa lei aplicada à álgebra booleana pode ser interpretada como: “o complementar de uma operação OU é igual à operação E dos seus complementares”. Ou seja:

- $\overline{A + B} = \overline{A} \cdot \overline{B}$

4.8 Exemplos de simplificação usando a álgebra booleana

a) $A \overline{B} C + \overline{A} B \overline{C} =$

$$= A' (B C + B C')$$

$$= A' (B (C + C'))$$

$$= A' (B (1))$$

$$= A' (B)$$

$$= A' B$$

$$\begin{aligned}
 \text{b) } \bar{A} B C + \bar{A} B \bar{C} + A B \bar{C} &= \\
 &= A C' (B + B') + A' B C' = \\
 &= A C (1) + A' B C' = \\
 &= A C' + A' B C' = \\
 &= C' (A + A' B) \\
 &= C' (A + B)
 \end{aligned}$$

$$\begin{aligned}
 \text{c) } S &= ABC + AC' + AB' \\
 S &= A (BC + C' + B') \\
 S &= A (BC + (CB)') \\
 S &= A (1) \\
 S &= A
 \end{aligned}$$

$$\begin{aligned}
 \text{d) } S &= A'B + A.B' + AB \\
 S &= A'B + A (B' + B) \\
 S &= A'B + A \\
 S &= (A' + A) . (B + A) \\
 S &= 1 . (B + A) \\
 S &= A + B
 \end{aligned}$$

4.9 Simplificação de expressão lógica através do Diagrama de Veitch Karnaugh (Mapa de Karnaugh)

Simplificação de circuitos lógicos através da álgebra de Boole é um método eficiente, mas que necessita de uma boa base matemática para ser aplicada de modo eficaz. Para chegar à expressão simplificada de um modo mais



Para saber mais sobre técnicas de simplificação de expressões lógicas pela álgebra booleana e teorema de De Morgan, além do método da soma de produtos, acesse o site <http://200.19.92.57/wschui/goodbit/Boole.htm>

Diagrama de Veitch Karnaugh

Foi criado por Edward Veitch em 1952 e aperfeiçoado pelo engenheiro de telecomunicações Maurice Karnaugh quando utilizava o diagrama original para simplificar circuitos de telefonia na empresa Bell (Fonte: GOLDEN MAP, 2012).

prático, iremos utilizar preferencialmente o método do **Diagrama de Veitch Karnaugh** ou simplesmente Mapa de Karnaugh. Esse método permite simplificar uma expressão lógica, retirando diretamente da tabela-verdade os possíveis estados de entrada e saída de um circuito lógico, obtendo com isso a expressão booleana simplificada.

4.9.1 Entendendo o Diagrama de Veitch Karnaugh

Antes de começarmos a estudar o Diagrama de Karnaugh vamos entender como ele é montando e como se faz para interpretá-lo de modo a conseguir a expressão reduzida.

Veja por exemplo a tabela-verdade abaixo, na qual os valores de S representam uma possível resposta de saída.

A	B	S
0	0	X
0	1	Y
1	0	Z
1	1	K

Ao montarmos o mapa de Karnaugh para a tabela acima, devemos observar quantas entradas ela possui, e posicionar as saídas de acordo com as entradas. Por exemplo:

No mapa ao lado podemos observar que há duas entradas (A e B), e cada entrada pode ter o valor 0 ou 1. Também podemos ver o posicionamento da saída de cada valor que cada entrada possui. X quando A=0 e B=0; Y quando A=0 e B = 1; Z quando A = 1 e B = 0; e K quando A = 1 e B = 1

A \ B	0	1
0	X	Y
1	Z	K

Após montado o mapa procuraremos as regiões onde a saída assume valor 1, agrupando-os em grupos de 1, 2 (pares), 4 (quadrados) ou 8 (oitavas) elementos.

A \ B	0	1
0	X	Y
1	Z	K

← Região A

↑ Região B

4.9.2 Mapa de Karnaugh para duas variáveis

Como exemplo, tomemos a seguinte tabela-verdade para duas variáveis de entrada:

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

Montando nosso mapa e localizando pares em que teremos saída 1, teremos então:

A \ B	0	1
0	0	1
1	1	1

← Par

← Par

Podemos observar então que a saída assume valor 1 nas regiões (pares) onde $A = 1$ "OU" onde $B = 1$. Sendo assim, podemos dizer que a expressão que representa a tabela-verdade será:

- $S = A + B$

Vamos agora tomar um novo exemplo:

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

Montando o mapa e localizando as saídas 1 teremos:

A \ B	0	1
0	1	0
1	1	0

Par

Nesse exemplo só temos um par formado na região onde (B=0).

Então nossa expressão será representada por:

- $S = \bar{B}$

Observação: Quando uma variável de uma região recebe o valor lógico 0, ele é representado pelo seu complemento da expressão (barra de negação).

4.9.3 Mapa de Karnaugh para três variáveis

O objetivo continua sendo localizar o número de regiões para a saída sendo 1, partindo de grupos de oitavas, quadras, duplas e em último caso, região de um único elemento, podendo uma mesma resposta pertencer a mais de uma região.

Para a montagem do mapa, utiliza-se, por exemplo, a seguinte estrutura:

C \ AB	00	01	11	10
0	Q	W	E	F
1	L	X	Y	Z

Quadra

Dupla
Observe que as saídas L e Z formam uma dupla

Tomemos como exemplo a seguinte expressão:

- $S = \bar{A} \bar{B} \bar{C} + A \bar{B} \bar{C} + A \bar{B} C + A B C + \bar{A} B C$

Vamos montar o mapa preenchendo com 1 as saídas equivalentes aos elementos das expressões acima, e depois localizando as regiões.

C \ AB	00	01	11	10
0	1		1	1
1			1	1

Temos então resposta 1 na região $A = 1$ (quadra), ou na região $B \text{ e } C = 0$ (dupla)

Logo, nossa expressão pode ser reduzida a:

- $S = A + \bar{B} \bar{C}$

Tomemos um novo exemplo:

Seja a tabela-verdade abaixo.

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Montando o mapa e localizando as regiões teremos:

A \ BC	00	01	11	10
0	0	0	0	0
1	1	0	1	1

Temos então uma dupla em $A = 1 \text{ e } B = 1$ e outra quadra em $A = 1 \text{ e } C = 0$

Logo, a expressão será: $S = A.B + A.C$



Para saber mais sobre o diagrama de Veitch Karnaugh, acesse o site http://www.joinville.ifsc.edu.br/~stefano/digital/Digital_Superior_parte5v1.pdf

Resumo

Nesta aula conhecemos os postulados e leis que regem a álgebra booleana. Fizemos reduções das expressões algébricas booleanas. Aprendemos a montar o mapa de Karnaugh e vimos como desenhar circuitos a partir do mapa de Karnaugh.

Atividades de aprendizagem

1. Simplifique as seguintes expressões:

a) $\overline{A}\overline{B}C + \overline{A}BC + ABC$

b) $\overline{A}B\overline{C}D + ABC\overline{D} + A\overline{B}CD$

c) $ABC + A\overline{B}\overline{C}$

d) $ABC + \overline{A}BC + ABC$

2. Prove pela álgebra de Boole a seguinte igualdade:

$$(A + B)(A + C) = A + BC$$

Dica: $A + 1 = 1$, então $1 + B + C = 1$

3. Determine a expressão e o circuito lógico pelo diagrama de Veitch Karnaugh, dadas as seguintes tabelas verdade:

a)

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

b)

A	B	C	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Aula 5 – Circuitos combinacionais

Objetivos

Caracterizar um circuito combinacional.

Conhecer os circuitos codificadores/decodificadores, multiplexadores / demultiplexadores, comparadores e meio somadores/meio subtratores.

Conhecer o funcionamento de cada um dos circuitos.

Aprender a realizar operações de adição e subtração com números binários.

5.1 Introdução

Os circuitos combinacionais são aqueles cujas saídas dependem única e exclusivamente das suas variáveis de entrada; portanto, são circuitos que não possuem dispositivo de memória.

Com o estudo dos circuitos combinacionais, poderemos mais tarde entender melhor o funcionamento de outros circuitos, tais como: somadores, subtratores, codificadores, decodificadores, etc.

5.2 Circuitos combinacionais de duas entradas

Para o processo de criação de um circuito combinacional, costuma-se seguir as seguintes etapas:

- Estudo da situação
- Estudo da tabela-verdade
- Expressão do circuito simplificado
- Representação do circuito

Vamos exemplificar o processo descrito acima com uma situação: instalação de uma controladora que dá acesso a uma impressora em um escritório.

Estudo da situação

Nossa controladora deve obedecer às seguintes características:

- Têm acesso à impressora um funcionário responsável e o patrão.
- O acesso do patrão à impressora é prioritário sobre o acesso do funcionário.

Estudo da tabela-verdade

Para o nosso estudo vamos convencionar:

- Patrão $\rightarrow P$
- Funcionário $\rightarrow F$
- Resposta da solicitação à impressora do patrão $\rightarrow R_p$
- Resposta da solicitação à impressora do funcionário $\rightarrow R_f$

CASO	P	F	R _p	R _f
1	0	0	0	0
2	0	1	0	1
3	1	0	1	0
4	1	1	1	0

Para o caso 1 $\rightarrow$ Ninguém solicita a impressora; então, não há resposta da impressora.

Para o caso 2 $\rightarrow$ Somente o funcionário solicita a impressora; então, a impressora responde a sua solicitação.

Para o caso 3 $\rightarrow$ Somente o patrão solicita a impressora; então, a impressora responde a sua solicitação.

Para o caso 4 $\rightarrow$ Tanto o patrão quanto o funcionário solicitam a impressora; então, a impressora responde à solicitação do patrão, já que ele é prioritário.

Expressão do circuito simplificado

Para a saída R_p teremos então:

P \ F	0	1
0	0	0
1	1	1

$$R_p = P$$

Para a saída Rf teremos então:

P \ F	0	1
0	0	1
1	0	0

$$R_f = \overline{P}F$$

Então a resposta da impressora será sempre uma situação ou outra, o que nos dá:

$$R = P + P.F$$

Representação do circuito (Figura 5.1)



Figura 5.1: Representação do circuito exemplo

Fonte: Elaborada pelos autores

5.3 Circuitos combinacionais de três entradas

Para circuitos combinacionais em geral, não importando o número de entradas, o processo de análise e construção do circuito será sempre o mesmo. Vamos ver um exemplo.

Vamos tomar o exemplo anterior da controladora da impressora, mas desta vez adicionando a figura da secretária do patrão no problema, de tal modo que:

- O comando para a impressora do patrão tem prioridade sobre todos.
- O comando para impressora da secretária tem prioridade apenas sobre o funcionário.

Desse modo teremos:

CASO	P	S	F	Rp	Rs	Rf
1	0	0	0	0	0	0
2	0	0	1	0	0	1
3	0	1	0	0	1	0
4	0	1	1	0	1	0
5	1	0	0	1	0	0
6	1	0	1	1	0	0
7	1	1	0	1	0	0
8	1	1	1	1	0	0

Caso 1 → Nenhum comando para a impressora; então, não há nenhuma resposta.

Caso 2 → Somente há comando para a impressora do funcionário; então, há resposta apenas para o comando dele.

Caso 3 → Há comando apenas para a impressora da secretária; então, há resposta apenas para o comando dela.

Caso 4 → Há comando para a impressora tanto do funcionário como da secretária; como a secretária tem prioridade, a impressora atenderá ao comando dela.

Caso 5, 6, 7 e 8 → Há comando do patrão para a impressora. Independentemente se há ou não solicitações da secretária e do funcionário, a impressora atende ao comando do patrão, já que é prioridade sobre todos os outros.

Resposta do comando do patrão (Rp)

P \ SF	0 0		0 1		1 1		1 0	
	0	0	0	1	1	0	1	0
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1

$$R_p = P$$

Resposta do comando da secretária

P \ SF	0 0	0 1	1 1	1 0
0	0	0	1	1
1	0	0	0	0

$$R_s = \overline{P} \cdot S$$

Resposta do comando do funcionário

P \ SF	0 0	0 1	1 1	1 0
0	0	1	0	0
1	0	0	0	0

$$R_f = \overline{P} \cdot \overline{S} \cdot F$$

Somando as três respostas teremos $\rightarrow R_i = P + \overline{P} \cdot S + \overline{P} \cdot \overline{S} \cdot F$

Cujo circuito será (Figura 5.2):

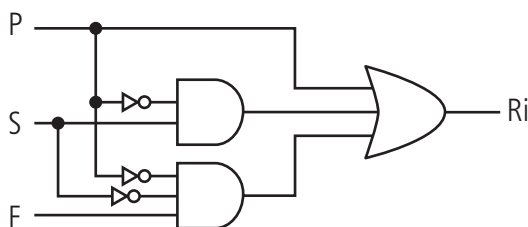


Figura 5.2: Circuito do exemplo

Fonte: Elaborada pelos autores

5.4 Circuitos aritméticos

5.4.1 Meio somador

Circuitos somadores, por definição, são capazes de realizar a soma de dois números em binário. São muito utilizados em circuitos onde haja necessidade de operações aritméticas.

Pelo fato de seu princípio se basear em soma de binários, é necessário que antes entendamos como é realizada essa operação.

Adição binária

ADIÇÃO DE DECIMAIS	ADIÇÃO DE BINÁRIOS
$0 + 0 = 0$	$0 + 0 = 0$
$0 + 1 = 1$	$0 + 1 = 1$
$1 + 0 = 1$	$1 + 0 = 1$
$1 + 1 = 2$	$1 + 1 = 10$

Quando em decimais somamos $1 + 1$, obtemos o valor 2. Em binário acontece o mesmo; porém, é bom lembrar que o número 2 em binário é representado por 10; então, costuma-se dizer que $1 + 1$ em binário é igual a 0 e vai um (*carry*).

Por exemplo:

A soma de 101010 e 111100 será:

$$\begin{array}{r} 111 \quad \leftarrow \text{Vai um (carry)} \\ 101010 \\ + 111100 \\ \hline 1100110 \end{array}$$

Ao montarmos a tabela-verdade para a soma de um *bit* temos:

A	B	SOMA	VAI UM
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Pela tabela-verdade notamos que a resposta SOMA será igual a uma porta OU-EXCLUSIVA, e a saída VAI UM (*carry*) será uma porta E; desse modo, teremos o seguinte circuito (Figura 5.3):

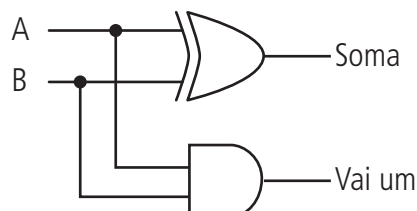


Figura 5.3: Circuito meio somador (*half adder*)

Fonte: Elaborada pelos autores

O circuito indicado é chamado de meio somador ou *half adder*, pois efetua a soma de apenas 1 *bit*. Para somas com mais de um *bit*, será necessário montar um circuito mais complexo, denominado de somador completo ou *full adder*.

5.4.2 Meio subtrator

Para que se possa entender o princípio de funcionamento de um subtrator binário, é necessário ver o conceito de empresta 1. Então, leia os itens abaixo.

- a) $0 - 0 = 0$
- b) $0 - 1 = 1$ (empresta 1)
- c) $1 - 0 = 1$
- d) $1 - 1 = 0$

O empresta 1 em binário é o mesmo conceito utilizado em sistema decimal. Por exemplo, ao efetuar $12 - 4$ teremos:

$$\begin{array}{r} {}^5 6 \quad 12 \\ - \quad 4 \\ \hline 5 \quad 8 \end{array} \quad \left\{ \begin{array}{l} \text{Como não efetuarmos a subtração de } 2 - 4, \text{ já que } 2 \text{ é} \\ \text{menor que } 4, \text{ pedimos } 1 \text{ emprestado ao número do lado,} \\ \text{no caso } 6, \text{ que automaticamente passa a valer } 5. \text{ Agora} \\ \text{teremos } 12 - 8 = 8, \text{ e } 5 - 0 = 5. \text{ Então o resultado passa} \\ \text{a ser } 58. \end{array} \right.$$

Agora vamos ver como isso acontece com números binários. Imagine então a seguinte subtração: $101 - 10$.

$$\begin{array}{r} \text{Terceiro} \quad \text{Segundo} \quad \text{Primeiro} \\ \text{elemento} \quad \text{elemento} \quad \text{elemento} \\ {}^0 1 \quad {}^1 0 \quad 1 \\ - \quad 1 \quad 0 \\ \hline 0 \quad 1 \quad 1 \end{array}$$

No primeiro elemento não há dificuldade, já que $1 - 0$ será igual a 1.

Agora vamos analisar o segundo elemento, onde há $0 - 1$. Nesse caso, por 0 ser menor que 1 ele pede 1 emprestado do terceiro elemento (1) que passa a valer 0. Agora teremos $10 - 1$ que será igual a 1. Atenção: Cuidado para não entender isso como 10 (dez) $- 1$ (um) pois se trata de binário, o certo é 10 (dois) $- 1$ (um) que é igual a 1.

Agora vamos montar nossa tabela-verdade para o nosso meio subtrator.

A	B	DIFERENÇA	EMPRESTA 1
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



Podemos implementar um somador completo a partir de dois meios somadores. Saiba mais sobre o circuito capaz de somar n bits, denominado somador completo ou *full adder* acessando o site http://www.mspc.eng.br/eledig/eldg0710.shtml#somad_compl_1



Assista ao vídeo sobre adição e subtração de números binários acessando o site: <http://www.youtube.com/watch?v=MeragDzjp5M>



Assim como o meio somador, este circuito só efetua a subtração de um *bit* de informação. Para um subtrator completo (*n bits – full subtrator*). Para saber mais sobre o circuito capaz de subtrair dois números de *n bits*, acesse o site <http://curto-circuitos.wikispaces.com/file/view/soma-sub.pdf>

De onde concluímos que a saída DIFERENÇA será um OU-EXCLUSIVO e a saída EMPRESTA 1 acontece quando A for 0 'E' B for 1. Desse modo nosso circuito será (Figura 5.4) a seguir.

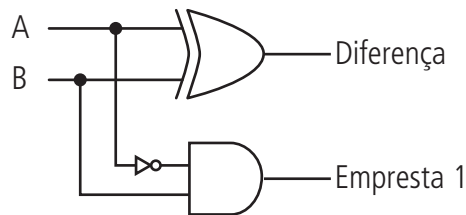


Figura 5.4: Circuito meio subtrator *half subtrator*)

Fonte: Elaborada pelo autores

5.5 Decodificador

5.5.1 Conceitos

Um decodificador é um dispositivo que converte um código qualquer desconhecido em um código conhecido pela máquina ou pelo usuário. Do ponto de vista da eletrônica digital, é um circuito lógico de várias entradas binárias e várias linhas de saída, em que somente uma linha de saída é acionada por vez para cada combinação das linhas de entrada.

Ou seja, um decodificador nada mais é do que um circuito combinacional que ativa uma saída diferente para cada código diferente colocado em suas entradas. Veja a tabela- verdade abaixo:

ENTRADAS		SAÍDAS			
A	B	S0	S1	S2	S3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Observe que a tabela indica uma saída diferente para cada caso de entrada que possui, o que por definição caracteriza um circuito decodificador. Desse modo teremos:

$$S0 = \overline{A} \cdot \overline{B} \quad S1 = \overline{A} \cdot B \quad S2 = A \cdot \overline{B} \quad S3 = A \cdot B$$

Cujo circuito equivalente está mostrado na Figura 5.5 a seguir.

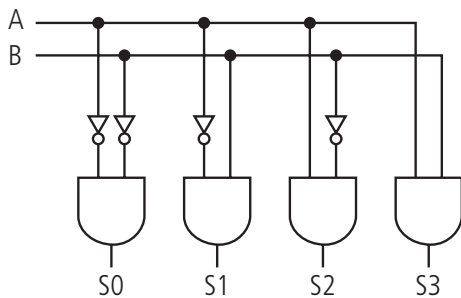


Figura 5.5: Circuito decodificador de duas entradas e quatro saídas

Fonte: Elaborada pelos autores

A Figura 5.6 é a representação gráfica de um decodificador de duas entradas e quatro saídas, como a do nosso exemplo.

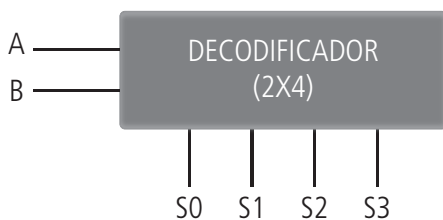


Figura 5.6: Representação gráfica de um decodificador de duas entradas e quatro saídas

Fonte: Elaborada pelos autores

5.6 Codificador

Esse tipo de circuito executa a função inversa do decodificador, ou seja, produz um código diferente em suas saídas para cada entrada ativada.

Vejamos a tabela-verdade abaixo, na qual destacaremos sua particularidade.

S1	S2	S3	S4	A	B
0	0	0	0		
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1		
0	1	0	0	1	0
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	1	1
1	0	0	1		

1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Como se percebe, a tabela-verdade não apresenta todas as 16 saídas possíveis. Acontece que para o nosso estudo de codificadores só nos interessam as entradas em que ocorre uma variável de entrada ativa por vez. Todas as demais possibilidades serão, então, consideradas irrelevantes.



Em alguns casos de circuitos combinacionais teremos situações que nunca acontecem e, portanto, não nos importaremos com os valores das entradas desses casos. Dizemos então que são casos **irrelevantes**, ou seja, tanto faz as entradas terem nível lógico 1 ou nível lógico 0 (CETEBE, 2005).

Assim sendo, reduziremos nossa tabela-verdade para:

S1	S2	S3	S4	A	B
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Verificamos que temos saída em A quando S2 ou S1 forem 1. E saída em B quando S3 ou S1 forem 1, então:

$$A = S1 + S2$$

$$B = S1 + S3$$

Cujo circuito será (Figura 5.7):

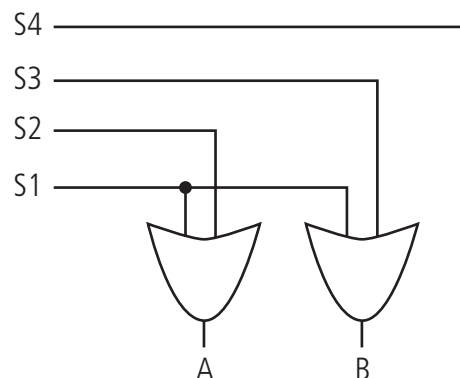


Figura 5.7: Circuito codificador quatro entradas e duas saídas

Fonte: Elaborada pelos autores

5.7 Multiplexadores

Em alguns circuitos digitais pode ser necessário que um elemento receba várias entradas e, de acordo com um processo de seleção, se define qual entrada será utilizada. Um exemplo grosseiro, mas que serve para um entendimento rápido, seria um CD de música com 13 músicas disponíveis para ouvir. Basta o ouvinte selecionar pelas teclas do CD *player* o número da faixa que ele deseja ouvir, para aquela música ser tocada.

Portanto, um multiplexador é um circuito combinacional que tem a finalidade de selecionar, por meio de variáveis de seleção, a entrada que será conectada à saída. A Figura 5.8 ilustra esse conceito. Observe que nela há cinco entradas (A, B, C, D e E), cada uma delas ligada a uma chave seletora 1, 2, 3, 4 e 5, respectivamente, e há ainda três entradas especiais, chamadas variáveis de seleção (X, Y e Z) que são as responsáveis por informar ao circuito qual das cinco chaves deve ser acionada, de modo a ligar a entrada selecionada com a saída única. Por exemplo, as variáveis X, Y e Z determinam que a chave 2 seja acionada, ligando a entrada B à saída. Ou ainda, X, Y e Z determinam que a chave 4 seja acionada, o que faz com que a entrada D seja ligada à saída.

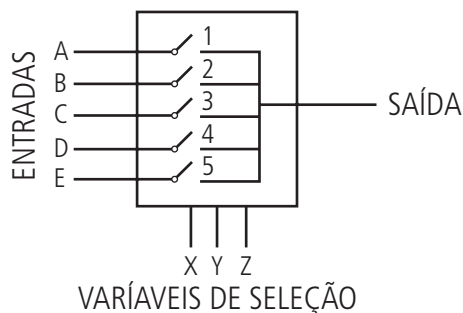


Figura 5.8: Esquema genérico de um multiplexador

Fonte: Elaborada pelos autores

Um multiplexador também recebe a denominação de MUX



5.7.1 Entradas & variáveis de seleção de um MUX

A Figura 5.9 demonstra a representação de um MUX de N entradas (chamado de MUX de N canais) para N variáveis de seleção.

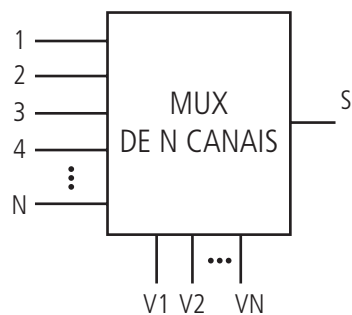


Figura 5.9: MUX de N canais

Fonte: Elaborada pelos autores

A relação matemática entre as entradas de um MUX e as variáveis de seleção se dá pela fórmula:

$N = 2^m \rightarrow$ Onde N representa o número de entradas (canais de entrada) e m o número de variáveis de seleção. Por exemplo: um MUX de duas variáveis de entrada A e B pode ter um total de 4 ($N=2^m \rightarrow N=2^2 = 4$) situações possíveis, quais sejam:

1. $A=0, B=0$
2. $A=0, B=1$
3. $A=1, B=0$
4. $A=1, B=1$

Sendo assim, esse MUX correspondente será de quatro canais (quatro entradas).

Ou ainda, um MUX de três variáveis de entrada, A , B e C , terá então oito canais ($N=2^m \rightarrow 2^3= 8$), já que as possibilidades agora são:

1. $A=0, B=0, C=0$
2. $A=0, B=0, C=1$
3. $A=0, B=1, C=0$
4. $A=0, B=1, C=1$
5. $A=1, B=0, C=0$
6. $A=1, B=0, C=1$
7. $A=1, B=1, C=0$
8. $A=1, B=1, C=1$

5.7.2 MUX de dois canais

Um MUX de dois canais deve ter apenas uma variável de seleção, já que para $N=2^m \rightarrow 2 = 2^m$, então $m=1$.

Sua representação então será (Figura 5.10):

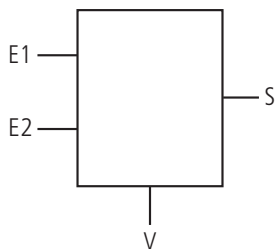


Figura 5.10: MUX de dois canais

Fonte: Elaborada pelos autores

Fazendo um estudo entre as possibilidades que a variável de seleção V pode assumir e as possibilidades de entradas que podem ser ligadas à saída S, teremos:

V	S
0	E1
1	E2

Observando a tabela-verdade e, por conceito de um MUX, sabemos que a saída S pode estar ligada a E1 ou a E2. Sendo que ela está ligada a E1 quando V.E1, e ligada a E2 quando V.E2. Desse modo teremos então a seguinte expressão booleana: $S = V.E1 + V.E2$, o que nos dá o seguinte circuito interno do MUX de dois canais (Figura 5.11):

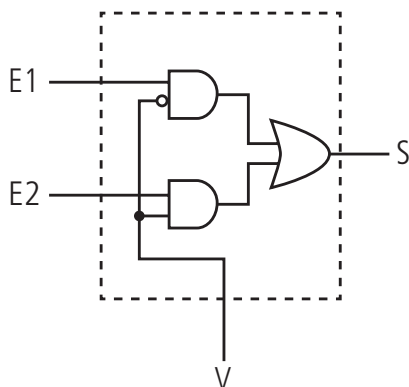


Figura 5.11: Circuito de um MUX de dois canais

Fonte: Elaborada pelos autores

5.7.3 MUX de quatro canais

Seguindo o mesmo roteiro do estudo da seção anterior (5.7.2) temos que para um MUX de três canais (representado na Figura 5.12), já que $N=2^m \rightarrow 4=2^m \rightarrow m=2$.

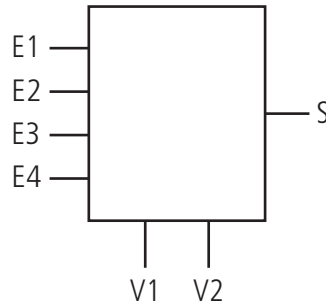


Figura 5.12: MUX de quatro canais

Fonte: Elaborada pelos autores

Já o estudo das possibilidades da variável e das entradas nos dá:

V1	V2	S
0	0	E1
0	1	E2
1	0	E3
1	1	E4

O que resulta na expressão booleana:

$$S = \underline{V1}.\underline{V2}.E1 + \underline{V1}.V2.E2 + V1.\underline{V2}.E3 + V1.V2.E4$$

Tendo seu circuito representado pela Figura 5.13 a seguir.

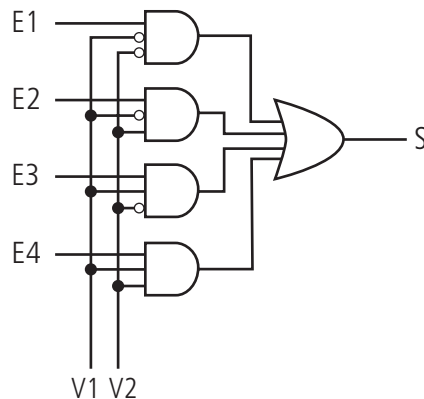


Figura 5.13: Circuito MUX quatro canais

Fonte: Elaborada pelos autores

5.8 Demultiplexadores

Demultiplexadores são circuitos combinacionais que efetuam a operação inversa dos multiplexadores, ou seja, têm em sua entrada apenas um sinal para várias saídas, em que uma delas é selecionada por meio de variáveis de seleção. Sua representação genérica pode ser vista na Figura 5.14 a seguir.

Um demultiplexador também recebe a denominação de DEMUX.

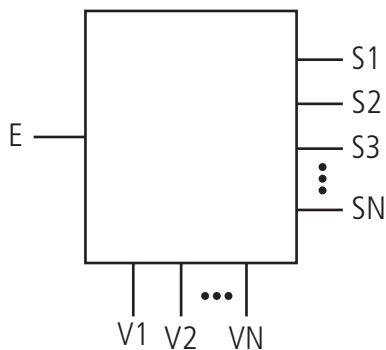


Figura 5.14: Representação genérica de um DEMUX

Fonte: Elaborada pelos autores

5.8.1 Saídas & variáveis de seleção de um DEMUX

A relação matemática que envolve as saídas e as variáveis de seleção de um DEMUX é semelhante à relação entradas x variáveis de seleção de um MUX, ou seja, ela se dá pela fórmula:

$N = 2^m \rightarrow$ Onde N é igual ao número de saídas e m o número de variáveis de seleção.

5.8.2 DEMUX de dois canais

Para um DEMUX de dois canais, teremos uma variável de seleção, já que $N = 2^m \rightarrow 2 = 2^m \rightarrow m = 1$ (Figura 5.15).

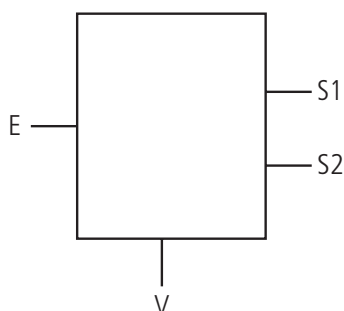


Figura 5.15: DEMUX de dois canais

Fonte: Elaborada pelos autores

Montando a tabela levando em consideração as possibilidades de saídas e da variável de seleção, temos:

V	S1	S2
0	E	0
1	0	E

De modo que S1 assume a saída da entrada quando $V = 0$ e $E=1$; além disso, S2 assume como saída de E quando $V = 1$ e $E = 1$; desse modo, teremos as seguintes expressões booleanas para cada saída.

- $S1 = V.E$
- $S2 = V.E$

Baseados nas expressões booleanas das saídas do DEMUX podemos representar seu circuito como (Figura 5.16):

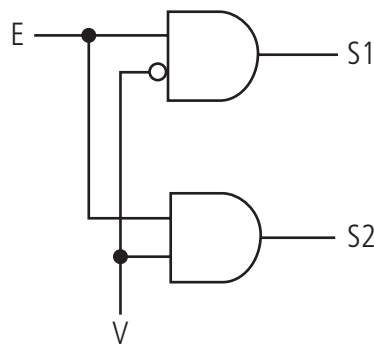


Figura 5.16: Circuito de um DEMUX de dois canais

Fonte: Elaborada pelos autores

5.8.3 DEMUX de quatro canais

Para um DEMUX de quatro canais, teremos duas variáveis de seleção, já que $N = 2^m \rightarrow 4 = 2^m \rightarrow m = 2$ (Figura 5.17).

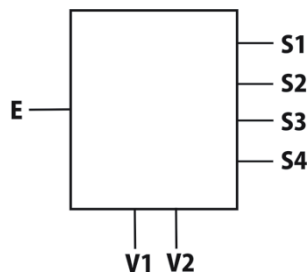


Figura 5.17: DEMUX de quatro canais

Fonte: Elaborada pelos autores.

O estudo das possibilidades das saídas com as possibilidades das variáveis de entrada resulta em:

V1	V2	S1	S2	S3	S4
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

Dessa forma, temos as seguintes expressões booleanas para as saídas:

- $S1 = V1.V2.E$
- $S2 = V1.V2.E$
- $S3 = V1.V2.E$
- $S4 = V1.V2.E$

E seu circuito então é (Figura 5.18):

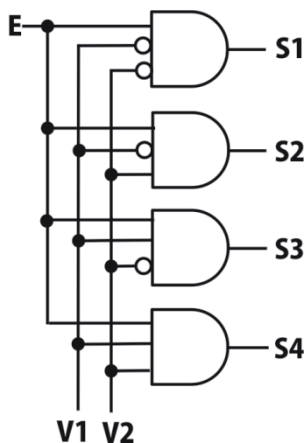


Figura 5.18: Circuito DEMUX de quatro canais

Fonte: Elaborada pelos autores



Para um estudo mais detalhado de circuitos combinacionais visite <http://pt.scribd.com/doc/75552448/Apostila-04-Circuitos-Combinacionais-Vs02>

Resumo

Nesta aula caracterizamos um circuito combinacional, conhecemos os circuitos codificadores/decodificadores, multiplexadores/demultiplexadores, comparadores e meio somadores/meio subtratores e o funcionamento de cada um dos circuitos; aprendemos a realizar operações de adição e subtração com números binários.

Atividades de aprendizagem

1. Baseado no texto explicativo da aula, descreva os seguintes componentes:
 - a) Meio somador
 - b) Multiplexador
 - c) Codificador

2. Um funcionário da Escola Técnica de Informática de Pirapora, quando envia um documento para impressão, tem a opção de realizá-la usando uma impressora jato de tinta ou uma impressora a *laser*. Crie o circuito combinacional que faça o controle do comando de impressão para o funcionário.

3. Efetue as seguintes operações matemáticas com números binários:
 - a) $10101 + 1100$
 - b) $110011 + 10101$
 - c) $111000 - 10101$
 - d) $110011 - 1101$

Aula 6 – Flip-flops

Objetivos

Distinguir circuito sequencial de circuito combinatório.

Compreender o funcionamento do *flip-flop*.

Conhecer diferentes tipos de *flip-flops*.

Desenhar o circuito interno dos diferentes *flip-flops*.

6.1 Introdução

Na aula passada estudamos os circuitos combinacionais que são a base da lógica combinacional. Apesar de serem extremamente práticos no sentido de cumprir seus objetivos, esses circuitos apresentam uma falha grave quando pensamos em computação, que seria falta de estado de memória.

A base de qualquer sistema computacional está vinculada à concepção do uso de memória. Seria impossível digitarmos qualquer coisa, vermos qualquer vídeo, ouvirmos qualquer música, visitarmos qualquer página na internet, enfim, realizar qualquer atividade usando nossos computadores sem o uso de memória.

O campo da eletrônica digital que estuda a criação e o uso de memória é chamado de lógica digital sequencial.

Como já vimos, os circuitos combinatórios nos permitem criar funções como decodificação, soma, entre outras. Porém, funções que dependem de tempo e memorização, por exemplo, simplesmente não podem ser implementadas com eles.

É nesse aspecto que os circuitos sequenciais apresentam vantagem, já que nesses circuitos o valor de uma saída depende não somente dos estados das entradas, mas também do estado anterior, isto é, o valor que a saída tinha antes da aplicação da combinação de valores nas entradas.



Para saber mais sobre a diferença entre a lógica sequencial e combinacional, acesse o site: http://pt.wikipedia.org/wiki/L%C3%B3gica_combinacional

Os circuitos sequenciais são, portanto, os circuitos cujos estados de saída dependem das suas entradas e/ou dos estados anteriores armazenados, e são normalmente circuitos pulsados, ou seja, utilizam-se de um *clock*.

6.2 Conceito de memória

Imagine um circuito somador que efetue a soma de dois números quaisquer. Esse circuito, ao receber as informações sobre os números a serem somados, efetuará a soma e enviará a resposta por meio de suas saídas. Mas e após isso? Se as entradas do circuito forem “desativadas”, o que acontecerá com o resultado obtido? Simplesmente ele se perderá.

Para que a informação fique retida, é necessário um novo circuito capaz de reter tal informação. Dessa forma, podemos definir como memória todo dispositivo que permite a perpetuação de uma informação por um período de tempo (Figura 6.1)

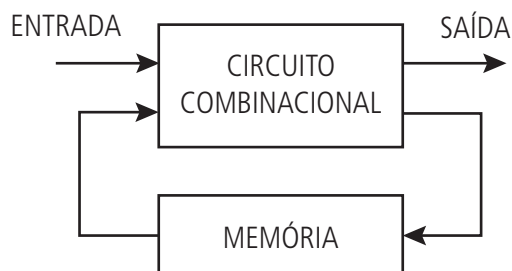


Figura 6.1: Esquema de atuação da memória

Fonte: Elaborada pelos autores

6.3 Bloco Elementar (*flip-flop*)

O bloco elementar da lógica sequencial é conhecido pelo seu nome em inglês, *flip-flop*. O *flip-flop* é um bloco que contém (Figura 6.2):

- duas entradas principais, A e B;
- uma entrada de controle (*clock*), CK;
- duas saídas complementares: Q e Q̄;
- dois componentes opcionais: a) uma entrada de pré-ajuste (*preset*), PR, e b) uma entrada de apagamento (*clear* ou *reset*).

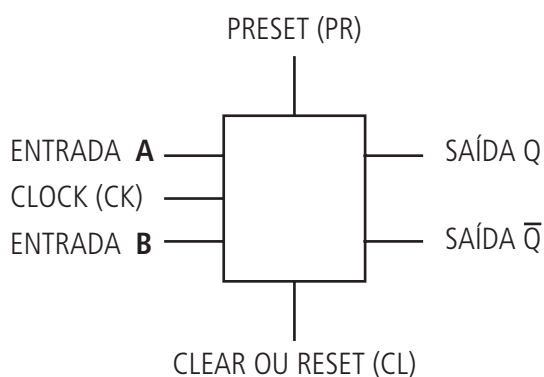


Figura 6.2: Flip-flop básico

Fonte: Elaborada pelos autores

As entradas de controle, pré-ajuste e apagamento serão mencionadas pelos seus nomes em inglês por ser prática usual da área.



Existem vários tipos de *flip-flops* (RS, JK, T e D, por exemplo), cuja distinção se faz pelas letras que representam as entradas A e B.

O *flip-flop* é um circuito lógico que pode armazenar um *bit* de informação. Por isso pode ser considerado uma célula de memória. Assim como as portas lógicas, os *flip-flops* são considerados blocos básicos de um sistema digital.

6.4 Flip-flop RS (Reset – Set) básico

O circuito *flip-flop* RS básico com portas NAND e mostrado na Figura 6.3:

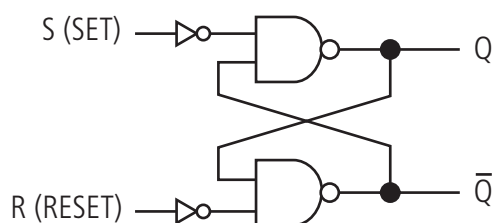


Figura 6.3: Circuito básico flip-flop RS

Fonte: Elaborada pelos autores

O circuito da Figura 6.3 mostra:

- Duas entradas: ESTABELECER (S – SET) e RESTABELECER (R – RESET).
- A saída Q é considerada a saída NORMAL e a saída $\bar{Q}$ é o seu complemento.

A tabela-verdade abaixo define a operação do *flip-flop* RS.

OPERAÇÃO	ENTRADAS		SAÍDAS	
	S	R	Q	$\overline{Q}$
Proibido	0	0	1	1
Set	0	1	1	0
Reset	1	0	0	1
Hold	1	1	Não altera	



Há um caso específico, quando S e R assumem valor 0, que apresenta saídas iguais a 1, o que por definição deve ser um caso impossível, já que Q e seu complemento não podem assumir o mesmo valor.

Como exemplo de circuito integrado (CI) utilizando o conceito de *flip-flop* RS básico, temos a série CI 7400, que foi a primeira série de CI largamente usada em computadores (*mainframes* principalmente) a partir das décadas de 1960 e 1970, e por décadas serviu de padrão para a construção de milhares de outros CIs, entre eles o 7402 com quatro portas NOR, o 7486 com quatro portas XOR e o 74266 com quatro portas XNOR, por exemplo (Figura 6.4).

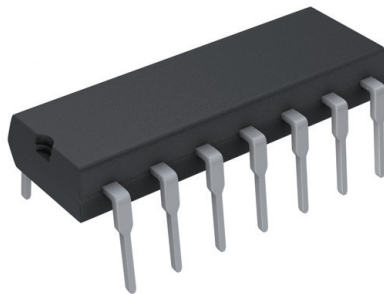


Figura 6.4: Circuito integrado da série 7400

Fonte: <http://www.westfloridacomponents.com/7400icguide.html>

A configuração de um CI 7400 de 14 pinos, visto na Figura 6.5, contém quatro portas NAND de duas entradas. Cada porta usa dois pinos para entrada e um pino para saída, sendo os dois contatos adicionais usados para o suprimento de energia (+5 V) e a conexão do terra (*ground-gnd*).

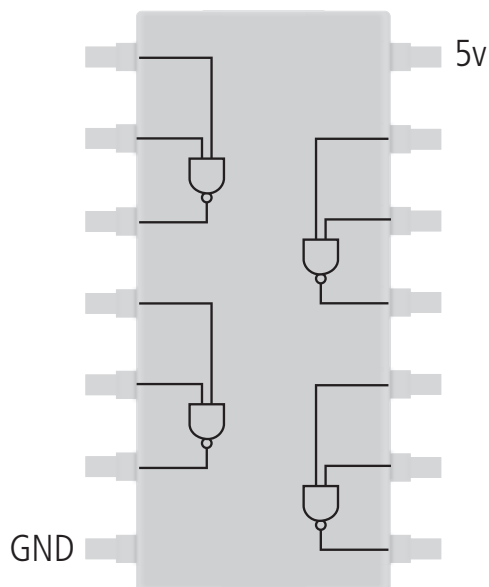


Figura 6.5: Circuito interno de um CI 4700 com portas NAND

Fonte: Elaborada pelos autores

6.5 Sinais de *clock* e *flip-flops* síncronos

Os sistemas digitais podem operar de forma síncrona ou assíncrona, ou seja, recebendo ou não um sinal de tempo. Nos sistemas assíncronos (que não utilizam tempo) as saídas mudam de nível lógico sempre que há mudança em suas entradas (em uma ou nas duas). Já nos sistemas síncronos (que utilizam tempo) as saídas são alteradas de acordo com um instante de tempo determinado por um circuito chamado *clock*.

Esse sistema altera o valor da entrada lógica no circuito de acordo com uma frequência predeterminada. Por exemplo, um *clock* de 1MHz oscila entre os valores 1 e 0 à razão de um milhão de vezes a cada segundo.

A Figura 6.6 representa um sinal de um circuito *clock* genérico. Note que seu valor se altera de 0 a 1 repetidamente a cada instante constante de tempo, mantendo-se com um determinado valor por um instante.

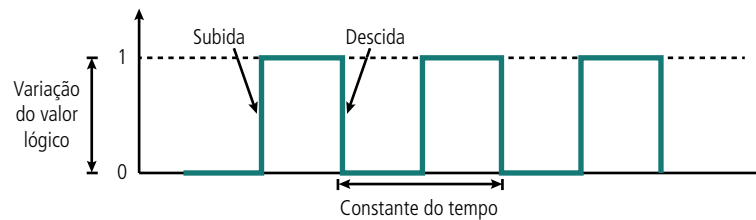


Figura 6.6: Sinal de um circuito clock

Fonte: Elaborada pelos autores

Desse modo, quando um *clock* (CLK) é inserido em um *flip-flop* como uma entrada a mais em seu circuito, suas saídas estarão sujeitas aos valores que ele enviar, ou seja, as saídas do *flip-flop* estão prontas para mudar de estado sempre que ocorre transição no sinal do CLK.

6.6 Flip-Flop RS síncrono

Um *flip-flop* RS síncrono é, por definição, um circuito RS padrão com uma entrada adicional para receber o valor lógico do *clock* (Figura 6.7).

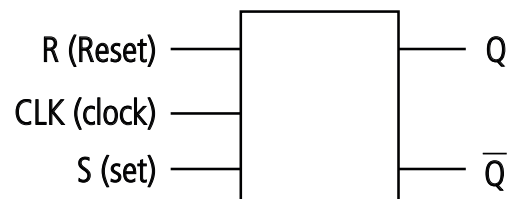


Figura 6.7: Representação de um flip-flop RS síncrono

Fonte: Elaborada pelos autores

Utilizando o circuito RS criado anteriormente com portas NAND e adicionando o circuito do *clock*, o circuito RS síncrono será representado como mostrado na Figura 6.8.

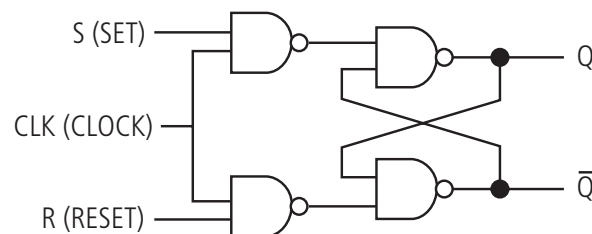


Figura 6.8: Circuito RS síncrono

Fonte: Elaborada pelos autores

A tabela-verdade do circuito RS síncrono então será:

OPERAÇÃO	ENTRADAS			SAÍDAS	
	CLK	S	R	Q	$\bar{Q}$
Hold	1 	0	0	Não altera	
Reset	1 	0	1	0	1
Set	1 	1	0	1	0
Proibido	1 	1	1	1	1

Podemos observar que o *flip-flop* RS síncrono tem um estado proibido quando $R = 1$, $S = 1$ e $CLK = 1$, já que as saídas de Q e do seu complemento não podem assumir o mesmo estado.



6.7 Flip-Flop JK

O *flip-flop* JK tem a propriedade de **comutar** sua saída sempre que as entradas J e K assumem valor 1 (Figura 6.9).

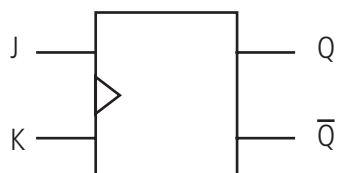


Figura 6.9: Representação de um *flip-flop* JK (o triângulo simboliza entrada *clock*)

Fonte: Elaborada pelos autores

Sua construção é feita tomando-se um *flip-flop* RS como base e adicionando uma realimentação das suas saídas, conforme pode ser visto na Figura 6.10 a seguir.

A-Z

Comutar

Permutar, substituir, trocar. Em eletricidade é usado quando há referência em alterar o sentido da corrente.

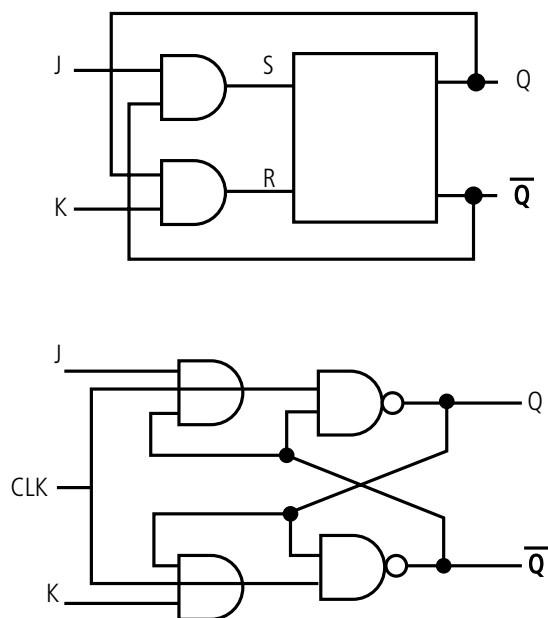


Figura 6.10: Diagrama (sem clock) e circuito (com clock) de um flip-flop JK

Fonte: Elaborada pelos autores

A tabela-verdade para o *flip-flop* JK é então:

OPERAÇÃO	SAÍDAS		SAÍDA
	J	K	Q
Hold	0	0	Qanterior (mantém)
Reset	0	1	0
Set	1	0	1
Comuta	1	1	$\bar{Q}$ anterior (troca)



Para a tabela-verdade do *flip-flop* JK é necessária uma interpretação diferente da que estamos acostumados a usar até agora, já que a saída Q não necessariamente mostra valores lógicos 1 e 0. Observe que quando J e K assumem valor 0 a resposta Q deve permanecer no estado em que se encontra, ou seja, não se altera. Já quando J e K assumem valor lógico 1, Q passará a ter o valor do seu complemento anterior, ou seja, muda de valor.

O *flip-flop* J-K recebeu este nome em homenagem a Jack Kilby, o homem que, em 1958, inventou o circuito integrado, trabalho agraciado posteriormente com o Prêmio Nobel em Física de 2000. “*Jump-Kill*” também é utilizado como analogia a “*set-reset*” (SENGER, 2012).

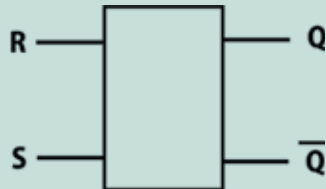
Resumo

Nesta aula vimos o que distingue um circuito sequencial de um circuito combinatório. Depois vimos o funcionamento do *flip-flop*. A partir daí conhecemos tipos diferentes de *flip-flops* aprendendo a desenhar o seu circuito interno.

Atividades de aprendizagem

1. Determine o valor lógico da saída Q para os estados a, b e c abaixo:

ESTADO	R	S
a	0	1
b	1	0
c	1	1



2. Os seguintes dados em série são aplicados ao *flip-flop* através das portas AND como indicado na Figura. Determine os dados em série resultantes que aparecem na saída Q. Existe um pulso de *clock* para cada tempo de *bit*. Considere a saída Q inicialmente em 0. Os *bits* mais à direita são aplicados primeiro.
3. O que difere um *flip-flop* síncrono de um assíncrono?



Para conhecer mais sobre *flip-flop* RS, JK e outros tipos de *flip-flops*, acesse <http://www.joinville.ifsc.edu.br/~mtaques/Digital/ELD2.prn.pdf> ou http://www.corradi.junior.nom.br/rev_flip_flop.pdf

Aula 7 – Circuitos sequenciais

Objetivos

Compreender o princípio de funcionamento de um contador.

Compreender o princípio de funcionamento de um registrador de deslocamento e as suas características.

Compreender a diferença entre transmissão de dados em série e a em paralela.

Conhecer os tipos de registradores quanto à entrada ou saída serial ou paralelas.

Compreender o princípio de funcionamento de memórias baseadas em *.flip-flops*.

7.1 Introdução

Conhecemos na aula passada os *.flip-flops* e por meio do seu estudo vimos que cada um tem seu funcionamento específico para o tratamento de um *bit* de informação.

Quando desejamos realizar um tratamento de mais de um *bit* de informação, podemos agrupar alguns *.flip-flops* para criar novas aplicações específicas. Esse agrupamento é comumente chamado de subsistemas. Juntos, os subsistemas formam sistemas maiores e mais complexos como, por exemplo, o computador.

Três subsistemas são fundamentais: o registrador, o contador e a memória.

7.2 Registradores

Um registrador tem a função de manipular e armazenar dados, na forma de *bits*.

A forma como os *.bits* são transmitidos pode ser dividida em duas categorias: modo serial e modo paralelo. No modo serial, uma sequência de *bits* é transmitida em série, ou seja, um *bit* de cada vez, seguindo um único caminho. Já no paralelo, a sequência de *bits* é transmitida simultaneamente, utilizando mais de um caminho (Figura 7.1).

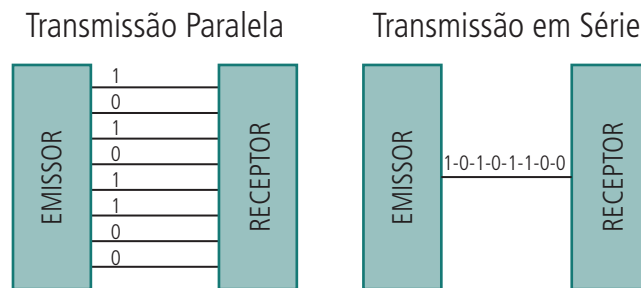


Figura 7.1: Transmissões em paralelo e em série

Fonte: Elaborada pelos autores

As transmissões (entrada ou saída) podem ser divididas em: da direita para esquerda, ou da esquerda para a direita.

Os registradores são classificados de acordo com o modo que operam a entrada e saída dos *.bits*.

7.2.1 Registrador série-série

Esse registrador recebe dados em forma serial e os transmite também em série (Figura 7.2).



Figura 7.2: Entrada e saída de dados de um registrador série-série

Fonte: Elaborada pelos autores



Um registrador série também é chamado de registrador de deslocamento.

7.2.2 Registrador série-paralelo

Esse registrador recebe os dados em série e transmite em paralelo (Figura 7.3).

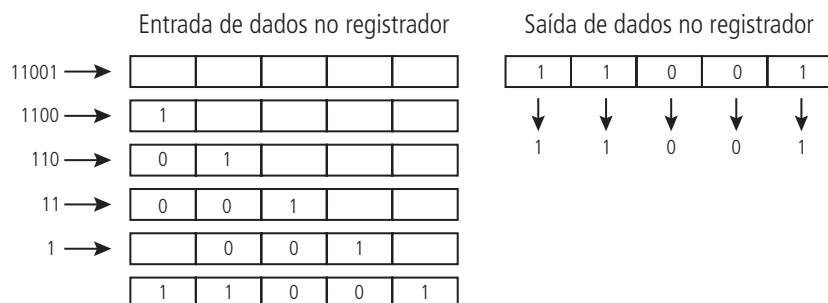


Figura 7.3: Entrada e saída de dados de um registrador série-paralelo

Fonte: Elaborada pelos autores

7.2.3 Registrador paralelo-série

Esse registrador recebe os dados em paralelo e transmite em série (Figura 7.4).

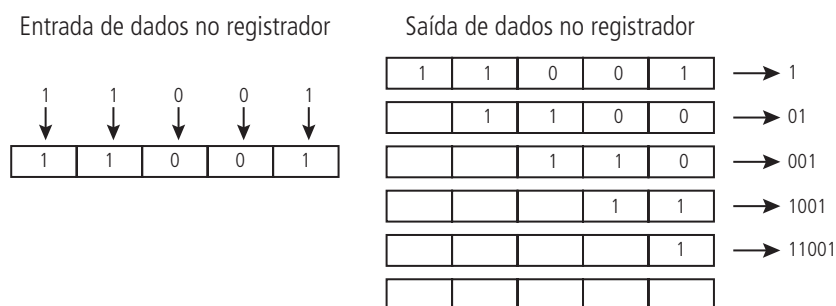


Figura 7.4 Entrada e saída de dados de um registrador paralelo-série

Fonte: Elaborada pelos autores

7.2.4 Registrador paralelo-paralelo

Esse registrador recebe e transmite dados em paralelo (Figura 7.5).

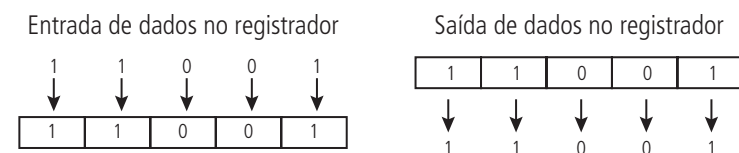


Figura 7.5: Entrada e saída de dados de um registrador paralelo-paralelo

Fonte: Elaborada pelos autores

7.2.5 Modo paralelo & modo serial de transmissão de um registrador

Antes de estudarmos os prós e contras de cada modo de transmissão, é importante que seja entendido que o número de *bits* que um registrador é capaz de armazenar é igual ao número de *flip-flops* que ele contém. Isso significa que nesse aspecto todos os registradores se igualam.

Porém, imagine um dado que contenha oito *bits* de informação. O armazenador precisará então possuir oito *flip-flops*. Para um registrador em série serão necessários oito ciclos para completar toda a transmissão (entrada ou saída) da informação, enquanto no paralelo, por enviar simultaneamente todos os oito *bits*, só um ciclo se faz necessário. Em contrapartida, para o registrador em série só há necessidade de se criar um único circuito para a transmissão (entrada ou saída). Já o registrador paralelo necessita de oito circuitos.

7.2.6 Circuito de um registrador de deslocamento (entrada série)

A Figura 7.6 é a representação simplificada de um registrador de deslocamento da esquerda para direita de quatro *bits*. Nela podemos observar quatro *flip-flops* (ff1, ff2, ff3, ff4).

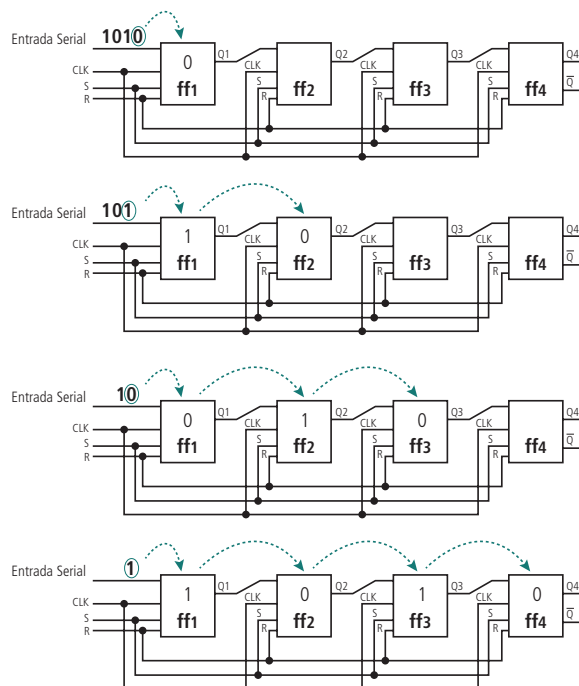


Figura 7.6: Circuito simplificado de um registrador da esquerda para direita

Fonte: Elaborada pelos autores

A Figura 7.6 mostra a sequência de entrada pela entrada serial do dado 1010 (quatro *bits*). Estando S (*set*) com valor 1 e R (*reset*) com valor 0 e quando a entrada do primeiro dado (*bit*) 0 é recebido, junto com o sinal de *clock*, esse valor é armazenado em ff1 e transferido para a saída Q1. Quando o segundo dado (1) é fornecido, junto com um novo sinal de *clock*, o valor inicial (0) que está em Q1 é transferido para o ff2 dando então saída em Q2, deixando ff1 livre para receber e armazenar o novo dado, que passa a ter uma nova saída Q1 igual ao novo dado recebido. E processo vai seguindo até que todos os *flip-flops* estejam armazenando seus respectivos valores.

Quando *cset* e *creset* assumem valor 1, todos os *flip-flops* do armazenador passam a ter valor 0 armazenado (Figura 7.7).

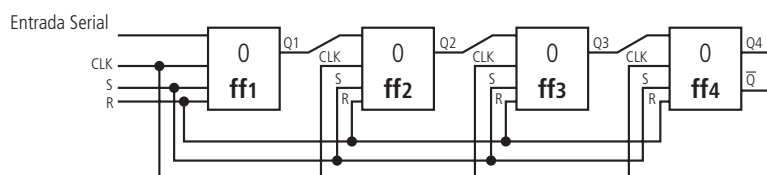


Figura 7.7: Registrador série com set=1 e reset=1

Fonte: Elaborada pelos autores

7.2.7 Circuito de um registrador de deslocamento (paralelo-paralelo)

A Figura 7.8 é a representação simplificada de um registrador de deslocamento de entrada e saída paralelas de quatro *bits*. Nela podemos observar quatro *flip-flops*: ff1, ff2, ff3, ff.

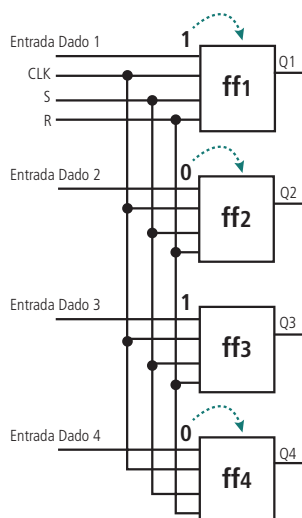


Figura 7.8: Registrador paralelo-paralelo

Fonte: Elaborada pelos autores

Podemos observar que para o mesmo dado anterior (1010) ser recebido no registrador será necessário apenas um pulso no *clock*, enquanto cada *flip-flop* recebe seu valor de dado (*bit*) correspondente, dando saída exatamente igual ao valor recebido.

Nesse registrador, os *flip-flops* dão saída 0 simultaneamente quando *Set* = 1 e *Reset* = 0 (Figura 7.9).

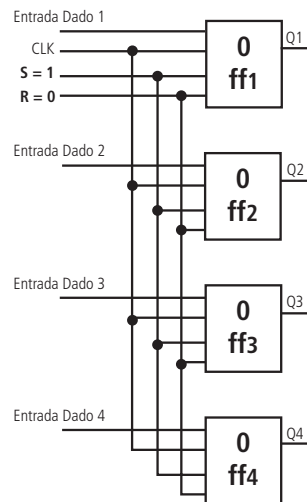


Figura 7.9: Registrador paralelo-paralelo com $cset=1$ e $creset=0$
Fonte: Elaborada pelos autores

7.2.8 Circuito de um registrador de deslocamento (série-paralelo)

A Figura 7.10 é a representação simplificada de um registrador de deslocamento de entrada série e saída paralela de quatro *bits*. Nela podemos observar quatro *flip-flops*: ff1, ff2, ff3, ff4.

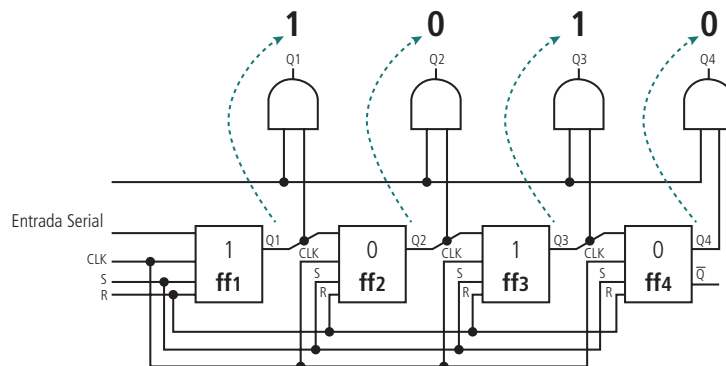


Figura 7.10: Circuito registrador série-paralelo
Fonte: Elaborada pelos autores

Como pode ser visto na Figura 7.10, o circuito de um registrador série-paralelo é o mesmo mostrado na seção 7.2.6 (Circuito de um registrador de deslocamento – entrada série) com o acréscimo de uma linha nova chamada Ativador de saída (*Output Enable – OE*).

O processo de recebimento do dado (1010) será então idêntico ao mostrado anteriormente (seção 7.2.6).

A função do OE é garantir que os dados (*bits*) só sejam transferidos quando todos os *bits* já estiverem com o registro efetuado; em outras palavras, quando todos os *flip-flops* estiverem dando saída a um *bit*. Nesse instante o circuito OE assume valor 1 que ativará quatro portas E, cada uma recebendo além do próprio OE o valor registrado nas respectivas saídas dos *flip-flops*. Desse modo, cada porta E dará saída exatamente igual AL valor Q de cada *flip-flop*.

7.2.9 Circuito de um registrador de deslocamento (paralelo-série)

A Figura 7.11 é a representação simplificada de um registrador de deslocamento de entrada paralela e saída série de quatro *bits*. Nela podemos observar quatro *flip-flops*: ff1, ff2, ff3, ff4.

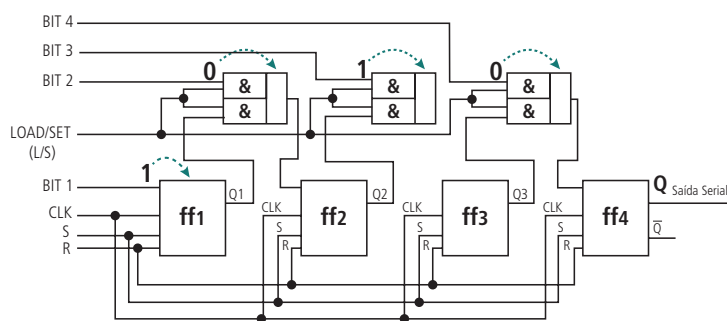


Figura 7.11: Registrador paralelo-série

Fonte: Elaborada pelos autores

No circuito da Figura 7.11 verificamos a existência de uma entrada LOAD/SET (carregar). Quando essa entrada assume valor 1, o registrador indicado na figura recebe o dado (quatro *bits*) pelas entradas BIT1, BIT2, BIT3 e BIT4 simultaneamente, quando o *flip-flop* equivalente recebe e repassa para sua saída Q. Após o carregamento dos quatro *flip-flops*, a entrada LOAD assume valor 0, passando então o circuito a operar como um registrador em série, quando a cada sinal do *clock* os *bits* vão sendo repassados (da esquerda para a direita) até a saída serial no *flip-flop*4.

7.2.10 Exemplos de registradores

Existem centenas de circuitos integrados (CIs) com atuação de registradores. Apenas para exemplificar citaremos três deles.

- a) 7495: Registrador de quatro *bits* (da esquerda para direita) com entrada e saída em paralelo (Figura 7.12)

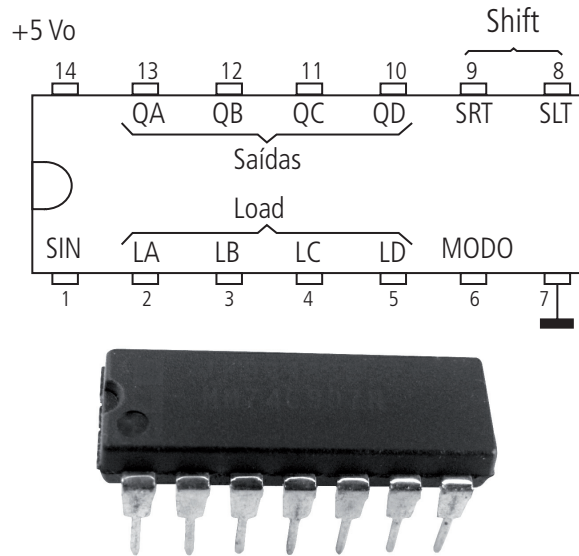


Figura 7.12: CI 7495

Fonte: Oxygen eletronic

- b) 74165: Registrador de oito *bits* com entrada paralela e saída serial (Figura 7.13)

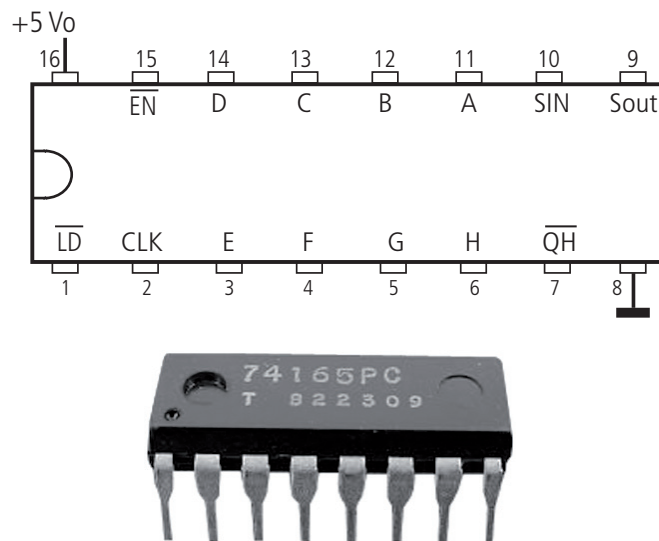


Figura 7.13: CI 74165

Fonte: Oxygen eletronic

- c) D4040: Registrador de oito *bits* com entrada paralela e saída serial (Figura 7.14)

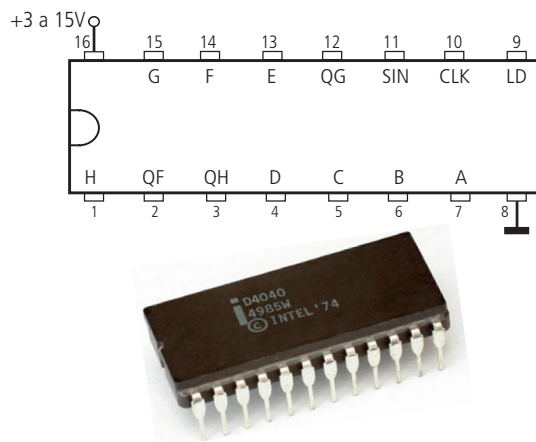


Figura 7.14 - CI D4040

Fonte: CPUWorld.com

7.3 Contadores

Em alguns circuitos digitais é necessário criar um circuito capaz de contar quantas vezes determinado evento se repetiu. Esse circuito específico pode ser obtido por meio de uma interligação específica de *flip-flops*. O circuito contador se caracteriza então como um subsistema sequencial capaz de contar, segundo uma determinada sequência, o número de pulsos que recebe em sua entrada.

7.3.1 Características dos contadores

Se levarmos em consideração a forma como os contadores atuam, ou ainda, o tipo de contagem que efetuam, podemos caracterizá-los como:

- a) **Tipo de controle:** síncronos e assíncronos – quando a atuação depende ou não de um *clock*.
- b) **Tipo de contagem:** progressivos (*up*) ou regressivos (*down*).
- c) **Tipo de código:** binário, hexadecimal, octal, decimal, **código de Gray**, etc.
- d) **Número máximo de contagem:** o número máximo em que o contador pode atuar.
- e) **Funcionamento livre ou autoparada:** quando seu funcionamento é autônomo, ou sua funcionalidade recebe um comando de encerramento específico.

A-Z

Código de Gray

É um sistema de código binário inventado por Frank Gray. O código é não ponderado quando de um número para outro apenas um *cbit* varia. Esse sistema de codificação surgiu quando os circuitos lógicos digitais se realizavam com válvulas termoiônicas e dispositivos eletromecânicos. O uso do código de Gray garante que qualquer mudança variaria apenas um *cbit*. Atualmente o código de Gray é utilizado em sistemas sequenciais mediante o uso dos Mapas de Karnaugh (BRAGA, 2002)

7.3.2 Contador de dois *bits* binário assíncrono

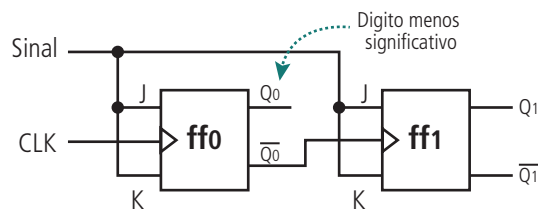


Figura 7.15: Circuito contado assíncrono de dois bits

Fonte: Elaborada pelos autores

Analisando o circuito acima (Figura 7.15) podemos verificar:

- a) por ser um contador de dois *bits*, usa-se dois *flip-flops*;
- b) os *flip-flops* são do tipo JK;
- c) os *flip-flops* trabalham de forma assíncrona, ou seja, em momentos diferentes já que as suas entradas de *clocks* são diferentes;
- d) o *clock* é aplicado apenas no primeiro *flip-flop*;
- e) os *flip-flops* são interligados de forma que, à exceção do primeiro, seus *clocks* recebem sinal proveniente do complemento da saída do seu antecessor;
- f) o *flip-flop* que recebe o sinal do *clock* será o correspondente ao dígito menos significativo do número gerado.

Quanto ao funcionamento do circuito temos:

1. Considerando o valor lógico da entrada igual a 1, a parte positiva do CLK (pulso de *clock*) faz com que a saída Q0 do ff0 vá para o nível ALTO (1). Na parte nula (0) do *clock*, não há efeito sobre o ff.
2. No segundo pulso do *clock* (parte alta), o valor de Q0 permutará (propriedade do *flip-flop* JK) assumindo valor lógico 0, fazendo com que seu complemento assumira valor 1, que por estar ligado ao *clock* do ff1, fará com que a saída Q1 assumira valor 1. Durante a parte nula do pulso, nada se altera.
3. No terceiro pulso do *clock*, durante a parte alta, o ff0 permutará seu valor, passando de 0 para 1, e, por conseguinte, seu complemento assume valor 0, que estando ligado ao *clock* do ff1 não modificará sua saída.

4. Finalmente, na parte alta do pulso 4 o ff0 novamente permutará seu valor de 1 para 0, e seu complemento de 0 para 1, que estando ligado ao *clock* de ff2 fará com que permute seu estado atual de 1 para 0. Nesse caso, todos os dois *flip-flops* retornam ao valor inicial 0. Quando isso ocorre, dizemos que o contador reciclou.

A Figura 7.16 mostra o diagrama de temporização do circuito em estudo.

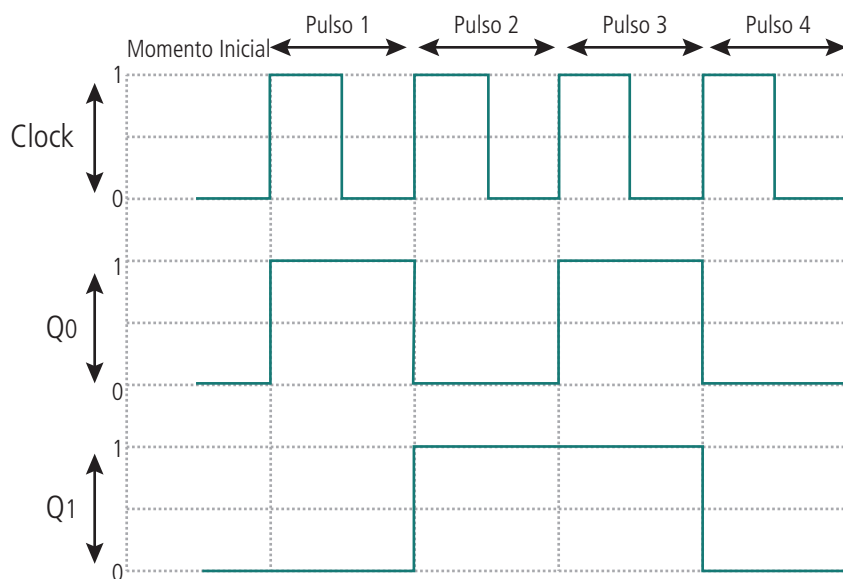


Figura 7.16: Diagrama de temporização do contador binário de dois bits com portas JK

Fonte: Elaborada pelos autores

Agora, montando a tabela-verdade do circuito, teremos:

CCLOCK	Q1	Q0
Momento inicial	0	0
Pulso 1	0	1
Pulso 2	1	0
Pulso 3	1	1
Pulso 4 (recicla)	0	0

Observe bem os valores das saídas Q1 e Q0, nessa ordem, na tabela-verdade e compare com os valores dos números binários abaixo:

- 002 = 010
- 012 = 110
- 102 = 210
- 112 = 310

Uma característica importante que podemos observar analisando a tabela-verdade do circuito contador em estudo é que: o número máximo que esse contador poderá contar está diretamente ligado ao número de *bits* que ele suporta, ou seja, o número de *flip-flops* que ele utiliza. No nosso exemplo temos um contador de dois *bits* (dois *flip-flops*) que nos fornece uma contagem de quatro números (00, 01, 10, 11; ou 0, 1, 2, 3).



Chamamos de **MÓDULO** a quantidade total de números que um circuito contador pode fornecer. E seu cálculo se dá pela fórmula $MÓDULO = 2^n$, onde n é o valor de *bits* (*flip-flops*) que ele utiliza.

O maior número fornecido por um contador será sempre seu módulo menos 1. Por exemplo, no nosso circuito de dois *bits*, seu módulo será $2^2 = 4$. Já o número máximo fornecido será $4 - 1 = 3$. Quando o número fornecido se iguala ao módulo, o contador então recicla, em outras palavras, retorna ao valor 0 reiniciando a contagem.

7.3.3 Contador de três *bits* binário assíncrono

Seguindo a mesma linha de estudo do tópico anterior, teremos (Figuras 7.17 e 7.18):

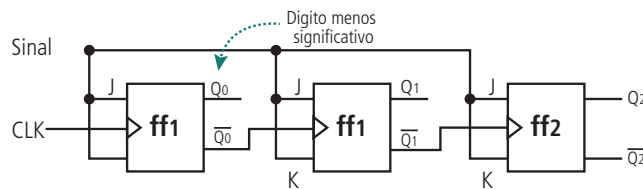


Figura 7.17: Circuito contador binário de três *bits*

Fonte: Elaborada pelos autores

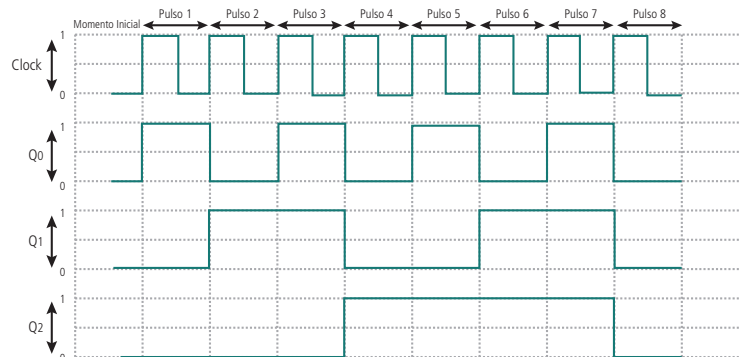


Figura 7.18: Diagrama de temporização do contador binário de três *bits* com portas JK

Fonte: Elaborada pelos autores

TABELA-VERDADE			
CCLOCK	Q2	Q1	Q0
Momento Inicial	0	0	0
Pulso 1	0	0	1
Pulso 2	0	1	0
Pulso 3	0	1	1
Pulso 4	1	0	0
Pulso 5	1	0	1
Pulso 6	1	1	0
Pulso 7	1	1	1
Pulso 8 (recicla)	0	0	0

Módulo: número de *bits* = 3, então Módulo = $2^3 \rightarrow 8$.

Número máximo a ser mostrado: $8 - 1 = 7$, ou 1112.

7.3.4 Contador assíncrono programável

Um contador digital pode ser programado por construção para mostrar uma contagem inferior ao seu número máximo. Por exemplo, um contador de quatro *bits* tem como número máximo 16, mas pode ser programado para registrar apenas até o 12, e quando chegar em 13 efetuar reciclagem.

Vamos entender como isso ocorre acompanhando o seguinte exemplo: um contador digital binário assíncrono de três *bits* deve contar até 5, assumindo a reciclagem quando a contagem atingir 6.

Bem, se o contador é de três *bits*, seu módulo será $2^3 = 8$ e seu número máximo será 7. Porém, é solicitado que antes de atingir seu módulo ele recicle. Para que isso ocorra, será incrementada uma porta E que fique recebendo os dados da contagem, e quando essa contagem atingir o número 6, a porta irá enviar um sinal às entradas CLEAR dos *flip-flops* (Figura 7.19).



Para conhecer outros tipos de contadores, acesse o [csite](http://www.joinville.ifsc.edu.br/~mtaques/Digital/ELD2.pm.pdf) <http://www.joinville.ifsc.edu.br/~mtaques/Digital/ELD2.pm.pdf>

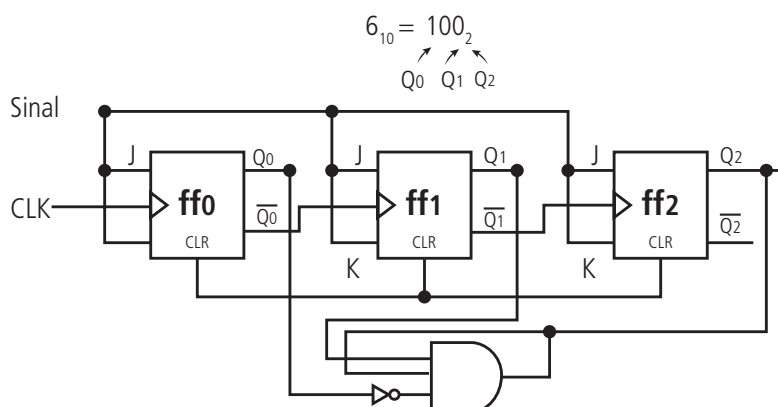


Figura 7.19: Circuito de um contador binário assíncrono limitado a 5

Fonte: Elaborada pelos autores

7.3.5 Exemplo de CI contador assíncrono

Como exemplo de contador binário de quatro *bits* assíncrono, podemos citar o 74LS93 mostrado na Figura 7.20 a seguir.

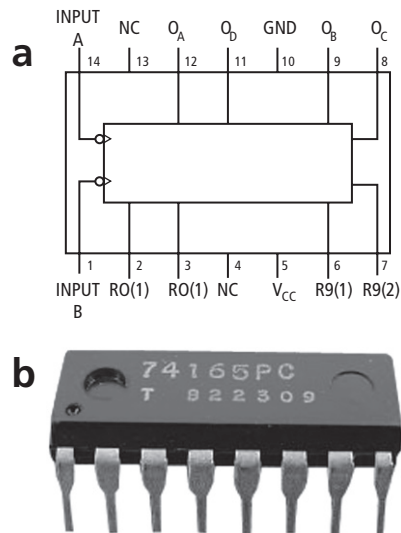


Figura 7.20: CI 74LS93

Fontes: a) National Xinying Industrial Vocacional High School; b) Des Moines Classifieds

7.4 Memórias

Entre os subsistemas sequenciais fundamentais em um sistema computacional, a memória é certamente a mais conhecida, pelo menos todos já ouviram falar em memória, embora pouquíssimos saibam realmente o que é e como atua.

Já abordamos anteriormente os registradores como forma de armazenamento de dados, porém em várias situações um sistema necessita de uma alternativa que assegure armazenamento de uma quantidade maior de informação que os registradores são capazes de suportar. Para isso, utilizamos as memórias. A memória é, então, parte de um sistema de armazenamento de uma grande quantidade de dados em binário.

7.4.1 Arranjo de uma memória

Uma memória é composta por elementos chamados de células. Cada célula é responsável pelo armazenamento de um *bit*. As células são construídas em blocos de linhas e colunas, como pode ser visto na imagem abaixo, que mostra um arranjo de 8 x 8 células, num total de 64 células (64 *bits*, ou oito *bytes*). Esse mesmo arranjo também poderia ser de quatro linhas e 16 colunas, ou mesmo uma coluna e 64 linhas (Figura 7.21).

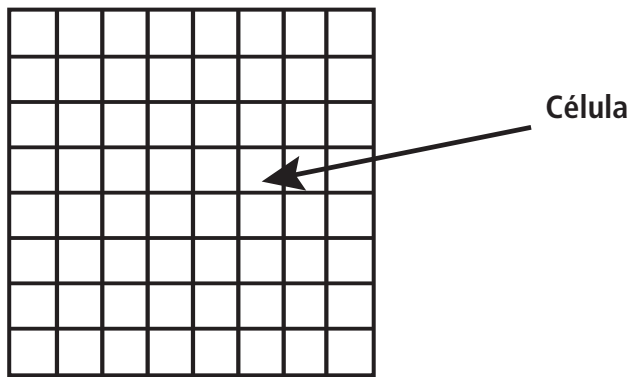


Figura 7.21: Arranjo de 64 células em oito linhas por oito colunas

Fonte: Elaborada pelos autores

O número total de células existente é o que determinará quantos *bits* essa memória será capaz de armazenar. Na Figura 7.21 a memória então tem a capacidade de 64 *bits*, ou oito *bytes*. As memórias atuais dos computadores pessoais trabalham na casa de 512 Mb (512 milhões de *bytes*) para cima.

7.4.2 Circuito interno de uma célula

A Figura 7.22 representa uma célula de memória típica usada pelas memórias RAM.

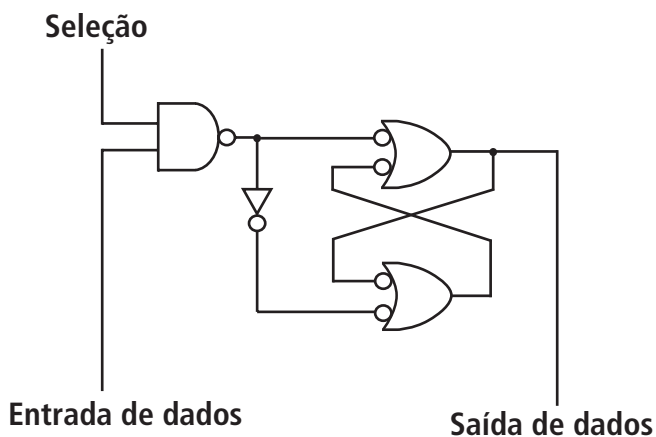


Figura 7.22: Circuito interno de uma célula de memória RAM

Fonte: Floyd (2007, p. 559)

Uma característica da célula de memória **RAM** é que enquanto houver nível de tensão alimentando a memória, e, por conseguinte, a célula, ela retém o *bit* de informação registrado internamente (0 ou 1). Porém, quando a tensão é retirada da célula o *bit* retido é perdido.

A-Z

RAM

Abreviação do inglês *cRandom Access Memory*, que traduzindo quer dizer Memória de Acesso Aleatório.

7.4.3 Endereço de memória

Para que determinado dado seja localizado (ou armazenado) é necessário que ele tenha um localizador. Esse localizador é chamado de endereço de memória. O endereço de memória de um *bit* será a localização da célula onde ele está armazenado (ou será armazenado), e essa localização é dada pela linha seguida pela coluna da célula. Para exemplificar, observe a Figura 7.23 que dá destaque a uma célula.

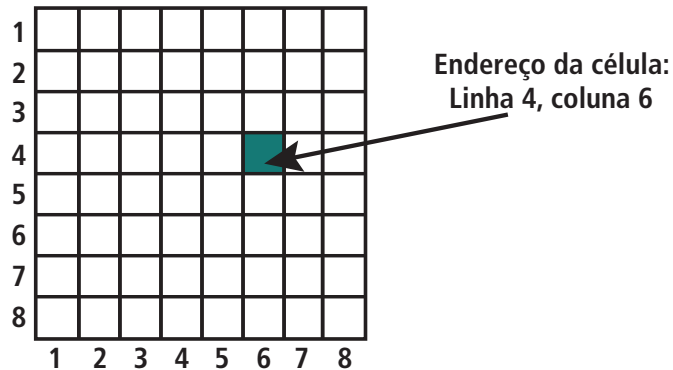


Figura 7.23: Endereço da célula

Fonte: Adaptada de Floyd (2007 p. 557)

Já um *byte* (grupo de 8 *bits*) pode ser localizado pela linha que ocupa, conforme demonstrado na Figura 7.24 a seguir.

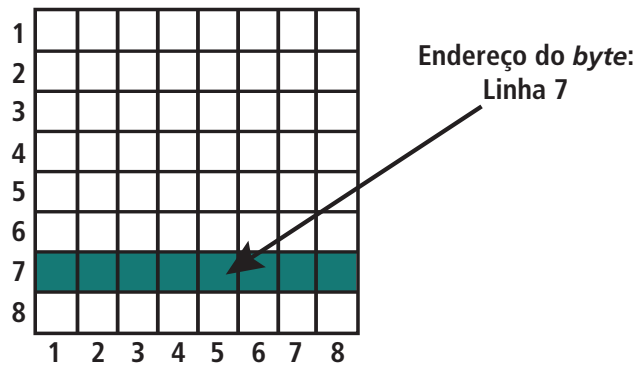


Figura 7.24: Endereço do byte

Fonte: Adaptada de Floyd (2007, p. 557)

7.4.4 Escrita e leitura

Duas são as operações efetuadas com uma memória. Quando ela armazena a informação, que chamamos de escrita, e quando ela fornece os dados armazenados, que chamamos de leitura.

Ao realizar a escrita de uma informação, a memória será preenchida em um endereço próprio para esta, que será novamente acessada na leitura por meio desse endereço.

Tanto a escrita quanto a leitura de dados são realizadas por um grupo de linhas de acesso à memória chamado de barramento de dados.

Processando a escrita

A Figura 7.25 ilustra o processo simplificado da escrita.

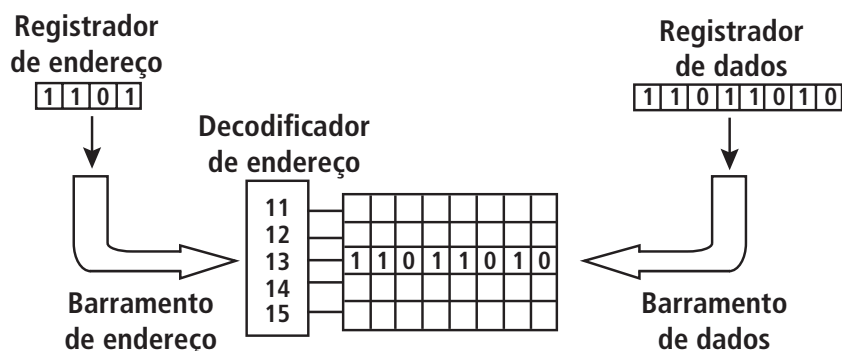


Figura 7.25: Processo de escrita na memória

Fonte: Adaptada de Floyd (2007, p.558)

1. Um código referente ao endereço onde o dado será armazenado (linha 13 ou 11012) é gerado no registrador de endereço e colocado no barramento de endereço.
2. Uma vez que o código do endereço esteja no barramento, ele é recebido pelo decodificador de endereço que o decodifica e seleciona na memória a posição especificada.
3. A memória então recebe um comando de escrita e o *byte* de dados armazenado no registrador de dados é colocado no barramento de dados e armazenado no endereço de memória selecionado (no caso, linha 13).

Caso no endereço especificado já exista um dado armazenado, este será sobrescrito pelo novo dado enviado pelo barramento de dados.



Processando a leitura

A Figura 7.26 ilustra o processo simplificado da leitura.

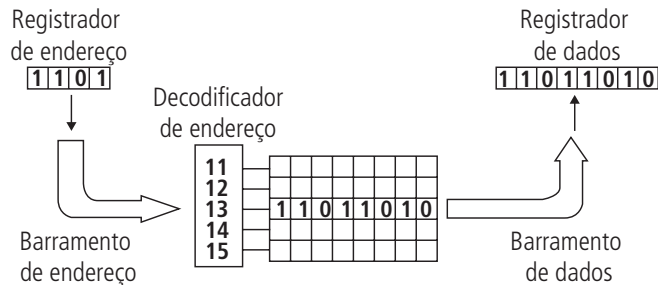


Figura 7.26: Processo de leitura da memória

Fonte: Adaptada de Floyd (2007, p. 558)

1. Um código existente no registrador de endereço referente ao endereço do dado a ser lido (no caso linha 13 ou 11012) é colocado no barramento de endereço.
2. Uma vez que o código do endereço esteja no barramento, o decodificador de endereço decodifica o endereço e seleciona a posição especificada na memória.
3. A memória então recebe um comando de leitura e uma cópia do *cbyte* de dados que está armazenado no endereço de memória selecionado (no caso linha 13) é colocada no barramento de dados e carregada no registrador de dados.



Após a leitura do dado na memória, o dado permanece na memória. Apenas uma cópia do que há no endereço especificado é enviada ao barramento de dados.

7.5 Exemplos de memórias

As principais memórias utilizadas nos computadores pessoais podem ser divididas pelo tipo de **encapsulamento** utilizado, podendo ser: DIP (*Dual In-line Package*), SOIC (*Small Outline Integrated Circuit*) e LCC (*Leaded Chip Carrier*), conforme mostra a Figura 7.27 a seguir.

A-Z

Encapsulamento

É basicamente uma proteção que envolve um circuito integrado. Sua principal função é garantir a conexão segura entre os circuitos e as placas físicas. Essa proteção é composta por pinos de metal que fixam um envoltório retangular feito de metal, plástico ou de outros materiais (Tecmundo).



Figura 7.27: Exemplos de memórias

Fontes: a) GPEC SRL; b) e c) Data Sheet.com

Há também os já conhecidos pentes de memórias que podem ser facilmente removíveis dos computadores (Figura 7.28).

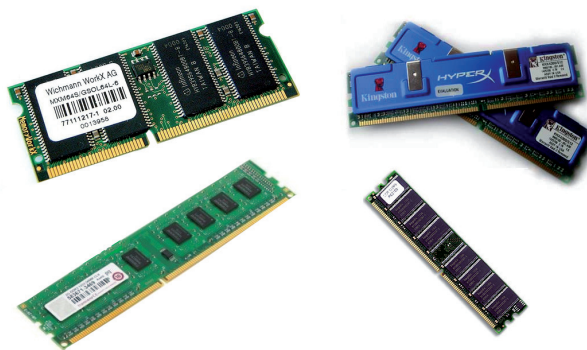


Figura 7.28: Pentes de memórias RAM

Fonte: Elaborada pelos autores

Resumo

Nesta aula vimos o princípio de funcionamento de um contador, também de um registrador de deslocamento, assim como as suas respectivas características. Depois estudamos transmissão de dados em forma serial e paralela. Conhecemos os tipos de registradores quanto à entrada ou saída serial ou paralelas; ao final, estudamos o princípio de funcionamento de memórias baseadas em *flip-flops*.

Atividades de aprendizagem

1. Um contador binário está no estado $Q_3Q_2Q_1Q_0 = 1010$. Qual é o próximo estado?
2. Considere que um marcador digital de pulsos esteja inicialmente “resetado” para 12 pulsos e marcando 0 no visor. Determine o estado binário do marcador após ocorrerem 62 pulsos do sinal do *clock*.
3. Em processo de leitura de dados da memória representada abaixo o registrador de endereço envia o seguinte código para o decodificador de endereço: 11000. Qual a informação que será passada para o barramento de dados?

22	1	0	0	1	1	0	0	0
23	1	0	0	0	1	0	1	0
24	1	1	0	1	1	0	1	0
25	1	0	0	1	1	1	1	0
26	0	1	0	1	1	0	1	1

4. A capacidade em *bits* de uma memória que tem 1024 endereços e pode armazenar oito *bits* em cada endereço é?
5. As afirmações abaixo se referem às memórias. Assinale as verdadeiras.
 - a) O *byte* é a menor unidade de dado que pode ser manipulada na memória.
 - b) Uma operação de escrita armazena dados na memória.
 - c) Uma operação de leitura faz com que dados da memória sejam copiados e em seguida apagados da memória.
 - d) Uma unidade de dado é localizada pelo seu endereço.
 - e) Uma RAM é volátil e tem capacidade de leitura/escrita.

Aula 8 – Conversores

Objetivos

Definir conversores A/D e D/A.

Compreender os princípios de funcionamento dos conversores A/D e D/A.

Conhecer o princípio de amostragem por pulso e sua importância para o conversor A/D.

Conhecer o princípio da quantização e sua importância para o conversor D/A.

Fazer as representações gráficas dos conversores A/D e D/A de n bits.

8.1 Conversores

Somos seres analógicos vivendo em um mundo analógico, cercados por um número cada vez maior de ferramentas digitais. Temos músicas, fotos, filmes, termômetros, livros, celulares, relógios, balanças, enfim, inúmeras coisas digitais refletindo o nosso mundo analógico. Entre esses dois “mundos” temos os conversores, cuja função é exatamente interligá-los de modo a tornar as ferramentas digitais possíveis e, principalmente, úteis.

Os conversores são divididos em duas categorias, de acordo com a sua função:

- Conversor Analógico/Digital (A/D) → Transforma sinais analógicos em digitais.
- Conversor Digital/Analógico (D/A) → Transforma sinais digitais em analógicos.

8.2 Entendendo ondas

Todas as coisas que podem, de um modo ou outro, serem mensuradas em um determinado tempo, podem ter suas medidas representadas em um gráfico em forma de onda.

Por exemplo, imagine a temperatura ambiente de uma cidade no decorrer de um dia, conforme mostra a Figura 8.1, representando a medição da temperatura em intervalos de duas horas. No início da madrugada a temperatura vai caindo, até amanhecer, quando o sol começa a aquecer, tendo seu ponto mais elevado por volta do meio-dia. Nesse instante, com o sol começando a descer em direção ao horizonte, a temperatura começa a cair. Vem o início da noite e com a total ausência do sol, a tendência da temperatura é continuar a diminuir.

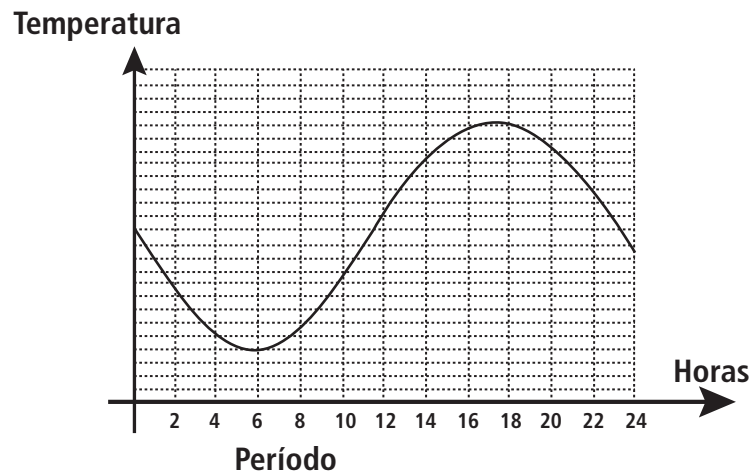


Figura 8.1: Onda representando medição de temperatura a cada duas horas

Fonte: Elaborada pelos autores

8.3 Características comuns aos conversores D/A e A/D

1. **Faixa dinâmica:** é a faixa de amplitude de operação do sinal analógico (normalmente uma tensão) dentro da região de trabalho do conversor. O sinal de entrada deve ser condicionado de forma a possibilitar sua máxima utilização dentro dessa faixa dinâmica.
2. **Resolução:** é a menor quantidade que pode ser convertida dentro da faixa dinâmica do sinal de entrada. É especificada pelo número de *bits* do conversor.

8.4 Conversor analógico/digital (A/D)

Um conversor analógico/digital é um sistema que faz a transposição dos valores analógicos para valores digitais.

Várias são as grandezas físicas que podemos mensurar em sua forma analógica: temperatura, pressão, velocidade, etc. Para que essas grandezas possam ser permutadas para grandezas digitais, antes de mais nada se faz necessário o uso de um dispositivo que as transforme em sinais elétricos. Esse dispositivo é o transdutor.

O transdutor então envia os sinais elétricos para um condicionador de sinal, cuja função é condicionar os sinais elétricos recebidos para uma escala que possa ser recebida então pelo conversor. A Figura 8.2 representa todo esse processo.

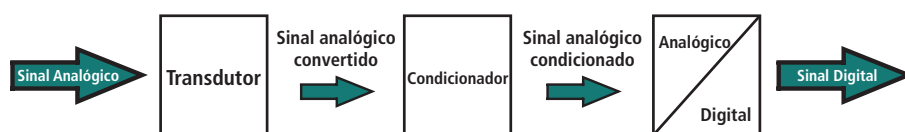


Figura 8.2: Processo de digitalização do sinal analógico

Fonte: Elaborada pelos autores

Para o nosso estudo não iremos considerar o processo do transdutor e do condicionador, apenas o que se processa no conversor A/D. Sempre que nos referirmos aos valores ou sinais analógicos de entrada do sistema, já fica pressuposto que todo o processo anterior já ocorreu.



8.4.1 Aproximação por amostragem de pulsos

Tomemos como exemplo um intervalo de uma curva de onda sonora, cujo valor altera constantemente a cada instante, resultando em uma curva semelhante à mostrada na Figura 8.3, em que também podemos localizar, na mesma curva, pulsos sendo realizados a cada instante T , resultando amostragem da curva em pulsos.

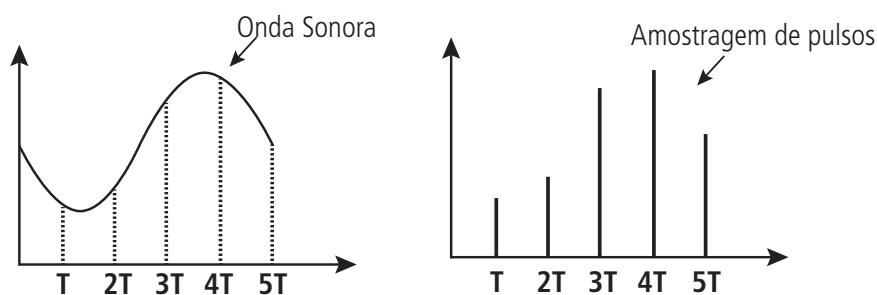


Figura 8.3: Amostragem de pulsos

Fonte: Elaborada pelos autores

Aproveitando o gráfico de amostragem de pulsos, e levando em consideração apenas ele, não há outro meio de saber o que acontece com os valores da onda entre os intervalos dos pulsos. Temos que presumir então que esses

valores não se alteram enquanto um novo valor não for detectado. Desse modo, somos obrigados a fazer uma aproximação na curva resultante usando somente os valores disponíveis, que resultará na curva identificada na Figura 8.4 a seguir.

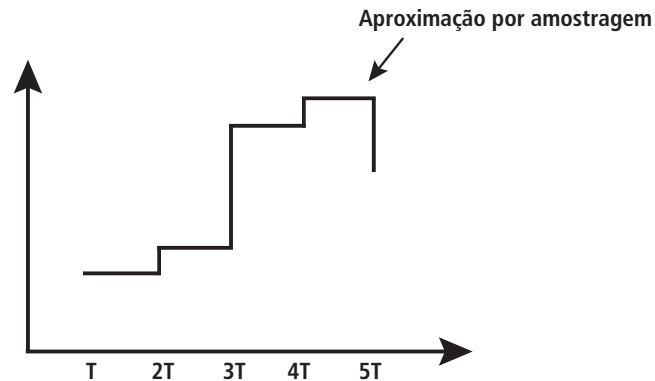


Figura 8.4: Aproximação da curva por amostragem

Fonte: Elaborada pelos autores

Como pode ser visto, a nossa curva ainda não está com uma representação nem próxima da realidade. Vamos então reduzir o intervalo do tempo para $\frac{1}{4}$ de T , resultando na Figura 8.5 a seguir.

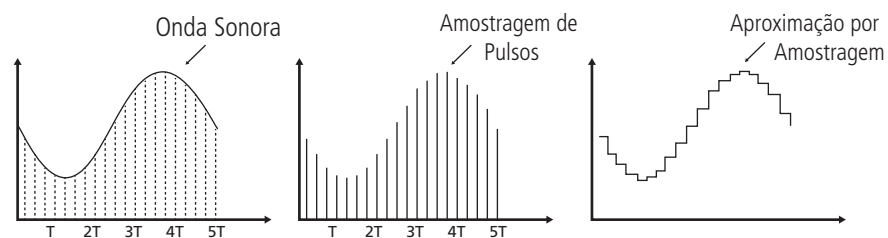


Figura 8.5: Aproximação por amostragem com $\frac{1}{4}$ de T

Fonte: Elaborada pelos autores

Comparando as aproximações por amostragem para os intervalos de pulso T (Figura 8.4) e $\frac{1}{4}$ de T (Figura 8.5) é fácil perceber que a segunda é bem mais próxima da onda analógica original. Isso acontece por causa de uma regra dos circuitos conversores que estabelece que quanto maior for a frequência utilizada, maior será a precisão da resposta. Por esse motivo, arquivos MP3 com frequência de 44,1 kHz (que seria similar a 44.100 pulsos de amostragem por segundo) tem uma fidelidade na qualidade maior que outro de 11kHz. O inconveniente em usar altas taxas de amostragem em busca de melhor qualidade do áudio é que resultará em um arquivo com maior número de informações, portanto maior em *bytes*.

8.4.2 Quantização

Uma determinada onda sonora, depois de passar por todo o processo (transdutor e condicionador), chega a um conversor analógico/digital de dois *bits*. A Figura 8.6 mostra então a onda em estudo sendo medida em *bits*.

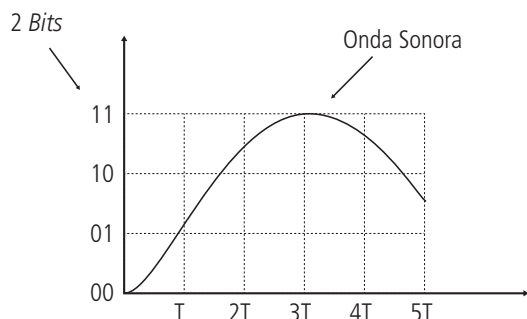


Figura 8.6: Onda sonora em um conversor A/D de dois *bits*

Fonte: Elaborada pelos autores

É importante lembrar que os valores binários só podem expressar os valores que os *bits* representam. Por isso, mesmo que no instante $2T$ o valor da onda já tenha ultrapassado 102, não é possível ao conversor quantizar esse valor. O mesmo valerá para o instante $5T$, por exemplo, em que o valor já está abaixo de 102, mas não há como quantizá-lo. A diferença entre o valor real e o valor quantizado é chamada de **erro de quantização**. A Figura 8.7 exhibe então a forma como o conversor de dois *bits* interpreta a onda em estudo.

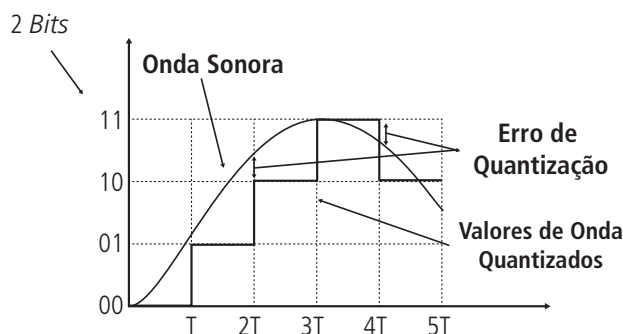


Figura 8.7: Valores interpretados pelo conversor A/D de dois *bits*

Fonte: Elaborada pelos autores

Vamos agora usar a mesma onda, porém com um conversor A/D de quatro *bits* (Figura 8.8).

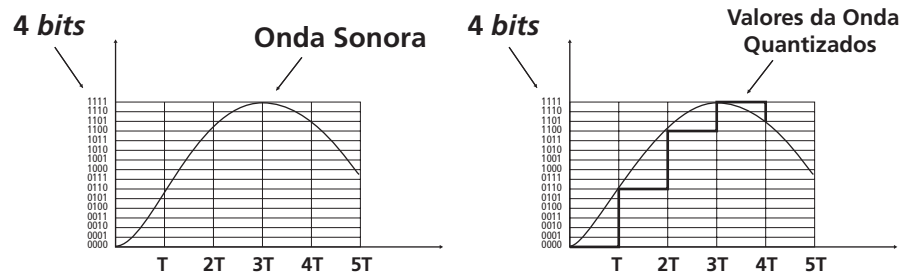


Figura 8.8: Valores interpretados por um conversor A/D de quatro *bits*

Fonte: Elaborada pelos autores

Comparando os valores quantizados pelos dois conversores, da Figura 8.7 e da Figura 8.8, é possível verificar que os erros de quantização do de quatro *bits* é bem menor que os do de dois *bits*. Isso porque quanto mais *bits* forem utilizados, menor será a faixa dinâmica de valores que ele quantizará, o que em eletrônica digital significa que terá uma melhor resolução.

A resolução é determinada pela fórmula $Res = 1/2^n$, sendo n o número de *bits* do conversor. E a partir da resolução podemos calcular qual é a faixa dinâmica de valores resultante do conversor que é determinada pelo produto entre a amplitude da faixa de trabalho e a resolução.

Exemplificando:

Supondo que ambos os conversores (dois *bits* e quatro *bits*) trabalhem com um sinal de entrada de 4V.

Então a resolução para o conversor de dois *bits* será:

$$Res = 1/2^n \text{ à } Res = 1 / 2^2 \text{ à } Res = 1 / 4 \text{ à } Res = 0,25.$$

E sua faixa dinâmica será de:

$$0,25 \times 4 \text{ V} = 1\text{V}.$$

Desse modo, o conversor A/D de dois *bits* do exemplo tem uma mudança de valores em sua saída a cada 1 Volt.

Vejamos agora o mesmo exemplo, porém usando o conversor de quatro *bits*.

$$Res = 1/2^n \text{ à } Res = 1 / 2^4 \text{ à } Res = 1 / 16 \text{ à } Res = 0,0625$$

A sua faixa dinâmica será então:

$$0,0625 \times 4 \text{ V} = 0,25\text{V}$$

Comparando as duas faixas, é claro observar que o conversor de quatro *bits* é bem mais preciso, pois ele muda sua resposta em consequência de uma alteração de entrada a cada 0,25 V.

8.4.3 Representação gráfica do conversor A/D

A representação gráfica de um conversor A/D deverá mostrar (Figura 8.9):

- a grandeza elétrica de entrada do conversor;
- o conversor; e
- o número de saídas do conversor, em paralelo.

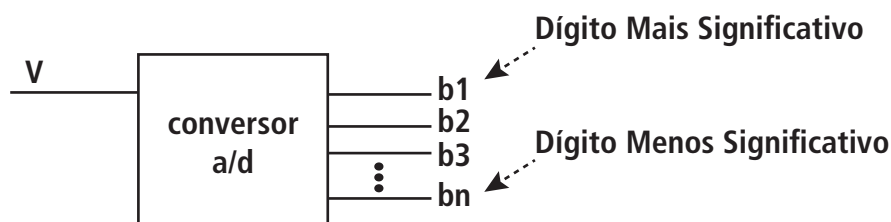


Figura 8.9: Representação gráfica de um conversor A/D para n bits

Fonte: Elaborada pelos autores

Exemplo: Conversor A/D de quatro *bits* com entrada de 10V (Figura 8.10).

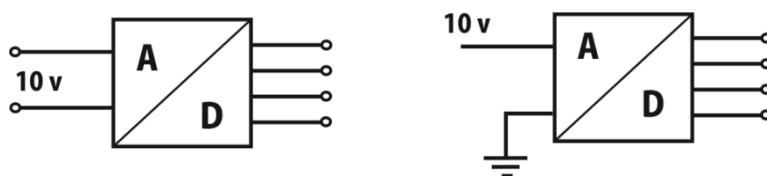


Figura 8.10: Exemplo de representação gráfica

Fonte: Elaborada pelos autores

8.5 Conversor digital/analógico (D/A)

A forma generalizada da conversão de sinais digitais em sinais analógicos segue o processo inverso do conversor A/D. O sinal digital é recebido pelo conversor D/A, que o transforma em sinal elétrico enviando para o condicionador. Este por sua vez repassa o sinal já condicionado para o transdutor, conforme mostra a Figura 8.11 a seguir.

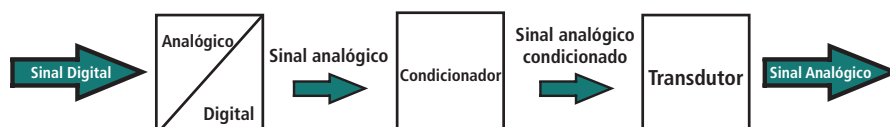


Figura 8.11: Processo de conversão do sinal digital em analógico

Fonte: Elaborada pelos autores

8.5.1 Resolução e amplitude máxima

Nos conversores D/A a resolução é uma importante característica para a transformação do sinal digital em sinal elétrico, pois é essa característica que determina o valor do sinal de saída. A Figura 8.12 mostra um mesmo sinal digital de três *bits* de entrada sendo recebido por três diferentes conversores, A, B e C, cada um com uma resolução diferente. Observe que os valores de saída de cada um para cada entrada é proporcional à sua resolução, o que, ao final, resulta em um valor de **amplitude máxima** também individual.

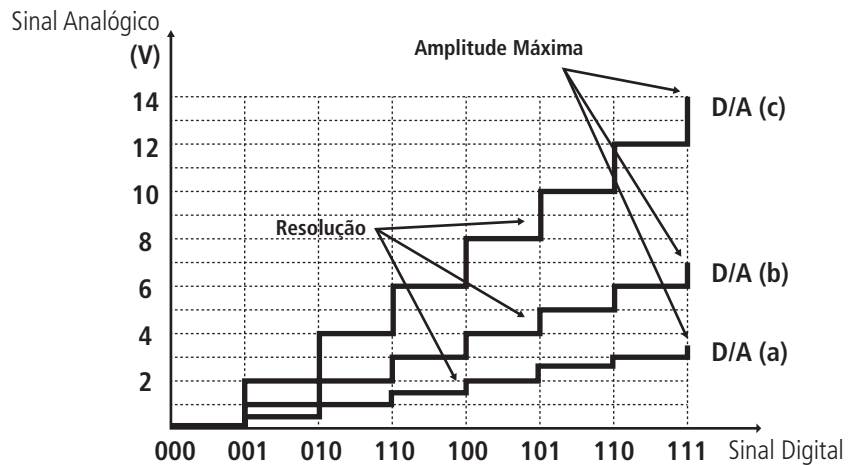


Figura 8.12: Três conversores D/A (a, b e c) com resoluções e amplitudes máximas próprias

Fonte: Elaborada pelos autores

A-Z

Extrapolação matemática

É uma técnica capaz de gerar um valor a partir de certos dados incompletos, com precisão relativa ao intervalo dos dados.

Existem várias técnicas para a conversão dos dados recebidos pelo conversor D/A em sinais elétricos. Cada uma envolve regras matemáticas específicas, de acordo com o melhor resultado obtido específico para a situação proposta. Mas, de forma geral, o resultado se dá por **extrapolação** dos valores binários recebidos e transformados em sinais analógicos, de acordo com a respectiva resolução do D/A. A Figura 8.13 ilustra como uma curva analógica é gerada a partir da extrapolação dos dados recebidos e gerados pelo conversor a cada pulso.

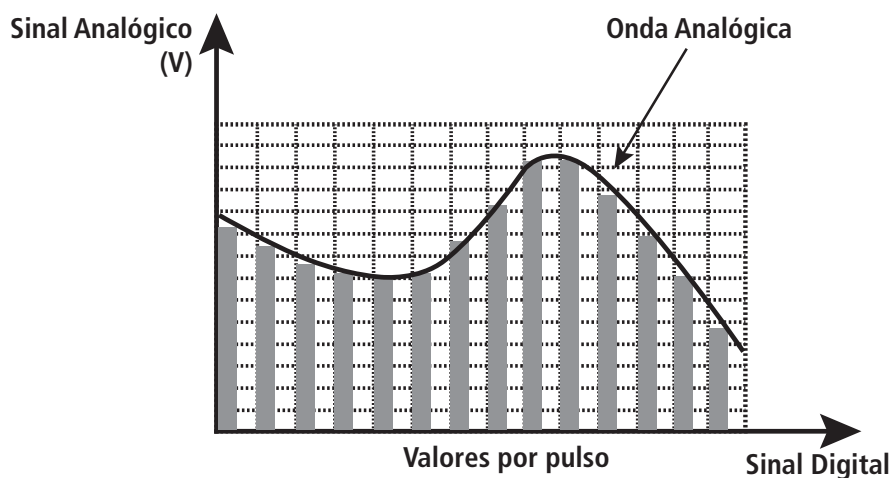


Figura 8.13: Onda analógica gerada por interpolação dos dados recebidos por pulso

Fonte: Elaborada pelos autores

8.5.2 Representação gráfica de um conversor D/A

A representação gráfica de um conversor D/A deverá mostrar (Figura 8.14):

- as entradas do conversor, em paralelo;
- o conversor; e
- a grandeza elétrica de saída do conversor.

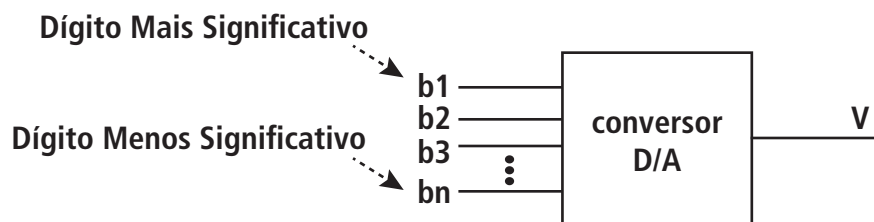


Figura 8.14: Representação gráfica de um conversor D/A para n bits

Fonte: Elaborada pelos autores

Exemplo: Conversor D/A de quatro bits com saída de 10V (Figura 8.15).

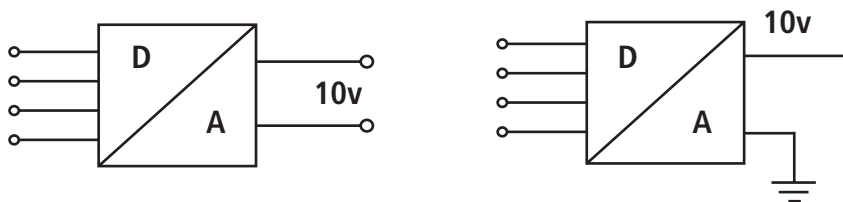


Figura 8.15: Exemplo de representação gráfica conversor D/A

Fonte: Elaborada pelos autores

8.6 Sinais digitais

8.6.1 Ruído

Ruído é um sinal interferente de natureza aleatória que provoca a degradação de sinal durante seu processamento. Este sinal pode ter diferentes origens: ser gerado pelo homem, ser intrínseco ou relacionado aos dispositivos eletrônicos. A Figura 8.16 mostra o ruído presente em uma onda sonora.

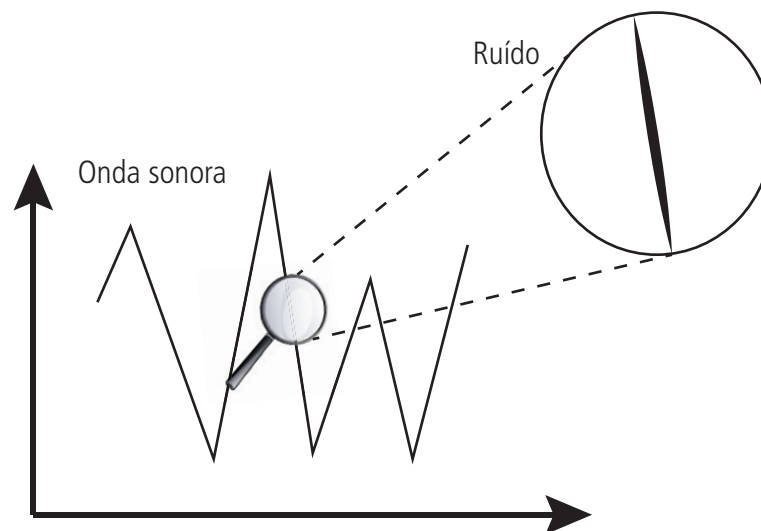


Figura 8.16: Ruído em uma onda sonora

Fonte: Elaborada pelos autores

8.6.2 Vantagens no uso de sinais digitais

Muitas são as situações em que sinais analógicos são transformados em sinais digitais para receber um tratamento lógico de modo a usar os dados recebidos em outros sistemas, ou mesmo para aperfeiçoá-los ou modificá-los. A utilização dos sinais digitais nos dá vantagens em relação aos sinais analógicos, pois eles nos permitem:

- aumentar a precisão e resolução da medida;
- melhorar a relação sinal/ruído;
- modificar os sinais para outros valores; e
- identificar e reduzir erros de medida.

8.6.3 Exemplos de conversores

Como exemplo de conversor analógico/digital, podemos citar o MCP3002 (Figura 8.17). Um conversor com resolução de dez *bits* com até 75.000 amostras por segundo.

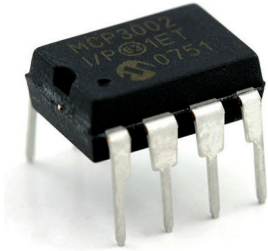


Figura 8.17: MCP3002 – conversor A/D

Fonte: Multilógica Shop

Já como exemplo de conversor digital/analógico, podemos citar o LTC2630 (Figura 8.18), um conversor com resoluções de oito, dez e 12 *bits*.



Figura 8.18: LTC2630 – conversor D/A

Fonte: Nei.com

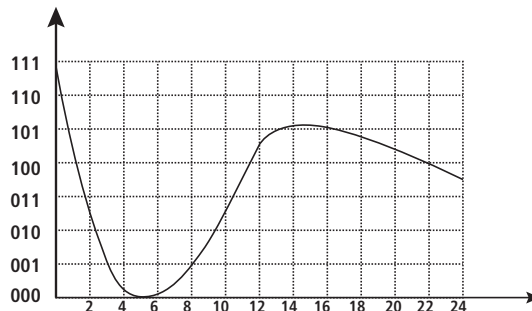
Resumo

Nesta aula vimos o que são conversores A/D e D/A e estudamos os princípios de funcionamento de cada um. Conhecemos o princípio de amostragem por pulso e sua importância para o conversor A/D, assim como o princípio da quantização e sua importância para o conversor D/A. Também fizemos as representações gráficas dos conversores A/D e D/A de n bits.

Atividades de aprendizagem

1. A Figura abaixo representa os valores de uma onda analógica codificados por um conversor A/D. Determine os valores registrados nos instantes t igual a:

- a) 4
- b) 8
- c) 12
- d) 16
- e) 20



2. Um conversor D/A de quatro bits trabalha recebendo sinais elétricos máximos de 10V. Determine:
 - a) a sua resolução;
 - b) a sua faixa dinâmica;
 - c) o valor do sinal analógico enviado por esse conversor para o condicionador quando o sinal recebido em um determinado instante for 1010.
3. Dois conversores A/D recebem e codificam os sinais de uma mesma grandeza analógica. Um dos conversores tem frequência de 4Khz, e o outro, de 12Kz. Qual dos dois resultará valores quantizados mais próximos dos valores da entrada analógica? Explique sua resposta.

Referências

ABSTRAÇÃO. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2012. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=Abstra%C3%A7%C3%A3o&oldid=32708149>>. Acesso em: 8 nov. 2012.

AXIOMA. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2012. Disponível em: <<http://pt.wikipedia.org/w/index.php?title=Axioma&oldid=32783238>>. Acesso em: 10 nov. 2012

BARBOSA, Carlos Ricardo dos Santos. **Eletrônica digital**. São Leopoldo: FIERG/SENAI, Serviço Nacional de Aprendizagem Industrial, 2009.

BRAGA, Newton C. Curso de eletrônica digital. **Saber Eletrônica**, n. 8, 2002. Número especial.

BRASIL. Ministério do Trabalho e Emprego. Secretaria de Políticas Públicas de Emprego. Instituto Educação e Pesquisa. Data Brasil. **Apostila de mecatrônica**. São Paulo: Catalisa, 2006.

CETEBE, Centro de Educação Tecnológica do Estado da Bahia. Unidade Camaçari. **Apostila de técnicas digitais I**. Camaçari: CETEBE, 2005.

FLOYD, Thomas L. **Sistemas digitais: fundamentos e aplicações**. 7. ed. Porto Alegre: Artmed, 2007.

GARCIAL, Paulo Alves; MARTINI, José Sidnei Colombo. **Eletrônica digital: teoria e laboratório**. 2. ed. São Paulo: Érica, 2010.

GOLDENMAP. Disponível em: <http://pt.goldenmap.com/Mapa_de_Karnaugh>. Acesso em: 20 abr. 2012.

GÜNTZEL, José Luís; NASCIMENTO, Francisco Assis do. **Introdução aos sistemas digitais**. v.2001/1. Disponível em: <<http://www.inf.ufsc.br/~guntzel/isd/isd1.pdf>>. Acesso em: 20 abr. 2012.

LEAL, Murilo Parreira. **Lógica matemática aplicada à engenharia da computação**. São Paulo: 2007.

SERVIÇO NACIONAL DA INDÚSTRIA – SENAI. **Instrumentação: eletrônica digital**. CPM – Programa de Certificação do Pessoal de Manutenção. Tubarão, ES: SENAI, 1999 .

SENGER, Luciano José. **Introdução à organização dos computadores**. Disponível em: <<http://www.ljsenger.net/2006/introorgec/intro12.pdf>>. Acesso em: 20 abr. 2012.

ZAPELINI, Wilson B. **Lógica combinacional**. Florianópolis: CEFET-SC, 2003.

Currículo dos professores-autores

Prof. Jodelson Moreira Julião

Formado em Engenharia Elétrica pela Universidade de Fortaleza (UNIFOR) com especialização MBA em Gestão Estratégica da Tecnologia da Informação pela Universidade Gama Filho. Há seis anos é professor do CETAM nos cursos de Técnico em Informática, Técnico em Edificação e Técnico em Eletroficações, além de Tutor de ensino a distância.

Prof. Rosumiro Trindade Junior

Tecnólogo em Eletrotécnica pelo Instituto de Tecnologia da Amazônia - UTAM, licenciado no Programa Especial de Formação Pedagógica para Formadores da Educação Profissional pela Universidade do Sul de Santa Catarina (UNISUL), técnico em Eletrônica/IFAM. Professor titular do SENAI-AM, nas áreas de Eletrônica e Automação Industrial.



ISBN 978-85-63576-41-5

