

Circuitos Digitais

Circuitos digitais

Rodrigo Vinícius Mendonça Pereira
Hugo Tanzarella Teixeira

© 2018 por Editora e Distribuidora Educacional S.A.
Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Editora e Distribuidora Educacional S.A.

Presidente
Rodrigo Galindo

Vice-Presidente Acadêmico de Graduação e de Educação Básica
Mário Ghio Júnior

Conselho Acadêmico
Ana Lucia Jankovic Barduchi
Camila Cardoso Rotella
Danielly Nunes Andrade Noé
Grasiele Aparecida Lourenço
Isabel Cristina Chagas Barbin
Lidiane Cristina Vivaldini Olo
Thatiane Cristina dos Santos de Carvalho Ribeiro

Revisão Técnica
Marcio Aparecido Artero

Editorial
Camila Cardoso Rotella (Diretora)
Lidiane Cristina Vivaldini Olo (Gerente)
Elmir Carvalho da Silva (Coordenador)
Letícia Bento Pieroni (Coordenadora)
Renata Jéssica Galdino (Coordenadora)

Dados Internacionais de Catalogação na Publicação (CIP)

Pereira, Rodrigo Vinícius Mendonça
P436c Circuitos digitais / Rodrigo Vinícius Mendonça Pereira,
 Hugo Tanzarella Teixeira. – Londrina : Editora e Distribuidora
 Educacional S.A., 2018.
 232 p.

ISBN 978-85-522-0715-3

1. Circuitos digitais. 2. Tecnologia. I. Pereira, Rodrigo
Vinícius Mendonça. II. Teixeira, Hugo Tanzarella. III.
Título.

CDD 600

Thamiris Mantovani CRB-8/9491

2018
Editora e Distribuidora Educacional S.A.
Avenida Paris, 675 – Parque Residencial João Piza
CEP: 86041-100 – Londrina – PR
e-mail: editora.educacional@kroton.com.br
Homepage: <http://www.kroton.com.br/>

Sumário

Unidade 1 Introdução aos circuitos digitais	7
Seção 1.1 - Conceitos de circuitos digitais	9
Seção 1.2 - Características dos circuitos digitais	28
Seção 1.3 - Visão geral das funções lógicas básicas	44
Unidade 2 Lógica combinacional	63
Seção 2.1 - Lógica programável e simulação	66
Seção 2.2 - Álgebra booleana e simplificação lógica	85
Seção 2.3 - Diagrama de Veitch-Karnaugh	101
Unidade 3 Lógica sequencial	121
Seção 3.1 - Flip-flops	123
Seção 3.2 - Contadores e registradores	143
Seção 3.3 - Máquinas de estados finitos	161
Unidade 4 Especificação e interfaceamento de circuitos integrados	179
Seção 4.1 - Especificações de circuitos integrados	182
Seção 4.2 - Aplicação de circuitos integrados	199
Seção 4.3 - Aplicação de circuitos integrados	216

Palavras do autor

Caro aluno, estamos vivendo momentos significativos na história da humanidade. Se você nasceu entre a década de 1980 e 1990, pertence à geração Y, também conhecida como geração do **milênio**. Essa geração foi a primeira que nasceu necessariamente em um mundo completamente tecnológico. Essa evolução somente foi possível devido aos significativos avanços da ciência — em especial, aos avanços na área de microeletrônica. Ano após ano, observamos saltos tecnológicos em dispositivos como celulares, tablets ou computadores pessoais, tanto em capacidade de processamento quanto nas aplicações que funcionam nesses sistemas.

Dessa forma, este livro traz uma análise básica, mas contemporânea, de um tema cada vez mais relevante: circuitos digitais.

O conteúdo aqui exposto guia você para um entendimento aprofundado sobre eletrônica digital e seus conceitos básicos, aspectos do sistema de numeração utilizado em sistemas computacionais, bem como suas operações e codificações. Apresentará também uma visão geral sobre funções lógicas e portas lógicas, detalhes sobre álgebra booleana e sua simplificação lógica. Além disso, serão abordados assuntos fundamentais, como diagrama de Veitch-Karnaugh, *flip-flops*, contadores e máquina de estados finitos. Por fim, entraremos no campo da especificação de circuitos integrados e o seu interfaceamento e aplicação.

Todo esse estudo possibilitará que você compreenda pontualmente detalhes sobre circuitos digitais. Isso permitirá que você obtenha confiança para tratar desse assunto com seus pares. Saiba que o tema é relevante, atual e, ao mesmo tempo, instigante. Dessa forma, não se limite a este livro, mas extrapole procurando outras fontes de informação e, principalmente, experimentando as abordagens aqui propostas. No momento no qual este livro está sendo produzido, é evidente o chamado **Boom do Hardware**. Esse movimento é caracterizado por grandes empresas inundando o mercado com sistemas completos para desenvolvedores extrapolarem a imaginação. Existe um investimento significativo da parte das grandes organizações em ambientes de programação gratuitos, com diversas bibliotecas free e dispositivos dos mais variados preços, processamento e aplicação.

Nós, brasileiros, somos um povo criativo. Já embarcamos nessa nova onda, sobretudo com boas ideias.

Caso você seja professor, este livro faz uma coletânea dos pontos mais relevantes sobre circuitos digitais e, em contrapartida, aborda tópicos fundamentais para que o aluno tenha um contato mais aprofundado com o mundo da microeletrônica. Este conteúdo permite uma abordagem mais prática, apresentando aplicações e ferramentas de simulação disponibilizadas de forma gratuita na web, propondo, assim, uma abordagem moderna. Dessa forma, a aprendizagem se baseia na descoberta gradativa do conteúdo pelo aluno, na qual são enfatizados conceitos-chave em conjunto com situações problemas que instigam o estudante aprofundar sobre um tema contemporâneo, factível e escalável. De leitura fácil e instigante, o presente texto é um convite ao estudo, sem deixar de lado o contexto atual e a criatividade da juventude. Seja bem-vindo ao mundo dos circuitos digitais.

Introdução aos circuitos digitais

Convite ao estudo

Estimado aluno, nas seções seguintes desta unidade iniciaremos nosso conhecimento sobre circuitos digitais. Inicialmente, em linhas gerais, iremos definir os conceitos de circuitos digitais. A construção desse conhecimento se inicia definindo o que é um circuito digital e em que são empregados, seguido de alguns exemplos de circuitos digitais no nosso cotidiano e quais as tecnologias envolvidas. Em seguida, iremos entender a diferença entre um circuito analógico e um circuito digital. Começaremos o texto entendendo os primórdios do maior sistema eletrônico do mundo – o sistema de telecomunicações –, e o quanto esse sistema faz uso tanto de circuitos analógicos quanto de circuitos digitais. Posteriormente, entenderemos como é definida a informação com relação aos níveis lógicos e às formas de ondas. O próximo tópico traz um conhecimento mais específico sobre como transformar a informação em circuitos lógicos através de portas lógicas e variáveis booleanas. Por fim, apontaremos nosso conhecimento para o futuro. Pontuaremos detalhes sobre circuitos integrados, circuitos lógicos e, finalmente, sistemas digitais. Tudo isso através de um contexto com base em exemplos.

Todo esse conhecimento justifica-se observando o nosso cotidiano. No presente momento em que você lê esse texto, bilhões de pessoas estão interagindo com algum circuito digital, ora ligando o celular, ora usando um relógio, ora assistindo a um vídeo na internet. A quantidade de informação gerada somente na internet rompeu a casa dos **zettabyte** (equivalente a um **sextilhão** de bytes). Portanto, circuitos digitais e toda eletrônica envolvida nesse volume de informação faz parte da revolução dos bytes nos últimos 100 anos. Você já pensou que faz parte dessa revolução?

E que você vive um momento da história na qual não existem mais barreiras entre idiomas, culturas e conhecimento? Todo esse cenário só foi possível porque milhões de cientistas levaram a ciência a um novo patamar, criando sistemas e incrementando as tecnologias de forma a facilitar o dia a dia do homem. Mas o que é um sistema digital? Como ele interage como um sistema analógico? Como um circuito com alguns sensores pode ser capaz de tomar uma decisão e acionar um motor? Como se dá essa interação?

Todas essas perguntas podem ser respondidas no contexto de circuitos digitais. Portanto, gradativamente podemos construir o seu pensamento, trabalhando conceitos básicos sobre circuitos digitais e evoluindo o pensamento até culminar em circuitos mais complexos para acionamento de motores, por exemplo.

Para tanto, vamos iniciar esta unidade de ensino contextualizando um cenário prático: considere que você está iniciando sua carreira na área de microeletrônica. Você acaba de ser contratado para fazer parte do time da linha de montagem de uma empresa de telecomunicações, na área de modens para a internet. Você é uma pessoa extremamente curiosa e não se limita a aprender somente o básico. De forma despretensiosa, por meio dos seus estudos nas horas vagas, acaba resolvendo um grande problema em um modem. Até onde você pretende chegar? Será que é possível subir de cargo ou mudar de setor? Quem sabe em breve você não estará em um time mais desafiador – o time do P&D (Pesquisa e Desenvolvimento)?

Como mencionado, circuitos digitais fazem parte da nossa vida. Conhecer, identificar e solucionar problemas em circuitos digitais não é uma tarefa fácil, exigindo muito estudo e dedicação. Para chegar ao time de P&D, você deve seguir uma trilha que passa pelo conhecimento, dedicação e profissionalismo. Esteja atento a todos os conceitos que serão apresentados aqui. Lembre-se: não se contente somente com o conhecimento deste material, extrapole. Boa sorte, mãos à obra e bons estudos.

Seção 1.1

Conceitos de circuitos digitais

Diálogo aberto

Caro aluno, esta seção tem o objetivo de direcionar o seu conhecimento de forma mobilizadora e através de um contexto muito próximo da sua realidade. Todo esse contexto é para conduzir você o mais próximo possível dos conceitos de circuitos digitais. Esses conceitos são a base para você futuramente começar a projetar sistemas digitais. Dessa forma, iremos compreender a diferença entre sinais digitais e sinais analógicos e os conceitos fundamentais sobre circuitos digitais, como esses sinais se enquadram em um circuito digital e como toda essa tecnologia evoluiu até os presentes dias. Vamos em especial compreender os níveis lógicos e as formas de ondas geradas por um circuito digital básico. Ainda, de fundamental importância, iremos compreender circuitos lógicos, portas lógicas e entender detalhes sobre variável booleana. Por fim, iremos mais a fundo, analisando conceitos sobre circuitos integrados, circuitos lógicos e finalmente sobre sistemas digitais.

Esta seção, apesar de introdutória, é fundamental. Ela explica conceitos básicos e necessários para o bom entendimento da disciplina.

Retomando o nosso contexto, imagine o seguinte cenário: você foi contratado por uma empresa da área de telecomunicações inicialmente para trabalhar com a montagem de produtos como modens, roteadores e switches. O seu trabalho consiste somente em soldar componentes desses produtos. À sua frente existe uma mesa muito grande, uma iluminação perfeita, uma bela estação de solda SMD e diversos componentes e placas de circuitos impressos prontos para serem soldados. Algumas coisas não estão claras para você: como uma sequência de pulsos elétricos é transformada em uma palavra? Como funcionam essas peças pretas que todos chamam de microchips? As perguntas são muitas. No seu intervalo, você observa um quadro na parede da empresa na qual aparecem um telégrafo, algumas pessoas felizes e alguns microchips.

Além disso, você percebe duas caixas de componentes sobre a sua mesa e decide saber mais sobre esses objetos. Uma forma é pesquisando na internet sobre a folha de dados (ou *datasheet*). Um dos componentes é um conversor analógico para digital e os outros são *flip-flops* tipo D e portas lógica AND e OR. Você agora precisa lidar com alguns questionamentos que lhe vem à mente.

A curiosidade leva você a pesquisar e questionar: o que são telégrafos? Você pesquisa como eles eram feitos, como é o diagrama de tempos do sinal de um telégrafo e como esses sinais eram definidos em digitais ou analógicos. Você pesquisa também sobre a diferença entre circuitos analógicos e circuitos digitais e como são definidos os níveis lógicos e as formas de onda digitais de um telefone convencional.

Ainda querendo saber mais, você busca informações sobre os componentes *flip-flops* tipo D e as portas lógica AND e OR que estão sobre a sua mesa, sobre encapsulamentos de circuitos integrados (ou CI). Qual a diferença de um CI DIP e um CI BGA? Todo esse questionamento vai ajudar você a entender muito como esses circuitos são feitos e como você irá soldá-los.

Lembre-se de que esse cenário é um aprofundamento ao seu conhecimento sobre circuitos digitais. Todo o conhecimento será feito com base em bastante leitura e pesquisa, e, para ser usado no futuro no seu dia a dia, devemos documentar todo esse aprendizado. Dessa forma, produza um relatório com a sua proposta de solução. Utilize a sua capacidade analítica e pontue os principais aspectos que você levantou. Use figuras, mas não se esqueça de explicá-las, bem como apontar as referências – devemos dar o mérito para que fez, é o mais justo. Use uma linguagem científica, não use 1^a pessoa e evite gerúndio.

Nas seções a seguir, serão apresentadas a base para esse conhecimento, mas não fique limitado somente a essa fonte de aprendizagem. Pesquise, converse e troque informações com outras pessoas. O mercado está favorável para funcionários proativos e esforçados. Boa sorte e bom trabalho.

Não pode faltar

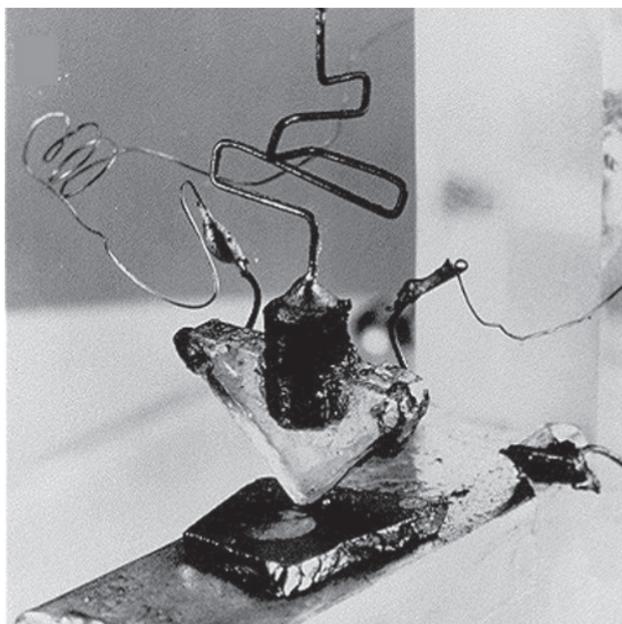
As seções a seguir discorrem sobre eletrônica digital e o seu contexto atual, sobre circuitos analógicos e digitais e os níveis lógicos e formas de onda. Após, iremos tratar sobre circuitos lógicos, portas lógicas

e variável booleana. Por fim, iremos entender circuitos integrados, circuitos lógicos e sistemas digitais.

Eletônica digital – contexto atual

O primeiro transistor fez 70 anos em 2017 (Figura 1.1). Segundo Vahid (2008), o transistor foi inventado nos laboratórios da empresa Bell Telephone (divisão AT&T) por John Bardeen e Walter Houser Brattain, em 1947. Consistia em um transistor de silício e germânio, dois elementos da tabela periódica que, após processos físico-químicos, adquirem propriedades semicondutoras (sólidos cristalinos cuja condutividade elétrica é intermediária entre condutor e isolante).

Figura 1.1 | O primeiro transistor – silício e germânio

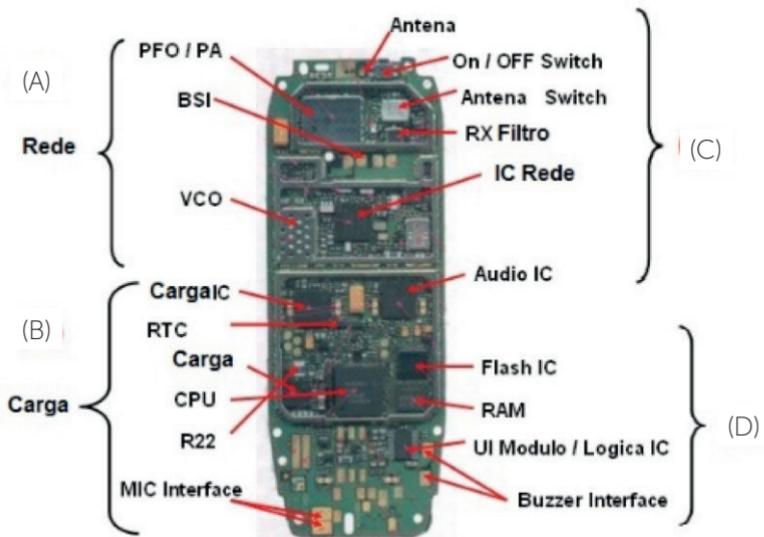


Fonte: <goo.gl/XXAEcF>. Acesso em: 18 set. 2017.

Prezado aluno, como uma tecnologia não tão antiga mudou a história da humanidade? Como esse desenvolvimento acarretou em tantas transformações na sociedade contemporânea? Como as últimas duas décadas foram responsáveis por essas mudanças? As respostas para esses questionamentos certamente passam pelos termos **eletônica digital e analógica**. Essas palavras tornaram parte do vocabulário de todos devido ao seu uso cotidiano. Tanto os circuitos

digitais e analógicos quanto as técnicas digitais e analógicas são utilizados em diversas áreas como: medicina, computação, aviação, entretenimento, telecomunicações, viagens espaciais, e assim por diante. Portanto, a eletrônica não caminha sozinha no mundo. Ela se divide em duas áreas – eletrônica analógica e eletrônica digital. Iremos aprofundar o conceito de cada área na seção a seguir. É importante ter em mente que essas duas áreas são complementares uma a outra – é fácil provar essa afirmação. Observe a Figura 1.2. Ela ilustra o nível de integração da eletrônica para uma tecnologia de celular. Na Figura 1.2 (A), é possível observar os circuitos responsáveis pela seção de conexão com a rede do aparelho. Na Figura 1.2 (B), essa seção é responsável também pela alimentação do mesmo. A Figura 1.2 (C) traz as etapas de filtros dos sinais recebidos pelo celular, a antena, o circuito de rede e áudio. Por fim, na Figura 1.2 (D), temos o *buzzer* (ou sirene), as memórias e os circuitos de que controlam as vibrações do telefone.

Figura 1.2 | Partes de celular e suas funções

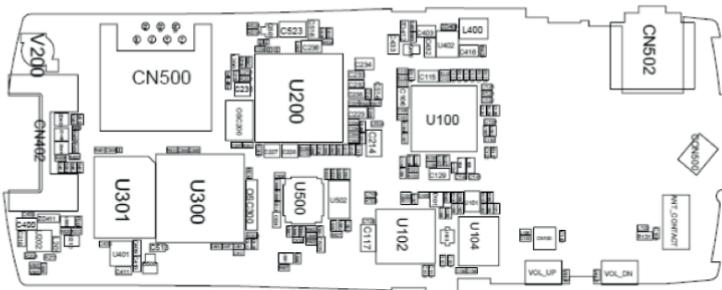


Fonte: <<http://www.mobilecellphonerepairing.com>>. Acesso em: 18 set. 2017.

Esse exemplo (Figura 1.2), portanto, ilustra bem a integração entre os diversos sistemas digitais e analógicos de um celular. Facilmente, observamos que sem a perfeita interação entre os diversos circuitos, não seria possível o correto funcionamento deste dispositivo.

Dentre as diversas etapas de desenvolvimento de um projeto, existe uma que se chama *layout* da placa. Nessa etapa, além de outros diagramas, é gerada uma figura que ajuda a entender como os componentes serão distribuídos na placa – essa figura é chamada de diagrama elétrico. A Figura 1.3 é um exemplo de diagrama elétrico.

Figura 1.3 | Diagrama elétrico de um celular



Fonte: Manual de Serviço Samsung GSM SGH-C110.

Observe nessa figura como os espaços são bem aproveitados e como os componentes eletrônicos são distribuídos. Não existe margem para erro. Sem o bom entendimento sobre eletrônica, não seria possível esse nível de integração.

Mas qual é o contexto da eletrônica digital em que vivemos? Não é possível determinar exatamente. Vivemos um contexto muito amplo, mas podemos apontar algumas tendências. Segundo o MIT Technology Review (2017), existem tendências que poderão afetar diretamente a nossa economia, medicina e cultura nos próximos quatro a 15 anos. Listamos algumas a seguir:

1. Reversão de paralises: através de implantes, cientistas pretendem reestabelecer movimentos de pessoas com lesões graves – disponível em 10 a 15 anos.
 2. Pagamento através de reconhecimento facial: detecção de padrões faciais autorizando o pagamento de contas – já disponível.
 3. Computadores quânticos na prática: Google, IBM, Intel, Microsoft, entre outras, investem massivamente em sistemas quânticos completos e, possivelmente, estarão disponíveis em quatro a cinco anos.

4. Selfie 360: câmeras permitem a gravação de imagens em 360 graus – já disponível.

5. Terapia genética 2.0: através de manipulações nos genes das células, será possível tratar câncer, problemas do coração, entre outras doenças – já disponível.

O mais interessante está no fato de as tecnologias acima possuírem, em algum momento, alguma relação com a eletrônica digital. No momento em que esse texto é escrito, a IBM anuncia a criação de microchips nó tecnológico de 5nm (BU, 2017). Estamos falando em bilhões de transistores em um microchip. Dessa forma, quando alguém perguntar acerca de eletrônica digital, em geral olhamos para as tendências do mercado e o nível de integração (nó tecnológico ou *half-pitch*) dos microchips para responder. Por isso, leia sempre sobre o assunto, seja curioso e dedicado.



Pesquise mais

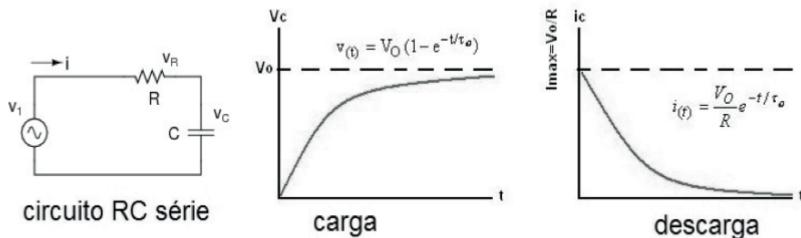
O termo nó tecnológico é uma forma de caracterizar circuitos integrados. O termo suscita controvérsias. No passado, referia-se ao menor tamanho da tecnologia dentro de um transistor, ou seja, o *gate* do transistor. Mas, em 2005, a ITRS – *International Technology Roadmap for Semiconductors* – optou por utilizar *half-pitch* para se referir à metade da menor distância entre distância entre a camada de metal 1 e a próxima camada de metal. Pesquise sobre o processo de fabricação de semicondutores e o uso do termo em inglês *ic fabrication process*. Procure vídeos a respeito do assunto. É fascinante.

Circuitos analógicos e digitais – níveis lógicos e formas de onda digitais

Neste momento, é importante definir dois conceitos: o que é um (i) **sinal analógico** e o que é (ii) **sinal digital**. Segundo Sedra et al. (2007), devemos ter em mente que um **sinal (x)** nada mais é que um valor ou uma amostra de algum sistema ou fenômeno físico observado em um **tempo (t)**. Sabendo disso, para os autores, um **sinal analógico X(t)** é um sinal que pode assumir diversos valores no **tempo (t)**, ou seja, ele pode assumir infinitas possibilidades. Um exemplo simples é o paladar. Por mais que o cozinheiro tente repetir os ingredientes, nunca o sabor de um prato é igual ao prato anterior: algo sempre varia, ou o sal ou a temperatura de cozimento do alimento. Já um **sinal digital x(t)** é aquele que sempre assume valores finitos (ou discretos) no **tempo (t)**, ou seja,

dentro de um conjunto de valores, ele assume um deles (SEDRA et al., 2007). O simples ato de contar a quantidade de moedas em uma carteira é um sinal digital, pois pode assumir só valores exatos. Agora, fica fácil definir o que é circuito analógico e um circuito digital. Segundo Tocci (2003), um circuito analógico é quando uma quantidade (valor, medida) é representada por um indicador proporcional continuamente variável. Em termos gerais, é um circuito cujo sinal é analógico. Um exemplo é um circuito resistor/capacitor em série (RC), cuja carga e descarga do capacitor é representada por um sinal analógico (Figura 1.4) em um gráfico.

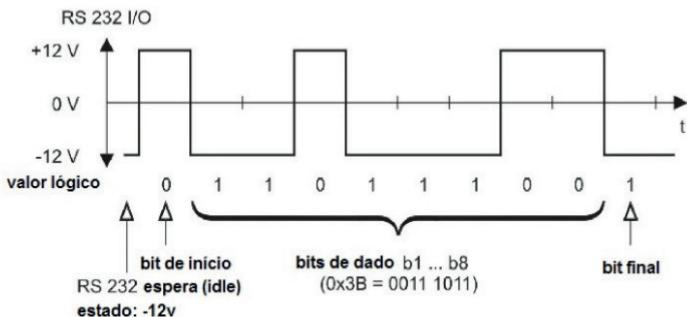
Figura 1.4 | Circuito analógico – resistor capacitor série



Fonte: elaborada pelo autor.

Já um circuito digital, segundo Tocci (2003), é um circuito cuja quantidade (valor, medida) é representada por símbolos chamados dígitos ou seja, este é um circuito cujos sinais são digitais. Um exemplo são circuitos que trabalham somente com dois estados: ligado/desligado (*on/off*), alto/baixo (*high/low*), verdadeiro/falso (*true/false*) entre outros, para representar dígitos (símbolos como um site: www.kroton.com.br) através de valores binários (Figura 1.5).

Figura 1.5 | Sinal digital

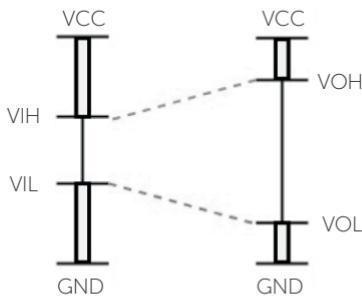


SINAL DIGITAL - SERIAL

Fonte: elaborada pelo autor.

Observando a Figura 1.5, é possível compreender que um sinal digital possui um nível lógico que define o dígito zero (+12 volts) e o dígito um (-12 volts). Estamos tratando agora dos níveis lógicos e como um dispositivo entende o valor representado por esses níveis lógicos em binário. Vamos começar definindo alguns detalhes. Um circuito digital trabalha com valores zeros (0) e uns (1). Considere como nosso padrão 0 volts (ou GND ou terra) para o dígito zero (0) e VCC para o dígito um (1). Você percebeu que existe uma infinidade de valores possíveis entre 0 volts e VCC? Dessa forma, é preciso saber como se dá essa condição intermediária (entre 0 e VCC). Vamos começar entendendo os níveis de tensões possíveis em circuitos digitais. Para tensões de circuitos digitais na entrada, é adotada a nomenclatura **VIH** (*V Input High*) para a menor tensão na qual ainda é possível representar o valor um (1). Para a maior tensão na qual ainda é representado o valor zero (0) na entrada, é chamado **VIL** (*V Input Low*). Algo similar ocorre na saída de um circuito digital. Os níveis de tensão podem variar entre **VCC** e **VOH** (*V Output High*) para indicar o valor um (1), e entre **VOL** (*V Output Low*) e **GND** para indicar zero. A Figura 1.6 ilustra todo esse conceito de representação de uma entrada e uma saída digital e seus respectivos níveis de tensão *V High* e *V Low*.

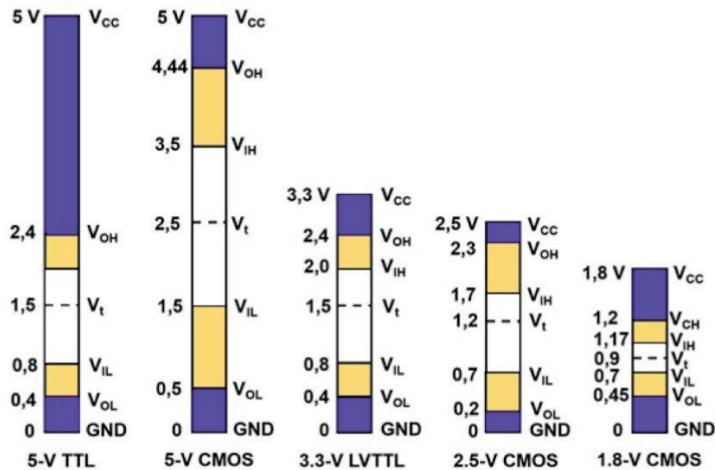
Figura 1.6 | Entrada e saída digitais



Fonte: elaborada pelo autor.

Ainda a respeito de níveis de tensão, há diferentes tecnologias que se aplicam aos circuitos digitais. Existem, por exemplo, tanto circuitos TTL e CMOS quanto o LTTL (*Low Power TTL*), BiCMOS, LVC (*Low Voltage CMOS*), entre outros. Para cada tecnologia, há um padrão entendimento de nível de tensão. A Figura 1.7 escalona as tecnologias com relação aos níveis de tensão (VCC, GND, V High e V Low).

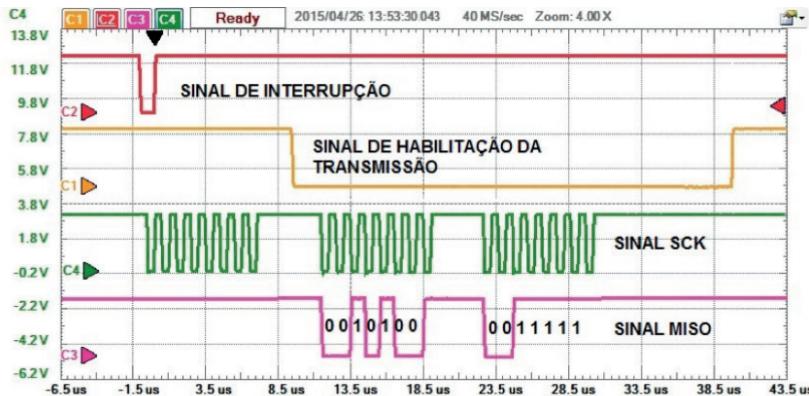
Figura 1.7 | Níveis de tensão digitais



Fonte: elaborada pelo autor.

Definido como os níveis de tensão representam dígitos binários, basta agora entender que podemos gerar esses níveis de tensão em forma de onda. A Figura 1.8 ilustra a transmissão de dados através do protocolo SPI (Serial Peripheral Interface). Cada sinal (SCK, MOSI e MISO) tem uma finalidade, mas ambos transmitem bits respeitando os níveis lógicos – *LOW* (0 volts) significa o bit zero (0) e *HIGH* (VCC) significa bit um (1).

Figura 1.8 | Forma de onda de um sinal digital – bits representando a informação



Fonte: elaborada pelo autor.

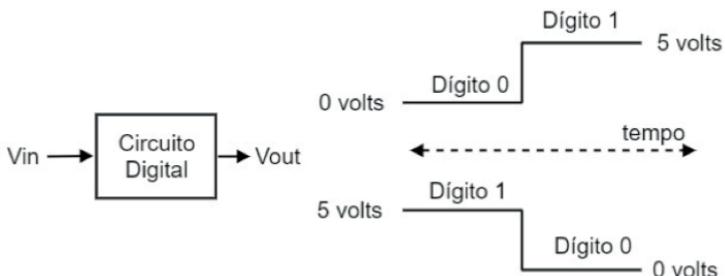
Agora, compare a forma de onda da Figura 1.5 com a forma de onda da Figura 1.8. Você consegue observar que o bit um (1) é representado

por (-)12 volts (menos doze volts) e o bit zero (0) por (+) 12 volts (mais doze volts)? Ocorre que no protocolo RS232 as tensões na faixa de +5 a +12 volts representam os níveis baixos (bit zero), enquanto tensões entre -5 a -12 volts representam os níveis altos (bit um). Essa lógica é utilizada para obtenção de maior segurança na transmissão dos dados em especial em locais ruidosos. Dessa forma, tensões positivas e negativas na transmissão de dados são chamadas de sinal **diferenciais**.

Circuitos lógicos, portas lógicas e a variável booleana

Segundo Tocci (2003), circuitos digitais são dispositivos projetados para produzir tensões de saída entre as faixas VCC – VOH e VOL – GND para os níveis um (1) e zero (0), respectivamente. Também são dispositivos projetados para responder a tensões de entrada entre as faixas VCC – VIH e VIL – GND para os níveis um (1) e zero (0), respectivamente (A Figura 1.6 exemplifica as referidas faixas de tensão). Para o autor, o modo como cada circuito digital interage a uma entrada é denominado lógica do circuito, ou seja, cada circuito segue regras determinadas pelas suas funções lógicas (ou portas lógicas) implementadas internamente.

Figura 1.9 | Circuito digital e os níveis binários (0 e 1)



Fonte: Adaptada de Tocci (2003).

A Figura 1.9 representa um circuito digital cuja entrada **Vin** produz uma entrada **Vout**. Tanto o sinal na entrada quanto a resposta na saída podem ser representados pelas duas formas de onda ao lado. Como mencionado anteriormente, dentro desse circuito existem funções lógicas. Tenha em mente que essas funções podem representar dois estados somente: **ligado** ou **desligado**; **zero** ou **um**; **aberto** ou **fechado**, ou seja, tanto na entrada quanto na saída desses blocos lógicos, são permitidos somente esses valores. A Tabela 1.1 lista as três principais portas lógicas utilizadas em circuitos digitais – porta AND (lógica E), OR

(lógica OU) e NOT (lógica NÃO ou INVERSA). Observe atentamente essa tabela. Estude cada porta lógica e a sua respectiva coluna. Essa tabela (Tabela 1.1) irá guiar nosso trabalho daqui para frente.

Reflita

No seu cotidiano, o que pode ser considerado uma função lógica? Por exemplo, no café da manhã, se você tiver leite e chocolate, você toma um achocolatado; ou, se você só tiver o leite, você toma somente o leite. Podemos considerar essas decisões como funções lógicas? A vida é feita de circuitos lógicos, ou seja, de decisões. Saiba tomá-las com cautela.

Tabela 1.1 | Blocos lógicos básicos

Porta	Símbolo	Tabela Verdade	Função Lógica	Expressão															
E AND		<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1	Assume valor 1 quando todas as entradas forem 1 e 0 nos outros casos	$S = A * B$
A	B	S																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OU OR		<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1	Assume valor 0 quando todas as entradas forem 1 e 1 nos outros casos	$S = A + B$
A	B	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NÃO (ou INVERSORA) NOT		<table border="1"> <thead> <tr> <th>A</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	S	0	1	1	0	Inverte o valor da variável de entrada A	$S = \bar{A}$									
A	S																		
0	1																		
1	0																		

Fonte: elaborada pelo autor.

Relembrando: em qualquer função (ou porta) lógica, somente são permitidos dois estados (0 ou 1) nas entradas nas saídas. Portanto, uma variável booleana é somente aquela variável (entrada A, B e saída S) que também só assume um dos dois estados permitidos (0 ou 1).



Com as portas lógicas da Tabela 1.1, é possível criar outras portas lógicas, por exemplo:

- A função NAND (Não E) é a combinação da porta AND com a porta NOT;
- A função NOR (Não Ou) é a combinação da porta OR com a porta NOT.

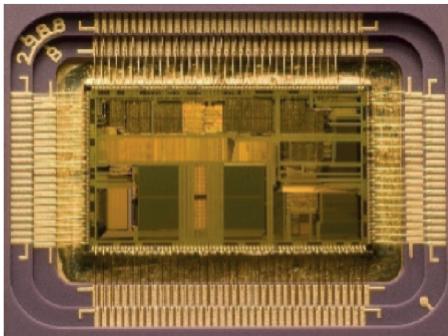
Não podemos esquecer a porta lógica XOR ou OU Exclusivo: essa porta é amplamente utilizada em criptografia, em somadores e segurança da informação. A lógica de uma porta XOR é: se as entradas forem iguais, a saída é zero (0). Se todas as entradas não forem iguais e ao menos uma delas for diferente, a saída é um (1).

O estudo das portas lógicas é fundamental em circuitos digitais. Não limite o seu conhecimento a somente o que foi exposto aqui. Procure aprofundar consultando outras fontes.

Circuitos integrados, circuitos lógicos e sistemas digitais

Segundo Vahid (2008), um circuito integrado (ou CI, microchip ou chip) consiste em um circuito eletrônico miniaturizado, feito basicamente de silício, do tamanho de uma unha.

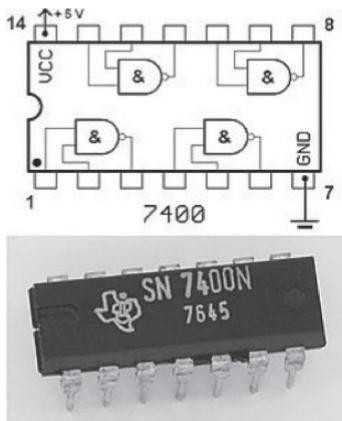
Figura 1.10 | Microprocessador Intel 80486DX2 com encapsulamento removido



Fonte: <https://pt.wikipedia.org/wiki/Circuito_integrado>. Acesso em: 18 set. 2017.

Foi a invenção dos circuitos integrados que permitiu o início de toda essa revolução que ocorreu no século 20 e ainda ocorrem nos dias atuais. Já os circuitos lógicos são os circuitos que fazem uso dos blocos lógicos explicados anteriormente. A Figura 1.11 exemplifica uma porta lógica NAND:

Figura 1.11 | Porta NAND – esquema do circuito integrado e hardware



Fonte: <https://pt.wikipedia.org/wiki/Porta_lógica>. Acesso em: 18 set. 2017.

Esses circuitos foram amplamente utilizados no início da eletrônica. Atualmente, podem ser usados, contudo a vida é mais fácil quando se programa um microcontrolador. Portanto, eles são importantes, mas não muito utilizados no dia a dia. Para Tocci (2003), sistemas digitais são uma combinação de dispositivos projetados para manipular informações lógicas ou valores físicos representados no formato digital. Podemos citar como sistemas digitais computadores pessoais, celulares, calculadoras avançadas, sistemas de áudio e vídeo.

Prezado aluno, o assunto não se encerra com o conteúdo apresentado aqui. Como mencionado anteriormente, pesquise outras fontes e questione os especialistas no assunto. Bons estudos.

Sem medo de errar

Relembrando a situação problema proposta: você foi contratado por uma empresa da área de telecomunicações para trabalhar com a montagem/solda de produtos como modens, roteadores e switches. Na sua bancada de trabalho, existem diversos componentes e placas de circuitos impressos prontos para serem soldados. Todo esse contexto leva você a se questionar sobre telecomunicações e os componentes que estão sobre a sua mesa.

Para responder aos seus questionamentos, inicie a sua pesquisa procurando sobre o telégrafo. Pesquise a história do telégrafo, como se deu o início dessa revolução nas telecomunicações e como eram feitos os telégrafos. Observe e compare com o sistema de

telecomunicações atual, por exemplo, rede móvel de celular versus rede cabeadas, transmissão por satélites e centrais de telefonia. Existe algo fundamental: o diagrama de tempos do sinal de um telegrafo. É importante essa análise para que você comece a perceber a diferença entre sinais digitais e analógicos. Observe como sinais digitais e analógicos diferem no tempo. Pesquise e compare como eram os sinais em uma rede como o telegrafo e como se dá atualmente em uma rede móvel.

A próxima etapa da sua pesquisa está em entender mais a respeito dos componentes que estão sobre a sua bancada. É fundamental que você saiba encontrar informações sobre eles. Na internet, pesquise sobre a folha de dados de um CI o 74LS74. Pesquise da seguinte forma: 74LS74 datasheet; ou pesquise sobre: *flip-flops tipo D*. As figuras a seguir são um exemplo de detalhes a que você deve estar atento.

Figura 1.12 | Diagrama lógico, distribuição de pinos e tabela verdade – CI 74LS74

DIAGRAMA LÓGICO		DISTRIBUIÇÃO DOS PINOS		TABELA VERDADE - TIPO D	
S1 5		V _{CC1} 1	16 V _{CC2}	CLOCK L X Q _{n+1}	
D1 7		Q1 2	15 Q2	H L L	
CE1 6		Q1 3	14 Q̄2	H H H	
R1 4		R1 4	13 R2		
C _C 9		S1 5	12 S2		
R2 13		CE1 6	11 C _{E2}		
CE2 11		D1 7	10 D2		
D2 10		V _{EE} 8	9 C _C		
S2 12					

Fonte: Motorola, MC10131 Datasheet. Disponível em <<https://www.bucek.name/pdf/mc10131.pdf>>. Acesso em: 2 out. 2017

Pesquise aspectos como as definições de tempo do componente e as condições de operação recomendadas. Expanda a sua pesquisa para portas lógica AND e OR.

Para finalizar, entenda os vários encapsulamentos possíveis para os componentes que você trabalha. Essa análise é fundamental, pois influencia em todo o projeto, tanto no custo quanto na montagem e possíveis manutenções.

Pesquise o custo entre um mesmo CI, mas com diferentes encapsulamentos. Quando se usa um e quando se usa outro encapsulamento?

Relembrando, esse cenário é um aprofundamento ao seu estudo sobre circuitos digitais. Dessa forma, todo o conhecimento será feito com base em bastante leitura, pesquisa e prática. A ideia aqui é que isso seja empregado no futuro no seu dia a dia. Uma forma de materializar o conhecimento é através de relatórios. Portanto, produza um relatório com todos os pontos levantados por você e não se esqueça de detalhar as portas AND e OR. Esse relatório deve conter todos os aspectos importantes para o bom entendimento e uso do componente. Não fique limitado somente a essa fonte de aprendizagem. Pesquise, converse e troque informações com outras pessoas. Boa sorte e bom trabalho.

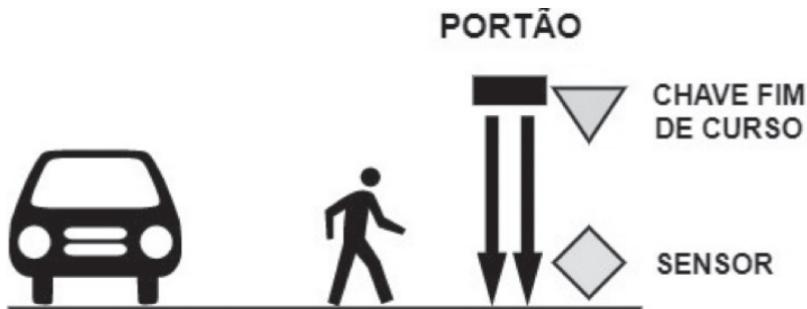
Avançando na prática

Meu portão automático

Descrição da situação-problema

Caro aluno, você está adquirindo um conhecimento muito amplo sobre circuitos digitais. Tanto que você já começa a observar soluções para problemas cotidianos. Você reparou que na casa da sua avó o portão automático pode ser melhorado. Ocorre que esse portão não possui um sensor de presença, ou seja, caso exista alguém abaixo dele, ele ignora e fecha. Observe a Figura 1.13. Ela inclui agora um sensor e uma chave de fim de curso.

Figura 1.13 | Sistema de abertura de portão



Fonte: elaborada pelo autor.

Qual seria uma lógica para a seguinte situação:

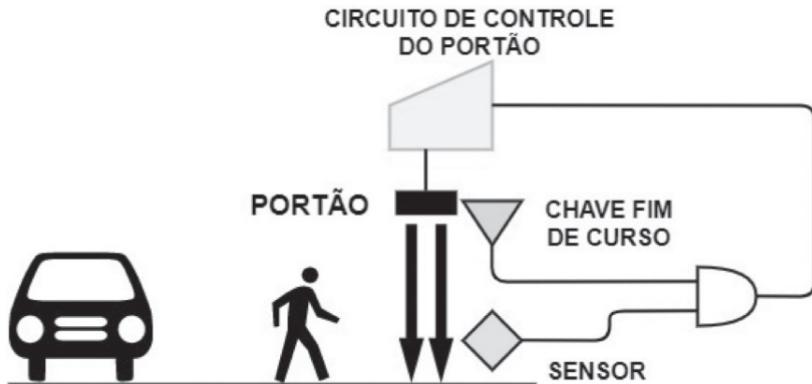
1. Portão no alto, chave ativa (em 1):
 - a) Sensor sem pessoas embaixo do portão (em 0);
 - b) Portão pode fechar.

2. Portão no alto, chave ativa (em 1):
 - a) Sensor com pessoas embaixo do portão (em 1);
 - b) Portão não pode fechar.
 3. Portão embaixo, chave desativada (em 0):
 - a) Portão pode abrir.
- Utilize as portas lógicas trabalhadas na unidade.

Resolução da situação-problema

Caro aluno, uma solução aparentemente simples é utilizar uma porta lógica AND tendo como entradas a CHAVE DE FIM DE CURSO e o SENSOR de presença. A saída seria utilizada na entrada do circuito que controla a subida e descida do portão (Figura 1.14).

Figura 1.14 | Solução de controle de abertura de portão



Fonte: elaborada pelo autor.

Existem diversas soluções. A princípio, essa solução pode resolver o problema. Pesquise como fazer esse teste fisicamente.

Faça valer a pena

1. Analise a notícia a seguir:

TV Digital: veja cronograma de desligamento do sinal analógico nas cidades Até o fim de 2017, oito capitais, além do DF, passam a receber apenas sinal digital. Quem tem TV analógica precisa ter conversor e antena digital.

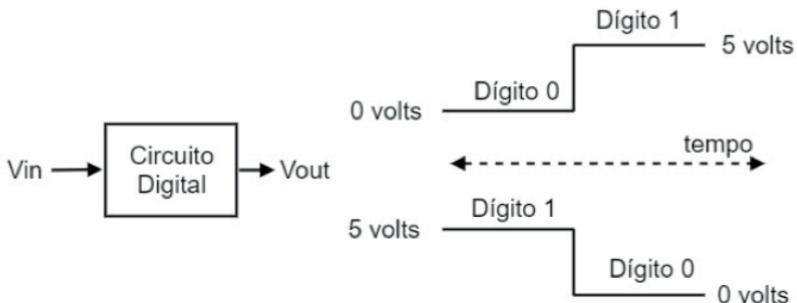
Fonte: G1. Disponível em: <<https://g1.globo.com/tecnologia/noticia/tv-digital-veja-cronograma-de-desligamento-do-sinal-analogico-nas-cidades.ghtml>>. Acesso em: 18 set. 2017.

Com relação a um sinal digital e um sinal analógico, podemos afirmar respectivamente:

- a) Sinal analógico $x(t)$ é aquele que sempre assume valores finitos (ou discretos) no tempo (t); sinal digital $X(t)$ é um sinal que pode assumir diversos valores no tempo (t).
- b) Sinal analógico $x(t)$ é aquele que sempre assume valores inconstantes no tempo (t); sinal digital $X(t)$ é um sinal que pode assumir diversos valores no tempo (t).
- c) Sinal digital $x(t)$ é aquele que sempre assume valores indeterminados no tempo (t); sinal analógico $X(t)$ é um sinal que pode assumir somente um valor no tempo (t);
- d) Sinal digital $x(t)$ é aquele que nunca assume valores finitos (ou discretos) no tempo (t); sinal analógico $X(t)$ é um sinal que não pode assumir diversos valores no tempo (t);
- e) Sinal digital $x(t)$ é aquele que sempre assume valores finitos (ou discretos) no tempo (t); sinal analógico $X(t)$ é um sinal que pode assumir diversos valores no tempo (t);

2. Segundo Tocci (2003), circuitos digitais são dispositivos projetados para produzir tensões de saída entre as faixas VCC – VOH e VOL – GND para os níveis um (1) e zero (0), respectivamente. Também são dispositivos projetados para responder a tensões de entrada entre as faixas VCC – VIH e VIL – GND para os níveis um (1) e zero (0), respectivamente. Para o autor, o modo como cada circuito digital interage a uma entrada é denominado lógica do circuito, ou seja, cada circuito segue regras determinadas pelas suas funções lógicas (ou portas lógicas) implementadas internamente.

Figura 1.15 | Circuito digital e os respectivos níveis dos lógicos



Fonte: adaptado de Tocci (2003).

A Figura 1.15 representa um circuito digital cuja entrada V_{in} produz uma saída V_{out} . Tanto o sinal na entrada quanto a resposta na saída podem ser representados pelas duas formas de onda ao lado. Dentro desse circuito, existem funções lógicas. Essas funções podem representar dois estados

somente: ligado ou desligado; zero ou um; aberto ou fechado, ou seja, tanto na entrada quanto na saída desses blocos lógicos, são permitidos somente esses valores.

Considerando o texto acima, qual será a mensagem (dígitos) enviada pela porta Vout se for lido na porta a seguinte sequência: GND (mais significativo) – VCC – VCC – GND – GND – VCC – GND – VCC (menos significativo)? Considere GND e VCC padronizados no texto.

- 1 (mais significativo) – 1 – 1 – 0 – 0 – 0 – 0 – 1 (menos significativo)
- 0 (mais significativo) – 1 – 1 – 0 – 0 – 1 – 0 – 1 (menos significativo)
- 1 (mais significativo) – 0 – 0 – 1 – 1 – 0 – 1 – 0 (menos significativo)
- 0 (mais significativo) – 1 – 0 – 1 – 0 – 1 – 0 – 0 (menos significativo)
- 1 (mais significativo) – 1 – 1 – 0 – 0 – 1 – 0 – 1 (menos significativo)

3. As três principais portas lógicas utilizadas em circuitos digitais – porta AND (lógica E), OR (lógica OU) e NOT (lógica NÃO ou INVERSORA). Considere a Tabela 1.2 a seguir:

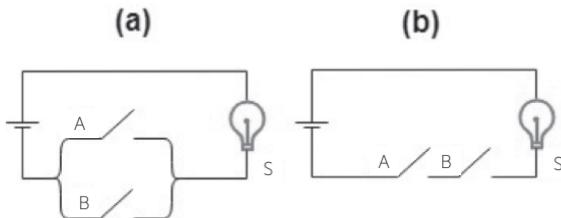
Tabela 1.2 | Blocos lógicos básicos

Porta	Símbolo	Tabela Verdade	Função Lógica	Expressão															
E AND		<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1	Assume valor 1 quando todas as entradas forem 1 e 0 nos outros casos	$S = A * B$
A	B	S																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OU OR		<table border="1"> <thead> <tr> <th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1	Assume valor 0 quando todas as entradas forem 1 e 1 nos outros casos	$S = A + B$
A	B	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NÃO (ou INVERSORA) NOT		<table border="1"> <thead> <tr> <th>A</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	S	0	1	1	0	Inverte o valor da variável de entrada A	$S = \bar{A}$									
A	S																		
0	1																		
1	0																		

Fonte: elaborada pelo autor.

Analise a Figura 1.16, considerando o texto-base:

Figura 1.16 | Circuito lógicos correspondentes



Fonte: elaborada pelo autor.

- a) Em (a) – corresponde a uma porta lógica OR. A lâmpada ficará acessa se e somente se ambas as entradas forem acionadas. Em (b) – corresponde a uma porta lógica AND. A lâmpada ficará acessa eventualmente quando ambas as entradas forem acionadas.
- b) Em (a) corresponde a uma porta lógica AND. A lâmpada ficará acessa se ambas as entradas forem acionadas. Em (b) corresponde a uma porta lógica OR. A lâmpada ficará acessa se A ou B ou ambas as entradas forem acionadas;
- c) Em (a) corresponde a uma porta lógica OR. A lâmpada ficará acessa se A ou B ou ambas forem acionadas. Em (b) corresponde a uma porta lógica AND. A lâmpada ficará acessa se e somente se ambas as entradas forem acionadas.
- d) Tanto (a) quanto (b) podem ligar a lâmpada independentemente da posição na qual estão as portas A e B.
- e) Não é possível abstrair portas lógicas observando o desenho. O que se conclui é que independentemente da posição das chaves as lâmpadas não serão acessas.

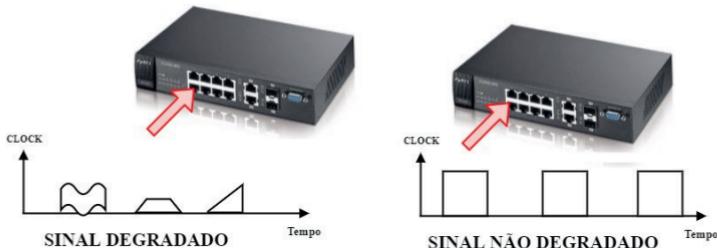
Seção 1.2

Características dos circuitos digitais

Diálogo aberto

Prezado aluno, relembrando: você foi contratado por uma empresa da área de telecomunicações para trabalhar com a montagem de produtos como modens, roteadores e switches. Ocorre que um dos switches está apresentando instabilidade quando conectado a mais de dois dispositivos. Você percebeu que o sinal de relógio na porta de saída de um dos switches está com a lógica correta. Ou seja, você consegue através do osciloscópio observar o sinal, mas, quando conectado com um segundo dispositivo, o sinal fica degradado. A Figura 1.17 ilustra esse problema.

Figura 1.17 | Exemplo de sinal degradado e não degradado



Fonte: elaborada pelo autor.

Você se questiona e começa a investigar o buffer que distribui o sinal de relógio entre o processador, a porta de saída e os outros dispositivos. Consiste em um NB3N551 (ON, 2012).

Em posse dessas informações, você inicia a sua pesquisa. Quais são as características DC importantes para o nosso sistema? Qual o FAN-OUT? O sistema está operando na temperatura correta?

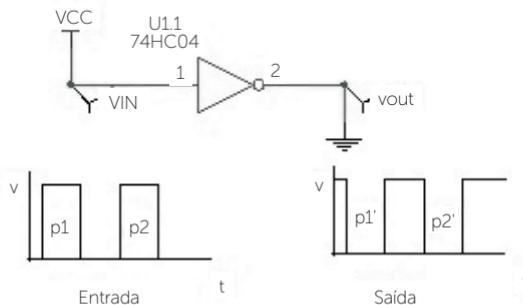
Esse cenário proposto tem por objetivo que você fique familiar com a leitura de folhas de dados de componentes e se imagine em uma situação que exija de você uma análise crítica. Portanto, tenha em mente que os possíveis problemas que a situação exemplifica não estão fora da realidade das pessoas que trabalham com circuitos digitais.

Dessa forma, elabore um relatório explicando os possíveis problemas encontrados e as soluções propostas. Para auxiliá-lo com essa tarefa, esta seção discorre sobre **margem de ruído** de um circuito digital. Em seguida, será analisado o **atraso de propagação** em um circuito digital. Após, iremos tratar sobre **dissipação de potência e produto atraso-potência** e, por fim, entenderemos o conceito de **FAN-IN** e **FAN-OUT** de um circuito digital. Lembre-se que os itens a seguir são empregados para caracterizar tanto o desempenho quanto a operação de circuitos lógicos. Utilize o texto-base como referência. Bons estudos.

Não pode faltar

Para começar, precisamos entender um pouco mais sobre o elemento básico no projeto de circuitos digitais – a porta inversora (ou porta NOT). A principal característica de uma porta inversora é a inversão lógica do sinal de entrada (V_{IN}) na saída (V_{OUT}), ou seja: quando V_{IN} é baixo (próximo a 0 Volts), a saída V_{OUT} obrigatoriamente será alta (próximo a V_{DD}) e vice-versa. A Figura 1.18 ilustra a uma porta inversora e o seu funcionamento.

Figura 1.18 | Porta inversora 74HC04 alimentada por uma fonte DC



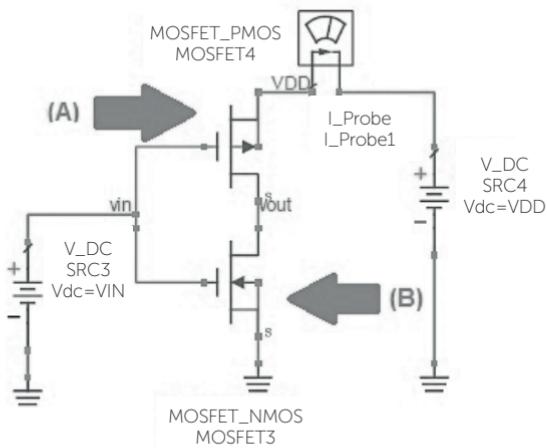
Fonte: elaborada pelo autor.

Nessa figura, é possível observar que os pulsos **p1** e **p2** na entrada são o inverso na saída do inverter (saída **p1'** e **p2'**).



Outra característica fundamental é entender como é montada eletronicamente uma porta lógica inversora. Dentre as diversas montagens possíveis para uma porta inversora, existe uma que utiliza dois transistores com tecnologia CMOS, sendo um **pMOS** e outro **nMOS** de forma complementar. A Figura 1.19 ilustra essa montagem.

Figura 1.19 | Porta Inversora CMOS

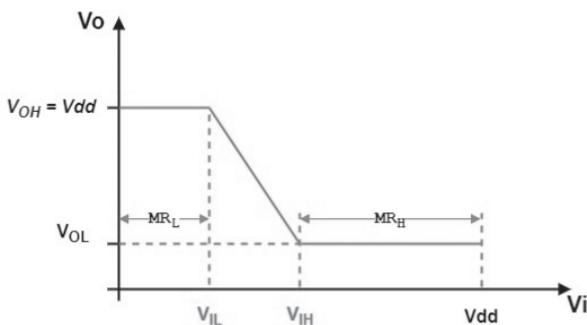


Fonte: elaborada pelo autor.

Na Figura 1.19, as setas apontam o transistor **pMOS** (em A) e **nMOS** (em B). Além disso, o sinal de entrada é descrito como **Vin** e a saída como **Vout**.

Conforme Sedra e Smith (2007), para quantificar a operação (ou funcionamento) de uma porta inversora devemos utilizar a característica (ou curva) de transferência de tensão (CTT). Essa curva é caracterizada por um gráfico tensão de saída versus tensão de entrada. Caro aluno, fique atento: estamos falando de uma porta inversora – se a entrada for baixa, a saída será alta e vice-versa. A Figura 1.20 apresenta a característica de transferência de tensão de um inveror.

Figura 1.20 | CTT de um inverter – parâmetros VOH, VOL, VIL e VIH



Fonte: adaptada de Sedra e Smith (2007).

Quadro 1.1 | Parâmetros CTT e suas respectivas funções

Parâmetros	Função
V_{OL}	Tensão de saída em nível baixo (<i>low-level output voltage</i>). Máximo nível de tensão de saída de um circuito lógico, no estado lógico 0.
V_{OH}	Tensão de saída em nível alto (<i>high-level output voltage</i>). Mínimo nível de tensão de saída de um circuito lógico, no estado lógico 1.
V_{IL}	Tensão de saída em nível baixo (<i>low-level input voltage</i>). Máximo nível de tensão requerido para o nível lógico 0 em uma entrada.
V_{IH}	Tensão de saída em nível alto (<i>high-level input voltage</i>). Mínimo nível de tensão requerido para o nível lógico 1 em uma entrada.

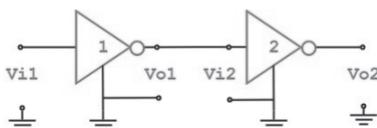
Fonte: elaborado pelo autor.

Observe atentamente na Figura 1.20 que o valor **VOH** não depende do valor exato de V_i , ou seja, quando V_i excede o valor de **VIL** a saída diminui e o inverter entra na região de amplificação linear ou também chamada de **região de transição**. Dessa forma, podemos concluir que: **VIL** é um parâmetro muito importante no CTT de um inverter e que **VIL** é o máximo valor que V_i pode assumir e ainda ser considerado pelo inverter um valor lógico **zero** (*low*). Já **VIH** é o valor mínimo que V_i pode assumir e ainda ser considerado pelo inverter como nível lógico **um** (*high*). Então, qual seria a CTT ideal de um inverter? Essa curva deve *(i)* maximizar as margens de ruído, *(ii)* distribuir igualmente entre as regiões de entrada de nível baixo (**VIL**) e nível alto (**VIH**), sendo o nível alto **VOH** o máximo valor de VDD e o nível baixo **VOL** o mínimo possível e próximo de zero. Em posse de toda essa introdução, conseguiremos entender como calcular a margem de ruído a seguir.

Margem de ruído

Analizando de forma estática uma porta inversora, segundo Sedra e Smith (2007), a grande vantagem em um circuito digital, quando comparado a um circuito analógico é a insensibilidade da saída deste circuito digital (V_{out}) com relação ao valor da entrada (V_{in}) do circuito digital. Tenha em mente que tanto um sistema digital quanto um sistema analógico sofrem oscilações de tensão, portanto quanto menos as entradas variarem mais preciso será o valor na saída do sistema. Considere agora dois inversores ligados em série da Figura 1.21.

Figura 1.21 | Inversores em série



Fonte: elaborada pelo autor.

Caso a saída do primeiro inverter esteja em nível alto ($Vo_1 = V_{OH}$), existe uma margem de segurança na entrada Vi_2 de forma que algum ruído elétrico não “atrapalhe” o segundo inverter. Isso ocorreria se o ruído por alguma razão levasse o valor Vi_2 abaixo do seu respectivo VIH . Essa margem é chamada de **margem de ruído para nível alto** ou **MRH**, sendo representada pela Equação 1.1:

$$MR_H = V_{OH} - V_{IH} \quad (\text{Equação 1.1})$$

De forma similar, se a saída do inverter estiver agora em nível baixo ($Vo_1 = V_{OL}$), o segundo inverter produzirá uma saída em nível alto, mesmo que o ruído altere o nível de Vo_1 até o limite de VIL . Essa margem é chamada de **margem de ruído para nível baixo** ou **MRL**, sendo representada pela Equação 1.2:

$$MR_L = V_{IL} - V_{OL} \quad (\text{Equação 1.2})$$

Reforçando, os parâmetros V_{OH} , V_{OL} , V_{IL} e V_{IH} definem a CTT de um componente (no caso um inverter), bem como as suas margens de ruídos e, portanto, medem a capacidade de esse componente tolerar variações de tensão nos sinais de entrada.



Pesquise mais

Caro aluno, circuitos integrados possuem diversas tecnologias de fabricação, como CMOS, bipolar, BiCMOS e GaAS. Pesquise mais a respeito dessas tecnologias em Tocci (2012), no capítulo 8. Descubra quais são os critérios para escolher uma família em detrimento da outra, como flexibilidade lógica, velocidade de operação, imunidade a ruidos, faixa de temperatura de operação, dissipação de potência e até mesmo o custo. Essas considerações são um bom começo para uma pesquisa mais aprofundada.

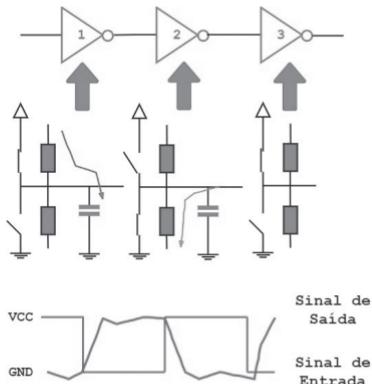
Atraso de propagação

Até o momento, analisamos a porta inversora de forma estática, ou seja, consideramos a porta inversora como uma caixa preta. Agora, vamos analisá-la de forma dinâmica, ou seja, considerando as respostas dadas pelos componentes internos da porta inversora. Vamos, então, analisar o seu comportamento dinâmico com relação ao atraso temporal entre o chaveamento de V_i (de alto para baixo e vice-versa) e a sua correspondente mudança na saída. Esse comportamento é chamado de **atraso de propagação**.

Tocci e Ronald (2012) definem de forma simples esse conceito: *como o tempo que leva para um sistema produzir uma saída apropriada após receber uma entrada*. Esse conceito é fácil de ser exemplificado: de forma geral, em um elevador, quando apertamos o botão para subir, após um certo tempo a porta fecha e depois subimos. Circuitos digitais reais sofrem do mesmo mal. Eles possuem um atraso com relação ao sinal de entrada e sua respectiva saída devido a fatores construtivos, o que é muito interessante, pois esses atrasos podem ser medidos.

De acordo com Sedra e Smith (2007), existem dois motivos para esses atrasos: os transistores que implementam as portas inversoras possuem tempos de chaveamento (entenda esse termo como ligar/desligar; VCC/GND ou transistor conduzindo/não conduzindo) diferentes de zero, e a capacidade entre a saída do inversor e o plano terra tem de ser carregada (ou descarregada conforme o caso). A Figura 1.22 ilustra os atrasos decorrentes do chaveamento.

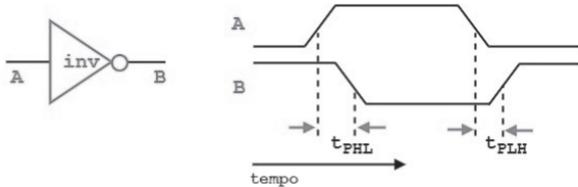
Figura 1.22 | Circuito inversor – chaveamento transistores



Fonte: elaborada pelo autor.

Na Figura 1.22, é possível observar que o sinal de saída possui um pequeno atraso quando comparado com o sinal de entrada. Portanto, temos um atraso na propagação do sinal. Esse atraso pode ser medido comparando o sinal de entrada com o de saída. Observe os dois sinais de uma inversora na Figura 1.23. Eles representam um pulso, cuja propagação possui seus respectivos **t_{PHL}** (tempo de atraso do estado lógico 1 para estado lógico 0) e **t_{PLH}** (tempo de atraso do estado lógico 0 para estado lógico 1).

Figura 1.23 | Medindo atraso de propagação porta inversora



Fonte: elaborada pelo autor.

Para Sedra e Smith (2007), esses atrasos são medidos entre os 50% das formas de onda de entrada e saída. O tempo de atraso de propagação médio (t_p) do sinal de saída é a média dos tempos de pulso t_{PHL} e t_{PLH} . A Equação 1.3 exemplifica esse tempo.

$$t_p = \frac{(t_{PHL} + t_{PLH})}{2} \quad (\text{Equação 1.3})$$

Todas essas informações devem estar contidas na folha de dados dos componentes. Consulte sempre que possível.

Dissipação de potência e produto atraso-potência

Caro aluno, nesse momento, vale lembrar algo bem característico de um circuito digital: ele aquece. Isso é fácil de perceber. Basta observar o momento em que o ventilador (também conhecido como cooler) do processador do computador é acionado. Certamente esse acionamento ocorre devido à necessidade de se dissipar a temperatura acumulada no processador, ou seja, dissipar o calor gerado pela atividade de chaveamento do processador.



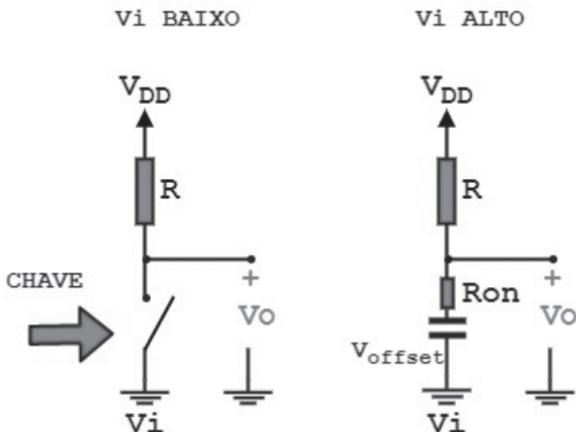
Refita

Já foi amplamente debatida a evolução da microeletrônica em textos anteriores. Nesse sentido, considerando que projetos de circuito digital VLSI têm por objetivo cada vez mais minimizar a área de silício e aumentar a quantidade de transistores, até que ponto podemos aumentar essa quantidade de transistores e diminuir a área? Qual implicação nas condições de PVT (Processo, Tensão e Temperatura)? Qual será o limite de tamanho que um transistor pode chegar? Refita sobre esses questionamentos, pois eles norteiam a microeletrônica há muito tempo.

Para Weste e Eshraghian (1993), a potência de uma porta lógica baseada na tecnologia CMOS pode ser dividida em quatro principais componentes: *(i)* potência estática (ou quiescente); *(ii)* potência dinâmica, devido às capacitâncias internas; *(iii)* potência dinâmica, devido às transições; e, por fim, *(iv)* potência dinâmica, devido às capacitâncias externas de carga do circuito.

Para o nosso estudo, iremos focar nas duas primeiras – *potência estática* e *potência dinâmica, devido às capacitâncias internas*. Primeiramente, observe a Figura 1.24. Essa figura representa uma porta inversora de forma simplificada. Quando a tensão **Vi** (Tabela 1.24 **Vi BAIXO**) for zero (Ground), a chave está aberta. E quando a tensão **Vi** (Tabela 1.24 **Vi ALTO**) for diferente de zero (Voffset) a chave está fechada e, portanto, conduzindo o sinal.

Figura 1.24 | Modelo simplificado de uma porta inversora



Fonte: adaptada de Sedra e Smith (2007).

Conforme Sedra e Smith (2007), quando Vi está no modo **BAIXO** a chave se encontra aberta e, portanto, sem chaveamento, mas, mesmo assim, existe potência dissipada no resistor R . Essa potência dissipada é conhecida como dissipaçāo de potēcia estática. O valor é dado pela Equação 1.4.

$$P_{\text{Estática}} = I_{\text{FUGA}} \times V_{DD} = \frac{V_{DD}^2}{R} \quad (\text{Equação 1.4})$$

Na segunda situação – Vi em modo **ALTO** –, sempre que a porta inversora chavear (ou seja, mudar para outro estado), é necessário que correntes transitem entre chaves para carregar (ou descarregar) as capacitâncias das cargas. Nesse momento, temos a dissipaçāo de potēcia dinâmica nas chaves. O valor dessa potêcia é dado pela equação 1.5.

$$P_{\text{Dinâmica}} = f \times C \times V_{DD}^2 \quad (\text{Equação 1.5})$$

Por fim, conforme Sedra e Smith (2007), existe outra medida da qualidade da tecnologia de fabricação do circuito CMOS. Corresponde ao chamado produto atraso-potêcia (ou DP – *Delay-Power Product*) e é dado pela Equação 1.5 (lembrando que t_P corresponde ao atraso de propagação médio).

$$DP = P_{\text{Dinâmica}} \times t_P \quad (\text{Equação 1.5})$$

Sendo que DP é dado em joules e, quanto menor o valor de DP, melhor a tecnologia quando analisado maior velocidade (t_p baixo) e menor consumo na porta ($P_{DINÂMICA}$). Assim, considerando uma tecnologia CMOS, na qual a maior parte da dissipação é dinâmica, o DP pode ser calculado pela Equação 1.6.

$$\sim C \times V_{DD}^2 \quad (\text{Equação 1.6})$$

Dessa forma, quanto menor o valor do DP para uma família, mais eficaz é a família lógica.

FAN-IN e FAN-OUT

O conceito a seguir trata da relação de quantidade de portas que um dispositivo de uma família de semicondutor pode ter tanto na entrada quanto na saída. O termo *fan-in* (leque de entrada) pode ser definido como o número de entradas que uma porta lógica possui, por exemplo: se uma porta AND tiver oito entradas, significa que ela possui *fan-in* igual a 8.

O conceito é similar para *fan-out*, mas não é igual. *Fan-out* (leque de saída) pode ser definido como o máximo de portas similares que uma porta pode acionar permanecendo dentro das especificações. Dessa forma, quanto mais entrada incluímos em uma porta de saída, menor será o V_{OH} e, consequentemente, menor a margem de ruído para nível alto ou M_{RH} .

Portanto, é importante saber quantas entradas em uma dada saída pode alimentar sem que isso comprometa a confiabilidade do sistema.

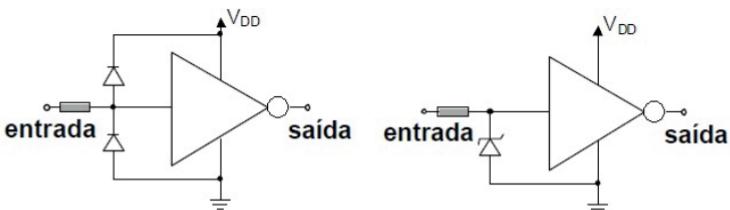


Exemplificando

Neste momento, é necessário exemplificar como proteger a entrada de um sistema. Primeiramente, entenda que, para transistores internos do CI, é fácil controlar as tensões de porta. Contudo, o VDD deve ter um valor baixo. No entanto, para transistores que estão ligados na entrada externa, a situação se torna um problema. Tendo em vista que as capacitâncias de porta em transistores CMOS são extremamente pequenas (cerca de dezenas de [fF] - fento-faraday), uma pequena quantidade de carga na entrada pode causar altas tensões e danos aos transistores. É o caso da energia estática acumulado no corpo humano. Isso é suficiente para danificar um CI CMOS.

Para proteger o CI CMOS dessas cargas e também de transientes de tensão (liga e desliga de fontes), são empregados circuitos limitadores de tensão com diodos. Esses circuitos são simples e, muitas vezes, já vêm integrados no CI. A Figura 1.25 exemplifica essa montagem.

Figura 1.25 | Exemplo de circuito de proteção de porta



Fonte: elaborada pelo autor.

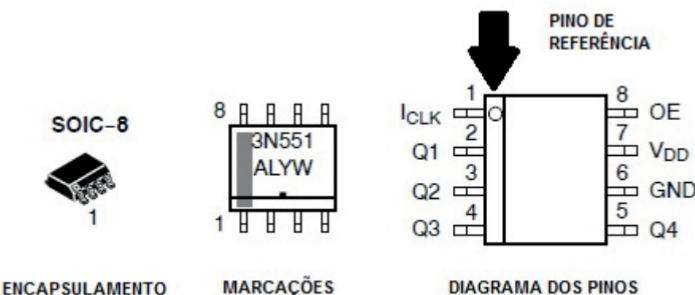
Caro aluno, a presente seção tratou de forma ampla aspectos importantes no nosso estudo sobre circuitos digitais. Tratamos sobre margem de ruído de um circuito digital e analisamos o atraso de propagação e a dissipação de potência e produto atraso-potência de um circuito digital. Por fim, entendemos o conceito de FAN-IN e FAN-OUT e a sua relevância em um projeto de circuito digital. Nesse contexto, fica o convite para você procurar extrapolar o que foi exposto aqui. converse com especialistas e, sobretudo, pratique. Boa sorte e bons estudos.

Sem medo de errar

Analizando o problema de instabilidade do switch quando conectado a mais de 2 dispositivos, existem alguns caminhos a serem analisados. O primeiro caminho você já tomou de forma correta – analisou a saída do sinal com o osciloscópio. Como são diversos sinais, você resolveu começar pelo principal sinal: o sinal de relógio que é distribuído para os outros dispositivos.

Analizando a folha de dados do dispositivo NB3N551, você pode começar descartando, por exemplo: V_{DD} – Positive Power Supply e a T_A – Operating Temperature Range, Industrial. A Figura 1.26 ilustra alguns detalhes do componente. Observe a importância da folha de dados de um componente. Consultando esta a folha, observe bem cada detalhe construtivo, como o encapsulamento escolhido, as marcações e, principalmente, onde fica o pino de referência (pino 1), o VDD e o GND.

Figura 1.26 | Diagramas do dispositivo NB3N551



Fonte: ON (2012).

Feita essa análise, o próximo passo é saber se a porta comporta mais de um dispositivo conectado em cada porta (Q1, Q2, Q3 e Q4).

A princípio, aqui está o erro. Quando um segundo dispositivo é ligado na porta de saída do NB3N551, o sinal fica degradado. Pesquisando a folha de dados, você consegue descobrir um pequeno detalhe, mas muito importante – o NB3N551 possui FAN-OUT de 4 –, mas isso não significa que cada porta possa conectar em mais de uma entrada de outro dispositivo sem que não seja analisado o SNR entre as portas. Extrapolare essa análise com relação aos sinais AC. converse com especialistas no assunto. Pesquise sobre buffers de isolamento. Inclua todos os dados coletados no seu relatório final.

Avançando na prática

Porta NAND 74AS20

Descrição da situação-problema

Um dos sistemas da empresa em que você trabalha necessita que se faça uma análise do número de portas de entrada NAND 74AS20 da fabricante Z que podem ser acionadas por outra porta de saída 74AS20 da mesma fabricante. Essa análise será fundamental, pois hoje as NAND que estão sendo usadas nos sistemas não serão mais produzidas pelo fabricante. Aqui vale mencionar algo muito comum no mundo da eletrônica: a DESCONTINUIDADE de componentes.

Com o decorrer do tempo, o processo de fabricação muito antigo vai ficando muito caro, por isso é vantajoso mudar para um processo, por assim dizer, mais novo. Assim, sempre que possível escolha um fabricante de componentes que vai garantir vida longa para o produto.

O que não foi o caso acima. Portanto, com essa análise, você vai descobrir se é possível mudar para a fabricante Z. Para isso, você separou alguns dispositivos e da série TTL 74xxx. Considere a Tabela 1.3 a seguir para fazer essa análise.

Tabela 1.3 | Portas lógicas TTL 74xxx, suas entradas e saídas

Séries TTL	Saídas		Entradas	
	<i>IOH</i>	<i>IOL</i>	<i>IIH</i>	<i>IIL</i>
74	-0.4 mA	16 mA	40 µA	-1.6 mA
74S	-1 mA	20 mA	50 µA	-2 mA
74LS	-0.4 mA	8 mA	20 µA	-0.4 mA
74AS	-2 mA	20 mA	20 µA	-0.5 mA
74ALS	-0.4 mA	8 mA	20 µA	-0.1 mA
74F	-1 mA	20 mA	20 µA	-0.6 mA

Fonte: elaborada pelo autor.

Resolução da situação-problema

Analizando a tabela acima, considere somente os valores do dispositivo que a empresa possui – no caso, é o 74AS. Os valores estão listados na Tabela 1.4. São eles:

Tabela 1.4 | Valores do componente 74AS

<i>IOH</i>	<i>IOL</i>	<i>IIH</i>	<i>IIL</i>
-2 mA	20 mA	20 µA	-0.5 mA

Fonte: elaborada pelo autor.

Inicialmente, vamos analisar os estados ALTO. Temos, então:

$$\text{FAN-OUT (nível ALTO)} = \frac{2mA}{20\mu A} = 100$$

Isso significa que o podem ser conectados até 100 dispositivos na saída do seu dispositivo FAN-OUT nível ALTO. Contudo, ainda temos o FAN-OUT nível baixo.

Nesse sentido, faça o mesmo cálculo para o nível BAIXO.

Caso o FAN-OUT nível BAIXO seja diferente do FAN-OUT nível ALTO, como pode ocorrer acima, devemos escolher o menor dos dois valores.

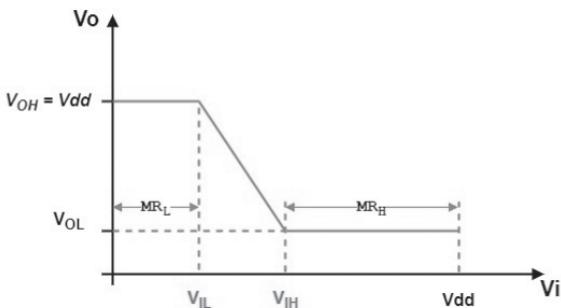
Lembrando que foi desenvolvimento da tecnologia de circuitos integrados como esse com que você trabalhou que permitiu o desenvolvimento da eletrônica digital e de toda essa modernidade que vivemos hoje. Bons estudos.

Faça valer a pena

1. Conforme Sedra e Smith (2007), para quantificar a operação (ou funcionamento) de uma porta inversora devemos utilizar a característica (ou curva) de transferência de tensão (CTT). Essa curva é caracterizada por um gráfico tensão de saída versus tensão de entrada.

Observe a figura a seguir:

Figura 1.27 | Curva de transferência de tensão (CTT) de uma inversora



Fonte: elaborada pelo autor.

Com relação aos parâmetros V_{OH} , V_{OL} , V_{IH} , V_{IL} , a alternativa correta é:

- a) V_{OL} consiste na tensão de entrada em nível baixo. Máximo nível de tensão de entrada de um circuito lógico, no estado lógico 0.
- b) V_{OH} consiste na tensão de saída em nível baixo. Mínimo nível de tensão de saída de um circuito lógico, no estado lógico 0.
- c) V_{IH} consiste na tensão de saída em nível baixo. Máximo nível de tensão requerido para o nível lógico 1 em uma entrada.
- d) V_{IL} consiste na tensão de saída em nível alto. Mínimo nível de tensão requerido para o nível lógico 1 em uma entrada.
- e) V_{DD} consiste na tensão mínima de operação de um circuito digital, em geral 0 (zero) volts.

2. A Figura 1.28 foi retirada da folha de dados do inversor CD4007 e exemplifica as características AC reais de um circuito inversor CD4007. Todas as informações a seguir foram retiradas da folha de dados desse componente. Considere um circuito MOS com todas as entradas protegidas contra descargas elétricas.

Figura 1.28 | Trecho da folha de dados do Inversor CD4007

TABLE 2. AC ELECTRICAL PERFORMANCE CHARACTERISTICS

PARAMETER	SYMBOL	CONDITIONS (NOTE 1, 2)	GROUP A SUBGROUPS	TEMPERATURE	LIMITS		UNITS
					MIN	MAX	
Propagation Delay	T _{PHL} T _{PLH}	VDD = 5V, VIN = VDD or GND	9	+25°C	-	110	ns
			10, 11	+125°C, -55°C	-	149	ns
Transition Time	T _{TTL} T _{TTHL}	VDD = 5V, VIN = VDD or GND	9	+25°C	-	200	ns
			10, 11	+125°C, -55°C	-	270	ns

NOTES:

1. CL = 50pF, RL = 200K, Input TR, TF < 20ns.
2. 55°C and +125°C limits guaranteed, 100% testing being implemented.

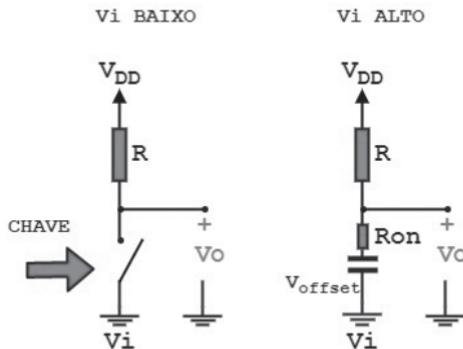
Fonte: National Semiconductor (1988).

Com relação ao comportamento dinâmico de uma porta inversora como a ilustrada no texto-base, podemos afirmar:

- a) A variável **t_{PHL}** corresponde ao tempo de atraso do estado lógico 1 para estado lógico 0 e a variável **t_{PLH}** corresponde ao tempo de atraso do estado lógico 0 para estado lógico 1.
- b) Esses atrasos são medidos entre os 50% das formas de onda somente na entrada do circuito, sendo a saída irrelevante para uma medida de tempo.
- c) O tempo de atraso de propagação médio (**t_P**) do sinal de saída é o produto do pulso alto para baixo (**t_{PHL}**) pelo tempo pulso baixo para alto (**t_{PLH}**).
- d) Os atrasos ocorrem devido à falta de padronização na fabricação dos circuitos digitais.
- e) Atrasos não são relevantes em um circuito digital, tendo em vista que eles operam em alta frequência e, portanto, podem ter o tempo desconsiderado

3. Para Weste e Eshraghian (1993), a potência de uma porta lógica baseada na tecnologia CMOS pode ser dividida em quatro principais componentes: **(i)** potência estática (ou quiescente); **(ii)** potência dinâmica, devido às capacitâncias internas; **(iii)** potência dinâmica, devido às transições; e, por fim, **(iv)** potência dinâmica, devido às capacitâncias externas de carga do circuito.

Figura 1.29 | Tensões de uma porta Lógica CMOS



Fonte: elaborada pelo autor.

Considerando a figura acima.

Com relação à dissipação de potência e produto atraso-potência, podemos afirmar de forma correta que:

- a) A potência de uma porta lógica baseada na tecnologia CMOS pode ser dividida em quatro principais componentes: (i) potência estática (ou quiescente); (ii) potência dinâmica, devido às capacitâncias internas; (iii) potência dinâmica, devido às transições; e, por fim, (iv) potência dinâmica, devido às capacitâncias externas de carga do circuito;
- b) Considerando a figura do texto-base. Quando Vi está no modo BAIXO, a chave se encontra fechada e, portanto, sem chaveamento, não dissipando potência no resistor R;
- c) Considerando a figura do texto. Quando Vi está no modo ALTO, sempre que a porta inversora abrir é necessário que correntes transitam entre chaves para somente carregar as capacitâncias das cargas. Nesse momento, temos a dissipação de potência estática nas chaves.
- d) Produto atraso-potência (ou **DP** – Delay-Power Product) não pode ser usado para medir a qualidade da tecnologia de fabricação de um circuito CMOS.
- e) Quanto maior o valor do **DP** para uma família, mais eficaz é a família lógica.

Seção 1.3

Visão geral das funções lógicas básicas

Diálogo aberto

Prezado aluno, relembrando: você trabalha em uma empresa da área de telecomunicações, atuando na montagem de produtos como modens, roteadores e switches, e percebeu que existe muita coisa a ser aprendida na área de circuitos digitais.

Surgiu uma pequena demanda para automatizar a contagem de produtos que saem do estoque. Sabendo disso, você pensou em montar um mostrador com dois displays de sete-segmentos, ou seja, ele vai contar do valor zero até o valor 99. Você tem em mente fazer com o mínimo de componentes. Para tanto, pensou em uma solução que utiliza um decodificador de 2 linhas para 4 linhas, um decodificador BCD para 7 segmentos e um multiplexador. Pesquisando diversas soluções, você concluiu que os componentes 74LS157; 74LS47 e 74LS139 encaixam como uma solução para o seu problema.

Em posse dessas informações, você inicia o seu trabalho. Quais são as características DC importantes do sistema? Como seria a lógica simplificada desse sistema? Qual a frequência que devemos colocar na entrada de seleção de dados para evitar cintilações, o famoso flickers (pesquise em outras fontes qual a frequência que o olho humano não percebe oscilações)?

Esse cenário proposto tem por objetivo que você se familiarize com a leitura de folhas de dados dos componentes 74LS157; 74LS47 e 74LS139. Além disso, possibilita que você comece a pensar em como fazer um projeto completo utilizando componentes. Essa situação não está fora da realidade das pessoas que trabalham com circuitos digitais.

Dessa forma, elabore um relatório explicando os limites de operação, como será a lógica implementada e qual a frequência de operação do sistema. Utilize o texto-base como referência e as folhas de dados dos componentes. Bons estudos!

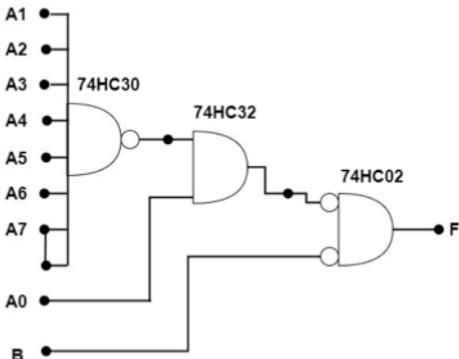
Não pode faltar

Caro aluno, até o presente momento vimos diversos tópicos sobre circuitos digitais. Trabalhamos, sobretudo, para entender o comportamento de cada porta lógica **AND**, **OR** e **NOT** (ou inversora) individualmente. Agora, daremos um passo maior. Estaremos combinando o uso dessas portas para formar circuitos mais complexos, ou seja, circuitos lógicos combinacionais. Tenha em mente que utilizaremos circuitos com funções lógicas fixas, como os circuitos integrados a seguir: **74XX283**; **74XX138**; **74XX148**; **74XX154**; **74XX47**; **74XX151**; **74XX139**; **74XX280**; **74XX85**; **74XX147**; **74XX157** e **74XX42**, para exemplificar nossos problemas. Alguns desses circuitos, por exemplo, são somadores, codificadores, decodificadores. Contudo, o foco deles é resolver problemas maiores, como o acionamento ou não de um alarme de incêndio, um aviso de abertura de uma sequência de portas ou a contagem da quantidade de pessoas em uma loja. Caso surja alguma dúvida, não hesite em pesquisar no conteúdo já trabalhado nas seções anteriores ou em buscar outras fontes, como a internet.

Circuitos com portas AND, OR e NOT

Prezado aluno, saiba que qualquer circuito lógico, independente da sua complexidade, pode ser construído usando as três portas lógicas básicas: AND, OR e NOT (ou inversora). Antes de entendermos como projetar um circuito, é importante defini-lo. Segundo Vahid (2008), um circuito consiste em uma interconexão de componentes elétricos, ou seja, estamos tratando aqui de diversos componentes conectados entre si. Vahid (2008) deixa ainda mais claro o que é um circuito lógico: *"Projetar um circuito que, para cada combinação diferente de entradas **a** e **b**, gere o valor apropriado de **f**".* A Figura 1.30 é um exemplo de circuito lógico que utiliza portas lógicas, havendo duas entradas **A** (entre 0 e 7) e **B**; e uma saída **F**. No meio, existem três portas lógicas (**74HC30**, **74HC32** e **74HC02**) formando o circuito digital.

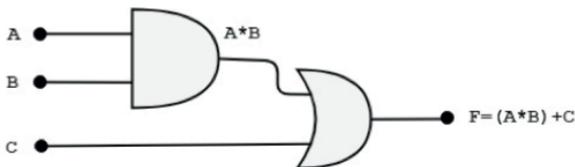
Figura 1.30 | Exemplo de circuito digital com portas de entrada e saída



Fonte: elaborada pelo autor.

Nesse sentido, usando as expressões booleanas de cada porta, podemos determinar facilmente a expressão lógica de saída. A Figura 1.31 ilustra uma expressão booleana utilizando duas portas lógicas, uma AND e uma OR.

Figura 1.31 | Expressão booleana



Fonte: elaborada pelo autor.

Observe que a saída é marcada pela função lógica (Função Lógica 1):

$$F = (A * B) + C \quad (\text{Função Lógica 1})$$

Caro aluno, se você observar com mais atenção a função lógica **F**, vai perceber que ela pode ser descrita da seguinte forma: entrada **A** conectada a uma entrada da porta lógica AND e entrada **B** conectada a outra entrada da porta lógica AND, cuja saída da porta AND é ligada na entrada da porta lógica OR. Já a entrada **C** é conectada na outra entrada OR. O resultado é a Função Lógica 1.



Precedência de operador: é comum um pouco de confusão quando se usam os operadores AND e OR. Existem algumas regras que devem ser obedecidas. Sempre que em uma expressão existir parênteses, essa será a operação realizada primeiro, por exemplo: $(A \cdot B) + C$; a expressão $(A \cdot B)$ será realizada primeiro. Outra regra: quando em uma expressão tiver lógica AND e OR, a operação que tiver a porta AND será realizada primeiro – a menos que não conflite com a regra anterior. Por exemplo: $A \cdot B + C$

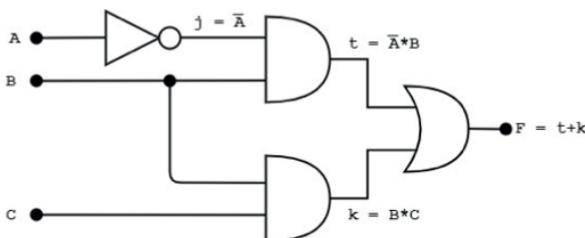
Segundo Tocci (2013), a melhor forma de entender o funcionamento de um circuito lógico é por meio de uma tabela-verdade. Dessa forma, uma tabela-verdade permite você: (i) analisar uma porta ou a combinação de cada conjunto lógico de cada vez e (ii) encontrar um erro de forma mais fácil. Observe a Figura 1.32. Ela descreve um circuito lógico combinacional completo. Analise em conjunto com a Tabela 1.5 e observe os valores intermediários entre as portas lógicas, pois eles são uma boa pista para solucionar a lógica final.



Pesquise mais

Pesquise mais sobre como montar uma tabela-verdade. Fique atento sobre como distribuir os valores binários nos sinais de entrada. É fundamental o amplo entendimento desse conceito. Pesquise outras fontes, por exemplo, o livro **Circuitos digitais – estude e use** (LOURENÇO et al., 1997, p. 51 a p. 54). Disponível em: <<https://biblioteca-virtual.com/detalhes/parceiros/5>>. Acesso em: 12 nov. 2017.

Figura 1.32 | Circuito lógico combinacional com portas AND, OR e NOT (inversora)



Fonte: elaborada pelo autor.

Tabela 1.5 | Tabela-verdade correspondente ao circuito da Figura 1.32

A	B	C	$j = \bar{A}$	$t = \bar{A} * B$	$k = B * C$	$F = t * k$
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	1	1

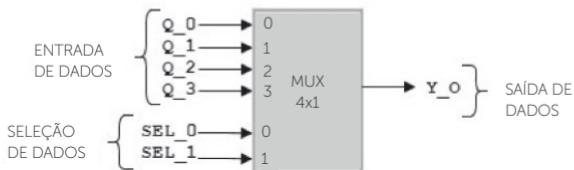
Fonte: elaborada pelo autor.

Se você construir esse circuito e o resultado não estiver correspondendo com o esperado, utilize a Tabela 1.5 para investigar onde está o problema. Lembre-se: é a tabela-verdade que lista as combinações possíveis de acordo com a lógica de entrada. Faça uso dela sempre que você precisar organizar a lógica.

Seleção de dados, codificação e decodificação

Segundo Floyd (2009), as 3 portas lógicas podem ser combinadas para criar sistemas digitais mais complexos. Uma das funções lógicas mais comuns são a seleção de dados, a codificação de dados e a sua correspondente decodificação. Quando estamos falando sobre seleção de dados, estamos nos referindo a um dispositivo chamado de multiplexador ou também conhecido como MUX. Multiplexadores consistem em dispositivos combinacionais, que permitem, por meio de uma chave seletora, que você selecione (escolha) um dos dados que estão na porta de entrada do circuito (que podem ser várias entradas de dados) e encaminhe para uma única porta de saída do circuito. A Figura 1.33 ilustra um multiplexador com 4 entradas de dados (**Q_0 à Q_3**) e uma saída (**Y_O**).

Figura 1.33 | Multiplexador 4x1 e dois bits de seleção



Fonte: elaborada pelo autor.

Observe um detalhe muito importante: a seleção dos dados. Essa seleção possui somente dois bits, o que permite selecionar no máximo quatro entradas de dados. Essa característica é fundamental em um multiplexador. Você pode observar todas as operações permitidas no multiplexador (Figura 1.33) na tabela a seguir.

Tabela 1.6 | Seleção de dados de uma multiplexador com 4 entradas e 1 saída

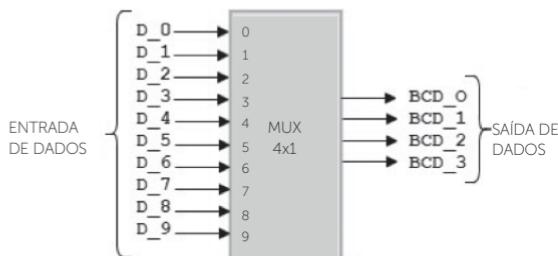
Seleção de Dados		Saída de Dados
SEL_1	SEL_0	Y
0	0	Q_0
0	1	Q_1
1	0	Q_2
1	1	Q_3

Fonte: adaptada de Floyd (2009, p. 347).

Observando a Tabela 1.6, fica mais claro que, se o valor binário **2b'01** for inserido na porta **SEL**, o valor de saída em **Y** será o valor em **Q_1**. Se em um multiplexador o sentido é de muitas portas para uma porta, no demultiplexador é o inverso – de uma porta para muitas portas.

Codificadores e decodificadores também são circuitos que operam utilizando lógica combinacional. Um circuito codificador recebe uma variável digital (tamanho **n bits**) como parâmetro de entrada e fornece na saída a mesma variável de entrada, mas representada de forma diferente (tamanho **x bits**), ou seja, codificada. O mesmo ocorre com um decodificador, mas no sentido inverso. Ou seja, na entrada do circuito inserimos uma variável digital (tamanho **x bits**) como parâmetro de entrada e obtemos na saída a mesma variável de entrada, mas agora decodificada (tamanho **n bits**). Um exemplo clássico é o codificador BCD (*Binary Coded Decimal* ou Decimal Codificado em binário). Nesse codificador, cada entrada representa um número decimal. Já a saída representa um número binário codificado para BCD (Figura 1.34).

Figura 1.34 | Codificador BCD



Fonte: elaborada pelo autor.

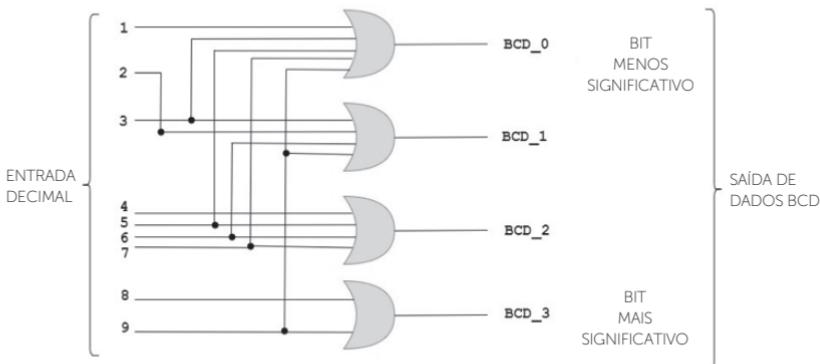
Tabela 1.7 | Conversão de decimal para binário

Dígito Decimal	CODIFICAÇÃO BCD			
	BCD_3	BCD_2	BCD_1	BCD_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Fonte: adaptada de Floyd (2009, p. 340).

Para contextualizar melhor, podemos implementar o circuito lógico necessário para a codificação de cada dígito decimal descrito na Tabela 1.7 para BCD, utilizando a porta lógica OR. Observe a Figura 1.35.

Figura 1.35 | Lógica do codificador BCD1



Fonte: adaptada de Floyd (2009, p. 341).

A Figura 1.35 resume da melhor forma possível as expressões lógicas da Tabela 1.7. A decodificação nada mais é que o processo inverso da codificação. Existem diversos codificadores no mercado. O CI 74HC147 é um exemplo de codificador de dígitos decimais de 1 a 9 para saídas BCD.

Somadores, subtratores e registradores

Caro aluno, esta seção é de fundamental importância. É através de somadores que as operações mais importantes na computação são realizadas. A adição em um sistema binário ocorre com a soma da primeira com a segunda parcela, resultando na soma e um bit de carro (o famoso “vai um” ou transbordo ou +1 ou carry).



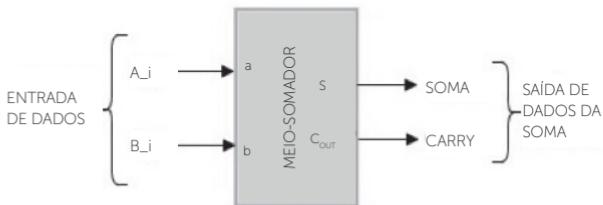
Refletia

Já sabemos que o sistema binário é uma representação numérica de base dois. A base utilizada por nós, seres humanos, é a base 10, de forma que as quantidades numéricas são representadas por 10 dígitos. Um dos problemas na interpretação desses números pelos computadores é a diferenciação entre números positivos e negativos e as consequências que tais dificuldades de sinalização trariam para as operações matemáticas são críticas. Agora, reflita como ocorre a soma ou subtração de negativos e números fracionários (conhecidos como ponto-flutuantes). Extrapolare o seu estudo pesquisando outras fontes.

Fonte: Floyd (2009).

Nesta seção, iremos detalhar o meio-somador. Ele é a base para outros somadores. A Figura 1.36 ilustra o símbolo do meio – somador. Já a Tabela 1.8 ilustra a lógica

Figura 1.36 | Símbolo de um meio-somador



Fonte: elaborada pelo autor.

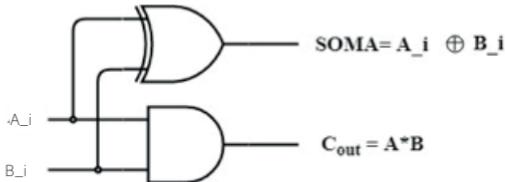
Tabela 1.8 | Meio-somador com carry de saída

A_i	B_i	C_{out}	Soma Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Fonte: adaptada de Floyd (2009, p. 347).

Observe que a saída Soma Σ só é 1 quando as entradas A_i e B_i não forem iguais. Essa soma é implementada com portas lógicas AND, uma porta OR e alguns NOT – essa lógica gera uma porta EX-OR ou OU EXCLUSIVO. A Figura 1.36 ilustra o diagrama lógico de um meio-somador.

Figura 1.37 | Diagrama lógico de um meio-somador



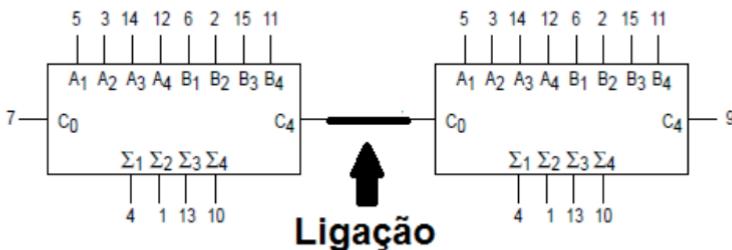
Fonte: adaptada de Floyd (2009, p. 341).

Com base nesse somador, é possível criar outros somadores mais complexos, como somador completo, somadores binários paralelo, somador de 4, 16 e N – bits.



Utilizando dois somadores 74LS283, é possível formar um somador paralelo de 8 bits. O que ocorre é que construtivamente o 74LS283 só permite a soma de 4 bits. Dessa forma, faz-se necessário adicionar mais um 74LS283 para a soma com 8 bits. Por exemplo, com as entradas $A_8A_7A_6A_5A_4A_3A_2A_1 = 10111000$ e $B_8B_7B_6B_5B_4B_3B_2B_1 = 10011110$ a soma dos dois números será $\Sigma_9\Sigma_8\Sigma_7\Sigma_6\Sigma_5\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 101010110$. Contudo, é necessária a ligação do bit de carro de saída do primeiro 74LS283 com o bit de carro de entrada do segundo 74LS283. A Figura 1.38 exemplifica essa ligação.

Figura 1.38 | Dois 74LS283 interligados pelos bits de carro



Fonte: elaborada pelo autor.

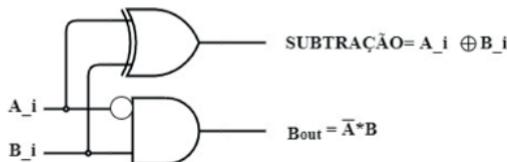
Caro aluno, subtraidores são obtidos de forma similar aos somadores. Observe a Tabela 1.9: a saída B_{out} corresponde ao bit de empréstimo (ou *Borrow*). A Figura 1.39 ilustra o diagrama lógico.

Tabela 1.9 | Meio-subtrator com empréstimo (*Borrow*) de saída

A _i	B _i	Bout	Soma
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Fonte: adaptada de Floyd (2009, p. 347).

Figura 1.39 | Diagrama lógico de um meio-subtrator



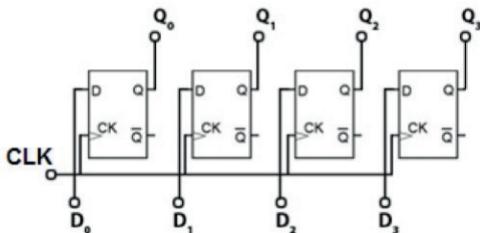
Fonte: adaptada de Floyd (2009, p. 341).

! Saiba mais

A subtração também pode ser executada utilizando complemento de dois, pois permite o uso dos circuitos mais simples, baratos e rápidos. O complemento de 1 e o complemento de 2 de um número binário são importantes porque eles permitem a representação de números negativos. Para saber mais sobre eles, leia a Seção 2.5 do livro **Sistemas digitais: fundamentos e aplicações** (FLOYD, 2009). Disponível em: <<https://biblioteca-virtual.com/detalhes/partceiros/5>>. Acesso em: 12 nov. 2017a.

Até o momento, tratamos de circuitos combinacionais. Agora, iremos tratar acerca de um componente: o registrador. Segundo Vahid (2008), um registrador é um componente sequencial que pode armazenar múltiplos bits. Um registrador é construído utilizando **N flip-flops**. A Figura 1.40 ilustra a estrutura interna de uma sequência de registradores e o seu respectivo símbolo.

Figura 1.40 | Registrador de 4 bits



Fonte: elaborada pelo autor.

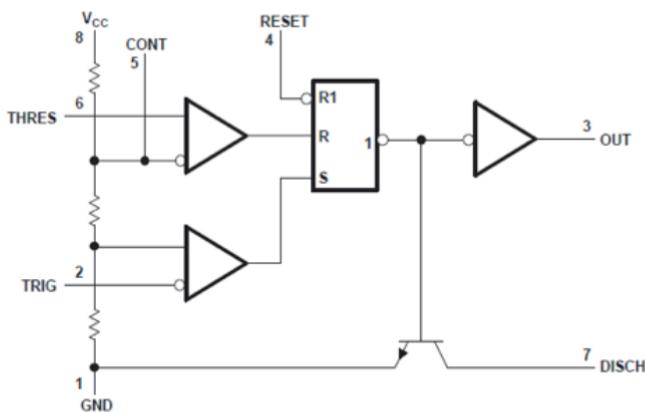
Observe na Figura 1.40 como um circuito sequencial é construído de forma simples, na qual através de quatro **flip-flops**, interconectados pelo mesmo sinal de relógio é possível contar até $2^4 = 16$, ou seja, do valor zero até o valor 15. Portanto registradores são fundamentais na computação. Ao longo do conteúdo, iremos abordar mais detalhes

sobre ele. Contudo, tenha em mente que em registrador é como se fosse uma barreira na qual o dado na entrada **Q** somente passa para a saída **D** quando um pulso ascendente ou descendente ocorre no sinal **CLK** (chamado de pulso de relógio ou pulso de *clock*).

Pulso de relógio

Prezado aluno, todo computador necessita de uma referência de tempo. Esse tempo é comumente chamado de relógio do sistema ou *clock* do sistema. Um exemplo de circuito que permite gerar uma referência de tempo é o componente 555. Esse componente operar como multivibrador monoestável, multivibrador astável (ou oscilador) ou multivibrador biestável, gerando pulso ascendentes ou descendentes, que por sua vez servem de referência de tempo para outro sistema. A Figura 1.41 ilustra como é o diagrama funcional de um dispositivo 555.

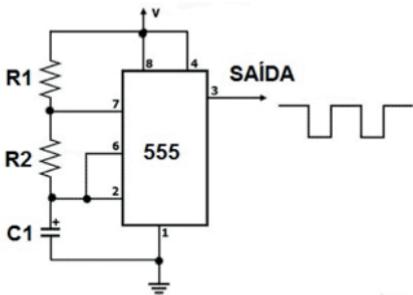
Figura 1.41 | Diagrama funcional 555



Fonte: <<http://www.ti.com/lit/ds/symlink/ne555.pdf>>. Acesso em: 11 out. 2017.

Como monoestável, o 555 funciona como disparador. As aplicações incluem temporizadores, chaves imunes a ruído, entre outras. Como astável (ou oscilador), o 555 funciona como oscilador. As aplicações incluem geradores de pulso, relógios de sistema, entre outras aplicações. Por fim, como biestável, pode funcionar como um flip-flop para interruptores imunes a ruído, por exemplo. Como um multivibrador astável, para gerar um pulso de relógio utilize o exemplo da Figura 1.42 e a Equação 1.7, a qual determina a frequência do pulso de saída. Fique atento à especificação de R1, R2 e C1.

Figura 1.42 | 555 como multivibrador astvel



Fonte: <<http://www.ti.com/lit/ds/symlink/ne555.pdf>>. Acesso em: 11 out. 2017

$$f = \frac{1.44}{(R_1 + 2R_2)C_1} \quad (\text{Equação 1.7})$$

Caro aluno, pesquise sempre a folha de dados dos componentes. Observe, por exemplo, como é possível variar o ciclo de trabalho (ou duty cycle) de um 555. Tenha em mente que esse componente possui limitações na frequência de saída. Procure aplicações para o temporizador, por exemplo, um pisca-pisca, ou projetos mais complexos, como botões imunes a ruído. Bons estudos e não se esqueça de extrapolar seu conhecimento.

Sem medo de errar

Caro aluno, como mencionado no início da unidade, surgiu uma pequena demanda para automatizar a contagem de produtos que saem do estoque da empresa na qual você trabalha. Como você já possui um bom conhecimento em sistemas digitais, você pensou em montar um mostrador com dois displays de sete-segmentos que vai contar do valor zero até o valor 99. Você sugeriu para essa solução utilizar um demultiplexador/decodificador de 2 linhas para 4 linhas, um decodificador BCD para 7 segmentos e um multiplexador. Seguindo essa configuração, os componentes que melhor se enquadram na sua proposta são os 74LS157; 74LS47 e 74LS139.

Neste sentido, a primeira coisa a ser feita é separar os componentes e as folhas de dados de todos eles. A operação básica pode ser descrita da seguinte forma.

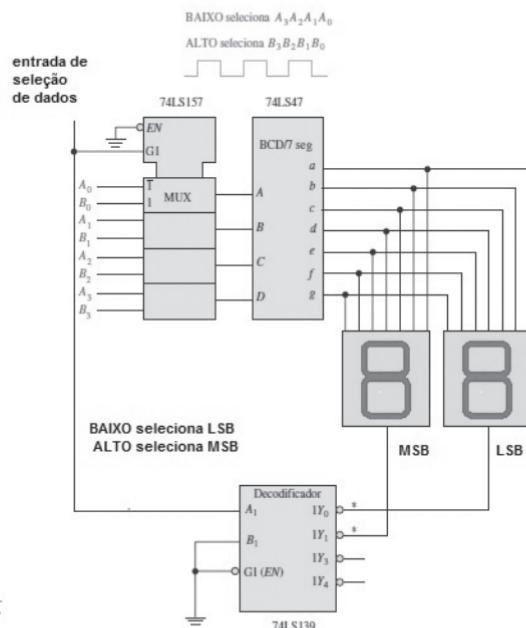
Observe, por exemplo, que basta um 74LS47 para decodificar dois dígitos BCD para 7 segmentos. Aqui, estamos propondo o método básico de multiplexação de displays. Esse método pode ser ampliado para todos os sistemas que possuam números como saída, ou seja, mostradores (Displays).

Analisando a folha de dados do multiplexador 74LS157, é possível observar que o LSB (bit menos significativo) e MSB (bit mais significativo) correspondem às entradas A e B, respectivamente. Fique atento nesse momento. Caso o número esteja fora do normal, é aqui que devemos analisar.

Por fim, observando o demultiplexador/decodificador de 2 linhas para 4 linhas 74LS139, você percebe que basta ligar o mesmo sinal de seleção dados na entrada A1 do 74LS139 e aterrkar as outras entradas. Pesquisando em outras fontes, você vai descobrir que a entrada de seleção de dados deve operar acima de 30Hz ou os olhos humanos vão perceber cintilações.

A figura a seguir ilustra a lógica simplificada do sistema proposto.

Figura 1.43 | Lógica simplificada do sistema



Fonte: adaptada de Floyd (2009, p. 352).

Extrapolde esse projeto e pesquise os sinais DC de funcionamento de cada componente, assim como sobre buffers de isolamento entre cada componente – por essa razão, a análise DC. Converse com especialistas no assunto e inclua todos os dados coletados no seu relatório final. Bom trabalho!

Avançando na prática

Oscilador 555

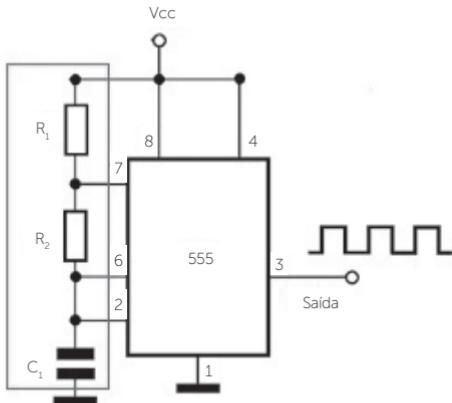
Descrição da situação-problema

Caro aluno, você desenvolveu um projeto de um contador de produtos que saem do estoque da empresa que trabalha. Contudo, existe a necessidade de gerar os pulsos de seleção de dados na entrada do multiplexador e do decodificador 2 linhas para 4 linhas. Para tanto, você pensou em utilizar um componente chamado 555. Dessa forma, qual seria o melhor modo de operação do 555 para funcionar como oscilador? Quais seriam os valores do capacitor e dos resistores para uma frequência de 144Hz? Analise a folha de dados e o conteúdo apresentado para solucionar esse problema.

Resolução da situação-problema

A melhor solução para essa situação é utilizar o componente 555 no modo astável. A Figura 1.44 exemplifica a montagem desse modo 555 astável.

Figura 1.44 | Componente 555 modo astável



Fonte: elaborada pelo autor.

Para resolver esse problema, devemos pensar em valores fáceis de encontrar no mercado, por exemplo: um resistor de 10K é muito comum. Nesse sentido, só precisamos encontrar o valor do capacitor. Observe a resolução a seguir.

$$f = \frac{1.44}{(R_1 + 2R_2)C_1} \Rightarrow 144 = \frac{1.44}{(10 \times 10^3 + 2 \times 10 \times 10^3) \times C_1} \Rightarrow C_1 \sim 50nF$$

Portanto, utilizando a equação de cálculo da frequência, é possível obter os seguintes valores para uma frequência de 144Hz: R1= R2 =10 k e C1=50 nF. Esses valores são valores comerciais e fáceis de encontrar no mercado. Analise o sinal de saída com um osciloscópio. Bons estudos!

Faça valer a pena

1. Todos os computadores necessitam de uma fonte de temporização para prover formas de onda de *clock* precisas. A seção de temporização controla toda a temporização do sistema e é responsável pela correta operação do hardware. A seção de temporização geralmente consiste em um oscilador controlado a cristal e contadores para divisão de frequência. "O uso de um oscilador de alta frequência que tem a frequência dividida para um valor menor proporciona maior precisão e estabilidade na frequência" (FLOYD, 2009, p. 422).

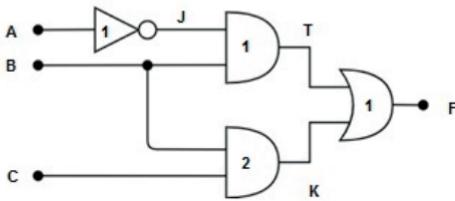
Tendo em vista um temporizador 555 configurado como modo astável (oscilador), determine a frequência de saída quando R1 = 2.2kΩ; R2 = 4.7kΩ e C1 = 0.022μF.

- a) 56.4 MHz
- b) 5.64 MHz
- c) 564 kHz
- d) 56.4 kHz
- e) 5.64 kHz

2. Segundo Vahid (2008), um circuito consiste em uma interconexão de componentes elétricos, ou seja, estamos tratando aqui de diversos componentes conectados entre si. Nesse sentido, um circuito combinacional é constituído por um conjunto de portas lógicas as quais determinam os valores das saídas diretamente a partir dos valores atuais das entradas.

Considere a figura a seguir:

Figura 1.45 | Circuito Combinacional



Fonte: elaborada pelo autor.

A lógica correspondente em J, T, K e F é:

- a) $J = \bar{A}; T = \bar{A} \times B; K = B + C; F = T + K$
- b) $J = \bar{A}; T = \bar{A} \times B; K = B + C; F = T + K$
- c) $J = \bar{A}; T = A \times B; K = B \times C; F = T + K$
- d) $J = \bar{A}; T = A \times B; K = B \times C; F = T + K$
- e) $J = A; T = A \times B; K = B \times C; F = T + K$

3. É através de somadores que as operações mais importantes na computação são realizadas. A adição em um sistema binário ocorre com a soma da primeira com a segunda parcela, resultando na soma e um bit de carro (o famoso “vai um” ou transbordo ou +1 ou carry).

Com relação à soma e subtração de duas parcelas, assinale a alternativa correta:

- a) A soma entre $A_8A_7A_6A_5A_4A_3A_2A_1 = 10111000$ e $B_8B_7B_6B_5B_4B_3B_2B_1 = 10011110$ a soma dos dois números será $\Sigma_9\Sigma_8\Sigma_7\Sigma_6\Sigma_5\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 101010110$.
- b) A subtração entre $A_8A_7A_6A_5A_4A_3A_2A_1 = 00111100$ e $B_8B_7B_6B_5B_4B_3B_2B_1 = 00011010$ produz como resultado sub= 00010011.
- c) Não é possível criar com meio-somadores outros somadores mais complexos.
- d) Em uma subtração, pode ocorrer o carry-out ou estouro de uma operação.
- e) Em uma soma, pode ocorrer o empréstimo ou borrow de uma operação.

Referências

BU. Huiming, 5 nanometer transistor inching their way into chips. IBM THINK Blog. 2017. Disponível em: <<https://www.ibm.com/blogs/think/2017/06/5-nanometer-transistors/>>. Acesso em: 22 set. 2017.

FLOYD, Thomas. **Sistemas digitais**: fundamentos e aplicações. São Paulo. Pearson Prentice Hall, 2009.

MIT Technology Review. (2017). These are the 10 breakthrough technologies you need to know about right now. [online] Disponível em: <https://www.technologyreview.com/lists/technologies/2017/> [Acesso em 19 Out. 2017].

NATIONAL Semiconductor. **CD4007M/CD4007C Dual Complementary Pair Plus Inverter datasheet**. Semiconductor Components Industries, LLC, 1988.

ON Semiconductor. **NB3N551 3.3 V / 5.0 V Ultra-Low Skew 1:4 Clock Fanout Buffer datasheet**. Semiconductor Components Industries, LLC, 2012.

SEDRA, Adel S.; SMITH, K. C. **Microeletrônica**. 5. ed. São Paulo: Pearson Prentice Hall, 2007.

TOCCI, Ronald J. **Sistemas Digitais**: Princípios e Aplicações. São Paulo: Ed. Pearson, 2003.

VAHID, F. **Sistemas Digitais**: Projeto, Otimização e HDLS. Porto Alegre: Bookman, 2008.

WESTE, N.H.E.; ESHRAGHIAN, K. **Principles of CMOS VLSI design**. 2. ed. Boston: Addison-Wesley Publishing Company, 1993.

Lógica combinacional

Convite ao estudo

Caro aluno, nas seções seguintes desta unidade iremos trabalhar o conhecimento acerca de lógica combinacional. Em linhas gerais, iremos construir o conhecimento sobre lógica programável, as linguagens de descrição de hardware e os simuladores envolvidos nessa etapa. Na seção 2.1, vamos dar início a esse conhecimento definindo o que é lógica programável. Em seguida, discutiremos sobre a gama de linguagens envolvidas nesse conhecimento e, por fim, iremos trabalhar os simuladores utilizados para testar os resultados gerados por esse conhecimento.

Na seção seguinte, iremos aprender sobre álgebra booleana e como é possível simplificar uma expressão. Isso somente é possível aprendendo lógica booleana, seus postulados e teoremas e como simplificar expressões booleanas.

Na sequência, na última seção, iremos mais a fundo na simplificação de expressões combinacionais. Aprenderemos sobre o diagrama de Veitch-Karnaugh ou simplesmente chamado de mapa de Karnaugh (lê-se Karnô).

Para compreender aonde queremos chegar, tenha em mente que sempre existe a necessidade do mercado de bons profissionais com conhecimento em lógica programável. Mas há um porém aqui: esses profissionais são raros, em especial os profissionais com conhecimento amplo sobre os mais variados assuntos, como IoT (do inglês *internet of things*), processamento de sinal digitais e telecomunicações. Dessa forma, na área de circuitos digitais, existem conceitos que devemos ter conhecimento, em lógica programável, se a nossa intenção for trabalhar com circuitos reprogramáveis, como CPLD (do inglês *complex programmable logic device* ou "dispositivo lógico complexo programável"), FPGAs (do inglês *field programmable*

gate array, em português “arranjo de portas programáveis em campo”) e SoCs (do inglês *system on chip*). Já a álgebra booleana e mapas de Veith-Karnaugh são a base para programar esses dispositivos.

Pense que você pode fazer parte desse time de profissionais. O caminho é árduo, mas, quando se tem vocação, o final é com um sentimento de vitória que não é possível descrever em algumas linhas. Mas o que é lógica combinacional? Como são essas linguagens? Como simular esses sistemas?

Essas perguntas podem ser respondidas no contexto desta unidade. Outras perguntas surgirão e também podem ser respondidas aqui. É esperado muitas outras dúvidas surjam e que você, aluno, tenha a curiosidade de pesquisar mais sobre o assunto. É esse o objetivo aqui – plantar uma semente e iniciar o conhecimento para que você aumente de forma sólida e independente o seu conhecimento.

Para tanto, vamos iniciar esta unidade de ensino contextualizando um cenário prático: considere que você já está há um bom tempo na área de microeletrônica. Já faz parte do time da linha de montagem da empresa de telecomunicações, na área de Modens para a internet. Você é uma pessoa extremamente curiosa e não se limita a aprender somente o básico. Observando o que está acontecendo ao seu redor, iniciou há 6 meses o estudo sobre lógica programável, inicialmente utilizando algumas ferramentas disponíveis na internet para gerar códigos em algumas linguagens como Verilog, VHDL (do inglês VHSIC *hardware description language*) e MyVHDL, em seguida usando umas ferramentas de testes e simulação. Contudo, em uma reunião geral, você soube que a empresa na qual você trabalha necessita customizar uma aplicação. Tendo em vista que essa aplicação necessita de hardware específico, ou seja, não é possível encontrar solução microprogramada com um custo acessível, a equipe de engenharia decidiu utilizar um CPLD. Essa é a sua chance de mudar de setor – sair da manutenção e ir para o setor de pesquisa e desenvolvimento. Como você é uma pessoa antenada e já havia estudado lógica programável, comentou com o seu chefe que teria interesse em participar do

projeto. Para testar a sua habilidade, foi-lhe dada a oportunidade de desenvolver um protocolo SERIAL-PARALELA utilizando HDL (do inglês Hardware Description Language). Como você irá programar? Será necessário alguma lógica booleana? Você terá de otimizar a lógica através de uma tabela-verdade?

Parabéns por ter extrapolado o conhecimento e iniciado o estudo com antecedência. Aproveite que o momento está a seu favor. Não perca essa oportunidade. A tarefa não é fácil. Exige muito estudo e dedicação. Para chegar ao time de P&D da sua empresa, você deve seguir uma trilha que passa pelo conhecimento, dedicação e profissionalismo. Esteja atento a todos os conceitos que serão apresentados aqui. E não se contente somente com o conhecimento apresentado aqui, extrapole. Mão à obra e boa sorte!

Seção 2.1

Lógica programável e simulação

Diálogo aberto

Caro aluno, esta seção tem o objetivo de direcionar o seu conhecimento sobre lógica programável, linguagem e simuladores de descrição de hardware (HDLs) e simuladores de circuitos digitais. Todo esse conhecimento é para conduzir você o mais próximo possível dos conceitos de circuitos digitais e da realidade do meio profissional. Os conceitos que serão expostos nesta seção são a base para você futuramente começar a projetar sistemas digitais.

Dessa forma, iremos entender como a lógica programável é estruturada, qual o contexto no qual está inserida, os dispositivos empregados e as suas arquiteturas. Na sequência, iremos dar início ao conceito de linguagem de descrição de hardware. Iremos compreender qual a necessidade dessas linguagens, quais são e finalizaremos esse tópico com um exemplo escrito utilizando duas linguagens, VHDL e VERILOG. Por fim, iremos trabalhar algo fundamental: a simulação de circuito descritos em HDL e circuitos digitais. Para isso, vamos utilizar ferramentas gratuitas e on-line disponíveis. Esta seção, apesar de introdutória, é fundamental para compreendermos como iremos simular circuitos digitais.

Nesse contexto, uma excelente maneira para entender essa proposta de aprendizado é imaginar o seguinte cenário: a empresa na qual você trabalha necessita customizar uma aplicação. Tendo em vista que essa aplicação necessita de hardware específico, ou seja, não é possível encontrar solução microprogramada com um custo acessível, a equipe de engenharia decidiu utilizar um CPLD. Essa é a sua chance de mudar de setor – sair da manutenção e ir para o setor de pesquisa e desenvolvimento. Como você é uma pessoa antenada e já havia estudado lógica programável, comentou com o seu chefe que teria interesse em participar do projeto. Para testar a sua habilidade, foi-lhe dada a oportunidade de desenvolver inicialmente uma lógica de teste (*testbench*) de um circuito combinacional que alerta a conexão ou a desconexão de uma fonte de alimentação em um dos servidores da empresa após 5 segundos. A solução é

puramente combinacional, mas necessita ser testada. Como você irá programar? Será necessário o uso de ferramentas de simulação? Qual a linguagem que você irá adotar? E qual a lógica que será simulada? Existem muitas incógnitas, mas, se a sua simulação for bem pensada, você irá obter todas as suas respostas.

Lembre-se de que esse cenário é uma proposta de aprofundamento ao seu conhecimento sobre circuitos digitais. E que todo o seu conhecimento será feito com base em bastante leitura e pesquisa de forma que ele seja utilizado no futuro no seu dia a dia. Dessa forma, devemos documentar todo esse aprendizado. Produza, então, um relatório com a sua proposta de solução. Utilize a sua capacidade analítica e pontue os principais aspectos que você levantou. Use figuras, mas não se esqueça de explicá-las, bem como apontar as referências (devemos dar o mérito para quem fez, é o mais justo). Sempre é bom lembrar: use uma linguagem científica, evite ao máximo o gerúndio. Existe um provérbio bávaro (região ao sul da Alemanha chamada de Baviera) que diz: *De que adianta correr quando se está no caminho errado?* Por isso, utilize as ferramentas de simulação para validar o caminho correto da sua solução.

Nas seções a seguir, será apresentada a base para esse conhecimento, mas não fique limitado somente a essa fonte. Pesquise, converse e troque informações com outras pessoas. O mercado está favorável para funcionários proativos e esforçados. Boa sorte e boa simulação!

Não pode faltar

Daremos início a um assunto muito interessante: lógica programável, suas linguagens e simuladores de circuitos digitais. Nesse sentido, precisamos entender qual é o contexto em que um projeto de sistema digital está inserido. Um projeto de sistema digital consiste em determinar o que chamamos de função lógica. Uma função lógica reproduz a lógica de funcionamento do sistema de acordo com estímulos que esse sistema recebe. É nesse momento que podemos criar um circuito lógico mais complexo que execute essas funções lógicas a partir de circuitos lógicos simples. Um exemplo bem simples é o controle da abertura da porta de um elevador. Pense no seguinte cenário envolvendo um elevador:

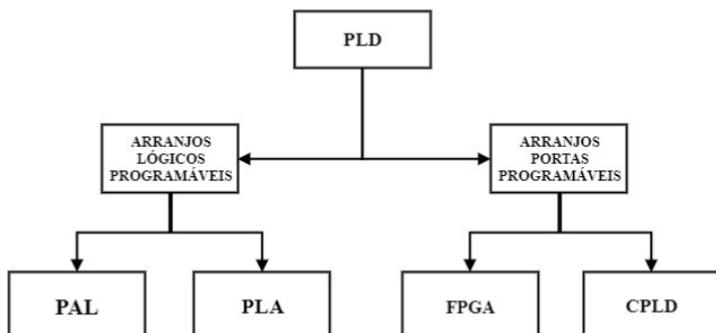
- O elevador está no térreo.
- Uma pessoa chama o elevador no sétimo andar.

Quando a porta do elevador deve abrir? De forma simplificada, a porta deve abrir somente quando: (i) o elevador estiver no sétimo andar; (ii) o elevador estiver parado e (iii) o elevador estiver alinhado com a porta. Qualquer outra situação pode colocar em risco a pessoa que chamou o elevador. A lógica programável vem para ajudar a construir funções lógicas que podem ser utilizadas em sistemas como o de controle de elevadores.

Lógica programável

Para entender lógica programável, temos de entender os dispositivos que estão ligados ao conceito. Os componentes que utilizam lógica programável possuem em sua estrutura interna centenas ou mesmo milhares de circuitos como portas lógicas, *flip-flops* e registradores, entre outros. Esses circuitos são interligados internamente através de estruturas bem complexas de barramentos de interconexão. *Preste atenção neste momento: essas interconexões são programáveis*. Essa é grande vantagem da lógica programável: de acordo com a necessidade do projeto, podemos mudar completamente o hardware que possuímos. Você pode pensar da seguinte forma: imagine essas interconexões como fusíveis, que, conforme o projeto do seu sistema, você pode ou não queimar as ligações entre uma porta lógica e outra. Longe de usarmos fogo para isso, usamos um software para ligar ou desligar essas interconexões. Por isso, chamamos de lógica programável.

Figura 2.1 | Organização dos dispositivos lógicos programáveis



Fonte: elaborada pelo autor.

PAL (do inglês *programmable array logic*) e PLA (do inglês *programmable logic array*) são dispositivos com blocos pré-fabricados com lógica AND e OR, ou NAND e NOR. Uma vez que a lógica seja programada em cada dispositivo, ela não pode mais ser alterada. Isso ocorre porque as interligações usadas sofrem um processo de ligação que provoca um curto-círcito na interconexão. Dispositivos do tipo PLA implementam lógica de soma de produtos. Já dispositivos PAL implementam o produto das somas.



Pesquise mais

Dispositivos com arranjos lógicos programáveis foram amplamente utilizados no início da computação, mas possuíam a desvantagem que, depois de gravados, não poderiam ser regravados. Nesse sentido, pesquise soluções que possam utilizar PALs PLAs. Uma boa referência é a folha de dados do dispositivo PAL16R8 e o PLA 82s100. Analise com calma detalhes como a descrição funcional, as características elétricas e, sobretudo, os diagramas de interconexão.

Já os “arranjos de portas programáveis em campo” consistem em circuitos integrados projetados para serem configurados após a fabricação – por isso, “programável em campo”. A grande maioria dos microchips que encontramos em nosso dia a dia, como os circuitos das televisões, celulares, entre outros, já vem como blocos de hardware pré-programados (ASIC). Portanto, tem as suas funcionalidades já definidas no ato de fabricação. Os FPGAs e os CPLDs permitiram, então, que as funcionalidades desses novos sistemas fossem definidas exclusivamente pelos usuários e não mais pelos fabricantes.



Refletia

Existem dispositivos chamados OTPs (do inglês *one-time programmable*), os quais não podem ser atualizados e reutilizados após a programação inicial. Isso não ocorre com FPGAs e CPLDs. Nesse sentido, qual a vantagem de utilizar um sistema em que a lógica não pode ser mais modificada? Isso tem algo relacionado com o custo de final de cada dispositivo? Qual a vantagem e desvantagem em utilizar FPGA ao invés de CPLD? Reflita sobre dois aspectos: o custo dos dispositivos OTPs e FPGAs e a diferença das tecnologias OTP, FPGA e CPLD.

Linguagem de descrição de hardware

Vamos agora entender como é possível descrever um circuito através de uma linguagem, ou como os especialistas falam: *vamos entender como é possível descrever hardware*. Inicialmente, essas linguagens foram a melhor forma encontrada para dar um significado mais preciso para cada um dos elementos de um circuito. Assim, quando as ferramentas de software fizerem a leitura da descrição, saberiam exatamente o que fazer. Observe a Figura 2.2: como uma aplicação rodando em um computador consegue entender que uma porta inversora deve inverter um sinal? E como essa aplicação entende que a saída do inverter deve ser ligada na porta AND? Só é possível esse entendimento através de linguagens especializadas que descrevem toda essa estrutura – são então as **linguagens de descrição de hardware**, ou **HDLs** (*Hardware Description Languages*).

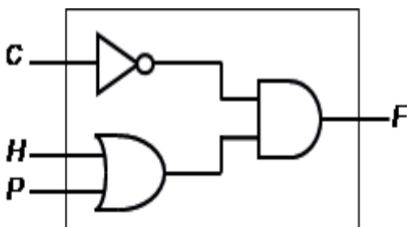


Vocabulário

Fique atento como os especialistas se referem a algo. Isso facilita o início de uma conversa e, sobretudo, demonstra conhecimento sobre o assunto. Em HDL, o próprio nome já dá a dica de como referenciar a um circuito descrito pela linguagem. O termo **estrutura** se refere a um circuito, significando também a interconexão entre os componentes.

Existem algumas linguagens básicas para descrever esses circuitos. Vamos entender duas delas inicialmente: (i) VHDL e (ii) VERILOG. Vamos utilizar o circuito da Figura 2.2 como exemplo base.

Figura 2.2 | Lógica combinacional de abertura ou fechamento de uma porta



Fonte: adaptada de Vahid (2008, p. 466).

A Figura 2.2 descreve uma lógica combinacional de um sistema que controla a abertura ($F=1$) de uma porta e o fechamento ($F=0$) de correr. A entrada **P** indica a presença de uma pessoa ($P=1$) ou a ausência ($P=0$). A entrada **H** significa que a porta está no modo manual. Se **H** for 1, a

porta permanece aberta e, se **H** for **0**, a porta funcionará de acordo com a presença de pessoas. Por fim, a entrada **C** significa que a porta deve permanecer fechada (**C=1**). Esse é o caso da loja fechada no fim do expediente; se **C=0**, a porta funciona de acordo com as entradas **H** e **P**. Vale lembrar que os dois últimos casos ocorrem somente quando o supervisor da loja autoriza. A descrição do circuito da Figura 2.2 em VHDL pode ser vista no Quadro 2.1 e em VERILOG no Quadro 2.2

Quadro 2.1 | Descrição do circuito da Figura 2.2 em VHDL

Inclui bibliotecas básicas da linguagem VHDL	<pre>library ieee; use ieee.std_logic_1164.all;</pre>
Descreve o nome do circuito ControlePorta e as respectivas portas de entrada e saída	<pre>entity ControlePorta is port (C, H, P: in std_logic; F: out std_logic); end ControlePorta;</pre>
Introduz no circuito ControlePorta as estruturas de cada componente INV, OR2 e AND2	<pre>architecture rtl of ControlePorta is</pre>
Fios internos para interligação entre componentes	<pre>component INV port (x: in std_logic; f: out std_logic); end component;</pre> <pre>component OR2 port (x, y: in std_logic; f: out std_logic); end component;</pre> <pre>component AND2 port (x, y: in std_logic; f: out std_logic); end component;</pre>
O circuito ControlePorta consiste internamente nos circuitos inv_1 ; OR2_1 e AND2_1	<pre>signal n1, n2: std_logic; --fios internos</pre> <pre>begin</pre> <pre>inv_1: INV port map (x => C, f => n1); OR2_1: OR2 port map (x => H, y => P, f => n2); AND2_1: AND2 port map (x => n1, y => n2, f => F)</pre>

Fonte: adaptado de Vahid (2008, p. 466).

Quadro 2.2 | Lógica combinacional de abertura ou fechamento de uma porta

Descreve o nome circuito ControlePorta e define as portas de entrada e saída	<pre>module ControlePorta (C, H, P, f); input C, H, P; output f;</pre>
Fios para interconexão entre os circuitos	<pre>wire n1, n2;</pre>
O circuito ControlePorta consiste internamente nos circuitos inv_1 , or2_1 , and2_1	<pre>INV inv_1 (C, n1); OR2 or2_1 (H, P, n2); AND2 and2_1 (n1, n2, f);</pre>

endmodule

Fonte: adaptado de Vahid (2008, p. 467).

As duas linguagens representam o mesmo circuito. Contudo, podemos observar que VERILOG é mais sucinta que VHDL. Aqui, é importante observar que ambas são importantes e que possuem algumas características em comum. O que ocorre é que em projetos em HDL necessariamente você deve descrever o nome do projeto, as portas de entrada e saída do circuito, os fios de interconexão, os circuitos que são utilizados e, por fim, o comportamento interno do circuito (no nosso caso, o sistema de controle de porta). Caro aluno, fazemos o convite para você analisar com calma todos os aspectos da linguagem ilustrados nos quadros apresentados. Procure outras fontes e pesquise outras linguagens, como *SystemVerilog*, *SystemC*, *MyVerilog*. Converse com especialistas sobre o assunto. As HDLs são linguagens poderosas principalmente para quem curte bons desafios na área da tecnologia e vive antenado com o futuro.

Simuladores de HDL

Existem diversas ferramentas de descrição e simulação de HDL. O que ocorre é que muitas dessas ferramentas estão voltadas para uma tecnologia específica de FPGAs. É o caso de duas grandes empresas que possuem uma boa fatia do mercado – a Xilinx e a Altera. Existem outras, mas vamos focar nessas por causa das ferramentas de software que elas utilizam. Essas duas empresas possuem suas respectivas IDEs de desenvolvimento de aplicações em HDL.



Vocabulário

IDE é uma sigla, do inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado. É o software utilizado para trabalhar com os dispositivos (FPGAs) de cada empresa.

Essas IDEs integram diversos softwares que permitem tanto o desenvolvimento e a simulação, quanto a integração com os FPGAs. A Tabela 2.1 lista algumas dessas plataformas de desenvolvimento e suas respectivas IDEs. Caro aluno, essas plataformas são um bom início para começar no mundo dos HDLs. Procure informar-se sobre como usá-las e como adquiri-las.

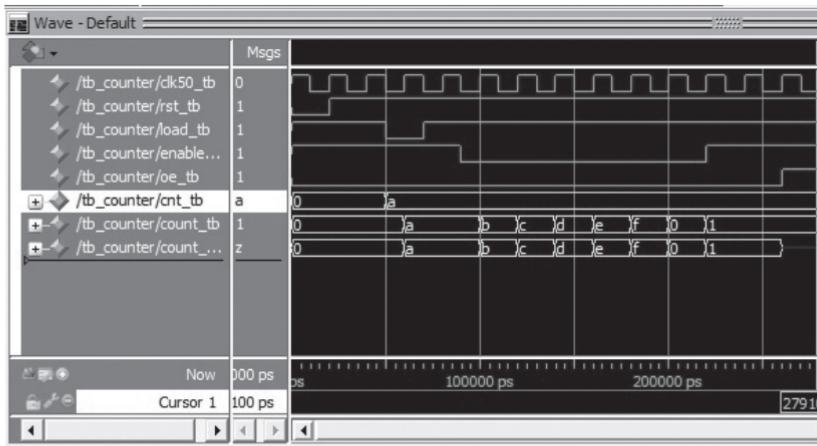
Tabela 2.1 | Plataformas de desenvolvimento e respectivas IDEs

Empresa	Plataforma	IDEs
ALTERA	DE0-Nano, DE2	Quartus II; ModelSim
XILINXS	Virtex-7; Spartan3	ISE, ISim

Fonte: elaborada pelo autor.

Com relação à simulação, uma ferramenta muito interessante para testar os circuitos desenvolvidos é o ModelSim (MODELSIM, 2017). Essa ferramenta possui uma versão gratuita bem interessante que pode ser explorada. Ela permite, por exemplo, simular os sinais de saída do circuito desenvolvido como os descritos na Figura 2.3.

Figura 2.3 | Captura de tela da ferramenta ModelSim

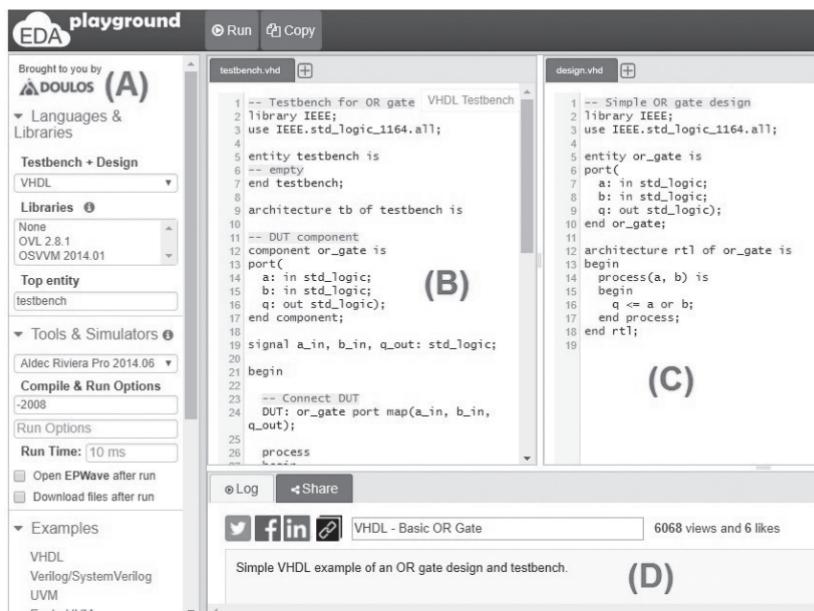


Fonte: elaborada pelo autor.

Observe na Figura 2.3 os sinais à esquerda. Eles correspondem aos sinais do circuito testado, no caso um contador. Os gráficos à direita são os resultados da simulação do circuito. Com essas ferramentas, você pode analisar erros, acertos ou mesmo características do circuito desenvolvido. É fundamental o teste. Sem um bom teste, não existe garantia que o circuito possa funcionar em um FPGA. Ainda assim, existem diversas iniciativas gratuitas na internet que exigem somente um pequeno cadastro. Uma delas e muito interessante é a EDA Playground. Com essa ferramenta on-line, você começar a testar suas habilidades em HDLs. Ela permite tanto a criação de circuitos em HDL quanto a

simulação utilizando várias metodologias de testes. A Figura 2.4 ilustra a interface da EDA Playground disponível na web. Em (A), está a área onde é possível escolher a linguagem HDL que será trabalhada. Em (B), a área onde é escrito o código que irá testar o circuito descrito em (C) e, por fim, (D) corresponde à área onde são impressos os resultados e os erros da simulação.

Figura 2.4 | Captura de tela do ambiente de desenvolvimento e simulação EDA Playground



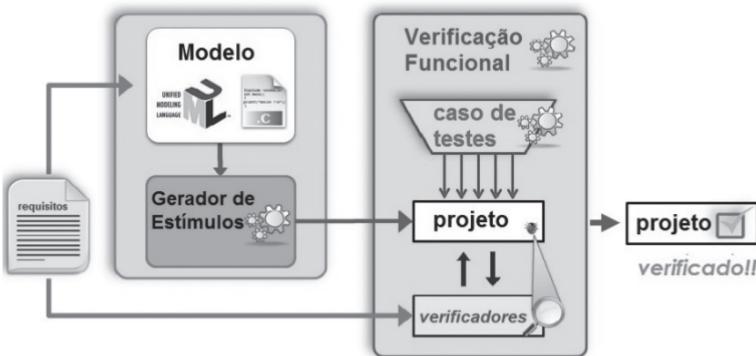
Fonte: EDA Playground.

Existem outras ferramentas de simulação de HDL disponíveis no mercado; basta procurar na internet pelo termo: **HDL simulator**. Em sua grande maioria, essas ferramentas só precisam de um pequeno cadastro e possuem exemplos de aplicações muito úteis para começar um projeto maior. Explore essas ferramentas testando seus exemplos e adaptando-os para o contexto da disciplina.



Existem diversas metodologias de verificação de código. O exemplo do modelo apresentado é o mais comum. Na metodologia (Figura 2.5), os estímulos são gerados por um modelo-padrão (Bloco Modelo). Esses sinais são inseridos no projeto (Bloco Projeto) e, durante a verificação funcional, testes são executados. Os sinais de saída são verificados (Bloco Verificadores), e, caso os requisitos do projeto sejam cumpridos, o projeto é dado como VERIFICADO e aprovado para ser usado.

Figura 2.5 | Exemplo de arquitetura de verificação

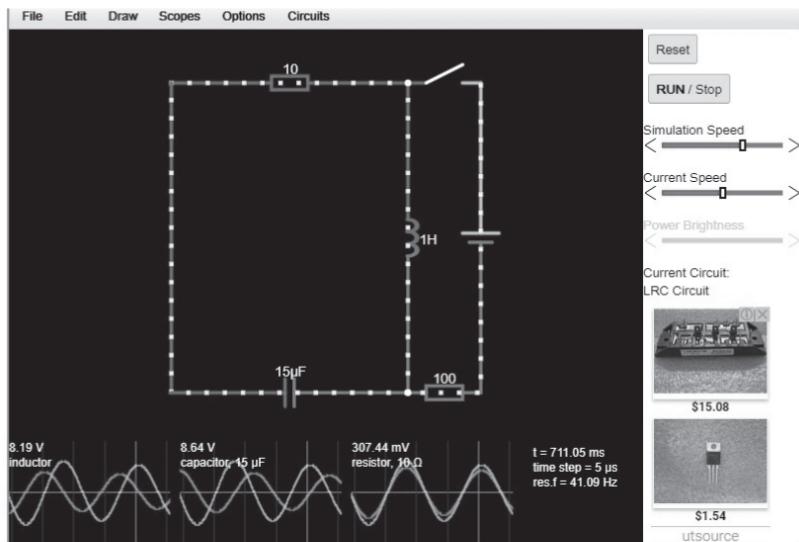


Fonte: elaborada pelo autor.

Simuladores de circuitos digitais

Caro aluno, até o projetista mais experiente necessita testar os projetos, e a melhor forma é em ambientes computacionais. As razões são óbvias. Você não precisa comprar os componentes e pode variar a quantidade sem que isso implique em custo e tempo de projeto ou mesmo que, ao testar modificações no projeto, os erros levem à queima de todo o circuito – portanto, ambientes de simulações computacionais são uma grande revolução na área de circuitos digitais. Temos como simuladores mais tradicionais o PSPICE, TINA-TI, Multisim-NI e Proteus. Todos são fabulosos, mas dependem de uma instalação no computador e da licença de uso (algumas licenças custam muito caro mesmo). Agora, imagine uma ferramenta on-line, gratuita e que você possa usar no seu navegador de internet favorito. O Circuit Simulator version 2.1.11js (CIRCUIT SIMULATOR, 2017) é um bom exemplo dessa tendência. A Figura 2.6 mostra a tela inicial do simulador.

Figura 2.6 | Simulador de circuito digital – circuit simulator



This is an electronic circuit simulator. When the applet starts up you will see an animated schematic of a simple LRC circuit. Red color indicates negative voltage. The moving yellow dots indicate current.

Fonte: Circuit Simulator (2017).

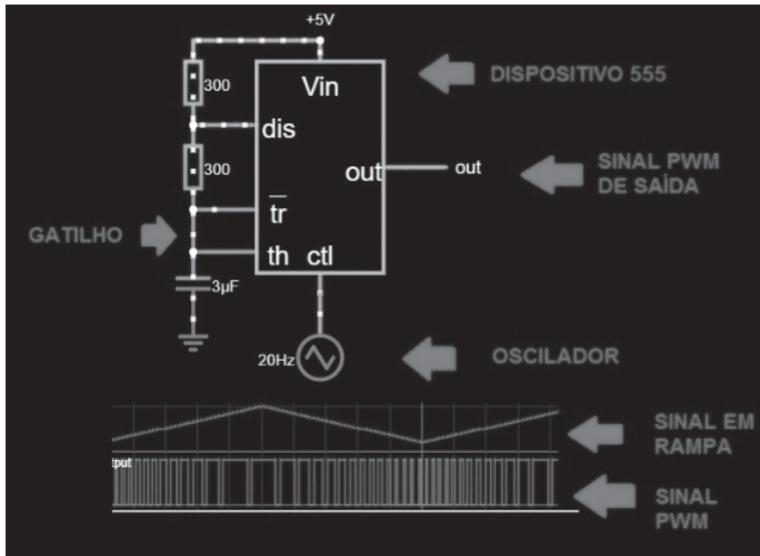
O exemplo da Figura 2.6 consiste em um circuito RLC – resistor, indutor e capacitor. Como você pode observar, depois de montado o circuito é possível variar os valores de cada componente e analisar os sinais de saída. Esse simulador é um dos muitos disponíveis na internet, como o EasyEDA, PartSim, CircuitLab, entre outros. Escolha o seu favorito.



Assimile

Caro aluno, o uso de componentes 555 está bem estabelecido pelos projetistas. Um emprego muito interessante é o emprego do 555 como um **multivibrador astável** – no qual a cada oscilação reiniciamos o sinal de gatilho do componente (sinal Trigger). Com as ferramentas de simulação, fica fácil testar o circuito.

Figura 2.7 | Componente 555 como multivibrador astável



Fonte: adaptada de Circuit Simulator (2017).

É possível a montagem de diversos circuitos com o 555. Seja curioso e simule.

Prezado aluno, o assunto desta seção é bem interessante. Ele possibilita que você tenha mecanismos para testar os circuitos digitais e pela internet – essa é *uma tendência que não tem mais volta*. Teste seu conhecimento com essas novas ferramentas. Bons estudos.

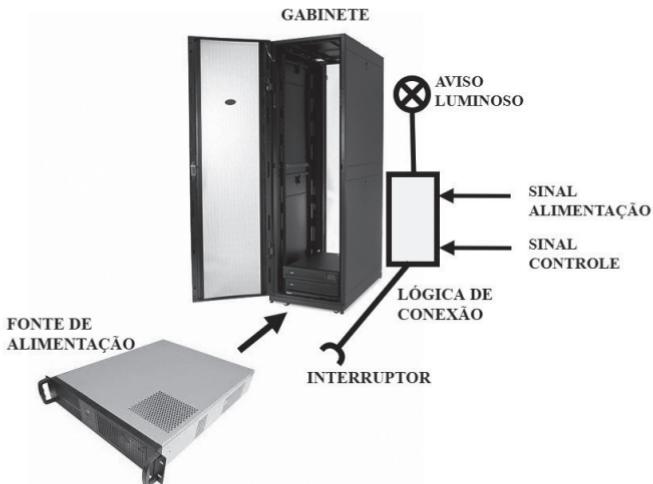
Sem medo de errar

Vamos relembrar o contexto da situação-problema que lhe foi proposto. A empresa na qual você trabalha necessita customizar uma aplicação. Tendo em vista que essa aplicação necessita de hardware específico, ou seja, não é possível encontrar solução microprogramada com um custo acessível, a equipe de engenharia decidiu utilizar um CPLD. Essa é a sua chance de mudar de setor – sair da manutenção e ir para o setor de pesquisa e desenvolvimento. Como você é uma pessoa antenada e já havia estudado lógica programável, comentou com o seu chefe que teria interesse em participar do projeto. Para testar a sua habilidade, foi-lhe dada a oportunidade de desenvolver inicialmente uma lógica de teste (*testbench*) de um circuito combinacional que

alerta a conexão ou a desconexão de uma fonte de alimentação em um dos servidores da empresa após 5 segundos. A solução é puramente combinacional, mas necessita ser testada. Como você irá programar? Será necessário o uso de ferramentas de simulação? Qual a linguagem que você irá adotar? E qual a lógica que será simulada? Existem muitas incógnitas, mas, se a sua simulação for bem pensada, irá obter todas as suas respostas.

Inicialmente, vamos entender a lógica que será testada. O sistema consiste em uma fonte de alimentação que, conectada ao gabinete do servidor, sinaliza através de um aviso luminoso que existe uma fonte conectada no servidor. A Figura 2.8 ilustra o problema a ser resolvido:

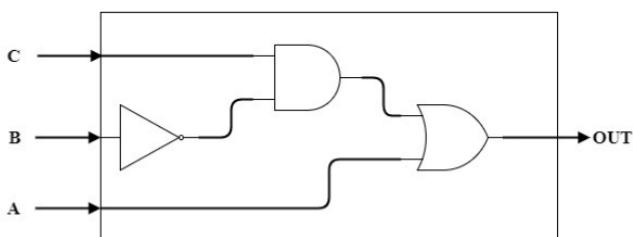
Figura 2.8 | Sistema de sinalização de gabinetes



Fonte: elaborada pelo autor.

A lógica a ser testada é ilustrada pela Figura 2.9:

Figura 2.9 | Lógica de conexão de fonte em gabinetes



Fonte: elaborada pelo autor.

Lembre-se de que (**A**) é a entrada do sinal de controle, ou seja, ficará em nível alto (em 1) sempre que houver alguma fonte secundária alimentando o gabinete. A entrada (**B**) consiste em sinal oriundo da fonte de alimentação que sinaliza que ela está funcionando corretamente. Por fim, o sinal (**C**) corresponde ao interruptor sinalizando que a fonte está inserida no gabinete corretamente.

Como você irá testar a lógica? A ideia é que você utilize uma linguagem de descrição de hardware para testar as funcionalidades desse circuito. Se ele for válido, a próxima etapa é criar um circuito com as portas lógicas. Inicialmente, você pode utilizar a lógica a seguir como referência de lógica de teste.

```

DECLARAÇÃO DAS          library IEEE; useIEEE.std_logic_1164.all;
BIBLIOTECAS E ENTIDADE      entity testbench is - VAZIO endtestbench;
INÍCIO DA ARQUITETURA      architecture tb of testbench is
DO TESTBENCH                component MINHA_LOGICA is port(INTERFACE DA MINHALÓGICA);
LÓGICA A SER                end component;
TESTADA                      signal a_in, b_in, c_in; out:std_logic;
SINAIS DO SISTEMA           begin
CONEXÃO COM O              DUT: MINHA_LOGICA port map(INTERFACE DA MINHALÓGICA);
DISPOSITIVO QUE             Process begin
SERÁ TESTADO                  a_in <= '0'; b_in <= '0'; c_in <= '0'; wait for 1 ns;
                                assert(out='0') report "Fail 0/0" severity error;
                                a_in <= '1'; b_in <= 'X'; c_in <= 'X';wait for 1 ns;
                                assert(q_out='1') report "Fail 1/X" severity error;
                                assert false report "Test done." severity note; wait;
                                end process;
                                endtb
LÓGICA DE
TESTE

```

Em posse da lógica de teste, crie agora a lógica de conexão. Não se esqueça de testar todos os possíveis casos de sinais de entrada da sua lógica. Utilize as ferramentas de verificação disponíveis na internet. Bom trabalho e boa sorte com os testes.

Avançando na prática

Simulação de ondas quadradas

Descrição da situação-problema

A sua solução para sinalizar o gabinete do servidor ficou excelente. Acreditamos que em breve você estará mudando de setor. Ocorre que a empresa possui mais uma demanda simples, mas fundamental, para sinalizar que os gabinetes do servidor estão sem a fonte de alimentação.

Consiste em um sistema que aciona um autofalante a uma frequência de 20 kHz. A lógica proposta inicialmente é utilizar um FPGA, contudo você tem certeza de que existe uma solução com um custo menor. Dessa forma, você pensa em usar seus conhecimentos de eletrônica para simular um circuito mais simples. Como é possível simular e garantir que o projeto funcione sem problemas? Quais são as frequências possíveis?

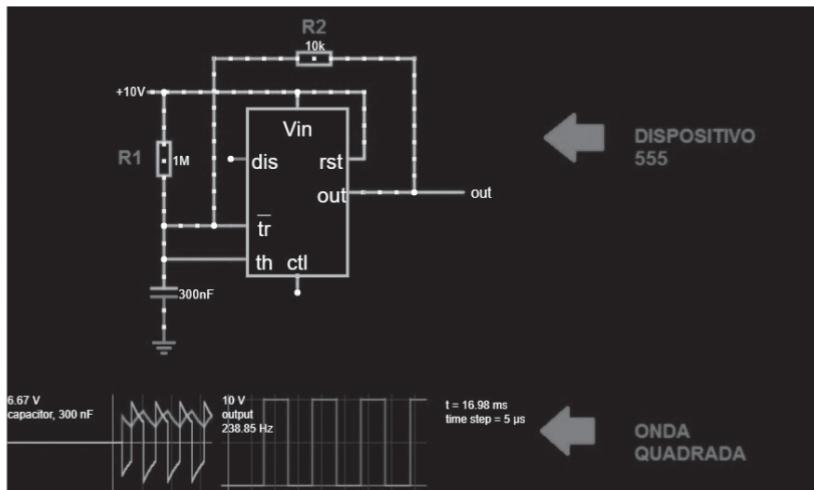
Agora, é o momento para pensar em como simular. Utilize o conhecimento adquirido na unidade. Certamente, existem soluções e simuladores on-line para resolver o problema.

Resolução da situação-problema

Um bom início é pesquisar soluções analógico-digitais na literatura. Uma solução mais simples e amplamente conhecida é o nosso dispositivo 555 funcionando como oscilador de onda quadrada na frequência que necessitamos.

Nesse sentido, utilize a ferramenta Circuit Simulator (2017) como descrito na Figura 2.10 para testar a sua solução e descobrir qual a configuração do oscilador para a frequência desejada.

Figura 2.10 | Oscilador 555 como gerador de onda quadrada



Fonte: adaptada de Circuit Simulator (2017).

Varie as resistências de R1 e R2. Analise qual a frequência resultante na saída OUT. Escreva um relatório com a sua proposta de solução e os resultados de simulação. Utilize a sua capacidade analítica e pontue os principais aspectos que você levantou. Use figuras, mas não se esqueça de explicá-las, bem como apontar as referências.

Faça valer a pena

1. Um projeto de sistema digital consiste em determinar o que chamamos de função lógica. Uma função lógica reproduz a lógica de funcionamento do sistema de acordo com estímulos que esse sistema recebe. É nesse momento que podemos criar um circuito lógico mais complexo que execute essas funções lógicas a partir de circuitos lógicos simples.

Considerando o texto-base, qual cenário melhor que descreve o emprego de uma função lógica SIMPLES?

- a) Um submarino nuclear navegando abaixo da calota polar.
- b) As sondas espaciais para Marte que analisam as composições de solo.
- c) Sistema de controle de trajetória de um carro autônomo.
- d) Um aparelho de ressonância magnética com resolução de 100[nm] (nanômetros).
- e) O controle de abertura e fechamento de um portão através de um sensor de presença e duas chaves de final de curso.

2. Lógica programável

Para entender lógica programável, temos de entender os dispositivos que estão ligados ao conceito. Os componentes que utilizam lógica programável possuem em sua estrutura interna centenas ou mesmo milhares de circuitos como portas lógicas, *flip-flops* e registradores, entre outros. Esses circuitos são interligados internamente através de estruturas bem complexas de barramentos de interconexão.

Linguagem de descrição de hardware

Inicialmente, essas linguagens foram a melhor forma encontrada para dar um significado mais preciso para cada um dos elementos de um circuito. Assim, quando as ferramentas de software fizerem a leitura da descrição, saberiam exatamente o que fazer.

Considerando o texto-base sobre lógica programável e linguagem de descrição de hardware, é correto afirmar:

a) **Lógica programável:** as interconexões entre os circuitos são fixas – característica principal de uma lógica programável.

Linguagem de descrição de hardware: VHDL e VERILOG são exemplos de HDLs.

b) **Lógica programável:** são exemplos de dispositivos com lógica programável PAL, PLA, FPGA e CPLD.

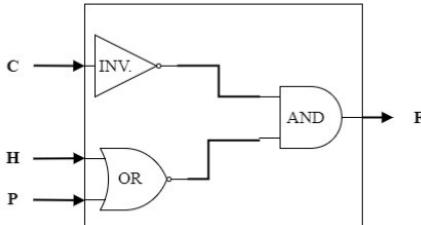
Linguagem de descrição de hardware: tanto VHDL quanto VERILOG necessitam incluir bibliotecas básicas da linguagem (Ex. IEEE).

c) **Lógica programável:** as interconexões entre os circuitos são programáveis – característica principal de uma lógica programável.

Linguagem de descrição de hardware: VHDL necessita incluir bibliotecas básicas da linguagem (Ex.: IEEE), enquanto VERILOG não é necessário incluir bibliotecas básicas.

d) **Lógica programável:** não é possível a criação de lógicas complexas, pois é uma tecnologia antiga e pouco usual.

Linguagem de descrição de hardware: a figura a seguir permite a descrição utilizando VHDL, mas não é possível utilizando VERILOG.



e) **Lógica programável:** PAL e PLA são dispositivos com blocos pré-fabricados com lógica AND e OR, ou NAND e NOR. Já FPGAs e CPLDs são dispositivos que já vêm como os blocos de hardware pré-programados

(ASIC) e, portanto, têm as suas funcionalidades já definidas no ato de fabricação e não permitem qualquer mudança na lógica.

Linguagem de descrição de hardware: a figura a seguir ilustra um circuito com três portas de entrada e uma porta de saída; três componentes básicos; e dois fios para interligação entre esses blocos.

Descreve o nome circuito
ControlePorta e define as portas de entrada e saída

Fios para interconexão entre os circuitos

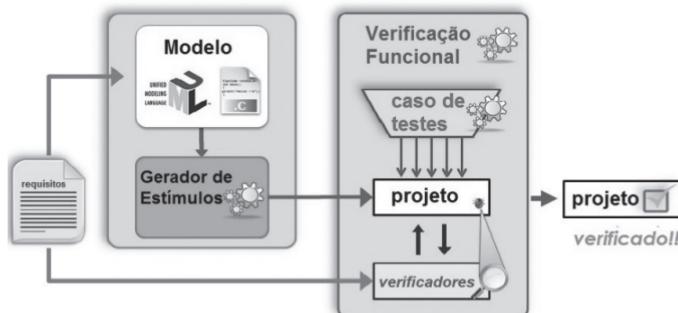
O circuito **ControlePorta** consiste internamente nos circuitos **inv_1**, **or2_1**, **and2_1**

```
module ControlePorta (C, H, P, f);
  input C, H, P;
  output f;
  wire n1, n2;
  INV inv_1 (C, n1);
  OR2 or2_1 (H, P, n2);
  AND2 and2_1 (n1, n2, f);
endmodule
```

3. Existem diversas ferramentas de descrição e simulação de HDL. O que ocorre é que muitas dessas ferramentas estão voltadas para uma tecnologia específica de FPGAs. É o caso de duas grandes empresas que possuem uma boa fatia do mercado: a Xilinx e a Altera.

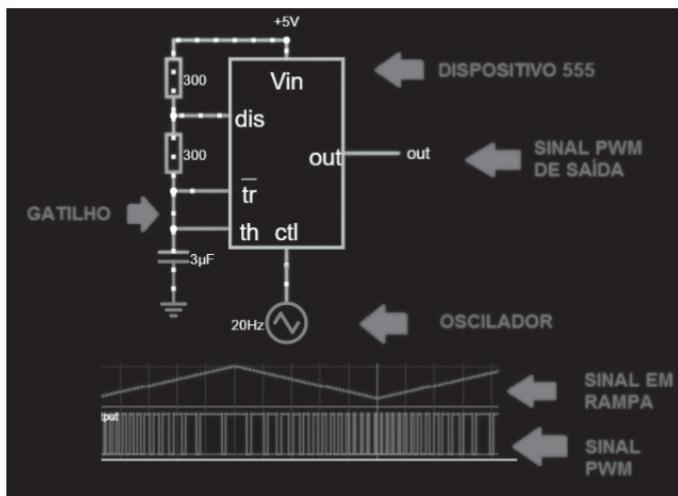
Considerando o contexto de simulação de lógica programável e circuitos digitais, é correto afirmar:

- Projetistas experientes não necessitam verificar modificações nos projetos, tendo em vista que conhecem bem a tecnologia que utilizam.
- Existem diversas ferramentas de simulação tanto para descrição de hardware quanto para circuitos digitais – todas são pagas e necessitam licença para um usuário comum.
- A figura a seguir exemplifica uma metodologia de verificação de um projeto. Nela, é possível observar que sinais de testes são injetados no circuito que será verificado. O resultado final poderá dizer se o circuito tem ou não problemas.



Fonte: elaborada pelo autor.

- d) Testar um circuito através de ambientes computacionais de simulação é algo perigoso, pois, ao testar modificações no projeto, os erros podem levar a queimar todo o circuito simulado e prejudicar o computador.
- e) A figura a seguir é um exemplo de simulação através de um ambiente computacional de simulação. Contudo, ambientes computacionais de simulação não permitem modificar facilmente os componentes do circuito, tendo em vista que cada componente possui um custo e prazo entrega para o cliente.



Fonte: adaptada de Circuit Simulator (2017).

Seção 2.2

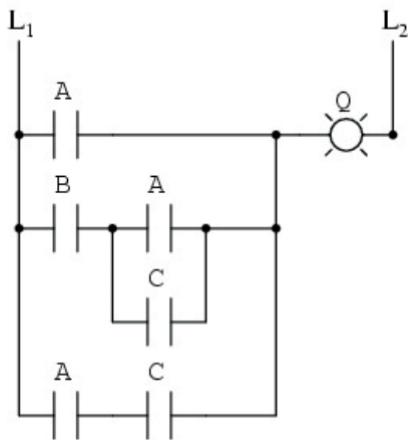
Álgebra booleana e simplificação lógica

Diálogo aberto

Caro aluno, esta seção tem o objetivo de direcionar o seu conhecimento sobre álgebra booleana, teoremas de De Morgan e simplificação de expressões booleanas. Toda essa informação é para conduzir você o mais próximo possível dos conceitos de circuitos digitais e da realidade do meio profissional. Os conceitos que serão expostos nesta seção são a base para você compreender lógica digital.

Nesse contexto, a melhor forma para entender essa proposta de aprendizado é imaginar o seguinte cenário: você desenvolveu um excelente trabalho na sua empresa. Irá sair da área de manutenção para o time de pesquisa e desenvolvimento – algo fantástico. Como a empresa tem intensão em diversificar o mercado, ela vem analisando a possibilidade de inserir produtos na indústria metalomecânica. Para isso, começou a projetar controladores lógicos programáveis (CPL, ou do inglês *Programmable Logic Controller*). Um CLP é um sistema de controle amplamente utilizado na indústria para comandar e monitorar máquinas ou processos industriais. Ocorre que o projetista chefe que analisa as aplicações desenvolvidas – o software – acredita que a lógica de um dos CLPs pode ser simplificada. Ele quer testar o seu conhecimento. Para isso, passou a seguinte lógica para você simplificar, ilustrada na Figura 2.11 (considere: na mesma linha, variáveis em série correspondem à lógica AND. Linhas em paralelo correspondem à lógica OR).

Figura 2.11 | CLP lógica



Fonte: elaborada pelo autor.

Essa lógica permite que o CLP funcione de acordo com a necessidade do projeto. Como você é uma pessoa antenada e já havia estudado circuitos digitais, aceitou prontamente o desafio. Como você irá começar? Será necessário o uso de ferramentas de simulação? Como será a simplificação? E qual a lógica final? Existem muitas incógnitas, mas uma simplificação bem pensada irá conter todas as suas respostas. Lembre-se de que esse cenário é uma proposta de aprofundamento ao seu conhecimento sobre circuitos digitais. E que todo o seu conhecimento será feito com base em bastante leitura. Dessa forma, devemos documentar todo esse aprendizado. Produza, então, um relatório com a sua proposta de solução. Utilize a sua capacidade analítica e pontue os principais aspectos que você levantou. Use figuras, mas não se esqueça de explicá-las, bem como apontar as referências (devemos dar o mérito para quem fez, é o mais justo). Sempre é bom lembrar: use uma linguagem científica e evite ao máximo o gerúndio.

Nas seções a seguir, será apresentada a base para esse conhecimento, mas não fique limitado somente a essa fonte de conhecimento. Pesquise, converse e troque informações com outras pessoas. O mercado está favorável para funcionários proativos e esforçados. Boa sorte e boa simulação.

Não pode faltar

Nesta seção, iremos avançar nossos conhecimentos sobre sistemas digitais. Serão abordados tópicos referentes aos postulados e propriedades e formas canônicas de expressões booleanas.



Vocabulário

Caro aluno, de maneira geral, no campo das exatas, uma forma canônica refere-se à forma normal e clássica de representar alguma relação. Como exemplo, temos a forma canônica das equações do 2º grau como:

$$ax^2 + bx + c = 0$$

Sendo a, b, c números reais e a não nulo.

Como visto nas unidades anteriores, os circuitos lógicos executam, ou mesmo, são equivalentes a, expressões booleanas. Essas expressões em geral representam situações ou problemas do nosso dia a dia (problemas reais). Contudo, os circuitos gerados por essas tabelas-verdade admitem, muitas vezes, simplificações – ou seja, a partir de alguns procedimentos conseguimos reduzir o número de portas lógicas do sistema. A vantagem dessa redução está na diminuição do grau de dificuldade na montagem e no custo final do circuito digital desenvolvido. Nesse sentido, o estudo da simplificação de circuitos lógicos necessita do conhecimento da teoria da álgebra de Boole. Isso é possível entendendo seus postulados, as propriedades, as equivalências, as constantes etc. O mesmo se aplica para o teorema de De Morgan. O que de fato ocorre é que, na álgebra de Boole, iremos encontrar os fundamentos dos circuitos de eletrônica digital.



Assimile

George Boole nasceu em Lincoln, em 2 de novembro de 1815, e faleceu em Ballintemple, em 8 de dezembro de 1864, na Inglaterra. Boole foi matemático e filósofo, considerado o criador da álgebra booleana – teoria fundamental para o desenvolvimento da computação moderna. Na sequência, temos Augustus De Morgan (não confunda com o famoso pirata espanhol). Nasceu em Madura, Índia, em 27 de junho de 1806, e morreu em Londres, em 18 de março de 1871. De Morgan foi matemático e lógico. Cabe a ele o mérito de ter formulado as leis de De Morgan e

foi o primeiro a tornar rigorosa a ideia da indução matemática. Já em 1938, coube a Claude Shannon aplicar todo esse conhecimento de álgebra para mostrar que as propriedades de circuitos elétricos poderiam ser representadas por uma álgebra booleana com dois valores distintos (TOCCI; WIDMER; MOSS, 2007, p. 50).

O conhecimento a seguir é muito importante. Como mencionado anteriormente, ele é a base da computação moderna. Fique atento e lembre-se dos conceitos aqui elencados.

Lógica booleana – constantes, variáveis e expressão

Precisamos definir o que é uma lógica booleana, ou também chamada de álgebra booleana. Devemos ter em mente que a álgebra booleana usa símbolos algébricos como "X", "Y", "Z", "p", "q", "r" para identificar palavras, frases ou as suposições. Ela, portanto, usa símbolos para criar um sistema algébrico com operações de soma e multiplicação que resultam em uma conclusão lógica. Não é difícil imaginar as seguintes premissas:

$E = \text{Estudantes}$

(Premissas 2.1)

$K = \text{Ensino Superior}$

Se analisarmos um conjunto de pessoas através das representações a seguir, podemos concluir:

$(1 - E)$ – representa todas as pessoas que não são estudantes (E).

$(E \times k)$ – representa todas as pessoas que são estudantes (E) e são do ensino superior (K).

$1 (\text{um})$ – representa pessoas que são estudantes (E) ou são do ensino superior (K).

A álgebra de Boole é fantástica ao definir essas afirmações em equações. Agora, vamos mais a fundo para entendermos como podemos definir as variáveis de Boole. Existem apenas duas constantes booleanas fundamentais:

0 (zero)

(Premissas 2.2)

1 (um)

Assim, uma variável booleana é representada por aqueles símbolos algébricos que comentamos no início (as letras "X", "Y", "Z", "p", "q", "r")

e só podem assumir dois valores, ou 0 (zero) ou 1 (um). Por exemplos: ou X é igual a 1 (um) ou X é igual a 0 (zero). Ou Z é igual a 1 (um) ou Z é igual a 0 (zero).



Reflita

Você consegue perceber que constantes podem ser associadas a estados que você já conhece bem – ligado (um) e desligado (zero); a V_{cc} (um) e a GND (zero); presença de algo (um) ou ausência de algo (zero)? Caro aluno, reflita sobre quais são outros estados possíveis que podem ser representados por constantes.

Esse momento é fundamental, pois é a partir dele que conseguimos definir o que é uma **expressão booleana**. Uma expressão booleana é, portanto, uma expressão algébrica (também chamada de expressão matemática) que envolve constantes e/ou variáveis booleanas, cujos resultados podem assumir apenas dois valores, ou 0 (zero) ou 1 (um). Alguns exemplos:

$$S = A \times B$$

$$S = A + B \times C$$

(Expressões booleanas 2.1)

$$S = (A \times C) + (B \times C)$$

Aqui, é possível entender que o resultado dos valores que A, B, C e D definem o valor final de S.



Exemplificando

Um exemplo de expressão booleana é a expressão $S = A \times B$.

A e B podem assumir 0 ou 1. Observe a seguir os valores possíveis de S:

$$S = A \times B = 1 \times 1 \Rightarrow S = 1$$

$$S = A \times B = 1 \times 0 \Rightarrow S = 0$$

(Expressões booleanas 2.2)

$$S = A \times B = 0 \times 1 \Rightarrow S = 0$$

$$S = A \times B = 0 \times 0 \Rightarrow S = 0$$

Assim, conseguimos perceber o quanto os valores de A e B influenciam na saída S.

Postulados e propriedades

Na álgebra booleana existe o que chamamos de postulados ou também conhecidos como axiomas. É a partir de axiomas que são estabelecidas várias propriedades da álgebra de Boole. Diversas propriedades como a negação (complemento ou lógica inversa), a

adição (porta lógica E) e soma (porta lógica OU). Essas propriedades podem ser verificadas como equivalências lógicas. Para demonstrar cada uma dessas propriedades, basta utilizar tabelas-verdades e constatar a sua equivalência. Fique atento a seguir – listaremos os principais postulados relacionados à álgebra de Boole.

Tabela 2.2 | Postulados da álgebra de Boole

Complemento de uma variável	Se $A = 0$ então $\bar{A} = 1$ Se $A = 1$ então $\bar{A} = 0$.
Adição com duas variáveis	$0 + 0 = 0$ $1 + 0 = 1$ $0 + 1 = 1$ $1 + 1 = 1$
Multiplicação com duas variáveis	$0 \times 0 = 0$ $1 \times 0 = 0$ $0 \times 1 = 0$ $1 \times 1 = 1$

Fonte: elaborada pelo autor.

Iremos tratar agora das propriedades da álgebra de Boole. Analise com muita atenção a Tabela 2.3. Ela lista as respectivas propriedades da álgebra de Boole e será fundamental a partir de agora.

Tabela 2.3 | Propriedades da álgebra de Boole

Propriedade	Complemento	Adição	Multiplicação
Identidade	$\bar{\bar{A}} = A$	$A + 0 = A$ $A + 1 = 1$ $A + A = A$ $A + \bar{A} = 1$	
Comutativa	-	$A + B = B + A$	$A \times B = B \times A$
Associativa	-	$A + (B + C) \Rightarrow$ $(A + B) + C \Rightarrow$ $A + B + C$	$A \times (B \times C) \Rightarrow$ $(A \times B) \times C \Rightarrow$ $A \times B \times C$

Distributiva	-	$A + (B \times C) \Rightarrow (A + B) \times (A + C)$	$A \times (B + C) \Rightarrow (A \times B) + (A \times C)$
--------------	---	-------------------------------------------------------	------------------------------------------------------------

Fonte: elaborada pelo autor.

Complementando, as leis da álgebra de Boole dizem respeito ao espaço booleano – ou seja, aos valores que uma variável pode assumir – e operações elementares desse espaço. Já as propriedades são deduzidas a partir das definições das operações elementares do espaço booleano. Caro aluno, as regras apresentadas devem ser bem estudadas e entendidas.

Teorema de De Morgan

Os teoremas propostos pelo matemático Augustus De Morgan ou somente De Morgan são de grande contribuição para a simplificação de expressões em álgebra booleana. Esses teoremas definem regras que são usadas para converter as operações lógicas E em OU e vice-versa.

- O complemento do produto é igual à soma dos complementos:

$$\overline{A \times B} = \bar{A} + \bar{B} \quad (\text{Teorema 1})$$

$$\overline{A + B} = \bar{A} \times \bar{B} \quad (\text{Teorema 2})$$

Todos os teoremas aqui apresentados estendem-se para **N** variáveis, ou seja:

$$\overline{(A \times B \times C \dots \times N)} = \bar{A} + \bar{B} + \bar{C} \dots + \bar{N}, \text{ para o Teorema 1}$$

$$\overline{(A + B + C \dots + N)} = \bar{A} \times \bar{B} \times \bar{C} \dots \times \bar{N}, \text{ para o Teorema 2}$$

Os teoremas de Augustus De Morgan são interessantes de serem comprovados. Analise a Tabela 2.4, ela ilustra a igualdade entre os dois termos para duas variáveis e validade do teorema.

Tabela 2.4 | Igualdade entre dois termos do teorema 2 de De Morgan

A	B	$\overline{A+B}$	$\bar{A} \times \bar{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Fonte: elaborada pelo autor.

Observe a coluna $\overline{A+B}$; o resultado é igual ao da coluna $\bar{A} \times \bar{B}$, portanto isso prova o teorema.



Pesquise mais

Caro aluno, observe que o número de combinações que as variáveis de entrada podem assumir pode ser calculado por 2^n , onde n é o número de variáveis de entrada. Nesse sentido, pesquise como criar uma tabela-verdade de uma equação booleana e como você deve avaliar os termos conforme a precedência. Pesquise em referências como Tocci, Widmer e Moss (2007), capítulo 2, página 50 em diante. Existem diversas tabelas-verdade que servem de referência para complementar o seu estudo.

Precisamos agora compreender mais três teoremas: o teorema da absorção e mais dois teoremas da identidade. Observe a seguir:

$$A + (A \times B) = A$$

(Teorema 3.1 – Absorção Soma)

$$A \times (A + B) = A$$

(Teorema 3.2 – Absorção Produto)

Por fim, a Tabela 2.5 lista os teoremas da identidade tanto para soma quanto para o produto.

Tabela 2.5 | Teoremas da identidade

Soma	Produto
$A + (\bar{A} \times B) = A + B$	$A \times (\bar{A} + B) = A \times B$
$\bar{A} + (A \times B) = \bar{A} + B$	$\bar{A} \times (A + B) = \bar{A} \times B$

Fonte: elaborada pelo autor.

Todos os conhecimentos sobre os teoremas são fundamentais para a próxima seção. Eles são a base para a simplificação de expressões booleanas. Prezado aluno, caso você tenha alguma curiosidade, pesquise outras fontes ou converse com outros especialistas. Não deixe de construir uma tabela-verdade para confirmar a sua solução.

Simplificação de expressões booleanas por fatoração

Agora, iremos empregar todo o conhecimento da álgebra booleana na simplificação de expressões. Como cada circuito digital que construímos corresponde a uma expressão, se você simplificar as expressões significa que você irá diminuir o tamanho do circuito. Isso é fantástico, pois essa simplificação do circuito gera economias tanto na quantidade de circuitos quanto na energia consumida por esses circuitos. Nesse sentido, existem duas formas para simplificar expressões booleanas:

- Fatoração.
- Mapas de Veitch-Karnaugh.

Iremos, nesta unidade, aprender a simplificar através da fatoração. A fatoração consiste, então, na aplicação dos postulados e propriedades da álgebra booleana em uma expressão, com o objetivo de deixar essa expressão menor que a anterior.



Assimile

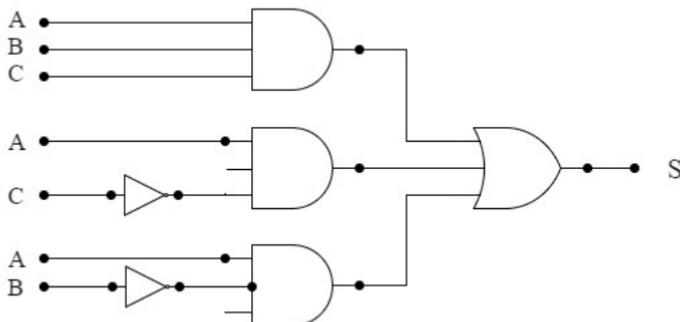
Existem várias notações (símbolos) para descrever uma variável de uma lógica booleana. Por exemplo, a lógica INVERSORA pode ser descrita como: $\sim A$, $\neg A$ e A' (lê-se A negado). A lógica OR pode ser descrita pelo símbolo \vee ; já a lógica E pode ser descrita pelo símbolo \wedge .

Analise a expressão a seguir:

$$S = A \cdot B \cdot C + A \cdot \bar{C} + A \cdot \bar{B} \quad (\text{Expressão 2.2})$$

Observe o circuito lógico final gerado pela expressão apresentada (Figura 2.12):

Figura 2.12 | Circuito lógico da Expressão 2.2



Fonte: elaborada pelo autor.

Caso você aplique os postulados através de uma fatoração, a expressão $A \cdot B \cdot C + A \cdot \bar{C} + A \cdot \bar{B}$ será reduzida somente a $S = A$. Vamos comprovar? Se $A \cdot B \cdot C + A \cdot \bar{C} + A \cdot \bar{B}$ pode ser reescrito como $A \cdot (B \cdot C + \bar{B} \cdot \bar{C})$ e como $(B \cdot C + \bar{B} \cdot \bar{C})$ é igual a 1, logo a saída S é igual a A . Você percebe como é importante a fatoração – diminuímos todo um circuito gigante para somente uma saída. Caro aluno, o exemplo apresentado é um caso ideal.

Vamos analisar mais dois exemplos. Lembrando que você deve ter em mente todos os conceitos de álgebra de Boole para resolvê-los.

Analise a expressão a seguir: $S = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}C$.

A melhor forma de iniciarmos é reorganizando algumas variáveis para visualizarmos melhor a expressão, no caso a variável \bar{C} da expressão do meio: $S = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C}B + A\bar{B}C$. Agora, podemos começar a simplificação isolando \bar{B} e B : $S = \bar{A}\bar{C}(\bar{B} + B) + A\bar{B}C$ como $(\bar{B} + B)$ é igual a 1; temos que a saída S como resultado final: $S = \bar{A}\bar{C} + A\bar{B}C$.

Agora, vamos analisar o segundo caso: $S = (A + B + C) \times (\bar{A} + \bar{B} + C)$.

Aqui, é necessário bastante atenção. Para simplificar essa equação, a melhor forma é transformarmos a multiplicação em soma. Por isso, multiplique os termos de forma distributiva.

Ficará assim:

$$S = A\bar{A} + A\bar{B} + AC + B\bar{A} + B\bar{B} + BC + C\bar{A} + C\bar{B} + CC$$

Agora, é só aplicar o restante dos postulados:

$$S = 0 + A\bar{B} + A.C + B. + 0 + B.C + C. + C.\bar{A} + C \Rightarrow$$

$$S = A\bar{B} + B. + C.(A + B + + + 1) \Rightarrow$$

$$S = A\bar{B} + B. + C.(1) \Rightarrow$$

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

Não existe melhor forma de aprender a simplificar expressões que não seja praticando. É necessário muito trabalho e estudo.



Pesquise mais

Existem duas formas de escrever uma expressão booleana: a forma normal e a forma canônica. Na forma normal, temos:

- Forma normal conjuntiva (FNC), produto de somas ou produto de maxtermos.
- Forma normal disjuntiva (FND), soma de produtos ou soma de mintermos.

Pesquise a diferença entre essas duas formas. Elas são fundamentais para entendermos como devemos escrever uma expressão booleana. Utilize como referência Tocci, Widmer e Moss (2007), capítulo 3, da página 58 até a 70. Nessas páginas, é detalhado de forma bem didática como uma expressão booleana deve ser descrita.

Uma vez obtida a forma normal da função booleana que você necessita, para simplificá-la você deve utilizar a manipulação algébrica. Você deve respeitar os postulados e as propriedades da álgebra booleana e praticar muito. Foque, portanto, na boa compreensão dos postulados desta seção. Os postulados são a base dos circuitos digitais contemporâneos. Converse com especialistas e, sobretudo, leia muito. Os livros que estão listados nas referências são leituras que certamente complementam o estudo. Leia com muita atenção o livro do Vahid (2008), pois ele aborda o conteúdo aqui apresentado de forma bem dinâmica e com exemplos ótimos. Bons estudos.

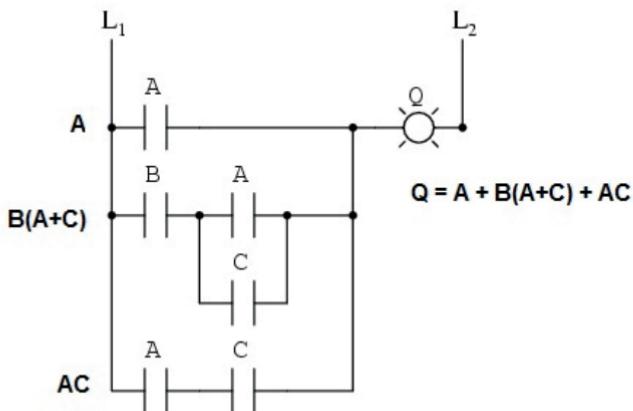
Sem medo de errar

Vamos relembrar o contexto proposto. A empresa em que você trabalha está diversificando o mercado, entrando com produtos para o setor industrial. Para isso, começou a projetar CLPs. Um controlador

lógico programável (CLP), ou do inglês PLC (*Programmable Logic Controller*), é um sistema de controle amplamente utilizado na indústria para comandar e monitorar máquinas ou processos industriais.

O projetista chefe que analisa as aplicações desenvolvidas acredita que a lógica de um dos CLPs pode ser simplificada. Ele quer testar o seu conhecimento. Para isso, ele pediu que você simplificasse a lógica da Figura 2.11. A primeira ação que você tem de tomar é montar a expressão algébrica da aplicação. A Figura 2.13 ilustra a expressão.

Figura 2.13 | Lógica CLP



Fonte: elaborada pelo autor.

Agora, basta simplificar a equação.

$$Q = A + B(A+C) + AC \Rightarrow$$

$$Q = A + BC + AC$$

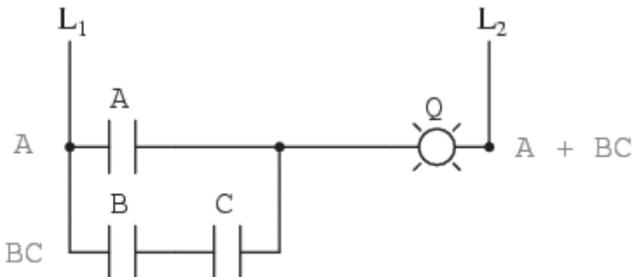
Como $A+AB$ é igual a A , temos

$$Q = A + B(A+C) + AC \Rightarrow$$

$$Q = A + BC + AC$$

Por fim, se $A+AC$ é igual a A , o resultado final é $Q=A+BC$. Portanto, o seu chefe tinha razão, é possível reduzir a expressão. Isso é importante porque reduz o tamanho da lógica dentro do CPLD, e, com isso, ele consome menos energia, além de sobrar mais espaço para projetos maiores. A figura a seguir ilustra o resultado final:

Figura 2.14 | Lógica CLP simplificada



Fonte: elaborada pelo autor.

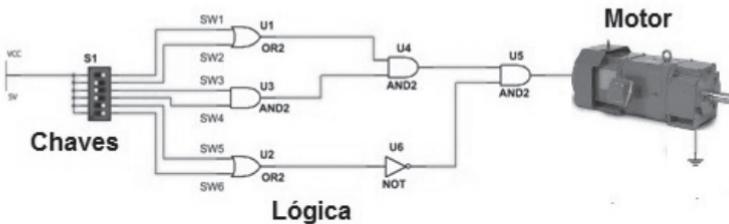
Avançando na prática

Mudanças para CLPs

Descrição da situação-problema

Existe uma demanda muito interessante em uma planta de uma fábrica. Algumas máquinas possuem o acionamento totalmente manual. Sempre que uma sequência de acionamento necessita ser acionada, uma pessoa altamente treinada precisa inserir uma sequência em grupo de interruptores, para depois acionar a máquina. Há um custo muito alto quando isso ocorre – uma pessoa deve ser treinada e alocada só para essa tarefa, e a máquina tem de ser desligada e acionada no momento certo. Nesse sentido, surgiu a oportunidade de modernização de um sistema. Esse sistema consiste em uma sequência de chaves que controlam o acionamento do motor. A ideia é substituir essas chaves por um CLP.

Figura 2.15 | Acionamento manual de motor



Fonte: elaborada pelo autor.

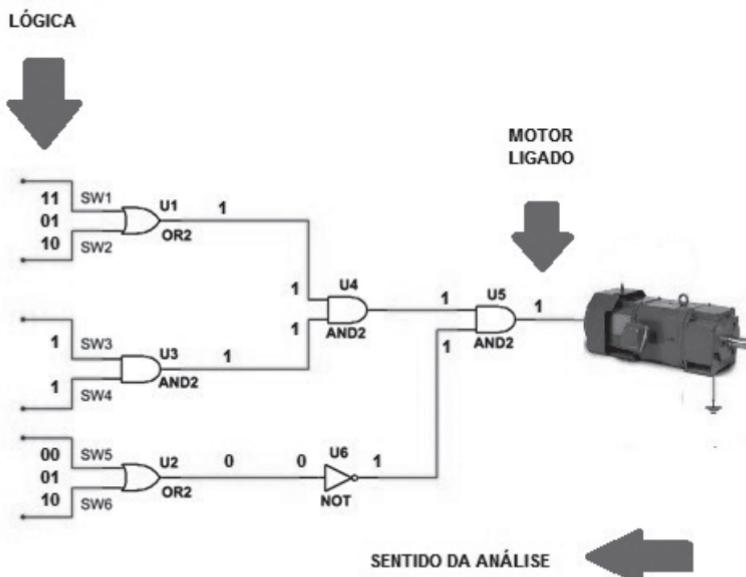
Para isso, você teria de transformar a lógica acima em uma expressão booleana, analisar quais os sinais que acionam o motor e, por fim, simplificar essa lógica.

Como é possível saber quais lógicas que acionam o motor? Será que uma tabela com os possíveis sinais resolveria isso? A solução está em analisar os sinais no sentido do motor para as chaves. Fique atento aos sinais. Bom trabalho!

Resolução da situação-problema

A primeira etapa para resolver esse problema é analisar com calma o circuito anterior. Analisando o circuito no sentido MOTOR para as CHAVES, é possível entender quais são os sinais que irão acionar o motor.

Figura 2.16 | Acionamento digital de motor



Fonte: elaborada pelo autor.

Toda a lógica é melhor visualizada quando colocada em uma tabela (Tabela 2.6).

Tabela 2.6 | Tabela com acionamentos de entrada e saída

Entradas			Saída
SW1 e SW2	SW3 e SW4	SW5 e SW6	
11	11	00	1
11	11	01	1
11	11	10	1
01	11	00	1
01	11	01	1
01	11	10	1
10	11	00	1
10	11	01	1
10	11	10	1

Fonte: elaborada pelo autor.

Agora, fica fácil montar a expressão e validar com uma ferramenta de simulação. Boa sorte e bons estudos.

Faça valer a pena

1. A lógica boolena ou também chamada de álgebra booleana utiliza símbolos algébricos como "X", "Y", "Z", "p", "q", "r" para identificar palavras, frases ou as suposições. Ela, portanto, usa símbolos para criar um sistema algébrico com operações de soma e multiplicação que resultam em uma conclusão lógica. Considere as seguintes premissas:

$$E = \text{Estudantes}$$

$$K = \text{Graduação}$$

(premissas)

$$B = \text{Bons}$$

Analizando o enunciado, a melhor sentença que valida as premissas é:

- a) $1 - (E \times K)$ significa que estamos falando que esse conjunto é de pessoas que são estudantes somente de graduação.
- b) $(E \times K)$ significa que estamos falando que esse conjunto é de estudantes e eles não são de graduação.
- c) $(E \times K)$ significa que estamos falando que todo esse conjunto é de estudantes de graduação.
- d) $(E \times K \times B) \cup (E \times K \times B)$ significa que estamos falando esse conjunto todo é de estudantes de graduação e são bons.
- e) $1 - (\bar{E} \times K)$ significa que estamos falando que esse conjunto é de pessoas que não são estudantes de graduação.

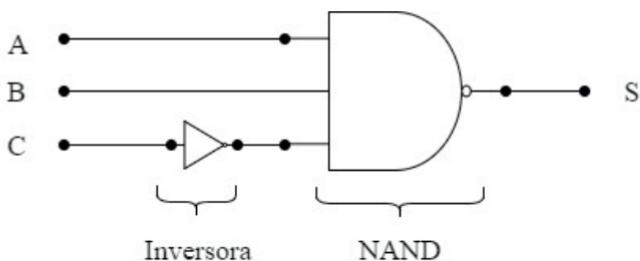
2. Os teoremas de Augustus De Morgan são utilizados para realizar a simplificação de expressões booleanas e também para desenvolver circuitos digitais. Eles foram propostos pelo matemático britânico Augustus De Morgan, no século XIX.

Simplifique a expressão $X = \overline{A + \bar{B}C}$ utilizando os dois teoremas de De Morgan. A melhor alternativa é:

- a) $X = \overline{A + B + C}$
- b) $X = A(B + C)$
- c) $X = A(B + C)$
- d) $X = \bar{A} + (B\bar{C})$
- e) $X = \bar{A}(B + \bar{C})$

3. Como cada circuito digital que construímos corresponde a uma expressão, se você simplificar as expressões significa que você irá diminuir o tamanho do circuito. Isso gera economias tanto na quantidade de circuitos quanto na energia consumida por eles.

Considere a lógica a seguir:



Encontre a expressão lógica para a saída do circuito acima e **simplifique-a** utilizando os teoremas de De Morgan.

- a) $S = A + B + C$
- b) $S = A\bar{B}C$
- c) $S = \overline{ABC}$
- d) $S = \overline{ABC}$
- e) $S = \bar{A} + \bar{B} + C$

Seção 2.3

Diagrama de Veitch-Karnaugh

Diálogo aberto

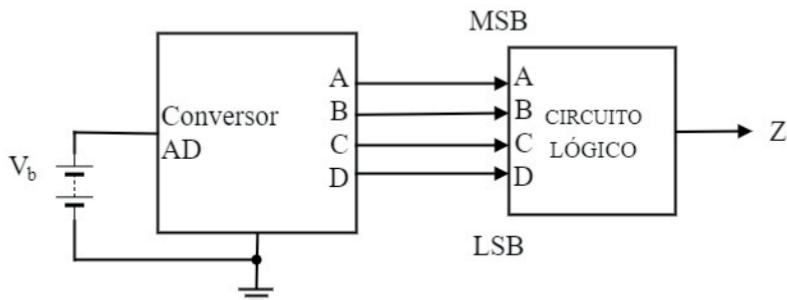
Prezado aluno, é importante para os profissionais que trabalham com tecnologia terem em vista como o mercado está se adaptando a todas as grandes mudanças que estão ocorrendo no mundo. Essa visão de como o mercado está sendo conduzido permite que cada profissional analise suas habilidades, vivências e experiências. Essa autorreflexão possibilita que o profissional se destaque cada vez mais no meio em que vive, como a empresa, universidade ou mesmo na sua comunidade.

Uma das formas para adaptar-se ao mercado é buscar conhecimento em uma área específica. Assim, nesta seção, iremos trabalhar um conteúdo muito interessante. Ele trata sobre circuitos lógicos digitais e os seus modos de simplificação e de expressão. Isso vai permitir que você construa sistemas através de lógica combinacional (portas lógicas) e que esses circuitos sejam o mais otimizado possível. Vamos, então, analisar o seguinte cenário, relembrando: você trabalha em uma empresa da área de telecomunicações, atuando na equipe de pesquisa e desenvolvimento. Você está sendo encarregado de analisar se o mercado de carros elétricos poderá ser um novo segmento de atuação da sua empresa. Dessa forma, você se propôs a desenvolver um projeto que irá monitorar a tensão CC (V_B) de um carro elétrico.

Ocorre que a saída do conversor é um número binário de quatro bits – **ABCD**. Esse valor corresponde à tensão da bateria em degraus de 1 V (um Volt), sendo a variável **A** o valor menos significativo (MSB). As saídas binárias do conversor são as entradas de um circuito lógico que gera uma saída em nível ALTO, sempre que o valor for maior que seis volts (em binário 0110).

Observe a Figura 2.17.

Figura 2.17 | Conversor AD e circuito lógico de carga de bateria



Fonte: elaborada pelo autor.

Em posse dessas informações, você inicia o seu trabalho. Quais são as características DC importantes do sistema? Como será a tabela-verdade? É possível simplificar a expressão resultante? Dessa forma, elabore um relatório explicando os limites de operação, como será a lógica implementada e qual a frequência de operação do sistema. Caro aluno, convidamos você a se preparar para o mercado compreendendo o conteúdo desta unidade. Na seção "Não pode faltar", iremos trabalhar o uso de maxtermos e mintermos, como desenvolver as tabelas-verdade e o diagrama de Karnaugh, além de como fazer minimização de soma dos produtos e o produto das somas. Por fim, iremos analisar detalhes sobre o diagrama de Karnaugh de 2, 3, 4 e 5 variáveis. Utilize o texto base como referência e as folhas de dados dos componentes. Bons estudos!

Não pode faltar

Prezado aluno, nesta unidade, iremos avançar nossos conhecimentos sobre sistemas digitais. Entenderemos como é possível representar uma expressão lógica e depois como simplificar da melhor forma possível essas expressões. Como visto anteriormente, as expressões booleanas representam uma lógica cujas variações nas entradas determinam um sinal de saída. O que ocorre é que muitas dessas expressões permitem a simplificação para lógicas menores. Essa redução implica em diversas vantagens, como a simplificação do circuito final, redução do grau de dificuldade do sistema e, sobretudo, redução de custo. Nesse sentido, a simplificação desses circuitos lógicos necessita de um bom conhecimento em maxtermos e mintermos, o uso de tabelas-verdade e do mapa de Veitch-Karnaugh. O conhecimento a seguir é muito

importante, pois ele serve como fundamento para a computação moderna. Mantenha o foco nesta unidade.

Maxtermos e mintermos

Mintermos e maxtermos correspondem a uma forma utilizada para reescrever uma expressão lógica. Essa forma é o início para padronizar a expressão a fim de se obter a expressão de forma mais simplificada. Nesse sentido, uma expressão escrita na forma mintermos é escrita na forma da soma de produtos – independentemente se as variáveis são complementadas (negadas). Reforçando: mintermos possuem somente termos com a lógica AND (produto).



Assimile

Fique atento. Na expressão, se a variável for zero (0), ela deve aparecer complementada (negada), e, se variável for um (1), deve aparecer não complementada (negada).

A Tabela 2.7 exemplifica o uso de três variáveis (X_1, X_2, X_3) escritas como **mintermos** (como produtos). Observe atentamente a coluna mintermos; nela, todas as variáveis X_1, X_2, X_3 estão escritas como produto entre si.

Tabela 2.7 | Tabela com os valores de mintermos

Linha	X_1	X_2	X_3	Mintermos
1	0	0	0	$\bar{X}_1 \bar{X}_2 \bar{X}_3$
2	0	0	1	$\bar{X}_1 \bar{X}_2 X_3$
3	0	1	0	$\bar{X}_1 X_2 \bar{X}_3$
4	0	1	1	$\bar{X}_1 X_2 X_3$
5	1	0	0	$X_1 \bar{X}_2 \bar{X}_3$
6	1	0	1	$X_1 X_3 X_3$
7	1	1	0	$\bar{X}_1 \bar{X}_2 X_3$
8	1	1	1	$X_1 X_2 X_3$

Fonte: elaborada pelo autor.

Agora, iremos entender como criar uma expressão da forma mintermos. Por exemplo, imagine que a lógica escolhida utilize as linhas 1, 3, 5 e 7 da Tabela 2.7. A expressão na sua forma canônica pode ser escrita como: $F(X_1X_2X_3) = \sum m(1,3,5,7)$, ou seja – a soma dos produtos das linhas 1, 3, 5 e 7 e, substituindo as linhas pelos respectivos valores da coluna **mintermos**, assim, chegaremos à seguinte expressão: $F(X_1X_2X_3) = \bar{X}_1\bar{X}_2\bar{X}_3 + \bar{X}_1X_2\bar{X}_3 + X_1\bar{X}_2\bar{X}_3 + \bar{X}_1\bar{X}_2X_3$.

Agora, vamos descobrir como é a forma maxtermos. A forma maxtermo é o oposto da mintermo, portanto maxtermo é escrito na forma do produto das somas. Maxtermos possuem somente termos com a lógica OR (soma). Mas fique atento! Na expressão $F(X_1X_2X_3)$, se alguma das variáveis, X_1 , X_2 , ou X_3 , for um (1), ela deve aparecer complementada, se alguma variável for zero (0) ela não deve ser complementada. A Tabela 2.8 exemplifica o uso de três variáveis (X_1 , X_2 , X_3) escritas como maxtermos (como soma). Observe na Tabela 2.7 nas colunas X_1 , X_2 , X_3 na primeira linha – temos todos os valores iguais a zero. Portanto, na coluna **maxtermos**, os valores não serão negados (diferente do que ocorre com mintermos – fique atento).

Tabela 2.8 | Tabela com os valores de maxtermos

Linha	X_1	X_2	X_3	Mintermos
1	0	0	0	$X_1 + X_2 + X_3$
2	0	0	1	$X_1 + X_2 + \bar{X}_3$
3	0	1	0	$X_1 + \bar{X}_2 + X_3$
4	0	1	1	$X_1 + \bar{X}_2 + \bar{X}_3$
5	1	0	0	$\bar{X}_1 + X_2 + X_3$
6	1	0	1	$\bar{X}_1 + X_2 + \bar{X}_3$
7	1	1	0	$\bar{X}_1 + \bar{X}_2 + X_3$
8	1	1	1	$\bar{X}_1 + \bar{X}_2 + \bar{X}_3$

Fonte: elaborada pelo autor.

Da mesma forma que o mintermo, agora iremos entender como criar uma expressão da forma maxtermos. Por exemplo, vamos supor que a lógica escolhida necessite das linhas 2, 4,

6 e 8. A expressão na sua forma canônica pode ser descrita como: $F(X_1X_2X_3) = \prod m(2,4,6,8)$; logo, substituindo as linhas pelos respectivos maxtermos, obteremos a seguinte expressão: $F(X_1X_2X_3) = (X_1 + X_2 + \bar{X}_3)(X_1 + \bar{X}_2 + \bar{X}_3)(\bar{X}_1 + X_2 + \bar{X}_3)(\bar{X}_1 + \bar{X}_2 + \bar{X}_3)$.

Nas próximas seções, aprenderemos como aplicar todo o seu conhecimento sobre álgebra de Boole e para simplificar essas expressões.



Assimile

Cada maxtermo é o complemento do seu correspondente minitermo e vice-versa. É possível provar essa afirmação. Por exemplo, se você escolher a quarta linha da tabela mintermos, para você transformar o termo em maxtermos basta fazer o complemento da linha. Observe:

$$\begin{aligned}\bar{X}_1X_2X_3 &= \overline{\bar{X}_1X_2X_3} = \bar{\bar{X}}_1 + \bar{X}_2 + \bar{X}_3 = \\ \therefore \bar{X}_1X_2X_3 &\Leftrightarrow X_1 + \bar{X}_2 + \bar{X}_3\end{aligned}$$

Utilize essa tática sempre que precisar.

Por fim, maxtermos e mintermos são padronizações que servem como base, por exemplo, na utilização de arranjos de portas lógicas e PLAs (do inglês, *programmable logic array*). Lembre-se: sempre é possível converter maxtermos para mintermos e vice-versa, basta aplicar o complemento em toda a expressão e o teorema de De Morgan. Por exemplo, se $M1 = A + B + \bar{C}$, aplicando o complemento $M1 = \overline{A + B + \bar{C}} = \overline{ABC} = \overline{ABC}$. Logo, o minitermo de M1 é \overline{ABC} .

Tabelas-verdade e o diagrama de Karnaugh

Caro aluno, iremos neste momento entender duas formas de visualizar e simplificar uma expressão lógica. A primeira é utilizando a tabela-verdade e a segunda através do diagrama de Karnaugh. Utilizamos a tabela-verdade para determinar o valor final de duas afirmações (também chamadas de proposições). Por exemplo, observe as duas proposições simples a seguir:

p : O número 25 é um quadrado perfeito

(Proposições)

q : Francisco é um estudante

Observando as duas proposições, é possível afirmar, então, que: Se o número 25 é um quadrado perfeito, então a raiz quadrada de 25 é

um número inteiro igual a 5. Ou mesmo que: Se Francisco é estudante, ou ele estuda Medicina ou não. Caro aluno, uma tabela-verdade é, então, utilizada para apresentar todos os valores lógicos possíveis que uma proposição simples (somente uma variável) ou composta (com duas variáveis ou mais) podem assumir, sendo que os únicos valores possíveis são: VERDADEIRO (**V**) ou FALSO (**F**). Iremos entender primeiramente como realizar operações lógicas utilizando uma ou duas proposições. Em seguida, iremos entender como criar uma tabela verdade. Inicialmente, utilizaremos operadores lógicos básicos para ligar duas proposições ou combiná-las.



Reflita

Quando falamos de lógica matemática com aplicações para operações em conjuntos, um conectivo lógico ou operador lógico é um símbolo ou palavra que é usada para conectar duas ou mais sentenças de modo que o sentido da sentença composta produzida dependa apenas das sentenças originais. São exemplos de conectivos lógicos \wedge : e, \vee : ou, \rightarrow : se... então, \leftrightarrow : se e somente se , \neg : negação. Considere as sentenças a seguir:

p: Francisco foi à escola ; **q**: Antônio foi à escola.

t: Francisco **e** Antônio foram à escola.

u: **Se** Francisco foi à escola, **então** Antônio foi à escola.

A estrutura lógica das afirmações **t** e **u** pode ser descrita da seguinte forma:

t : $p \wedge q$ e **u** : $p \rightarrow q$. Essas estruturas lógicas representam as afirmações em linguagem natural descritas acima. Reflita o quanto no nosso dia a dia utilizamos lógica matemática e o quanto ela pode ser útil.

Considere as principais operações e conectivos lógicos listados no Quadro 2.3 e as seguintes premissas:

p: Marcos é engenheiro; (Premissa 1)

q: Pedro é médico; (Premissa 2)

Quadro 2.3 | Conectivos lógicos e respectivos exemplos

Operação	Conectivo	Estrutura Lógica	Exemplos
Negação	\neg	Não p	Marcos não é engenheiro
Conjunção	\wedge	p e q	Marcos é engenheiro e Pedro é médico
Disjunção Inclusiva	\vee	p ou q	Marcos é engenheiro ou Pedro é médico
Condicional	\rightarrow	Se p então q	Se Marcos é engenheiro então Pedro é médico
Bicondicional	\leftrightarrow	p se e somente se q	Marcos é engenheiro se e somente se Pedro é médico

Fonte: elaborada pelo autor.

Dessa forma, a Tabela 2.9 ilustra como são as tabelas-verdade de uma proposição simples e uma proposição composta:

Tabela 2.9 | Tabelas-verdade

$$\neg p \quad p \wedge q$$

$\neg p$	
p	$\neg p$
V	F
F	V

p	q	$p \wedge q$
V	V	V
F	V	F
V	F	F
F	F	F

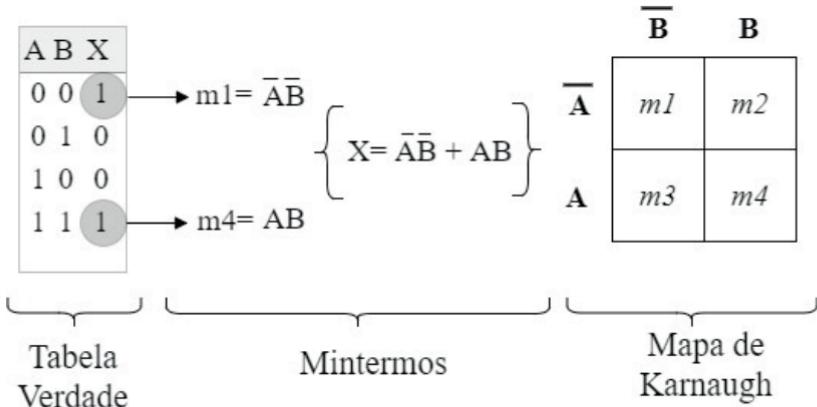
Fonte: elaborada pelo autor.

A tabela simples negação ilustra a negação da proposição p , e a tabela composta ilustra a conjunção entre a proposição p e a proposição q . Observe que o número de combinações possíveis na tabela composta é 2^n , sendo n o número de proposições. Ou seja, a tabela simples permite representar somente dois valores e a tabela composta permite representar somente quatro valores possíveis.

A segunda forma de visualizar uma expressão lógica é através do diagrama ou mapa de Karnaugh. Um mapa de Karnaugh utiliza tabelas-verdades para simplificar uma expressão lógica ou converter uma tabela-verdade no seu circuito lógico correspondente. Nesta seção, aprenderemos as regras iniciais para utilizarmos um Mapa de Karnaugh para problemas com somente duas entradas. Visualmente, um mapa de Karnaugh possui o mesmo número de quadrados que o de mintermos, ou seja, para uma função booleana de duas variáveis, existem quatro mintermos. Logo, o mapa de Karnaugh será constituído por quatro

quadrados representando cada quadro um minitermo (m_1 , m_2 , m_3 e m_4). A Figura 2.18 ilustra uma tabela-verdade, a expressão lógica resultante em minitermos e o mapa de Karnaugh com a distribuição do minitermos.

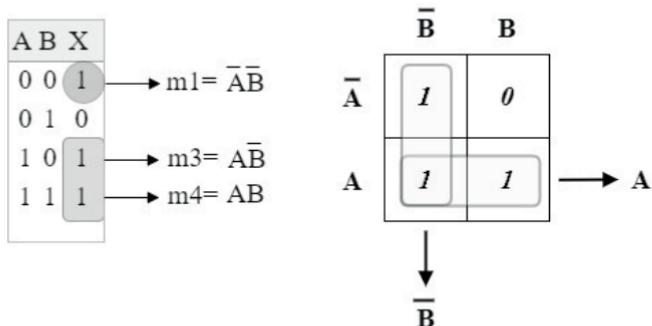
Figura 2.18 | Mapa de Karnaugh com duas variáveis



Fonte: elaborada pelo autor.

Agora, você precisa preencher no mapa de Karnaugh os respectivos níveis lógicos da coluna X da tabela-verdade. Com o mapa já construído, deve-se separar os minitermos considerando somente os campos que possuem 1 para solução final. Eles devem ser agrupados em conjuntos com quantidade de elementos múltiplos de 2 (2, 4, 6 8 etc.), e, para isso ocorrer, os elementos têm de estar lado a lado tanto na horizontal como na vertical, como exemplificado na Figura 2.19.

Figura 2.19 | Tabela-verdade e mapa de Karnaugh



Fonte: elaborada pelo autor.

O resultado final será a soma dos campos verticais que não variaram, no caso \bar{B} , e dos campos horizontais que também não variaram, no caso A. O resultado final da simplificação da tabela-verdade utilizando o mapa de Karnaugh será: $X = \bar{B} + A$. Caro aluno, na última seção desta unidade, iremos tratar mais a respeito do mapa de Karnaugh, em especial com múltiplas variáveis.

Minimização de soma dos produtos e o produto das somas

Como mencionado anteriormente, a simplificação ou manipulação algébrica da expressão lógica implica na redução do número de portas lógicas do circuito – contudo, sem que isso acarrete na alteração do valor lógico final, ou seja, simplificar a expressão não pode mudar o valor da saída. ***Se isso ocorrer, algo está errado.*** Assim, uma vez obtida a forma normal de uma expressão booleana, é possível simplificá-la por meio de manipulação algébrica, através dos postulados e propriedades da álgebra de Boole.



Exemplificando

Para entender melhor os postulados da álgebra booleana, observe a expressão $F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C}$ da Figura 2.20. Através de duas propriedades da álgebra de Boole, é possível simplificar a expressão anterior.

Figura 2.20 | Simplificação da expressão

$$\begin{aligned}
 F &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C} \\
 F &= \underbrace{\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C}_{\text{Uma vez que } A + A = A} + \underbrace{A\bar{B}\bar{C} + AB\bar{C}}_{\text{Aplicando a propriedade}} \\
 F &= \bar{A}B(\bar{C} + C) + A\bar{B}\bar{C} + B\bar{C}(A + \bar{A}) && \left. \begin{array}{l} \text{A}(B + C) = AB + AC \\ \text{Como } (A + \bar{A}) = 1 \end{array} \right\} \\
 F &= \bar{A}B + A\bar{B}\bar{C} + B\bar{C}
 \end{aligned}$$

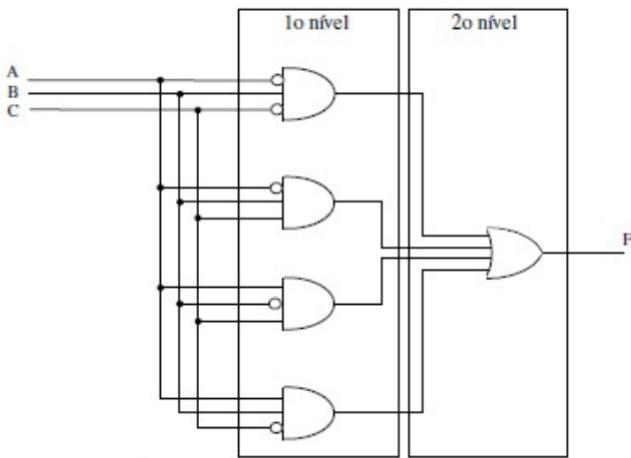
Fonte: elaborada pelo autor.

A expressão $F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C}$ continua tendo a mesma lógica, mas agora contém menos portas lógicas envolvidas $F = \bar{A}B + A\bar{B}\bar{C} + B\bar{C}$. Existem diversas vantagens em simplificar uma expressão, como confiabilidade maior, menor consumo de energia e maior velocidade operacional, já que possuem menos circuitos. (Lembre-se: quanto mais portas lógicas, maior é o atraso de propagação de um sinal.)

A Figura 2.21 ilustra a expressão não simplificada

$$F = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$$

Figura 2.21 | Circuito lógico antes da simplificação

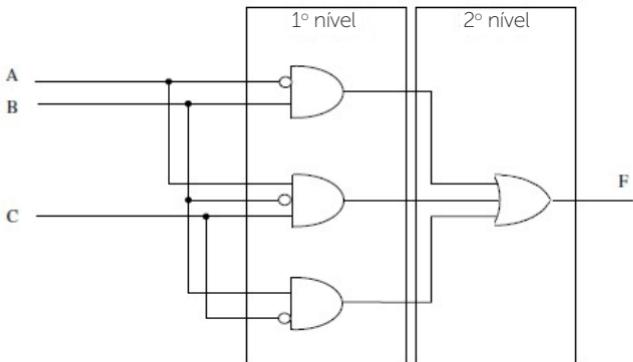


Fonte: elaborada pelo autor.

Já a Figura 2.22 ilustra a expressão simplificada

$$F = \bar{A}B + A\bar{B}C + AB\bar{C}$$

Figura 2.22 | Circuito lógico após a simplificação



Fonte: elaborada pelo autor.



Vamos analisar a complexidade dos circuitos apresentados. Para isso, é necessário somar a quantidade de portas lógicas com o mesmo número de portas de entradas e multiplicar pelas respectivas quantidades de portas de entrada. Por exemplo: observe o circuito da Figura 2.21, existem quatro portas lógicas com três entradas e uma porta lógica com quatro entradas, $4 \times 3 + 1 \times 4$. Isso produz um circuito com complexidade 16. O circuito da Figura 2.21 possui duas portas lógicas com duas entradas e duas portas lógicas com três entradas $2 \times 2 + 2 \times 3$; logo, possui complexidade igual a 10. Portanto, o circuito da Figura 2.21 é menos complexo que o circuito da Figura 2.20.

Diagrama de Karnaugh de 2, 3, 4 e 5 variáveis

Aprendemos nas seções anteriores detalhes do mapa de Karnaugh para duas variáveis. Iremos agora entender como proceder com o mapa de Karnaugh para 3, 4 e 5 variáveis. Contudo, mapas de Karnaugh acima de quatro entradas tornam-se muito complicados, sendo então recomendada a análise por meio de programas de computador. Inicialmente, trataremos de um mapa de Karnaugh com três variáveis. Utilize a seguinte Tabela 2.8 para esse exemplo, sendo **A**, **B** e **C** as entradas e **F** a saída:

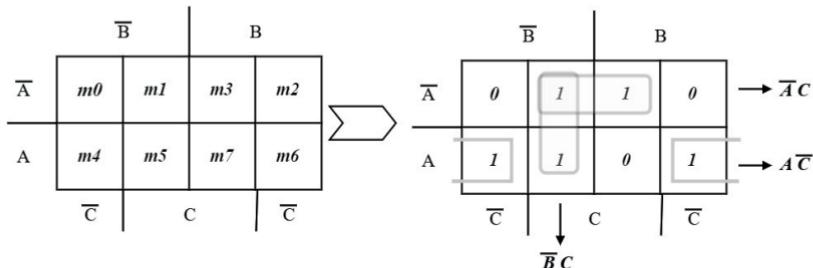
Tabela 2.8 | Tabela-verdade de uma expressão com três variáveis

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

Fonte: elaborada pelo autor.

Na tabela, selecionando os elementos que estão no nível lógico 1 na saída **F**, é possível obter a seguinte expressão: $F = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + ABC$, a qual é possível simplificar através do mapa de Karnaugh. Quando existem três variáveis, o mapa deve ser representado como ilustrado na Figura 2.23 (observe a distribuição do mintermos; fique atento).

Figura 2.23 | Mapa de Karnaugh com três variáveis



Fonte: elaborada pelo autor.

Assim, caro aluno, completando o mapa com as saídas obtidas da tabela-verdade na coluna F , temos a expressão simplificada: $F = \bar{A}C + A\bar{C} + \bar{B}C$, a qual possui o mesmo resultado da não simplificada $F = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}$.

Vamos analisar agora um mapa de Karnaugh com quatro variáveis. Observe a distribuição das variáveis e dos mintermos na Figura 2.24. Fique atento especialmente às últimas linhas.

Figura 2.24 | Mapa de Karnaugh para quatro variáveis

	\bar{C}	C			
\bar{A}	m_0	m_1	m_3	m_2	\bar{B}
A	m_4	m_5	m_7	m_6	B
	m_{12}	m_{13}	m_{15}	m_{14}	
	m_8	m_9	m_{11}	m_{10}	\bar{B}
\bar{D}		D		\bar{D}	

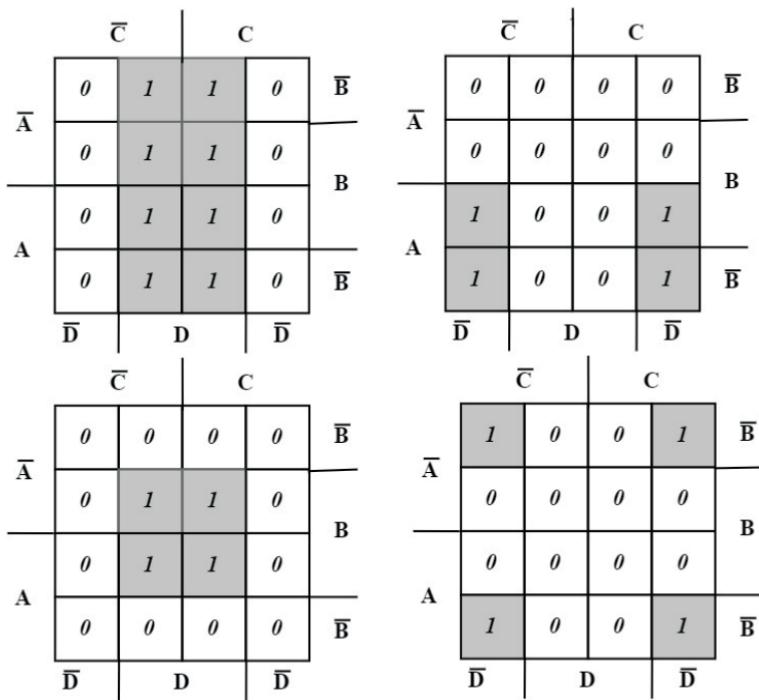
Fonte: elaborada pelo autor.



Exemplificando

A Figura 2.25 exemplifica os possíveis agrupamentos de um mapa de Karnaugh com quatro variáveis. Esteja atento aos agrupamentos que ficam nas laterais (fora do quadro) do mapa de Karnaugh, eles também precisam ser considerados. Obtenha o máximo de números uns em agrupamentos sucessivos.

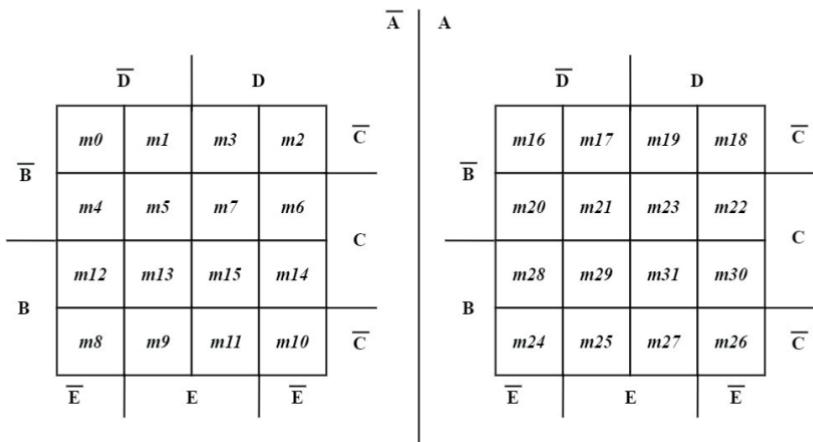
Figura 2.25 | Agrupamentos de mapas de Karnaugh com quatro variáveis



Fonte: elaborada pelo autor.

Agora, chegamos ao mapa de Karnaugh com cinco variáveis. É recomendado o uso de computadores para mapas acima de quatro variáveis. Isso se dá pela complexidade e pela necessidade de muita atenção para a distribuição dos mintermos e por causa da simplificação da expressão. Observe a complexidade do mapa de Karnaugh com cinco variáveis na Figura 2.26.

Figura 2.26 | Mapa de Karnaugh com cinco variáveis



Fonte: elaborada pelo autor.

Um mapa de Karnaugh com cinco variáveis consiste em dois mapas de quatro variáveis separados pela primeira variável e acrescido da quinta variável. Caso você se depare com uma situação como essa, analise a necessidade de alguma das variáveis e, caso não seja possível excluí-la, use alguma ferramenta computacional para resolver o problema.

Prezado aluno, o assunto não se esgota com o que foi apresentado aqui. Procure avançar no conteúdo pesquisando outras fontes. A álgebra booleana foi e sempre será fundamental para o entendimento da computação. Bons estudos!

Sem medo de errar

Como mencionado no início da unidade, surgiu a possibilidade de a empresa na qual você trabalha olhar novos mercados. Existe a necessidade de monitorar a tensão da bateria de um carro elétrico. Para isso, será necessário que um circuito fique monitorando a saída de um conversor analógico-digital e que envie um pulso quando o valor for maior que 6 volts (o 6 está excluído). A solução para esse problema passa inicialmente em estabelecer a tabela-verdade do circuito. Observe a Tabela 2.8, que lista os valores dos sinais de entrada $ABCD$ e a respectiva saída Z .

Tabela 2.9 | Tabela-verdade

	A	B	C	D	Z	
(0)	0	0	0	0	0	-
(1)	0	0	0	1	0	-
(2)	0	0	1	0	0	-
(3)	0	0	1	1	0	-
(4)	0	1	0	0	0	-
(5)	0	1	0	1	0	-
(6)	0	1	1	0	0	-
(7)	0	1	1	1	1	$\bar{A}BCD$
(8)	1	0	0	0	1	$A\bar{B}\bar{C}\bar{D}$
(9)	1	0	0	1	1	$A\bar{B}\bar{C}D$
(10)	1	0	1	0	1	$A\bar{B}CD$
(11)	1	0	1	1	1	$A\bar{B}CD$
(12)	1	1	0	0	1	$AB\bar{C}\bar{D}$
(13)	1	1	0	1	1	$AB\bar{C}D$
(14)	1	1	1	0	1	$ABC\bar{D}$
(15)	1	1	1	1	1	$ABCD$

Fonte: elaborada pelo autor.

A tabela-verdade fornece a seguinte expressão na forma de soma de produtos:

$$Z = \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD + AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD$$

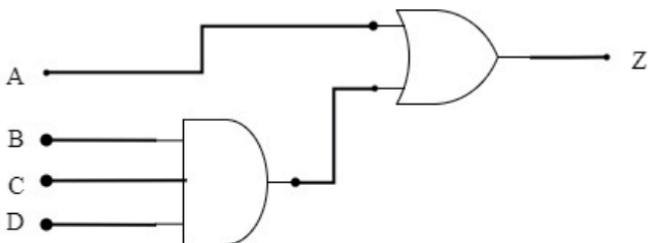
Agora, temos a tarefa mais difícil, a simplificação e eliminação de termos. O resultado final da simplificação é $Z = \bar{A}BCD + A$. Contudo, essa expressão ainda pode ser reduzida utilizando o teorema que diz que $X + \bar{X}Y = X + Y$, sendo que $X = A$ e $Y = BCD$.

Portanto, a expressão final será:

$$Z = BCD + A$$

Aqui fica bem claro o quanto é importante simplificar uma expressão lógica. A Figura 2.27 exemplifica a lógica final.

Figura 2.27 | Lógica do sistema de monitoramento de bateria de carro elétrico



Fonte: elaborada pelo autor.

Avançando na prática

Comparação de binários

Descrição da situação-problema

Você desenvolveu uma excelente solução para carregamento de carros elétricos. Provavelmente, a empresa que você trabalha vai ampliar o mercado e atuar no segmento de carros elétricos. A próxima etapa é comparar quando duas baterias estão com a mesma tensão, a fim de que, através de mecanismos de análise de carga, o consumo da bateria seja mais bem distribuído. Para tanto, você precisa saber que um número binário pode ser representado pela notação X_1X_0 e pode representar qualquer valor entre 00, 01, 10 e 11. De forma similar, você pode representar um outro número através da notação Y_1Y_0 , que pode também representar qualquer valor entre 00, 01, 10 e 11. Projete um circuito lógico usando as entradas X_1X_0 Y_1Y_0 , cuja saída seja nível ALTO apenas quando dois números binários X_1X_0 e Y_1Y_0 forem iguais. Diante desse contexto, uma tabela-verdade pode ajudar na formulação dessa lógica? A porta lógica XNOR simplificaria o circuito? Inclua no seu relatório final a solução com e sem simplificação.

Resolução da situação-problema

O primeiro passo para resolver o problema é construir uma tabela-verdade com as dezesseis possibilidades. A Tabela 2.9 lista todas as possíveis possibilidades.

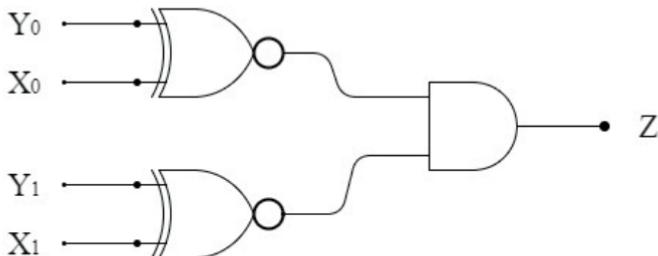
Tabela 2.10 | Tabela-verdade do comparador entre dois números binários

X_1	X_0	Y_1	Y_0	Z
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0
1	1	1	1	1

Fonte: elaborada pelo autor.

A saída Z terá nível alto sempre que os valores X_1X_0 e Y_1Y_0 forem iguais. Nesse sentido, analisando a Tabela 2.10, existem quatro momentos em que isso ocorre. Analise esses momentos e aplique a soma de produtos para achar a equação. Uma vez montada a expressão, compare com o circuito lógico da Figura 2.27 e analise se o uso da porta lógica XNOR compensa.

Figura 2.28 | Lógica XNOR para comparação de dois números binários



Fonte: elaborada pelo autor.

Inclua uma tabela-verdade comparando a solução com XNOR em seu relatório final. Explique qual a função da porta lógica AND ao final do circuito. Bons estudos!

Faça valer a pena

1. Mintermos e maxtermos correspondem a uma forma utilizada para reescrever uma expressão lógica. Essa forma é o início para padronizar a expressão para se obter a expressão de forma mais simplificada.

Qual é o mintermo do maxtermo $M_1 = A + B + C + D$?

- a) $ABCD$
- b) $ABC\bar{D}$
- c) $\bar{A}BC\bar{D}$
- d) $\bar{A}\bar{B}CD$
- e) $\bar{A}\bar{B}\bar{C}D$

2. Existem duas formas de visualizar e simplificar uma expressão lógica. A primeira é utilizando a tabela-verdade e a segunda através do diagrama de Karnaugh. Através da tabela-verdade, é possível, por exemplo, determinar o valor final de duas afirmações (também chamada de proposições).

Quais operações descrevem as proposições a seguir?

João não é pintor

João é professor **e** Gustavo é piloto

João é professor **se e somente se** Gustavo é piloto

- a) $p : \neg p; p \wedge q; p \rightarrow q$
- b) $p : p; p \vee q; p \rightarrow q$
- c) $p : \neg p; p \vee q; p \leftrightarrow q$
- d) $p : \neg p; p \wedge q; p \leftrightarrow q$
- e) $p \wedge q; p \wedge q; p \leftrightarrow q$

3. Mapa de Karnaugh é um método de simplificação gráfico criado por Edward Veitch (1952) e aperfeiçoado pelo engenheiro de telecomunicações Maurice Karnaugh. Chamamos esse diagrama de mapa, visto esse ser um mapeamento biúnico a partir de uma tabela-verdade da função que está a ser analisada.

Ele é utilizado para simplificar uma equação lógica ou para converter uma tabela-verdade no seu circuito lógico correspondente.

Fonte: Wikipédia, Mapa de Karnaugh.

Encontre a expressão que melhor se adéqua ao mapa de Karnaugh a seguir.

		\bar{B}			B	
\bar{A}	0	0	1	1		
<hr/>						
A	0	1	1	0		
	\bar{C}		C		\bar{C}	

- a) $S = \bar{A}B + AB + BC$
- b) $S = AC$
- c) $S = ABC$
- d) $S = \bar{A}B + BC$
- e) $S = \bar{A}B + AC$

Referências

ATMEL Semiconductors. **ATF22LV10C datasheet**. 2017.

CIRCUIT Simulator, 2017. Disponível em: <<http://www.falstad.com/circuit/>>. Acesso em: 24 out. 2017.

CARRO, Luigi. **Projeto e prototipação de sistemas digitais**. Porto Alegre: UFRGS, 2001.

FLOYD, Thomas. **Sistemas digitais**: fundamentos e aplicações. Porto Alegre. Bookman, 2009.

LOURENÇO, Antônio Carlos de; CRUZ, Eduardo Cesar Alves; FERREIRA, Sabrina Rodero; CHOUERI JR., Salo. **Circuitos digitais**: estude e use. São Paulo: Érica, 1997.

MODELSIM PE Student Edition. **Mentor**, 2017. Disponível em: <https://www.mentor.com/company/higher_ed/modelsim-student-edition>. Acesso em: 12 nov. 2017.

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. **Sistemas digitais**: princípios e aplicações. 10. ed. São Paulo: Pearson, 2007.

VAHID, Frank. **Sistemas digitais**: projeto, otimização e HDLS. Porto Alegre: Bookman, 2008.

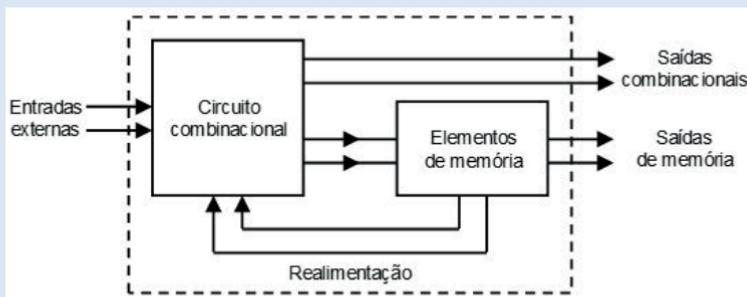
WESTE, N. H. E.; ESHRAGHIAN, K. **Principles of CMOS VLSI design**. Boston, Massachusetts, EUA. 2. ed. Reading, Addison-Wesley Publishing Company, 1993.

Lógica sequencial

Convite ao estudo

Na unidade anterior, estudamos circuitos considerados combinacionais, porque suas saídas, em qualquer instante de tempo, dependem apenas dos níveis lógicos presentes nas entradas no mesmo instante. Nesta unidade, conheceremos os circuitos lógicos sequenciais. Segundo Szajnberg (2014), um circuito sequencial difere de um combinacional pela presença de memória e de uma ou mais malhas de realimentação. A Figura 3.1 mostra um diagrama de blocos de um sistema digital hipotético, que agrupa um circuito combinacional com elementos de memória.

Figura 3.1 | Diagrama de blocos de um sistema hipotético



Fonte: elaborada pelo autor.

Para Vahid (2008), um circuito sequencial é um circuito cujas saídas dependem não somente das entradas atuais, mas também do seu estado atual, que é o conjunto de todos os bits armazenados no circuito. O estado do circuito, por sua vez, depende da sequência passada de valores que apareceram nas entradas do circuito.

Os sinais de controle nos circuitos sequenciais são gerados sequencialmente e se relacionam não somente com as funções

de entrada presentes, mas também com os estados anteriores do sistema armazenado pela memória. Os estados são fornecidos justamente pelas vias de realimentação saída-entrada. Logo, a saída de um sistema sequencial, em qualquer instante, depende da entrada presente e das condições dos sinais da entrada no passado.

Nesse contexto, como forma de fixar os conceitos que serão apresentados, vamos pensar na seguinte situação: você desenvolveu algumas habilidades trabalhando em uma empresa de telecomunicações, começou na linha de montagens e sua curiosidade e vontade de aprender o levou para a equipe de pesquisa e desenvolvimento. Agora, percebendo um nicho no mercado, você e alguns amigos decidiram montar uma *start-up* de desenvolvimento de projetos de sistemas de automação. Vocês sabem que, para conquistar o mercado, terão que ser criativos e, sobretudo, econômicos em seus projetos. Você está pronto para esse desafio?

Para auxiliá-lo, esta unidade tem por objetivo apresentar alguns conceitos e elementos fundamentais para que, além de ter domínio do conteúdo relacionado aos circuitos combinacionais, você seja também capaz de conhecer e utilizar dispositivos de memória na implementação de circuitos lógicos sequenciais. Fique atento aos conceitos que serão apresentados nesta unidade.

Pronto para mais uma? Esperamos que sim! Bons estudos.

Seção 3.1

Flip-flops

Diálogo aberto

Para construir um circuito sequencial, precisamos de um bloco construtivo que nos capacite a armazenar um bit. Armazenar um bit significa que podemos armazenar um bit no bloco e depois voltar para ver o que foi armazenado. Esta seção começa com o estudo dos fundamentos da lógica sequencial. O comportamento de um circuito sequencial não depende somente das entradas em um determinado instante; o estado atual do circuito também afeta a sua saída.

Imagine que sua *start-up* de desenvolvimento de projetos de sistemas de automação foi contratada por uma rede de mercados local para um primeiro projeto: construir o sistema de botão de chamada de auxílio para o caixa – como o esquema visto na Figura 3.2, a seguir.

Figura 3.2 | Sistema do botão de chamada de auxílio para o caixa



Fonte: elaborada pelo autor.

O operador de caixa pode apertar o botão “chamar” para que uma lâmpada vermelha seja acesa acima do seu caixa. Isso indica ao auxiliar que ele precisa de algum tipo de ajuda. A lâmpada permanecerá acesa, mesmo que o botão “chamar” seja solto. Para apagar a lâmpada, é preciso apertar o botão “cancelar”.

Para ajudar a realizar esse projeto, abordaremos aqui os dispositivos lógicos biestáveis, que são o *latch* e o *flip-flop*. Os dispositivos biestáveis são dispositivos de armazenamento temporário que têm dois estados estáveis, chamados SET e RESET. A diferença básica entre *latches* e *flip-flops* é a forma com que eles comutam de um estado para o

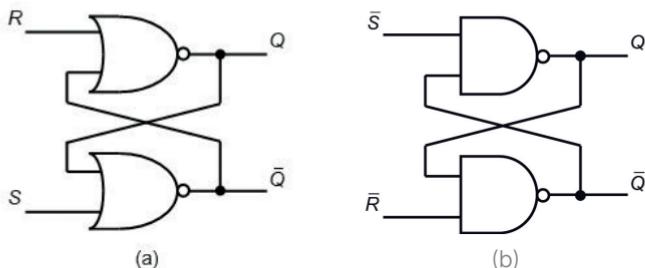
outro. O *flip-flop* é a unidade básica para construção de contadores, registradores e outras lógicas de controle sequencial e é usado em certos tipos de memórias.

Desejamos bons estudos e ótimo trabalho!

Não pode faltar

Segundo Floyd (2007), um **latch** é um tipo de dispositivo lógico de armazenamento temporário que tem dois estados estáveis e por isso é chamado de **biestável** ou **multivibrador**. Um **latch S-R** (SET-RESET) com entrada ativa em nível ALTO é composto de duas portas NOR com acoplamento cruzado, conforme pode ser visto na Figura 3.3 (a). Um **latch $\bar{S}-\bar{R}$** com entrada ativa em nível BAIXO, por sua vez, é composto por duas portas NAND com acoplamento cruzado, conforme pode ser visto na Figura 3.3 (b).

Figura 3.3 | Latch S-R (a) com entrada ativa em nível ALTO; (b) com entrada ativa em nível BAIXO



Fonte: elaborada pelo autor.

Para entendermos o funcionamento de um latch, vamos considerar o latch S-R composto por portas NOR da Figura 3.3 (a). De acordo com o funcionamento de uma porta NOR de duas entradas, podemos dizer que o nível lógico 0 em uma das entradas faz com que a saída seja a outra entrada complementada. Já o nível lógico 1 em uma das entradas força a saída a permanecer no estado lógico 0.

Vamos recordar a tabela-verdade da porta NOR na Tabela 3.1.

Tabela 3.1 | Tabela-verdade da porta NOR

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Fonte: elaborada pelo autor.

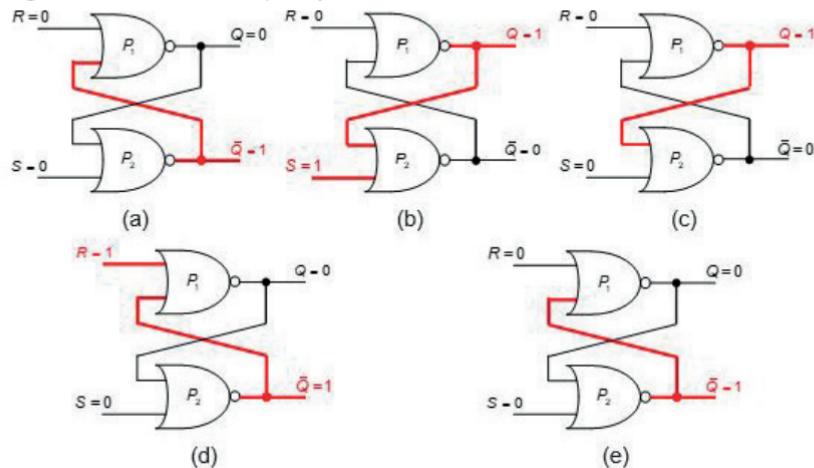
O *latch* mostrado na Figura 3.3 (a) tem duas entradas, S e R , e duas saídas, Q e \bar{Q} . Vamos iniciar considerando que as duas entradas e a saída Q estão no nível lógico 0. Como a saída Q é conectada de volta na entrada da porta P_2 e a entrada S está no nível 0, a saída \bar{Q} vai para o nível 1. Essa saída é acoplada de volta na entrada da porta P_1 , garantindo que sua saída seja 0, como mostrado na Figura 3.4 (a). Quando a saída Q é 0, o *latch* está no estado de *RESET* e permanece nesse estado indefinidamente até que um nível 1 seja aplicado na entrada S .

Com a entrada S é igual a 1, a saída da porta P_2 é forçada para 0. O acoplamento entre as portas faz com que uma entrada da porta P_1 seja 0. Como a entrada R está no nível 0, então a saída Q é levada para o nível 1 e o acoplamento entre as portas garante que a saída \bar{Q} seja 0, como pode ser visto na Figura 3.4 (b).

O nível 1 na saída Q , de fato, garante que a saída \bar{Q} seja 0 mesmo quando o nível 1 da entrada S seja removido, como pode ser visto na Figura 3.4 (c). Quando a saída Q é 1, o *latch* está no estado de *SET* até que um nível 1 seja aplicado na entrada R .

Fazendo a entrada R igual a 1, a saída Q é levada novamente para 0, o acoplamento entre as portas, e o fato de que a entrada S está no nível 0 fazem com que a saída \bar{Q} seja levada para 1, como vemos na Figura 3.4 (d). Como podemos notar, o *latch* foi levado novamente para o estado *RESET*, pois mesmo quando a entrada R volta para o nível 0, a saída Q mantém-se no nível 0, como pode ser visto na Figura 3.4 (e), que é exatamente igual à Figura 3.4 (a).

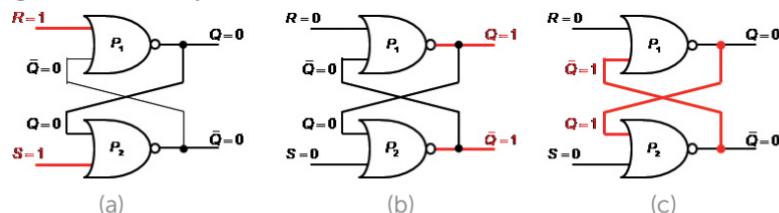
Figura 3.4 | Modos de operação do latch S-R básico



Fonte: elaborada pelo autor.

Quando as duas entradas S e R são colocadas no nível lógico 1 ao mesmo tempo, tanto a saída Q quanto \bar{Q} são forçadas para 0 (Figura 3.5 (a)), violando a condição básica de complementariedade entre essas duas saídas. Mas o problema mais sério ocorre quando tornamos as entradas S e R 0 novamente. Nesse caso, é possível que as duas entradas retornem ao valor 0 simultaneamente. Quando isso ocorre, as portas P_1 e P_2 terão em todas as suas entradas o nível lógico 0, fazendo com que suas saídas mudem para 1, como mostrado na Figura 3.5 (b). No entanto, as portas são realimentadas com as saídas, que estão agora no nível lógico 1, forçando as saídas para 0, como vemos na Figura 3.5 (c). Os níveis 1 nas portas levarão as saídas para 1, e esse ciclo se repetirá, criando um estado oscilatório no *latch*. No entanto a oscilação não é uma característica desejável em um bloco de armazenamento de memória.

Figura 3.5 | Oscilação devido ao estado inválido no latch S-R básico



Fonte: elaborada pelo autor.

De modo similar à tabela-verdade do sistema combinacional, a **tabela de estado** enumera as entradas e saídas com colunas adicionais para os estados presentes e seguintes do circuito sequencial, como podemos ver na Tabela 3.2 para o *latch S-R*.

Tabela 3.2 | Tabela de estado do *latch S-R*

S	R	$Q(t)$	$Q(t+1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	Indefinido
1	1	1	Indefinido

Fonte: elaborada pelo autor.

A tabela excitação, por sua vez, lista os valores das entradas do circuito sequencial, bem como as mudanças entre os estados presentes e seguintes, como é visto na Tabela 3.3 para o *latch S-R*.

Tabela 3.3 | Tabela excitação do *latch S-R*

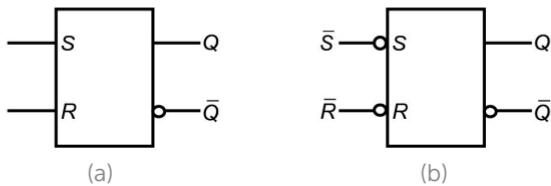
S	R	$Q(t+1)$	Estado
		$Q(t)$	
0	0	0	Repouso
0	1	0	RESET
1	0	1	SET
1	1	?	Inválido

Fonte: elaborada pelo autor.

A tabela excitação é geralmente usada no projeto e na síntese de circuitos sequenciais.

Os símbolos lógicos para os *latches* com entradas ativas em nível ALTO e entradas ativas em nível BAIXO são mostrados na Figura 3.6.

Figura 3.6 | Símbolo lógico para o *latch*: (a) S-R com entradas ativas em nível ALTO; (b) $\bar{S}-\bar{R}$ com entradas ativas em nível BAIXO



Fonte: elaborada pelo autor.



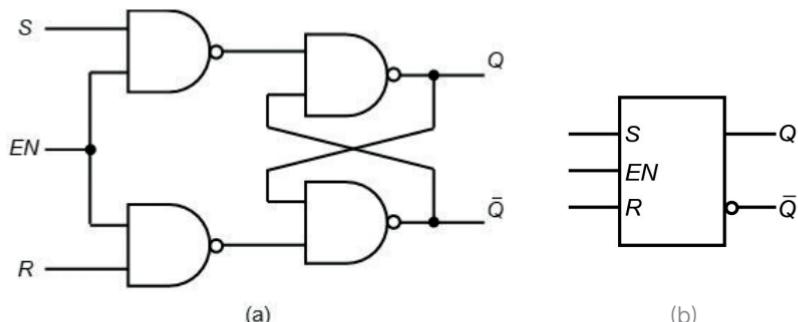
Pesquise mais

O latch $\bar{S}-\bar{R}$ com portas NAND tem funcionamento idêntico ao do latch S-R, porém sua entrada é ativada em nível BAIXO. Para uma análise de funcionamento, semelhante à feita aqui para o latch $\bar{S}-\bar{R}$, consulte a Seção 7.1 do livro:

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

Em um **latch controlado**, a transição entre os estados ocorre somente sob a ação da subida de um pulso em uma entrada de habilitação, *EN* – essa entrada também pode ser chamada de *gatilho* e, portanto, a letra *G* também pode ser usada para indicar essa entrada. O diagrama lógico e o símbolo lógico para um **latch S-R controlado** podem ser vistos na Figura 3.7.

Figura 3.7 | Latch S-R controlado (a) diagrama lógico; (b) símbolo lógico

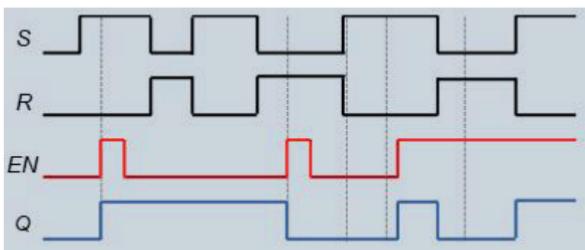


Fonte: elaborada pelo autor.



As entradas S e R controlam o estado para o qual o *latch* irá quando um nível ALTO é aplicado na entrada EN . O *latch* não mudará de estado enquanto EN estiver em nível BAIXO. Porém, durante todo o tempo em que essa entrada permanecer em nível ALTO, a saída é determinada pelos estados das entradas S e R . Nesse circuito, o estado inválido ocorre quando S e R tiverem simultaneamente nível ALTO. Um exemplo de diagrama de tempo do *latch* S-R controlado pode ser visto na Figura 3.8.

Figura 3.8 | Funcionamento do *latch* S-R controlado

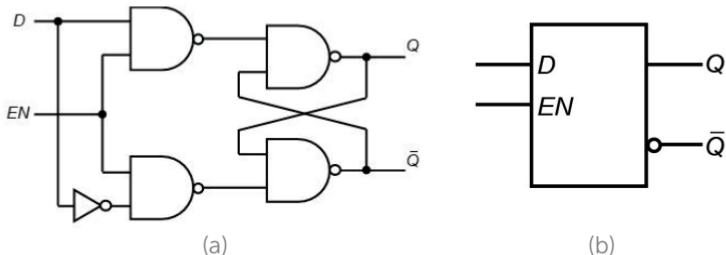


Fonte: elaborada pelo autor.

Os gráficos com "nível lógico versus tempo" fornecem não só uma representação visual do desenvolvimento de sinais no tempo, mas também uma demonstração gráfica de comparação entre sinais em vários pontos de um circuito de chaveamento. Os diagramas de tempo são muito usados em sistemas sequenciais.

O ***latch D*** é um outro tipo de *latch* controlado. Ele teve sua origem devido à necessidade de se evitar, no *latch* S-R, a ocorrência do estado proibido. Por isso, o *latch* D tem apenas uma entrada além da EN. A entrada mencionada é denominada de entrada D (dado). O diagrama lógico e o símbolo lógico para um *latch* D são mostrados na Figura 3.9.

Figura 3.9 | Latch D controlado (a) diagrama lógico; (b) símbolo lógico



Fonte: elaborada pelo autor.

Quando a entrada D for de nível ALTO e a entrada EN for de nível ALTO, o *latch* será levado para o estado *SET*. Quando a entrada D for de nível BAIXO e a entrada EN for de nível ALTO, o *latch* será levado para o estado *RESET*. Dito de outra forma, a saída Q segue a entrada D quando EN for de nível ALTO.

Em lógica sequencial, temos dois modos básicos de operação: **assíncrono** e **síncrono**. No modo assíncrono, os circuitos que constituem o sistema funcionam com tempos independentes entre si. Os estados de saída são gerados imediatamente após a aplicação dos sinais de entrada, controlados pela realimentação direta, usando estritamente os retardos de propagação dos decodificadores do estado seguinte.

Os *latches* são um exemplo de dispositivo assíncrono. No modo síncrono, os ***flip-flops*** são chaveados por um trem de pulsos denominado relógio do sistema ou, como é internacionalmente conhecido, “clock do sistema” (*system clock* ou simplesmente *clock*). O *clock* representa o sinal de comando de um sistema sequencial e consiste em um sinal digital permanente, geralmente de alta frequência. A Figura 3.10 traz um exemplo de um sinal de *clock*. A seta evidencia qual borda é utilizada para acionar o *flip-flop*.

Figura 3.10 | Trem de pulsos de um sinal de *clock*: (a) borda positiva; (b) borda negativa



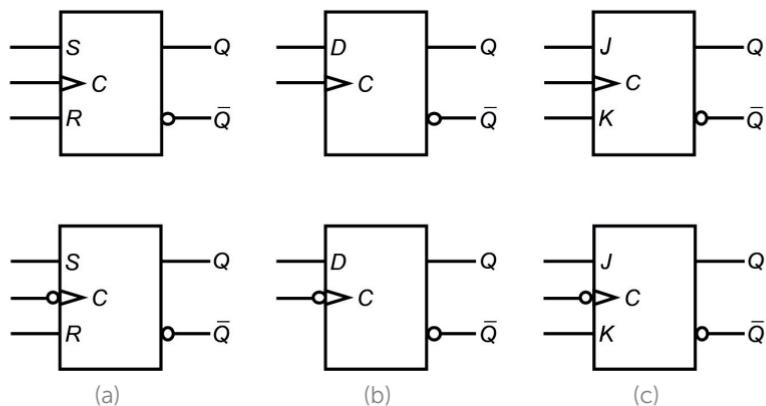
Fonte: elaborada pelo autor.

Quando um *latch* controlado é adaptado para funcionar de maneira síncrona, ele passa a ser chamado de *flip-flop S-R disparado por borda*.

Apresentaremos, a seguir, três tipos de *flip-flops* disparados por borda: S-R, D e J-K. Na prática, o *flip-flop* S-R não está disponível na forma de CI, mas ele é a base para os *flip-flops* D e J-K, por isso é importante entendermos o seu funcionamento.

A Figura 3.11 traz os símbolos lógicos para esses *flip-flops*. O pequeno triângulo dentro do bloco indica a entrada de *clock* (C) e é denominado indicador de entrada dinâmica. Os *flip-flops* podem ser disparados tanto na borda positiva (parte superior da figura) quanto na negativa (parte inferior da figura), indicados pelo pequeno círculo na entrada C (FLOYD, 2007).

Figura 3.11 | Símbolos lógicos de *flip-flops* disparados por borda (a) S-R (b) D (c) J-K



Fonte: adaptada de Floyd (2007, p. 394).

No ***flip-flop S-R disparado por borda*** (Figura 3.11(a)), as entradas S e R são denominadas síncronas, uma vez que os bits nas entradas são transferidos para a saída do *flip-flop* apenas na borda de disparo do pulso de *clock*.

Quando $SR=10$, somente na borda de disparo do pulso de *clock* é que o *flip-flop* será levado para o estado *SET*, com a saída $Q=1$. O mesmo ocorre quando $SR=01$: a saída somente será $Q=0$ na próxima borda de disparo do pulso de *clock*, estando o *flip-flop* no estado *RESET*. Quando $SR=00$, a saída não muda de estado, permanecendo no estado anterior. O mesmo fenômeno oscilatório descrito para o *latch S-R* ocorre para o *flip-flop* quando $SR=11$, por isso essa condição é considerada inválida.



Agora que você conhece o funcionamento do *latch* S-R controlado e do *flip-flop* S-R, qual é exatamente a diferença entre esses dois dispositivos?

A Tabela 3.4 traz a tabela de excitação do *flip-flop* S-R disparado por borda positiva. Lembre-se: o *flip-flop* somente muda de estado na borda de disparo de um pulso de *clock*, portanto qualquer alteração nas entradas do *flip-flop* não alteram o seu estado, com exceção do pequeno intervalo em torno da transição do *clock*.

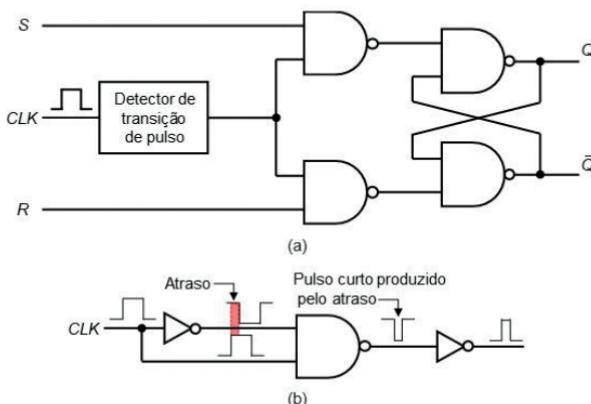
Tabela 3.4 | Tabela excitação do *flip-flop* S-R disparado por borda positiva

S	R	C	$Q(t+1)$	Estado
0	0	X	$Q(t)$	Repouso
0	1	↑	0	RESET
1	0	↑	1	SET
1	1	↑	?	Inválido

Fonte: elaborada pelo autor.

Na Figura 3.12 (a) é possível ver uma implementação simplificada de um *flip-flop* S-R disparado por borda. Usaremos essa implementação para demonstrar o conceito de disparo por borda. O *flip-flop* S-R difere do *latch* S-R controlado apenas pelo fato de que o primeiro possui um detector de transição de pulso (FLOYD, 2007). Um exemplo básico de detector de transição de pulso é mostrado na Figura 3.12 (b).

Figura 3.12 | Disparo por borda: (a) diagrama lógico simplificado do *flip-flop* S-R disparado por borda positiva; (b) exemplo de detector de transição de pulso

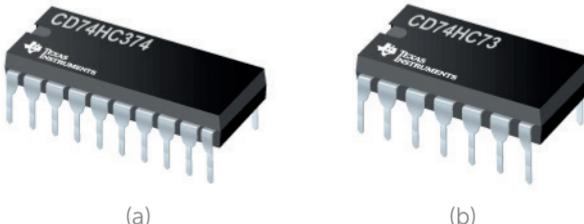


Fonte: adaptada de Floyd (2007, p. 394).

Esse tipo de detector de transição de pulso faz uso do atraso ocasionado pela primeira porta inversora. Assim, o pulso de *clock* invertido chega alguns nanosegundos depois na porta NAND em relação ao pulso original. Esse circuito produz um pico de duração muito curta na transição positiva do pulso de *clock*. Em um *flip-flop* disparado pela borda negativa, o sinal do *clock* é invertido primeiro, produzindo um pico na borda negativa.

Os *flip-flops* D e J-K são comercializados na forma de CI e são mais utilizados que o tipo S-R, tal como o *flip-flop* tipo D CD74HC374, mostrado na Figura 3.13 (a), e o *flip-flop* tipo J-K, na Figura 3.13 (b), ambos da Texas Instruments. No entanto, eles são derivados do *flip-flop* S-R.

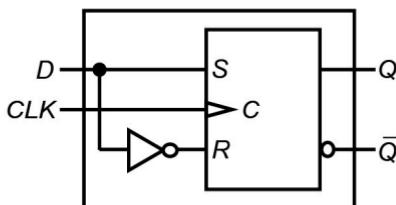
Figura 3.13 | Circuito integrado de flip-flop (a) tipo-D (b) tipo J_K



Fontes: (a) <<http://www.ti.com/graphics/folders/partimages/CD74HC374.jpg>>; (b) <<http://www.ti.com/graphics/folders/partimages/CD74HC73.jpg>>. Acesso em: 21 fev. 2018.

O **flip-flop D** (Figura 3.10 (b)) é usado quando um único bit de dado (1 ou 0) deve ser armazenado. Um *flip-flop* D básico é obtido por um *flip-flop* S-R, invertendo a entrada S e conectando esse sinal à entrada R, como pode ser visto na Figura 3.14. Observe que esse tipo de *flip-flop* possui apenas uma entrada além do *clock*. Caso a entrada D seja 1, quando um pulso de *clock* é aplicado, o *flip-flop* irá para o estado *SET* e o nível 1 da entrada será armazenado pelo *flip-flop*. Caso exista um nível 0 na entrada D quando um pulso de *clock* é aplicado, o *flip-flop* irá para o estado *RESET* e o nível 0 da entrada será armazenado pelo *flip-flop*.

Figura 3.14 | *Flip-flop D* disparado por borda positiva



Fonte: elaborada pelo autor.

A excitação do *flip-flop D* disparado por borda positiva pode ser vista na Tabela 3.5. A operação de um dispositivo disparado por borda negativa é evidentemente a mesma, exceto pelo disparo ocorrer na borda de descida do pulso de *clock*. Lembre-se: a saída Q segue a entrada D na borda ativa ou de disparo do *clock*.

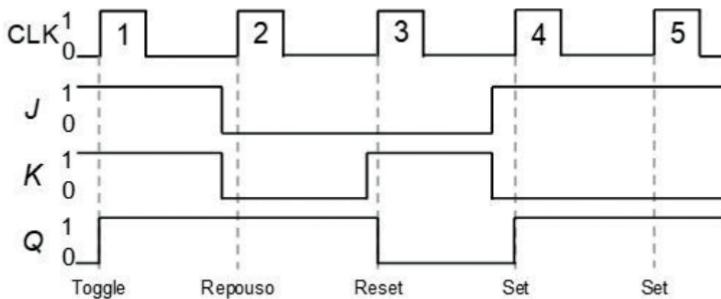
Tabela 3.5 | Tabela de excitação do *flip-flop D* disparado por borda positiva

D	C	$Q(t+1)$	Estado
0	↑	0	RESET
1	↑	1	SET

Fonte: elaborada pelo autor.

O **flip-flop J-K** (Figura 3.11 (c)) é versátil e amplamente usado, além disso funciona da mesma forma que o *flip-flop* S-R nas condições de operação de *SET*, *RESET* e repouso, como indicado na Figura 3.15.

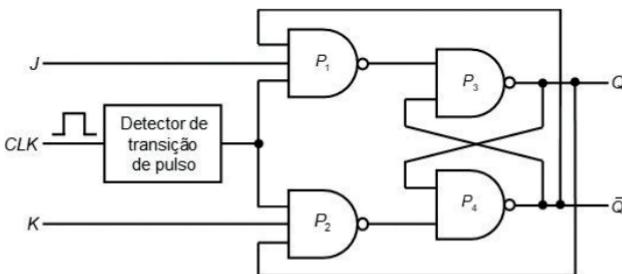
Figura 3.15 | Operação de um *flip-flop* tipo J-K



Fonte: adaptada de Floyd (2007, p. 401).

A diferença entre os dois é que o *flip-flop* J-K não possui um estado inválido, como acontece com o *flip-flop* S-R. A Figura 3.16 mostra a lógica interna básica para um *flip-flop* J-K disparado por borda. Note que a saída Q é conectada de volta na entrada da porta P_2 e que a saída \bar{Q} é conectada de volta na entrada da porta P_1 . As duas entradas de controle são denominadas J e K em homenagem a Jack Kilby, inventor do circuito integrado (FLOYD, 2007).

Figura 3.16 | Diagrama lógico simplificado para um *flip-flop* J-K disparado por borda positiva



Fonte: elaborada pelo autor.

Um *flip-flop* J-K também pode ser do tipo disparado por borda negativa, caso no qual a entrada de clock é invertida.

A operação lógica do *flip-flop* J-K é a mesma que a do tipo S-R para as condições de *SET* e *RESET*. A diferença ocorre quando as entradas J e K estiverem ambas em nível ALTO. Para entender isso, considere que o *flip-flop* esteja no estado de *RESET*. O nível lógico

1 em \bar{Q} habilita a porta P_1 , assim uma transição no *clock* leva o *flip-flop* para o estado *SET*, de modo que agora existe um nível 1 em Q, permitindo que a próxima transição do *clock* passe pela porta P_2 e reinicie o *flip-flop*. Podemos notar que, a cada transição sucessiva do *clock*, o *flip-flop* muda para o estado oposto ao anterior. Esse estado é denominado **toggle** (comutação). A Tabela 3.6 traz a tabela de excitação do *flip-flop* J-K disparado por borda positiva. Observe que não existe estado inválido como ocorre com o *flip-flop* S-R. A tabela de excitação para um dispositivo disparado por borda negativa é idêntica a esta tabela, exceto pelo *flip-flop* ser disparado na borda de descida do pulso de *clock*.

Tabela 3.6 | Tabela excitação do *flip-flop* J-K disparado por borda positiva

<i>J</i>	<i>K</i>	<i>C</i>	$Q(t+1)$	Estado
0	0	↑	$Q(t)$	Repouso
0	1	↑	0	RESET
1	0	↑	1	SET
1	1	↑	$\bar{Q}(t)$	Toggle

Fonte: elaborada pelo autor.

Um *flip-flop* J-K conectado para o modo *toggle* é denominado algumas vezes de ***flip-flop T*** e tem a tabela de excitação mostrada na Tabela 3.7.

Tabela 3.7 | Tabela excitação do *flip-flop T* disparado por borda positiva

<i>T</i>	<i>C</i>	$Q(t+1)$	Estado
0	↑	$Q(t)$	Repouso
1	↑	$\bar{Q}(t)$	Toggle

Fonte: elaborada pelo autor.



Pesquise mais

O desempenho, os requisitos de operação e as limitações dos *flip-flops* são especificados por diversas características de operação ou parâmetros encontrados nas folhas de dados dos dispositivos. Geralmente, essas especificações são aplicáveis a todos os *flip-flops* CMOS e TTL. (FLOYD, 2007, p. 406)

Para saber mais sobre esse assunto, leia a Seção 7.3 do livro:

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

Sem medo de errar

Sua *start-up* de desenvolvimento de projetos de sistemas de automação fechou seu primeiro contrato: construir o sistema de botão de chamada de auxílio para o caixa por uma rede de mercados local.

É necessário ter em mente que uma pequena lâmpada vermelha acima do caixa deve acender quando o operador de caixa pressionar o botão “chamar” e que a lâmpada deve permanecer acesa, mesmo após o botão de chamada ter sido solto, de modo que precisamos de um meio para “lembra” que o botão foi pressionado. Isso pode ser feito usando um bloco de armazenamento de um bit que armazena 1, quando o botão de chamar é apertado, ou 0, quando o botão de cancelar é apertado. A saída desse bloco é, então, ligada à lâmpada vermelha, como foi indicado já na Figura 3.2.

Com o conhecimento que você já adquiriu até esse momento, você sabe que poderia usar um *latch* S-R básico ou um *latch* S-R controlado, ou ainda um *flip-flop* do tipo D para construir esse bloco de armazenamento. Como você pode notar, um *latch* S-R básico é mais simples, em termos construtivos, que um *flip-flop* do tipo D, mas essa simplicidade acarreta alguns problemas.

Esteja ciente de que, como afirma Vahid (2008), os projetistas raramente usam blocos de armazenamento de bits que não sejam os *flip-flops* do tipo D, por isso avance nas suas pesquisas e responda a seguinte reflexão para que seja possível concluir o seu projeto: se o *flip-flop* tipo D possui apenas uma entrada, como utilizar no seu sistema de duas entradas, o botão “chamar” e o botão “cancelar”?

Avançando na prática

Aplicação de *latches* em chaves mecânicas

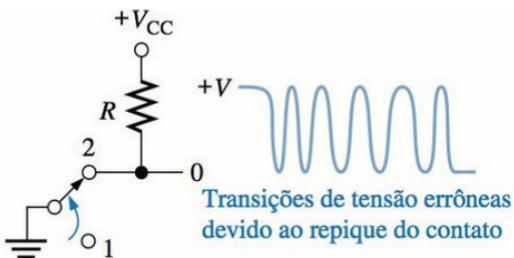
Descrição da situação-problema

Imagine que você trabalha analisando projetos em uma equipe de desenvolvimento. Você recebeu para análise um projeto de um controle remoto que não está funcionando adequadamente.

Em uma primeira análise, você isolou a parte do circuito que está com defeito: trata-se de um *latch* $\bar{S}\bar{R}$ utilizado para eliminar oscilações devido ao contato de uma chave mecânica do controle.

Segundo Floyd (2007), quando a chave é acionada fechando um contato, esse contato vibra fisicamente, oscilando por algum tempo antes de estabilizar. Embora isso ocorra em um intervalo relativamente curto, as oscilações produzem picos de tensão em geral não aceitáveis em circuitos digitais. Essa situação é ilustrada na Figura 3.17.

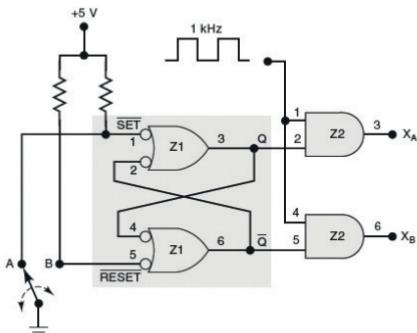
Figura 3.17 | Situação de debounce ou repique



Fonte: Floyd (2007, p. 391).

Um *latch* feito com portas NAND (ou OR negativa) pode ser usado para eliminar os efeitos do repique de uma chave, como mostra a Figura 3.18.

Figura 3.18 | Circuito para eliminar o repique



Fonte: Tocci; Widner; Moss (2011, p. 183).

Ao testar o circuito, você percebe que ele funciona corretamente quando a chave está na posição B, porém, quando a chave é colocada na posição A, a saída Q não vai para o estado 1, conforme era esperado. Quais são as possíveis causas para esse mau funcionamento?

Resolução da situação-problema

As saídas do *latch* mostrado na Figura 3.18 controlam a passagem de um sinal formado por pulsos retangulares com frequência de 1 kHz por meio das saídas X_A e X_B das portas AND. Quando a chave é colocada na posição A, o latch é “setado” ($Q=1$). Isso habilita os pulsos de 1 kHz a chegarem à saída X_A , enquanto o nível BAIXO em Q mantém $X_B=0$. Quando a chave é colocada na posição B, o latch é reiniciado ($Q=0$), mantendo $X_A=0$, enquanto o nível ALTO em Q habilita a passagem dos pulsos para X_B .

Você percebeu que, quando a chave está na posição B, o circuito funciona corretamente, porém, com a chave na posição A, a saída Q não vai para o estado 1, conforme era esperado. Existem algumas possibilidades de problema.

1. Circuito aberto internamente em Z1-1, que impediria que a saída Q respondesse à entrada \overline{SET} .
2. Falha em um componente interno da porta NAND Z1, que impediria que ela funcionasse corretamente.
3. A saída Q está fixa em nível BAIXO. Isso pode ser causado por:
 - a. Z1-3 em curto interno com GND.

- b. Z1-4 em curto interno com GND.
- c. Z2-2 em curto interno com GND.
- d. A saída Q em curto externo com GND.

Uma verificação com ohmímetro, medindo-se entre a saída Q e GND, determinará a existência de qualquer uma dessas condições apresentadas; e uma verificação visual pode revelar algum curto externo.

Um curto interno ou externo entre \bar{Q} e V_{cc} é descartado, pois se \bar{Q} estivesse em curto com V_{cc} , isso não evitaria que a saída Q fosse levada para nível ALTO, quando a entrada SET fosse para nível BAIXO. Como a saída Q não está indo para nível ALTO, esse não é o defeito. O motivo que faz com que a saída \bar{Q} esteja fixa em nível ALTO é a saída Q estar fixa em nível BAIXO, mantendo a saída \bar{Q} em nível ALTO por meio da porta NAND, na parte inferior do diagrama.

Faça valer a pena

1. Em um circuito combinacional, o valor lógico presente na saída é determinado unicamente pelos valores lógicos presentes nas entradas em cada momento. O comportamento de um circuito sequencial, por sua vez, não depende somente das entradas em um determinado instante; o estado atual do circuito também afeta a sua saída.

Nesse contexto, avalie as afirmações a seguir:

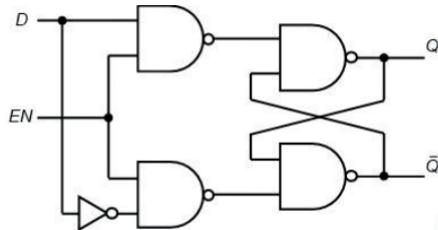
- I. Dentre os circuitos sequenciais, os dispositivos biestáveis são os mais importantes e têm esse nome por apresentarem dois estados estáveis, chamados de SET e RESET.
- II. Os dispositivos biestáveis, por terem a capacidade de armazenar um bit de informação, são bastante úteis como dispositivos de memória.
- III. Circuitos sequenciais biestáveis dividem-se entre *latches* e *flip-flops*, e a diferença entre esses dispositivos dá-se pela maneira como cada um deles comuta entre seus estados.
- IV. Enquanto em um *flip-flop* sua saída pode ser alterada a qualquer instante de tempo, por isso eles são considerados dispositivos assíncronos, em um *latch* a saída somente é alterada quando houver variação no sinal de *clock*, por isso os *latches* são considerados síncronos.

É correto o que se afirma em:

- a) I e III, apenas.
- b) II e IV, apenas.
- c) I, III e IV, apenas.
- d) I, II e III, apenas.
- e) I, II, III e IV.

2. Em um *latch* controlado, a transição entre os estados ocorre somente sob a ação da subida de um pulso em uma entrada de habilitação, EN. O *latch* D é um tipo de *latch* controlado. Ele teve sua origem devido à necessidade de se evitar, no *latch* S-R, a ocorrência do estado proibido. Por isso, o *latch* D tem apenas uma entrada além da EN. O diagrama lógico para um *latch* D controlado é mostrado na Figura 3.19:

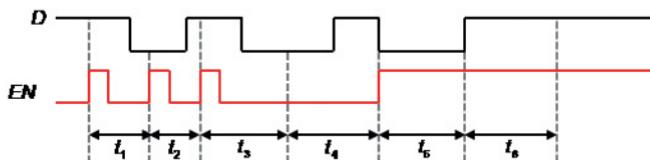
Figura 3.19 | Diagrama lógico para o *latch* D controlado



Fonte: elaborada pelo autor.

Considere que as formas de onda de D e EN, mostradas na Figura 3.20, são aplicadas nas entradas do *latch* visto na Figura 3.19. Então determine os valores da saída Q nos instantes t_1, t_2, t_3, t_4, t_5 e t_6 e assinale a alternativa que contém a sequência correta.

Figura 3.20 | Formas de onda de entrada

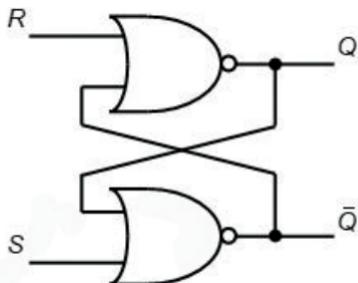


Fonte: elaborada pelo autor.

- a) 101101.
- b) 010010.
- c) 101010.
- d) 001101.
- e) 100001.

3. Segundo Floyd (2007), um *latch* é um tipo de dispositivo lógico de armazenamento temporário que tem dois estados estáveis e por isso é chamado de **biestável** ou **multivibrador**. Um *latch S-R* (SET-RESET) com entrada ativa em nível ALTO é composto de duas portas NOR com acoplamento cruzado, conforme pode ser visto na Figura 3.21.

Figura 3.21 | *Latch S-R* com entrada ativa em nível ALTO



Fonte: elaborada pelo autor.

Nesse contexto, considere as afirmações a seguir:

I. A tabela de excitação, por sua vez, lista os valores das entradas do circuito sequencial, bem como as mudanças entre os estados presentes e seguintes. Para o *latch S-R*, a excitação é mostrada na Tabela 3.8.

Tabela 3.8 | Tabela de excitação

S	R	$Q(t+1)$	Estado
0	0	$Q(t)$	Repouso
0	1	0	RESET
1	0	1	SET
1	1	?	Inválido

Fonte: elaborada pelo autor.

II. Considere que as entradas *S* e *R* do *latch* estão nos níveis lógicos 1 e 0, respectivamente. Em um segundo instante, a entrada *S* é levada para 0, e a partir desse momento observa-se na saída \bar{Q} o nível lógico 0.

III. Quando as duas entradas *S* e *R* são colocadas no nível lógico 1 ao mesmo tempo, tanto a saída *Q* quanto \bar{Q} são forçadas para 0, violando a condição básica de complementaridade entre essas duas saídas.

É correto o que se afirma em:

- a) I e II, apenas.
- b) II e III, apenas.
- c) I e III, apenas.
- d) I, apenas.
- e) I, II e III.

Seção 3.2

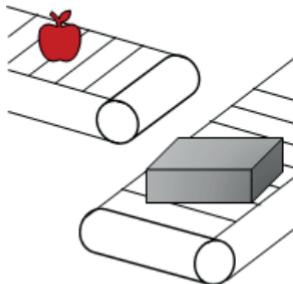
Contadores e registradores

Diálogo aberto

Os *flip-flops* podem ser utilizados para realizar operações de contagem. Esse arranjo com *flip-flops* recebe o nome de contador. A quantidade de *flip-flops* usados e a forma na qual eles são combinados determinam o número de estados e a sequência específica de estados que o contador percorre durante um ciclo completo. Os registradores de deslocamento são um tipo de circuito lógico muito parecido com os contadores digitais. Os registradores são geralmente usados para armazenamento de dados digitais e não possuem uma característica interna de sequência de estado como os contadores.

Sua start-up de projetos de sistemas de automação já executou um primeiro projeto para uma rede de mercados local, um sistema de botão de chamada de auxílio para o caixa. Esse projeto deu a vocês uma certa visibilidade, e um produtor de maçãs entrou em contato com a sua empresa pois ele deseja automatizar a contagem de maçãs empacotadas nas caixas para transporte. Atualmente ele trabalha com um par de esteiras transportadoras operadas manualmente, como pode ser visto no esquema da Figura 3.22. Um operador posiciona uma caixa logo abaixo da esteira que carrega as maçãs. Em seguida, ele ativa a esteira das maçãs e inicia a contagem das que caem na caixa.

Figura 3.22 | Empacotamento de maçãs com esteiras transportadoras



Fonte: elaborado pelo autor.

Seu cliente gostaria de automatizar essa etapa de contagem e açãoamento das esteiras, devendo cada caixa conter 48 maçãs. Para começar, pense em uma solução usando algum tipo de sensor de presença que seja capaz de identificar quando uma maçã passa pela esteira. Além disso, você já sabe que os *latches* e *flip-flops* são dispositivos capazes de armazenar bits, sendo assim dispositivos de memória, mas como usá-los para criar um circuito capaz de contar? Para auxiliá-lo com esse projeto, vamos aprender mais sobre o uso de *flip-flops* como contadores e registradores de deslocamento.

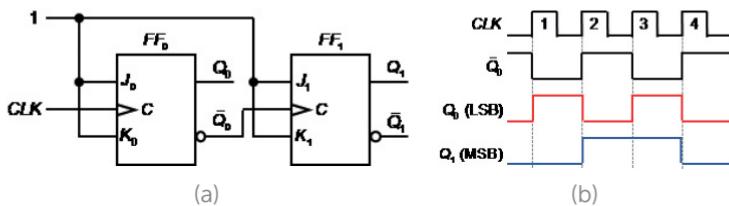
Não pode faltar

Os contadores podem ser classificados como **assíncronos** ou **síncronos**, de acordo com como eles recebem os pulsos de *clock*. Dentro dessas categorias, os contadores podem ainda ser classificados pela sequência com a qual percorrem os estados, pelo número de estados ou pelo número de *flip-flops* no contador.

Nos **contadores assíncronos**, o primeiro *flip-flop* recebe um sinal de *clock* externo, e cada *flip-flop* subsequente recebe o *clock* por meio da saída do *flip-flop* anterior.

A Figura 3.23 (a) mostra um contador binário de dois bits conectado para operação assíncrona. Os *flip-flops* são conectados para operação *toggle* ($J=1$ e $K=1$), e o *clock* (CLK) é aplicado na entrada de *clock* (C) apenas do primeiro *flip-flop* (FF_0), que é o bit menos significativo (LSB). O segundo *flip-flop* (FF_1) é disparado pela saída \bar{Q}_0 do FF_0 . Enquanto o FF_0 muda de estado na borda positiva de cada pulso de *clock*, o FF_1 muda apenas quando disparado por uma transição positiva da saída \bar{Q}_0 do FF_0 . Uma vez que há o tempo de atraso de propagação inherente dos *flip-flops*, a transição do pulso de *clock* (CLK) de entrada e a transição da saída \bar{Q}_0 do FF_0 nunca podem mudar exatamente ao mesmo tempo. Portanto, os dois *flip-flops* nunca são disparados simultaneamente (FLOYD, 2007). Por isso, a operação do contador é assíncrona.

Figura 3.23 | Contador binário assíncrono de dois bits: (a) esquema de montagem; (b) diagrama de temporização



Fonte: adaptada de Floyd (2007, p. 444).

A Figura 3.23 (b) ilustra as mudanças de estado nas saídas dos *flip-flops* em resposta aos pulsos de *clock*. Por questões de simplicidade, as transições de Q_0 e Q_1 e os pulsos do *clock* são mostrados como eventos simultâneos, mas sabemos que existe um pequeno atraso entre as transições do *clock* e Q_0 e entre as transições de Q_0 e Q_1 . A sequência dos estados do contador realiza uma sequência de números binários, conforme listada na Tabela 3.9.

Tabela 3.9 | Sequência de estados binários para o contador de dois bits assíncrono

CLK	Q_1	Q_0
Valor inicial	0	0
1	0	1
2	1	0
3	1	1
4 (volta ao estado inicial)	0	0

Fonte: elaborada pelo autor.

O contador da Figura 3.23 (a) conta, de fato, os pulsos de *clock* até três e, no quarto pulso, ele recicla seu estado para o estado inicial ($Q_0 = 0$, $Q_1 = 0$). O termo reciclagem, usualmente empregado na operação de contadores, refere-se à transição do contador do seu estado final para o original.



Faça você mesmo

Como era de se esperar, com dois *flip-flops* $2^2 = 4$, o contador de dois bits exibe quatro estados diferentes. É possível criar contadores capazes de contar qualquer quantia de potências de 2. Faça o

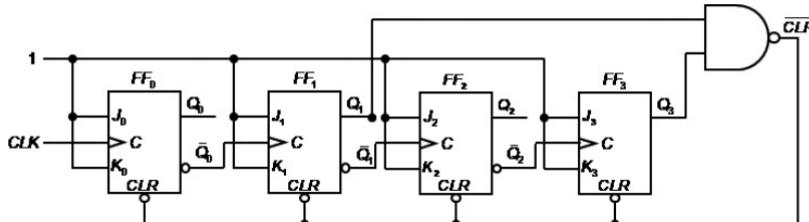
esquema de montagem, o diagrama de temporização e a tabela com a sequência de estados binários para um contador de três bits. Quantos estados esse contador possui?

Os contadores também podem ser projetados de modo a ter um número de estados menor em sua sequência que o valor máximo de 2^n . Esse tipo de sequência é chamado de sequência truncada. Para se obter uma sequência truncada, é preciso forçar que o contador recicle seu estado antes que ele complete um ciclo passando por todos os estados possíveis.

Um contador de dez estados, chamado contador de década, tem sua sequência de contagem de zero (0000_2) a nove (1001_2). Além disso, ele é um contador de década BCD, porque a sua sequência de dez estados produz o código BCD. Esse tipo de contador é útil em aplicações com display nas quais o código BCD é necessário para conversões com leituras em decimal.

Por exemplo, em um contador de década BCD, o estado é reciclado para o estado 0000 após o estado 1001. Um contador de década requer quatro *flip-flops* (três *flip-flops* são insuficientes, pois $2^3 = 8$). Uma forma de fazer um contador reciclar após a contagem do nove (1001) é decodificar a contagem dez (1010) com uma porta NAND e conectar a saída da porta NAND nas entradas de *clear* (\overline{CLR}) dos *flip-flops*, como mostra a Figura 3.24.

Figura 3.24 | Contador de década assíncrono

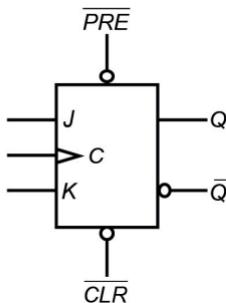


Fonte: adaptada de Floyd (2007, p. 449).

Conforme Floyd (2007, p. 402):

A maioria dos flip-flops em circuitos integrados também tem entradas assíncronas. Essas são entradas que afetam o estado do flip-flop independente do clock. Elas são normalmente denominadas preset (PRE) e clear (CLR) [...]. Um nível ativo na entrada preset irá [levar o flip-flop para o estado SET] e um nível ativo na entrada clear irá levar o flip-flop [para o estado RESET]. Um símbolo lógico para um flip-flop J-K com entradas preset e clear é mostrado na [Figura 3.25]. Essas entradas são ativas em nível BAIXO, conforme indicado pelos pequenos círculos. Essas entradas de preset e clear têm que ser mantidas em nível ALTO para a operação síncrona.

Figura 3.25 | Símbolo lógico para um flip-flop J-K com entradas *preset* e *clear*



Fonte: adaptada de Floyd (2007, p. 402).

Observe que apenas as saídas Q_1 e Q_3 são necessárias para reciclar a contagem. Esse tipo de arranjo é chamado de **decodificação parcial**, pois não foram usadas todas as saídas para reciclar a contagem. Quando o contador chega ao estado dez (1010), a saída da porta de decodificação vai para 0 e assincronamente reinicia todos os flip-flops.

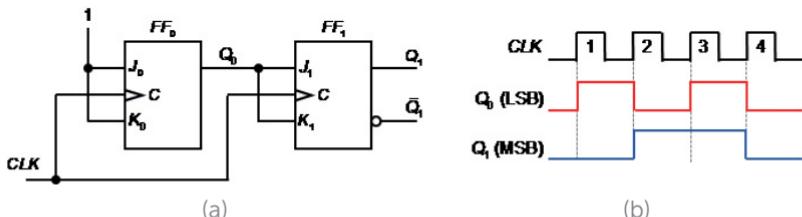
Nos **contadores síncronos**, o mesmo sinal de *clock* é fornecido simultaneamente para todos os flip-flops. Na Figura 3.26 (a), temos como exemplo um contador binário síncrono de dois bits. Note que foi preciso um arranjo diferente na ligação das entradas J_1 e K_1 para que o FF_1 consiga uma sequência binária.

Considere que os dois *flip-flops* estão reiniciados, de modo que o contador esteja inicialmente no estado binário 0. Quando a borda positiva do primeiro pulso de *clock* é aplicada, FF_0 comuta e Q_0 passa para o nível lógico 1. No FF_1 , as entradas J_1 e K_1 estão em nível baixo, porque Q_0 – saída na qual as entradas de FF_1 estão conectadas – ainda não foi para o nível lógico 1. Lembre-se de que existe um atraso de propagação entre a borda de disparo do *clock* até que a saída Q realmente comute. Assim, $J_1 = 0$ e $K_1 = 0$, quando é aplicada a borda de subida do primeiro pulso de *clock*. Essa é uma condição de repouso, e por isso FF_1 não muda de estado (FLOYD, 2007).

Quando a borda de subida do segundo pulso de *clock* ocorre, FF_0 comuta e Q_0 vai para o nível baixo. Como FF_1 tem um nível 1 nas entradas J_1 e K_1 na borda de disparo desse pulso de *clock*, o *flip-flop* comuta, e Q_1 vai para o nível lógico 1. Quando a borda de subida do terceiro pulso de *clock* ocorre, FF_0 comuta novamente para o estado *SET*, e FF_1 permanece no estado *SET*.

Por fim, na quarta borda de subida do *clock*, Q_0 e Q_1 vão para o nível lógico 0, porque os dois estão na condição *toggle* em suas entradas J e K . Assim, o contador recicla para o seu estado original. A Figura 3.26 (b) mostra o diagrama de temporização desse contador.

Figura 3.26 | Contador binário síncrono de dois bits: (a) esquema de montagem; (b) diagrama de temporização



Fonte: adaptada de Floyd (2007, p. 452-453).

Observe que os atrasos de propagação não são indicados na Figura 3.26 (b). Embora os atrasos sejam um fator importante na operação de um contador síncrono, eles normalmente são omitidos no diagrama de temporização geral por questão de simplicidade de representação. Entretanto, em circuitos digitais de alta velocidade, esses pequenos atrasos são considerações importantes no projeto e na análise de defeito (FLOYD, 2007).



Um contador binário de três bits tem a sua sequência de estados representada na Tabela 3.10. Vamos projetar um contador binário **síncrono** de três bits?

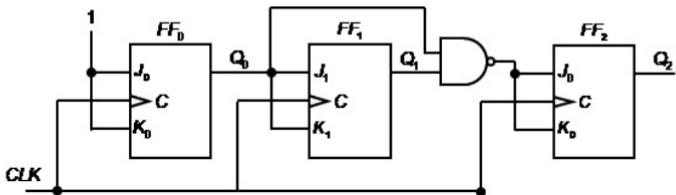
Tabela 3.10 | Sequência de estados binários para o contador de dois bits assíncrono

CLK	Q_2	Q_1	Q_0
Valor inicial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recicla)	0	0	0

Fonte: elaborada pelo autor.

Observe que Q_0 alterna os bits a cada pulso de *clock*. Para produzir essa operação, o FF_0 deve ser mantido no modo *toggle*. Note que Q_1 vai para o estado oposto cada vez que Q_0 for nível 1. Segundo Floyd (2007), para produzir essa operação, Q_0 é conectada diretamente nas entradas J_1 e K_1 do FF_1 . Por fim, no FF_2 , a saída Q_2 muda de estado somente quando Q_0 e Q_1 são nível lógico 1. Essa condição pode ser detectada por uma porta AND e aplicada nas entradas J_2 e K_2 do FF_2 . O esquema de montagem desse contador pode ser visto na Figura 3.27.

Figura 3.27 | Contador binário síncrono de três bits



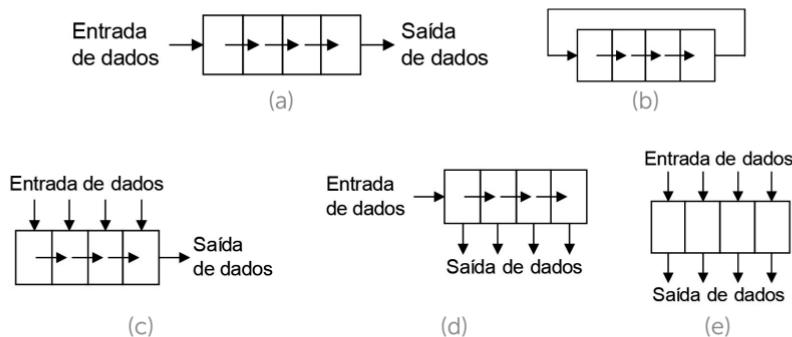
Fonte: adaptada de Floyd (2007, p. 454).

Você é capaz, agora, de projetar um contador binário síncrono de quatro bits? E um contador de década síncrono de quatro bits? Temos a certeza de que sim! Exercite seus conhecimentos projetando esses contadores.

Um registrador é um circuito digital com duas funções básicas: **armazenar** e **movimentar dados**.

Segundo Floyd (2007), a **capacidade de armazenamento** de um registrador é o número total de bits de dados digitais que ele pode reter. A capacidade de armazenamento de um registrador o torna um importante tipo de dispositivo de memória. Ela pode ser expandida aumentando-se o número de *flip-flops* no registrador – e cada *flip-flop* equivale a um bit de capacidade a mais. E é a **capacidade de deslocamento** de um registrador que permite o movimento de dados de um *flip-flop* para outro com a aplicação de pulsos de *clock*. Na Figura 3.28, temos exemplos dos tipos de movimentos de dados em registradores de deslocamento.

Figura 3.28 | Movimentos básicos de dados em registradores de deslocamento: (a) entrada serial/saída serial; (b) rotação (buffer circular); (c) entrada paralela/saída serial; (d) entrada serial/saída paralela; (e) entrada paralela/saída paralela



Fonte: elaborada pelo autor.



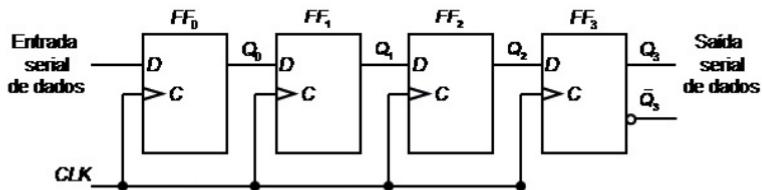
Refletia

Em geral, qual é a diferença entre um contador e um registrador de deslocamento? Qual *flip-flop* seria mais adequado para ser utilizado em um registrador?

O registrador de deslocamento com **entrada serial/saída** serial recebe um bit de cada vez numa única linha, ou seja, de forma serial. Ele disponibiliza na sua saída a informação armazenada, também um bit de cada vez, de forma serial. A Figura 3.29 mostra um dispositivo de quatro bits implementado com *flip-flops D*. Com

quatro estágios é possível armazenar até quatro bits de dados nesse registrador de deslocamento.

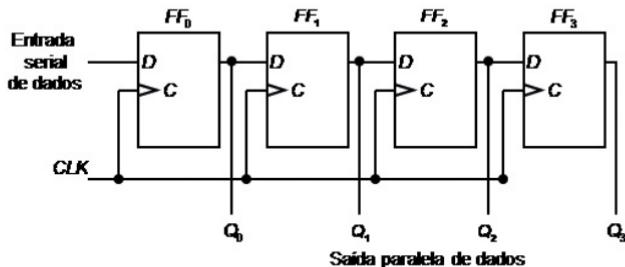
Figura 3.29 | Registrador com entrada serial/saída serial de quatro bits



Fonte: adaptada de Floyd (2007, p. 511).

Em um registrador de deslocamento com **entrada serial/saída paralela**, os bits de dados são inseridos de forma serial, porém, uma vez que os dados são armazenados, cada bit é disponibilizado simultaneamente na saída de forma paralela. A Figura 3.30 mostra um registrador de deslocamento com entrada serial/saída paralela de quatro bits.

Figura 3.30 | Registrador com entrada serial/saída paralela de quatro bits



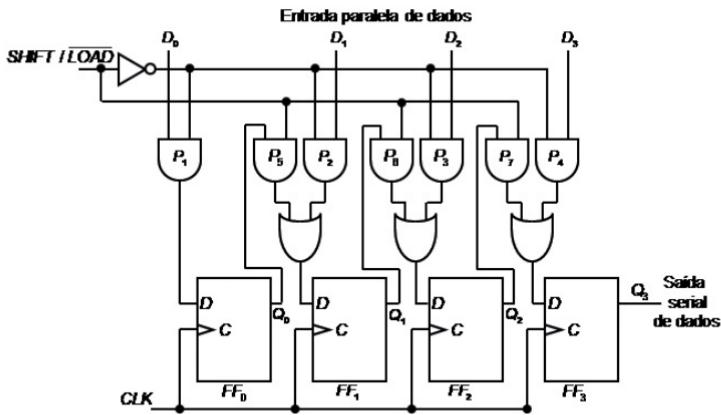
Fonte: adaptada de Floyd (2007, p. 515).

Para um registrador de deslocamento com **entrada paralela/saída serial**, os bits são inseridos todos ao mesmo tempo nos seus respectivos flip-flops, em vez de bit a bit em uma única linha, como acontece com a entrada serial de dados. A saída se dá bit a bit. A Figura 3.31 mostra um registrador de deslocamento com entrada paralela/saída serial de quatro bits. Há, além das quatro linhas de entradas de dados (D_0, D_1, D_2 e D_3), uma entrada **SHIFT / LOAD**, que permite carregar os quatro bits em paralelo no registrador.

As portas de P_1 a P_4 são habilitadas quando **SHIFT / LOAD** é 0, permitindo que os bits de dados sejam aplicados na entrada D do

seu respectivo *flip-flop*. Quando um pulso de *clock* é aplicado, os estados desses *flip-flops* são atualizados adequadamente, de modo que todos os quatro bits são armazenados simultaneamente. Quando ***SHIFT / LOAD*** for igual a 1, as portas de P_1 a P_4 são desabilitadas e as portas de P_5 a P_7 são habilitadas, deslocando os bits de dados de um *flip-flop* para o outro.

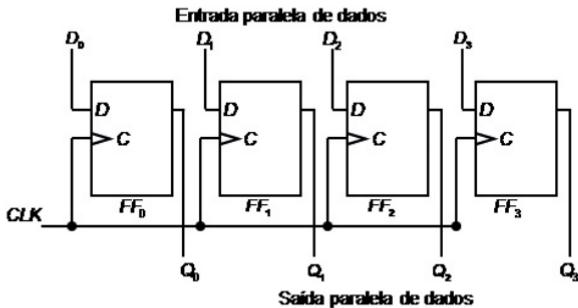
Figura 3.31 | Registrador com entrada paralela/saída serial de quatro bits



Fonte: adaptada de Floyd (2007, p. 518).

O registrador de deslocamento com entrada **paralela/saída paralela** possui o aspecto construtivo, e o seu entendimento é muito mais simples. No instante seguinte à entrada de todos os bits de dados, eles são disponibilizados nas saídas. A Figura 3.32 mostra um registrador com entrada paralela/saída paralela de quatro bits.

Figura 3.32 | Registrador com entrada paralela/saída paralela de quatro bits



Fonte: adaptada de Floyd (2007, p. 518).



Pesquise mais

Os registradores de deslocamento são importantes em aplicações que envolvem o armazenamento e a transferência de dados em sistemas digitais. Nesta seção, apenas introduzimos o assunto. Para saber mais sobre eles, leia o Capítulo 9 do livro:

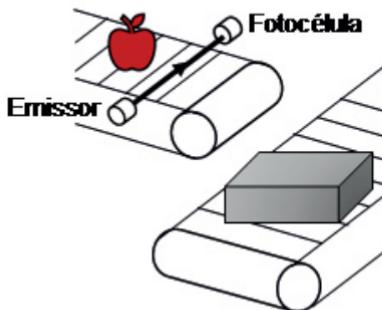
FLOYD, Thomas L. **Sistemas digitais**: fundamentos e aplicações. 9. ed. Porto Alegre: Bookman, 2007.

Sem medo de errar

Após a implementação do projeto sistema de botão de chamada de auxílio para o caixa em uma rede de mercados locais, um produtor de maçãs, parceiro dessa rede, tomou conhecimento da sua start-up e os contratou para automatizar a contagem de maçãs no encaixotamento para transporte. Atualmente um operador posiciona uma caixa logo abaixo da esteira que carrega as maçãs; em seguida ele ativa a esteira das maçãs e inicia a contagem das que caem na caixa, como foi visto na Figura 3.22. Cada caixa deve conter 48 maçãs.

Inicialmente é preciso pensar em como detectar a passagem das maçãs pela esteira. Uma das maneiras de fazer isso é posicionar um emissor de feixe de luz e uma fotocélula em um determinado ponto de uma esteira de transporte, como pode ser visto na Figura 3.33. Assim, quando uma maçã passa por esse ponto, ele interrompe o feixe, e essa interrupção é convertida em um sinal lógico que aciona um contador.

Figura 3.33 | Esteira de transporte com sensor de presença



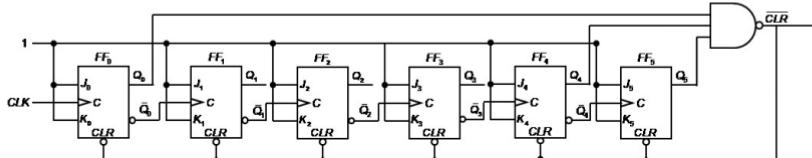
Fonte: elaborada pelo autor.

Agora, é necessário projetar um circuito contador capaz de contar até 48, para contar a quantidade de maçãs que passa pela esteira. O primeiro passo é identificar quantos *flip-flops* são necessários para realizar essa contagem.

Para determinar a quantidade N de *flip-flops* necessária para contar até 48, temos que $2^N \geq 48$. Como $2^5 = 32$, 5 *flip-flops* são insuficientes, mas $2^6 = 64$, de modo que com 6 *flip-flops* seria possível contar até $110000_2 = 48_{10}$. Portanto, 6 é a menor quantidade de *flip-flops* possível para montar um contador que conte até 48. Claro que seria possível utilizar mais do que 6 *flip-flops*, mas seria um desperdício de recurso.

No entanto, um contador com 6 *flip-flops* conta até 64, por isso é necessário que o seu circuito retorne ao estado inicial (000000_2) quando chegar à contagem de 49 (110001_2). Como vimos, uma forma de fazer um contador retornar ao estado inicial é decodificar a contagem final, nesse caso (110001_2) com uma porta NAND, e conectar a saída da porta NAND nas entradas de clear (CLR) dos *flip-flops*, como pode ser visto na Figura 3.34.

Figura 3.34 | Circuito contador de maçãs



Fonte: elaborada pelo autor.

Além disso, a mesma saída ligada às entradas de *clear* pode ser usada para acionar a esteira das caixas, de modo a mover uma nova caixa vazia para receber novas maçãs.

Antes de finalizar, vamos pensar em alguns aspectos: quais circuitos integrados comerciais poderiam ser utilizados para montar esse circuito? Faça uma pesquisa e responda a essa pergunta. Além disso, você pode imaginar que o contador é um circuito que deve ser utilizado em diversas aplicações, então será que existe um contador encapsulado em um único CI? Se existe, será que existe um que conte até 48? Se não, existe uma maneira de fazer com que um eventual CI de um contador conte até 48? Continue suas buscas, pois você está apenas começando!

Avançando na prática

Controle de vagas de um estacionamento

Descrição da situação-problema

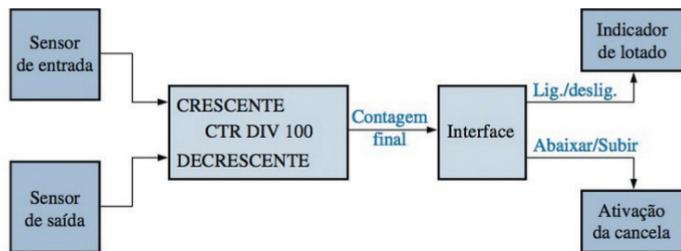
Vamos pensar no seguinte cenário: você foi contratado por um pequeno shopping local para desenvolver um sistema de controle de estacionamento de veículos. O problema consiste em fazer um projeto de monitoração dos espaços disponíveis em uma centena de vagas em um estacionamento e prover a sinalização da condição de lotado por meio de uma indicação luminosa e abaixando a cancela na entrada. Proponha um sistema, com base em circuitos integrados (CI) da série 7400 disponíveis no mercado.

Resolução da situação-problema

Para resolver esse problema, você propôs um sistema que consiste em:

1. Dois sensores de presença, um na entrada e outro na saída do estacionamento.
2. Um contador crescente/decrescente.
3. Um circuito lógico para ligar ou desligar o sinal de LOTADO e baixar ou subir a cancela na entrada.

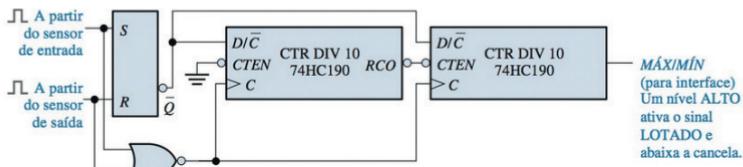
Figura 3.35 | Diagrama de blocos funcionais para o controle de vagas do estacionamento



Fonte: Floyd (2007, p. 483).

O contador crescente/decrescente é obtido ao ligar dois contadores de década 74HC190 (TEXAS, 2003) conectados em cascata, conforme pode ser visto no diagrama lógico na Figura 3.36:

Figura 3.36 | Diagrama lógico para um contador crescente/decrecente de módulo 100 para o controle de estacionamento de veículos



Fonte: Floyd (2007, p. 483).

Cada veículo que entra no estacionamento aciona o sensor de entrada e leva o *latch S-R* para o estado *SET*, de modo que $\bar{Q}=0$, levando o contador a funcionar no modo *crescente*. O sinal do sensor também é levado para uma porta NOR que aciona os contadores pela entrada do *clock*, quando a saída da porta muda de 0 para 1. Cada vez que isso ocorre, o contador incrementa uma unidade.

Quando cem veículos entrarem no estacionamento, a saída MÁX/MÍN do contador vai para 1, ativando um circuito que acende o sinal de lotado e abaixa a cancela.

Quando um veículo sai, o sensor de saída é acionado, levando o *latch S-R* para o estado *RESET*. Agora $\bar{Q}=1$ leva o contador para o modo *decrecente*. O sinal desse sensor também é levado para uma porta NOR, que aciona os contadores pela entrada do *clock* quando a saída da porta muda de 0 para 1. Cada vez que isso ocorre, o contador decrementa uma unidade.

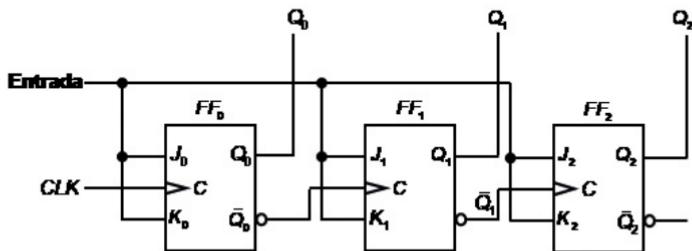
Se o estacionamento estiver lotado e um veículo sair, a saída MÁX/MÍN vai para nível BAIXO, desligando o sinal *lotado* e subindo a cancela.

Faça valer a pena

1. Os *flip-flops* podem ser utilizados para realizar operações de contagem. Esse arranjo com *flip-flops* recebe o nome de contador. A quantidade de *flip-flops* usados e a forma na qual eles são combinados determinam o número de estados e a sequência específica de estados que o contador percorre durante um ciclo completo.

Os contadores podem ser classificados como **assíncronos** ou **síncronos**, de acordo com como eles recebem os pulsos de *clock*. Nesse contexto, considere o circuito sequencial da Figura 3.377 e analise as afirmações a seguir:

Figura 3.37 | Circuito sequencial



Fonte: elaborada pelo autor.

I. O circuito da Figura 3.377 é um contador assíncrono de 3 bits, capaz de contar de 0 até 7.

II. Para que o circuito da Figura 3.377 funcione como um contador, a **entrada** do circuito deve ser mantida em nível lógico 1, assim, com as entradas J e K dos *flip-flops* em 1, eles funcionarão no modo de operação *toggle* ou comutação. Nesse estado, a cada transição sucessiva do *clock*, o *flip-flop* muda para o estado oposto ao anterior.

III. Os contadores também podem ser vistos como circuitos divisores de frequência, uma vez que a frequência da saída de cada *flip-flop* é a metade da frequência do sinal aplicado ao seu *clock*.

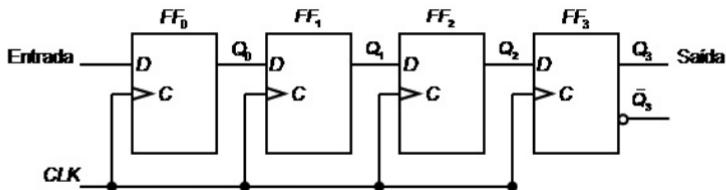
É correto o que se afirma em:

- a) I, apenas.
- b) II e III, apenas.
- c) I e III, apenas.
- d) I e II, apenas.
- e) I, II e III.

2. Um registrador é um circuito digital com duas funções básicas: *armazenar* e *movimentar dados*. A capacidade de armazenamento de um registrador é o número total de bits de dados digitais que ele pode reter. Já a *capacidade de deslocamento* de um registrador permite o movimento de dados de um *flip-flop* para outro com a aplicação de pulsos de *clock*.

Seja o circuito da Figura 3.38, considere que ele esteja inicialmente zerado (todos os bits em nível 0).

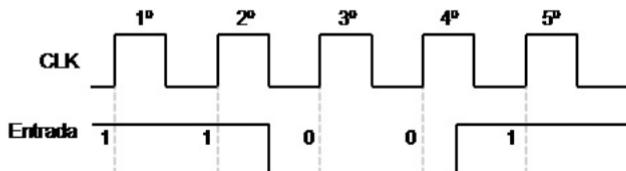
Figura 3.38 | Circuito sequencial



Fonte: elaborada pelo autor.

Agora considere, também, a sequência de dados de entrada da Figura 3.39.

Figura 3.39 | Sequência de dados de entrada



Fonte: elaborada pelo autor.

Nesse contexto, analise as afirmações a seguir:

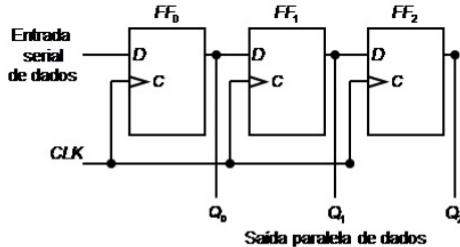
- I. O circuito da Figura 3.38 é um registrador de serial/serial de 4 bits.
- II. Após o 3º pulso de *clock*, a sequência de dados armazenada no registrador é $Q_3Q_2Q_1Q_0 = 0100$.
- III. Após o 5º pulso de *clock*, o bit armazenado no *flip-flop* FF_0 no primeiro pulso de *clock* é eliminado na saída do *flip-flop* FF_3 .

É correto o que se afirma em:

- a) I e III, apenas.
- b) I e II, apenas.
- c) II e III, apenas.
- d) I, apenas.
- e) I, II e III.

3. Em um registrador de deslocamento com entrada serial/saída paralela, os bits de dados são inseridos de forma serial, porém, uma vez que os dados são armazenados, cada bit é disponibilizado simultaneamente na saída de forma paralela, como pode ser visto na Figura 3.40.

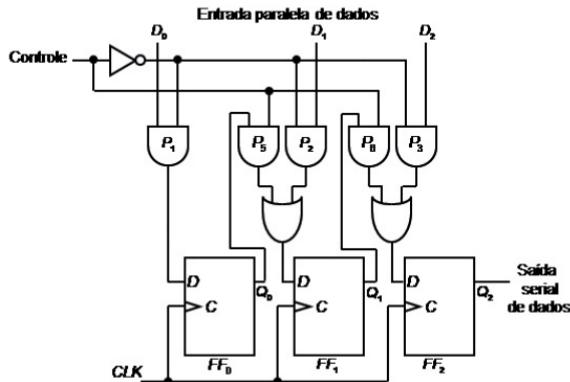
Figura 3.40 | Registrador com entrada serial/saída paralela



Fonte: elaborada pelo autor.

Em um registrador de deslocamento com entrada paralela/saída serial, no entanto, os bits são inseridos todos ao mesmo tempo nos seus respectivos *flip-flops*, em vez de bit a bit em uma única linha, como acontece com a entrada serial de dados. A saída se dá bit a bit, como pode ser visto na

Figura 3.41 | Registrador com entrada serial/saída paralela

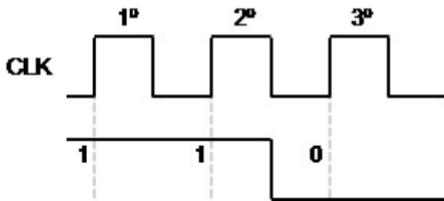


Fonte: elaborada pelo autor.

Nesse contexto, avalie as seguintes afirmações:

- Ambos os registradores possuem capacidade de armazenamento de 3 bits.
- Considere a sequência de entrada da Figura 3.42 para o registrador com entrada serial/saída paralela (inicialmente zerado) da Figura 3.40.

Figura 3.42 | Sequência de entrada



Fonte: elaborada pelo autor.

A saída observada ao final de cada pulso de clock é: 1º pulso: $Q_2Q_1Q_0 = 001$; 2º pulso $Q_2Q_1Q_0 = 011$; e 3º pulso $Q_2Q_1Q_0 = 110$.

III. A entrada “Controle” no circuito da Figura 3.41 é responsável por controlar o armazenamento ou o deslocamento dos dados nesse circuito.

IV. Quando “Controle” é igual a 1, os bits de dados são deslocados da esquerda para a direita; quando “Controle” é igual a 0, os bits nas entradas D_0 , D_1 e D_3 são armazenados nos *flip-flops*.

É correto o que se afirma em:

- a) I, II e IV, apenas.
- b) II, III e IV, apenas.
- c) I e III, apenas.
- d) II e III, apenas.
- e) I, II, III e IV.

Seção 3.3

Máquinas de estados finitos

Diálogo aberto

Os sistemas sequenciais podem ser utilizados em projetos para diversas finalidades. Eles são caracterizados por circuitos que executam sequências de estados predefinidos de acordo com os pulsos de *clock* e com os outros sinais de entrada. Os contadores já estudados são máquinas de estado destinados a executar sequências numéricas regulares.

Sua *start-up* de projetos de sistemas de automação já conquistou espaço no mercado, e vocês receberam uma solicitação de projeto de um cliente importante, uma empresa de engenharia e eletrônica, cujos principais produtos são componentes automotivos.

Eles estão desenvolvendo uma nova injeção eletrônica para lançar no mercado. A injeção eletrônica é um sistema de gerenciamento eletrônico de alimentação de combustível para o motor a combustão de um veículo automotor. Um dos atuadores de marcha-lenta da injeção eletrônica é o motor de passo. O motor de passo (Figura 3.43) é basicamente um motor de corrente contínua de duas bobinas, com a particularidade de que, com pulsos de tensão aplicados sequencialmente em cada uma delas, o eixo gira um ângulo predeterminado. Para que o motor realize rotações com um número determinado de graus especificados pelo seu cliente, os eletroímãs em torno do motor de passo devem ser energizados segundo uma sequência específica de modo a atrair o motor magnetizado para a posição seguinte da rotação.

Figura 3.43 | Motor de passo para injeção eletrônica



Fonte: <<https://www.mte-thomson.com.br/produtosmte/motor-de-passo-iac/>>. Acesso em: 21 fev. 2018.

A sequência em questão pode ser representada como uma sequência de 4 bits: 0001, 0011, 1100 e 1000. Depois de 1000, o circuito deve retornar a 0001 e repetir a sequência. Sua tarefa é projetar um circuito com quatro saídas que realize essa sequência. Para auxiliá-lo com esse projeto, vamos aprender mais sobre o uso e a implementação das máquinas de estado finito.

Preparado?

Bons estudos.

Não pode faltar

Os sistemas sequenciais também podem ser projetados para aplicações específicas. Nesses casos, geralmente são chamados de *máquinas de estados finitos*. Os contadores são casos particulares de máquinas de estado. A máquina de estados finitos, de forma geral, é um conceito abstrato que, por meio de um método estruturado, auxilia a análise e a síntese de sistemas sequenciais sincronizados. Representa um sistema digital que, quando acionado por um pulso de um *clock*, move-se de um estado lógico para outro.

As máquinas de estado finito têm um nome que pode causar alguma confusão. O termo “finito” está presente para diferenciar as máquinas de estado finito de uma representação similar usada na matemática, que pode ter um número infinito de estados. O termo “máquina” é usado no sentido matemático, ou de ciência da computação, sendo um objeto conceitual que pode executar uma linguagem abstrata. Nesse caso, esse sentido de máquina não é o de hardware.

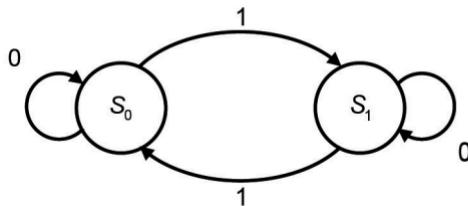
Em tese, todos os circuitos síncronos, incluindo os registradores e contadores movidos por um *clock*, podem ser representados por máquinas de estados finitos. É sempre recomendável solucionar um problema em um nível funcional mais alto, valendo-se da metodologia das máquinas de estados finitos. O projeto de um circuito sequencial pode ser dividido em cinco etapas:

1. Desenhar o diagrama de estados do circuito.

Um **diagrama de estado** representa um modelo gráfico de um sistema sequencial e especifica as transições entre os estados do sistema, as quais, por sua vez, descrevem como um estado particular passa para outro. Um sistema sequencial pode convenientemente

ser representado por um diagrama de estado. Um diagrama é traçado na forma de um conjunto de círculos, em que cada um deles corresponde a um estado. A Figura 3.44 traz como exemplo um diagrama de estado genérico para um sistema com uma única entrada e dois estados possíveis, S_0 e S_1 . As setas indicam as transições entre os estados para cada ação (entrada) em um determinado estado.

Figura 3.44 | Diagrama de estado genérico



Fonte: elaborada pelo autor.

2. Identificação dos estados e cálculo da quantidade de *flip-flops* necessários, dada por:

$$n_{ff} = \text{int}(\log_2 n_s), \quad (3.1)$$

em que n_{ff} é o número de *flip-flops* e n_s é o número de estados. A função `int()` indica o número inteiro menor ou igual à variável.

3. Obter as expressões lógicas, com o uso de tabelas-verdade e/ou mapas de Karnaugh, das variáveis de entrada dos *flip-flops* em função do estado atual e das variáveis de entrada.

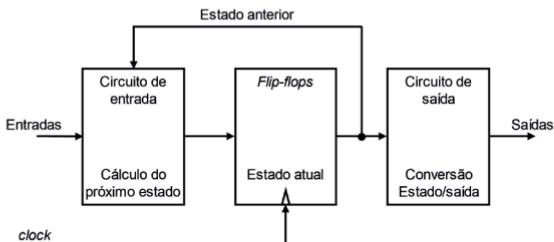
4. Obter as expressões lógicas, com o uso de tabelas-verdade e/ou mapas de Karnaugh, das saídas dos *flip-flops* em função do estado atual.

5. Desenhar o circuito.

Antes de praticarmos, é importante sabermos que existem dois modelos tradicionais para a implementação de máquinas de estados. São eles: as **máquinas Moore** e as **máquinas Mealy**.

Nas **máquinas Moore**, as saídas dependem **apenas** do estado atual registrado internamente no sistema. As mudanças das saídas ocorrem apenas na próxima transição de borda do *clock*. A Figura 3.45 apresenta um diagrama de blocos de uma máquina Moore.

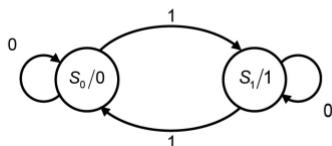
Figura 3.45 | Diagrama de blocos de uma máquina Moore



Fonte: elaborada pelo autor.

É possível notar, pela Figura 3.45, que o *clock* comanda a memória que envia para o circuito de saída, o estado atual, que é resultado da combinação entre as entradas e o estado anterior. Assim, as saídas assumem um novo estado somente na próxima transição de borda do *clock*. A Figura 3.46 traz um diagrama de estado genérico de uma máquina Moore para um sistema com uma entrada, uma saída e dois estados possíveis. Note que a saída é indicada em cada estado.

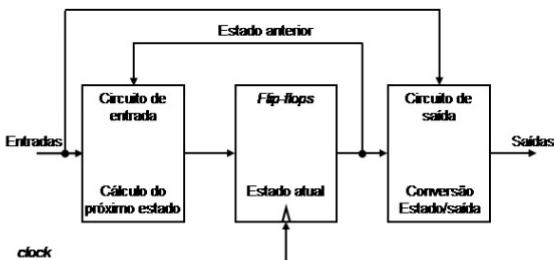
Figura 3.46 | Diagrama de blocos de uma máquina Moore



Fonte: elaborada pelo autor.

Nas **máquinas Mealy**, as saídas dependem tanto das entradas quanto do estado atual registrado internamente no sistema. Se houver mudanças nas entradas, as saídas também se alterarão. A Figura 3.47 apresenta em blocos a estrutura de um sistema sequencial que utiliza o modelo de Mealy.

Figura 3.47 | Diagrama de blocos de uma máquina Mealy



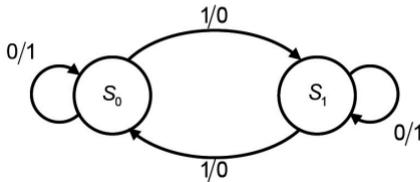
Fonte: elaborada pelo autor.



Nas **máquinas Moore**, as saídas dependem **apenas** do estado atual registrado internamente no sistema. Já nas **máquinas Mealy**, as saídas dependem tanto das entradas quanto do estado atual registrado internamente no sistema.

Agora, podemos notar que o *clock* comanda o registrador que envia os estados armazenados resultantes das entradas e o estado anteriormente existente para estágio de saída, porém as saídas também recebem as informações diretamente das entradas. Nesse caso, as saídas podem ser alteradas pela mudança dos estados de entradas, mesmo antes da atuação do *clock*. A Figura 3.48 traz um diagrama de estado genérico de uma máquina Mealy para um sistema com uma entrada, uma saída e dois estados possíveis. Note que agora a saída é indicada em cada transição de estado.

Figura 3.48 | Diagrama de estado genérico para uma máquina Mealy



Fonte: elaborada pelo autor.

As máquinas de estado também podem ser classificadas como sistemas síncronos ou assíncronos, conforme as características de aplicação do sistema de *clock* nos circuitos. Como já vimos, os sistemas sequenciais síncronos são aqueles em que os circuitos operam de maneira sincronizada com a entrada *clock*, ou seja, todos os elementos, *flip-flops* ou registradores de estado são comandados pelo mesmo pulso de *clock* em função do acionamento no tempo, uma vez que têm ligação dessa entrada em comum. Já os sistemas sequenciais assíncronos são aqueles que atuam de maneira assíncrona, ou seja, sem sincronismo entre os elementos do circuito, sendo que as variações internas e de saída ocorrem em instantes diferenciados com relação ao pulso de *clock*. Notamos pelas estruturas estudadas que a atuação do modelo de Moore é síncrona e que, no modelo de Mealy, as saídas podem atuar assincronamente (CAPUANO, 2014).



Você é capaz de pensar em aplicações em que é mais conveniente utilizar um dos modelos de máquina de estado finito apresentado (Moore ou Mealy) a despeito do outro?

Para efetuarmos projetos de sistemas sequenciais, de um modo geral, devemos seguir o procedimento descrito, ou seja, determinar o diagrama de estados; levantar a tabela-verdade da sequência de estados proposta; colocar na tabela os estados a serem assumidos nas entradas dos *flip-flops*, utilizando a tabela de projetos já desenvolvida; utilizar os mapas de Karnaugh para simplificá-los e assim elaborar o circuito com as expressões simplificadas. Para entender melhor o procedimento, vamos elaborar um projeto no exemplo a seguir.



Exemplificando

Vamos projetar uma máquina de estados para atuar como contador de três bits para efetuar a contagem crescente ($E = 0 \rightarrow 0$ a 7) ou decrescente ($E = 1 \rightarrow 7$ a 0). O circuito deverá possuir também uma saída S , que sinaliza se a contagem é crescente ($S = 1$) ou decrescente ($S = 0$). Para isso, use *flip-flops* do tipo J-K. A Tabela 3.11 traz a sequência de contagem proposta e a Tabela 3.12 traz a tabela-verdade do *flip-flop* J-K.

Tabela 3.11 | Sequência de contagem proposta

E	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	1	1	0
1	1	1	1
1	1	0	0
1	1	0	1
1	0	1	0
1	0	1	1
1	0	0	0
1	0	0	1

Fonte: elaborada pelo autor.

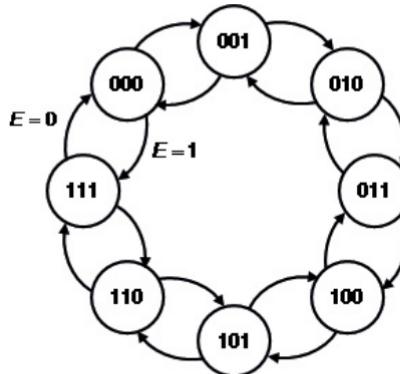
Tabela 3.12 | Tabela-verdade do *flip-flop* J-K

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Fonte: elaborada pelo autor.

Passo 1: a Figura 3.49 mostra o diagrama de estados para o problema proposto. Note que, para evitar que o diagrama ficasse visualmente poluído, as entradas $E=0$ e $E=1$ foram representadas apenas nas transições entre os estados 000 e 111, mas elas se repetem de forma idêntica nas transições dos demais estados.

Figura 3.49 | Diagrama de estados



Fonte: elaborada pelo autor.

Passo 2: a partir da máquina de estados, sabemos que existem oito estados e que serão necessários três flip-flops.

Passo 3: a Tabela 3.13 reproduz a máquina de estados, apresentando os estados e sua tradução para a saída dos flip-flops J-K.

Tabela 3.13 | Tabela-verdade completa do projeto

E	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0	S
0	0	0	0	0	X	0	X	1	X	1
0	0	0	1	0	X	1	X	X	1	1
0	0	1	0	0	X	X	0	1	X	1
0	0	1	1	1	X	X	1	X	1	1
0	1	0	0	X	0	0	X	1	X	1
0	1	0	1	X	0	1	X	X	1	1
0	1	1	0	X	0	X	0	1	X	1
0	1	1	1	X	1	X	1	X	1	1
1	1	1	1	X	0	X	0	X	1	0
1	1	1	0	X	0	X	1	1	X	0
1	1	0	1	X	0	0	X	X	1	0
1	1	0	0	X	1	1	X	1	X	0
1	0	1	1	0	X	X	0	X	1	0
1	0	1	0	0	X	X	1	1	X	0
1	0	0	1	0	X	0	X	X	1	0
1	0	0	0	1	X	1	X	1	X	0

Fonte: elaborada pelo autor.

A Figura 3.50 apresenta os mapas de Karnaugh correspondentes a cada entrada dos flip-flops e da saída S.

Passo 4: na Figura 3.50 também estão listadas as expressões simplificadas para cada entrada dos flip-flops e da saída S.

Figura 3.50 | Mapas de Karnaugh com simplificações (a) J_2 (b) K_2 (c) J_1 (d) K_1 (e) J_0 (f) K_0 (g) S

		$Q_1 Q_0$	00	01	11	10
		EQ_2	00	01	11	10
00					1	
01	X	X		X	X	
11	(X)	X	X	X	X	
10	1					

$$J_2 = E\bar{Q}_1\bar{Q}_2 + \bar{E}Q_1Q_2$$

(a)

		$Q_1 Q_0$	00	01	11	10
		EQ_2	00	01	11	10
00			X	X	X	X
01					1	
11	1					
10	(X)	X	X	X	X	

$$K_2 = E\bar{Q}_1\bar{Q}_2 + \bar{E}Q_1Q_2$$

(b)

		$Q_1 Q_0$	00	01	11	10
		EQ_2	00	01	11	10
00			1	(X)	X	X
01			1	(X)	X	X
11	1			X	X	(X)
10	1			X	(X)	X

$$J_1 = E\bar{Q}_0 + \bar{E}Q_0 = E \oplus Q_0$$

(c)

		$Q_1 Q_0$	00	01	11	10
		EQ_2	00	01	11	10
00			X	(X)	1	
01	X		X	(X)	1	
11	(X)	X	X			
10	X	X				1

$$K_1 = E\bar{Q}_0 + \bar{E}Q_0 = E \oplus Q_0$$

(d)

		$Q_1 Q_0$	00	01	11	10
		EQ_2	00	01	11	10
00			1	X	X	1
01	1		X	X	X	1
11	1	X	X	X	1	
10	1	X	X	X	1	

$$J_0 = 1$$

(e)

		$Q_1 Q_0$	00	01	11	10
		EQ_2	00	01	11	10
00			X	1	1	X
01	X			1	1	X
11	X	1	1	1	X	
10	X	1	1	1	X	

$$K_0 = 1$$

(f)

$Q_1 Q_2$	00	01	11	10
$E Q_2$	1	1	1	1
00	1	1	1	1
01				
11				
10				

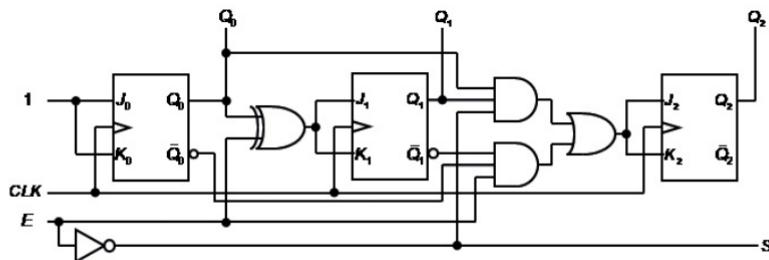
$$S = \bar{E}$$

(g)

Fonte: elaborada pelo autor.

Passo 5: o circuito final é apresentado na Figura 3.51.

Figura 3.51 | Circuito final



Fonte: adaptada de Capuano (2014, p. 116).



Pesquise mais

Caro aluno, esperamos que tenha gostado dessa jornada de aprendizado pelo mundo dos circuitos sequenciais. O que vimos até aqui mal "arranha a superfície" desse tema tão desafiador e atual. De fato, podemos dizer que apenas introduzimos o assunto. Portanto, não pare por aqui. Continue sua pesquisa. Um bom ponto de partida para se aprofundar no assunto é o livro:

VAHID, Frank. **Sistemas digitais**: projetos, otimização e HDLs. Porto Alegre: Bookman, 2008. 560 p.

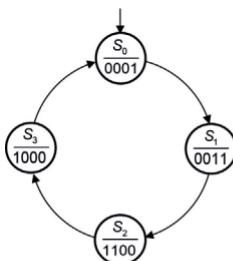
Sem medo de errar

Sua empresa está em um momento crucial: vocês foram contratados por uma importante empresa de engenharia e eletrônica, cujo principais produtos são componentes automotivos. O projeto que foi solicitado a vocês é um gerador de sequências para controlar o motor de passo da nova injeção eletrônica que eles vão produzir.

O gerador de sequência deve produzir a seguinte saída: 0001, 0011, 1100 e 1000, de forma cíclica.

Você inicia o seu projeto com um esboço do diagrama de estados, com quatro estados denominados de S_0 , S_1 , S_2 e S_3 , para o problema proposto, como o mostrado na Figura 3.52.

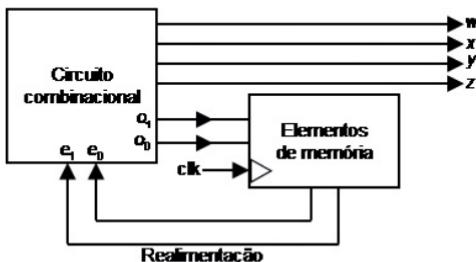
Figura 3.52 | Diagrama de estado do gerador de sequência



Fonte: elaborada pelo autor.

Para auxiliar no projeto, você faz também um diagrama de bloco do controle do gerador de sequência, na Figura 3.53. Ao fazer esse exercício de projeto, você estabelece que o controle do gerador de sequência precisa de um gerador de estado de dois bits (ou seja, dois flip-flops) para representar os quatro estados possíveis, uma vez que, de (3.1), temos $n_f = \text{int}(\log_2 4) = 2$.

Figura 3.53 | Diagrama de bloco do controle do gerador de sequência



Fonte: elaborada pelo autor.

Seu circuito possui quatro saídas lógicas, w, x, y e z, além de o_0 e o_1 , que representam os estados. O circuito combinacional não possui entrada externa, somente e_0 e e_1 , que representam os estados. Para montar a tabela de estados para o controle do gerador de sequência (Tabela 3.14), você codificou os estados como: $S_0:00$, $S_1:01$, $S_2:10$ e $S_3:11$.

Tabela 3.14 | Tabela de estados para o controle do gerador de sequência

Entradas		Saídas					
e_1	e_0	w	x	y	z	o_1	o_0
S_0	0 0	0	0	0	1	0	1
S_1	0 1	0	0	1	1	1	0
S_2	1 0	1	1	0	0	1	1
S_3	1 1	1	0	0	0	0	0

Fonte: elaborada pelo autor.

Apenas observando a Tabela 3.14, você pode deduzir as equações lógicas para o circuito combinacional, de modo que:

$$w = e_1$$

$$x = e_1 \bar{e}_0$$

$$y = \bar{e}_1 e_0$$

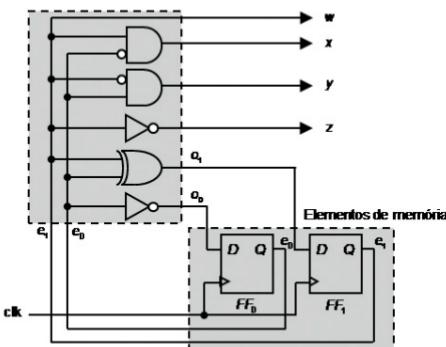
$$z = \bar{e}_1$$

$$o_1 = e_1 \text{ xor } e_0$$

$$o_0 = \bar{e}_0$$

O circuito denominado “elementos de memória” pode ser feito com um registrador de 2 bits feito com *flip-flops* do tipo D. Assim, o circuito final é mostrado na Figura 3.54

Figura 3.54 | Arquitetura do controle do gerador de sequência



Fonte: elaborada pelo autor.

Por fim, faça uma pesquisa buscando quais são os CLs comerciais que você deve usar na montagem do seu projeto.

Avançando na prática

Atualização do sistema de controle de vagas de um estacionamento

Descrição da situação-problema

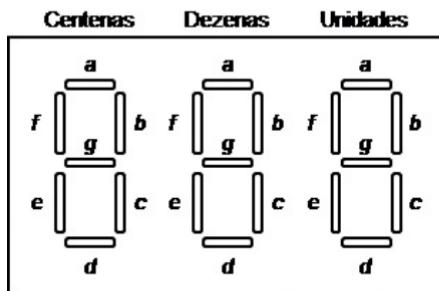
Em seu último trabalho, você foi contratado por um pequeno shopping local para desenvolver um sistema de controle de estacionamento de veículos. O problema consistia em fazer um projeto de monitoração dos espaços disponíveis em uma centena de vagas de um estacionamento e prover uma indicação luminosa da condição de “lotado”, abaixando a cancela que dá acesso à entrada. O cliente, embora satisfeito com sua solução, decidiu mudar o conceito do seu pedido. Nessa nova situação, o cliente pediu que, além do que já havia sido feito, fosse adicionado na entrada do estacionamento um painel indicando o número de vagas.

Resolução da situação-problema

Em uma atualização do projeto, o cliente pediu que fosse adicionado na entrada do estacionamento um painel indicando o número de vagas.

Para o painel, você usará um display de sete segmentos triplo (um para cada dígito de uma centena), como pode ser visto na Figura 3.55.

Figura 3.55 | Display de sete segmentos triplo

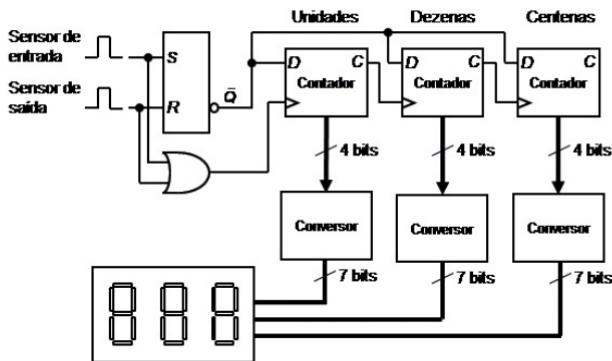


Fonte: elaborada pelo autor.

Cada dígito no display é acionado individualmente e, para facilitar o seu projeto, você decide usar três contadores de década de quatro bits (conta de 0 a 9) crescente/decrescente encadeados. Cada contador deve ter uma entrada para indicar a direção da contagem, crescente para $D=0$ e decrescente para $D=1$. O contador deve ter também uma saída, C, que indica com um nível lógico 1 que a contagem foi reciclada (1001 para 0000 ou 0000 para 1001).

Os quatro bits de saída de cada contador são ligados a decodificadores BCD-7 segmentos, e as saídas do decodificador alimentam o display. A Figura 3.56 traz o diagrama lógico para o circuito proposto.

Figura 3.56 | Diagrama lógico para o circuito contador de carros no estacionamento



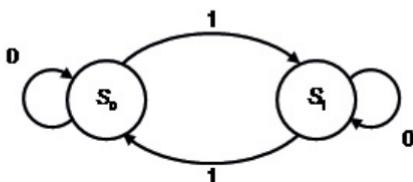
Fonte: elaborada pelo autor.

Embora exista no mercado esse tipo de contador encapsulado em um único circuito integrado (CI), projete o circuito lógico de cada contador utilizando o procedimento apresentado nesta seção.

Faça valer a pena

1. Um diagrama de estado representa um modelo gráfico de um sistema sequencial e especifica as transições entre os estados do sistema, as quais, por sua vez, descrevem como um estado particular passa para outro. Um sistema sequencial pode convenientemente ser representado por um diagrama de estado, como o diagrama de estado genérico representado na Figura 3.57.

Figura 3.57 | Diagrama lógico para o circuito contador de carros no estacionamento



Fonte: elaborada pelo autor.

Nesse contexto, avalie as afirmativas a seguir:

- Em um diagrama de estados, cada círculo representa um estado, de modo que, no diagrama do sistemas representado na Figura 3.57, há dois estados, S_0 e S_1 .
- As setas representam as transições de estado, de um determinado sistema, devido a uma ação (entrada) em um certo estado.
- O sistema da Figura 3.57, por possuir apenas dois estados, pode ser representado com apenas 1 bit, ou seja, apenas 1 flip-flop.
- Considerando o sistema no estado S_0 , uma entrada igual a 1 leva o sistema para o estado S_1 .

É correto o que se afirma em:

- a) I e III, apenas.
- b) II e IV, apenas.
- c) II, III e IV, apenas.
- d) I, II e III, apenas.
- e) I, II, III e IV.

2. Uma máquina de estados finitos é um sistema algébrico usado para modelar programas de computador ou circuitos lógicos. Em tese, todos os circuitos síncronos, incluindo os registradores e contadores movidos por um *clock*, podem ser representados por máquinas de estados finitos.

Sobre os modelos de máquinas de estado finito, avalie as afirmações a seguir:

I. Existem dois modelos tradicionais para a implementação de máquinas de estados. São eles: as máquinas Moore e as máquinas Mealy. Embora a implementação desses dois modelos ocorra de forma diferente, o comportamento das duas máquinas modelando um mesmo sistema é idêntico.

II. Nas máquinas Moore, a saída depende da entrada e do estado atual, de modo que o uso de uma máquina Moore geralmente leva a uma redução no número de estados.

III. Nas máquinas Mealy, a saída depende somente da entrada, o que costuma simplificar o comportamento da máquina.

É correto o que se afirma em:

- a) I, apenas.
- b) II e III, apenas.
- c) I e II, apenas.
- d) I e III, apenas.
- e) I, II e III.

3. As máquinas de estado também podem ser classificadas como sistemas síncronos ou assíncronos, conforme as características de aplicação do sistema de *clock* nos circuitos. Os sistemas sequenciais síncronos são aqueles em que os circuitos operam de maneira sincronizada com a entrada *clock*. Já os sistemas sequenciais assíncronos são aqueles atuam de maneira assíncrona, ou seja, sem sincronismo entre os elementos do circuito.

Nesse contexto, avalie as afirmativas a seguir:

I. Em um sistema assíncrono, todos os elementos, *flip-flops* ou registradores de estado são comandados pelo mesmo pulso de *clock*, em função do acionamento no tempo, uma vez que têm ligação dessa entrada em comum.

II. Em um sistema síncrono, as variações internas e de saída ocorrem em instantes diferenciados com relação ao pulso de *clock*.

III. Uma máquina Moore é considerada um sistema síncrono.

IV. Uma máquina Mealy é considerada um sistema assíncrono.

É correto o que se afirma em:

- a) I e II, apenas.
- b) III e IV, apenas.
- c) I e III, apenas.
- d) II e IV, apenas.
- e) I, II, III e IV.

Referências

CAPUANO, Francisco Gabriel. **Sistemas digitais:** circuitos combinacionais e sequenciais. São Paulo: Érica, 2014.

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

SZAJNBERG, Mordka. **Eletônica digital:** teoria, componentes e aplicações. Rio de Janeiro: LTC, 2014.

TEXAS INSTRUMENTS. **CD54HC190, CD7HC190, CD54HC191, CD74HC191, CD54HCT191, CD74HCT191 Synchronous up/down counters with down/up mode control datasheet.** Dallas, 2003. 31 p. Disponível em: <<http://www.ti.com/lit/ds/symlink/cd74hc190.pdf>>. Acesso em: 21 fev. 2018.

TOCCI, Ronald J.; WIDNER, Neal S.; MOSS, Gregory L. **Sistemas digitais:** princípios e aplicações. 11. ed. São Paulo: Pearson Prentice Hall, 2011.

VAHID, Frank. **Sistemas digitais:** projetos, otimização e HDLs. Porto Alegre: Bookman, 2008.

Especificação e interfaceamento de circuitos integrados

Convite ao estudo

Caro aluno, nas seções desta unidade trabalharemos o conhecimento acerca da especificação, como se dá o interfaceamento e como utilizar circuitos integrados das famílias TTL e CMOS. Em linhas gerais, temos que ter em mente o que é um circuito integrado digital e como os níveis de tensão influenciam na definição da tecnologia escolhida.

Em seguida, vamos entender o uso de circuitos digitais MOS e CMOS como chaves, e também em dispositivos como LEDs, campainhas, relés, motores e solenoides.

Na sequência, entenderemos como isolar digitalmente um circuito utilizando isoladores digitais e optoisoladores. Na sequência, na última seção, também vamos aprofundar nosso conhecimento acerca da utilização de sensores de efeito Hall e de como encontrar problemas em circuitos digitais simples. Por fim, aprenderemos como utilizar um circuito digital para controlar um servo motor.

Para compreender onde queremos chegar, tenha em mente que, na indústria, sempre existe a necessidade de bons engenheiros com conhecimento em circuitos digitais. Profissionais que conseguem avaliar problemas como o acoplamento elétrico entre dois níveis de tensão distintos, ou mesmo problemas intermitentes em um projeto digital, são muito bem remunerados. Dessa forma, o texto aqui exposto serve como início para que você amplie seus conhecimentos no assunto. Você pode fazer parte desse time de profissionais digitais. É um trabalho demorado e contínuo, mas o retorno

profissional é interessante. Sendo assim, é interessante fazer algumas perguntas: o que são circuitos CMOS e TTL? O que são níveis lógicos e o que é margem de ruído? Como utilizar um circuito TTL e CMOS como chave?

Essas perguntas devem ser respondidas nesta unidade. Outras perguntas devem surgir e também podem ser respondidas aqui. Contudo, muitas outras dúvidas devem surgir, e o importante é que você, aluno, tenha curiosidade e amplie seu conhecimento de forma sólida e independente.

Uma maneira de ampliar o conhecimento é inserindo um cenário de forma a contextualizar o mundo em que vivemos com questões atuais e relevantes.

Considere, então, que a sua empresa está bem consolidada e bem aceita no mercado, e dinheiro já não é mais um problema. Nesse sentido, você percebeu uma boa oportunidade de investimento em pequenas empresas iniciantes em uma startup. Essas empresas podem, no futuro, render milhões, caso a ideia seja bem aceita no mercado. Para isso, você avalia investir um certo montante de dinheiro em uma boa ideia. Uma empresa chamou a sua atenção. Essa empresa atua na área agrícola, e sua solução de automação, a princípio, gera uma economia substancial de rações para frangos. Você decide avaliar todos os detalhes do projeto de hardware que eles possuem. Para isso, você pediu um detalhamento de toda a solução, em especial a do sistema de acionamento dos relés que controlam os alimentadores dos frangos, pois você quer entender algumas questões.

Os níveis de tensão FAN-IN e FAN-OUT das portas lógicas utilizadas no circuito antitrepidação de contagem de frangos estão corretos? Como é feito o acionamento do relês e dos motores de passo do sistema dos alimentadores? Como os optoisoladores entre a lógica de acionamento e o relé funcionam? Essas são algumas questões que merecem sua atenção, antes que você invista em uma empresa iniciante. De qualquer forma, parabéns por ter chegado tão longe com a sua empresa. Agora você pode ajudar outras empresas a

crescerem. Não perca essa oportunidade. A satisfação de ser o investidor anjo de uma empresa bem-sucedida é indescritível. Até agora todo o conhecimento que você adquiriu fez parte inteiramente da sua dedicação e profissionalismo. Esteja atento a todos conceitos que serão apresentados aqui. Não se contente somente com o conhecimento apresentado aqui; extrapole. Boa sorte e sucesso nessa empreitada.

Seção 4.1

Especificações de circuitos integrados

Diálogo aberto

Caro aluno, vamos aprofundar nossos conhecimentos sobre circuitos digitais TTL e CMOS. Esses dispositivos são importantes por serem a base de qualquer conhecimento sobre projetos digitais. Neste capítulo, é apresentado um conhecimento fundamental – a definição dos níveis lógicos que esses dispositivos podem assumir, bem como os detalhes construtivos e funcionais das famílias TTL e CMOS.

Nesse sentido, para contextualizar o assunto, considere o seguinte cenário: a sua empresa está bem consolidada e bem aceita no mercado, dinheiro já não é mais um problema. Nesse sentido, você percebeu uma boa oportunidade de investimento em pequenas empresas iniciantes em uma startup. Essas empresas podem, no futuro, render milhões, caso a ideia seja bem aceita no mercado. Para isso, você avalia investir um certo montante de dinheiro em uma boa ideia. Uma empresa chamou a sua atenção.

Essa empresa atua na área agrícola, e sua solução de automação, a princípio, gera uma economia substancial de rações para frangos. Você decide avaliar todos os detalhes do projeto de hardware que eles possuem. Para tanto, você pediu um detalhamento de toda a solução, em especial a do sistema de acionamento dos relés que controlam os alimentadores dos frangos. A primeira questão que você tentará entender é identificar se o sistema é robusto e identificar se os níveis de tensão utilizadas no circuito antitrepidação de contagem de frangos estão corretos (Figura 4.1).

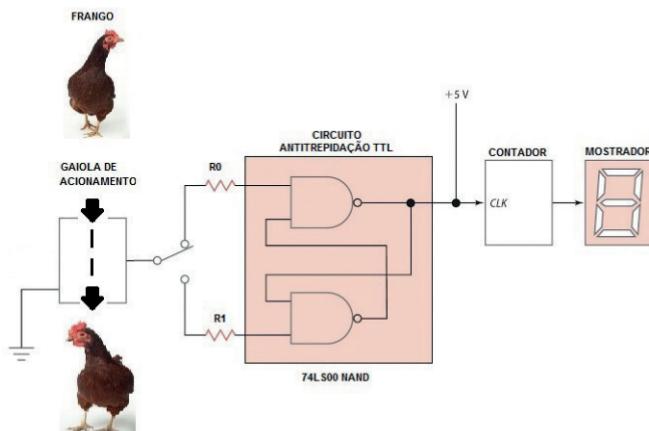
Considere as seguintes especificações:

- Portas lógicas SN74LS00 NAND, desenvolvida pela Texas Instruments. Documento: SDLS025D –DECEMBER 1983–REVISED MAY 2017.

- Tensão de alimentação do Sistema: **+5 Volts**.

- Tensão de acionamento do circuito acionador: **+3.5 Volts**.

Figura 4.1 | Sistema de contagem de frangos com circuito antitrepidação



Fonte: elaborada pelo autor.

Considere que somente um frango entra de cada vez no circuito de acionamento. Analise as questões levantadas e crie um relatório com as especificações do circuito antitrepidação. Essa análise é fundamental, pois possibilita que você comece a entender como obter as informações corretas sobre os circuitos na folha de dados. Bom trabalho.

Não pode faltar

Caro aluno, circuitos digitais são implementados utilizando diversas tecnologias. É o caso dos circuitos da família TTL, que são fabricados utilizando transistores bipolares (FLOYD, 2007). Já os circuitos da família CMOS utilizam transistores unipolares.



Assimile

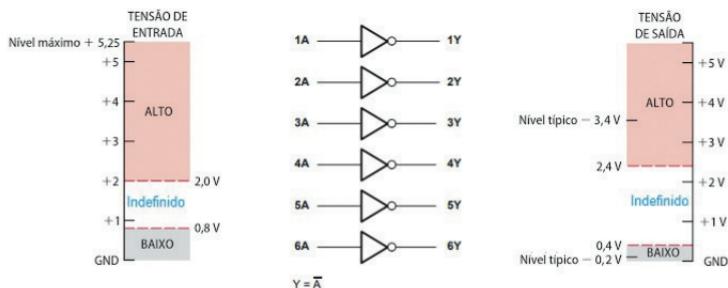
Circuitos digitais podem ser classificados conforme o seu principal componente eletrônico usado na fabricação. Eles podem ser classificados como Transistores Bipolares e Transistores Unipolares. Transistores Bipolares são transistores que utilizam junção NPN e PNP como principal elemento do circuito. Já Transistores Unipolares utilizam o efeito de campo, ou seja, canal P e canal N como principal elemento do circuito.

Em qualquer campo da área da microeletrônica, quando iniciamos a análise de um novo dispositivo começamos analisando algumas características básicas como a tensão, a impedância ou resistência ou a corrente desse novo dispositivo. Iremos verificar na próxima seção as características de tensão.

Níveis Lógicos e Margem de Ruído

Segundo Tocci (2011), temos que ter em mente que um circuito entende níveis de tensão específicos. Dependendo da tensão que se encontra na porta de entrada ou saída do dispositivo, o nível lógico é considerado 1 (alto) ou zero (baixo). A Figura 4.2 ilustra os níveis de tensão de entrada e de saída do CI **74LS04** da família lógica TTL.

Figura 4.2 | Níveis TTL do CI 74LS04



Fonte: elaborada pelo autor

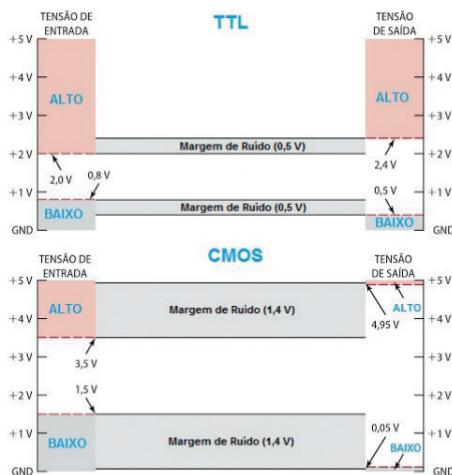
O fabricante desta porta inversora especifica que a tensão de entrada (Figura 4.2 – Tensão de Entrada) deve estar entre **0,8 volt** e **GND** para que o sinal corresponda ao Nível Lógico Baixo e opere corretamente. Em contrapartida, para que o dispositivo corresponda ao Nível Lógico Alto, a tensão de entrada deve estar no máximo **5,25 volts** e **2,0 volts**. Já a região entre **2,0 volts** e **0,8 volt** corresponde a uma região indeterminada ou indefinida. Caso algum sinal esteja operando na região de indefinição, o fabricante desta porta inversora não garante que o dispositivo irá entender o nível lógico que está na porta de entrada, portanto qualquer sinal nessa região deve ser evitado. Por exemplo, um sinal de **1,2 volts** está na região de indefinição e, portanto, o sinal de saída será imprevisível.

Já as tensões saídas do inversor (Figura 4.2 – Tensão de Saída) correspondem ao Nível Lógico Baixo quando o sinal de saída está na faixa aproximada de **0,2** a **0,4 volt**. Já uma saída com Nível

Lógico Alto corresponde a qualquer sinal na faixa entre **2,4 volts** e o valor máximo permitido de **5,25 volts**. Quanto maior for o valor da resistência menor será a corrente, portanto maior será a tensão de saída (lei de ohm), ou seja, o valor da tensão de saída está ligado à corrente de carga. Já a faixa entre **0,4 e 2,4 volts** corresponde à região de indeterminação do circuito. Por exemplo, se o valor de saída for de **1,8 volt**, podem ocorrer inconsistências na leitura do sinal. Todos esses valores de tensão pertencem à família TTL. Por outro lado, a família CMOS possui uma faixa ampla que em geral varia entre +3 volts até +15 volts. Um inverter CMOS entende entre 70% e 100% do valor da tensão de alimentação (V_{DD}) como Nível Alto, e entre 30% e 0% como Nível Baixo.

Tocci (2011) analisa um aspecto interessante que deve ser comentado: cada dispositivo e tecnologia é sensível a tensões indesejadas (ou espúrias). Essa característica é chamada de Imunidade a Ruidos. Quando estamos tratando de circuitos digitais, estamos tratando de Margem de Ruído. A Figura 4.3 compara as margens de ruídos de dois circuitos da família TTL e CMOS, respectivamente. Observe e compare os valores das margens de ruídos para cada família. Podemos concluir que dispositivos CMOS possuem uma margem de ruído maior quando comparados a um circuito TTL – portanto circuitos CMOS são mais robustos a ruídos.

Figura 4.3 | Comparação da margem de ruído em circuitos da família TTL e CMOS



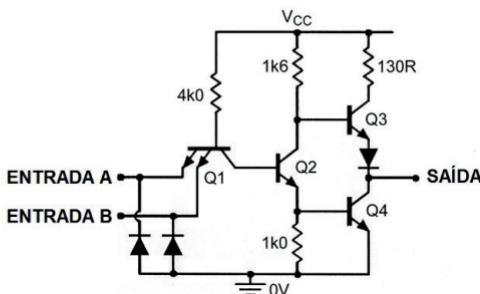
Fonte: adaptada de Tocci (2011, p. 430).

Mas o que são ruídos? Para Tocci (2011), ruídos são tensões ou flutuações espúrias não desejadas que ocorrem geralmente nos cabos de conexão, nas trilhas das placas de circuitos impressos ou mesmo no interior dentro do circuito integrado. O fato é que essas tensões geram erros e podem ocasionar falhas simples ou mesmo graves em um sistema digital. Ruídos devem ser evitados sempre – engenheiros passam anos estudando alguns aspectos do ruído em determinadas tecnologias, e aqui está um campo bem promissor e com um bom retorno financeiro.

Família TTL

Circuitos TTL (*Transistor–Transistor Logic*) são dispositivos pertencentes a uma família de circuitos digitais construídos com transistores bipolares (BJTs) e resistores. Inicialmente, esses componentes foram desenvolvidos pela empresa Texas Instruments, contudo hoje diversos fabricantes de semicondutores produzem esses circuitos. Sabemos que estamos utilizando um circuito da família TTL quando as séries que marcam o circuito começam com o número “54” e “74”, sendo o primeiro para uso militar e o segundo para uso civil/comercial. Então sempre que um circuito começar com o número “74”, podemos associá-lo à família TTL. A Figura 4.4 é um exemplo da arquitetura interna de uma porta lógica NAND como a utilizada em qualquer CI 74LS00.

Figura 4.4 | Esquemático da porta lógica NAND



Fonte: elaborada pelo autor.

A principal característica de uma família TTL é o fato da alimentação ser **+5 volts**. Dessa forma, observando as características de níveis lógicos e margem de ruído, em um dispositivo TTL o Nível Lógico Zero será a ausência de tensão ou **0 Volt** e Nível Lógico

Um será a tensão de **+5 Volts**, ou seja, a tensão de alimentação (V_{cc}). Por fim, em um projeto que utilize circuitos TTL, é importante prestar atenção nas correntes que são requeridas tanto nas entradas quanto nas saídas. Isso porque a fonte de alimentação de um circuito TTL deve garantir valores de corrente I_{IH} , I_{IL} , I_{CCH} e I_{CCL} , especificados nas folhas de dados do circuito integrado.

Os circuitos TTL apresentam subdivisões, e estas e suas respectivas características estão listadas na Tabela 4.1.

Tabela 4.1 | Subdivisões TTL

Identificação	Subdivisão	Característica
54xx/74xx	Standard	-
54Lx/74Lx	Low Power	Baixo Consumo
54Hx/74Hx	High Speed	Alta Taxa de Dados
54Sx/74Sx	Schottky	-
54LS/74LS	Low Power Schottky	-

Fonte: elaborada pelo autor.

A escolha do componente vai depender de diversos fatores de projeto, como custo por unidade (**54xx/74xx** – Standard), necessidade de alta velocidade de dados (**54Hx/74Hx** – High Speed) e baixo consumo de energia (**54Lx/74Lx** – Low Power).



Pesquise mais

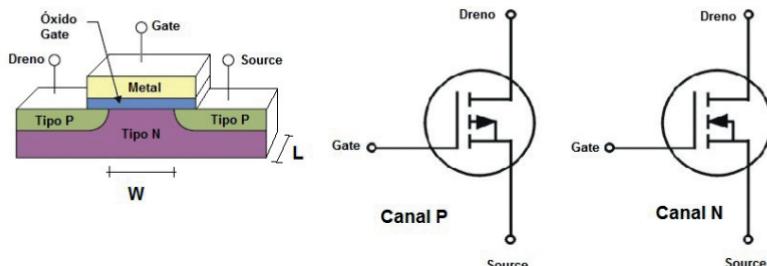
Pesquise a folha de dados da porta lógica TTL NAND - DM74LS00 Quad 2-Input NAND Gate. Agosto 1986. Esse componente, mesmo tendo sido criado em 1986, ainda é fabricado e vendido atualmente. Analise o diagrama de conexão, a tabela de funções e as características elétricas especificada pelo fabricante. Fique atento aos limites de tensão e corrente do dispositivo. Extrapole, analisando os atrasos de propagação do circuito.

Tecnologias MOS e CMOS

Segundo Tocci (2011), circuitos baseados na tecnologia MOS são construídos utilizando um composto químico chamado Óxido Semicondutor Metálico. Com esse composto, é possível criar os chamados transistores de efeito de campo de semicondutor de

óxido metálico, mais conhecidos como MOSFETs. Circuitos MOS apresentam uma arquitetura simples quando comparados com TTL, ocupando menos espaço em uma pastilha de silício. Dessa forma, um número maior de transistores MOS podem ser colocados em uma mesma área de silício que seria ocupada por transistores BJTs. Quando se fala em escala de integração de dispositivos, tecnologias baseadas em MOS são amplamente utilizadas em escala LSI (do inglês *Large Scale Integration*) e VLSI (do inglês *Very large scale integration*). Microprocessadores, coprocessadores, memória e circuitos de tempo em geral utilizam tecnologia MOS. Circuitos à base de MOS ainda podem ser divididos em NMOS e PMOS, ou seja, o canal que compõe um MOSFET, por exemplo, pode ser feito de material semicondutor do tipo N ou do tipo P. A Figura 4.5 exemplifica a arquitetura interna em corte e o esquema elétrico de um MOSFET.

Figura 4.5 | Corte Transversal e esquema elétrico de um MOSFET



Fonte: elaborada pelo autor.

Quando um projetista de semicondutores desenvolve um transistor do tipo MOS, ele deve considerar, entre outros parâmetros de dimensionamento do transistor, a largura do canal W . A largura desse canal é responsável por limitar a passagem de corrente no transistor. Já o comprimento do canal L está relacionado com o tempo que um elétron leva para percorrer o canal, limitando a resposta em frequência do MOSFET.

Caro aluno, vamos questionar o seguinte cenário. Dois transistores, um com canal P e o outro com canal N. O que aconteceria se conectássemos esses dois transistores? Como seria essa ligação? E para qual finalidade?

Esses questionamentos podem ser respondidos quando analisamos

dispositivos baseados em tecnologia CMOS ou Óxido Semicondutor Metálico com Simetria Complementar. Esses dispositivos utilizam transistores MOS com canal N e canal P interconectados entre si. Existem vantagens em utilizar esses dispositivos: (i) consomem pouca energia; (ii) apresentam baixo custo, (iii) projeto simples; (iv) níveis lógicos amplos na saída; e (v) uma significativa margem de ruído. Contudo, têm desvantagem com relação à velocidade de saída dos dados: circuitos CMOS são mais lentos quando comparado com circuitos Bipolares.

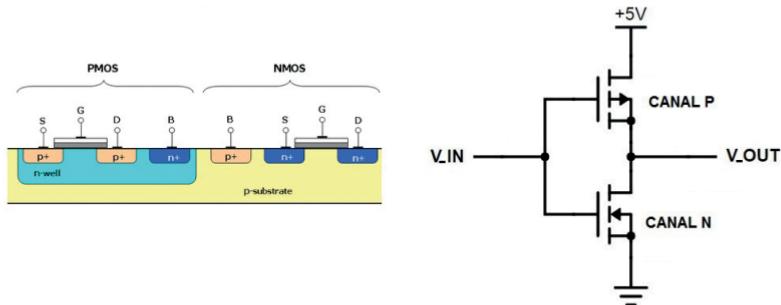


Refletia

Circuitos CMOS são extremamente sensíveis a eletricidade estática. Uma pequena descarga estática ou uma tensão transitória fora do padrão pode danificar a camada de dióxido de silício – essa camada age como um pequeno capacitor e pode ser rompida facilmente. Reflita sobre as condições de armazenagem desses componentes: como espumas e embalagens antiestáticas podem minimizar esse problema? De que forma o aterramento de instrumentos como ferro de solda evitam descargas elétricas no circuito? Por que desligar a fonte de alimentação do circuito e manter os sinais de entrada em níveis inferiores aos da tensão? Analise sobretudo como esses cuidados podem influenciar na decisão de um projeto.

Um dispositivo muito comum que utiliza tecnologia CMOS é ilustrado na Figura 4.6. Essa figura mostra um corte transversal e a arquitetura interna de uma porta CMOS inversora.

Figura 4.6 | Corte transversal e arquitetura porta CMOS inversora



Fonte: elaborada pelo autor.

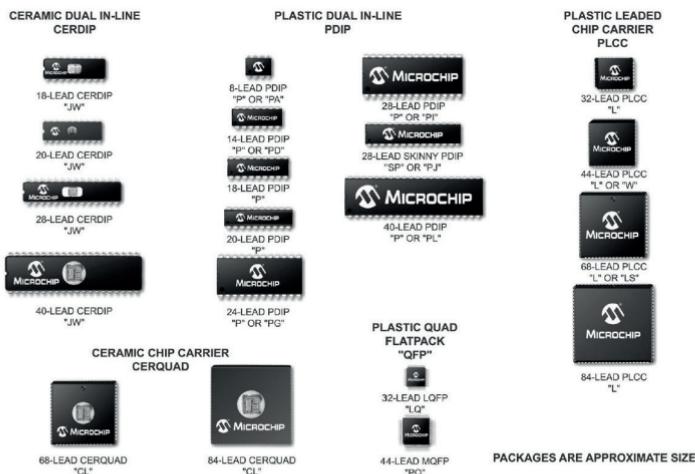
É possível observar no diagrama que representa a arquitetura da porta inversora que o transistor na parte superior representa um MOSFET com canal P, enquanto o transistor na parte inferior representa um MOSFET com canal N. Dessa forma, quando a tensão de entrada (V_{IN}) é levada ao Nível Lógico Baixo, o MOSFET superior começa a conduzir, ao passo que o MOSFET inferior é desligado, de modo que a saída V_{OUT} é levada ao Nível Lógico Alto. Todavia, quando a tensão de entrada (V_{IN}) é levada ao Nível Lógico Alto, o MOSFET superior é desligado, ao passo que o MOSFET inferior é ligado, de modo que a saída V_{OUT} é levada ao Nível Lógico Baixo. Essa dinâmica caracteriza uma porta inversora clássica. Uma característica bem interessante é a tensão de alimentação desses dispositivos. Ela pode variar entre +5 V, +6 V, +9 V e até +12 V.



Exemplificando

Todo dispositivo apresenta um encapsulamento, que é uma estrutura onde é fixada a pastilha de silício como dispositivo. Sempre que chegamos na etapa da especificação do componente, devemos ficar atentos ao tipo de encapsulamento por razões como o (i) tipo de solda disponível para aquele projeto, (ii) como será a manutenção do sistema e (iii) disponibilidade do produto. A Figura 4.7 exemplifica alguns dos diversos tipos de encapsulamentos utilizados na indústria.

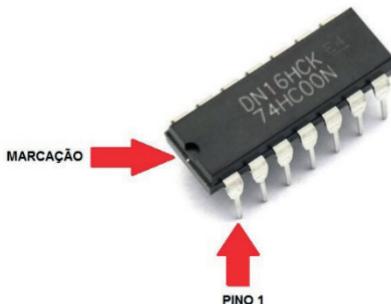
Figura 4.7 | Encapsulamentos disponíveis no mercado



Fonte: Rako (2016, p. 1).

Por fim, sempre devemos consultar a folha de dados do componente para, por exemplo, identificarmos os respectivos pinos e funções de cada porta e os níveis de tensões de alimentação. A Figura 4.8 aponta para duas importantes referências em um encapsulamento: a marcação de referência e o pino 1.

Figura 4.8 | Referências em um encapsulamento



Fonte: Rako (2016, p. 1).

Outras especificações de CIs digitais

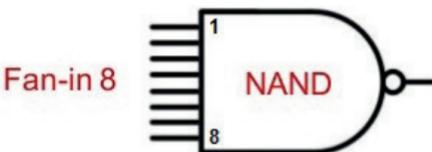
Caro aluno, até o momento tratamos sobre níveis de tensão e margem de ruído de um dispositivo TTL ou CMOS. Contudo, circuitos digitais também apresentam outras especificações importantes que devemos considerar em um projeto de sistema digital, tais como FAN-IN e FAN-OUT, capacidade de corrente, atraso na propagação do sinal e sobretudo a potência dissipada.

Segundo Tocci (2011), quando analisamos a PORTA DE SAÍDA de um dispositivo, devemos verificar a capacidade de corrente que essa porta pode fornecer. Essa capacidade limita a quantidade de portas de entrada que podem ser conectadas a essa porta de saída – portanto, estamos falando de FAN-OUT. Se um dispositivo possuir um FAN-OUT 15, significa que ele pode acionar até 15 outros dispositivos da mesma subfamília. A capacidade de corrente na saída está diretamente ligada ao FAN-OUT, o que limita a potência e a corrente no coletor do transistor BJT.

Quando analisamos as PORTAS DE ENTRADA de um dispositivo,

dizemos que um dispositivo possui FAN-IN igual a 5 quando ele possui cinco portas de entrada para uma das lógicas. O diagrama da Figura 4.9 ilustra uma porta lógica NAND com 8 entradas, portanto uma FAN-IN 8. Quanto maior o FAN-IN, mais lento é o dispositivo, tendo em vista que a sua capacidade aumenta com o número de portas.

Figura 4.9 | Diagrama de uma porta lógica NAND com FAN-IN 8



Fonte: elaborada pelo autor.

O atraso na propagação do sinal está relacionado à velocidade ou rapidez com que um sinal chega na saída como resposta à mudança do sinal de entrada do dispositivo. Como mencionado anteriormente, circuitos TTL são mais rápidos que circuito CMOS. Contudo, existe uma família de circuitos CMOS chamada FACT CMOS (Tecnologia CMOS Avançada), que supera muitos dispositivos TTL. Outra solução aceita pela indústria é o uso famílias que utilizam a tecnologia ECL (Lógica Emissor Acoplado) e à base de arseníeto de gálio; esses dispositivos são extremamente rápidos. Por fim, analisando o consumo de um circuito TTL, devemos considerar que esses dispositivos apresentam um atraso na ordem de 10ns, e circuitos CMOS, atrasos na ordem de 40 a 60ns. Contudo, circuitos CMOS consomem na ordem de 0,001mW, enquanto circuitos TTL consomem na ordem de 10mW. Outro aspecto que deve ser levado em consideração é o consumo com o aumento da frequência de operação do dispositivo. Existem muitas pesquisas sobre o assunto, mas o que de fato ocorre é que a escolha da tecnologia FACT CMOS pode ser uma boa opção entre velocidade e potência consumida.

Todo projeto digital requer uma boa análise dos aspectos que foram levantados nesta unidade – tanto os projetos simples quanto os mais complexos. Portanto, não fique limitado ao que foi exposto aqui: procure outras fontes e converse com especialistas. Bons estudos e persista, pois o assunto é muito interessante.

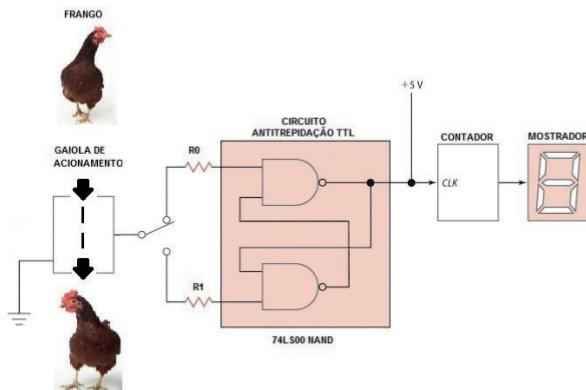
Sem medo de errar

Caro aluno, a sua empresa já um sucesso no mercado de soluções de tecnologia. Agora você está tendo a oportunidade de ajudar outras empresas menores a crescer. Surgiu a oportunidade de você investir um certo montante de dinheiro em uma empresa nascente em uma *startup* de renome. Relembrando, essa empresa atua na área agrícola e sua solução de automação, a princípio, gera uma economia substancial de ração para frangos. Você decide avaliar todos os detalhes do projeto de hardware que eles possuem. Para isso, você pediu um detalhamento de toda a solução, em especial a do sistema de acionamento dos relés que controlam os alimentadores dos frangos. A primeira questão que você tentará entender é identificar se o sistema é robusto e identificar se os níveis de tensão utilizados no circuito antitrepidação de contagem de frangos estão corretos (Figura 4.10).

Considere as seguintes especificações:

- Portas lógicas SN74LS00 NAND. Documento: SDLS025D – DECEMBER 1983–REVISED MAY 2017.
- Tensão de alimentação do sistema: **+5 Volts**.
- Tensão de acionamento do circuito acionador: **+5 Volts**.
- Corrente máxima nas portas de entrada: **+1 mA**.
- Resistores **R0** e **R1**: **10kΩ**.

Figura 4.10 | Sistema de contagem de frangos com circuito antitrepidação



Fonte: elaborada pelo autor.

Considere que somente um frango entra de cada vez no circuito de acionamento.

Para responder a esses questionamentos, você deve analisar com muita atenção a folha de dados do componente SN74LS00 NAND. Em posse da folha de dados SN74LS00 (1983) obtida no site da fabricante, serão analisadas inicialmente as condições de operação recomendadas para o circuito SNLS74LS00 (Figura 4.11).

Figura 4.11 | Condições de operação CI SN74LS00 NAND

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	SN54xx00	4.5	5	5.5	V
		SN74xx00	4.75	5	5.25	
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage	SNx400, SN74LS400, and SNx4S00		0.8		V
		SN54LS00		0.7		
I_{OH}	High-level output current	SN5400, SN54LS00, and SN74LS00		-0.4		mA
		SNx4S00		-1		
I_{OL}	Low-level output current	SNx400		16		
		SN54LS00		4		mA
		SN74LS400		8		
		SNx4S00		20		

6.6 Electrical Characteristics: SN4L500

over operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{IK}	$V_{CC} = \text{MIN}$ and $I_i = -18 \text{ mA}$			-1.5	V
V_{OH}	$V_{CC} = \text{MIN}$, $V_{IL} = \text{MAX}$, and $I_{OH} = -0.4 \text{ mA}$	2.5	3.4		V
V_{OL}	$V_{CC} = \text{MIN}$ and $V_{IH} = 2 \text{ V}$	$ I_{OL} = 4 \text{ mA}$	0.25	0.4	V
		$ I_{OL} = 8 \text{ mA (SN74LS00)}$	0.35	0.5	
I_i	$V_{CC} = \text{MAX}$ and $V_i = 7 \text{ V}$		0.1		mA
I_{iH}	$V_{CC} = \text{MAX}$ and $V_i = 2.7 \text{ V}$		20		μA
I_{iL}	$V_{CC} = \text{MAX}$ and $V_i = 0.4 \text{ V}$		-0.4		mA
I_{os}	$V_{CC} = \text{MAX}$	-20		-100	mA
I_{och}	$V_{CC} = \text{MAX}$ and $V_i = 0 \text{ V}$		0.8	1.6	mA
I_{ocl}	$V_{CC} = \text{MAX}$ and $V_i = 4.5 \text{ V}$		2.4	4.4	mA

Fonte: adaptada de SN74LS00 (1983).

Analizando a Figura 4.11, podemos concluir que os projetistas consideraram o circuito de uso comercial, ou seja, o SN74xxxx. Existe a possibilidade da escolha do SN54xxxx, mas considerando todos os aspectos de importação e restrição de uso militar imposto à família SN54xxxx e o custo de cada dispositivo (é consenso que famílias SN54xxxx são mais caras que famílias SN74xxxx), essa família foi descartada. Observe que **+5 Volts** é um nível de tensão adequado escolhido pelos engenheiros – esse valor pode ser verificado na Figura 4.11 no parâmetro V_{CC} nom. A princípio, a proposta feita pela empresa é adequada e merece sua atenção. Dessa forma, analise

mais detalhes da folha de dados e retome a unidade para consolidar o seu conhecimento sobre FAN-IN e FAN-OUT de um sistema. Em seu relatório, inclua todas as informações, como corrente máxima de operação em cada porta, valores que não podem ser excedidos nas portas de entrada e dimensionamento de resistores. Aproveite para analisar se esses requisitos estão adequados para o projeto proposto.

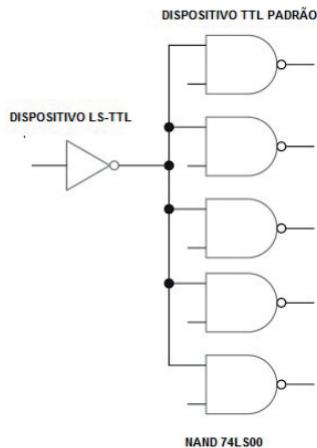
Avançando na prática

Análise da capacidade de corrente de um circuito TTL

Descrição da situação-problema

Caro aluno, você está considerando a possibilidade de investir em uma empresa nascente e tem verificado as soluções já desenvolvidas por ela. Para isso, você tem feito uma análise bem criteriosa dos projetos. Nesse sentido, um circuito (Figura 4.12) chamou sua atenção, e você pretende investigar se o FAN-IN e o FAN-OUT da solução é adequado. Você utilizará as informações retiradas na folha de dados já listados na Tabela 4.2.

Figura 4.12 | Circuito analisado



Fonte: elaborada pelo autor.

Tabela 4.2 | Características de corrente de entrada e saída dos circuitos TTL e LS-TTL

Família de Dispositivo	Corrente de Saída	Corrente de Entrada
TTL PADRÃO	$I_{OH} = 400 \mu A$ $I_{OL} = 16 mA$	$I_{IH} = 40 \mu A$ $I_{IL} = 1,6 mA$
LS-TTL	$I_{OH} = 400 \mu A$ $I_{OL} = 8 mA$	$I_{IH} = 20 \mu A$ $I_{IL} = 400 \mu A$

Fonte: elaborada pelo autor.

Você gostaria de saber se a porta inversora é capaz de acionar cinco portas NAND simultaneamente tanto quando o sinal for Nível Lógico Alto e Nível Lógico Baixo.

Resolução da situação-problema

Caro aluno, essa análise é importante. Caso o sistema esteja operando no limite, todo o projeto tem que ser refeito. Outro aspecto importante diz respeito a futuras modificações. Sempre devemos deixar uma pequena margem no projeto para evitar retrabalho, neste caso significa que o FAN-IN e FAN-OUT devem apresentar alguma margem.

Analisando o FAN-OUT em alto da porta inversora, considere:

- $I_{OH} = 400 \mu A$ do Circuito LS-TTL.
- $I_{IH} = 40 \mu A$ do Circuito TTL Padrão.

Logo o FAN-OUT em alto será $\frac{I_{OH}}{I_{IH}} : \frac{400\mu A}{40\mu A} \Rightarrow 10$

Analisando o FAN-OUT em baixa da porta inversora, considere:

- $I_{OL} = 8 mA$ do Circuito LS-TTL.
- $I_{IL} = 1,6 mA$ do Circuito TTL Padrão.

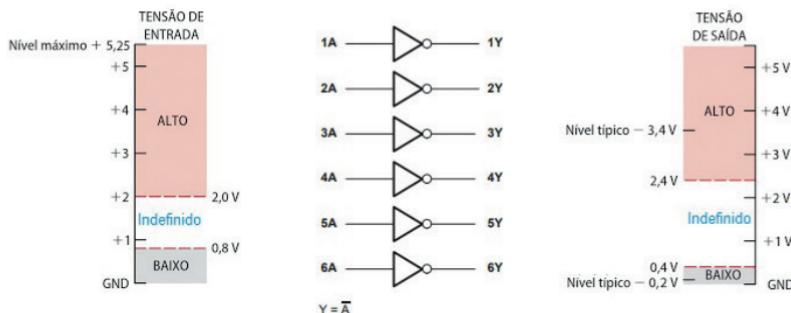
Logo o FAN-OUT em baixa será $\frac{I_{OL}}{I_{IL}} : \frac{8mA}{1,6mA} \Rightarrow 5$

Portanto a porta inversora pode acionar até dez dispositivos em nível alto e até cinco dispositivos em nível baixo. O sistema suporta múltiplos acionamentos, mas não existe margem para ampliar o sistema. Aqui você terá que decidir se continua ou não. Considere a possibilidade de conversar com a empresa e analisar as perspectivas futuras. Boa sorte e bom trabalho.

Faça valer a pena

1. Dependendo da tensão que se encontra na porta de entrada ou saída do dispositivo, o nível lógico é considerado 1 (alto) ou zero (baixo). A figura a seguir ilustra os níveis de tensão de entrada e de saída do CI 74LS04 da família lógica TTL.

Figura | Níveis TTL do CI 74LS04



Fonte: elaborada pelo autor.

Em relação a níveis lógicos, qual é a afirmação verdadeira?

- a) Observando a figura do texto-base, a tensão de saída entre **+2V** e **+1V** é considerada uma tensão confiável para determinar um nível lógico.
- b) O fabricante desta porta inversora especifica que a tensão de entrada (figura do texto-base – Tensão de Entrada) deve estar entre **0,8 volts** e **GND** para que o sinal corresponda ao Nível Lógico Alto e opere corretamente.
- c) Dispositivos CMOS possuem uma margem de ruído maior quando comparados a um circuito TTL, portanto circuitos CMOS são mais robustos a ruídos.
- d) Ruídos são tensões não desejadas que ocorrem somente na lógica combinacional de um circuito.
- e) Ruídos não precisam ser evitados, pois são eventos aleatórios que acontecem esporadicamente, portanto não devemos nos preocupar com eles.

2. Sabemos que estamos utilizando um circuito da família TTL quando as séries que marcam o circuito começam com o número "54" e "74", sendo o primeiro para uso militar e o segundo para uso civil/comercial. Então, sempre que circuito começar com o número "74" podemos associar à família TTL.

Com relação aos circuitos TTL, podemos afirmar:

- a) Circuitos TTL (Transistor–Transistor Logic) são dispositivos pertencentes a uma família de circuitos digitais construídos com transistores bipolares (BJTs) e resistores.
- b) A principal característica de uma família TTL é o fato da alimentação estar entre +15/-15 volts;
- c) Circuitos TTL são divididos somente em standards e *low powers*.
- d) Circuitos **54Lx/74Lx** são do tipo *high power*, apresentando alto consumo de energia.

Já circuitos **54Hx/74Hx** são do tipo *low speed* e apresentam baixa taxa de dados.

- e) Em um projeto que utilize circuitos TTL, é importante prestar atenção às correntes que são requeridas tanto nas entradas quanto nas saídas. Já a fonte de alimentação de um circuito TTL não influencia nos valores de corrente e cumpre somente o papel de fornecer a tensão de referência do sistema.

3. Circuitos baseados na tecnologia MOS são construídos utilizando um composto químico chamado Óxido Semicondutor Metálico. Com esse composto é possível criar os chamados transistores de efeito de campo de semicondutor de óxido metálico, mais conhecidos como MOSFETs.

Com relação aos circuitos MOS e CMOS, podemos afirmar:

- a) Circuitos MOS apresentam uma arquitetura mais complexa quando comparados com circuitos TTL, portanto ocupam mais espaço em uma pastilha de silício.
- b) Circuitos à base de MOS ainda podem ser divididos em NMOS e PMOS.
- c) Tecnologia CMOS ou Óxido Semicondutor Metálico com Simetria Complementar: esse tipo de dispositivo utiliza somente transistores MOS com CANAL N interconectados entre si.
- d) Existe desvantagem em utilizar esses dispositivos, pois (i) consomem pouca energia; (ii) têm baixo custo; (iii) apresentam projeto simples; (iv) têm níveis lógicos amplos na saída; e (v) uma significativa margem de ruído.
- e) Circuitos CMOS são mais rápidos quando comparados com circuitos bipolares.

Seção 4.2

Aplicação de circuitos integrados

Diálogo aberto

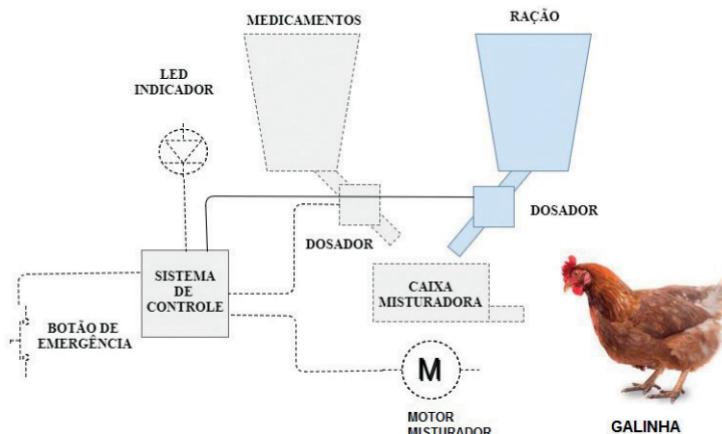
Caro aluno, na seção anterior trabalhamos conceitos pontuais sobre a especificação de circuitos integrados. Aprofundamos nossos conhecimentos sobre circuitos digitais TTL e CMOS e concluímos a importância desses dispositivos, pois são a base de qualquer conhecimento sobre projetos digitais. Ainda na seção anterior, foi apresentado o conceito sobre os níveis lógicos que esses dispositivos podem assumir, bem como os detalhes construtivos e funcionais das famílias TTL e CMOS.

Agora iremos dar mais um passo. Temos que ter em mente que circuitos TTL e CMOS são circuitos simples e robustos. Além disso, soluções que necessitam dessas características utilizam esses circuitos em conjunto com outros dispositivos. Dessa forma, quando um circuito TTL ou CMOS necessita trocar informação com outros dispositivos, existe a necessidade de pensarmos em um método para que essas informações cheguem ao destino de forma correta. Para isso, é necessário criar um cenário propício – isso é chamado de interfaceamento. Dispositivos como relés, solenoides, LEDs ou botões necessitam dos meios corretos para representarem corretamente nível lógico alto e nível lógico baixo.

Nesse sentido, considere o seguinte cenário: sua empresa já está bem consolidada no mercado, e você presume que outras empresas podem alavancar mais as suas vendas. Então você analisou a oportunidade de investimento em empresas iniciantes de uma *startup*. Você sabe que esse perfil de empresas pode, no futuro, render milhões, caso a ideia seja bem aceita no mercado. Você avalia investir em uma empresa que atua na área agrícola, e sua solução de automação a princípio gera uma economia substancial de ração para frangos. Avaliando o potencial da equipe, você sugere que juntamente com a alimentação seja distribuído um medicamento que auxilia no tratamento da gripe aviária. Para isso, a empresa terá que acrescentar ao sistema um mecanismo que, em conjunto com

a ração, mistura o medicamento.

Figura 4.14 | Sistema de mistura de ração de aves



Fonte: elaborada pelo autor.

Existem inúmeras soluções possíveis, considerando que a empresa não pretende refazer todo o sistema, mas somente adicionar um dispositivo que já possui um dosador preciso de medicamento. Para começar, considere que o dispositivo dosador possa ser acionado por um relé 12v/10A. Dessa forma, desenvolva a interface que aciona o motor da caixa misturadora de ração através do sistema de controle. Sabemos que o motor é acionado com +12 volts (DC). Não esqueça de acrescentar ao sistema de controle um LED indicando o funcionamento do dosador e um botão de interrupção de emergência para interromper o funcionamento do sistema caso ocorra algum problema. Para auxiliá-lo neste projeto, iremos aprender mais sobre como fazer o correto interfaceamento desses dispositivos. Elabore um relatório com a solução e as interfaces escolhidas.

Não pode faltar

Caro aluno, vamos relembrar que na seção anterior pontuamos conceitos sobre especificação de circuitos integrado TTL e CMOS. Percebemos que esses dispositivos são a base para projetos digitais. Por fim, entendemos como identificar os níveis lógicos que esses dispositivos podem assumir, seus detalhes construtivos e funcionais.

Vamos agora compreender como esses circuitos integrados digitais das mesmas famílias lógicas são desenvolvidos para interfacear entre si de forma prática e fácil.



Vocabulário

A língua portuguesa é muito dinâmica. Existem algumas palavras que são adotadas de outros idiomas ou derivadas de outras palavras para criar um novo significado. Segundo o dicionário on-line Priberam da Língua Portuguesa, a palavra “interface” significa:

substantivo feminino

1. [Informática] Dispositivo (material e lógico) graças ao qual se efetuam as trocas de informações entre dois sistemas.
2. [Didático] Limite comum a dois sistemas ou duas unidades que permite troca de informações.
3. [Por extensão] Interlocutor privilegiado entre dois serviços, duas empresas, etc.(DICIONÁRIO PRIBERAM, disponível em <<https://www.priberam.pt/dlpo/interface>>. Acesso em: 06 abr. 2018).

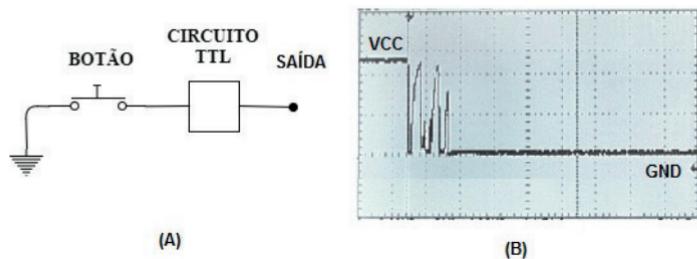
Nesta seção vamos derivar a palavra interface em outras duas: (i) interfacear e (ii) interfaceamento. No presente texto, essas palavras têm o significado de “Dispositivo (material e lógico) graças ao qual se efetuam as trocas de informações entre dois sistemas.”

Por exemplo: as portas de saída de um circuito integrado da família TTL 74LS00 podem facilmente se conectar com as portas de entrada de um circuito da mesma família TTL como o 74LS04, observando os limites de FAN-IN e FAN-OUT. Contudo, o interfaceamento utilizando famílias lógicas digitais e dispositivos corriqueiros como botões, LEDs, relês ou mesmo outras famílias, necessita de um pouco mais de cuidado e atenção. Essa unidade trata de algumas formas de interfaceamento entre circuitos digitais e esses dispositivos.

Emprego de circuitos digitais MOS e CMOS com chaves

A forma mais simples de inserir informação em qualquer sistema é através de botões. Como exemplos temos o teclado do computador, o botão do controle remoto do portão ou o botão de acionamento do elevador. Uma interface simples como chave é ilustrada na Figura 4.15. Essa interface é conhecida como ativa em baixo. Observe que quando você aperta o botão da Figura 4.14, a entrada do circuito TTL será aterrada, ou seja, o nível lógico será baixo. Já no momento em que a chave da Figura 4.15 (A) for liberada, o circuito TTL irá “flutuar”, ou seja, oscilar em torno do nível lógico baixo (Figura 4.15 (B)). Isso ocorre devido à arquitetura coletor-aberto do circuito TTL. De qualquer forma, temos que ter em mente que entradas TTL que oscilam não são confiáveis, tendo em vista que o valor é indeterminado.

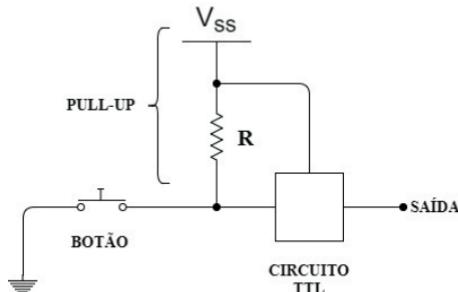
Figura 4.15 | Interface simples chave ativo em baixo com entrada coletor-aberto (A) e exemplo gráfico de um sinal oscilando (B)



Fonte: elaborada pelo autor.

A solução encontrada para corrigir esse problema de indeterminação em uma entrada TTL é acrescentar um resistor pull-up entre a chave e o circuito TTL. Esse resistor tem por objetivo elevar a tensão para V_{CC} quando a chave estiver aberta. A Figura 4.16 mostra essa pequena melhoria no circuito.

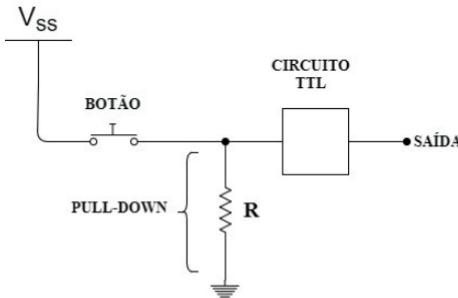
Figura 4.16 | Interface simples chave ativo em baixo com pull-up



Fonte: elaborada pelo autor.

Outro circuito comum utilizado como chave é o circuito ativo em alta. Nesse circuito, quando o botão é acionado, a tensão para V_{CC} é aplicada diretamente na entrada do circuito TTL. No momento que a chave é liberada, a entrada é levada para o nível lógico baixo por um resistor de pull-down. Os resistores de pull-down devem ser calculados observando a folha de dados do componente para não extrapolar os limites de corrente do circuito TTL. Uma chave ativa em alta é ilustrada na Figura 4.17.

Figura 4.17 | Interface simples chave ativa em alta com pull-down



Fonte: elaborada pelo autor.



Exemplificando

Considerando as características elétricas de uma porta lógica inversora padrão como os descritos na Figura 4.18, é possível calcular o resistor R de pull-down na entrada de um circuito chave ativa em alta cujo V_{CC} é +5 volts.

Figura 4.18 | Características elétricas de uma porta lógica padrão

PARAMETER	TEST CONDITIONS‡	SN5404			SN7404			UNIT
		MIN	TYP\$	MAX	MIN	TYP\$	MAX	
V_{IK}	$V_{CC} = \text{MIN}, I_I = -12 \text{ mA}$			-1.5			-1.5	V
V_{OH}	$V_{CC} = \text{MIN}, V_{IL} = 0.8 \text{ V}, I_{OH} = -0.4 \text{ mA}$	2.4	3.4		2.4	3.4		V
V_{OL}	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, I_{OL} = 16 \text{ mA}$		0.2	0.4		0.2	0.4	V
I_I	$V_{CC} = \text{MAX}, V_I = 5.5 \text{ V}$			1			1	mA
I_{IH}	$V_{CC} = \text{MAX}, V_I = 2.4 \text{ V}$			40			40	μA
I_{IL}	$V_{CC} = \text{MAX}, V_I = 0.4 \text{ V}$			-1.6			-1.6	mA
I_{OS}	$V_{CC} = \text{MAX}$	-20		-55	-18		-55	mA
I_{CCH}	$V_{CC} = \text{MAX}, V_I = 0 \text{ V}$		6	12		6	12	mA
I_{CCL}	$V_{CC} = \text{MAX}, V_I = 4.5 \text{ V}$		18	33		18	33	mA

Fonte: elaborada pelo autor.

Considerando os seguintes parâmetros:

- $V_{CC} +5$ volts.
- Corrente máxima na entrada para nível lógico baixo: - 1.6 mA.

Aplicando a lei de Ohm é possível obter o valor do resistor:

$$U = R \times i \Rightarrow 5 = R \times 1.6 \text{ mA} \therefore R = 312.5 \Omega$$

Observe que o sinal negativo da corrente significa que ela sai do circuito, portanto é desconsiderada no cálculo. Comercialmente, os resistores mais próximos ao valor calculado são de 330Ω , sendo então a melhor escolha para esse circuito.

Emprego de circuitos digitais MOS e CMOS com LEDs

A maioria dos sistemas eletrônicos, se não a sua totalidade, apresentam algum indicador de funcionamento como LEDs (do inglês *light emitting diode* ou "diodo emissor de luz"). Um LED é um excelente mecanismo indicativo, pois utiliza níveis de tensão e corrente reduzidos.



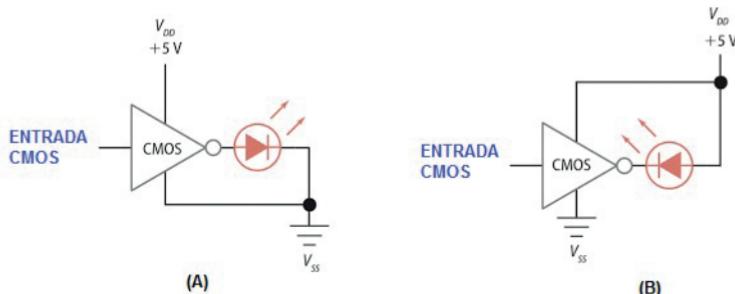
Assimile

Existem diversos tipos de LEDs no mercado, e quando se pensa em um projeto que envolva LEDs é importante considerar os seguintes aspectos: (i) a fonte de alimentação, que deve ser confiável e estável para evitar retrabalho no projeto; (ii) o material ótico usado, pois LEDs utilizados em iluminação são envolvidos por uma resina ou vidro, e esse material deve ser de boa qualidade porque pode sofrer envelhecimento e desgastes antes do tempo; (iii) bons dissipadores e estrutura de

fixação e proteção, estruturas que devem ser robustas para aguentar altas temperaturas, sol e chuva, ambientes corrosivos, raios ultravioleta e poeira, entre outras intempéries; e, por fim, (iv) potência do LED em watts adequada, pois cada LED é utilizado para um tipo de iluminação e, portanto, apresentam seus respectivos consumos. Um LED de baixo consumo, por exemplo, drena uma corrente de aproximadamente 20 a 30 mA, quando alimentado com 2 volts.

O uso de circuitos CMOS com LEDs indicadores é simples. Quando o circuito CMOS é alimentado com +5 volts, o próprio circuito limita a corrente, não havendo a necessidade de um resistor em série para limitar a corrente. Na Figura 4.19 (a) é possível observar que quando a saída do inverter estiver em alta, o LED acende, acontecendo o contrário na Figura 4.19 (b), quando a saída do inverter estiver em baixa, o LED acende.

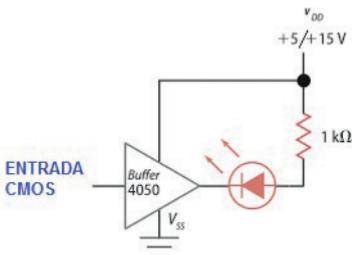
Figura 4.19 | Interface CMOS +5 volts e LED



Fonte: elaborada pelo autor.

Quando a alimentação do circuito CMOS for maior, entre +10 e +15 volts, por exemplo, existe a necessidade de um resistor em série entre o circuito e o LED para limitar a corrente. O uso de circuitos da série 4000, como os buffers CI 4049 e CI 4050, pode ser uma boa opção quando existe a necessidade de saídas com mais correntes (Figura 4.20). O buffer 4050, por exemplo, apresenta V_{dd} de +5 a +15 volts e corrente de I_{OH} 4,7 mA (corrente na saída em alta) e I_{OL} 22 mA (corrente na saída em low).

Figura 4.20 | Interface CMOS como buffer e LED

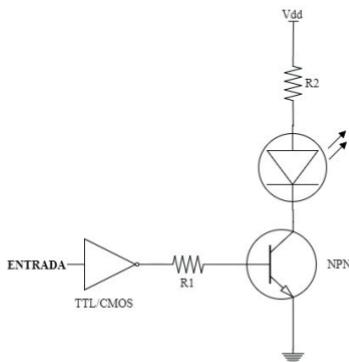


Fonte: elaborada pelo autor.

A mesma análise pode ser utilizada para circuito da família TTL, bastando observar os limites de tensão e alimentação, e calculando os valores dos resistores para limitar as correntes drenadas.

Uma solução mais elegante utiliza um transistor para isolar o circuito, tanto TTL quanto CMOS. Dessa forma, caso o LED sofra alguma sobretensão e venha a queimar, basta analisar a saída do transistor para detectar o problema. Essa solução é apresentada na Figura 4.21 e utiliza um transistor NPN como chave.

Figura 4.21 | Interfaceamento com circuito TTL ou CMOS e transistor



Fonte: elaborada pelo autor.

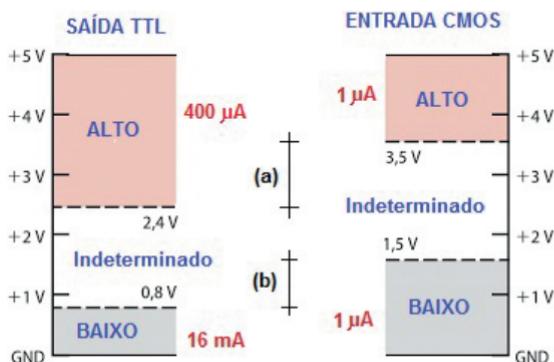
Os resistores R_1 e R_2 limitam a corrente e devem ser calculados considerando o limite tanto do circuito TTL ou CMOS quanto os limites do transistor NPN.

Interfaceamento entre CIs TTL e CMOS

Interfacear um circuito TTL com um circuito CMOS não acontece

de forma direta. Cada dispositivo apresenta níveis de tensão e corrente distintos. Essas diferenças fazem com que os níveis lógicos não sejam compatíveis. Um circuito TTL padrão, por exemplo, opera entre +5 volts e +2,4 volts e corrente de 400 μ A para o nível lógico alto como saída. Contudo, um circuito CMOS padrão, alimentado em +5 volts, opera entre +5 volts e +3,5 volts e corrente de 1 μ A para o nível lógico alto como entrada. Essa diferença fica mais clara observando a Figura 4.22 (a) entre as duas tensões na faixa alta e a diferença na Figura 4.22 (b) entre as duas tensões na faixa baixa.

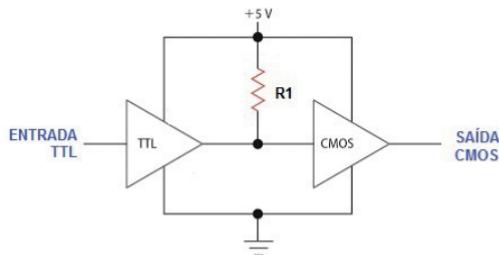
Figura 4.22 | Comparativo dos níveis de tensão e corrente entre um circuito TTL e um circuito CMOS



Fonte: elaborada pelo autor.

Essas diferenças de faixas fazem com que o circuito necessite de um resistor de pull-up, no caso $R1=1,5\text{ k}\Omega$, para elevar a tensão de saída do circuito TTL (Figura 4.23).

Figura 4.23 | Interfaceamento entre um circuito TTL e um circuito CMOS utilizando um resistor de pull-up para elevar a tensão de saída TTL



Fonte: elaborada pelo autor.

Outra forma de resolver é utilizando circuitos da família CMOS **74HCTxx**. Estes circuitos apresentam alta velocidade e são compatíveis diretamente com saídas TTL, como é o caso da porta inversora 74HCT34. Agora, caro aluno, existe um porém: circuitos CMOS são alimentados com níveis de tensão até +15 volts. Até o momento, consideramos os circuitos CMOS sendo alimentados com +5 volts. Então como podemos interfacear um circuito TTL com um circuito CMOS com alimentações distintas? E como podemos interfacear a saída de um circuito CMOS com a entrada de um circuito TTL? Para responder à primeira pergunta, precisamos considerar que entre o circuito TTL e o circuito CMOS devemos subir a tensão. Podemos fazer isso de duas formas: a primeira, utilizando um transistor NPN com um resistor no coletor do transistor para elevar a tensão da saída TTL (Figura 4.24 (a)). A segunda forma é utilizando um *buffer* coletor aberto para converter tensão TTL para CMOS, e podemos, nesse caso, usar circuitos como os SN7406 ou SN7416 (Figura 4.24 (b)) com um resistor de pull-up na saída do buffer para elevar a tensão do sinal.

Figura 4.24 | Interfaceamento circuito TTL com um circuito CMOS com diferentes alimentações



Fonte: elaborada pelo autor.

Para responder à segunda pergunta, precisamos considerar que o circuito TTL tem uma tensão menor que o circuito CMOS. Portanto, entre a saída do circuito CMOS e a entrada do circuito TTL devemos utilizar um *buffer* para abaixar a tensão, como os circuitos da família CMOS CD4049 e CD4050. Esses circuitos são apropriados para essa conversão lógica e não necessitam de resistores de pull-up.



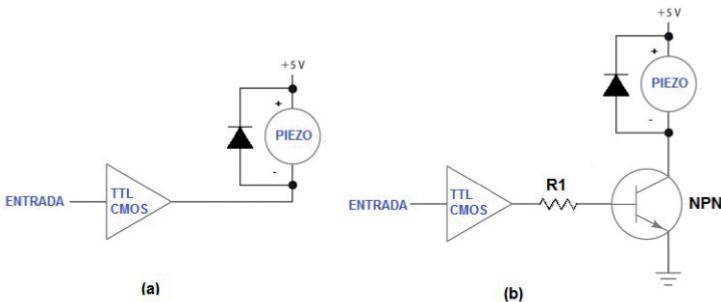
Circuitos da série CMOS FACT possuem características de corrente de saída muito interessantes. Esses circuitos podem acionar diretamente circuitos CMOS, TTL, PMOS e NMOS. Portanto facilitam muito a vida do projetista. Para saber mais sobre eles, investigue a folha de dados do circuito 74AC04:

FAIRCHILD Semiconductor. **74AC04, 74ACT04 Hex Inverter**. Fairchild Semiconductor, 2007. 10 p. Disponível em: <<http://www.onsemi.com/pub/Collateral/74ACT04-D.pdf>>. Acesso em: 12 jan. 2018.

Emprego de circuitos digitais com campainhas, relés, motores e solenoides

Sistemas digitais muitas vezes são empregados para resolver problemas simples como o acionamento de uma campainha, relés, motores ou mesmo um solenoide. Esses sistemas apresentam um interfaceamento simples. O acionamento de uma Campainha Piezoelétrica (cujo consumo varia entre 3 e 5 mA quando acionada), por exemplo, pode utilizar um circuito CMOS ou TTL diretamente, bastando um diodo em paralelo com a Campainha Piezoelétrica como proteção contra tensões reversas (Figura 4.25(a)). Outra forma de acionamento de uma Campainha Piezoelétrica é usando um transistor NPN como chave para isolar o sistema (Figura 4.25(b)).

Figura 4.25 | Interfaceamento TTL ou CMOS com uma campainha piezoelétrica

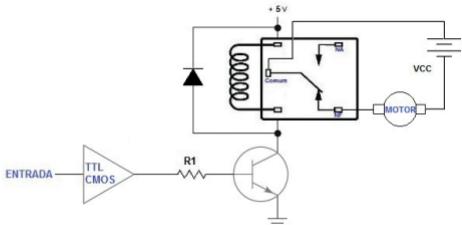


Fonte: elaborada pelo autor.

Já o relés é muito empregado em sistemas que requerem isolamento entre alta tensão e um circuito TTL ou CMOS. O

emprego de um diodo também é necessário para evitar picos de tensão induzida. A Figura 4.26 ilustra o interfaceamento de um circuito TTL ou CMOS com um relé; já o relé, por sua vez, aciona um motor DC.

Figura 4.26 | Interfaceamento circuito TTL ou CMOS com um relé e um motor DC



Fonte: elaborada pelo autor.

Quando a entrada TTL ou CMOS é acionada, o transistor começa a condução, fazendo com que a bobina do relé seja induzida, acionando mecanicamente a chave comum do relé. Por sua vez, o relé fecha o circuito e o motor é acionado – é assim que o relé produz isolamento e aciona um motor. O mesmo circuito pode ser utilizado para acionar um solenoide. Para isso, basta substituir o relé pelo solenoide, mantendo o diodo de tensão reversa.



Refletia

Quais são as falhas mais comuns em um circuito digital? Existe mal funcionamento interno de um circuito digital? Analisar entradas ou as saídas se estão para GND ou VCC é uma forma de verificação? Entradas e saída abertas são possíveis erros? E o curto-circuito entre dois pinos, pode ser uma falha?

Caro aluno, o assunto não se esgota aqui. Todos os mecanismos ilustrados nessa unidade podem apresentar variações ou mesmo ser melhorados. É importante salientar que circuitos TTL e CMOS são circuitos que, mesmo com o passar dos anos, ainda são soluções utilizadas por conta da simplicidade e da robustez que possuem. Pesquise mais sobre o assunto e coloque em prática o que você aprendeu. Bons estudos.

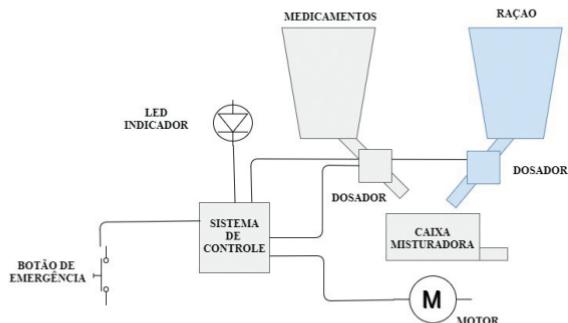
Sem medo de errar

Caro aluno, vamos relembrar: a sua empresa já está bem consolidada no mercado. Você considera que outras empresas podem alavancar mais as suas vendas. Avaliou, então, a oportunidade de investimento em empresas iniciantes em uma startup. É o caso de uma empresa que atua na área agrícola, e sua solução de automação, a princípio, gera uma economia substancial de ração para frangos. Você então sugere que juntamente com a alimentação seja distribuído um medicamento que auxilia no tratamento da gripe aviária. Dessa forma, considerando todos os requisitos você deve, inicialmente:

1. Descobrir quais os níveis lógicos com que cada dispositivo que será acrescentado ao sistema de controle opera, ou seja, compreenda como o dosador entende nível lógico alto e nível lógico baixo. Você deve fazer isso tanto para os botões quanto para o LED.
2. Descobrir quais são as alimentações de cada sistema, por exemplo, se um relé pode ser acionado com +5 volts ou +12 volts. Essa análise é importante para a escolha do relé. Portanto, você deve ficar atento se o sistema como um todo apresenta as referidas alimentações.
3. Descobrir quais os níveis lógicos com que as portas de entrada e saída do sistema operam. Essa análise é fundamental para a escolha do modo de acionamento.
4. Verificar se existe a necessidade de isolar o sistema dos dispositivos para evitar que uma falha queime o sistema de controle. Para isso, será necessário escolher uma família de circuito digital – TTL ou CMOS. Utilize a análise anterior nessa escolha.
5. Testar a lógica separadamente do sistema de controle.

A Figura 4.27 ilustra os dispositivos interligados ao sistema de controle. Utilize as soluções propostas na unidade para isolar o sistema de controle.

Figura 4.27 | Sistema de Mistura de Ração de Aves e Remédios



Fonte: elaborada pelo autor.

Apresente em seu relatório os esquemas elétricos das interfaces, com seus respectivos cálculos de tensão, corrente e resistores de pull-up e pull-down (caso necessário). Considere que o sistema de acionamento seja baseado em um microcontrolador padrão de 8 bits, portanto, faça a escolha desse microcontrolador e proponha todo o desenvolvimento com base nessa MCU. Considere que o sistema possua uma fonte de alimentação de +5/-5 volts, +9/-9 volts e +12/-12 volts e 200 watts de potência.

Avançando na prática

Imunidade a ruído

Descrição da situação-problema

Vamos imaginar o seguinte cenário agora: o sistema de alimentação de aves que você está produzindo em conjunto com uma empresa inicial pode ser colocado muito próximo de uma rede de transmissão de energia, ou mesmo próximo a um gerador de energia (gerador CA – corrente alternada). Esses sistemas produzem campos elétricos e magnéticos parasitas que, por sua vez, induzem tensões nos fios de conexão entre os circuitos do sistema. A Tabela 4.3 especifica as tensões de entrada e saída para um circuito padrão da família TTL. Utilize esses valores para determinar:

- A maior amplitude de um pico de tensão (ruído) aceito pelo sistema quando uma saída em nível lógico alto estiver acionando uma entrada.
- A maior amplitude de um pico de tensão (ruído) aceito pelo

sistema quando uma saída em nível lógico baixo estiver acionando uma entrada.

Tabela 4.3 | Tensões de entrada e saída para um circuito padrão da família TTL

Parâmetro	Tensão Mínima (v)	Tensão Típica (v)	Tensão Máxima (v)
V_{OH}	2,5	3,5	-
V_{OL}	-	0,2	0,5
V_{IH}	2,1	-	-
V_{IL}	-	-	0,9

Fonte: elaborada pelo autor.

Resolução da situação-problema

Caro aluno, essa análise é fundamental para futuros diagnósticos do sistema. Em posse dos resultados, é possível determinar o quanto o sistema deve ser reforçado para evitar retrabalho ou mesmo a troca do produto pelo cliente.

Para responder a questão, (a) devemos considerar:

- Se a saída estiver em nível alto, ela pode estar baixa até +2,5 volts (V_{OH} mínimo). Ocorre que, se a tensão de entrada entende como nível alto mínimo o valor +2,1 volts (V_{IH} mínimo), um pico de tensão ocasionado por tensão induzida pode levar o nível lógico abaixo de +2,1 volts, caso a amplitude desse pico seja maior que:

$$V_{NH} = V_{OH}(\text{mín}) - V_{IH}(\text{min}) \rightarrow 2,5 - 2,1 = 0,4 \text{ V}$$

- Se a saída estiver em nível baixo, ela pode estar alta até +0,5 volts (V_{OL} máximo). Ocorre que, se a tensão de entrada entende como nível baixo mínimo o valor +0,91 volts (V_{IL} máximo), um pico de tensão ocasionado por tensão induzida pode levar o nível lógico acima de +0,9 volts, caso a amplitude desse pico seja maior que:

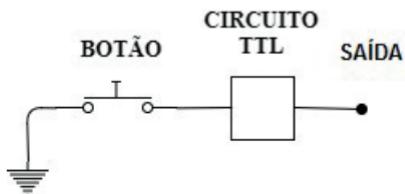
$$V_{NH} = V_{IL}(\text{máx}) - V_{OL}(\text{máx}) \rightarrow 0,9 - 0,5 = 0,4 \text{ V}$$

Em posse desses valores, fique atento para que o seu projeto não extrapole essas medidas.

Faça valer a pena

1. Uma interface simples como chave é ilustrada na figura abaixo. Essa interface é conhecida como ativa em baixo. Observe que quando você aperta o botão, a entrada do circuito TTL será aterrada, ou seja, o nível lógico será baixo. Já no momento em que a chave da figura for liberada, o circuito TTL irá “flutuar”, isto é, oscilar entre o nível lógico baixo. Isso ocorre devido à arquitetura coletor-aberto do circuito TTL. Portanto, entradas TTL que oscilam não são confiáveis, tendo em vista que o valor é indeterminado.

Figura | Interface simples chave ativo em baixo com entrada coletor-aberto



Fonte: elaborada pelo autor.

A melhor solução para mitigar esse problema é:

- a) O uso de capacitores de acoplamento em paralelo com o botão.
- b) O uso de um botão especial que não oscila devido a sua velocidade de ação.
- c) O uso de um circuito CMOS comparador em paralelo para determinar o nível lógico do botão.
- d) O uso de um resistor de pull-down.
- e) O uso de um resistor de pull-up.

2. Interfacear um circuito TTL com um circuito CMOS não acontece de forma direta. Cada dispositivo apresenta níveis de tensão e corrente distintos. Essas diferenças fazem com que os níveis lógicos não sejam compatíveis. Um circuito TTL padrão, por exemplo, opera entre +5 volts e +2,4 volts e corrente de 400 μ A para o nível lógico alto como saída. Contudo, um circuito CMOS padrão, alimentado em +5 volts, opera entre +5 volts e +3,5 volts e corrente de 1 μ A para o nível lógico alto como entrada. Observe esses níveis na figura a seguir.

Figura | Comparativo dos níveis de tensão e corrente entre um circuito TTL e um circuito CMOS



Fonte: elaborada pelo autor.

A maior amplitude de um pico de tensão (ruído) aceito pelo sistema quando uma saída em nível lógico alto estiver acionando uma entrada:

- a) A maior amplitude será: 0,9 volts.
- b) A maior amplitude será: 0,6 volts.
- c) A maior amplitude será 1,1 volts.
- d) A maior amplitude será -0,9 volts.
- e) A maior amplitude será -0,6 volts.

3. Circuitos CMOS são alimentados com níveis de tensão até +15 volts. Para interfacear um circuito TTL com um circuito CMOS com alimentações distintas, é necessário subir a tensão da saída TTL ou converter a tensão TTL para CMOS.

A solução que mais se adequa para interfacear a saída de um circuito TTL com a entrada de um circuito CMOS com diferentes alimentações é:

- a) O uso de resistores de pull-down.
- b) O uso de resistores de pull-up.
- c) O uso de capacitores de acoplamento.
- d) O uso de um transistor NPN com um resistor no coletor do transistor para elevar a tensão da saída TTL ou o uso de um buffer coletor aberto para converter tensão TTL para CMOS como os circuitos SN7406 ou SN7416, com um resistor de pull-up na saída do buffer para elevar a tensão do sinal.
- e) Não é possível esse tipo de interfaceamento.

Seção 4.3

Aplicação de circuitos integrados

Diálogo aberto

Caro aluno, nessa unidade estamos trabalhando o conhecimento acerca da especificação, do interfaceamento e da maneira de utilização de circuitos integrados das famílias TTL e CMOS. Em linhas gerais, estamos aprofundando nosso conhecimento sobre o uso de circuitos digitais MOS e CMOS como chaves e também em dispositivos como LEDs, campainhas, relés, motores e solenoides. Também entenderemos um pouco mais sobre como isolar digitalmente um circuito utilizando isoladores digitais e optoisoladores, como utilizar sensores de efeito Hall e como encontrar problemas em circuitos digitais simples. Por fim, aprofundaremos nosso saber acerca de um servomotor. A melhor forma de estimular o conhecimento de qualquer pessoa é inserindo um cenário com questões atuais e relevantes.

Considere, então, que a empresa iniciante que atua na área agrícola gerou uma economia de 20% nas rações e reduziu em 95% a mortalidade das aves com a sua solução de automação. O mercado está pedindo mais. A hora é agora, e você decidiu iniciar um novo projeto que vai revolucionar o mercado de ovos. Em uma granja moderna existem de 12 a 14 galinhas do tipo Lohmann LSL e Lohmann Brown em um sistema confinamento em gaiola vertical. Ocorre que cada galinha apresenta um ciclo de crescimento, desenvolvimento, pré-postura e início de postura. No ciclo de postura, a produção aumenta rapidamente até atingir um pico na 28 e 32 semanas de vida da galinha. A partir daí, a postura começa a decair em média de 6 a 9% a cada 10 semanas até o final do ciclo de postura da galinha. Por exemplo: em um lote com 100 galinhas, são recolhidos 90 ovos por dia. Se a produção começar a ficar abaixo de 50 a 60%, significa que esse lote (gaiola) é antieconómico e deve ser descartado. Sabendo disso, você desenvolveu um método utilizando circuitos digitais, para identificar qual galinha não está mais botando ovo, podendo ser substituída por outra. Para isso,

você vai utilizar, além do sistema que verifica qual galinha não está mais na postura, um servomotor para separar a galinha improdutiva e um sistema de comporta acionado por relé para abrir um alçapão.

Algumas questões você deve tratar com a equipe de desenvolvimento. Como você isolará o sistema digital (cujo custo é altíssimo) que atua com nível lógico TTL do sistema de acionamento do alçapão que apresenta uma tensão maior? Como será a integração com o servo motor? O sistema utiliza um sensor Hall para identificar se a galinha colocou o ovo; como você ampliará o sinal? Essas são algumas questões desafiadoras que a nova empresa vai responder para ter sucesso. De posse de todas as informações, escreva um relatório respondendo as questões acima. Não esqueça de fazer uma tabela relacionando os sinais e suas funções. Tenha como referência um microcontrolador de 8 bits de uso geral. Utilize o mais barato que você puder e justifique com uma tabela essa comparação. Parabéns por ter investido em uma empresa nascente! Bons empresários sabem o valor de boas ideias. O mérito de todo o sucesso é seu, e saiba que para realizar coisas grandes, comece pequeno e não tenha medo de errar. Desafie-se um pouco a cada dia. Boa sorte e sucesso hoje e sempre.

Não pode faltar

Caro aluno, na seção anterior focamos nosso aprendizado nos circuitos TTL e CMOS e nas soluções que necessitam desses dispositivos. Descobrimos que quando um circuito TTL ou CMOS necessita trocar informação com outros dispositivos, existe a necessidade de pensarmos em um método para que essas informações cheguem ao destino de forma correta, ao qual chamamos de interfaceamento. Portanto aprendemos que dispositivos como relés, solenoides, LEDs ou botões necessitam dos meios corretos para representar corretamente nível lógico alto e nível lógico baixo.

Nessa seção, iremos mais a fundo no conhecimento sobre interfaceamento. Inicialmente, compreenderemos como isolar um circuito digital que opera a baixa tensão de sistemas, os quais operam com tensões e correntes elevadas, utilizando alguns dispositivos como optoisoladores e relés. Contudo, relés são dispositivos que possuem partes mecânicas que eventualmente podem gerar picos

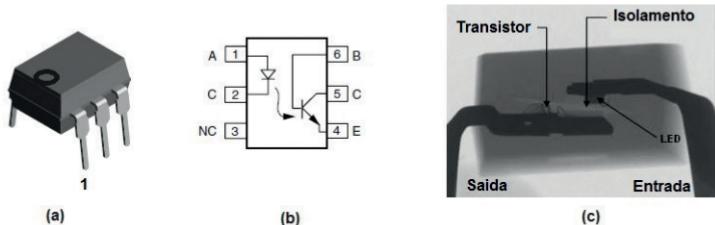
de tensão devido à indutância da bobina utilizada na abertura e fechamento do relé. Além disso, com o passar do tempo, esse mecanismo de abertura e fechamento tende a perder as suas propriedades mecânicas, ocasionando o mau funcionamento do relé. Uma solução encontrada pela indústria para resolver esse problema é o emprego de isoladores digitais ou optoisoladores. Esses isoladores digitais e óticos são utilizados, por exemplo, em lâmpadas frias, para isolar o acionamento digital dos sistemas de alta tensão.

Isolador digital versus optoisoladores

Segundo Tocci (2012), optoisoladores consistem em uma solução de isolamento de sinal, incontestável há mais de quatro décadas. Contudo, isoladores digitais fabricados em tecnologia de processo semicondutor de óxido metálico complementar (CMOS) estão ganhando força na indústria devido ao seu bom desempenho, a sua confiabilidade e ao seu nível de integração com outros sistemas. Circuitos optoisoladores digitais estão disponíveis em diversas versões. Esses sistemas estão disponíveis desde de dispositivos de quatro pinos ao custo de centavos de dólares, até dispositivos que custam algumas dezenas de dólares e atuam em sistemas que requerem alta velocidade.

Optoisoladores comuns e com preços mais acessíveis são em geral dispositivos de seis pinos baseados em LED. Esses dispositivos exigem que o LED na entrada seja ativado com uma corrente de aproximadamente 10 a 30 mA. Uma vez ativado o LED, a luz emitida sensibiliza o fototransistor dentro do dispositivo, que, por sua vez, começa a conduzir, levando saída para nível lógico alto. Analisando a Figura 4.28, conseguimos observar o diagrama de pinos (Figura 4.28 (a)), o circuito (Figura 4.28(b)) e, por fim, uma foto em corte, tirada utilizando um raio-X de um optoisolador (Figura 4.28(c)).

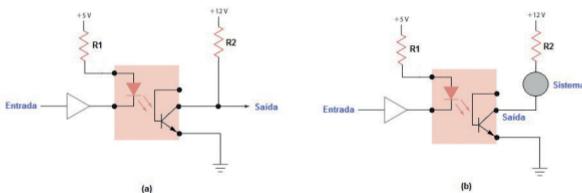
Figura 4.28 | Diagrama de pinos, circuito elétrico e foto de um optoisolador



Fonte: adaptado de Silicon Labs (2017).

Observe na Figura 4.28 (a) o pino 1, que sinaliza a entrada do LED. Em (b), veja como o circuito é distribuído em um encapsulamento DIP de seis pinos, e, por fim, em (c), observe o espaço entre o LED e o fototransistor. Fica mais claro o emprego de um optoisolador na Figura 4.29.

Figura 4.29 | Sistema com Optoisolador

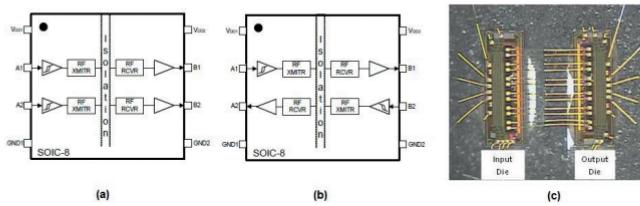


Fonte: elaborada pelo autor.

Observe na Figura 4.29 (a) as diferentes tensões em R1(+5 volts) e em R2 (+12 volts) e na Figura 4.29 (b) a inclusão de um sistema que opera a +12 volts.

Isoladores digitais baseados em CMOS apresentam o funcionamento análogo aos optoisoladores. Contudo, isoladores digitais são superiores em desempenho, em confiabilidade e nas taxas de transmissão com isolamento faixa dos kVolts, pois são baseados em isolamento capacitivo (Figura 4.30). Já optoisoladores são limitados, pois utilizam tecnologia mais antiga, baseada em LEDs. Nesse sentido, soluções baseadas em LEDs podem sofrer oscilações de corrente devido à variação da temperatura no sistema e devido ao processo de envelhecimento do LED. Esses fatores reduzem, portanto, a confiabilidade dos optoisoladores.

Figura 4.30 | Diagrama de pinos e circuito de um isolador digital



Fonte: adaptada de Silicon Labs (2017).

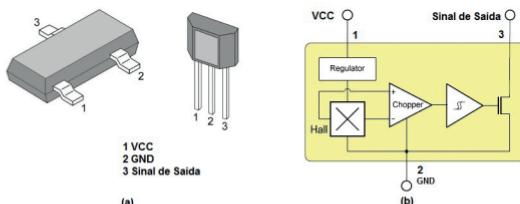
Observe na Figura 4.30 (a) e (b) o isolamento entre os circuitos, baseado em capacitores CMOS. Além disso, veja que um mesmo dispositivo possui dois canais, tanto em um sentido quanto em outro. Por fim, a Figura 4.30 (c) ilustra como é a arquitetura interna da tecnologia de um isolador digital CMOS. Sempre analise a folha de dados desses componentes para obter mais detalhes sobre as características elétricas e sobre o funcionamento.

Utilizando sensores de efeito Hall

Caro aluno, vamos compreender detalhes sobre o sensor de efeito Hall. Esse sensor é amplamente empregado em aplicações como sistemas de ignição de automóveis, impressoras, detectores de posição e comutadores de motores CC, entre outras aplicações.

Um sensor de efeito Hall básico consiste em um dispositivo semicondutor, cuja polarização proporciona uma circulação de corrente de polarização no sensor, como mostra a Figura 4.31. Ocorre que a tensão de polarização é proporcional ao campo magnético aplicado no sensor. Portanto, se o campo magnético próximo ao sensor for nulo, a saída do sensor será nula. Caso o campo varie, a saída do sensor varia. É simples assim. Contudo, como a saída do sensor é muito baixa, o sinal deve ser amplificado.

Figura 4.31 | Definição dos pinos e diagrama funcional do sensor de efeito Hall



Fonte: elaborada pelo autor.

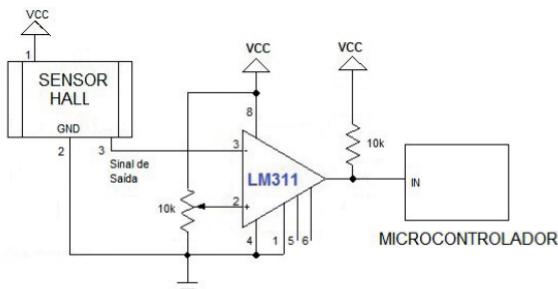
Observe na Figura 4.31 (a) os diferentes encapsulamentos possíveis em um sensor de efeito Hall e as suas respectivas distribuições de pinos. Já na Figura 4.31 (b) é descrito o diagrama funcional desse sensor, integrando um regulador de tensão, o sensor Hall, um comparador para anular offset dinâmico e, por fim, um Schmitt Trigger.



Exemplificando

Um uso muito interessante para esse sensor está na medição de radiação, como por exemplo em salas de raio-X de hospitais. Um sistema simples para medição de campo magnético em salas hospitalares pode ser montado utilizando um amplificador comparador LM311, um sensor Hall e um microcontrolador, como o descrito na Figura 4.32.

Figura 4.32 | Radiômetro utilizando sensor de efeito Hall



Fonte: elaborada pelo autor.

Observe na Figura 4.32 que a saída do comparador pode ser ligada diretamente no microcontrolador caso VCC seja + 5 volts.

Uma vez recebido o sinal pela entrada analógica do microcontrolador, é necessário convertê-lo de analógico para digital e exibi-lo em um mostrador.

Sensores de efeito Hall têm vantagens: são imunes à lama, poeira e água, quando bem acondicionados. São fáceis de ser fixados e apresentam um alcance maior na detecção de objetos. Contudo, como desvantagem, é necessário amplificar o sinal através de outros circuitos (mais circuitos, maior o custo, etc) e podem sofrer ação de outros campos magnéticos, aumentando ou diminuindo o campo magnético e gerando imprecisão na medida.

Encontrando problemas em circuitos digitais simples

Caro aluno, circuitos digitais podem dar problemas. Uma parcela desses problemas pode ser diagnosticada com uma inspeção visual, ou seja, com um pouco de calma é possível observar qualquer anormalidade no circuito. Essas anormalidades podem ser identificadas visualmente procurando no sistema por componentes inflados (como se fossem explodir) ou rachados, componentes com aparência de queimados ou mesmo a placa de circuito impresso queimada ou empenada, fios ou fusíveis rompidos. Toda e qualquer anormalidade deve ser investigada. Para isso, é preciso ter em mente que devemos conhecer um pouco do sistema. Isso é possível com o que chamamos de diagrama esquemático, o qual é possível observar na Figura 4.33. Um esquemático nada mais é que um diagrama com os componentes e as ligações entre esses componentes. O entendimento e a leitura de um esquemático são fundamentais para iniciarmos qualquer etapa de manutenção de um sistema. Caso você não possua o esquemático, é possível obter informações de cada componente buscando as folhas de dados de cada componente. Essa abordagem é mais demorada, mas é muito comum.

De posse dessas informações, a próxima etapa é usarmos um multímetro, também conhecido como multímetro, para analisarmos a alimentação e o plano terra de cada componente.



Assimile

Como é possível analisar uma falha em um circuito?

Primeiramente, você deve observar alguns passos:

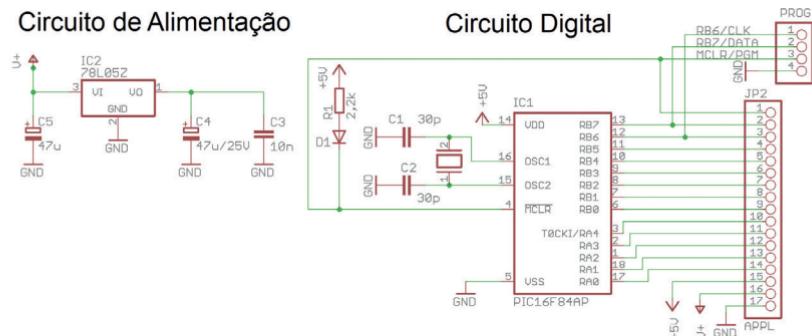
1. Garantir que o circuito está sendo alimentado com a tensão correta, e, para isso, você deve analisar a fonte de alimentação com o multímetro. Observe se o valor da saída da fonte corresponde ao valor necessário para alimentar o circuito digital.
2. Analise se a alimentação está chegando a cada circuito corretamente.
3. Com o multímetro medindo continuidade, analise se o plano terra não está rompido.
4. Com o multímetro medindo continuidade, analise se o plano terra não está ligado à alimentação. Caso isso esteja ocorrendo, temos um curto circuito entre o plano terra e o V_{CC} . A origem deve ser investigada.

5. Agora utilizando um osciloscópio, analise as portas do circuito digital. Para isso, você deve saber qual o sinal esperado na saída. Por exemplo: caso você esteja analisando a saída de um sinal serial, você deve observar uma forma de onda entre o nível lógico zero e o nível lógico um, com os dados esperados.

6. Com o osciloscópio, não deixe de analisar o circuito gerador do sinal de relógio. O sinal de relógio é fundamental para o funcionamento do circuito.

Caso você encontre algum problema, retire o componente e substitua por um novo. Se possível, analise o componente retirado para identificar o problema. Com o diagnóstico do problema, você pode saber como ele foi originado e corrigi-lo em uma próxima etapa de projeto.

Figura 4.33 | Diagrama esquemático de um sistema digital



Fonte: elaborada pelo autor.

Analisar um circuito é uma tarefa trabalhosa. Existem diversos métodos e ferramentas de depuração de sistemas, portanto não se prenda somente ao que foi explicado aqui. converse com pessoas mais experientes e tenha em mente que só com muita prática é possível encontrar problemas em sistemas.

Circuitos digitais e servomotores

Caro aluno, nessa seção trataremos sobre servomotores e o seu controle através de um circuito digital padrão. Vamos começar entendendo um pouco sobre o motor de corrente contínua (motor CC). O principal aspecto de um motor CC é que o sentido da rotação é definido pelo sentido da corrente. Podemos verificar isso de forma bem simples: basta desligarmos o motor CC, invertermos os pinos

de alimentação e ligarmos novamente o motor CC, e o sentido de rotação do motor é invertido. Essa sequência não é prática. Menos prático e mais complicado é se precisarmos controlar a velocidade e posição de um motor CC de forma precisa. Portanto, o motor CC sem um controle não é o mais adequado para esses requisitos.

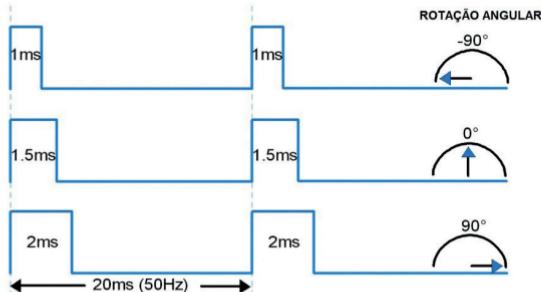
Para Tocci (2012), um servomotor é a solução encontrada para resolver esse problema de controle de velocidade e posição. Um servomotor consiste em um (i) Sistema Atuador – motor elétrico (CC ou AC) e engrenagens; (ii) Sensor – indica a posição angular do eixo do motor; e, por fim, um (iii) Circuito de Controle – controle em malha fechada. Alguns servomotores podem girar somente até 180 graus para ambos os lados, outros, porém, têm liberdade de 360 graus, possibilitando giros completos. Servomotores apresentam uma interface simples com três fios: dois fios de alimentação (VCC e GND) e um fio de controle.

É utilizado para controlar o servomotor o protocolo PWM (Modulação por Largura de Pulso), sendo que a largura determina a posição que o eixo irá se deslocar (SG90, 2017):

- Largura máxima – eixo rotaciona +90 graus com referência ao eixo central.
- Largura mínima – eixo rotaciona -90 graus com referência ao eixo central.
- Demais larguras – o eixo adota posição proporcional à largura.

Todo sinal de controle do servomotor, em geral, é repetido a cada 50 Hz (ou 20ms), sendo a largura do pulso entre 1 e 2ms. Observe na Figura 4.34 o sinal de controle utilizado em um servomotor.

Figura 4.34 | Sinal de controle em um servomotor



Fonte: elaborada pelo autor.

Um termo muito empregado em sinais PWM, que se refere ao tempo total em que o sinal permanece em nível lógico alto ou estado ativo durante um ciclo, é o chamado Ciclo de Trabalho ou *Duty Cycle*. Um *duty cycle* é expresso em porcentagem e pode ser calculado utilizando a seguinte formula: $duty_cycle = \frac{\tau}{T}$; sendo τ o intervalo de tempo em que o sinal permanece em estado ativo e T o tempo total do ciclo.



Refletia

Você é capaz de pensar em aplicações para um servomotor no seu dia a dia? Seria possível gerar um sinal PWM utilizando um microcontrolador de 8 bits de uso geral?

Por fim, servomotores são amplamente utilizados em robótica, aeromodelismo e automodelismo, entre outros sistemas que necessitam de um controle preciso do movimento.



Pesquise mais

Caro aluno, esperamos que tenha gostado do conteúdo aqui apresentado. Essa jornada pelo mundo dos circuitos digitais é só o começo; o assunto não se esgota aqui. Todo o conteúdo apresentado marca o início de um estudo mais aprofundado, portanto convido você a pesquisar mais sobre cada ponto que aqui foi abordado. Continue o seu estudo através de mais pesquisa em websites, conversando com especialistas ou adquirindo revistas especializadas. Não podemos deixar de mencionar que um bom ponto de partida para examinar o assunto são os livros:

TOCCI, Ronald J. **Sistemas digitais: princípios e aplicações**. 11. ed. São Paulo: Pearson Prentice Hall, 2011.

VAHID, Frank. **Sistemas digitais: projetos, otimização e HDLs**. Porto Alegre: Bookman, 2008.

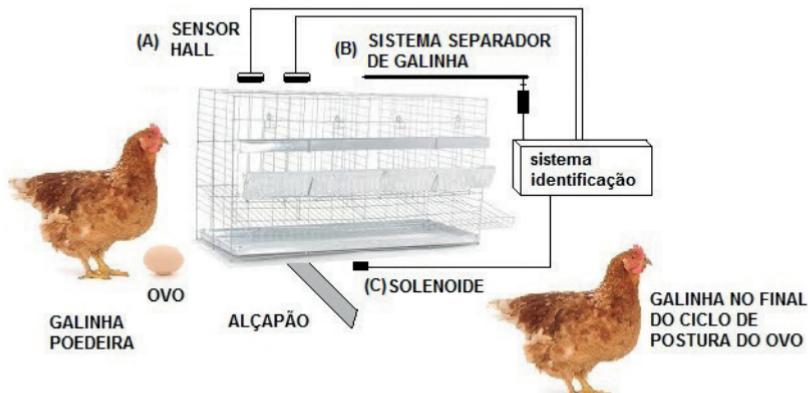
Sem medo de errar

Relembrando: você investiu em uma empresa iniciante que atua na área agrícola, e a sua solução de automação gerou uma economia de 20% nas rações e reduziu em 95% a mortalidade das aves. Você decidiu iniciar um novo projeto que vai revolucionar o mercado de ovos. Consiste em um sistema que, utilizando circuitos digitais, identifica qual galinha não está mais botando ovo, podendo substituí-la por outra. Para isso, você vai utilizar, além do sistema que verifica qual galinha não está mais na postura, um servomotor para separar a galinha improdutiva e um sistema de comporta acionado por relé para abrir um alçapão. Obtenha uma visão geral analisando a Figura 4.35.

Inicialmente, você deve entender como deve se integrar cada sistema. Temos que pensar em três sistemas distintos: **(A)** análise do sensor Hall; **(B)** sistema separador de galinha e **(C)** acionamento do solenoide.

Vamos focar inicialmente nas portas de saída e entrada de dados do sistema de identificação. O sistema de identificação deve obrigatoriamente ter uma porta de saída e uma de entrada de sinal TTL. Para o sensor Hall, devemos ter uma porta que permita a entrada de um sinal analógico. Logo, amplifique o sinal utilizando um amplificador comparador – um exemplo de circuito amplificador pode ser encontrado na folha de dados do LM331.

Figura 4.35 | Sistema de identificação e separação de galinhas



Fonte: elaborada pelo autor.

Para o sistema de separação de galinhas, utilize PWM para acionar o servomotor. Considere que a largura de pulso de um servomotor está entre 1 e 2ms e que o período de repetição de cada ciclo é de 20ms (50 Hz). Considere também que um servomotor possui três fios de interface, sendo dois de alimentação e um fio para o PWM.

Para o acionamento do Solenoide, você pode usar duas abordagens: a primeira é utilizar um circuito TTL seguido de um optoisolador ou um isolador digital seguido do solenoide, e segunda abordagem é utilizar um optoisolador ou um isolador digital seguido de um relé em série com solenoide. Esta última abordagem é mais robusta mas utiliza um relé, o que pode ser um problema caso esse relé se desgaste com o tempo.

Nessa solução, procure utilizar o que foi aprendido nesta unidade. Faça um relatório com os esquemas elétricos e as especificações de cada porta de saída e entrada. Utilize como referência um microcontrolador de 8 bits de uso geral. Gere uma tabela com a lista de componentes (também conhecida como B.O.M., do inglês *Bill Of Materials*). Esta lista deve conter o máximo de informação dos componentes utilizados, que são detalhes como nome, número do projeto, quantidade, custo, distribuidor e observações gerais, entre outros detalhes. Tenha em mente que, quanto mais informações estiverem descritas no B.O.M., melhor será para o desenvolvedor e para a equipe de compras.

Avançando na prática

Ajuste da velocidade de um motor CC com PWM

Descrição da situação-problema

Vamos imaginar a seguinte situação: você foi contratado para desenvolver um sistema que ajusta a velocidade de uma esteira de distribuição de malas em um aeroporto. Você pode interromper ou aumentar a velocidade de 5% até 95% da velocidade máxima do motor. Proponha um sistema com base no circuito integrado 555 e um resistor variável para esse controle de velocidade. Pesquise na folha de dados do componente para saber mais detalhes, caso necessário.

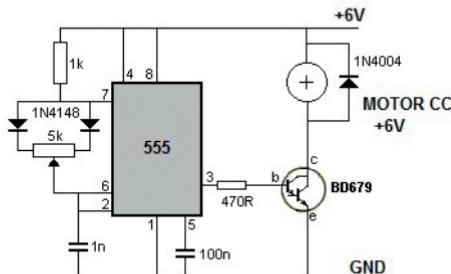
Resolução da situação-problema

Para resolver esse desafio, você propôs uma solução de sistema, que consiste em:

- Um circuito 555.
- Um motor CC de 6 volts.
- Controle através de um resistor variável.

Observe na Figura 4.36 um circuito gerador do sinal PWM adaptado para um motor CC.

Figura 4.36 | Circuito 555 gerador de sinal PWM



Fonte: elaborada pelo autor.

Fique atento, pois o capacitor temporizador será em função das características do motor CC utilizado, assim o motor não vibrará muito no momento da parada nem em baixas velocidades. Fique atento à temperatura no transistor; caso necessário, coloque um dissipador de temperatura. Experimente outras abordagens utilizando outros transistores.

Faça valer a pena

1. Considere as especificações de um relé de uso geral abaixo:

- 1 bobina de contato reversível para 12 A.
- Montagem direta em circuito impresso.
- Baixo custo.
- Produção automatizada.
- Disponível na versão selada.

Com base nessas especificações, analise o enunciado a seguir.

Uma alternativa para substituição dos relés é o emprego de optoacopladores

ou isoladores digitais. Em relação a esses dispositivos, podemos afirmar que:

- a) Optoisoladores comuns e com preços mais acessíveis são em geral dispositivos de seis pinos baseados em relés. Esses dispositivos exigem que o relé na entrada seja ativado com uma corrente de aproximadamente 10 a 30 mA. Uma vez ativado o relé, este aciona o fototransistor dentro do dispositivo, que por sua vez começa a conduzir levando saída para nível lógico alto.
- b) Isoladores digitais baseados em CMOS têm o funcionamento totalmente diferente quando comparado a optoisoladores, e, portanto, não podem ser empregados em sistemas similares.
- c) Optoisoladores não são limitados, pois utilizam tecnologia mais moderna baseadas em relés de alta velocidade.
- d) Isoladores digitais são superiores em desempenho quando comparados a optoisoladores. Possuem alta confiabilidade e taxas de transmissão com isolamento faixa dos kVolts, pois são baseados em isolamento capacitivo.
- e) Fatores que reduzem a confiabilidade dos optoisoladores: envelhecimento da bobina do relé, temperatura e umidade.

2. Circuitos digitais podem dar problemas. Uma parcela desses problemas podem ser diagnosticados com uma inspeção visual, ou seja, com um pouco de calma devemos observar qualquer anormalidade no circuito. Essas anormalidades podem ser identificadas visualmente procurando no sistema componentes inflados (como se fossem explodir) ou rachados, componentes com aparência de queimados ou mesmo a placa de circuito impresso queimada ou empenada, fios ou fusíveis rompidos.

Com relação ao diagnóstico de um problema em um circuito digital, qual alternativa que melhor se enquadra:

- a) Inspecionar a fonte de alimentação é uma prática desnecessária, pois todo circuito é alimentado com a tensão correta.
- b) Analisar o plano terra com relação à alimentação com um multímetro, no intuito de investigar curto circuito, é uma análise válida.
- c) Analisar os sinais de saída do circuito digital não é uma prática comum, portanto, desnecessária.
- d) Inspecionar a continuidade da linha de alimentação que chega até o circuito digital costuma ser a razão dos problemas em um circuito digital.
- e) Circuitos digitais não possibilitam inspeção e reparo.

3. Um motor CC tem o sentido da rotação definido pelo sentido da corrente. Podemos verificar de forma bem simples essa afirmação. Basta desligarmos o motor CC, invertermos os pinos de alimentação e ligarmos novamente o motor CC, e o sentido de rotação do motor é invertido.

Servomotores são uma solução encontrada para resolver o problema de controle de velocidade e posição de motores CC. Neste sentido, podemos afirmar sobre servomotores:

- a) Consistem em sistema com (i) sistema atuador – motor elétrico (CC ou AC) e engrenagens; (ii) sensor – indica a posição angular do eixo do motor; e, por fim, um (iii) circuito de controle – controle em malha fechada.
- b) Servomotores permitem que o eixo rode somente +90 graus para esquerda, -90 graus para a direita e nenhum valor intermediário.
- c) Servomotores têm uma interface simples com quatro fios: dois fios de alimentação (VCC e GND), um fio de controle e um fio de realimentação.
- d) Servomotor utiliza o protocolo UART (universal asynchronous receiver/transmitter), sendo que o tamanho dos dados determina a posição que o eixo irá se deslocar.
- e) Servomotores são amplamente utilizados na medicina, em mergulho autônomo e em voo em planadores, entre outros sistemas que necessitam de velocidade no movimento de eixos.

Referências

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

IMPERIAL COLLEGE LONDON. **SERVO MOTOR SG90**. Londres: 2017. 1 p. Disponível em: <http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf> Acesso em: 04 fev. 2018.

NATIONAL Semiconductors. **MM54HCT34/MM74HCT34 Non-Inverter**. Arlington: National Semiconductors, 1988. 6 p. Disponível em: <<http://pdf.datasheetcatalog.com/datasheet/nationalsemiconductor/DS005359.PDF>>. Acesso em: 06 abr. 2018.

RAKO, P. **IC Packages and thermal design**. EDN Networks, mar. 2016. Disponível em: <<https://www.edn.com/electronics-blogs/anablog/4441750/IC-packages-and-thermal-design>>. Acesso em: 12 abr. 2018.

SILICON LABS. **Isolator vs. Optocoupler Technology**. Austin, Texas: Silicon Labs, 2017. 4 p. Disponível em: <<https://www.silabs.com/whitepapers/isolator-vs-optocoupler-technology>>. Acesso em: 17 jan. 2018.

TEXAS INSTRUMENTS. **SN5406, SN5416, SN7406, SN7416 HEX INVERTER BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS**. Dallas: Texas Instruments, 1983. 19 p. Disponível em: <<http://www.ti.com/lit/ds/symlink/sn5406.pdf>>. Acesso em: 06 abr. 2018.

TEXAS INSTRUMENTS. **CD4049UB and CD4050B CMOS Hex Inverting Buffer and Converter**. Dallas: Texas Instruments, 1998. 35 p. Disponível em: <<http://www.ti.com/lit/ds/symlink/cd4049ub.pdf>>. Acesso em: 06 abr. 2018.

TEXAS INSTRUMENTS. **SNx400, SNx4LS00, and SNx4S00 Quadruple 2-Input Positive-NAND Gates**. Dallas: Texas Instruments, 2017. Disponível em <<http://www.ti.com/lit/ds/symlink/sn74s00.pdf>>. Acesso em 05 abr. 2018.

TOCCI, Ronald J.; WIDMER, Neal S. **Sistemas digitais**: princípios e aplicações. [S.l.: s.n.], 2011.

Anotações

ISBN 978-85-522-0715-3



9 788552 207153 >