

CURSO DE ELETRÔNICA

ELETRÔNICA DIGITAL - I

Newton C. Braga



VOLUME

3

NCB

**CURSO DE ELETRÔNICA
ELETRÔNICA DIGITAL - PARTE 1**

NEWTON C. BRAGA



Instituto NCB

www.newtoncbraga.com.br

contato@newtoncbraga.com.br

CURSO DE ELETRÔNICA -

Eletrônica Digital - 1

Autor: Newton C. Braga

São Paulo - Brasil - 2012

Palavras-chave: Eletrônica - Engenharia Eletrônica
- Componentes - Educação Tecnológica

Diretor responsável: Newton C. Braga

Diagramação e Coordenação: Renato Paiotti

MAIS INFORMAÇÕES

INSTITUTO NEWTON C. BRAGA

<http://www.newtoncbraga.com.br>

NOTA IMPORTANTE

Esta série de livros fornece conhecimentos básicos de eletrônica para cursos regulares, cursos a distância e para autodidatas consistindo, portanto numa literatura cuja finalidade é apoio, iniciação ou complementação de conhecimentos. Sua aquisição não implica no direito a obtenção de certificados ou diplomas os quais devem ser emitidos pelas instituições que adotam o livro ou ainda ministram cursos de outras formas. Da mesma forma o autor ou a editora não se responsabilizam por eventuais problemas que possam ser causados pelo uso indevido das informações nele contidas como o não funcionamento de projetos, ferimentos ou danos causados a terceiros de forma acidental ou proposital, ou ainda prejuízos de ordem moral ou financeira. Os eventuais experimentos citados quando realizados por menores devem ter sempre a supervisão de um adulto. Todo cuidado foi tomado para que o material utilizado seja encontrado com facilidade na época da edição do livro, mas as mudanças tecnológicas são muito rápidas o que nos leva a não nos responsabilizarmos pela eventual dificuldade em se obter componentes para os experimentos quando indicados em outros livros desta série.

Copyright by
INSTITUTO NEWTON C. BRAGA
1ª edição

Todos os direitos reservados. Proibida a reprodução total ou parcial, por qualquer meio ou processo, especialmente por sistemas gráficos, microfilmicos, fotográficos, reprográficos, fonográficos, videográficos, atualmente existentes ou que venham a ser inventados. Vedada a memorização e/ou a recuperação total ou parcial em qualquer parte da obra em qualquer programa juscibernético atualmente em uso ou que venha a ser desenvolvido ou implantado no futuro. Estas proibições aplicam-se também às características gráficas da obra e à sua editoração. A violação dos direitos autorais é punível como crime (art. 184 e parágrafos, do Código Penal, cf. Lei nº 6.895, de 17/12/80) com pena de prisão e multa, conjuntamente com busca e apreensão e indenização diversas (artigos 122, 123, 124, 126 da Lei nº 5.988, de 14/12/73, Lei dos Direitos Autorais).

ÍNDICE

1 - Eletrônica Analógica e Digital -

Sistemas de Numeração	11
1.1 - Analógico e digital	12
1.2 – Lógica digital	14
1.3 – Sistemas de numeração	16
1.4 – Numeração binária	18
1.4.1 - Quilo, Mega e Giga	24
1.5 – Binários menores que 1	25
1.5.1 - Potências Negativas de 2	26
1.6 – Formas diferentes de usar o sistema binário ..	26
1.6.1 - Sistema BCD (Decimal Codificado em Binário)	26
1.6.2 – Código Biquinário	28
1.6.3 – Código Excesso 3 (XS3)	28
1.6.4 - Código Aiken ou 2421	29
1.6.5 – Código Gray	29
1.6.6 – Código 987654321	30
1.6.7 - ASCII	31
1.6.8 – Código Fortran	32
1.6.9 - Códigos de erros	33
1.6.10 – Binário com Paridade	33
1.6.11 - Código 2 de 5	34
1.7 - Sistema Hexadecimal	34
1.8 – Aritmética Binária	37
Termos em inglês	40

2 - A Álgebra de Boole

2 - A Álgebra de Boole	43
2.1 - A algebra de Boole	44
2.2 - Os níveis lógicos	45
2.3 - Operações Lógicas	46
2.3.1 – Raciocínio Lógico e notações	48
2.4 - Função Lógica NÃO ou Inversora	50
2.5 - Função Lógica E (AND)	52
2.6 - Função lógica OU	54
2.7 - Função NÃO-E (NAND)	56
2.8 - Função Não-Ou (NOR)	59
2.9 - Função Ou-exclusivo (Exclusive-OR)	61

2.10 - Função Não-Ou exclusivo ou coincidência (Exclusive-NOR)	62
2.11 - Propriedades das operações lógicas	63
Relações Booleanas	73
Teoremas de De Morgan:	73
2.12 - Fazendo tudo com portas NAND ou Não-OU	74

3 - Famílias de Circuitos, Lógicos Digitais

3 - Famílias de Circuitos, Lógicos Digitais	79
3.1 - O transistor como chave eletrônica	80
3.2 - Melhorando o desempenho	81
3.3 - A família TTL	83
3.3.1. - Consumo & Velocidade	86
3.4 - Outras Características da Família TTL	86
3.4.1 - Correntes de entrada:	86
3.4.2 - Correntes de saída	88
3.4.3 - Fan In e Fan Out	89
3.4.4 - Velocidade	91
3.5 - Subfamílias TTL	92
3.6 - Compatibilidade entre as subfamílias	95
3.7 - Open Collector e Totem-Pole	96
3.8 - Tri-State	98
Termos em Inglês:	101

4 - A Família de Circuitos

Integrados CMOS	103
4.1 – Os circuitos integrados CMOS	104
4.2 - Aplicações digitais	107
4.3 – Consumo e velocidade	108
4.4 – Famílias e Subfamílias CMOS	112
4.5 – Sensibilidade ao manuseio	113
4.6 - As Configurações CMOS	116
4.7 -Especificações	118
4.7.1 – Desacoplamento	121
4.8 – Interfaceando	121
4.9 – Fontes de alimentação	126
Termos em Inglês	132

5 - Combinando Funções Lógicas (Lógica Combinacional) 135

5.1 - As tabelas verdade	136
5.2 - Lógica Combinacional	137
5.3 - Como Projetar Um Circuito Combinacional .	140
5.4 - Simplificando e Minimizando	148
5.5 - Diagramas de Karnaugh	149
Termos em inglês	155

6 - Os Elementos Biestáveis 157

6.1 - Os Flip-Flops	158
6.2 - Flip-flop R-S	159
6.2.1 - Repiques	164
6.3 - Flip-flops RS com Clock Mestre-Escravo	164
6.4 - O flip-flop J-K Mestre-Escravo	170
6.5 - O flip-flop tipo D	175
6.6 - Flip-flop tipo T	175
6.7 - Transformando Flip-Flops	177
6.8 - Nos equipamentos digitais	179
6.9. Os flip-flops antigos	180
Termos em inglês:	181

7 - Os Flip-Flops e Funções, Integradas em Circuitos Integrados 183

7.1 - Os flip-flops TTL	184
7.2 - Os flip-flops CMOS	194
7.3 - Funções lógicas TTL	200
7.4 - Funções lógicas CMOS	204
7.5 - A Função Tri-State Expansível do 4048	209
Termos em Inglês:	212

A - RESPOSTAS - FONTES PARA INTEGRADOS TTL 215

A.1 - Fontes para integrados TTL	215
Fonte TTL de 5 V x 1 A	215
Fonte TTL 5 V x 5 - LM301	216
Fonte de 5 V x 500 mA	216

Em 1972, já com experiência no ensino de eletrônica em cursos presenciais, fui contratado por uma grande organização de ensino por correspondência para renovar seu curso prático de eletrônica. Completado esse trabalho, fui trabalhar na Editora Saber em 1976 onde passei a publicar nas páginas da Revista Saber Eletrônica o primeiro Curso de Eletrônica em Instrução Programada, uma novidade que atraiu a atenção de milhares de leitores que tiveram sua formação inicial totalmente apoiada nos ensinamentos que então disponibilizamos. O sucesso deste curso fez com que em diversas ocasiões posteriores o curso fosse repetido e atualizado nas páginas da mesma revista e na revista Eletrônica Total. Neste intervalo publicamos a primeira edição completa desse curso que recebeu o nome de Curso Básico de Eletrônica e chegou até sua quinta edição, posteriormente sendo em 2009 transformado numa apostila. No entanto, desde a primeira edição e o primeiro curso na revista, muita coisa mudou, e se bem que diversas atualizações fossem feitas, chegou o momento de se fazer algo novo, adaptado aos novos tempos da eletrônica, num formato mais atual e com conteúdo que seja mais útil a todos que desejarem aprender o básico da eletrônica. Desta forma o conteúdo do curso anterior foi separado em dois, Curso de Eletrônica - Eletrônica Básica (já publicado – Vol 1 da série) e Curso de Eletrônica - Eletrônica Analógica (que é este volume – Vol 2), os quais devem ser completados com uma nova versão do Curso de Eletrônica Digital. O Curso de Eletrônica Digital foi remodelado, sendo dividido em dois volumes. Este é o primeiro e depois do segundo teremos um quinto volume da série com uma parte prática. Assim, neste primeiro volume do Curso de Eletrônica Digital, abordamos todo o conhecimento adquirido nos volumes anteriores passando isto para esta tecnologia além de incluir mais informações sobre novas tecnologias, novos componentes e novas aplicações. Podemos dizer que este livro, como os demais, podem ser considerados a plataforma de iniciação ideal para muitos cursos, dos técnicos às disciplinas eletivas, da reciclagem de conhecimentos até aqueles que desejam ter na eletrônica uma segunda atividade ou precisam deles para o seu trabalho em área relacionada.

Desde 1999, quando criamos a primeira versão deste Curso de Eletrônica Digital que pudesse servir de iniciação aos que desejassem ter conhecimentos desta tecnologia, ela passou por grandes transformações. Do fim da válvula ao transistor, quando começamos e os primeiros circuitos integrados, a eletrônica evoluiu para a tecnologia dos CIs de alto grau de integração, os FPGAs, os DSPs, microcontroladores e as montagens em superfície. Assim, nosso livro Curso de Eletrônica Digital, com o primeiro volume tratando dos conceitos básicos e componentes das tecnologias TTL e CMOS, ele pode ser considerado um curso atualizado com finalidades um pouco diferentes das que visava na época de sua criação original. A eletrônica digital está presente numa infinidade de aplicações de uma forma muito avançada, no entanto, para que possamos dominar essas novas aplicações, precisamos conhecer seus fundamentos. Desta forma, este Curso de Eletrônica digital é um curso de fundamentos que devem ser aplicados nos ramos específicos nos quais o profissional vai se especializar. Estes ramos incluem a automação (mecatrônica), telecomunicações, instrumentação, eletrônica embarcada, e muitos outros. Assim, nosso curso visando justamente às necessidades de conhecimento que a preparação para essas áreas pedem, tem uma abordagem direta e rápida de conceitos que, em princípio, exigem o conhecimento prévio dados nos dois primeiros volumes da série, Eletrônica Básica e Eletrônica Analógica.

Capítulo 1 - Nesta lição estudaremos as diferenças entre o que é analógico e o que é digital, com uma introdução à lógica digital e os sistemas de numeração, com ênfase a numeração binária. Também veremos como utilizar os binários e o sistema hexadecimal, além das regras da aritmética binária.

Capítulo 2 – O assunto desta lição é a Álgebra de Boole, base de toda a tecnologia digital. Estudaremos também as funções lógicas utilizadas na prática, analisando suas propriedades e suas aplicações. Estas funções são as E, OU, Não-E e Não-Ou além de outras.

Capítulo 3 – A terceira lição deste curso analisa o transistor como chave eletrônica e como ele pode ser utilizado nos circuitos digitais. Teremos uma introdução à família TTL, suas subfamílias e também analisaremos a compatibilidade com outras famílias. Trataremos de algumas características importantes desta família como as saídas tri-state.

Capítulo 4 – A lição 4 dedica-se aos componentes da família lógica CMOS, analisando-se suas aplicações e características. Vere-

mos quais são as funções disponíveis e suas configurações. Também veremos como interfacear os componentes desta família com outros circuitos.

Capítulo 5 – O assunto desta lição é a lógica combinacional, ou seja, como os circuitos lógicos digitais podem ser combinados para realizar funções complexas. Aprenderemos como simplificar os circuitos e como usar os diagrama da Karnaugh para esta finalidade.

Capítulo 6 – A importância dos flip-flops na tecnologia digital é muito grande e existem diversos tipos destes blocos. Analisaremos nesta lição como funcionam os diversos tipos de flip-flops e onde eles podem ser usados. Também veremos como eles podem ser transformados.

Capítulo 7 – A importância dos flip-flops ficou patente na lição anterior. Nesta lição estudaremos tanto os flip-flops em tecnologia TTL como CMOS e como usá-los. Também trataremos de algumas funções lógicas disponíveis em tecnologia TTL e CMOS.

Enfim, o conteúdo estudado pode ser considerado como mais um degrau de uma escada que levará os interessados a um mundo de conhecimento técnico capaz de significar sua realização profissional e muito mais que isso, a satisfação pessoal de dominar as mais importantes tecnologias de nosso tempo.

Newton C. Braga



» Eletrônica Analógica e Digital

» Sistemas de Numeração

Nesta primeira lição de nosso curso trataremos do relacionamento entre a eletrônica analógica (que estudamos nos dois primeiros volumes desta série de Cursos) e a eletrônica digital. Veremos quais são as diferenças entre as duas eletrônicas, de modo a poder entender melhor o que realmente se denomina eletrônica digital. Teremos uma noção básica sobre o modo como os circuitos digitais processam numericamente as informações funcionam e como os sistemas de numeração, com ênfase na numeração binária e hexadecimal. Esta lição será então formada pelos seguintes itens:

- 1.1 – Analógico e digital
- 1.2 – Lógica digital
- 1.3 – Sistema de numeração
- 1.4 – Numeração binária
- 1.5 - Binários menores que 1
- 1.6 – Formas diferentes de usar binários
- 1.7 – Sistema hexadecimal
- 1.8 – Aritmética binária

Objetivos desta lição:

- Diferenciar grandezas analógicas e digitais
- Entender o que é lógica digital
- Estudar os diferentes sistemas de numeração
- Conhecer a numeração binária
- Contato com alguns códigos binários importantes
- Aprender como funciona a numeração hexadecimal
- Entender como as operações de soma, subtração, multiplicação e divisão com binários são realizadas

DSP

Uma tecnologia muito importante em nossos dias é a que faz uso dos DSPs ou Digital Signal Processor ou Processadores Digitais de Sinais. Estes circuitos processam sinais analógicos, convertendo-os para a forma digital e depois retornam com estes sinais processados para sua forma original analógica. Trata-se de um “interfaceamento” digital do mundo analógico. A Texas Instruments possui uma ampla linha de processadores de sinais digitais, como os da série TMS320.

**1.1 - Analógico e digital**

Por que digital? Esta é certamente a primeira pergunta que qualquer leitor que está “chegando agora”, e tem apenas alguma base teórica da Eletrônica, principalmente da eletrônica analógica, como a ensinada nos primeiros volumes desta série, faria ao encontrar o nosso curso.

Por este motivo, começamos nosso curso justamente por explicar as diferenças entre as duas eletrônicas, de modo que elas fiquem bem claras. Devemos lembrar que em muitos equipamentos, mesmo classificados como analógicos ou digitais, encontraremos os dois tipos de circuitos. É o caso dos computadores, processadores, equipamentos de telecomunicações, automatismos e instrumentos de laboratório, e muitos outros que, mesmo sendo classificados como “máquinas estritamente digitais”, podem ter em alguns pontos de seus circuitos configurações analógicas.

Uma definição encontrada nos livros especializados atribui o nome “Eletrônica Digital” aos circuitos que operam com quantidades que só podem ser incrementadas ou decrementadas em passos finitos.

Um exemplo disso é dado pelos circuitos que operam com impulsos. Só podemos ter números inteiros de pulsos sendo trabalhados em qualquer momento em qualquer ponto do circuito. Em nenhum lugar encontramos “meio pulso” ou “um quarto de pulso”.

A palavra digital também está associada a dígito (do latim digitus de “dedo”) que esta associado à representação de quantidades inteiras. Não podemos usar os dedos para representar meio pulso ou um quarto de pulso.

Na eletrônica analógica trabalhamos com quantidades ou sinais que podem ter valores que variam de modo contínuo numa escala. Os valores dos sinais não precisam ser inteiros. Por exemplo, um sinal de áudio, que é analógico, varia suavemente entre dois extremos enquanto que um sinal digital só pode variar aos saltos, conforme mostra a figura 1.

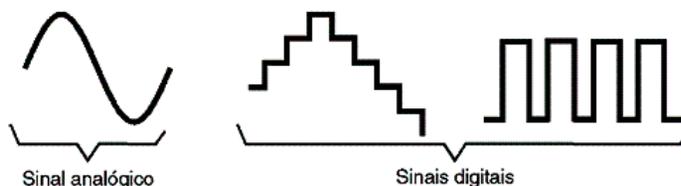


Figura 1 – Um sinal digital varia em saltos, ou seja, assume valores discretos

Conforme o leitor pode perceber, a diferença básica entre os dois tipos de eletrônica está associada inicialmente ao tipo de sinal com que elas trabalham e no que elas fazem com os sinais.

De uma forma resumida podemos dizer que:

- A eletrônica digital trabalha com sinais que só podem assumir valores discretos ou inteiros.

· A eletrônica analógica trabalha com sinais que podem ter qualquer valor entre dois limites.

Onde Encontramos A Eletrônica Digital

COMPUTADORES - os computadores atuais são digitais em sua totalidade, e praticamente não se usa outro tipo de configuração, a não ser no interfaceamento com o mundo exterior, pois somos analógicos. No entanto, nem sempre foi assim. Nas primeiras décadas deste século, quando os circuitos eram ainda valvulados os primeiros computadores eram máquinas analógicas. A imprecisão, e algumas outras dificuldades técnicas, que estes computadores apresentavam fizeram com que eles logo fossem substituídos pelos circuitos digitais que nós usamos hoje.

TELECOMUNICAÇÕES - Antigamente também, todos os equipamentos de telecomunicações trabalhavam diretamente com sinais analógicos vindos de microfones, câmeras de TV e outras fontes. No entanto, atualmente a maioria das transmissões de informações sem fio, por ondas de rádio, e por meios físicos como fibras ópticas e cabos, ocorre na forma digital. Os sinais analógicos são convertidos em digitais e, com isso, transmitidos de forma muito mais eficiente.

INDÚSTRIA E AUTOMAÇÃO - Antigamente, os controles de máquinas industriais eram simples, não passando de interruptores e chaves ou, no máximo, dispositivos que controlavam diretamente sinais analógicos. Também neste caso, tivemos uma evolução com o uso de microprocessadores, microcontroladores e DSPs (Digital Signal Processors ou Processadores Digitais de Sinais). Neles, os sinais analógicos de sensores e controle são convertidos em sinais digitais e usados nos equipamentos.

INSTRUMENTAÇÃO – A maioria dos instrumentos de laboratório são digitais. Com uma precisão maior e a possibilidade de processar as medidas feitas, eles ainda podem ter recursos para gravar ou enviar dados através da internet ou outros meios de transmissão, com recursos muito maiores do que os equivalentes analógicos.

ELETRÔNICA DE CONSUMO – Muitos dos eletro-eletrônicos que utilizamos hoje possuem chips de controle que nada mais são do que microcontroladores, operando de forma totalmente digital. Isso ocorre com seu aquecedor a gás, seu forno de microondas, seu relógio digital, calculadora, TV, DVD player, e muito mais.

ELETRÔNICA EMBARCADA – Todos os veículos modernos possuem um microcontrolador que gerencia o funcionamento de suas

Nosso mundo digital – a todo instante estamos em contato com equipamentos digitais. Na foto temos apenas alguns deles.



partes, do motor aos sistemas de segurança e navegação, incluindo-se o GPS e sistemas de entretenimento.

ELETRÔNICA MÉDICA – Uma vasta quantidade de equipamentos médicos utiliza recursos digitais no seu controle, com um grau de sofisticação elevado, pela capacidade dos circuitos empregados. São equipamentos de raios-X, tomografia, etc.

1.2 – Lógica digital

Todos os equipamentos que usam circuitos digitais funcionam obedecendo a um tipo de comportamento baseado no que se denomina Lógica.

Diferentemente dos circuitos amplificadores comuns que simplesmente amplificam, atenuam ou realizam algum tipo de processamento simples dos sinais, os circuitos digitais não processam os sinais baseados em uma finalidade simples que é determinada quando são fabricados.

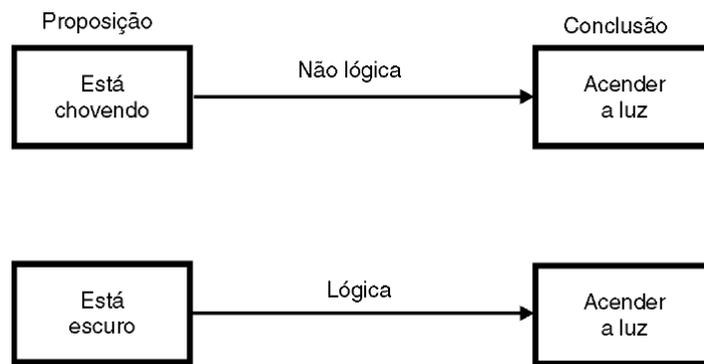
Os circuitos digitais de todos os equipamentos que fazem uso desta tecnologia são capazes de combinar os sinais, tomando decisões segundo um comportamento lógico.

É evidente que, se o leitor deseja realmente entender as coisas que ocorrem com os circuitos digitais, deve partir exatamente do aprendizado do comportamento que desejamos que eles tenham, ou seja, um comportamento lógico e isso implica em saber que tipo de comportamento é este.

Podemos dizer que a partir da lógica podemos tirar conclusões a partir de fatos conhecidos, ou tomar decisões a partir de fatos conhecidos.

Por exemplo, a decisão de “acender uma lâmpada quando está escuro” é uma decisão lógica, pois a proposição e a conclusão são fatos relacionados.

Ao contrário, a decisão de “acender uma lâmpada porque está chovendo” não é uma decisão lógica, pois os fatos envolvidos não têm relação.



Elementos simples de lógica são a base de funcionamento dos circuitos digitais.

Figura 2 – Decisões que envolvem lógica e que não são lógicas

Outros exemplos de lógica:

- Acionar um relé quando a temperatura chega a um determinado valor
- Tocar uma campainha quando uma peça chegar a uma determinada posição.
- Abrir uma porta quando a senha correta for digitada num teclado

Evidentemente, os fatos relacionados acima são simples e servem para exemplificar como as coisas funcionam.

Na eletrônica dos controles lógicos de máquinas, como os microprocessadores, microcontroladores, DSPs e computadores, o que temos é a aplicação da lógica digital, ou seja, de circuitos que operam tomando decisões em função de coisas que acontecem no seu próprio interior ou em dispositivos que estejam ligados a eles.

As decisões são extremamente simples, mas se combinarmos muitas delas poderemos ter comportamentos muito complexos, como os encontrados nos computadores, máquinas industriais, microcontroladores, robôs, etc.

É claro que estes equipamentos e seus circuitos digitais não podem entender coisas como está escuro ou está chovendo e tomar decisões.

Os circuitos lógicos digitais trabalham com sinais elétricos.

Assim, os circuitos lógicos digitais nada mais fazem do que receber sinais com determinadas características e em sua função tomam decisões que nada mais são do que a produção de outro sinal elétrico.

Mas, se os sinais elétricos são digitais, ou seja, representam quantidades discretas, e se a lógica é baseada em tomada de decisões, o próximo passo no entendimento da eletrônica digital, com sua lógica é justamente partir para o modo como as quantidades discretas são representadas e entendidas pelos circuitos eletrônicos.

Definição de lógica

Uma definição comum de lógica é aquela que a coloca como “a ciência das leis do pensamento”, ou seja, do conhecimento das leis do pensamento. No entanto, uma definição mais adequada seria a de que “a lógica a ciência das leis do raciocínio”, no sentido de que ela garante que nossos pensamentos ocorram de forma correta levando a conhecimentos verdadeiros.

1.3 – Sistemas de numeração

O modo como contamos as quantidades vem do fato de possuímos 10 dedos. Assim, tomando os dedos das mãos, podemos contar objetos com facilidade até certo ponto.

O ponto crítico ocorre quando temos quantidades maiores do que 10. O homem resolveu o problema passando a indicar também a quantidade de mãos, ou de vezes em que os dez dedos são usados.

Assim, quando dizemos que temos 27 objetos, o 2 indica que temos “duas vezes as mãos cheias” ou duas dezenas mais 7 objetos. O 2 tem peso 10. Da mesma forma, quando dizemos que temos 237 objetos, o 2 agora indica que temos “duas dezenas de pares de mãos cheias” ou duas centenas enquanto que o 3 indica que temos mais 3 pares de mãos cheias e finalmente o 7, mais 7 objetos, conforme mostra a figura 3.

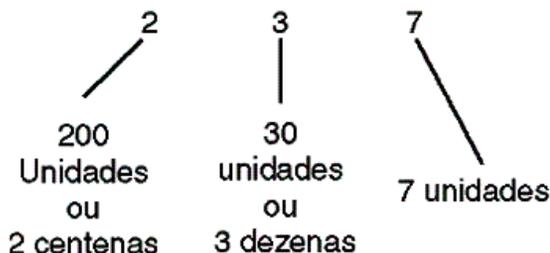


Figura 3 – O algarismo tem um valor que depende da sua posição relativa no número representado.

Em outras palavras a posição dos algarismos na representação dos números tem um peso e no nosso sistema de numeração que é decimal, esse peso é 10, conforme mostra a figura 4.

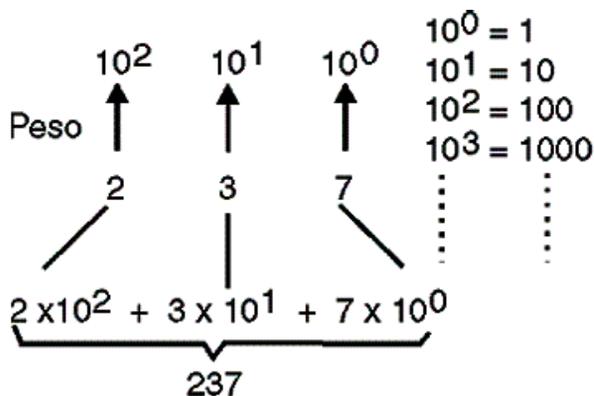


Figura 4 - No sistema de numeração decimal, os pesos dos algarismos são potências de 10

O que aconteceria se tivéssemos um número diferente de dedos, por exemplo, 2 em cada mão?

Isso significaria, em primeiro lugar, que no nosso sistema de base 4 (e não base 10) só existiriam 4 algarismos para representar os números: 1, 2, 3 e 4, conforme mostra a figura 5.

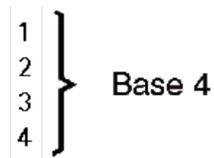


Figura 5 – Apenas 4 algarismos estariam disponíveis para representar as quantidades

Para representar uma quantidade maior do que 4, teríamos de usar mais de um algarismo.

Assim, para indicar 7 objetos na base 4, teríamos “uma mão cheia com 4” e mais 3. Isso daria 13, conforme mostra a figura 6.

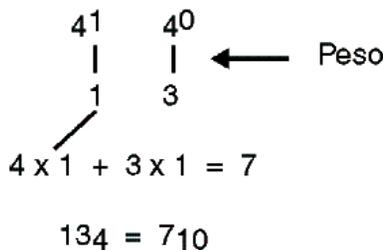


Figura 6 – Representando uma quantidade maior que 4. Para 7 teríamos 13_4 ,

Notação

O índice “4” no 13_4 indica que o valor representado está na base 4. Da mesma forma 7_{10} , o 10 indica que o valor representado está na base 10.

Veja então que no “13” na base 4, o 1 tem peso 4, enquanto que o 3 tem o seu valor normal.

De uma forma generalizada dizemos que, dependendo da base do sistema, os algarismos têm “pesos” que correspondem à sua posição no número, e que estes pesos são potências da base.

Por exemplo, para a base 10, cada algarismo a partir da direita tem um peso que é uma potência de 10 em ordem crescente, o que nos leva à unidade (10 elevado a 1), à dezena (dez elevado ao expoente 1), à centena (dez elevado ao quadrado), ao milhar (dez elevado ao cubo) e assim por diante, conforme mostra a figura 7.

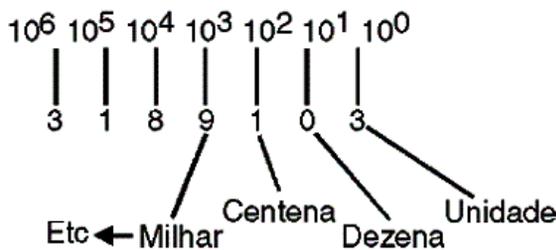


Figura 7 – Os pesos dos algarismos aumentam da direita para a esquerda

Em eletrônica digital, costumamos dizer que o dígito mais à direita, por representar a menor potência ou ter menor peso, é o dígito (ou bit*) menos significativo ou LSB (Least Significant Bit), enquanto que aquele que está mais à esquerda é o mais significativo ou MSB (Most Significant Bit).

* O bit que é o dígito binário (na base 2) será estudado mais adiante

Para a base 4, os dígitos têm potências de 4 conforme mostra a figura 8.

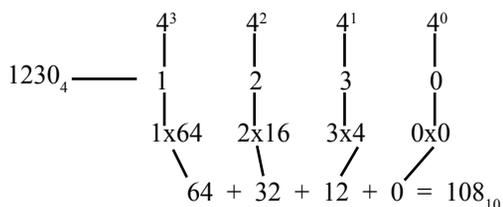


Figura 8 – Os pesos na base 4

História

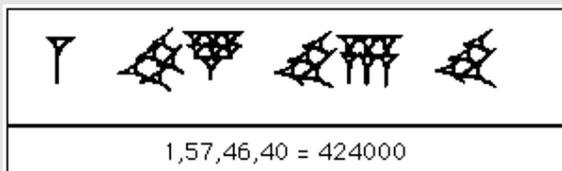
Nem sempre os povos usaram a base 10 para a contagem. Na Babilônia, por exemplo, a base era o 12, daí a dúzia, a divisão do dia em 12 + 12 horas e da hora em 60 minutos. Alegam os defensores da base 12, que ela pode ser dividida mais vezes, do que na base 10, antes de chegarmos a números que não sejam inteiros.

𐀀 1	𐀁 11	𐀂 21	𐀃 31	𐀄 41	𐀅 51
𐀆 2	𐀇 12	𐀈 22	𐀉 32	𐀊 42	𐀋 52
𐀌 3	𐀍 13	𐀎 23	𐀏 33	𐀐 43	𐀑 53
𐀒 4	𐀓 14	𐀔 24	𐀕 34	𐀖 44	𐀗 54
𐀘 5	𐀙 15	𐀚 25	𐀛 35	𐀜 45	𐀝 55
𐀞 6	𐀟 16	𐀠 26	𐀡 36	𐀢 46	𐀣 56
𐀤 7	𐀥 17	𐀦 27	𐀧 37	𐀨 47	𐀩 57
𐀪 8	𐀫 18	𐀬 28	𐀭 38	𐀮 48	𐀯 58
𐀰 9	𐀱 19	𐀲 29	𐀳 39	𐀴 49	𐀵 59
𐀶 10	𐀷 20	𐀸 30	𐀹 40	𐀺 50	

Numeração na base 12, usada pelos babilônicos

424000 na Babilônia

Eis o modo como os babilônios escreviam 42400 que na sua base 12, seria 1,57.46,40.



De qualquer forma, fica até hoje, como herança daqueles povos a divisão do ano em 12 meses, do dia em 12 + 12 horas, das horas em 60 minutos e a própria dúzia.

Os circuitos eletrônicos não possuem dedos. É evidente também que não seria muito fácil projetar circuitos que sejam capazes de reconhecer 10 níveis de uma tensão, ou de outra grandeza elétrica, sem o perigo de que, qualquer pequeno problema os leve a uma confusão.

Uma pequena variação da tensão nestes circuitos pode mudar um 3 para 4 ou vice-versa, afetando os cálculos que ele tenha que realizar.

Muito mais simples para os circuitos eletrônicos é trabalhar com um sistema de numeração que esteja mais de acordo com o seu princípio de funcionamento e isso realmente é feito.

Um circuito eletrônico pode ter ou não ter corrente, pode ter ou não ter tensão, pode receber ou não um pulso elétrico.

Também é muito mais fácil diferenciarmos dois estados de elementos indicadores como uma lâmpada acesa ou apagada, uma campainha em silêncio ou tocando.

Ora, os circuitos eletrônicos são mais apropriados para operar com sinais que tenham duas condições possíveis, ou seja, que representem dois dígitos ou algarismos.

Também podemos dizer que as regras que regem o funcionamento dos circuitos que operem com apenas duas condições possíveis são muito mais simples.

Houve época em que se tentou trabalhar com as quantidades na forma original analógica, com a criação de computadores capazes de realizar cálculos complexos, mas com o tempo ficou claro que trabalhar com duas condições possíveis apenas para os circuitos, adotando uma lógica digital, era muito mais vantajoso, por diversos motivos.

Assim, o sistema adotado nos circuitos eletrônicos digitais modernos é o sistema binário ou de base 2 onde apenas dois dígitos são usados, correspondentes a duas condições possíveis de um circuito: 0 e 1.

Mas, como podemos representar qualquer quantidade usando apenas dois algarismos?

A idéia básica é a mesma usada na representação de quantidades no sistema decimal: atribuir pesos aos dígitos conforme sua posição no número.

Para entendermos melhor como tudo isso funciona, vamos tomar como exemplo o valor 1101 que em binário representa o número 13 decimal (*) e ver como isso ocorre.

O primeiro dígito da direita nos indica que temos uma vez o

Computadores analógicos

Usando amplificadores operacionais (veja Volume 2 – Eletrônica Analógica) foram construídos computadores capazes de realizar operações matemáticas com estes componentes. Estes computadores analógicos, originalmente foram criados para calcular tabelas de tiros, para aplicações militares. Na foto, um computador analógico vendido em kit na década de 50.



Computador analógico vendido em kit da década de 1950.

(*) Para não fazermos confusões em relação ao tipo de base que está sendo usada para representar um número ou quantidade, é comum colocarmos ao lado, como índice, a base que está sendo usada. Assim, ao falarmos em 1101 em binário, escrevemos simplesmente 1101_2 e para representar 13 em decimal, escrevemos 13_{10} . Esta forma de indicarmos as bases de um número, será adotada em nossas lições daqui por diante.

peso deste dígito ou 1. O zero do segundo dígito da direita para a esquerda, indica que não temos nada com o peso 2. Agora o terceiro dígito da direita para a esquerda, e que tem peso 4, é um 1, o que indica que temos “uma vez quatro”. Finalmente o primeiro dígito da esquerda que é um 1, e que está na posição de peso 8, nos diz que temos “uma vez oito”.

Somando uma vez oito, com uma vez quatro e uma vez um, temos o total que é justamente a quantidade que conhecemos em decimal como treze.

Veja então que, conforme mostra a figura 9, na numeração binária, os dígitos vão tendo pesos, da direita para a esquerda que são potencias de 2, ou seja, dois elevado ao expoente zero que é um, dois elevado ao expoente 1 que é 2; dois ao quadrado que é 4 e assim por diante.

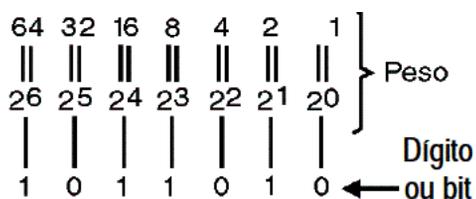


Figura 9 – Os pesos aumentam da direita para a esquerda, segundo potências de 2

Para o leitor, basta lembrar que a cada dígito que nos deslocamos para a esquerda, seu peso dobra, conforme mostra a figura 10.

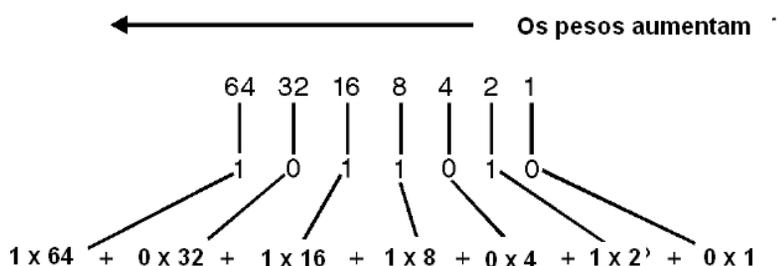


Figura 10 – Uma representação binária

Como não existe um limite para os valores dos pesos, isso significa que é possível representar qualquer quantidade em binário, por maior que seja, simplesmente usando a quantidade apropriada de dígitos.

Para 4 dígitos podemos representar números até 15; para 8 dígitos podemos ir até 255; para 16 dígitos podemos ir até 65 535 e assim por diante.

O leitor deve lembrar-se desses valores limites para 4, 8, 16 e 32 dígitos de um número binário, pois eles têm uma grande importância nas aplicações digitais modernas.

Tabela com potências de 2

Damos a seguir uma tabela com as potências de 2, até o expoente 30. Os valores encontrados nessa tabela são comumente usados em muitos cálculos envolvendo eletrônica digital.

Potência de 2	Valor
2 ⁰	1
2 ¹	2
2 ²	4
2 ³	8
2 ⁴	16
2 ⁵	32
2 ⁶	64
2 ⁷	128
2 ⁸	256
2 ⁹	512
2 ¹⁰	1024
2 ¹¹	2048
2 ¹²	4096
2 ¹³	8192
2 ¹⁴	16 384
2 ¹⁵	32 768
2 ¹⁶	65 536
2 ¹⁷	131 072
2 ¹⁸	262 144
2 ¹⁹	524 288
2 ²⁰	1 048 576
2 ²¹	2 097 152
2 ²²	4 194 304
2 ²³	8 388 608
2 ²⁴	16 777 216
2 ²⁵	33 554 432
2 ²⁶	67 108 864
2 ²⁷	134 217 728
2 ²⁸	268 435 456
2 ²⁹	536 870 912
2 ³⁰	1 073 741 824

A seguir damos a representação binária dos números decimais até 17 para que o leitor tenha uma idéia de como tudo funciona:

decimal	binário	decimal	binário
0	0	9	1001
1	1	10	1010
2	10	11	1011
3	11	12	1100
4	100	13	1101
5	101	14	1110
6	110	15	1111
7	111	16	10000
8	1000	17	10001
		...	etc.

Bit

Bit é uma contração de Binary Digit ou Dígito Binário, consistindo na menor quantidade de informação, ou na unidade de informação. Esta unidade é usada tanto em eletrônica digital como computação, em telecomunicações e na teoria da informação. O principal múltiplo do bit é o byte, que consiste no agrupamento de 8 bits. Também é utilizado em alguns casos o nibble que consiste no agrupamento de 4 bits. Os bytes também são chamados de octetos. Na tabela abaixo damos alguns múltiplos do bit e do byte, observando-se que 1 kb (quilo-bit) corresponde exatamente 1 000 bits pois os prefixos quilo, mega, etc são válidos para o SI na base 10. Para a base 2 houve uma nova padronização do IEC que pode ser encobrada no verbete “byte” (consulte). Na nova tabela, 1 quilobit é abreviado por 1 Kib e vale 1 024 bits, sendo o valor indicado para a base 2. Na tabela de potências de 2 dada mais adiante, pode-se ver os valores exatos de bits para os múltiplos.

Para o leitor que pretende entender de eletrônica digital aplicada aos computadores, equipamentos industriais, médicos e muitos outros existem momentos em que é preciso saber converter uma indicação em binário para o decimal correspondente.

Podemos dar como exemplo o caso de certas placas que são usadas no diagnóstico de computadores e de máquinas industriais as quais possuem um conjunto de LEDs que acendem indicando um número correspondente a um código de erros.

Os LEDs apagados indicam o algarismo 0 e os LEDs acesos indicam o algarismo 1, conforme mostra o exemplo da figura 11, em que temos uma placa com um display e um conjunto de 4 LEDs para indicação em binário.



Figura 11- Placa de diagnóstico com indicação em binário (4 LEDs)

Vamos supor que num diagnóstico a sequência de acendimento dos LEDs seja 1010. De início o leitor precisa saber por onde começar a leitura, ou seja, se o de menor peso é o da direita ou da esquerda.

Nas indicações dadas por instrumentos ou mesmo na representação da valores binários como, por exemplo, na saída de um circuito é preciso saber qual dos dígitos tem maior peso e qual tem menor peso. Isso é feito com uma sigla que é adotada normalmente e que refere-se ao dígito, no caso denominado bit.

Assim, para o dígito de menor peso ou bit menos significativo é adotada a sigla LSB (Least Significant Bit) e para o mais significati-

vo é adotada a sigla MSB (Most Significant Bit), conforme mostra a figura 12.

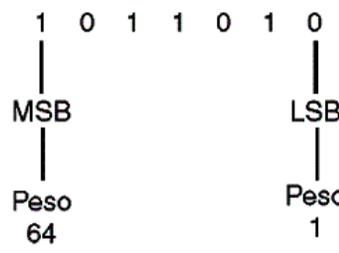


Figura 12 – As extremidade de um número binário

O que fazemos é somar os valores dados pelos dígitos multiplicados pelo peso de sua posição. No caso do valor tomado como exemplo, 1010110, temos:

Dígito x Peso = Valor

- 1 x 64 = 64
- 0 x 32 = 0
- 1 x 16 = 16
- 0 x 8 = 0
- 1 x 4 = 4
- 1 x 2 = 2
- 0 x 1 = 0

Somando os valores temos:

$$64 + 16 + 4 + 2 = 86$$

O valor decimal de 1010110 é 86, ou usando a notação mais apropriada:

$$1010110_2 = 86_{10}$$

Observe que, para realizar uma conversão, tudo que o leitor tem de fazer é lembrar que a cada dígito que saltamos para a esquerda seu peso dobra na sequência 1, 2, 4, 8, 16, 32, 64, 128, etc.

Na prática, também pode ocorrer o problema inverso que é a transformação de um valor expresso em decimal (base 10) para a base 2 ou binário.

Para esta transformação podemos fazer uso de um algoritmo muito simples que memorizado pelo leitor pode ser de grande utilidade, dada sua praticidade.

Vamos ver então como usar o algoritmo para a conversão de um decimal para binário como, por exemplo, o 116. O que fazemos é uma série de divisões sucessivas, conforme mostra a figura 13.

Algoritmo
 Algoritmo nada mais é do que uma sequência de operações simples que seguem uma determinada regra cuja finalidade é realizar uma operação mais complexa. Quando você soma os números um sobre o outro (da mesma coluna) ao fazer uma soma e passar para cima os dígitos que excedem o 10, fazendo o conhecido “vai um” você nada mais está fazendo do que usar um algoritmo. Os microprocessadores, microcontroladores, muitos circuitos digitais, máquinas industriais, equipamentos “inteligentes” usam muitos tipos de algoritmos quando fazem suas operações, se bem que a maioria não precisa ser conhecida dos leitores agora.

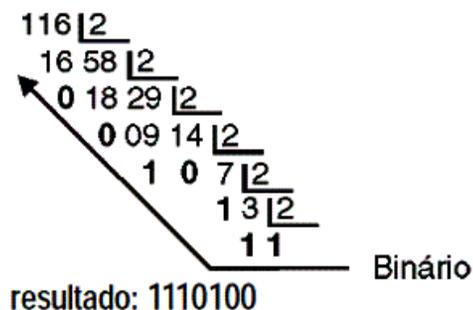


Figura 13 – Convertendo decimal para binário

Veja então que vamos dividindo os números por 2 até o ponto em que chegamos a um valor menor que 2 e que, portanto, não mais pode ser dividido.

O resultado desta última divisão, ou seja, seu quociente é então o primeiro dígito binário do número convertido. Os demais dígitos são obtidos lendo-se os restos da direita para a esquerda da série de divisões que realizamos. Tudo muito simples e rápido.

1.4.1 - Quilo, Mega e Giga

Em eletrônica digital, assim como na eletrônica analógica tradicional, é comum o uso de prefixos para representar quantidades muito grandes ou muito pequenas.

Dessa forma, os prefixos Quilo (k) para representar milhares, Mega (M) para representar milhões e Giga (G) para representar bilhões também são usados em eletrônica digital.

No entanto, diferentemente, do que ocorre com a eletrônica analógica, esses prefixos representam quantidades “levemente” diferentes.

O que ocorre é que quando usamos quilo (k) para representar 1 000 (mil), na base 10, 1000 é um valor inteiro, o que não ocorre com a base 2.

Assim, o valor mais próximo da base 2 é 1024. Isso significa que, em eletrônica digital, o valor quilo, na realidade representa 1024. Uma memória que tenha uma capacidade de armazenar 1 quilo bit ou 1 kb, na verdade, armazena 1024 bits.

Da mesma forma, quando falamos que um programa de computador exige um espaço de memória de 1 Mega bytes, na verdade, ele exige um espaço de 1 048 576 bytes, já que a potência de 2 mais próxima de 1 milhão é 1 048 576.

Em outras palavras, para informática e eletrônica digital o quilo (k) vale 1024, o mega (M) vale 1 048 576 e o Giga vale 1 073 741 824.

Abreviações

Abreviamos o bit com b e byte com B. Da mesma forma, lembre-se que quilo (e não kilo) é abreviado por k minúsculo.

Novos padrões de prefixos

Se bem que não sejam ainda muito usados existe uma nova padronização do IEC que procura eliminar a confusão que ocorre nesse caso. Esta padronização do IEC (International Electrotechnical Commission), datada do ano 2000, estabelece o uso dos prefixos para bits e bytes, conforme a tabela abaixo. Veja que os prefixos quilo, mega, tera, etc. são válidos apenas para a base 10 do SI. Para a base 2 existem novos prefixos que são usados conforme indica a tabela:

IEC para base 2 (binário)

Nome	Pref.	Valor
kibi	Ki	$2^{10} = 1024$
mebi	Mi	$2^{20} = 1\ 048\ 576$
gibi	Gi	$2^{30} = 1\ 073\ 741\ 824$
tebi	Ti	$2^{40} = 1\ 099\ 511\ 627\ 776$
pebi	Pi	$2^{50} = 1\ 125\ 899\ 906\ 842\ 624$
exbi	Ei	$2^{60} = 1\ 152\ 921\ 504\ 606\ 846\ 976$
zebi	Zi	$2^{70} = 1\ 180\ 591\ 620\ 717\ 411\ 303\ 424$
yobi	Ui	$2^{80} = 1\ 208\ 925\ 819\ 614\ 629\ 174\ 706\ 176$

1.5 – Binários menores que 1

Para o leitor, talvez seja difícil entender como usando quantidades que só podem ser inteiras, como dado pela definição de digital no início desta lição, seja possível representar quantidades menores que um, ou seja, números “quebrados” ou fracionários.

É claro que isso é possível na prática, pois se assim não fosse os computadores, equipamentos digitais de controle e automação e as calculadoras não poderiam realizar qualquer operação com estes números e sabemos que isso não é verdade.

O que se faz é usar um artifício que consiste em empregar potências negativas de um número inteiro, para representar quantidades que não são inteiras. Desta forma, é possível usar dígitos binários para representar quantidades fracionárias sem problemas.

Vamos dar um exemplo tomando como exemplo o número 0,01101 em binário. A própria existência de um “0,” já nos sugere que se trata de um número menor que 1 e portanto fracionário.

O que ocorre é que os dígitos deste número têm pesos que correspondem à potências de 2 negativas, e que nada mais são do que frações, conforme a seguinte sequência:

Dígito x Peso = Valor

$$0, \times 1 = 0$$

$$0 \times 1/2 = 0,5$$

$$1 \times 1/4 = 0,25$$

$$1 \times 1/8 = 0,125$$

$$0 \times 1/16 = 0,0625$$

$$1 \times 1/32 = 0,03125$$

Ponto e Vírgula Decimal

Alertamos que a separação da parte inteira da parte “quebrada” de um número, no nosso país é feita por uma “vírgula” enquanto que em outros países, como nos de língua inglesa, é feita por um ponto. Assim, falamos em vírgula decimal enquanto que eles falam em ponto decimal.

Somando os valores relativos temos:

$$0,25 + 0,0625 + 0,03125 = 0,29375$$

O número decimal representado é, portanto, 0,29375.

Veja que, usando tantos dígitos quantos sejam necessários podemos representar com qualquer precisão um número decimal menor que 1.

1.5.1 - Potências Negativas de 2

Na tabela dada a seguir temos algumas potências negativas de 2, usadas para representar números menores que 1.

Potência Negativa de 2	Decimal
2^0	1
2^{-1}	0,5
2^{-2}	0,25
2^{-3}	0,125
2^{-4}	0,062 5
2^{-5}	0,031 25
2^{-6}	0,015 625
2^{-7}	0,007 812 5
2^{-8}	0,003 906 25
2^{-9}	0,001 953 125
2^{-10}	0,000 976 562 5
2^{-11}	0,000 488 281 25
2^{-12}	0,000 244 140 625
2^{-13}	0,000 122 070 312 5
2^{-14}	0,000 061 035 156 25
2^{-15}	0,000 030 517 578 125

1.6 – Formas diferentes de usar o sistema binário

A utilização de circuitos eletrônicos com determinadas características, e a própria necessidade de se adaptar o sistema binário à representação de valores que sejam convertidos rapidamente para o decimal, e mesmo outros sistemas, levou ao aparecimento de algumas formas diferentes de utilização dos binários denominados “puros”.

Estas formas são encontradas em diversos tipos de equipamentos digitais incluindo os próprios computadores.

1.6.1 - Sistema BCD (Decimal Codificado em Binário)

BCD é a abreviação de Binary Coded Decimal e se adapta melhor aos circuitos digitais por características que ficarão mais claras no

decorrer de nosso curso. O que se faz é transformar cada dígito decimal de um número em um grupo de quatro dígitos binários (bits) independentemente do valor total do número que vai ser representado.

Assim, partimos da seguinte tabela:

Dígito decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Assim, se quisermos representar em BCD o número 23,25 não o convertemos da forma convencional por divisões sucessivas mas sim, tomamos cada dígito decimal e o convertemos no BCD equivalente conforme se segue:

2	3	2	5
0010	0011	0010	0101

Veja então que, para cada dígito decimal sempre teremos quatro dígitos binários ou bits e também que os valores 1010, 1011, 1100, 1101 e 1111 não existem neste código.

Esta representação foi muito interessante quando as calculadoras se tornaram populares, pois podia ser usada para todas as operações com números comuns e os 5 códigos que não eram usados dos valores que não existiam foram adotados para indicar as operações! (figura 14)

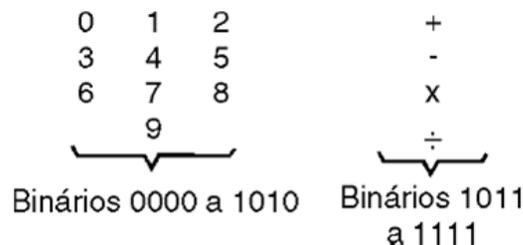


Figura 14 – Uso do código BCD

Usando representações desta forma, operavam os primeiros computadores que muito apropriadamente eram chamados de computadores de “4 bits”.

Computadores modernos

Os computadores modernos não se contentam com apenas 4 bits, pois eles permitem apenas a representação de 4 símbolos ou operações além dos algarismos de 0 a 9. Depois de passar pelos computadores de 8 bits, tivemos os de 16 bits e agora são comuns os de 32 bits

1.6.2 – Código Biquinário

Existem diversos outros códigos utilizados em equipamentos eletrônicos digitais que merecem destaque. Outro deles é o Código Biquinário.

Neste código temos 7 bits com pesos diferentes. Os dois primeiros bits, ou dígitos, indicam se o número está entre 0 e 5 ou entre 5 e 9.

Por exemplo, se o número é 7, temos nos dois primeiros bits 1 e 0, pois o número é maior que 5. Os outros bits indicam, quanto deve ser somado ao 5 para se obter o 7. Assim, usamos o bit de peso 2 para esta finalidade pois:

$$5 + 2 = 7$$

Temos então como representação do valor 7 neste código

$$\begin{array}{r} 10 \\ \underbrace{\quad} \\ 5 \end{array} + \begin{array}{r} 00100 \\ \underbrace{\quad} \\ 2 \end{array} = 7$$

A tabela abaixo nos dá as representações dos decimais de 0 a 9 neste código.

Decimal	Pesos 5043210
0	0100001
1	0100010
2	0100100
3	0101000
4	0110000
5	1000001
6	1000010
7	1000100
8	1001000
9	1010000

1.6.3 – Código Excesso 3 (XS3)

Podemos dizer que se trata de uma variação do BCD. Nele, a diferença é que a cada dígito representado, é acrescentado 3 ao valor correspondente em BCD, conforme a seguinte tabela.

Veja então que, para representar o valor 5, somamos 3, e com isso, usamos o BCD 8 (1000).

A vantagem do uso deste código está na facilidade com que ele pode ser usado em operações matemáticas.

Decimal	Exc-3	BCD
0	0011	0000
1	0100	0001
2	0101	0010
3	0110	0011
4	0111	0100
5	1000	0101
6	1001	0110
7	1010	0111
8	1011	1000
9	1100	1001

1.6.4 - Código Aiken ou 2421

Este código é utilizado para representar números decimais, mudando apenas o peso dos dígitos em relação ao BCD, conforme a seguinte tabela.

Decimal	2421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	1100
7	1101
8	1110
9	1111

Uma aplicação deste código está nos circuitos de sintonia em que podem ser usadas chaves para selecionar o valor da capacitância obtida numa associação de capacitores (Ver Volume 1 – Eletrônica Básica).

1.6.5 – Código Gray

O código Gray se caracteriza pelo fato da passagem de qualquer número para o seguinte sempre ser feita com a mudança de um único dígito.

Assim, por exemplo, quando passamos de 0111 (7 em decimal) para 1000 (8 em decimal) os quatro dígitos mudam. No código gray a passagem do 7 para 8 muda apenas um dígito pois o 7, 0100 e o 8, 1100.

A vantagem da mudança de apenas um dígito, na passagem de um número para o seguinte (ou volta ao anterior), está na possibili-

dade de se ter maior velocidade dos circuitos e menor consumo. Conforme veremos nas próximas lições, o consumo de um circuito digital ocorre na maior parte, no momento em que ele realiza uma operação lógica, ou seja, muda de estado.

Uma aplicação deste código está nos encoders ópticos, que são dispositivos ópticos em que se tem uma codificação através de zonas claras e escuras que permitem medir sua velocidade de rotação, ou ainda determinar sua posição, conforme mostra a figura 15.

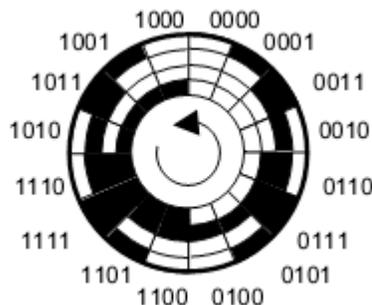


Figura 15 – Exemplo de aplicação do código Gray num encoder

Na tabela abaixo temos alguns valores decimais, e os correspondentes binários e no código Gray.

Decimal	Binário	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

1.6.6 – Código 987654321

No código 9876543210, temos uma decodificação de “uma saída de 10”, ou seja, um único bit será 1 enquanto os demais serão 0, conforme o dígito a ser representado, conforme a seguinte tabela:

Decimal	9876543210
0	0000000001
1	0000000010
2	0000000100
3	0000001000
4	0000010000
5	0000100000
6	0001000000
7	0010000000
8	0100000000
9	1000000000

A desvantagem deste código está no fato de precisarmos de 8 bits para representar os dígitos de 0 a 9.

1.6.7 - ASCII

Temos também códigos mais complexos como o ASCII (American Standard Code for Information Interchange), muito usado em informática.

Levando em conta que, com 8 bits podemos representar 256 valores diferentes, foi criado um código de 8 bits que correspondem aos principais símbolos (letras, números, sinais de pontuação, etc.) mais utilizados num sistemas de comunicações.

Assim, quando usamos o teclado do computador, a cada tecla está associado um byte correspondente a este código, o qual é decodificado e enviado ao computador. O mesmo ocorre, quando selecionamos uma tecla de letra, número ou outro símbolo gráfico no teclado do telefone celular, para digitar uma mensagem. Desta forma, quando digitamos a letra A, o byte enviado é 0100 0001.

Nesta tabela temos ainda a indicação dos valores correspondentes em binário no sistema de numeração base 16, ou hexadecimal, que estudaremos mais adiante.

Significado	Binário	HEXA	Significado	Binário	HEXA
Em branco	0010 0000	20	0	0011 0000	30
!	0010 0001	21	1	0011 0001	31
>>	0010 0010	22	2	0011 0010	32
#	0010 0011	23	3	0011 0011	33
\$	0010 0100	24	4	0011 0100	34
%	0010 0100	25	5	0011 0101	35
&	0010 0110	26	6	0011 0110	36
'	0010 0111	27	7	0011 0111	37
(0010 1000	28	8	0011 1000	38
)	0010 1001	29	9	0011 1001	39
-	0010 1010	2A	A	0100 0001	41
+	0010 1011	2B	B	0100 0010	42
,	0010 1100	2C	C	0100 0011	43
_	0010 1101	2D	D	0100 0100	44
.	0010 1101	2E	E	0100 0101	45
/	0010 1111	2F	F	0100 0110	46
:	0011 1010	3A	G	0100 0111	47
;	0011 1011	3B	H	0100 1000	48
<	0011 1100	3C	I	0100 1001	49
=	0011 1101	3D	J	0100 1010	4A
>	0011 1110	3E	K	0100 1011	4B
?	0011 1111	3F	L	0100 1100	4C
			M	0100 1101	4D
			N	0100 1110	4E
			O	0100 1111	4F
			P	0101 0000	50
			Q	0101 0001	51
			R	0101 0010	52
			S	0101 0011	53
			T	0101 0100	54
			U	0101 0101	55
			V	0101 0110	56
			W	0101 0111	57
			X	0101 1000	58
			Y	0101 1001	59
			Z	0101 1010	5A

Esta tabela contém os 128 primeiros códigos, havendo uma extensão para os códigos até 256.

Veja que nesta tabela também temos a indicação do símbolo em hexadecimal (hex) que é outro sistema que estudaremos no item 1.7, pela sua importância.

1.6.8 – Código Fortran

Nos primeiros computadores digitais, os programas eram digitados e passados a cartões perfurados do tipo Hollerith. Cada linha do programa correspondia a um cartão que depois eram empilhados consistindo assim no programa completo.

Levados a uma leitora, estes cartões eram “puxados” um a um para seu interior, onde depois de lidos, rodavam o programa apresentando o resultado numa folha impressa.

Na figura 16 temos um exemplo desses cartões perfurados com o código usado.

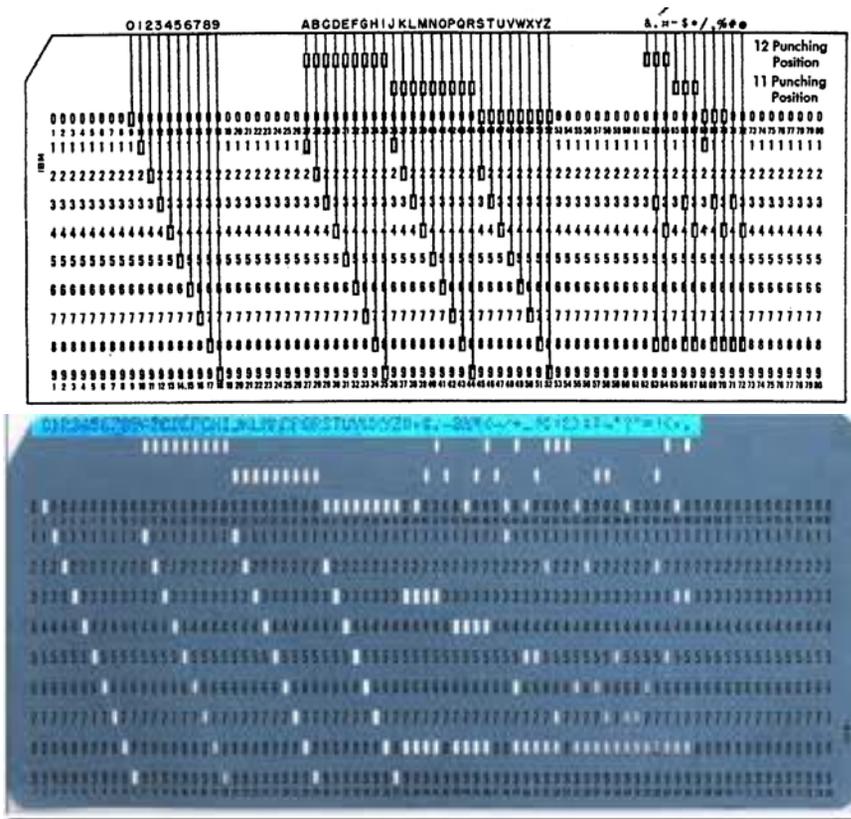
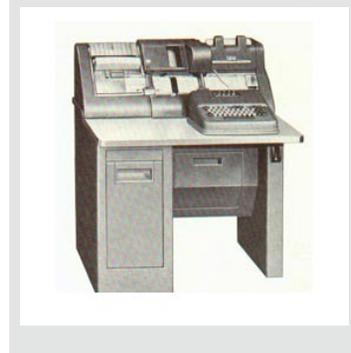


Figura 16 – Cartões Hollerith usados em computadores digitais antigos

Perfuradora

Na foto abaixo uma antiga perfuradora de cartões, usada em computadores antigos para "digitação" dos programas nos cartões Hollerith. O autor deste livro deu seus primeiros passos em programação, utilizando uma máquina destas em 1970 na Escola Politécnica da USP, onde aprendeu Fortran.



1.6.9 - Códigos de erros

Na transmissão de dados na forma digital, é de extrema importância garantir a integridade da informação. Para esta finalidade existem recursos que permitem saber se um número na forma binária foi transmitido corretamente ou não. Para isso são agregados códigos especiais, denominados códigos de erro.

1.6.10 – Binário com Paridade

Uma maneira de simples de se detectar a integridade de um valor binário transmitido em BCD consiste em se acrescentar um bit que indique a paridade. Assim, esse bit será 1, se a soma dos bits dos quatro bits transmitidos for ímpar e será 0, se a soma for par. Como exemplo, damos dois casos possíveis, onde p é o bit de paridade.

Códigos**muito complexos**

Além dos códigos que vimos, podemos contar com outros de grande complexidade como o código de Hamming. Este código permite detectar não apenas se houve a transmissão de um bit errado, como também, dentro do número digital transmitido, qual é esse bit.



Richard Hamming – criador do código que leva seu nome

decimal	BCD	p
0	0000	0
1	0001	1
2	0010	1
3	0011	0
4	0100	1
5	0101	0
6	0110	0
7	0111	1
8	1000	1
9	1001	0

1.6.11 - Código 2 de 5

Neste código existem sempre dois dígitos 1, conforme a seguinte tabela:

decimal	2 de 5
0	00011
1	00101
2	00110
3	01001
4	01010
5	01100
6	10001
7	10010
8	10100
9	11000

Desta forma, se numa transmissão de dados, aparecer um bloco de 5 bits com um número diferente de 2 de bits 1, então pode-se saber que os dados chegaram de forma incorreta.

1.7 - Sistema Hexadecimal

Os bits dos circuitos digitais são agrupados em conjuntos de 4 (nibble), desta forma temos os microprocessadores e computadores de 4, 8, 16 e 32 bits, etc.

Também observamos que, com 4 bits podemos obter representações binárias de 16 números, e não somente de 10. Vimos que os 5 excedentes poderiam ser usados para representar operações nas calculadoras.

Isso significa que, muito mais compatível com a numeração binária ou operação binária dos circuitos digitais, como os computadores, é a representação de valores no sistema hexadecimal ou de base 16.

E, de fato isso é feito: abrindo muitos programas de um computador vemos que suas características, como posições de memória ou quantidade de memória, são feitas neste sistema.

Isso significa que todos que trabalham com eletrônica digital precisam conhecer este sistema e, mais do que isso, saber como fazer conversões dele para o decimal e vice-versa além de conversões para o sistema binário.

Na tabela abaixo damos as representações dos dígitos deste sistema tanto com equivalentes decimais como binários:

Decimal	Binário	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Na figura 17 temos a apresentação dos valores de 0 a 15 em displays de 7 segmentos, utilizando a codificação hexadecimal.



Figura 17 – Displays de 7 segmentos apresentando numeração em hexadecimal.

Observe que, como não existem símbolos para de dígitos para 10,11,12,13,14 e 15, foram usadas as letras A,B,C,D,E e F.

Como fazer as conversões: os mesmos procedimentos que vimos para o caso das conversões de decimal para binário e vice-versa são válidos para o caso dos hexadecimais mudando-se apenas a base.

Vamos dar exemplos:

Como converter 4D5 em decimal:

Os pesos no caso são: 256, 16 e 1. (a cada dígito para a esquerda multiplicamos o peso do anterior por 16 para obter novo peso).

Temos então:

$$4D5 = (4 \times 256) + (13 \times 16) + (1 \times 5) = 1237$$

Observe que o “D” corresponde ao 13. O número decimal equivalente ao 4D5 hexadecimal ou “hex” como é muitas vezes representado, é 1237.

$$4D5 \text{ (hex)} = 1237 \text{ (dec)}$$

A conversão inversa, ou seja, de decimal para hexadecimal é feita por divisões sucessivas. Tomemos o caso de 1256, mostrado na figura 18.

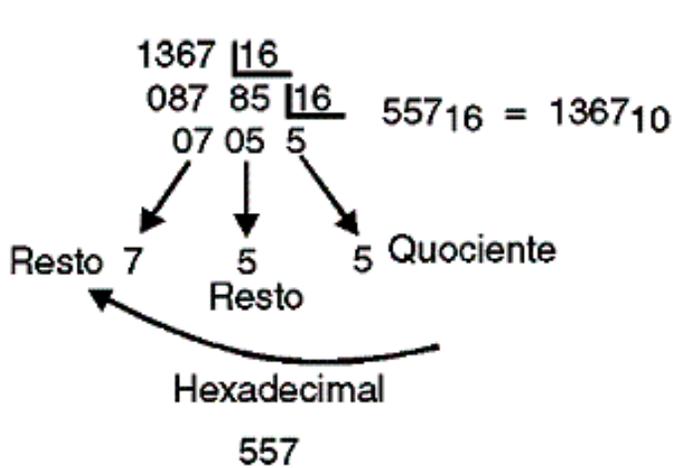


Figura 18 – Convertendo o decimal 1367 em hexadecimal

Veja então que basta ler o quociente final e depois os restos das divisões sucessivas, sempre lembrando que os que excederem 10, devem ser “trocados” pelas letras equivalentes.

Na tabela dada a seguir temos algumas potências de 16 convertidas para decimal.

Potência de 16	Decimal
16^0	1
16^1	16
16^2	256
16^3	4096
16^4	65 536
16^5	1 048 576
16^6	16 777 216

1.8 – Aritmética Binária

Da mesma forma que podemos realizar operações com números decimais, podemos também somar, subtrair, multiplicar ou dividir números binários.

Para trabalhar com números binários os procedimentos (algoritmos) são bastante semelhantes aos usados com os números decimais e, evidentemente, podemos aplicar estes procedimentos a números em qualquer base.

Para realizar operações aritméticas com números binários, basta lembrar algumas regras básicas que são dadas a seguir:

As operações com números binários seguem certas regras que são dadas a seguir:

a) Soma:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ (vai um)}$$

b) Subtração:

$$0 - 0 = 0$$

$$0 - 1 = 1 \text{ (empresta 1)}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

c) Multiplicação:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

d) Divisão:

$$0 : 0 = 0$$

$$0 : 1 = 0$$

$$1 : 0 = \text{não permitido}$$

$$1 : 1 = 1$$

Funções lógicas

Conforme veremos, existem circuitos digitais que contêm funções lógicas básicas e estas funções são capazes de realizar somas, multiplicações e outras operações com números binários. Graças a estas funções é que podemos implementar circuitos capazes de realizar operações lógicas ou aritméticas.

Vamos tomar como exemplo a soma binária de 6 (0110) com 7 (1110). Colocando um número sobre o outro, como na figura 19 (a).

Somando a última coluna da direita, seguindo as regras dadas acima, temos que $0 + 1 = 1$ e “vai 0” para ser transportado para a segunda coluna, como mostra a figura 19 (b)

A seguir somamos os três dígitos da segunda coluna: $0 + 1 + 1 = 1$ e “vai 1”, que é transportado para a terceira coluna, como vemos na figura 19 (c).

Somamos agora os três dígitos da terceira coluna $1 + 1 + 1 = 1$ e “vai 1”, que é transportado para a quarta coluna, conforme mostra a figura 19 (d).

Finalmente somamos os dígitos da quarta coluna: $1 + 0 + 0 = 1$ e “vai 0”. Com isso chegamos resultado final na figura 19(e)

É fácil então converter 1101, que é o resultado final, em decimal, chegando ao resultado esperado: 13.

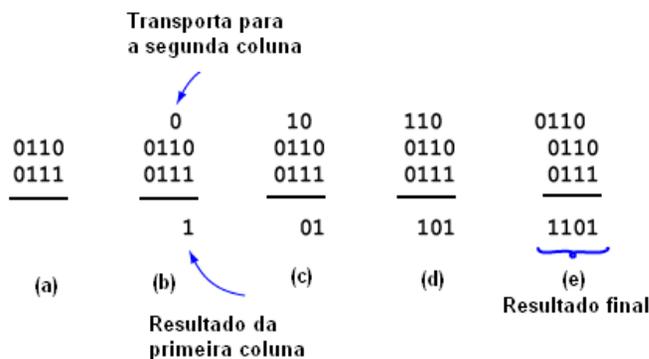


Figura 19 – A soma binária

Um exemplo de subtração binária é dado a seguir. Observe que na primeira coluna, como não podemos tirar 1 de 0, “emprestamos 1” da segunda coluna (da direita para a esquerda), exatamente como fazemos com os números decimais. Na figura 20 temos um exemplo de subtração binária.

$$\begin{array}{r}
 10010 = 18 \\
 - \\
 1001 = 9 \\
 \hline
 01001 = 9
 \end{array}$$

Figura 20 – Uma subtração binária

Para a multiplicação, podemos dar o seguinte exemplo, observando, que da mesma forma que na multiplicação decimal, multiplicamos os números da primeira linha por cada um da segunda linha, conforme mostra a figura 21.

$$\begin{array}{r}
 1101 = 7 \\
 101 = 5 \\
 \hline
 1101 \\
 0000 \\
 1101 \\
 \hline
 1000001 = 35
 \end{array}$$

Figura 21 – A multiplicação binária

Finalmente podemos dar um exemplo de divisão binária na figura 22:

$$\begin{array}{r}
 1011011 \quad | \quad 111 \\
 1000 \quad \quad 1101 \\
 00111 \\
 000
 \end{array}$$

Figura 22 – Uma divisão binária

Existem processos de cálculos simples que são feitos utilizando-se os seus complementos. Os complementos de um número são a diferença entre a base e o número. Por exemplo:

- O complemento de 3 na base 10 é 7
- O complemento de 1 na base 2 é 1
- O complemento de 2 na base 2 é 0

Na próxima lição veremos como estas operações podem ser utilizadas de maneira mais efetiva através do que denominamos Álgebra Booleana.

Leibniz

Descobridor da aritmética binária. Ainda jovem, ele escreveu sobre a numeração binária e suas operações. Ele relacionou os caracteres do I Ching com a numeração binária até 11111. Muitas outras descobertas no campo da matemática foram feitas por ele.



Gottfried Leibniz
(Julho 1646 - Junho 1716)

Termos em inglês

Os termos utilizados em eletrônica digital muitas vezes são mantidos na sua forma original em inglês na literatura técnica em português, se bem que em muitos casos tenham tradução. Vejamos alguns mais importantes.

Analog – analógica

Logic – lógica

Quantities – grandezas

Numbering system – sistema de numeração

Code – código

Binary – binário

BCD- Binary Coded Decimal – Decimal Codificado em Binário

Power of two – potência de 2

Sum – soma

Subtraction – subtração

Termos para pesquisa:

- Sistemas de numeração
- A origem dos números
- Leibniz
- Operações com números binários
- Algoritmos
- Códigos binários
- ASCII

EXERCÍCIOS

1. Convertendo 64 em BCD obtemos:

- a) 0010 1000
- b) 0110 0010
- c) 0010 0101
- d) 0110 0101

2. Convertendo 01001 para decimal obtemos:

- a) 9
- b) 10
- c) 14
- d) 6

3. A que número decimal corresponde o BCD 1000 0001?

- a) 129
- b) 65
- c) 81
- d) 64

4. O número F1 em hexadecimal corresponde a:

- a) 32
- b) 17
- c) 43
- d) 128

5) No Código de Gray, o valor 7 corresponde a:

- a) 0100
- b) 1000
- c) 0111
- d) 1010

6) O quinto bit num código que detecta erros é denominado:

- a) LSB
- b) MSB
- c) Bit de paridade
- d) Bit de erro

7) Podemos trabalhar com complementos de um número binário para:

- a) Aumentar a precisão das operação
- b) Reduzir erros
- c) Facilitar as operação aritméticas
- d) Aumentar a velocidade das operações



» A Álgebra de Boole

No capítulo anterior vimos as diferenças entre a eletrônica analógica e a eletrônica digital e as suas principais aplicações. Vimos também como as quantidades podem ser representadas com apenas dois dígitos, no sistema denominado binário e como é possível realizar operações aritméticas com eles. Vimos também que existem diversas formas de se codificar os números binários. Neste capítulo veremos como as operações com os números na forma digital podem ser feitas, com as regras estabelecidas pela Álgebra de Boole. Veremos ainda a representação destas funções tomando contato com a simbologia usada em eletrônica digital. Este capítulo conta então com os seguintes itens:

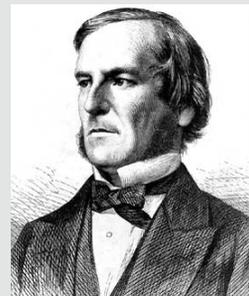
- 2.1 – A Álgebra de Boole
- 2.2 – Os níveis lógicos
- 2.3 – Operações lógicas
- 2.4 – Função lógica Não (inversora)
- 2.5 – Função lógica E (AND)
- 2.6 – Função lógica OU (OR)
- 2.7 – Função lógica Não-E (NAND)
- 2,8 – Função lógica Não-OU (NOR)
- 2.9 – Função lógica OU-Exclusivo (Exclusive OR)
- 2.10 – Função lógica Não-OU-Exclusivo (Exclusive NOR)
- 2.11- Propriedades das operações lógicas
- 2.12 – Fazendo tudo com funções NAND

Objetivos:

- Entender como operações lógicas podem ser feitas com circuitos eletrônicos
- Conhecer a álgebra de Boole
- Familiarizar-se com as principais operações lógicas
- Conhecer a simbologia usada na representação das funções lógicas

George Boole

Matemático britânico (irlandês) que desenvolveu sua álgebra apresentando-a num livro chamado "Investigações sobre as leis do pensamento", publicado em 1854. Neste livro ele tratava pela primeira vez do que se denominou lógica simbólica.



George Boole – 1815 - 1864

Outras Lógicas

Se bem que a eletrônica digital moderna se baseie quase que totalmente na álgebra de Boole, ela não é a única a encontrar aplicações práticas. Existem, por exemplo, circuitos especiais denominados “inteligentes” que se baseiam em outros tipos de lógica.

Uma dessas lógicas que encontra uma aplicação moderna em dispositivos de controle, robótica e que agregam o que se denomina “inteligência artificial” aos dispositivos, é a denominada “Lógica Fuzzy” ou “Lógica Difusa”. Nela, além dos dois estados possíveis, admite-se um terceiro como “pode ser, mais ou menos, ou sei lá”.

2.1 - A álgebra de Boole

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

No entanto, se bem que a Álgebra de Boole, como foi chamada, pode resolver problemas práticos de controle, fabricação de produtos, instrumentação, etc., na época não havia eletrônica e nem mesmo as máquinas eram avançadas o suficiente para utilizar seus princípios.

A Álgebra de Boole veio a se tornar importante com o advento da Eletrônica, especificamente da eletrônica digital que gerou os modernos computadores, equipamentos digitais e de telecomunicações com que estamos familiarizados.

Boole estabelece em sua teoria que só existem no universo duas condições possíveis ou estados, para a análise de qualquer situação, e estes dois estados são opostos.

Assim, uma lâmpada só pode estar acesa ou apagada; uma torneira só pode estar aberta ou fechada; uma fonte só pode ter ou não ter tensão na sua saída; uma pergunta só pode ter como resposta verdadeiro ou falso (sim ou não).

Dizemos de uma maneira simples que, na Álgebra de Boole, as variáveis lógicas só podem adquirir dois estados:

0 ou 1

Verdadeiro ou Falso

Aberto ou Fechado

Alto ou Baixo (HI ou LO)

Ligado ou Desligado

Na eletrônica digital, partimos justamente do fato de que um circuito só pode trabalhar com dois estados possíveis, ou seja, podemos encontrar neste circuito dois tipos de sinais: presença do sinal ou ausência do sinal o que faz com que ela se adapte perfeitamente aos princípios da Álgebra de Boole.

Tudo que um circuito lógico digital pode fazer está previsto pela Álgebra de Boole. Desde as mais simples operações ou decisões, como acender um LED quando dois sensores são ativados de determinada forma, ou quando uma tecla é pressionada até, girar no espaço uma imagem tridimensional.

Partindo de princípios simples, com regras muito bem definidas, Boole mostrou que desde operações simples como somar e subtrair até as mais complexas, que envolvem os cálculos realizados pelos computadores, podem ser realizadas com base no conceito de sim ou não ou ausência ou presença de um sinal.

2.2 - Os níveis lógicos

Partimos então do fato de que nos circuitos lógicos digitais só podemos encontrar duas condições possíveis: presença e ausência de sinal. Fazemos isso para definir alguns pontos importantes para o nosso entendimento.

Nos circuitos digitais, a presença de uma tensão será indicada como 1 ou HI (de HIGH ou Alto), enquanto que a ausência de uma tensão será indicada por 0 ou LO (de LOW ou baixo).

O 0 ou LO será sempre uma tensão nula, ou ausência de sinal num ponto do circuito, mas o nível lógico 1 ou HI pode variar conforme o circuito considerado, conforme mostra a figura 23.

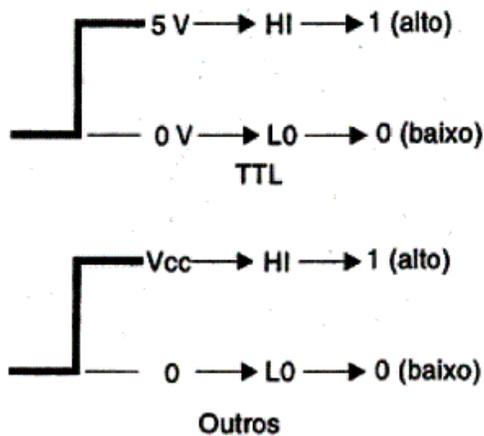


Figura 23 – Nos circuitos lógicos temos apenas dois níveis de tensão possíveis

Na maioria de equipamentos digitais, denominados TTL, a tensão usada na alimentação de todos os circuitos lógicos é de 5 volts, assim o nível 1 ou HI de seus circuitos será sempre uma tensão de 5 volts.

No cerne de alguns processadores e em outras aplicações alimentadas por bateria pode ser encontrada uma tensão de alimentação menor, da ordem de 3,3 volts e até menos, o que significa que nestes circuitos um nível 1 ou HI sempre corresponderá a uma tensão desse valor.

Existem ainda circuitos digitais que empregam componentes de tecnologia CMOS e que podem ser alimentados por tensões entre 3 e 15 volts tipicamente. Nestes casos, conforme mostra a figura 24, um nível lógico 1 ou HI poderá ter qualquer tensão entre 3 e 15 volts, dependendo apenas a alimentação usada.

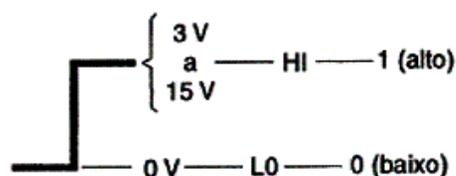


Figura 24 – Níveis de tensão em alguns tipos de circuitos digitais

Álgebra Clássica e Álgebra de Boole

O que diferencia a Álgebra Clássica da Álgebra de Boole é que, enquanto a primeira trabalha com relações quantitativas a segunda, trabalha com relações lógicas.

Na verdade, a idéia de se associar a presença de tensão ao nível 1 a ausência ao nível 0 é uma mera questão de convenção. Nada impede que adotemos um critério inverso e projetemos os circuitos sem que isto impeça seu funcionamento normal.

E isto realmente existe:

Assim, quando dizemos que ao nível alto (1) associamos a presença de tensão e ao nível baixo a ausência de tensão (0) estamos falando do que se denomina “lógica positiva”.

Se associarmos o nível baixo ou 0 a presença de tensão e o nível alto ou 1 à ausência de tensão, estaremos falando de uma “lógica negativa”, conforme mostra a figura 25.

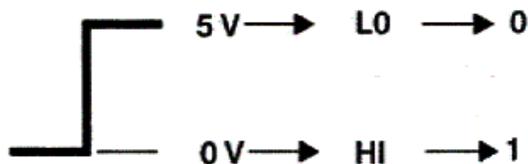


Figura 25 – Exemplo de lógica negativa

Para não causar nenhum tipo de confusão nos nossos leitores, todo o nosso curso tratará exclusivamente da lógica positiva, assim como dos dispositivos eletrônicos tomados como exemplo.

Veja então que, na nossa lógica podemos associar os seguintes estados de um circuito aos valores 0 e 1:

0 volt
Falso
Desligado
Nível baixo ou LO (De low, “baixo” em inglês.)

5 volts (ou outra tensão positiva, conforme o circuito)
Verdadeiro
Ligado
Nível alto ou HI (De high, “alto” em inglês.)

2.3 - Operações Lógicas

No dia-a-dia estamos acostumados a realizar diversos tipos de operações algébricas, sendo as mais comuns as que envolvem números, ou seja, quantidades que podem variar ou variáveis. Assim, estamos muito mais acostumados a trabalhar com a Álgebra Clássica.

Partindo desta idéia, podemos representar uma soma como:

$$Y = A + B$$

Onde o valor que vamos encontrar para Y depende dos valores que vamos atribuir as letras A e B.

Dizemos que temos neste caso uma função algébrica e que o valor Y é a variável dependente, pois seu valor vai depender justamente dos valores de A e B são as variáveis independentes.

Na eletrônica digital, entretanto, existem operações mais simples do que a soma, e que podem ser perfeitamente implementadas levando em conta a utilização da Álgebra Booleana.

É interessante observar que, com um pequeno número destas operações podemos chegar a uma infinidade de operações mais complexas como, por exemplo, as utilizadas nos microprocessadores, microcontroladores, e muitos outros circuitos digitais que, repetidas em grande quantidade, ou levadas a um grau de complexidade muito grande, nos fazem até acreditar que a máquina seja “inteligente”!

Na verdade, é a associação de determinada forma das operações simples que nos leva ao comportamento muito complexo de muitos circuitos digitais, conforme mostra a figura 26.

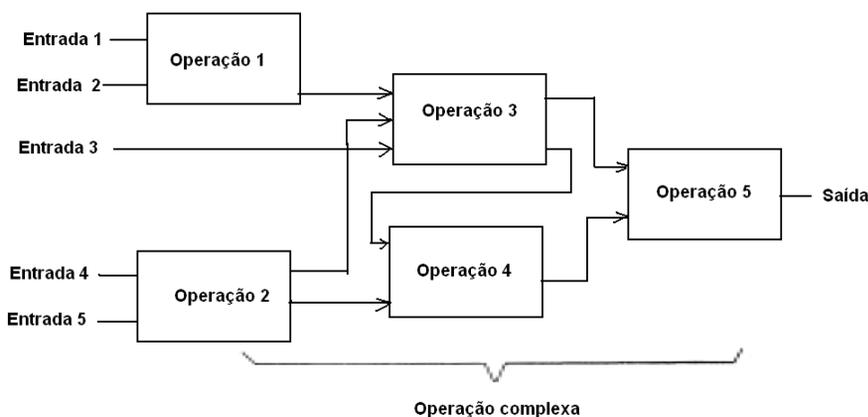


Figura 26 – Diversas operações simples resultam numa operação complexa

Partindo de um exemplo, podemos tomar um circuito de processamento digital que é formado por uma grande quantidade de pequenos blocos, denominados portas ou funções, como mostra a figura 27, em que temos entradas e saídas.

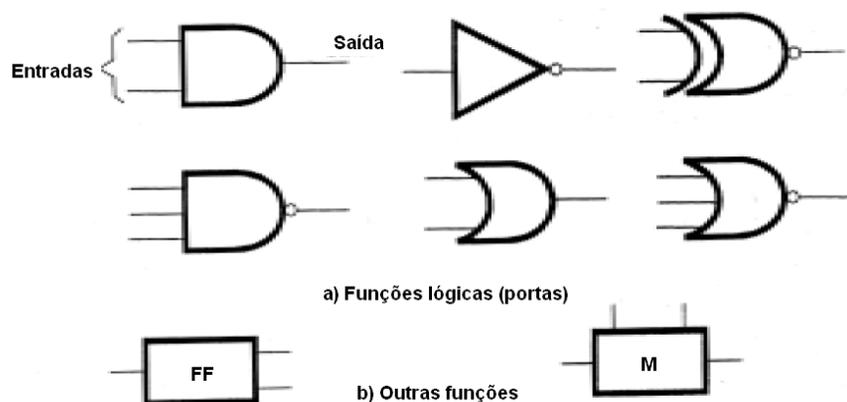


Figura 27 – Blocos contendo portas ou funções

O que vai aparecer na saída é determinado pela função e pelo que acontece nas entradas. Em outras palavras, a resposta que cada circuito lógico, para uma determinada entrada ou entradas, depende do que ele é ou de que “regra booleana” ele segue.

Isso significa que, para entender como os circuitos digitais funcionam desde os mais simples até os equipamentos que possuem um microprocessador que realizam as mais complexas operações, teremos de começar entendendo como ele faz as operações mais simples com as denominadas portas e quais são elas.

Por este motivo, depois de definir estas operações lógicas, associando-as à Álgebra de Boole, vamos estudá-las uma a uma.

2.3.1 – Raciocínio Lógico e notações

No trabalho com eletrônica digital, quando tratamos do funcionamento dos circuitos básicos, como estamos fazendo aqui, é importante que o leitor se familiarize com o jargão usado ao se tratar do raciocínio lógico.

Assim as questões relacionadas com o raciocínio lógico são sempre formadas por proposições que provam, dão razão ou suporte, consistindo em afirmações que expressam um pensamento de sentido denominado completo.

Essas proposições, conforme vimos ao tratar da Álgebra de Boole, têm, sempre duas possibilidades: elas podem ter um sentido positivo ou negativo.

Podemos então dizer:

José acende uma lâmpada (sentido positivo)

Mário não usa o osciloscópio (sentido negativo)

Os dois exemplos caracterizam uma afirmação ou proposição.

Lógica e Matemática

É importante que o aluno observe a íntima relação que os circuitos digitais possuem com a matemática. Dessa forma, um bom profissional da eletrônica digital deve obrigatoriamente ter um bom conhecimento de matemática. Verificamos hoje, que no campo da programação e de computadores, por exemplo, uma boa parte dos profissionais tem formação em matemática.

Partindo daí, chegamos à base das estruturas lógicas que consiste em saber o que é verdade ou mentira, ou usando um jargão mais apropriado, o que é verdadeiro ou falso.

Assim, existem alguns princípios básicos que regem as proposições, cujos resultados devem ser sempre os mesmos.

a) Princípio da contradição

Nenhuma proposição pode ser, ao mesmo tempo, falsa e verdadeira.

b) Terceiro excluído

Dadas duas proposições lógicas contraditórias, somente uma delas é verdadeira. Uma proposição ou é verdadeira ou falsa, não existindo um terceiro valor lógico. Para a proposição “ele usa o osciloscópio” ou “ele não usa o osciloscópio”, não existe um meio termo.

No estudo da lógica, e também da eletrônica digital, é comum o uso dos Conectivos Lógicos. Estes são símbolos que unem das proposições uma a outra, ou ainda as transformam numa terceira proposição. Os principais Conectivos Lógicos são: o:

- (~) “não”: negação.
- (\wedge) “e”: conjunção.
- (\vee) “ou”: disjunção.
- (\rightarrow) “se...então”
condicional (\leftrightarrow).
- “se e somente se”: bicondicional.

Todos estes conectivos vão se traduzir em funções lógicas, ou circuitos que funcionam seguindo as leis da lógica que veremos a partir de agora.

Veja que existe uma notação para a lógica que é diferente da usada em eletrônica digital e também em matemática. Ao estudar as funções, a partir de agora, para facilitar o entendimento dos leitores, incluindo os que têm formação em matemática, daremos as notações possíveis.

A notação matemática vem da teoria dos conjuntos. Trata-se do ramo da matemática que estuda os conjuntos ou coleções de elementos.

Esta teoria foi inicialmente estudada por Georg Cantor e Richard Dedekind em 1870 e hoje é adotada nos cursos de matemática do mundo inteiro.

A lógica de classes, que veremos mais adiante, pode ser considerada um pequeno fragmento da teoria dos conjuntos.

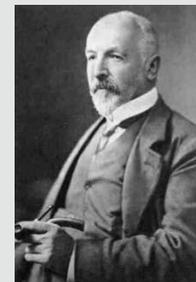
Nessa lógica, temos as seguintes notações equivalentes para a lógica e para a teoria dos conjuntos:

$$A \cap B \text{ equivale a } a \wedge b. \text{ (A intersecção B)}$$

$$A \cup B \text{ equivale a } a \vee b. \text{ (A união B)}$$

Georg Cantor

Matemático russo, que estudando na Suíça, trabalhou na teoria dos números, chegando ao que hoje se denomina teoria dos conjuntos, de grande importância para a matemática.



Georg Ferdinand Ludwig Cantor (1845 – 1918)

Símbolos

Para representar as funções lógicas, a notação usada em matemática que vimos acima não é a mesma usada em lógica. Ao estudar as funções, a seguir, tomaremos contato com as representações mais usadas em eletrônica digital.

$A \subset B$ equivale a $a \rightarrow b$.
 $a \in P$ equivale a $P(a)$.
 \emptyset equivale a F (Falso).
 U equivale a V (Verdadeiro).
 $\setminus A$ equivale a $\sim a$.

Tabela verdade

As tabelas verdade mostram o que ocorre com a(s) saída(s) de uma função lógica ou mesmo de um conjunto de funções que realizam operações lógicas em função de sua entrada. Nela são combinadas todas as condições possíveis das entradas e o que ocorre em sua função com a(s) saída(s).

2.4 - Função Lógica NÃO ou Inversora

Encontramos também nos manuais a indicação desta função com a palavra inglesa correspondente que é NOT.

O que esta função faz, é negar uma afirmação, ou seja, como em Álgebra Booleana só existem duas respostas possíveis para uma pergunta, esta função “inverte” a resposta, ou seja, a resposta é o “inverso” da pergunta. O circuito que realiza esta operação é denominado inversor.

Se a entrada for “sim” ou nível lógico 1, a saída será “não”, ou nível lógico 0.

Levando em conta então que este circuito diz sim, quando a entrada é não, ou que apresenta nível 0, quando a entrada 1 e vice versa, podemos associar a ele uma espécie de tabela que será de grande utilidade sempre que estudarmos qualquer tipo de circuito ou função lógica.

Esta tabela nos mostra o que ocorre com a saída da função quando colocamos na entrada todas as combinações possíveis de níveis lógicos.

Dizemos que se trata de uma “tabela verdade” (nos manuais em inglês esta tabela aparece com o nome de “Truth Table”), e para a porta NOT ou inversora ela é mostrada a seguir:

Entrada	Saída
0	1
1	0

Os símbolos adotados para representar esta função são mostrados na figura 28.

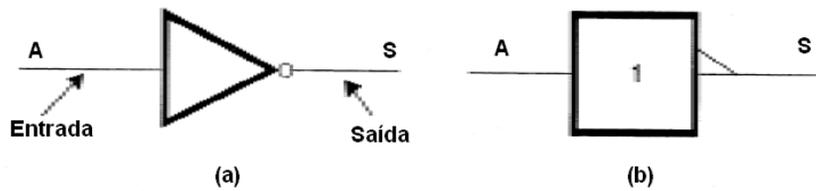


Figura 28 – Símbolos para a função NÃO – em (a) o mais comum e em (b) a simbologia adotada pelo IEEE.

O adotado normalmente em nossas publicações, e que será usado neste curso, é o mostrado (a), mas existem muitos manuais técnicos e mesmo diagramas em que são adotados outros. Os leitores devem conhecer todos, dada a possibilidade de se ter em mãos manuais de equipamentos de procedências as mais diversas.

Sendo A a entrada e A a saída podemos escrever a equação booleana do inversor como:

$$S = \overline{A}$$

Onde: S é a saída
A a entrada

Esta função pode ser simulada por um circuito simples que o leitor pode facilmente entender e que é mostrado na figura 29.

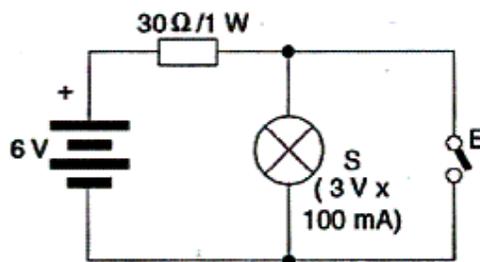


Figura 29 – Circuito que simula uma função NÃO

O mesmo circuito, utilizando um LED é mostrado na figura 30.

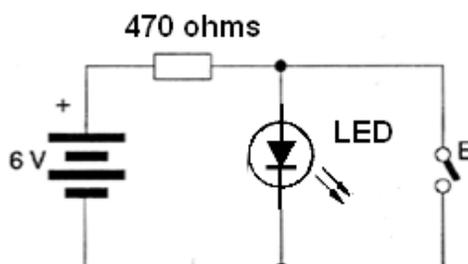


Figura 30 – Circuito Equivalente com LED

Neste circuito temos uma lâmpada, que acesa, indica o nível 1 na saída, e apagada, indica o nível 0 na saída. Quando a chave está aberta, indicando que a entrada é nível 0, a lâmpada está acesa indicando que a saída é nível 1.

Por outro lado, quando a chave é fechada o que representa uma entrada 1, a lâmpada apaga indicando que a saída é 0.

Esta maneira de se simular funções lógicas com lâmpadas (ou LEDs) indicando a saída e chaves indicando a entrada é bastante usada em cursos técnicos com finalidade didática pela facilidade com que podemos entender seu funcionamento.

Existem outros tipos de circuitos simples que podem ser usados para implementar uma porta com finalidade didática, usando relés, transistores ou outros componentes de fácil obtenção.

Basta então lembrar que:

Entrada: Chave aberta = 0

Chave fechada = 1

Saída: lâmpada apagada = 0

lâmpada acesa = 1

Na figura 31 temos um circuito com relé para simular esta função. A carga pode ser um LED em série com um resistor de 470 ohms ou uma lâmpada e as baterias podem ser de 6 V.

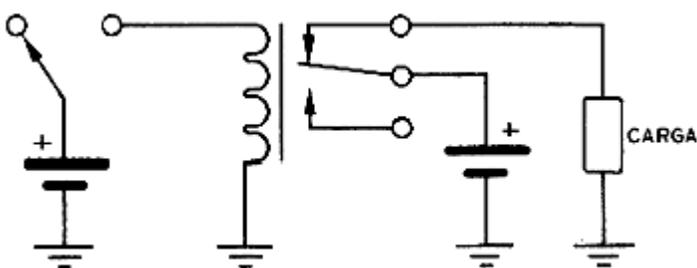


Figura 31 - Função inversor com relé

Conectivos lógicos

Será muito importante observar sempre na descrição do funcionamento das funções lógicas o uso dos conectivos lógicos como o "se, e somente se" indicando uma condição única que deve ser satisfeita na sua operação.

2.5 - Função Lógica E (AND)

A função lógica E, também conhecida pelo seu nome em inglês AND, pode ser definida como aquela em que a saída será 1 se, e somente se, todas as variáveis de entrada forem 1.

Veja que neste caso, as funções lógicas E podem ter duas, três, quatro, ou quantas entradas quisermos, sendo representada pelos símbolos mostrados na figura 32.

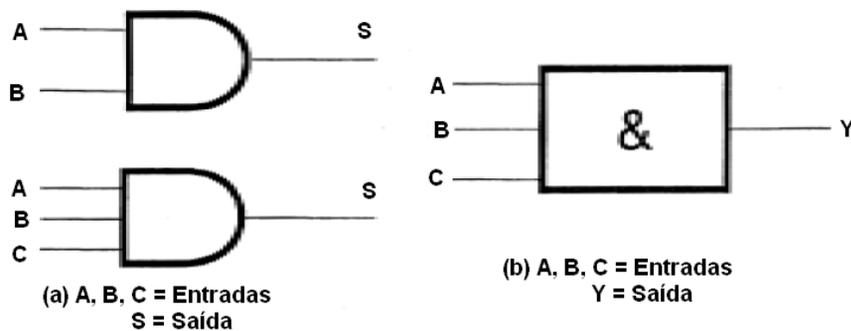


Figura 32- Os símbolos da função lógica E (AND)

As funções lógicas também são chamadas de “portas” ou “gates” (do inglês), já que correspondem a circuitos que podem controlar ou deixar passar os sinais sob determinadas condições.

Tomando como exemplo uma porta ou função E de duas entradas, podemos escrever a seguinte tabela verdade:

ENTRADAS		SAÍDA
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

A equação Booleana para esta função é:

$$S = A.B$$

Onde: S é a saída

A e B são as entradas

Na figura 33 mostramos o modo de se simular o circuito de uma porta E usando chaves e uma lâmpada comum.

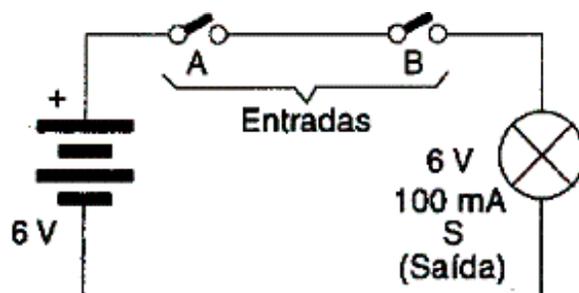


Figura 33 – Simulação de uma porta E com lâmpada

Também podemos simular esta função com LED, conforme mostra a figura 34.

Na prática

O uso de chaves numa função equivalente, pode ser usado na prática num sistema muito simples. Por exemplo, podemos usar duas chaves na conexão da figura 33, acionando uma fechadura eletrônica. A fechadura só pode ser ativada pela segunda chave, se a primeira (de segurança) estiver fechada.

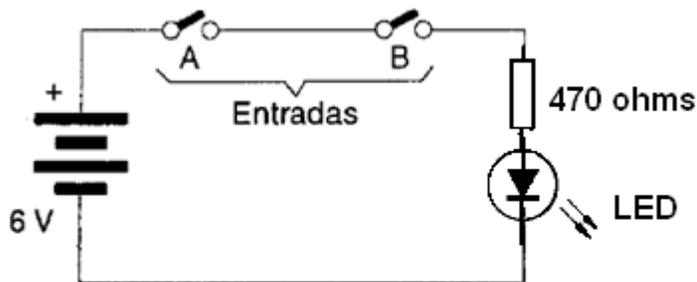


Figura 34 – Circuito de simulação com LED

Veja que é preciso que uma S1 E S2 estejam fechadas para que a saída (lâmpada ou LED) seja ativada.

Para uma porta E de três entradas podemos escrever a seguinte tabela verdade:

Entradas			Saída
A	B	C	S
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

Para que a saída seja 1, é preciso que todas as entradas sejam 1.

Veja ainda o leitor que para uma porta E de 2 entradas, temos 4 combinações possíveis para os sinais aplicados. Para uma porta E de 3 entradas, temos 8 combinações possíveis para o sinal de entrada.

Para uma porta de 4 entradas, temos 16 combinações possíveis de entradas e assim por diante.

2.6 - Função lógica OU

A função OU, ou ainda OR (do inglês), é definida como aquela em que a saída estarno nível alto se uma, ou mais de uma das entradas, estiver no nível alto. Esta função é representada pelos símbolos mostrados na figura 35.

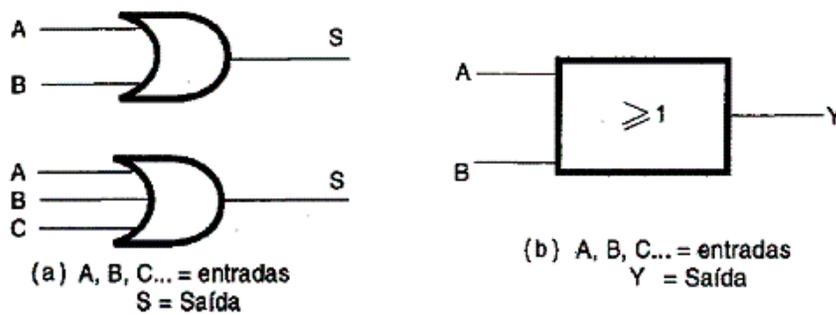


Figura 35 – A função lógica OU (OR)

O símbolo adotado normalmente nas nossas publicações é o mostrado em (a). No entanto, existem diagramas e manuais em que a representação adotada é a mostrada em (b).

Para uma porta OU de duas entradas podemos elaborar a seguinte tabela verdade:

Entradas		Saída
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Veja então que a saída estará no nível 1 se uma OU outra entrada estiver no nível 1.

A equação booleana será:

$$S = A + B$$

Um circuito simples com chaves e lâmpada para simular esta função é dada na figura 36

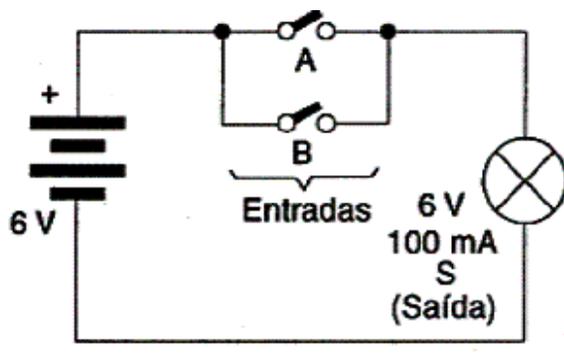


Figura 36 – Circuito simples de porta OU usando lâmpada

Quando uma ou outra chave estiver fechada, (entrada 1) a lâmpada receberá corrente (saída 1), conforme desejamos.

Também, nesse caso, podemos simular a função com outros componentes comuns como transistores e relés.

Na figura 37 temos um circuito de simulação usando LED em lugar da lâmpada indicadora.

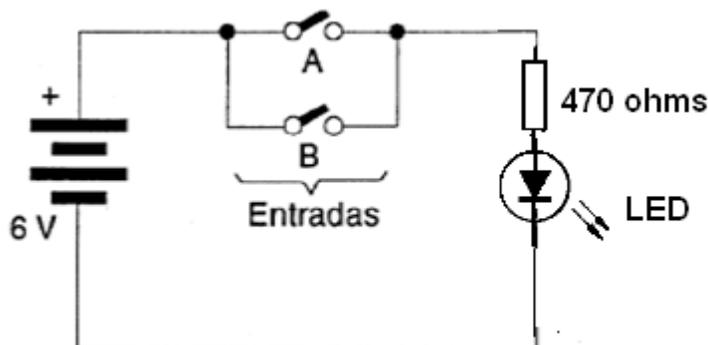


Figura 37 - Função OU com LED

Para mais de duas variáveis podemos ter portas com mais de duas entradas. Para o caso de uma porta OU de três entradas teremos a seguinte tabela verdade:

Entradas			Saída
A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Na prática

Como exemplo desta função numa aplicação prática simples podemos citar um alarme em que dois ou mais sensores normalmente abertos (NA) são ligados a uma sirene. Qualquer um deles que seja acionado, fará com que o alarme toque.

Observe que, nessa função, teremos a saída no nível abaixo com apenas uma combinação possível das entradas: todas as entradas no nível baixo.

2.7 - Função NÃO-E (NAND)

As funções E, OU e NÃO (inversor) são a base de toda a Álgebra Booleana e todas as demais podem ser consideradas como derivadas delas.

Assim, uma primeira função importante, derivada das anteriores, é a obtida pela associação da função E com a função Não, ou seja, a negação da função E que é denominada Não-E ou do inglês NAND.

Na figura 38 temos os símbolos adotados para representar esta função.

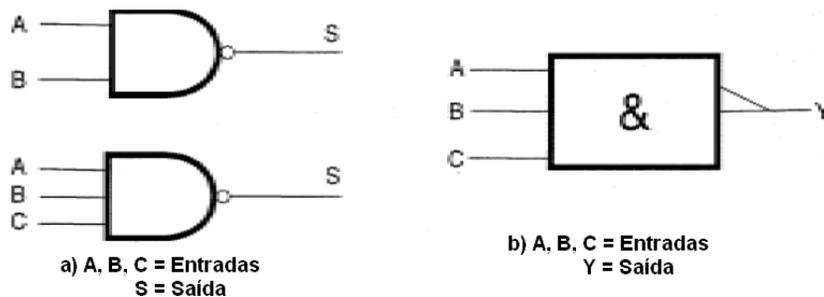


Figura 38 – A função Não-E ou NAND

Observe a existência de um pequeno círculo na saída da porta para indicar a negação no caso da simbologia que adotamos (a). O triângulo na saída do símbolo mostrado em (b) indica a negação da função.

A equação booleana para esta função será escrita como:

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

Onde: S é a saída

A e B são as entradas

O traço sobre a soma indica a negação

Podemos dizer que para a função NAND a saída estará no nível 0 se, e somente se, as entradas estiverem no nível 1.

A tabela verdade para uma porta Não-E ou NAND de duas entradas é a seguinte:

Entradas		Saída
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Na figura 39 temos um circuito simples com chaves que simula esta função.

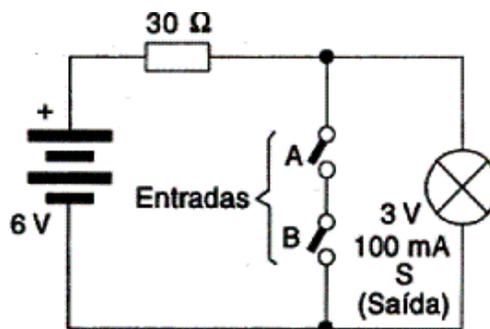


Figura 39 – Circuito com lâmpada que simula uma porta ou função NAND

A mesma função, simulada com um circuito usando LED é mostrada na figura 40.

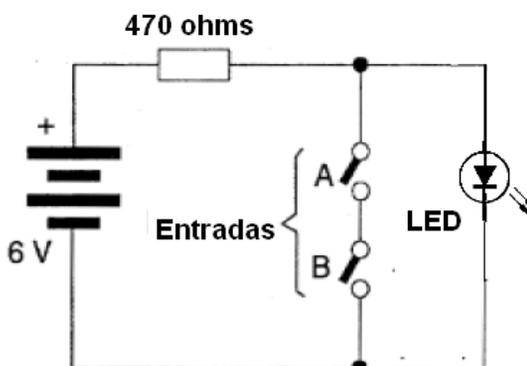


Figura 40 – Simulação da função NAND com LED

Veja que a lâmpada (ou LED) só apagará (saída 0) quando as duas chaves estiverem fechadas (1) curto-circuitando assim sua alimentação. O resistor é usado para limitar a corrente da fonte.

Também neste caso podemos ter a função NAND com mais de duas entradas. Para o caso de 3 entradas teremos a seguinte tabela verdade:

Entradas			Saída
A	B	C	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

A equação booleana para uma função NAND ou Não-E de 3 entradas será escrita como:

$$S = \overline{A + B + C}$$

Na prática
Podemos dar como exemplo prático de aplicação desta função, de forma simplificada, usando chaves, uma fechadura de código, em que, no circuito dado na figura 38, as chaves A e B seriam o código e em lugar do LED um alarme. O alarme só será desativado se as duas chaves correspondentes ao código forem acionadas.

2.8 - Função Não-Ou (NOR)

Esta é a negação da função OU, obtida pela associação da função OU com a função Não ou inversor. O termo inglês usado para indicar esta função é NOR, e seus símbolos são mostrados na figura 41.

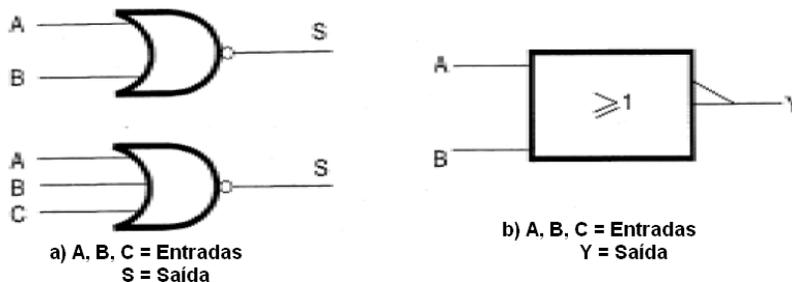


Figura 41 – Função Não-OU ou NOR

Em (a) temos a simbologia mais adotada em nossas publicações, e em (b) um segundo tipo de simbologia que pode ser encontrada em muitos manuais técnicos.

Podemos definir sua ação da seguinte forma: a saída será 1 se, e somente se, todas as variáveis de entrada forem 0.

Uma tabela verdade para uma função NOR de duas entradas é mostrada a seguir:

Entradas		Saída
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

A equação booleana para esta função é:

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

Um circuito simples usando chaves e lâmpada para simular esta função é mostrado na figura 42.

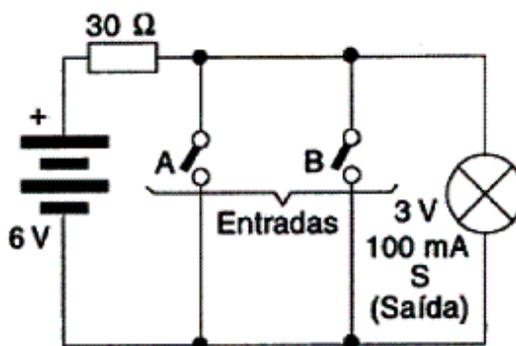


Figura 42 – Circuito NOR com lâmpada

Neste caso também podemos elaborar uma versão com LED. Conforme mostra a figura 43.

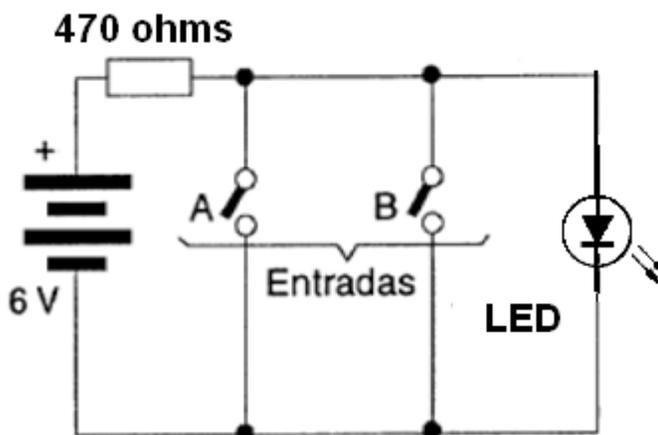


Figura 43- Função NOR com LED indicador

Observe que a lâmpada (ou LED) só se mantém acesa (nível 1) se as duas chaves (S1 e S2) estiverem abertas (nível 0).

Como nos demais exemplos, podemos implementar esta função experimentalmente usando relés, transistores e outros componentes comuns. Da mesma forma que nas funções anteriores podemos ter portas NOR com mais de duas entradas. Para o caso de três entradas teremos a seguinte tabela verdade:

Entradas			Saída
A	B	C	S
0	0	0	1
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	0

Na prática

Esta função, na sua forma simples com chaves, pode ser usada desativar uma carga quando uma ou outra chave for acionada.

Observe que, para esta função todas as saídas estarão no nível alto exceto quando as três entradas estiverem, ao mesmo tempo, no nível baixo.

2.9 - Função Ou-exclusivo (Exclusive-OR)

Uma função de grande importância para o funcionamento dos circuitos lógicos digitais, e especificamente aqueles que realizam operações matemáticas, como os microprocessadores e microcontroladores, é a denominada Ou-exclusivo, ou usando o termo inglês, “Exclusive-Or”.

Esta função tem a propriedade de realizar a soma de valores binários ou ainda encontrar o que se denomina “paridade” de valores binários, o que será visto futuramente.

Na figura 44 temos os símbolos adotados para esta função.

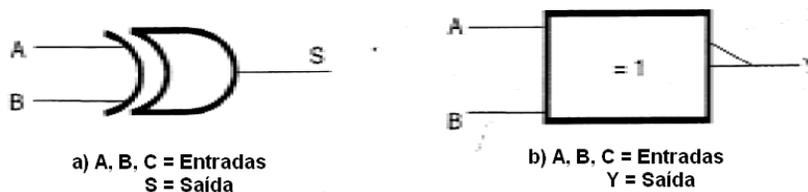


Figura 44 – A função OU-exclusivo ou Exclusive-OR

Da mesma forma que nos casos anteriores, temos simbologias diferentes que podem ser encontradas nos manuais e diagramas. O leitor deve conhecer as duas simbologias, para poder trabalhar eficientemente com circuitos digitais.

Podemos definir sua ação da seguinte forma: a saída será 1 se, e somente se, as variáveis de entrada forem diferentes. Isso significa que, para uma porta Exclusive-OR de duas entradas teremos saída 1 se as entradas forem 0 e 1 ou 1 e 0, mas a saída será 0 se as entradas forem ambas 1 ou ambas 0, conforme mostra a seguinte tabela verdade:

Entradas		Saída
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

A equação booleana para esta função é:

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

Na prática

Um sistema “inteligente” de acionamento de lâmpadas pode ser elaborado com esta função. Onde, a lâmpada acende se uma ou outra pessoa acionar o seu interruptor, mas não acende, se as duas pessoas acionarem os interruptores, ao mesmo tempo.

Esta função é derivada das demais no sentido de que podemos “montá-la” usando portas conhecidas, conforme mostra a figura 45.

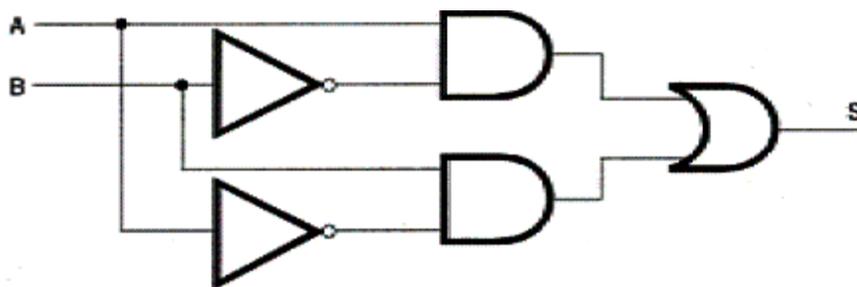


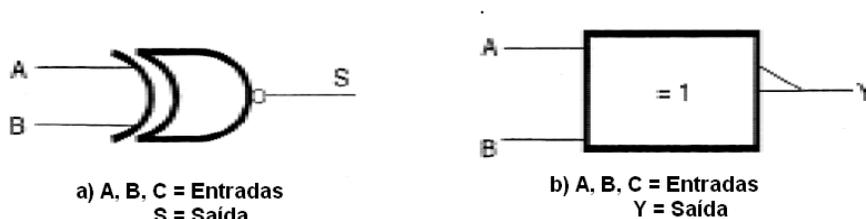
Figura 45 – Usando inversores, portas AND e uma porta OR para obter uma função OU-Exclusivo

Se bem que esta função tenha seu próprio símbolo, e possa ser considerada um “bloco” independente nos projetos, podemos sempre implementá-la com um circuito equivalente como o mostrado acima.

Na verdade, conforme veremos nas lições futuras existem diversos circuitos integrados usados em aplicações digitais que já levam em seu interior esta função.

2.10 - Função Não-Ou exclusivo ou coincidência (Exclusive-NOR)

Podemos considerar esta função como o “inverso” do Ou-exclusivo. Sua denominação em inglês é, Exclusive-NOR sendo representada pelo símbolo mostrado na figura 46.



a) A, B, C = Entradas
S = Saída

b) A, B, C = Entradas
Y = Saída

Figura 46 – A função Não-Ou Exclusivo ou Exclusive-NOR

Observe o círculo que indica a negativa da função anterior, se bem que essa terminologia não seja apropriada neste caso.

No caso da simbologia mostrada em (b) o triângulo na saída indica a negação.

Esta função pode ser definida como a que apresenta uma saída igual a 1 se, e somente se as variáveis de entrada forem iguais.

Uma tabela verdade para esta função é a seguinte:

Entrada		Saída
A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

Podemos implementar esta função usando outras já conhecidas conforme mostra a figura 47.

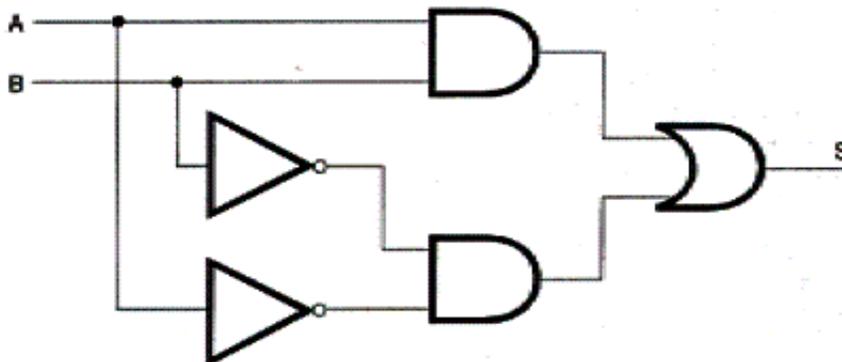


Figura 47 – Implementação da função Não-Ou Exclusivo a partir de outras funções

Na prática

Um sistema de controle de um motor de uma máquina que só funcione se dois sensores estiverem ativados ou desativados, mas que não funciona apenas um dos sensores estiver ativado, é uma idéia de aplicação imediata desta função.

Também observamos que podemos encontrar essa função já fazendo parte de muitos circuitos integrados destinados à aplicações digitais.

2.11 - Propriedades das operações lógicas

As portas realizam operações com os valores binários aplicados às suas entradas.

Assim, conforme vimos, podemos representar estas operações por uma simbologia apropriada facilitando assim o projeto dos circuitos e permitindo visualizar melhor o que ocorre quando associamos muitas funções. No entanto para saber associar as diversas portas e com isso realizar operações mais complexas é preciso conhecer as propriedades que as operações apresentam.

Exatamente como no caso das operações com números decimais, as operações lógicas com a Álgebra Booleana se baseiam numa série de postulados e teoremas algo simples.

Os principais são dados a seguir e prová-los fica por conta dos leitores que desejam ir além. Para entender, entretanto, seu significado não é preciso saber como provar sua validade, mas sim memorizar seu significado. Partindo das representações lógicas que já estudamos, podemos enumerar as seguintes propriedades das operações lógicas:

Estas propriedades são dadas na forma de Postulados. Todo profissional que trabalha com circuitos lógicos digitais deve conhecer, portanto, os Postulados da Álgebra Booleana.

a) Leis da Tautologia

“A repetição por soma ou multiplicação não alteram o valor real de um elemento.”

O circuito mostrado na figura 48 corresponde a esta lei.

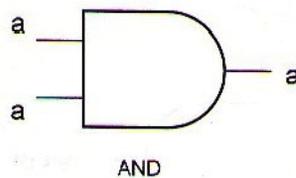
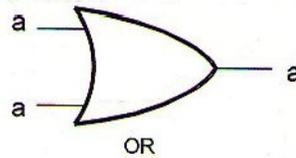


Figura 48 - Tautologia

Notações

Observamos novamente que as notações comuns que usamos em álgebra booleana são diferentes das notações usadas em matemática, especificamente na teoria dos conjuntos, e também da notação lógica.

$$a + a = a$$

$$a \times a = a$$

Notação matemática (teoria dos conjuntos)

$$a \cap a = a$$

$$a \cup a = a$$

Notação lógica:

$$a \wedge a = a$$

$$a \vee a = a$$

b) Leis da Comutação

“A conjunção e a disjunção não são afetadas pela mudança sequencial”.

A figura 49 mostra os diagramas correspondentes a esta função.

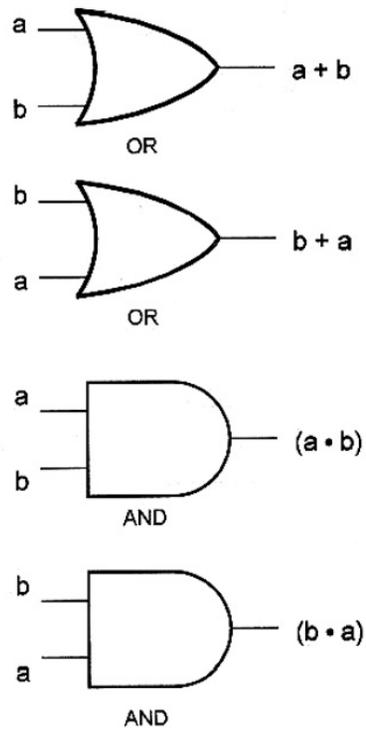


Figura 49 – Diagramas correspondentes as leis da comutação

$$a + b = b + a$$

$$a \times b = b \times a$$

Notação matemática:

$$a \cup b = b \cup a$$

$$a \cap b = b \cap a$$

Notação lógica:

$$a \vee b = b \vee a$$

$$a \wedge b = b \wedge a$$

Regrinhas conhecidas

Veja que esta lei corresponde à conhecida regrinha que aprendemos em matemática em que “a ordem das parcelas não afeta a soma” e “a ordem dos fatores não afeta o produto”.

c) Leis da Associação

“O agrupamento não afeta a disjunção ou a conjunção”.

A figura 50 mostra os circuitos equivalentes.

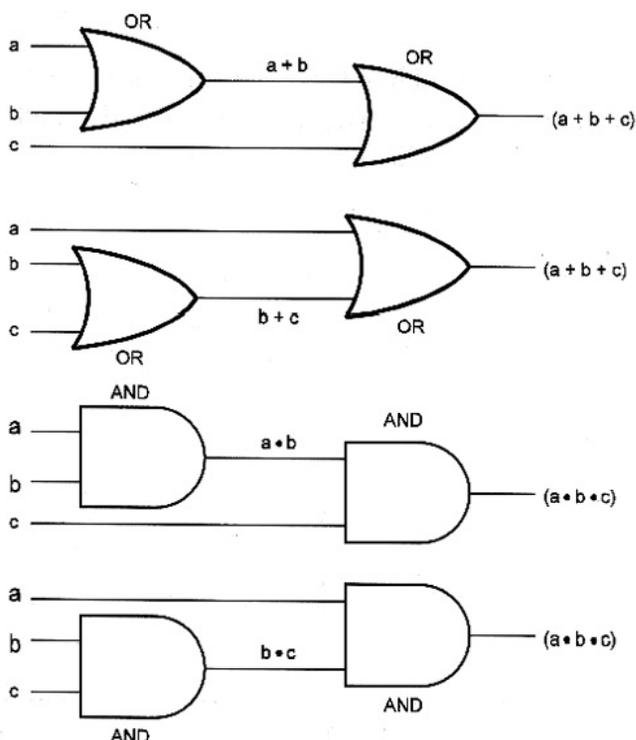


Figura 50 – Leis da associação

$$a + (b + c) = (a + b) + c$$

$$a \times (b \times c) = (a \times b) \times c$$

Notação matemática:

$$a \cup (b \cup c) = (a \cup b) \cup c$$

$$a \cap (b \cap c) = (a \cap b) \cap c$$

Notação lógica:

$$a \vee (b \vee c) = (a \vee b) \vee c$$

$$a \wedge (b \wedge c) = (a \wedge b) \wedge c$$

d) Leis da Distribuição

“Um elemento é adicionado a um produto pela adição do elemento a cada membro do produto, e uma soma é multiplicada por um elemento pela multiplicação de cada elemento da soma por este elemento”.

O diagrama do circuito equivalente é mostrado na figura 51.

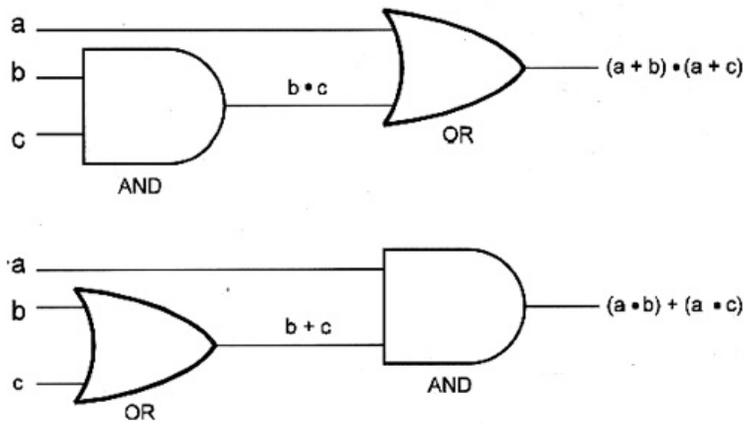


Figura 51 – Leis da distribuição

$$a + (b \times c) = (a + b) \times (a + c)$$

$$a \times (b + c) = (a \times b) + (a \times c)$$

Notação Matemática:

$$a \cup (b \cap c) = (a \cup b) \cap (a \cup c)$$

$$a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$$

Notação Lógica:

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

e) Leis da Absorção

“A disjunção de um produto por um de seus membros é equivalente a este membro. A conjunção de uma soma por um de seus membros é equivalente a este membro”.

O circuito lógico equivalente é mostrado na figura 52.

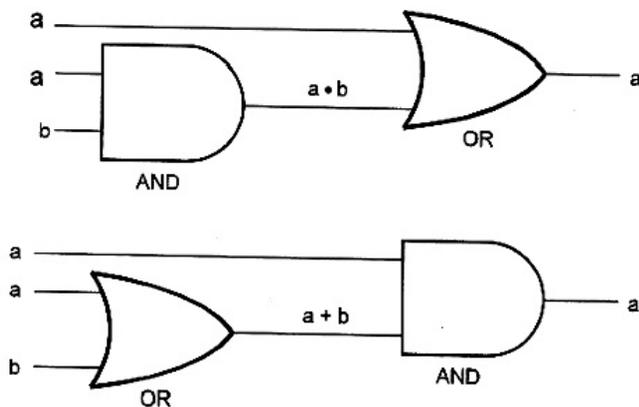


Figura 52 – Leis da absorção

$$a + (a \times b) = a$$

$$a \times (a + b) = a$$

Notação matemática:

$$a \cup (a \cap b) = a$$

$$a \cap (a \cup b) = a$$

Notação lógica:

$$a \vee (a \wedge b) = a$$

$$a \wedge (a \vee b) = a$$

f) Leis da Classe Universo

“A soma consistindo de um elemento e a classe universo é equivalente a classe universo. O produto consistindo de um elemento e a classe universo é equivalente ao elemento”.

Os circuitos equivalentes são dados na figura 53.

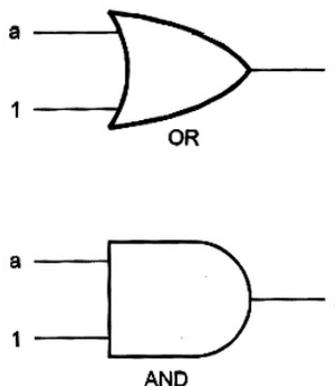


Figura 53 – Leis da classe universo

$$a + 1 = 1$$

$$a \times 1 = a$$

Notação matemática:

$$a \cup 1 = 1$$

$$a \cap 1 = a$$

Notação lógica:

$$a \vee 1 = 1$$

$$a \wedge 1 = a$$

d) Leis da Classe Nulo

“A soma consistindo de um elemento e a classe nulo é equivalente ao elemento. O produto consistindo de um elemento e a classe nulo é equivalente à classe nulo”.

As funções equivalentes são mostradas na figura 54.

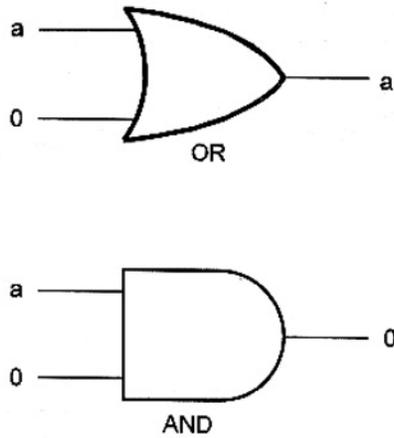


Figura 54 – Leis da classe nulo

$$a + 0 = a$$

$$a \times 0 = 0$$

Notação matemática:

$$a \cup 0 = a$$

$$a \cap 0 = 0$$

Notação lógica:

$$a \vee 0 = a$$

$$a \wedge 0 = 0$$

g) Leis da Complementação

“A soma consistindo de um elemento e o seu complemento é equivalente à classe universo. O produto consistindo de um elemento e seu complemento é equivalente à classe nulo”.

Na figura 55 temos a representação dessas leis.

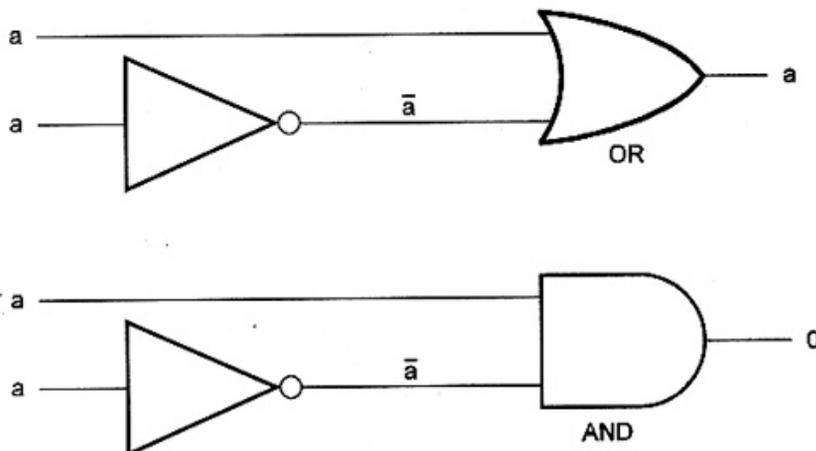


Figura 55 – Leis da complementação

$$a + \bar{a} = 1 \quad a \times \bar{a} = 0$$

h) Leis da Contraposição

“Se um elemento **a** é equivalente ao complemento de um elemento **b** isso implica que o elemento **b** é equivalente ao complemento do elemento **a**”.

A figura 56 representa o diagrama lógico equivalente.

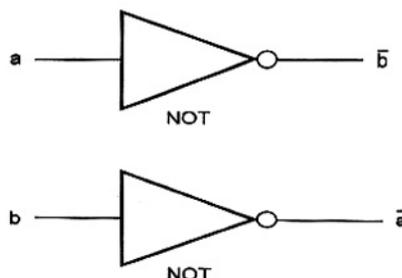


Figura 56 – Leis da contraposição

$$a = \bar{b} \rightarrow b = \bar{a}$$

Notação matemática:

$$a = b' \Rightarrow b = a'$$

i) Lei da Dupla Negação

“O complemento da negação de um elemento é equivalente ao elemento”.

A figura 57 mostra o circuito lógico equivalente.

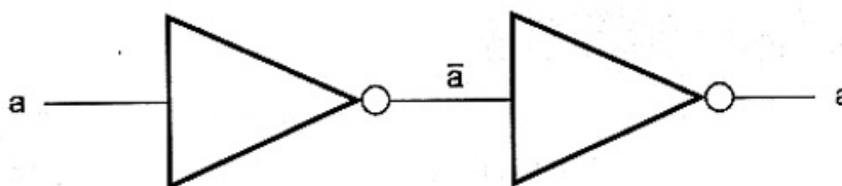


Figura 57 - Lei da dupla negação

$$a = \bar{\bar{a}}$$

Notação matemática:

$$a = a''$$

Notação lógica:

$$a = -a'$$

j) Leis da Expansão

“A disjunção de um produto composto dos elementos **a** e **b**, e um produto composto do elemento **a** e o complemento do elemento **b**, é equivalente ao elemento **a**. A conjunção de uma soma

composta dos elementos **a** e **b**, e a soma composta do elemento **a** e o complemento do elemento **b**, é equivalente ao elemento **a**”.

A figura 58 mostra os circuitos lógicos equivalentes.

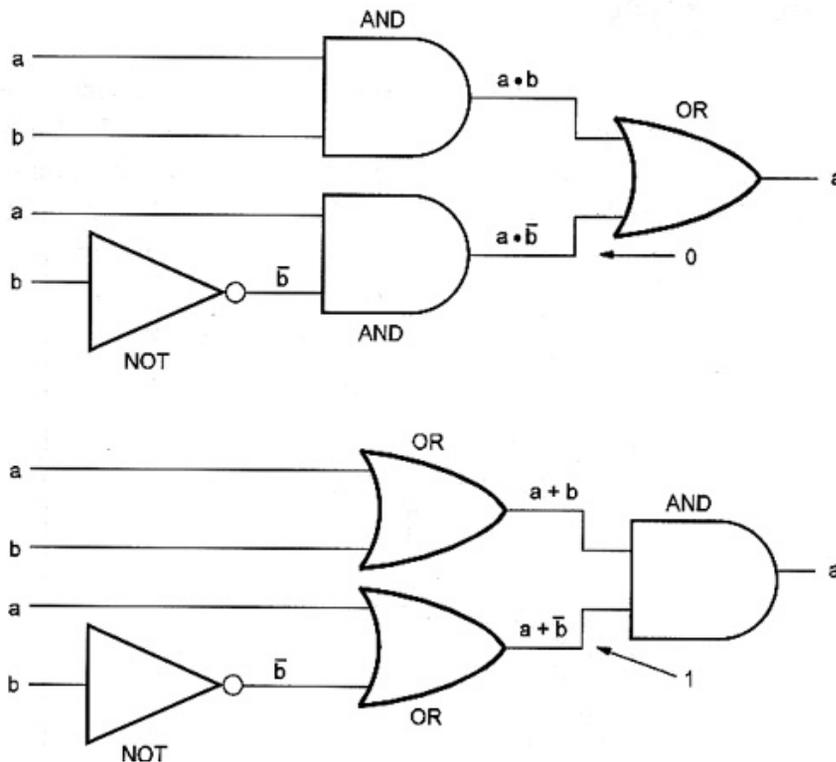


Figura 58 – Leis da expansão

$$(a \cdot b) + (a \cdot \bar{b}) = a$$

$$(a + b) \cdot (a + \bar{b}) = a$$

Notação matemática:

$$(a \cap b) \cup (a \cap b') = a$$

$$(a \cup b) \cap (a \cup b') = a$$

Notação lógica:

$$(a \wedge b) \vee (a \wedge \sim b) = a$$

$$(a \vee b) \wedge (a \vee \sim b) = a$$

k) Leis da Dualidade

“O complemento de uma soma composta dos elementos **a** e **b** é equivalente a conjunção do complemento do elemento **a** e o complemento do elemento **b**. O complemento de um produto composto pelos elementos **a** e **b** é equivalente a disjunção do complemento do elemento **a** e o complemento do elemento **b**”.

Na figura 59 temos os circuitos lógicos equivalentes.

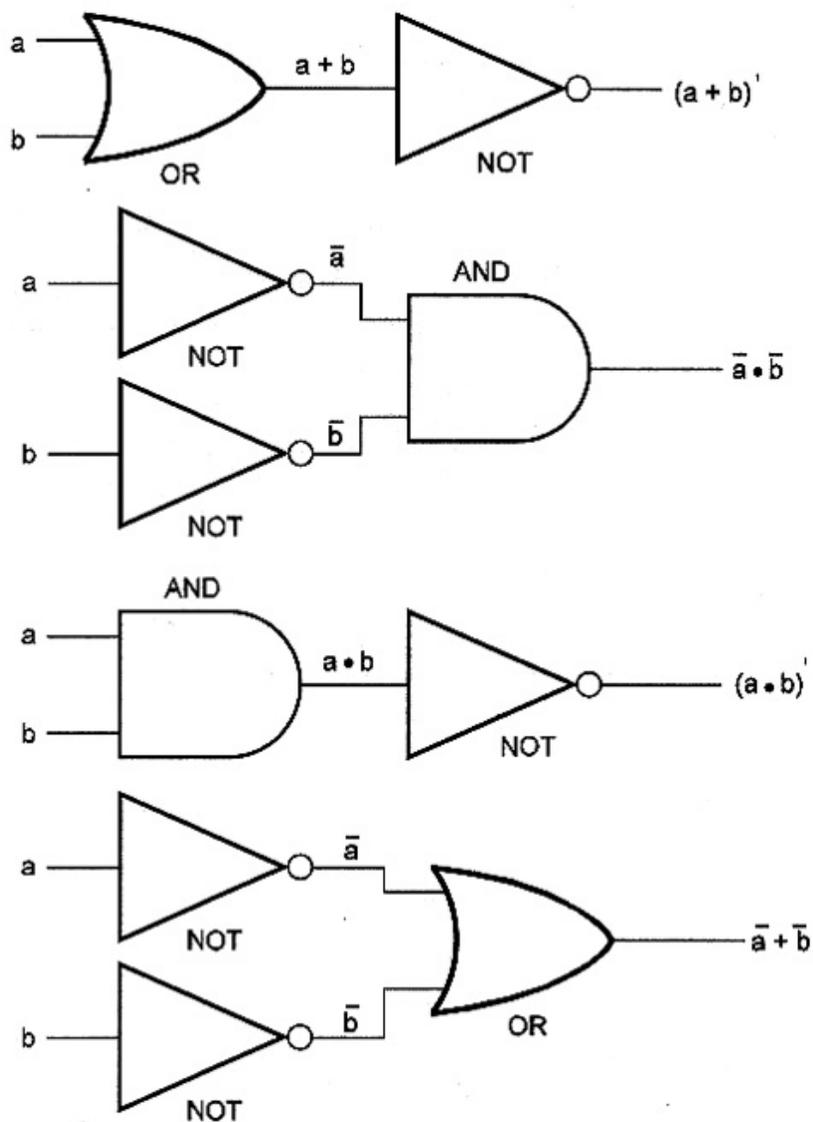


Figura 59 – Lei da dualidade

$$(a + b)' = \bar{a}\bar{b}$$

$$(a \cdot b)' = \bar{a} + \bar{b}$$

Notação matemática

$$(a \cup b)' = a' \cap b'$$

$$(a \cap b)' = a' \cup b'$$

Notação lógica:

$$\sim (a \vee b) = \sim a \wedge \sim b$$

$$\sim (a \wedge b) = \sim a \vee \sim b$$

Relações Booleanas

Damos a seguir algumas relações importantes que o leitor deve memorizar.

a) Ponto Identidade

- Adição:
- $a + 0 = a$
- $a + 1 = 1$
- $a + a = a$
- Multiplicação
- $0 \times a = 0$
- $1 \times a = a$
- $a \times a = a$

b) Comutativa

- Adição:
- $(a + b) = (b + a)$
- multiplicação:
- $a \times b = b \times a$

c) Associativa

- Adição:
- $(a + b) + c = a + (b + c)$
- Multiplicação
- $(a \times b) \times c = a \times (b \times c)$

d) Distributiva

- $a + (b \times c) = a \times (b + c)$
- $a + b \times c = (a + b) \times (a + c)$

e) Absorção

- $a \times (a + b) = a + a \times b = a$

Teoremas de De Morgan:

“Aplicando a operação NÃO a uma operação E, o resultado obtido é igual ao da operação OU aplicada aos complementos das variáveis de entrada”.

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

“Aplicando a operação Não a uma operação OU o resultado é igual ao da operação E aplicada aos complementos das variáveis de entrada”.

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

Matemática

Veja que todas essas relações são bem conhecidas da matemática nas operações comuns. Por exemplo, na comutativa temos que “a ordem das parcelas não altera o resultado da soma”, e assim por diante.

De Morgan

Nascido na Índia, filho de ingleses, estudou na Inglaterra e desenvolveu diversos trabalhos no campo da lógica.



Augustus De Morgan
(1806 – 1871)

2.12 - Fazendo tudo com portas NAND ou Não-OU

As portas Não-E, pelas suas características, podem ser usadas pra se obter qualquer outra função que estudamos. Esta propriedade torna essas portas blocos universais nos projetos de circuitos digitais já que, na forma de circuitos integrados, as funções NAND são fáceis de obter e, além disso, baratas.

A seguir, vamos mostrar de que modo podemos obter as funções estudadas simplesmente usando portas NAND.

* Inversor

Para obter um inversor a partir de uma porta NAND basta unir suas entradas ou colocar uma das entradas no nível lógico 1, conforme mostra a figura 60.

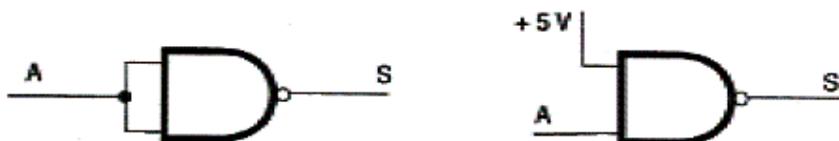


Figura 60 – Inversor com a porta NAND ou Não OU

* Uma porta E (AND) é obtida simplesmente agregando-se à função Não-E (NAND) um inversor em cada entrada, conforme mostra a figura 61.

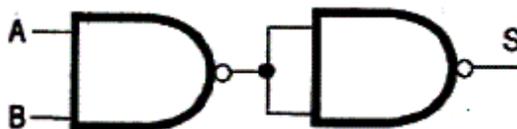


Figura 61 – Função E (AND) obtida com duas portas Não-E ou NAND

* A função OU pode ser obtida com o circuito mostrado na figura 62.

O que se faz é inverter a saída, depois de aplicá-las a uma porta NAND.

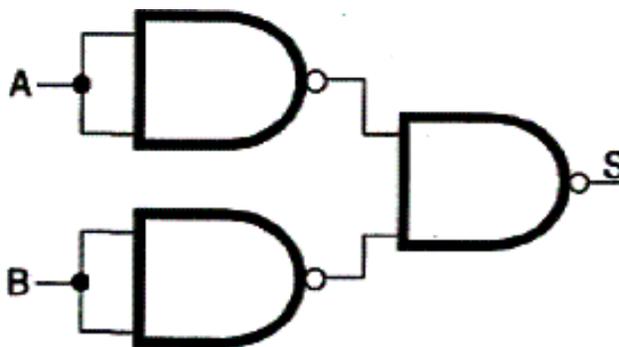


Figura 62 – Função OU implementada com portas Não-E (NAND)

Conclusão

Os princípios em que se baseiam os circuitos lógicos digitais podem parecer algo abstratos, pois usam muito de matemática, e isso pode desestimular os leitores. No entanto, eles são apenas o começo. Fazendo um esforço para entendê-los o leitor certamente será recompensado, pois estes princípios estão presentes em tudo que seu computador faz e em muitos equipamentos denominados digitais. Nos próximos capítulos, quando os princípios estudados começam a tomar uma forma mais concreta aparecendo em circuitos que realmente funcionam e fazem alguma coisa, ficará fácil entendê-los melhor.

Termos em inglês

Os termos utilizados em eletrônica digital muitas vezes são mantidos na forma original, em inglês, em muitos documentos. É o caso das portas ou funções lógicas que é normal que sejam expressas como AND, NAND, OR, NOR, além de outras. Alguns termos importantes:

Gate – portas
2-Input – duas entradas
Boolean equation – equação booleana
Truth table – tabela verdade
Exclusive OR- Ou exclusiva
Postulates – postulados
Laws – leis
Comutation – comutação
Association – associação
Distribution – distribuição
Absortion – absorção
Classe – class
Null – nulo
Contraposition – contraposição
Double negation – dupla negação
Expansion- expansão
Duality - dualidade

Termos para pesquisa

- Teorema de De Morgan
- Funções lógicas
- Portas lógicas
- Operações lógicas
- Postulados

EXERCÍCIOS

1. Se associarmos à presença de uma tensão o nível lógico 1 e a sua ausência o nível 0 teremos que tipo de lógica:

- a) Digital b) Positiva c) Negativa d) Booleana

2. Na entrada de uma função lógica NÃO aplicamos o nível lógico 0.

A saída certamente ser :

- a) 0 b) 1 c) Pode ser 0 ou 1 d) Estará indefinida

3. O circuito que realiza a operação lógica NÃO é denominado:

- a) Porta lógica
b) Inversor
c) Amplificador digital
d) Amplificador analógico

4. Se na entrada de uma porta NAND aplicarmos os níveis lógicos 0 e 1 a saída certamente será:

- a) 0
b) 1
c) Pode ser 0 ou 1
d) Estará indefinida

5. Em qual das seguintes condições de entrada a saída de uma porta OR será 0:

- a) 0,0
b) 0,1
c) 1,0
d) 1,1

6. Qual é o nome da função lógica em que obtemos uma saída 1 quando as entradas tiverem níveis lógicos diferentes, ou seja, forem 0 e 1 ou 1 e 0.

- a) NAND
b) NOR
c) AND
d) Exclusive OR

7. Qual é a porta que pode ser utilizada para se implementar qualquer função lógica:

- a) Inversor (NÃO)
b) AND
c) NAND
d) OR



» Famílias de Circuitos Lógicos Digitais

No capítulo anterior conhecemos os fundamentos da Álgebra de Boole que regem o funcionamento dos circuitos lógicos digitais encontrados na prática em muitos equipamentos, como computadores, instrumentos de medida, controles e automatismos industriais, robôs, etc. Vimos de que modo umas poucas funções básicas funcionam e sua importância na obtenção de funções mais complexas. Mesmo sendo um assunto um pouco abstrato, por envolver princípios matemáticos, o leitor pode perceber que é possível simular o funcionamento de algumas delas com circuitos eletrônicos relativamente simples, usando chaves e lâmpadas. Os circuitos eletrônicos modernos, entretanto, não usam chaves e lâmpadas, mas sim dispositivos muito rápidos que podem estabelecer os níveis lógicos nas entradas das funções com velocidades incríveis, e isso lhes permite realizar milhões, ou mesmo bilhões de operações em cada segundo. Que tipo de circuitos são usados e como estes circuitos são encontrados na prática, em blocos básicos, que juntados podem levar a elaboração de circuitos muito complicados como os encontrados nos computadores, robôs, controles industriais, e em muitas outras aplicações, é o assunto deste nosso capítulo. Nele o leitor já vai começar a tomar contacto com componentes práticos das famílias usadas na montagem dos equipamentos digitais de todos os tipos. São estes os componentes básicos que podem ser encontrados em equipamentos que usam tecnologia digital de todos os tipos. Este capítulo consta dos seguintes itens:

- 3.1 - O transistor como chave eletrônica
- 3.2 - Melhorando o desempenho
- 3.3 - A família TTL
- 3.4 - Outras características da família TTL
- 3.5 - Sub-famílias TTL
- 3.6 - Compatibilidade entre as famílias
- 3.7 - Open Collector e Totem Pole
- 3.8 - Tri-state

Objetivos

- * Entender como os transistores podem ser usados nos circuitos digitais
- * Familiarizar-se com uma das mais importantes famílias de circuitos digitais, a TTL
- * Conhecer algumas características elétricas dos circuitos integrados TTL
- * Ver como ocorre a compatibilidade entre esta família e outras

3.1 - O transistor como chave eletrônica

Um transistor pode funcionar como um interruptor, deixando passar ou não uma corrente, conforme a aplicação de uma tensão de controle em sua entrada.

Assim, na simulação dos circuitos que estudamos, e em que usamos chaves, é possível utilizar transistores com uma série de vantagens.

No caso das chaves, o operador era responsável pela entrada do sinal, pois ele é que, atuando com suas mãos sobre a chave, deveria estabelecer o nível lógico de entrada, mantendo esta chave aberta ou fechando conforme desejasse 0 ou 1.

Se usarmos um transistor, teremos uma vantagem importante. O transistor pode operar com a tensão ou nível lógico que seja produzido por outra função, e não necessariamente por uma pessoa que precise acionar uma chave.

Assim, as funções lógicas implementadas com transistores têm a vantagem de poderem ser interligadas umas nas outras, pois o sinal que aparece na saída de cada uma pode ser usado como entrada para outra. Mais do que isso, podemos ter uma sequência de etapas que funcione automaticamente, processando os níveis lógicos, à medida que passam de uma etapa para outra.

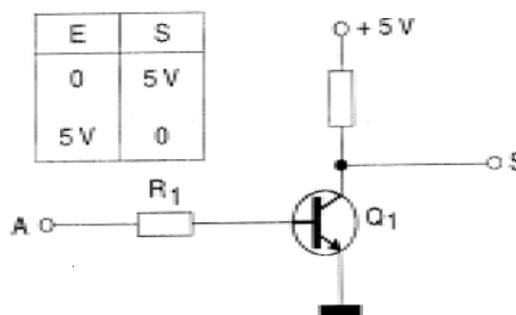


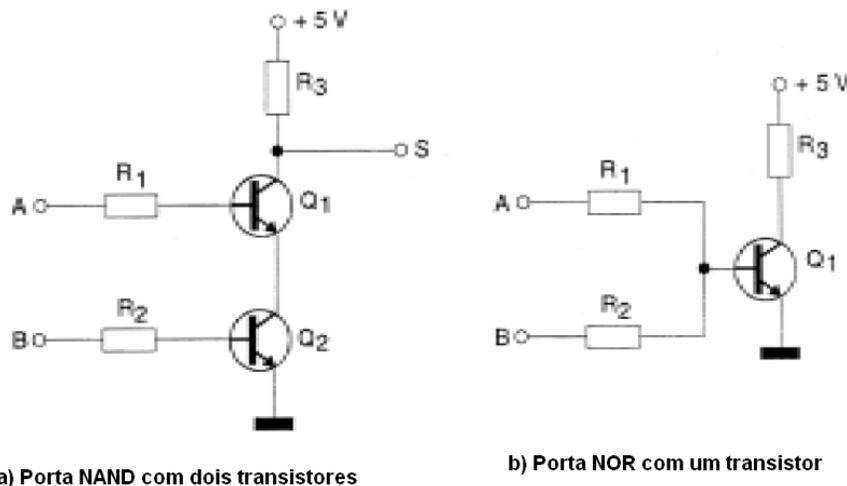
Figura 63 - O transistor como chave numa função inversor

Na figura 63 damos um exemplo interessante de como podemos obter um inversor usando um transistor.

Aplicando o nível 1 na base do transistor, ele conduz até o ponto de saturar, o que faz com que a tensão no seu coletor caia a 0. Por ou-

tro lado, na ausência de tensão na sua base, que corresponde ao nível 0 de entrada, o transistor se mantém cortado, e a tensão no seu coletor se mantém alta o que corresponde ao nível lógico 1.

Outras funções podem ser conseguidas com transistores conforme mostra a figura 64.



a) Porta NAND com dois transistores

b) Porta NOR com um transistor

Figura 64 -Outras funções implementadas com transistores

Em (a) temos uma porta NAND de duas entradas. Será preciso que as duas entradas sejam levadas ao nível alto para que a saída seja baixa. Em qualquer outra combinação de níveis lógicos de entrada, um dos transistores, pelo menos, não conduz e a saída se mantém no nível alto.

Em (b) temos uma porta NOR com um transistor bipolar apenas. Quando uma ou outra entrada vai ao nível alto, o transistor conduz, e com isso a saída vai ao nível baixo. A saída só estará no nível alto quando as duas entradas estiverem no nível baixo também.

Por estes exemplos, percebemos que a elaboração de um circuito lógico digital capaz de realizar operações complexas usando transistores é algo que pode ser conseguido com relativa facilidade. É claro que outros dispositivos equivalentes como transistores de efeito de campo também podem ser usados com a mesma finalidade.

3.2 - Melhorando o desempenho

No entanto, usar transistores em circuitos que correspondam a cada função, de uma maneira não padronizada, pode trazer algumas dificuldades.

Nos primeiros tempos da eletrônica digital, cada função era montada com seus transistores, diodos e resistores na sua plaquinha para depois, todas serem interligadas. No entanto, este procedimento se revelou inconveniente por diversos motivos.

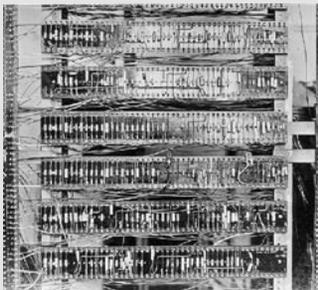
O primeiro deles era a complexidade que o circuito adquiria se tivesse de realizar muitas funções. Reunir uma grande quantidade de

Transistores de efeito de campo

Conforme veremos nas próximas lições, todas as funções implementadas com transistores bipolares também existem com transistores de efeito de campo, na forma de circuitos integrados. Uma característica do FET (transistor de efeito de campo), muito importante quando temos um circuito de alta complexidade, é o seu baixo consumo.

Primeiros computadores

Os primeiros computadores digitais usavam válvulas como elementos ativos para implementar suas funções lógicas. Com o aparecimento do transistor, muito menor e com menos consumo, a montagem dos computadores foi facilitada. Mas, mesmo assim, os primeiros computadores transistorizados tinham uma placa para cada função lógica. Na foto, as placas do primeiro computador transistorizado, o TX-O (Transistorized Experimental Computer) de 1956



circuitos numa única placa exigia um projeto muito bem elaborado da placa a ser utilizada.

O segundo era a necessidade de se padronizar o modo como cada circuito ou função deveria funcionar. Seria muito importante estabelecer que todos os circuitos operassem com a mesma tensão de alimentação e também fornecer sinais que os demais pudessem reconhecer como reconhecer os sinais gerados pelos outros.

O desenvolvimento da tecnologia dos circuitos integrados, possibilitando a colocação num único invólucro de diversos componentes já interligados, foi um passo fundamental para possibilitar um desenvolvimento muito rápido da eletrônica digital.

De fato, o que se fez foi criar uma série de circuitos integrados que contivessem numa única pastilha de silício ou “chip” as funções lógicas digitais mais usadas, e de tal maneira projetadas, para que todas fossem compatíveis entre si, ou seja, operassem com as mesmas tensões e reconhecessem os mesmos sinais.

Usando estes chips seria possível projetar equipamentos capazes de realizar operações lógicas complexas, simplesmente escolhendo as funções básicas para sua implementação e depois interligando-as da forma exigida para se obter os efeitos desejados.

Surgiram então “séries” de circuitos integrados digitais que se popularizaram justamente pela facilidade com que se tornou possível desenvolver qualquer projeto de eletrônica digital.

Estas séries de circuitos integrados eram conhecidas por “Famílias Lógicas”.

Elas funcionavam como “blocos básicos”, a partir das quais os projetistas teriam facilidade em encontrar todos os elementos para montar seus circuitos digitais, desde os mais simples até alguns de grande complexidade.

Na verdade, os blocos foram se tornando cada vez mais complexos, possibilitando a elaboração de circuitos de grande complexidade que só foram substituídos pelo aparecimento de blocos de uma geração superior, já contendo funções completas como os microprocessadores e microcontroladores e DSPs.

Assim, conforme mostra a figura 65, precisando montar um circuito que usasse uma porta AND duas NOR e inversores, o projetista teria disponíveis componentes prontos, ou seja, circuitos integrados (chips) compatíveis entre si, contendo estas funções, e de tal forma que poderiam ser interligadas sem a necessidade de componentes intermediários.

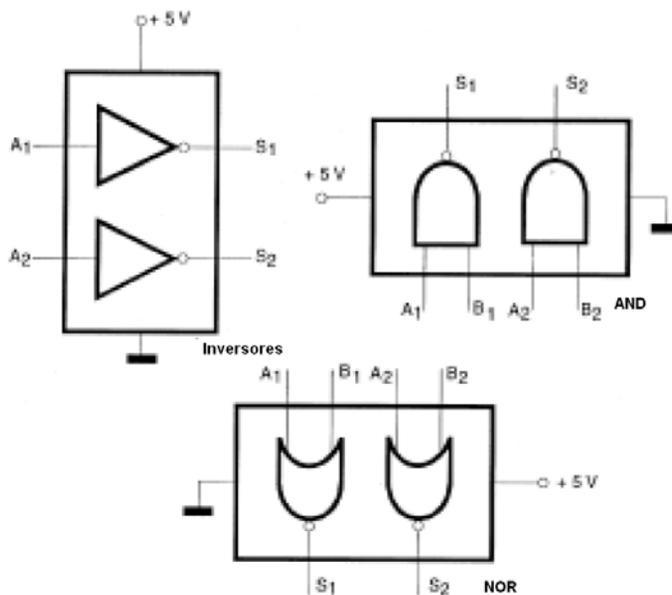


Figura 65 - Circuitos integrados com funções independentes prontas para uso

O sucesso do advento dessas famílias foi enorme, pois além do menor tamanho dos circuitos e menor consumo de energia, havia ainda a vantagem do menor custo e obtenção de maiores velocidades de operação, assim como confiabilidade.

Diversas famílias foram criadas desde o advento dos circuitos integrados, recebendo cada uma delas uma denominação, conforme a tecnologia empregada.

As principais famílias lógicas desenvolvidas foram:

- * RTL ou Resistor Transistor Logic
- * RCTL ou Resistor Capacitor Transistor Logic
- * DTL ou Diode Transistor Logic
- * TTL ou Transistor Transistor Logic
- * C-MOS ou Complementary Metal Oxide Semiconductor
- * ECL ou Emitter Coupled Logic

Atualmente as mais usadas são as Famílias TTL e CMOS que aparecem numa grande quantidade de equipamentos digitais que vão dos circuitos mais simples até computadores aos robôs, equipamentos de telecomunicações e de controles de máquinas industriais.

3.3 - A família TTL

A família TTL foi originalmente desenvolvida pela Texas Instruments, mas hoje, muitos fabricantes de semicondutores produzem seus principais circuitos.

Esta família é principalmente reconhecida pelo fato de ter duas séries que começam pelos números 54, para os componentes de uso militar, e 74 para os componentes de uso comercial. Assim, podemos

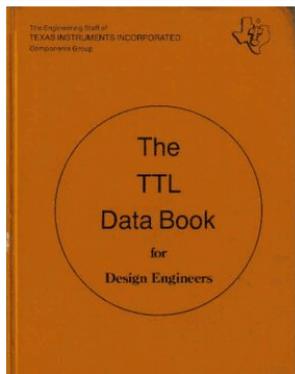
Circuitos integrados modernos

Na figura abaixo temos circuitos integrados que contém chips com milhões de funções lógicas integradas, já interligadas realizando funções complexas. (veja o curso básico e o curso de eletrônica analógica - Volume 1 e 2 desta série)



Circuitos integrados TTL

A família TTL foi criada pela Texas Instruments em 1961 como o nome TCTL. A primeira série, com características militares foi a 5400, seguida pela série comercial 7400, apresentada em 1962, e que até hoje usada, com suas variações, que estudaremos no próximo item e nos próximos capítulos. Na foto, um manual TTL da Texas, como o autor deste livro possui, de enorme utilidade em projetos.



rapidamente associar qualquer componente que comece pelo número “74” ou “54” à família TTL.

Na figura 66 mostramos uma porta típica TTL. Trata-se de uma porta NAND de duas entradas que logo chama a atenção pelo fato de usar um transistor de dois emissores.

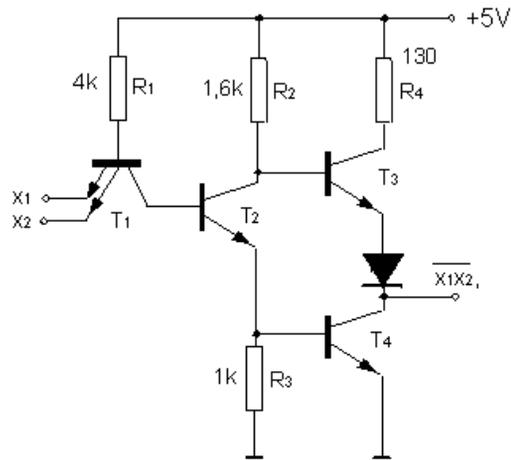


Figura 66 - Porta NAND (Não-E) TTL

A característica mais importante desta família está no fato de que ela é alimentada com uma tensão de 5 Volts.

Assim, para os componentes desta família o nível lógico 0 é sempre a ausência de tensão ou 0 V, enquanto que o nível lógico 1 é sempre uma tensão de +5 V.

Na prática, entretanto, deve-se considerar a resistência interna dos próprios circuitos, o que nos leva a considerar, de uma forma mais apropriada, nível 0, uma tensão “próxima” de 0 V e nível alto, uma tensão “próxima” de 5 V.

Isso significa que os níveis lógicos, para serem reconhecidos, devem estar dentro de faixas bem definidas.

Conforme mostra a figura 67, uma porta TTL, ou qualquer outra função dessa família, reconhecerá como nível 0 as tensões que estiverem entre 0 e 0,8 V, e como 1 as que estiverem numa outra faixa, entre 2,4 e 5 V.

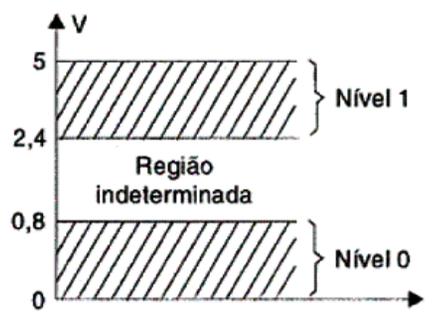


Figura 67 - Faixas de operação da lógica TTL d 5 V

Entre estas duas faixas existe uma região indefinida ou que deve ser evitada. Não se pode prever de que modo vai ser reconhecido um sinal que esteja nessa faixa.

Existem centenas de circuitos integrados TTL disponíveis no mercado para a realização dos projetos. A maioria deles está em invólucros DIL de 14, 16 e 20 pinos, conforme mostra a figura 68.

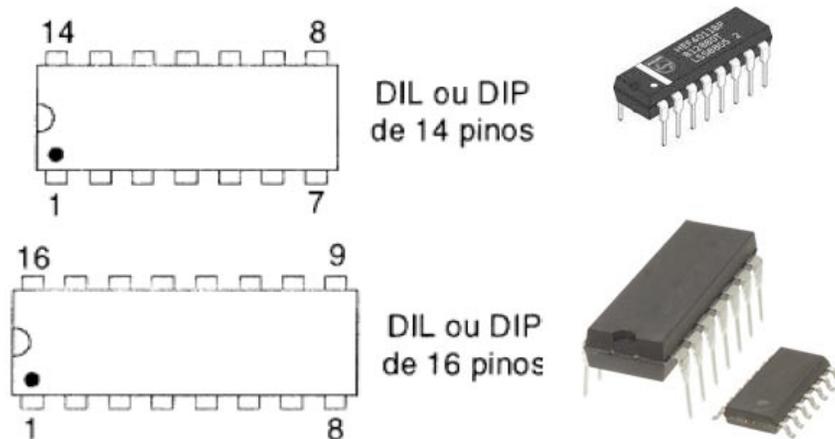


Figura 68 - Uma boa parte dos circuitos integrados TTL está disponível nestes invólucros

Também podemos encontrar funções TTL em invólucros SMD (para montagem em superfície) que são indicados para a produção em massa, feita por máquinas automáticas, conforme mostrado na mesma figura.

Assim, as funções mais simples que constam das portas, e que são disponíveis numa certa quantidade em cada integrado, são as que utilizam circuitos integrados de poucos pinos.

No entanto, à medida que novas tecnologias foram sendo desenvolvidas, permitindo a integração de uma grande quantidade de componentes, surgiu a possibilidade de se colocar num circuito integrado, não apenas umas poucas portas e funções adicionais que serão estudadas futuramente, como também flip-flops, decodificadores e outros mas já interligados de uma determinada forma que seja muito usada ou que tenha uma aplicação específica.

Diversas etapas no aumento da integração foram obtidas recebendo nomes que hoje são comuns quando falamos de equipamentos digitais e computadores em geral.

Temos então as seguintes classificações para os graus de integração dos circuitos digitais:

SSI - Small Scale Integration ou Integração em Pequena Escala que corresponde a série normal dos primeiros TTL. Esta série contém de 1 a 12 portas lógicas num mesmo componente ou circuito integrado.

FPGAs

Atualmente existe a tendência de serem usados circuitos integrados em que as funções lógicas estão todas no chip, mas sem conexão. A conexão entre elas, que determina o que o circuito vai fazer, é feita através de programação. Estes circuitos podem ser extremamente complexos com milhares ou mesmo milhões de funções. Eles partem da idéia que vimos, de que usando portas NAND podemos criar qualquer outra função e, além disso, fazem sua interligação. Na foto, um chip FPGA da Altera.



MSI - Medium Scale Integration ou Integração de Média Escala em que temos num único circuito integrado de 13 a 99 portas ou funções lógicas.

LSI - Large Scale Integration ou Integração em Grande Escala e que corresponde a circuitos integrados contendo de 100 a 999 portas ou funções lógicas.

VLSI - Very large Scale Integration ou Integração em Escala Muito grande que corresponde aos circuitos integrados que contenham mais de 1000 portas ou funções lógicas.

ULSI - Ultra Large Scale of Integration ou Integração em Escala Extremamente Grande que corresponde a circuitos que contenham centenas de milhares ou mesmo milhões de portas ou funções lógicas, tais como DSPs, microcontroladores, microprocessadores e outros.

3.3.1. - Consumo & Velocidade

Quanto maior for a velocidade de um circuito lógico digital, maior é o seu consumo de energia. Da mesma forma, quanto maior for a quantidade de funções integradas num chip, maior é o seu consumo.

Para os projetistas dos chips digitais é, portanto, de fundamental importância pensar nesses dois fatores principalmente quando se trata de um chip que vai ser usado em equipamento alimentado por bateria.

Assim, uma tendência atual é que os dispositivos que tenham escalas muito altas de integração como os microprocessadores e DSPs usem tecnologias que não sejam a TTL, como a CMOS (que veremos mais adiante nesse curso) que se caracteriza por ter um consumo de energia muito mais baixo.

No começo, havia uma restrição ao uso do CMOS dada sua velocidade muito menor do que os chips de tecnologia TTL, mas atualmente, dispositivos CMOS já operam em velocidades muito altas, nada deixando a dever aos dispositivos de tecnologia TTL.

3.4 - Outras Características da Família TTL

Para usar corretamente os circuitos integrados TTL, e mesmo saber testá-los, quando apresentam algum problema de funcionamento, é importante conhecer algumas de suas características adicionais.

Analisemos as principais características, lembrando os níveis lógicos de entrada e saída que são admitidos:

3.4.1 - Correntes de entrada:

Quando uma entrada de uma função lógica TTL está no nível 0, flui uma corrente da base para o emissor do transistor multiemissor da ordem de 1,6 mA, conforme mostra a figura 69.

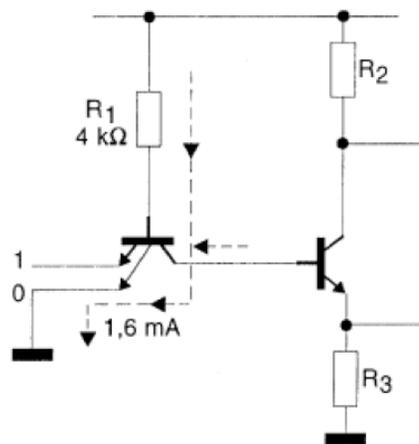


Figura 69 - Corrente de entrada com o nível baixo (0 V)

Esta corrente deve ser levada em conta em qualquer projeto, pois ela deve ser suprida pelo circuito que vai excitar a porta. Quando a entrada de uma porta lógica TTL está no nível alto flui uma corrente no sentido oposto da ordem de $40 \mu\text{A}$, conforme mostra a figura 70.

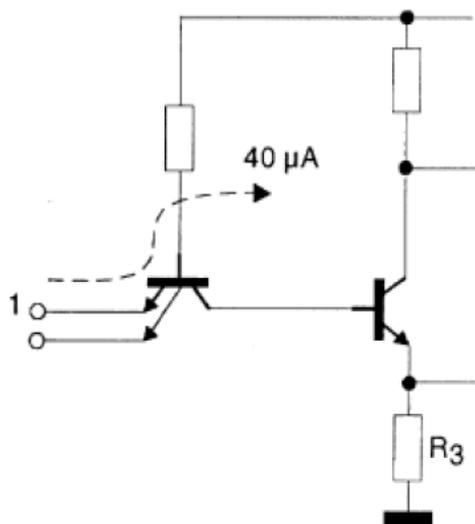


Figura 70 - Corrente de entrada no nível alto (5 V).

Veja que esta corrente corresponde à polarização inversa (fuga) da junção base-emissor do transistor multiemissor de entrada.

Esta corrente vai circular quando a tensão de entrada estiver com um valor superior a 2,0 V.

Se o dispositivo que vai excitar uma entrada de uma função lógica for outro circuito TTL, não precisamos nos preocupar muito com isso, pois ele já é projetado para fornecer as correntes necessárias. O problema é maior quando o dispositivo que deve excitar a entrada é um circuito de outra tecnologia como, por exemplo, uma etapa transistorizada que use um sensor, ou mesmo diretamente um sensor.

Interfaceamento

O interfaceamento entre circuitos lógicos de famílias diferentes exige cuidados, assim como de outros tipos de circuitos. Para a realização deste interfaceamento existem regras que devem ser consultadas.

3.4.2 - Correntes de saída

Quando a saída de um circuito TTL vai ao nível 0 (ou baixo) flui uma corrente da ordem de 16 mA, conforme mostra o circuito equivalente da figura 71.

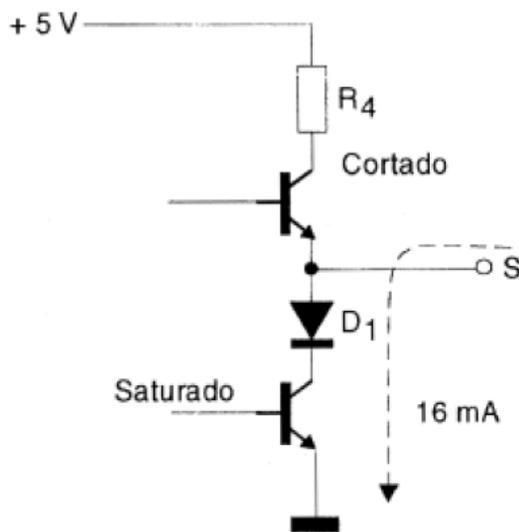


Figura 71 - A corrente de saída no nível baixo

Isso significa que uma saída TTL no nível 0, ou baixo, pode drenar de uma carga uma corrente máxima de 16 mA, ou seja, pode “absorver” uma corrente máxima desta ordem.

Por outro lado quando a saída de uma função TTL está no nível 1, ou alto, ela pode fornecer uma corrente máxima de 400 uA, conforme mostra a figura 72.

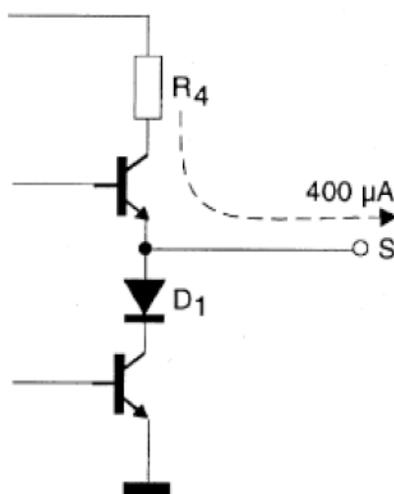


Figura 72 - Corrente de saída no nível alto

Veja então que podemos obter uma capacidade muito maior de excitação de saída de uma porta TTL quando ela é levada ao nível 0 do

que no nível 1. Isso justifica o fato de que em muitas funções indicadoras, em que ligamos um LED na saída, por exemplo, fazemos com que ele seja aceso quando a saída vai ao nível 0 (e portanto a corrente é maior), e não ao nível 1 conforme mostra a figura 73.

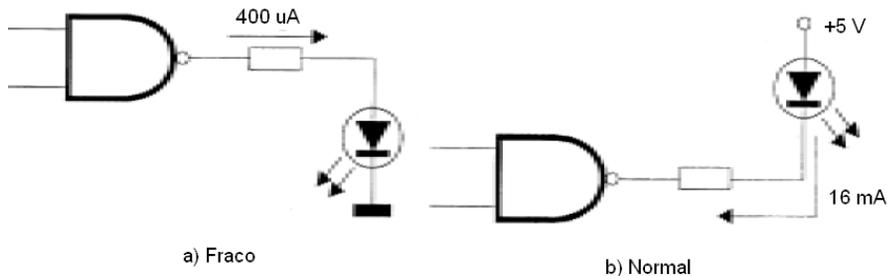


Figura 73 - Acionamento no nível alto e no nível baixo

Também damos preferência ao acionamento no nível baixo quando a carga que exige uma corrente maior é um transistor excitador, ou outro dispositivo que precise de mais corrente.

Se o acionamento precisa ser feito no nível alto, precisaremos usar uma etapa adicional que tenha um ganho maior de corrente.

3.4.3 - Fan In e Fan Out

Fan In e Fan Out são termos técnicos que especificam duas características de extrema importância para os projetos que usam circuitos integrados da família TTL. Eles dizem de uma forma técnica, tudo o que podemos ligar na entrada ou na saída de uma função lógica TTL.

O que ocorre é que a saída de uma função não precisa estar obrigatoriamente ligada a uma única entrada de outra função. A mesma saída pode ser usada para excitar diversas funções ao mesmo tempo.

Como a entrada de cada função precisa de certa corrente, e a saída da função que vai excitar tem uma capacidade limitada de fornecimento ou de drenar corrente, é preciso estabelecer um limite para a quantidade de entradas que podem ser excitadas, conforme mostra a figura 74.

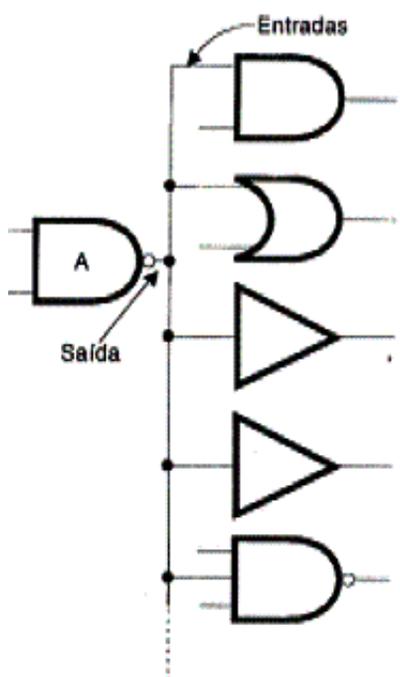


Figura 74 - Uma saída pode excitar várias entradas

Assim, levando em conta as correntes nos níveis 1 e 0 das entradas e saídas, definimos o FAN OUT como o número máximo de entradas TTL que podemos ligar a uma saída TTL.

Para os componentes da família TTL Normal ou Standard que é a que estamos estudando o FAN OUT é 10.

Isso significa que uma única saída de um circuito integrado TTL pode excitar até 10 entradas TTL ao mesmo tempo.

Por outro lado, também pode ocorrer que na entrada de uma função lógica TTL seja preciso ligar mais de uma saída TTL. Considerando novamente que circulam correntes nestas ligações, e que os circuitos têm capacidades limitadas de condução, precisamos saber até que quantidade de ligações pode ser feita em cada entrada.

Usamos então o termo FAN IN para indicar a quantidade máxima de saídas que podemos ligar a uma entrada, conforme mostra a figura 75.

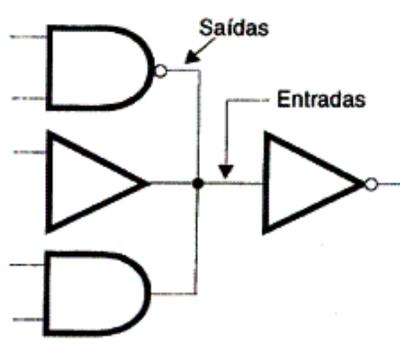


Figura 75 - O Fan IN nas portas TTL

Na prática, por problemas que podem ocorrer se uma saída vai ao nível baixo quando outra está no nível alto, não se recomenda que em nenhum projeto tenhamos FAN IN de mais de 1, a não ser que recursos de proteção sejam usados como, por exemplo, diodos, cuja finalidade, é evitar que as saídas das funções excitadoras possam ser colocadas em curto.

3.4.4 - Velocidade

Os circuitos eletrônicos possuem uma velocidade limitada de operação a qual depende de diversos fatores.

Esses fatores incluem o tempo de trânsito dos portadores de carga nos materiais semicondutores de que é formado o chip, da capacitância que, com a resistência de entrada dos circuitos, forma um circuito de tempo, além das indutâncias parasitas.

No caso específico dos circuitos TTL, temos principalmente de considerar a própria configuração das portas que apresentam indutâncias e capacitâncias parasitas as quais influem na sua velocidade de operação de uma forma bastante acentuada.

Assim se levarmos em conta a configuração típica de uma porta, conforme mostra o circuito da figura 76, vemos que se for estabelecida uma transição muito rápida da tensão de entrada, as tensões nos diversos pontos do circuito não se alteram com a mesma velocidade.

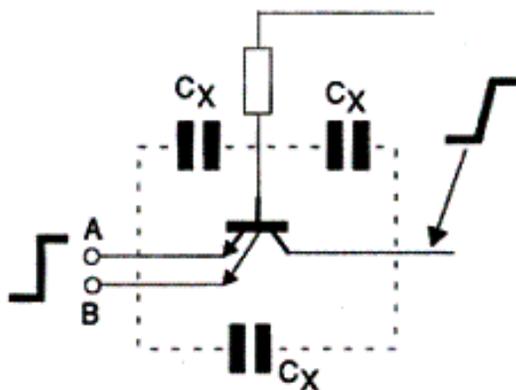


Figura 76 - As capacitâncias parasitas de uma entrada TTL

Este sinal tem antes de carregar as capacitâncias parasitas existentes, de modo que a tensão de entrada suba gradualmente, demorando certo tempo que deve ser considerado em qualquer projeto de alta velocidade no qual está o dispositivo.

Da mesma forma, à medida que o sinal vai passando pelas diversas etapas do circuito, temos de considerar os tempos que os componentes levam para comutar, justamente em função das capacitâncias e indutâncias parasitas existentes.

O resultado disso é que para os circuitos integrados TTL, há um retardo, não desprezível, entre o instante em que o sinal passa do nível

Regras de interfaceamento

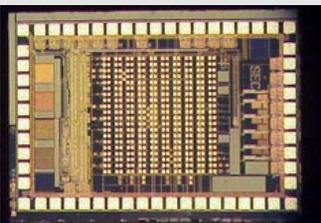
Mais uma vez lembramos que é preciso ter muito cuidado quando interligamos diversos circuitos digitais.

Dissipação

Um outro problema que deve ser considerado é que na subida ou na descida do sinal aos níveis lógicos alto e baixo, temos um trecho em que o componente apresenta uma variação de resistência, responsável pela dissipação de calor. Assim, a maior parte da energia dissipada pelo componente ocorre na comutação, por isso, quanto mais comutações ele realiza, mais calor ele dissipa. Essa é uma das limitações para a velocidade máxima de operação dos circuitos lógicos.

Velocidade Cada vez Maior

As velocidades de operação dos chips lógicos aumentam cada vez mais. Existe um limite teórico, que é determinado pelo o próprio tamanho dos componentes da pastilha. Quanto menor for o tamanho, mais rápido podem passar os sinais através deles. Segundo se acredita, o limite está próximo de ser alcançado, mas novas tecnologias, como o uso de litografia ultravioleta a Laser na fabricação dos CIs, possibilitando a integração de componentes ainda menores e o uso de materiais em que os elétrons podem se deslocar a velocidades maiores, como o Arseneto de Gálio, deve levar a circuitos digitais com velocidades muitas vezes maiores do que as que temos hoje. Temos ainda a intervenção da plasmônica que pode ler a chips ópticos onde os sinais são transferidos através da luz e não de correntes. Na foto, um chip de silício com um dispositivo plasmônico integrado



0 para o 1 na entrada, e o instante em que o sinal na saída responde a este sinal, passando do nível 1 para o 0, no caso de um inversor tomado como exemplo.

Da mesma forma, existe um retardo entre o instante em que o sinal de entrada passa do nível 1 para o 0, e o instante em que o sinal de saída passa do nível 0 para o 1, no caso do mesmo inversor tomado como exemplo.

Esses dois tempos são mostrados na figura 77 e são muito importantes nas especificações dos circuitos TTL, principalmente quando trabalhamos com o projeto de dispositivos muito rápidos.

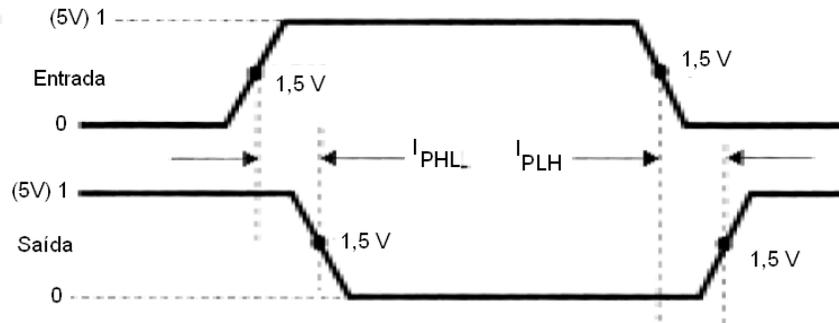


Figura 77 - Os tempos de transição dos sinais numa porta TTL

Esses tempos são ainda mais prolongados se o dispositivo reunir muitas funções lógicas.

O problema principal que esses sinais causam, e que podemos adiantar para o leitor, é que, se dois sinais que devam chegar ao mesmo tempo a um certo ponto do circuito, não o fizerem porque um se retarda mais do que o outro, ao passar por determinadas funções, isso pode gerar interpretações erradas do próprio circuito que funcionará de modo anormal.

O circuito estará ainda com um 0 numa entrada (a), aguardando a chegada de um 1, quando a (b) já recebeu o 1 de outro circuito, fornecendo então uma saída que corresponde a 0 e 1 de entrada, e não 1 e 1. Isso pode mudar todo o comportamento do circuito levando-o a resultados que não são os esperados.

3.5 - Subfamílias TTL

Os primeiros circuitos integrados TTL que foram desenvolvidos, logo se mostraram impróprios para certas aplicações como, por exemplo, quando se desejava maior velocidade, menor consumo de energia ou ainda os dois fatores reunidos. Naquela época, as outras tecnologias que poderiam substituir os circuitos integrados TTL ainda eram pouco desenvolvidas.

Isso fez com que, mantendo as características originais de compatibilidade entre os circuitos, e mantendo as mesmas funções bási-

cas, fossem criadas subfamílias, agregando uma característica adicional diferenciada.

Assim, a partir da família original, que foi denominada “Standard” ou “Normal”, surgiram diversas subfamílias. Para diferenciar essas subfamílias foram adicionadas no número que identifica o componente, depois do 54 ou 74 com que todos começam, uma letra ou duas letras.

Temos então a seguinte tabela que identifica as subfamílias da família TTL standard:

Indicação: 54/74

Família/Subfamília: Normal ou Standard

Características: baixo custo, média velocidade

Indicação: 54L/74L

Família/Subfamília: Low Power

Característica: baixo consumo

Indicação: 574H/74H

Família/Subfamília: High Speed

Característica: alta velocidade

Indicação: 54S/74S

Família/Subfamília: Schottky

Característica: alta velocidade

Indicação: 54LS/74LS

Família/Subfamília: Low Power Schottky

Característica: alta velocidade e baixo consumo

A versão standard ou normal é a que apresenta os componentes com o custo mais baixo e que também dispõe da maior quantidade de funções.

No entanto, a versão LS é a que se adapta mais aos circuitos de computadores e instrumentação rápida e equipamentos portáteis, pois tem a mesma velocidade dos componentes da família standard com muito menor consumo.

Algumas características podem ser comparadas para que os leitores tenham uma idéia das diferenças:

* **Velocidade**

A velocidade de operação de uma função TTL normalmente é especificada pelo tempo que o sinal leva para se propagar através do circuito. Em uma linguagem mais simples, trata-se do tempo que ocorre entre o instante que aplicamos os níveis lógicos na entrada e o instante em que obtemos a resposta, conforme mostra a forma de onda que vimos na figura 77.

Diodos Schottky

Os diodos Schottky são dispositivos que começam a conduzir com uma tensão muito baixa, da ordem de 0,2 V ou menos, por isso, podem resultar em dispositivos muito rápidos quando usados em comutação. Usando a sua tecnologia na junção de transistores, podemos elaborar circuitos digitais muito rápidos. (veja Volume 2 - Eletrônica Analógica)

Para os circuitos da família TTL é comum especificar estes tempos em nanossegundos ou bilionésimos de segundo.

Assim, temos:

Família/Sub-Família	Tempo de propagação (ns)
TTL Standard	10
Low Power	33
Low Power Schottky	10
High Speed	6
Schottky	3

Em muitos manuais de componentes TTL, entretanto, em lugar do tempo de propagação em nanossegundos, podemos encontrar a velocidade ou frequência máxima de operação do dispositivo, a qual será dada em Megahertz. Assim, uma porta TTL comum que tenha um tempo de propagação de 10 ns, quando convertemos esse valor para termos de frequência, obtemos uma frequência máxima de operação da ordem de 100 MHz.

* Dissipação

Outro ponto importante no projeto de circuitos digitais é a quantidade de energia ou a potência consumida e, portanto, dissipada na forma de calor.

Em especial, a potência consumida se torna muito importante quando pensamos em dispositivos alimentados por baterias.

Na prática, em nossos dias, como os dispositivos de tecnologia CMOS têm muito menor consumo que os equivalentes TTL, eles têm sido preferidos quando se trata de projetos alimentados por pilhas e baterias.

Outra vantagem destes dispositivos é que eles podem operar com tensões muito mais baixas do que as usadas pelos circuitos de tecnologia TTL.

Também devemos levar em conta os casos em que devemos usar uma grande quantidade de funções num circuito.

Quando usamos uma grande quantidade de funções, esta característica se torna importante tanto para o dimensionamento da fonte como para o próprio projeto da placa e do aparelho que deve ter meios de dissipar o calor gerado.

Podemos então comparar as dissipações das diversas famílias, tomando como base uma porta ou “gate”:

Família/Subfamília	Dissipação por Porta (mW)
Standard	10
Low Power	1
Low Power Schottky	2
High Speed	22
Schottky	20

O leitor já deve ter percebido que um problema importante que se manifesta é que, quando aumentamos a velocidade o consumo também aumenta. O projetista deve, portanto, ser cuidadoso em escolher a subfamília que una as duas características na medida certa de sua precisão, incluindo nisso o preço dos componentes.

Também deve estar atento sobre a possibilidade de se utilizar outras tecnologias, com vantagem, como por exemplo, a CMOS que estudaremos mais adiante.

* **Imunidade ao ruído**

Outra característica importante que deve ser observada ao se utilizar uma certa família de circuitos integrados TTL é a sua imunidade ao ruído.

Ambientes ruidosos, como dentro de um veículo, uma fábrica ou ainda na rua, podem exigir que os circuitos utilizados sejam imunes a presença de ruídos. Picos de ruídos podem ser interpretados como sinais e causar um processamento errôneo.

A imunidade ao ruído é expressa em milivolts (mV) ou volts (V), indicando qual é a amplitude máxima do ruído que pode ser sobreposta ao sinal lógico, sem que isso cause uma interpretação errônea da função.

3.6 - Compatibilidade entre as subfamílias

Um ponto importante que deve ser levado em conta quando trabalhamos com a família Standard e as sub-famílias TTL é a possibilidade de interligarmos os diversos tipos.

Isso realmente ocorre já que todos os circuitos integrados da família TTL, e também das subfamílias são alimentados com 5 volts.

O que devemos observar, e com muito cuidado, é que as correntes que circulam nas entradas e saídas dos componentes das diversas subfamílias são completamente diferentes, o que quer dizer que, quando passamos de uma para outra tentando interligar os seus componentes, as regras de Fan In e Fan Out mudam completamente.

Na verdade não podemos falar de Fan-in e Fan-out quando interligamos circuitos de famílias diferentes. O que existe é a possibilidade de se elaborar uma tabela, a partir das características dos componentes, em que a quantidade máxima de entradas de determinada subfamília pode ser ligada na saída de outra subfamília.

Esta tabela , dada a seguir:

Ruídos

Veja nos volume anteriores as lições que tratam sobre EMI (Interferência Eletromagnética) e outros ruídos que podem interferir no funcionamento dos circuitos eletrônicos.

		Saída				
		74L	74	74LS	74H	74S
Entrada	74L	20	40	40	50	100
	74LS	2,5	10	5	12,5	12,5
	74	10	20	20	25	50
	74H	2	8	4	10	10
	74S	2	8	4	10	10

Observamos, por esta tabela, que uma saída 74 (Standard ou Normal) pode excitar convenientemente 10 entradas 74LS (low Power Schottky).

Na figura 78 mostramos como isso pode ser feito.

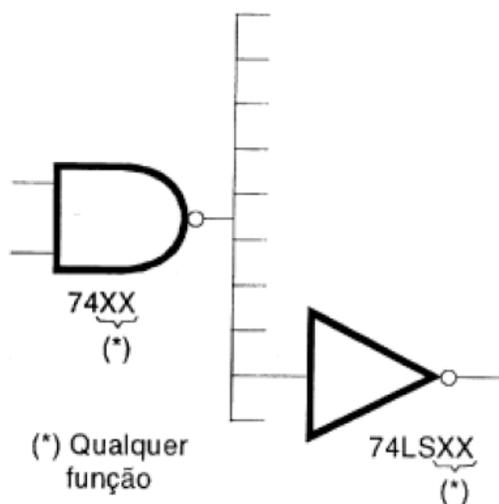


Figura 78 - Interfaceamento entre funções

3.7 - Open Collector e Totem-Pole

Os circuitos comuns TTL que estudamos até agora, e que tem a configuração mostrada na figura 54 deste capítulo, são denominados Totem Pole.

Nestes circuitos temos uma configuração em que um ou outro transistor conduz a corrente conforme o nível estabelecido na saída seja 0 ou 1.

Este tipo de circuito apresenta um inconveniente se ligarmos duas portas em paralelo, conforme mostra a figura 79.

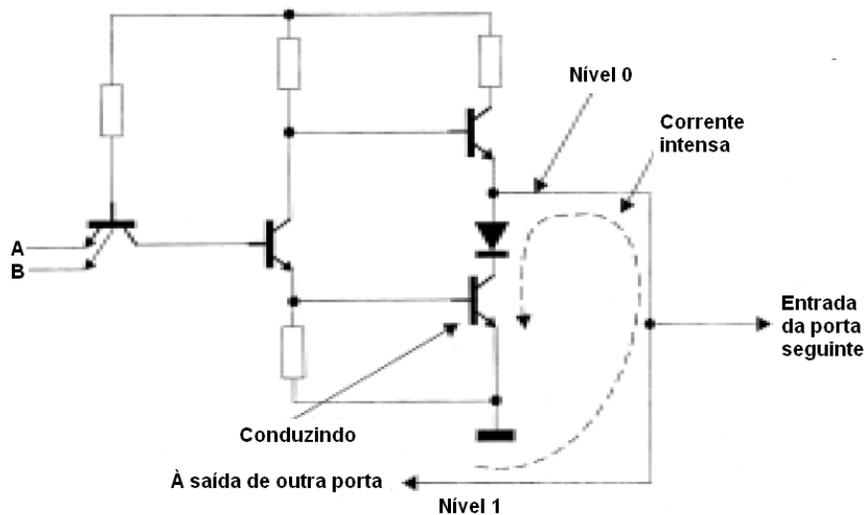


Figura 79 - Conflito de níveis na saída de uma porta

Se uma das portas tiver sua saída indo ao nível alto (1) ao mesmo tempo em que a outra vai ao nível baixo (0), estabelece-se um curto-circuito na saída que pode causar sua queima. A corrente intensa que passa a circular pelo circuito, sem qualquer limitação é a causa do problema.

Isso significa que os circuitos integrados TTL com esta configuração nunca podem ter suas saídas interligadas da forma indicada.

Esse é um dos problemas que impede que uma entrada seja excitada por diversas saídas TTL “totem pole” ligadas em paralelo.

Em algumas aplicações, entretanto, é preciso ligar diversas saídas a uma entrada para excitação, não ao mesmo tempo, mas em instantes diferentes. Para isso, existe a possibilidade de se elaborar circuito em que as saídas de portas sejam interligadas com transistores internos numa configuração diferente. Isso é conseguido com a configuração denominada Open Collector ou Coletor Aberto que é mostrada na figura 80.

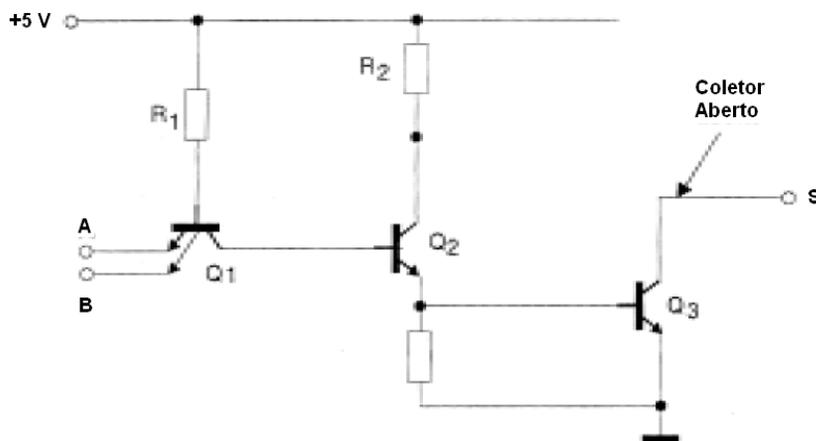


Figura 80 - Saída Open-Collector ou coletor aberto

Os circuitos integrados TTL que possuem esta configuração, são indicados como “open collector” e quando são usados, exigem a ligação de um resistor externo denominado “pull-up”, normalmente de 2000 ohms ou próximo disso.

Como o nome em inglês diz, o transistor interno está com o “coletor aberto” (open collector) e para funcionar precisa de um resistor de polarização. O valor 2 000 ohms ou 2 200 ohms é padronizado para fornecer a polarização que o circuito precisa para funcionar, sem problemas.

A vantagem desta configuração está na possibilidade de interligarmos portas diferentes num mesmo ponto, conforme mostra a figura 81.

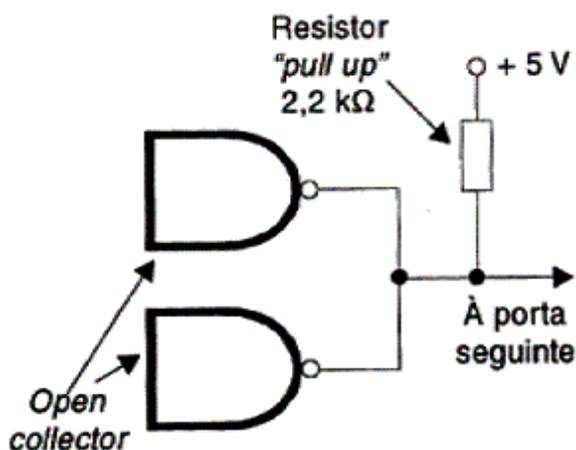


Figura 81 - O resistor pull-up

No entanto, a desvantagem está na redução da velocidade de operação do circuito que se torna mais lento, com a presença do resistor, pois ele tem uma certa impedância que afeta o desempenho do circuito, “atrasando” o sinal lógico que atravessa a função.

3.8 - Tri-State

Tri-state significa “terceiro estado” ou “três estados” e é uma configuração que também pode ser encontrada em alguns circuitos integrados TTL, principalmente de uso em informática e controles industriais que envolvem o uso de microcontroladores, microprocessadores e DSPs.

Os circuitos lógicos com saídas tri-state possuem características que se adaptam a sua ligação a circuitos externos, que tanto podem ser saídas como entradas de sinais.

Na figura 82 temos um circuito típico de uma porta NAND tri-state que vai nos servir como exemplo.

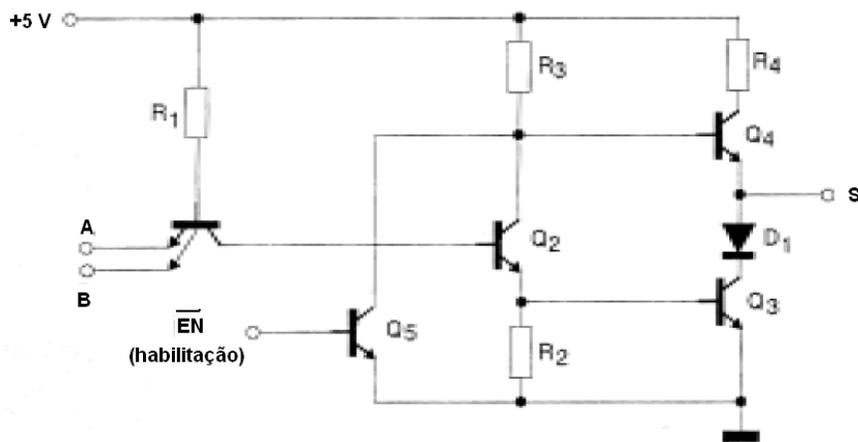


Figura 82 - Uma porta NAND Tri-state

O que ocorre é que podem existir aplicações em que duas portas tenham suas saídas ligadas num mesmo circuito, conforme mostra a figura 83.

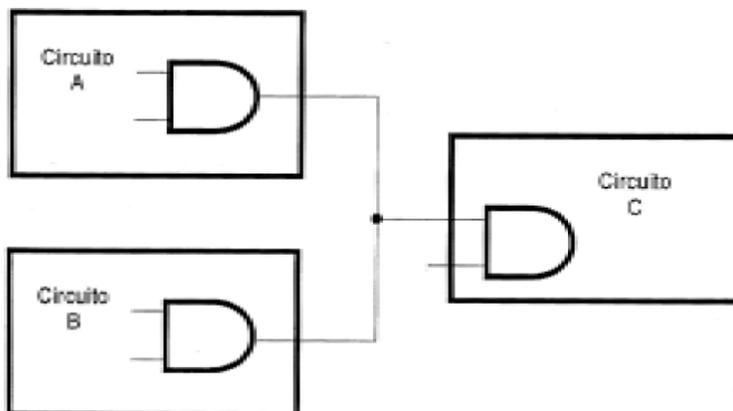


Figura 83- Quando A estiver enviando sinais para C, B deve estar desativada

Uma porta está associada a um primeiro circuito e a outra porta a um segundo circuito. Quando um circuito envia seus sinais para a porta o outro deve ficar em espera.

Ora, se o circuito que está em espera ficar no nível 0 ou no nível 1, estes níveis serão interpretados pela porta seguinte como informação e isso não pode ocorrer.

O que deve ocorrer é que quando uma porta estiver enviando seus sinais a outra porta, ela deve estar numa situação em que na sua saída não tenhamos nem 0 e nem 1, ou seja ela deve ficar num estado de circuito desligado, circuito aberto ou terceiro estado (tri-state).

Isso é conseguido através de uma entrada de controle denominada “habilitação”, que em inglês é dito “enable”, e abreviada por EN.

Assim, quando EN está no nível de habilitação 0, no circuito da figura 80, o transistor Q5 não conduz, e nada acontece no circuito que funciona normalmente.

No entanto, se EN for levada ao nível 1, o transistor Q5 satura levando tanto Q3 como Q4 ao corte, ou seja, os dois passam a se comportar como circuitos abertos, independentemente dos sinais de entrada. Na saída Y o que temos é então um estado de alta impedância ou circuito aberto.

Podemos então concluir que uma função tri-state apresenta três estados possíveis na sua saída:

Nível lógico 0
Nível lógico 1
Alta Impedância

As funções tri-state são muito usadas nos circuitos de computadores, nos denominados barramentos de dados ou “data bus”, onde diversos circuitos devem aplicar seus sinais ao mesmo ponto ou devem compartilhar a mesma linha de transferência desses dados.

Com o uso de saídas tri-state ligadas a estes barramentos não ocorrem problemas tanto quando dados devem entrar como devem sair.

Em muitas máquinas industriais, que devem coletar informações de diversos circuitos sensores ao mesmo tempo e, portanto, ligados ao mesmo ponto, este tipo de lógica também é muito usada, ocorrendo o mesmo com robôs e outros equipamentos mecatrônicos.

O circuito que está funcionando deve estar habilitado e os que não estão funcionando, para que suas saídas não influenciem nos demais, devem ser levadas sempre ao terceiro estado.

Na figura 84 temos um exemplo de aplicação em que são usados circuitos tri-state.

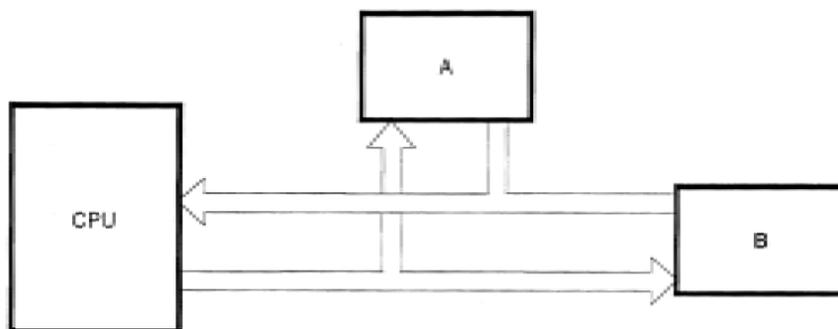


Figura 84 - Circuito de interface com componentes tri-state

Uma unidade de processamento de um computador envia e recebe dados para/de diversos periféricos, usando uma única linha ou barramento (bus). Todos os circuitos ligados a estas linhas devem ter suas saídas do tipo tri-state.

Termos em Inglês:

Em eletrônica digital, muitos dos termos técnicos originais em inglês são mantidos mesmo na literatura e documentação em português. Dentre destacamos os termos “gate”, FAN-IN, FAN-OUT, Tri-state, Pull-Up e muitos outros. No entanto, há uma certa quantidade deles que merecem uma tradução:

Logic - lógica
Low Power - baixa potência
High Speed - alta velocidade
Standard - Padrão
Family - família
Series - série
Open - aberto
Totem - Totem (um sobre o outro)
Pull - colocar (por)
Up - acima
Down - abaixo

Termos para pesquisa:

- * História do TTL
- * Famílias de circuitos digitais
- * Totem pole
- * Pull-up
- * Tri-state

QUESTIONÁRIO

1. Quais são as duas principais famílias de circuitos lógicos digitais obtidas na forma de circuitos integrados?
 - a) CMOS e TTL
 - b) Schottky e LS
 - c) AO e Solid State
 - d) FET e Bipolar

2. Qual é a tensão de alimentação dos circuitos integrados da família TTL Standard?
 - a) 3 a 15 V
 - b) 1,5 V
 - c) 5 V
 - d) 12 V

3. Circuitos integrados que contenham grande quantidade de funções, mais de 1 000, e são usados principalmente nos modernos computadores, máquinas de controle industrial e mecatrônica são denominados:
 - a) SSI
 - b) MSI
 - c) LSI
 - d) VLSI

4. Um circuito integrado TTL tem uma capacidade maior de corrente na sua saída quando:
 - a) No nível 1
 - b) No nível 0
 - c) As capacidades são iguais nos dois níveis
 - d) A capacidade depende da função

5. A família TTL de alta velocidade tem seus componentes com a sigla:
 - a) 74L
 - b) 74H
 - c) 74S
 - d) 74LS

6. Para que tipos de configurações, não podemos ligar duas saídas juntas?
 - a) Todas
 - b) Totem pole
 - c) Open Collector
 - d) Nenhuma delas

7. Que estado encontramos numa saída de uma função TTL Tri-state quando a entrada de habilitação não está ativada?
 - a) Nível 0
 - b) Nível 1
 - c) Nível 0 ou 1
 - d) Alta impedância



» A Família de Circuitos Integrados CMOS

No capítulo anterior mostramos aos leitores que os circuitos integrados digitais são organizados em famílias, de modo a se manter uma compatibilidade de características que permita sua interligação direta, sem a necessidade de qualquer componente adicional. Vimos na ocasião que as famílias contam com dezenas, ou mesmo centenas de funções, que funcionam como blocos ou tijolos, a partir dos quais podemos “construir” qualquer circuito eletrônico digital, por mais complexo que seja. Na verdade, os próprios blocos tendem a ser cada vez mais completos e complexos, com a disponibilidade de circuitos integrados que contenham milhares, dezenas de milhares, centenas de milhares e até milhões de funções, já interligadas de modo a exercer uma tarefa que seja muito utilizada. É o caso dos circuitos integrados VLSI e ULSI de apoio encontrados nos computadores, controles de máquinas industriais (PLCs), DSPs, FPGAs onde milhares ou mesmo milhões de funções lógicas já estão interligadas de modo a exercer dezenas ou centenas de funções comuns nestes equipamentos. No capítulo anterior também estudamos a família TTL e suas subfamílias que são muito comuns na maioria dos equipamentos eletrônicos, analisando as principais funções disponíveis e suas características elétricas. No entanto, existem outras famílias, e uma que é muito utilizada é justamente a que vamos estudar nesta lição de nosso curso, denominada a família CMOS. Se bem que as duas famílias CMOS e TTL tenham características diferentes, elas não são incompatíveis. Na verdade, conforme veremos elas podem ser interligadas em determinadas condições que o leitor deve conhecer, e que também serão abordadas neste capítulo. Como estas duas famílias correspondem praticamente a quase tudo que se pode fazer em matéria de circuitos digitais de forma simples e prática, o seu conhecimento dará ao leitor as bases necessárias para o trabalho com este tipo de componente.

Itens:

- 4.1- Os circuitos integrados CMOS
- 4.2 – Aplicações digitais
- 4.3 – Consumo e velocidade
- 4.4 – Sub-famílias CMOS

- 4.5 – Sensibilidade e manuseio
- 4.6 – As configurações CMOS
- 4.7 – Especificações
- 4.8 - Interfaceando
- 4.9 – Fontes de alimentação

Objetivos:

- Conhecer a família de circuitos digitais CMOS
- Entender como funcionam as principais funções e como utilizá-las na prática
- Saber como manusear os circuitos integrados CMOS
- Saber como ligar circuitos desta família a circuitos lógicos digitais de outras famílias, e também de outros tipos
- Interpretar as especificações dos circuitos integrados digitais comerciais

4.1 – Os circuitos integrados CMOS

CMOS significa “Complementary Metal-Oxide Semiconductor”, sigla que corresponde a um tipo de tecnologia que utiliza transistores de efeito de campo, ou “field effect transistor” (FET), em lugar dos transistores bipolares comuns (como nos circuitos TTL), para elaboração dos circuitos integrados digitais.

Existem vantagens e desvantagens no uso de transistores de efeito de campo, mas os fabricantes, com o desenvolvimento de novas tecnologias de fabricação, conseguem pouco a pouco eliminar as diferenças que existem entre as duas famílias, aumentando ainda mais as suas velocidades e reduzindo seu consumo de energia.

De uma forma geral, podemos dizer que existem aplicações em que é muito mais vantajoso usar circuitos integrados TTL e aplicações em que é melhor usar circuitos integrados CMOS.

Os transistores de efeito de campo usados nos circuitos integrados CMOS ou MOSFETs têm a estrutura básica mostrada na figura 85 onde também aparece seu símbolo.

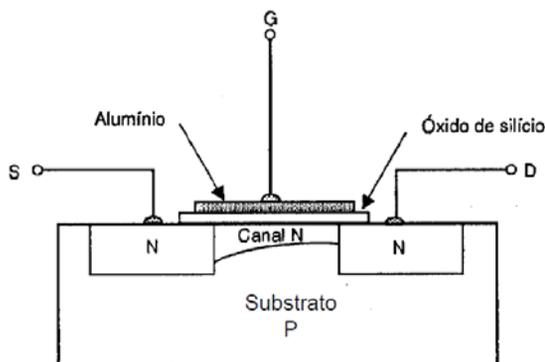


Figura 85 – Estrutura de um transistor MOS de canal N

Esta estrutura tem passado por alterações a cada geração de novos componentes, no sentido de se obter maior velocidade e menor consumo. No entanto, o princípio de funcionamento que passamos a estudar, se mantém.

Conforme podemos ver, o eletrodo de controle é a comporta ou gate (g), onde se aplica o sinal que deve ser amplificado ou ainda deve ser usado para chavear o circuito.

O transistor é polarizado então de modo a haver uma tensão entre a fonte ou “source” (s) e o dreno ou “drain” (d).

Fazendo uma analogia com o transistor bipolar, podemos dizer que a comporta do MOSFET equivale à base do transistor bipolar, enquanto que o dreno equivale ao coletor, e a fonte ao emissor, conforme mostra a figura 86.

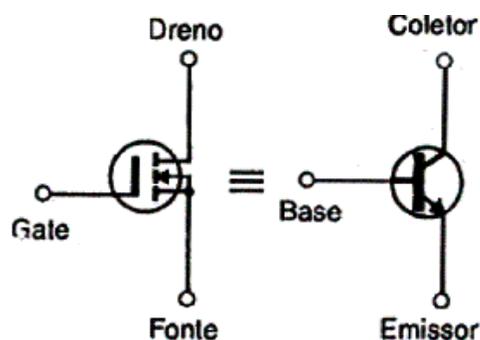


Figura 86 – Equivalência entre o transistor MOS e o bipolar

Observe que entre o eletrodo de comporta, que consiste numa placa de alumínio, e a parte que forma o substrato ou canal por onde passa a corrente, não existe contato elétrico e nem junção de materiais semicondutores, mas sim uma finíssima camada de óxido de alumínio ou óxido metálico, que justamente dá nome ao dispositivo (metal-oxide).

A polaridade do material semicondutor usado no canal, que é a parte do transistor por onde circula a corrente controlada, determina seu tipo e também a polaridade da tensão que a controla.

Assim, encontramos na prática transistores de efeito de campo tipo MOS de canal N, e transistores de efeito de campo tipo MOS de canal P.

É interessante observar que os transistores de canal N são mais rápidos do que os de canal P, pois os portadores negativos de cargas (elétrons) são mais rápidos que as lacunas.

Na verdade, os próprios transistores MOS podem ainda ser divididos em dois tipos: enriquecimento e empobrecimento que levam a dois tipos de representação. Para nosso curso, entretanto, é suficiente lembrar que existem transistores MOS tipo P e tipo N.

Na figura 87 temos os símbolos adotados para representar os dois tipos de transistores.

Mais sobre transistores

Para saber mais sobre o funcionamento dos transistores MOS e bipolares sugerimos consultar o Curso de Eletrônica – Eletrônica Analógica – Vol 2 do mesmo autor deste livro.

Amplificador de corrente e amplificador de tensão

Os transistores bipolares são típicos amplificadores de corrente, pois corrente de base determina corrente de coletor. Os MOSFETs, como as válvulas, são típicos amplificadores de tensão, pois tensão de gate determina corrente de dreno.

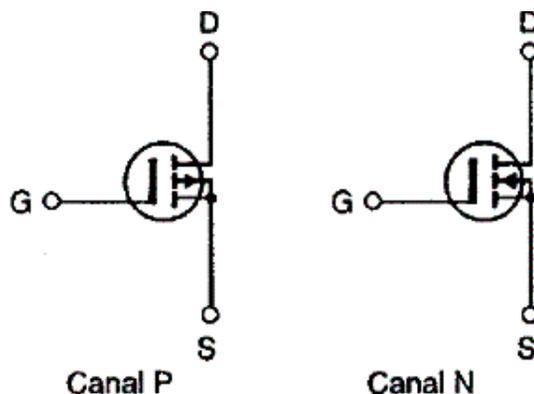


Figura 87 – Tipos de transistores MOS com seus símbolos (observe a direção da seta)

Podemos dizer, de uma forma que não pode ser levada muito longe, que estes transistores são equivalentes aos tipos NPN e PNP bipolares, se considerarmos o sentido de circulação das correntes que eles controlam.

Num transistor de efeito de campo, a corrente que circula entre a fonte e o dreno pode ser controlada pela tensão aplicada à comporta. Isso significa que, diferentemente dos transistores bipolares em que a corrente de coletor depende da corrente de base, no transistor de efeito de campo, a corrente do dreno depende da tensão de comporta.

Assim, no tipo P uma tensão positiva de comporta aumenta sua condução, ou seja, faz com que ele sature enquanto que no tipo N uma tensão negativa de comporta é que o leva à saturação.

Mais uma vez fazendo uma comparação com os tipos bipolares, podemos dizer então que, enquanto os transistores bipolares são típicos amplificadores de corrente, os FETs ou transistores de efeito de campo MOS são típicos amplificadores de tensão.

Esta diferença leva o transistor de efeito de campo CMOS a apresentar características muito interessantes para aplicações tanto em eletrônica digital como analógica.

Uma delas está no fato de que a impedância de entrada do circuito é extremamente elevada, o que significa que precisamos praticamente somente de tensão para controlar os dispositivos CMOS. Assim, é preciso uma potência extremamente baixa para o sinal que vai excitar a entrada de um circuito integrado CMOS, já que praticamente nenhuma corrente circula por este elemento.

Essa característica torna o transistor CMOS um dispositivo capaz de operar com quantidades de energia muito baixas, ou seja, a ter um consumo apropriado para as aplicações que sejam alimentadas por pilhas e baterias.

A outra está no fato de que, diferentemente dos transistores bipolares que só começam a conduzir quando uma tensão da ordem de 0,6 V vence a barreira de potencial de sua junção base-emissor, os

FETs não têm esta descontinuidade de características, o que os torna muito mais lineares em qualquer aplicação que envolva amplificação de sinais.

Na figura 88 temos as curvas características de um MOSFET de canal N.

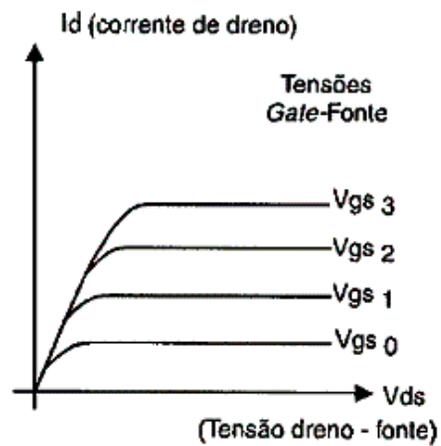


Figura 88 – Característica do transistor MOSFET

No entanto, ao lado das vantagens, os transistores de efeito de campo MOS também têm suas desvantagens. Uma delas é que o eletrodo de comporta (gate) se comporta como a placa de um capacitor. Assim, quando aplicamos ao transistor uma tensão para comutá-lo, antes que ela tenha algum efeito sobre o componente é preciso carregar este capacitor, o que leva algum tempo.

Esta capacitância de comporta é, portanto, o principal responsável pela baixa velocidade desses componentes.

Tecnologias modernas vêm conseguindo reduzir bastante estas capacitâncias, e os circuitos que usam transistores MOS estão se tornando cada vez mais rápidos, em alguns casos, já podendo ser comparados aos circuitos bipolares usados na tecnologia TTL.

4.2 - Aplicações digitais

Da mesma forma que podemos elaborar funções lógicas básicas usando transistores bipolares comuns, também podemos fazer o mesmo tomando por base nos transistores de efeito de campo MOS. A tecnologia CMOS (Complementary MOS) permite que os dispositivos tenham características excelentes para aplicações digitais.

CMOS significa que em cada função temos configurações em que transistores de canal N e de canal P são usados ao mesmo tempo, ou seja, usamos pares complementares conforme mostra o diagrama do inversor lógico mostrado na figura 89. (O C de complementar significa o uso dos dois tipos de transistores)

Subfamílias CMOS

Nos próximos itens teremos informações sobre subfamílias CMOS tão rápidas como as TTL, mas com o baixo consumo que caracteriza os componentes CMOS.

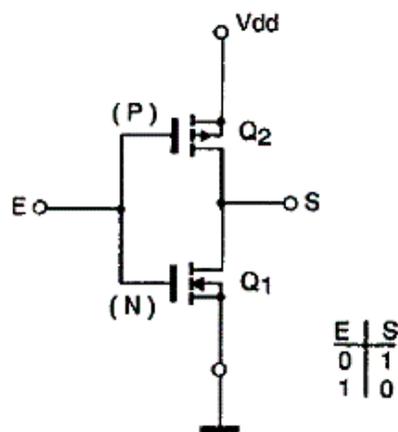


Figura 89 – Um inversor CMOS

Conforme explicamos no item anterior, a polaridade da tensão que controla a corrente principal nos transistores de efeito de campo MOS depende justamente do tipo de material usado no canal, que pode ser tanto do tipo P quando do tipo N.

Assim, se levarmos em conta que nos circuitos digitais temos dois níveis de sinal possíveis podemos perceber que, dependendo do nível deste sinal aplicado à comporta dos dois transistores ao mesmo tempo, quando um deles estiver polarizado no sentido de conduzir plenamente a corrente (saturado), o outro estará obrigatoriamente polarizado no sentido de cortar esta corrente (corte).

No circuito indicado na figura 88, quando a entrada A estiver no nível baixo (0), o transistor Q2 conduz enquanto que Q1 permanece no corte. Isso significa que Vdd que é a tensão de alimentação positiva é colocada na saída o que corresponde ao nível alto ou 1.

Por outro lado, quando na entrada aplicamos o nível alto, que corresponde ao Vdd (tensão de alimentação), é o transistor Q1 que conduz e, com isso, o nível baixo ou 0 V (também indicado por Vss), que será colocado na saída.

Conforme sabemos, estas características correspondem justamente a função inversora.

Um inversor CMOS é, portanto, uma configuração muito simples de se implementar, pois usa apenas dois transistores complementares.

A partir desta configuração, uma grande quantidade de funções pode ser implementada com base em transistores CMOS.

4.3 – Consumo e velocidade

Analisando o circuito inversor tomado como base para nossas explicações, vemos que ele apresenta duas características importantes.

A primeira é que sempre um dos transistores estará cortado, qualquer que seja o sinal de entrada (alto ou baixo), o que significa

que praticamente não circula corrente alguma entre o Vdd e o ponto de terra (0 V). A única corrente que circulará é eventualmente de um circuito externo excitado pela saída, conforme mostra a figura 90.

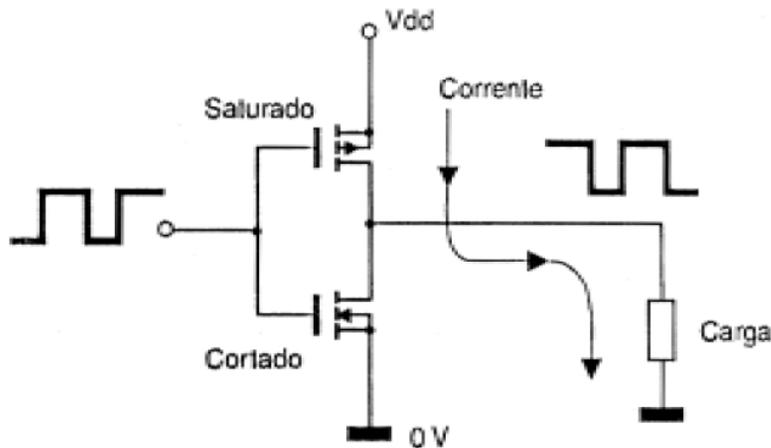


Figura 90 – A única corrente é a que passa pelo circuito externo

Isso significa um consumo extremamente baixo para este par de transistores em condições normais, já que na entrada do bloco seguinte, que também será um circuito CMOS, teremos uma impedância elevadíssima e praticamente nenhuma corrente circula. Este consumo é da ordem de apenas 10 nW (nW = nanowatt = 0,000 000 001 watt).

A corrente maior circula justamente no momento em que os transistores comutam, pois conforme vimos para excitar a etapa seguinte que é a entrada de um CMOS, deve ser carregado o capacitor que corresponde ao eletrodo de comporta, conforme explicaremos em detalhes mais adiante.

É fácil perceber que, se integrarmos 1 milhão de funções destas num circuito integrado, ele vai consumir apenas 1 mW! É claro que na prática temos fatores que tornam maior este consumo como, por exemplo, eventuais fugas, a necessidade de um ou outro componente especial de excitação que exija maior corrente, e a própria velocidade de operação que determina a velocidade com que o capacitor virtual de entrada deve ser carregado e descarregado.

Conforme vimos, ao lado das boas características ele também tem seus problemas, e justamente um deles está no fato de que o eletrodo de controle (comporta) que é uma placa de metal fixada no material semiconductor e isolada por uma camada de óxido, que funciona como a armadura ou placa de um capacitor, conforme mostra a figura 91.

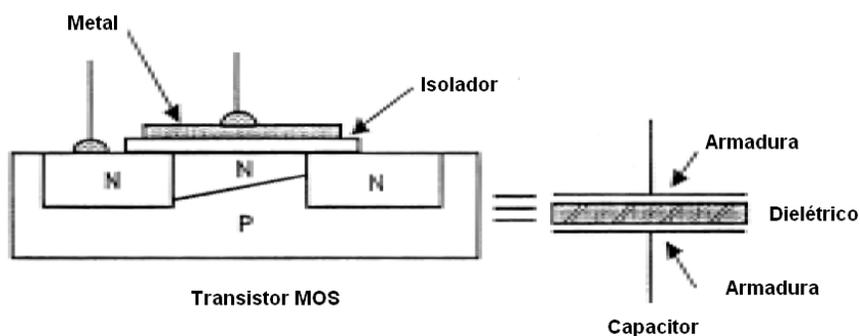


Figura 91 – Um transistor MOS se comporta como um capacitor

Isso significa que, ao aplicarmos um sinal de controle a uma função deste tipo, a tensão não sobe imediatamente até o valor desejado, mas precisa de certo tempo, tempo necessário para carregar o “capacitor” representado pelo eletrodo de comporta.

Se bem que o eletrodo tenha dimensões extremamente pequenas, se levarmos em conta as impedâncias envolvidas no processo de carga e também a própria disponibilidade de corrente dos circuitos excitadores, o tempo envolvido no processo não é desprezível e certo atraso na propagação do sinal ocorre.

O atraso nada mais é do que a diferença de tempo entre o instante em que aplicamos o sinal na entrada e o instante em que obtemos um sinal na saída.

Nos circuitos integrados CMOS típicos, como os usados nas aplicações digitais, para um inversor como o tomado como exemplo, este atraso é da ordem de 3 nanossegundos (3 ns).

Isso pode parecer pouco nas aplicações comuns, mas se um sinal tiver de passar por centenas de portas antes de chegar a certo ponto em que ele seja necessário, a soma dos atrasos pode causar diversos problemas de funcionamento, se não for prevista.

Veja, entretanto ainda, que a carga de um capacitor num circuito de tempo, como o mostrado na figura 92 até um determinado nível de tensão depende também da tensão de alimentação.

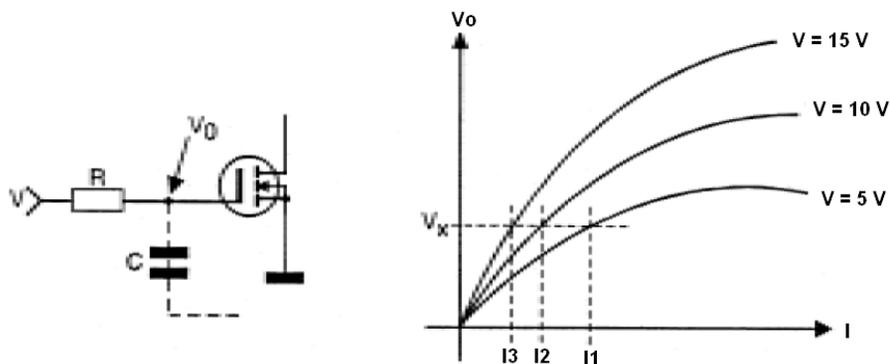


Figura 92- Carga do capacitor em função da tensão

Assim, com mais tensão a carga é mais rápida e isso nos leva a uma característica muito importante dos circuitos CMOS digitais que deve ser levada em conta em qualquer aplicação: com maior tensão de alimentação os circuitos integrados CMOS são mais rápidos.

Desta forma, enquanto que nos manuais de circuitos integrados TTL encontramos uma velocidade máxima única de operação para cada tipo (mesmo porque sua tensão de alimentação é fixa de 5 volts), nos manuais CMOS encontramos as velocidades associadas às tensões de alimentação (já que os circuitos integrados CMOS podem ser alimentados por uma ampla faixa de tensões).

Um exemplo de disso pode ser observado nas características de um circuito integrado CMOS formado por seis inversores (hex inverters) onde temos as seguintes frequências máximas de operação:

4049 - Seis inversores

O circuito integrado CMOS 4049 é formado por seis inversores, com a pinagem do invólucro DIL de 14 pinos mostrada na figura 93.

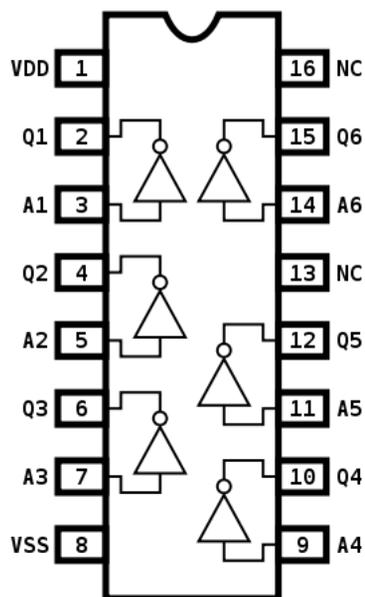


Figura 93 – O circuito integrado 4049

Frequência máxima de operação:

Com Vdd = 5 V - 1,66 MHz (típico)

Vdd = 10 V - 4,00 MHz (típico)

Vdd = 15 V - 5,00 MHz (típico)

Veja então que o circuito é muito mais rápido quando o alimentamos com uma tensão de 15 V do que quando o alimentamos com uma tensão de apenas 5 volts. Este fato é muito importante quando, por exemplo, devemos elaborar um oscilador com circuito integrado CMOS que opere no seu limite de velocidade ou quando vamos utilizar circuitos CMOS com outros de tecnologias mais rápidas.

Velocidade máxima

A velocidade máxima de operação de um circuito lógico CMOS depende da tensão de alimentação.

Capacitância de entrada

Esta capacitância dos transistores deste tipo não afeta apenas sua velocidade. Para polarizar o MOSFET com um sinal, precisamos "carregar" o capacitor de entrada, e isso não só leva tempo (um retardo), como também consome energia. Assim, a corrente que circula na entrada do componente, que deveria não existir, pois ele é controlado por tensão, serve para carregar este capacitor parasita. Uma forma de aumentar a velocidade destes componentes é justamente diminuindo esta capacitância.

Na figura 94 temos um gráfico que relaciona a velocidade de comutação com a corrente consumida.

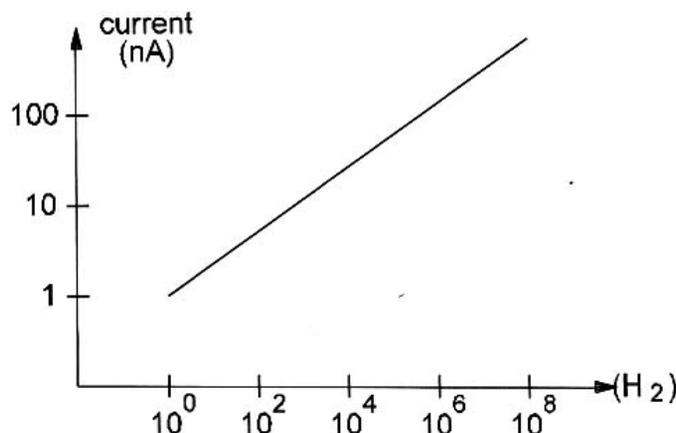


Figura 94 – Corrente consumida x velocidade de comutação

Na verdade, tecnologias especiais têm conseguido reduzir a capacitância de entrada dos circuitos CMOS a valores muito baixos.

Hoje são disponíveis famílias de integrados CMOS que podem operar com baixas tensões e com velocidades que se aproximam daquelas obtidas com algumas famílias TTL.

A grande vantagem está na manutenção do baixo consumo, mesmo em velocidades elevadas de operação.

4.4 – Famílias e Subfamílias CMOS

As famílias CMOS Standard começam com o 4000. Da mesma forma que no caso das famílias TTL temos também sufixos que indicam as subfamílias.

Também é importante notar que os números 4000 indicativos do tipo são acompanhados de um prefixo (conjuntos de letras) que indicam o fabricante, por exemplo, CD4001, etc.

Da mesma forma que no caso da família dos circuitos integrados TTL, também nos CMOS foram criadas sub-famílias com características especiais, algumas até com a mesma pinagem dos equivalentes TTL, seguindo, portanto a mesma nomenclatura.

Estas famílias, criadas a partir de 1972, visavam a substituição direta dos circuitos TTL por CMOS equivalentes, sem a necessidade de se modificar um layout de placa, pois a pinagem era a mesma.

Temos então as seguintes subfamílias CMOS com nomenclatura TTL, com suas características:

- **Standard 74C00** – esta é a família normal CMOS com nomenclatura TTL, mas que se encontra atualmente obsoleta. Observe a presença do “C” entre o 74 e o número do componente, para indicar que se trata de CMOS.

- **High Speed 74HC00** (High Speed significa alta velocidade) – Esta subfamília apareceu em 1980. Com os dispositivos desta família temos a mesma velocidade dos dispositivos TTL standard, mas com as características de consumo dos dispositivos CMOS.

- **High Speed 74HCT00** (alta velocidade) – esta subfamília se caracteriza pelo fato dos dispositivos terem entradas compatíveis com as saídas TTL.

- **Advanced High Speed 74AC00** (alta velocidade avançada) – com tempos de propagação típicos de apenas 5 ns

- **Advanced High Speed 74ACT00** (alta velocidade avançada) – com entradas compatíveis com TTL e tempos de propagação de 7 ns.

A tabela abaixo dá as principais características destas subfamílias quando comparadas com as CMOS convencionais da série 4000.

	Faixa de tensões	Corrente quiescente-por porta	Tempo de propagação por porta	Frequência máxima de operação	Fan-out para entradas TTL LS
4000B	3 – 15 V	0,01 μ A	125 ns (5V) 50 ns (10 V) 40 ns (15 V)	2 MHz (5 V) 5 MHz (10 V) 9 MHz (15 V)	1
4000UB	3 – 15 V	0,01 μ A	90 ns (5 V) 50 ns (10 V) 40 ns (15 V)	3 MHz (5 V) 5 MHz (10 V) 8 MHz (15 V)	1
74HC00	2 – 6 V	0,02 μ A	8 ns	40 MHz	10
74HCT00	4,5 – 5,5 V	0,02 μ A	10 ns	-	10
74AC00	2 – 6 V	0,02 μ A	6 ns	100 MHz	10
74ACT	4,5 – 5,5 V	0,02 μ A	7 ns	-	60
TTL LS	4,75 – 5,25 V	0,5 mA	9 ns	40 MHz	20

4.5 – Sensibilidade ao manuseio

A presença de uma finíssima camada de óxido, isolando a comporta do substrato, extremamente sensível a descargas elétricas, torna os dispositivos que usam transistores MOS muito delicados.

De fato, a própria carga elétrica acumulada em ferramentas ou no nosso corpo, quando caminhamos num tapete num dia seco, ou ainda atritamos objetos em nossa roupa, pode ser suficiente para danificar de modo irreversível dispositivos MOS. Para que o leitor tenha uma idéia, caminhando num carpete num dia seco, seu corpo pode acumular uma carga estática que atinge potenciais de até mais de 10 000 volts.

Se você tocar numa torneira ou uma maçaneta de porta a descarga de seu corpo neste percurso de terra pode lhe causar um forte choque.

Se, da mesma forma, você tocar num terminal de um dispositivo CMOS, a carga de seu corpo que escoar por este dispositivo pode facil-

Cargas Estáticas

As cargas elétricas acumuladas nos corpos, principalmente nos dias secos, são o principal inimigo dos componentes eletrônicos modernos. Cuidados com o manuseio, armazenamento e transporte de componentes são fundamentais para manter sua integridade. Todo profissional que trabalha com componentes eletrônicos deve saber como tratar estes delicados componentes, evitando que queimem por descargas estáticas. Veja o volume 1 desta série, Eletrônica Básica, para mais informações sobre ESD e cargas estáticas.



Pulseiras anti estáticas

mente destruir a finíssima camada de óxido que separa a comporta do substrato e, com isso, o componente estará inutilizado.

Em outras palavras, os dispositivos que usam transistores CMOS são extremamente sensíveis à descargas estáticas, conforme mostra a figura 95.

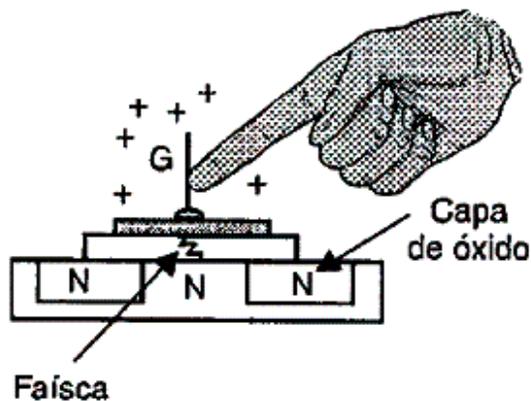


Figura 95 – Como cargas estáticas destroem componentes CMOS

Assim, a primeira preocupação no uso e manuseio destes componentes é evitar que, de qualquer modo, apareçam as tensões perigosas capazes de causar danos entre os terminais dos componentes.

Para os transistores CMOS, e outros componentes que usam terminais axiais, existe a possibilidade de dotá-los de um pequeno anel de metal que curto-circuita seus terminais, conforme mostra a figura 96, e que somente é retirado depois que o componente é soldado na placa de circuito impresso.

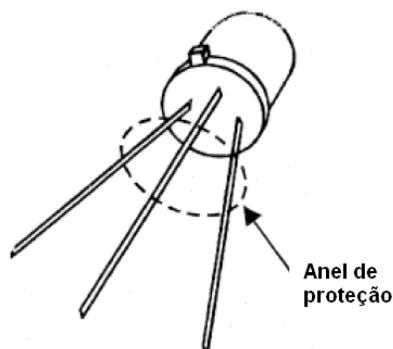


Figura 96 – Protegendo um transistor (FET ou MOSFET) sensível contra descargas estáticas

Este anel coloca em curto os terminais do componente, mantendo-os todos no mesmo potencial, evitando assim que apareçam as tensões capazes de danificar a pastilha sensível interna.

Para os circuitos integrados com invólucros DIL, e de outros tipos, existem diversas formas de se fazer seu transporte sem o perigo de cargas estáticas acumuladas em objetos possam lhe causar danos.

Uma delas consiste no uso de uma esponja condutora na qual

seus terminais são enfiados e assim mantidos em curto, conforme mostra a figura 97.

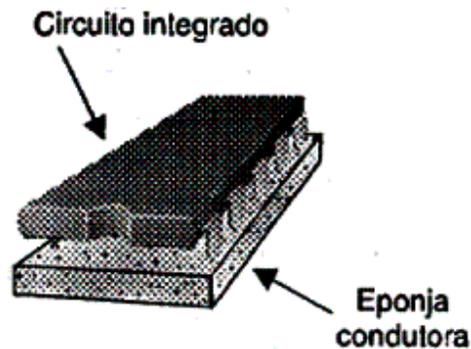


Figura 97 – Usando uma esponja condutora

Os circuitos integrados CMOS devem ser mantidos nestas esponjas até o momento de serem usados, sob pena de que algum toque acidental com o dedo carregado de estática provoque danos.

Outra possibilidade consiste em se transportar os circuitos integrados CMOS em embalagens de plásticos anti-estáticos conforme mostra a figura 98.

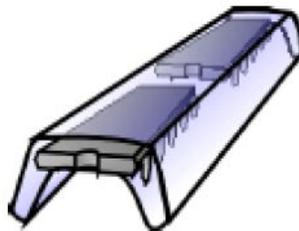


Figura 98 – Embalagem anti-estática para circuitos integrados

De qualquer forma, a regra geral é: NUNCA toque com os dedos nos terminais de componentes CMOS, sejam eles circuitos integrados ou transistores.

O laboratório ou oficina em que se trabalha com circuitos integrados CMOS deve tomar precauções especiais para que em nenhum ponto ocorram acúmulos de cargas estáticas. As bancadas de trabalhos com componentes desta tecnologia devem ter partes metálicas aterradas e os próprios profissionais devem usar recursos que permitam descarregar cargas de seu corpo.

Em empresas de trabalhos com circuitos CMOS é comum o uso de pulseiras de metálicas pelos profissionais, sendo estas pulseiras ligadas a um fio terra. (Veja o volume 2 – Eletrônica Analógica, lição que trata dos FETs).

Para o leitor, que apenas eventualmente se vê diante de tais componentes, é necessário apenas lembrar que não se deve tocar nos terminais dos componentes e com isso já se tem uma boa garantia para sua integridade.

Memórias RAM

Lembramos que os pentes de memória de computadores, microprocessadores usados em muitas máquinas industriais e equipamentos de telecomunicações, DSPs e outros usam estes transistores e, portanto, não devem ter os seus terminais diretamente tocados.



Um pente de memória ou cartão de memória RAM

Outro ponto importante que deve ser levado em conta é que se recomenda não deixar nenhuma entrada de um circuito integrado CMOS desligada, principalmente nas aplicações mais sensíveis.

O que ocorre é que a sensibilidade destas entradas é suficientemente alta para que tensões induzidas no próprio circuito sejam captadas levando os dois transistores a um estado intermediário entre o corte e a saturação ou ainda entrem em oscilação na frequência do sinal captado.

Isso, além de elevar o consumo do circuito integrado, pode causar instabilidades que afetam o funcionamento geral do circuito.

Uma regra prática consiste em se levar as entradas das funções não usadas num integrado a níveis definidos de tensão, ou seja, ligar ao V_{dd} ou ainda ao ponto de 0 V.

O nível que deve ser estabelecido nas entradas não usadas depende da função considerada. Deve-se tomar cuidado para que, em se fazendo isso, as saídas não usadas não sejam ligadas a níveis opostos ao estabelecidos pela ligação. Recomenda-se, portanto, manter as saídas desligadas. Na figura 99 temos exemplos que dependem da porta.

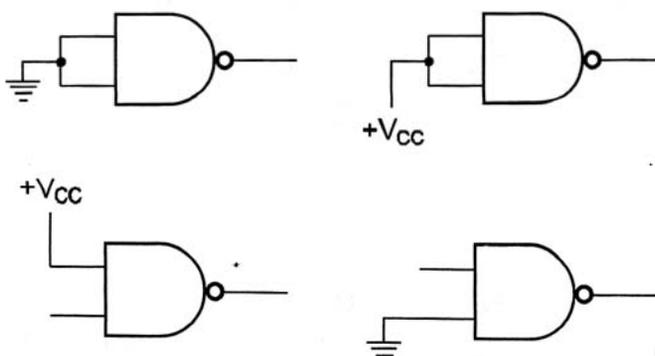


Figura 99 – Como conectar as entradas não usadas

Outro ponto importante a ser considerado é que tensões negativas abaixo de V_{ss} (0 V) não devem ser aplicadas aos circuitos de tecnologia MOS, pois elas podem danificá-los de modo irreversível.

4.6 - As Configurações CMOS

Elaborar funções lógicas a partir de transistores CMOS é muito mais simples do que fazer o mesmo com transistores bipolares.

De fato, se analisarmos o circuito de uma função lógica TTL, veremos que são usados, além dos transistores bipolares, diversos outros componentes passivos como resistores (em boa quantidade), diodos e até mesmo capacitores em alguns poucos casos.

Com o uso dos transistores CMOS, a elaboração de funções é muito mais simples, pois só precisamos de um tipo de componente praticamente: os transistores. Não encontramos na maioria das funções básicas senão transistores.

Podemos começar tomando como exemplo a figura 100 em que temos a configuração usada para uma porta NOR de 2 entradas CMOS na qual são usados quatro transistores.

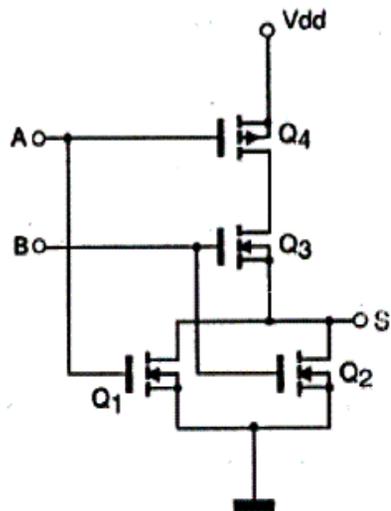


Figura 100 – Porta NOR CMOS

Observe a simplicidade desta função CMOS, quando comparada a funções equivalentes TTL. Com os circuitos CMOS precisamos apenas de transistores para ter a função desejada, enquanto que na equivalente TTL precisamos de muito mais do que transistores.

É importante observar que a não utilização de resistores traz enormes vantagens quando se pensa em termos de potência dissipada. Resistores convertem energia elétrica em calor e a sua ausência num circuito significa que não temos tanta geração de calor e, portanto, desperdício de energia.

Na figura 101 temos a configuração usada para uma porta NAND de duas entradas CMOS em que também usamos apenas 4 transistores.

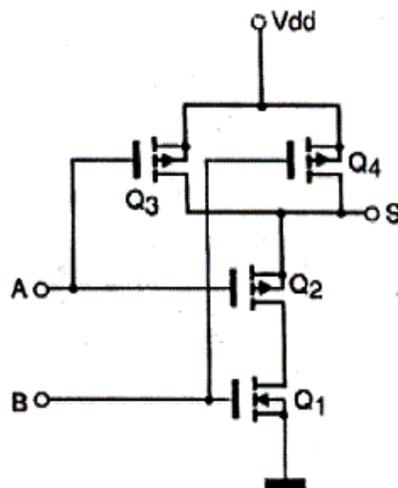


Figura 101 – Porta NAND CMOS

Neste circuito, quando ambas as entradas, ou uma delas estiver no nível baixo (0), um ou os dois transistores de canal P estarão em condução e a saída ficará no nível alto.

Quando as duas entradas estiverem no nível 1, entretanto, então os dois transistores de canal N vão conduzir ao mesmo tempo levando a saída para o nível baixo.

Para as outras funções lógicas temos configurações do mesmo tipo, mudando apenas a disposição e a quantidade de transistores usados.

Tomando estas duas funções como exemplo, achamos que o leitor já pode ter uma idéia de como elas são feitas como funcionam.

Mais informações sobre todas as funções lógicas normalmente podem ser encontradas nos manuais dos fabricantes.

Esses manuais normalmente podem ser “baixados” a partir dos sites dos fabricantes na Internet, ou ainda do site do autor deste livro.

Também podem ser obtidas informações com a digitação diretamente do tipo de componente em mecanismos de buscas como o Google,

4.7 -Especificações

A principal família de circuitos integrados CMOS é a 4000 onde a maioria dos componentes são designados por números que começam por 4, como 4001, 4011, 4017, 4096, etc. As exceções existem para os componentes mais avançados da linha, mas é fácil saber que eles são compatíveis pois normalmente são incluídos nos mesmos manuais dos demais.

Os circuitos integrados CMOS comuns funcionam com tensões de alimentação de 3 a 15 volts. Lembramos que existem séries CMOS mais antigas com o sufixo A em que a tensão de alimentação fica na faixa de 3 a 12 volts e algumas séries que vão até 18 V.

Também devemos observar que existem as subfamílias que já estudamos em itens anteriores.

De qualquer forma, em dúvida sobre qualquer característica de um circuito integrado CMOS que tenha algum sufixo que possa indicar variações nas especificações normais, é sempre bom consultar seu manual ou ainda a Internet.

A maioria dos fabricantes coloca databooks disponíveis com dados de todos os seus componentes na própria Internet, bastando digitar o tipo no seu “search” (procura).

O site do autor WWW.newtonbraga.com.br, disponibiliza um mecanismo de busca para datasheet em que basta digitar o tipo do componente, para se acessar o documento.

Da mesma forma que no caso dos circuitos integrados TTL, preciso saber interpretar algumas das principais especificações que são:

a) Tensão de saída - no nível lógico baixo (0) a tensão de saída se aproxima de 0 V sendo no máximo de 0,01 volt para os tipos comuns com alimentação na faixa de 5 a 10 V. No nível lógico alto, a tensão de saída é praticamente a tensão de alimentação Vdd ou no máximo 0,01 volt menor.

b) Corrente de saída - diferentemente dos circuitos integrados TTL, em que temos uma capacidade maior de drenar corrente na saída do que de fornecer, para os circuitos integrados CMOS a capacidade de drenar e de fornecer corrente de saída é praticamente a mesma. Assim, para uma alimentação de 5 volts as saídas podem fornecer (quando no nível alto) ou drenar (quando no nível baixo) uma corrente de até 0,88 mA, e essa corrente sobe para 2,25 mA quando a alimentação passa para 10 V. Para uma tensão de 15 V a corrente máxima drenada ou fornecida tem um valor típico de 8,8 mA. Estas correntes, conforme mostra a figura 102 são designadas por IOL e IOH nas folhas de especificações dos circuitos integrados CMOS. Lembramos que existem algumas funções especiais em que, além dessas correntes terem valores diferentes dos indicados, também podem ser diferentes quando a saída está no nível alto ou no nível baixo. São algumas funções especiais são indicadas especificamente para excitar uma carga drenando ou fornecendo uma corrente.

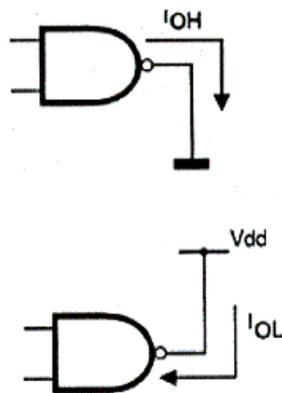


Figura 102 – As correntes de saída dos circuitos integrados CMOS

c) Corrente de fuga na entrada - se bem que a comporta esteja isolada do circuito dreno-fonte, com uma resistência que teoricamente seria infinita, na prática pode ocorrer uma pequena fuga. Esta, da ordem de 10 pA (1 picoampère = 0,000 000 000 001 ampère) para uma alimentação de 10 V deve ser considerada quando precisamos calcular a corrente de entrada de um circuito CMOS numa aplicação mais crítica.

d) Potência - os circuitos integrados CMOS consomem muito menos energia que os circuitos integrados TTL. Para os tipos comuns a corrente de alimentação, especificada por Idd, é normalmente da

ordem de 1 nA tipicamente com um máximo de 0,05 μ A para alimentação de 5 V o que corresponde a uma dissipação de 5 nW em média para alimentação de 5 V e 10 nW para alimentação de 10 V.

e) Velocidade - os tipos comuns CMOS são muito mais lentos que os TTL, mas famílias especiais estão aparecendo com velocidades cada vez maiores e em muitos casos estas se aproximam dos mais rápidos TTLs. As frequências máximas, conforme já explicamos dependem das tensões de alimentação e das funções, já que maior número de componentes para atravessar significa um atraso maior do sinal. Assim, nos manuais encontramos a especificação de velocidade dada tanto em termos de frequência quanto em termos de atraso do sinal. Para o caso do atraso do sinal, observamos que ele pode estar especificado tanto para uma transição do nível alto para o nível baixo como vive-versa e em alguns circuitos ou tensões de alimentação podem ocorrer diferenças. De qualquer forma, as famílias normais de circuitos integrados CMOS possuem velocidades que no máximo alcançam 10 MHz diferentemente de algumas subfamílias TTL que podem alcançar mais de 100 MHz.

Todas essas características são importantes quando se realiza qualquer projeto que utilize circuitos integrados CMOS.

Também é preciso considerar que as faixas de tensões de entrada reconhecidas como nível e nível baixo dependem da tensão de alimentação, o que nos leva à curva de transferência mostrada na figura 103.

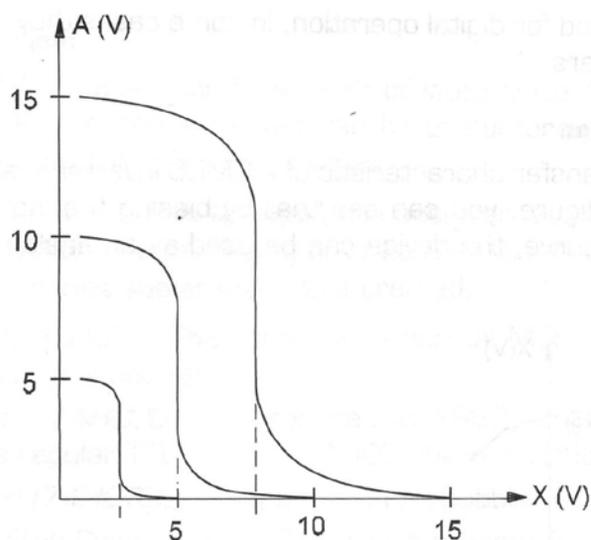


Figura 103 – Curva de transferência de um inversor CMOS com entrada A e saída X

Diferentemente dos circuitos integrados TTL, em que temos uma tensão fixa de alimentação, quando trabalhamos com circuitos

integrados CMOS é preciso estar atento ao que acontece com a velocidade, correntes de saída e níveis lógicos quando usamos diferentes tensões de alimentação.

Também é preciso observar que são muitos os fabricantes de circuitos integrados CMOS atualmente. Assim, um determinado componente de um fabricante pode ter características levemente diferentes do mesmo componente feito por outro fabricante.

Na maioria dos casos, as diferenças são pequenas demais para que ocorra alguma anormalidade de funcionamento num projeto, mas se na substituição de um componente por outro alguma coisa acontecer, o profissional deve atentar para esse detalhe.

Uniformidade

Uma garantia de que tudo vá bem num projeto é a uniformidade das características dos componentes usados. Assim, se possível, o projetista deve trabalhar com componentes de uma única família.

4.7.1 – Desacoplamento

As transições rápidas dos níveis de sinais que ocorrem num circuito integrado CMOS, como em qualquer outro, causam picos de corrente que podem se propagar pelo circuito.

O resultado desta propagação é uma instabilidade dos demais circuitos integrados do equipamento que podem ser levados indevidamente a uma mudança de estado.

Para se evitar este problema, o que se faz normalmente é desacoplar a alimentação do circuito integrado com a ajuda de um capacitor cerâmico, normalmente de 100 nF. Este capacitor, conforme mostra a figura 104, deve ser ligado entre o positivo e o negativo (GND) da alimentação.

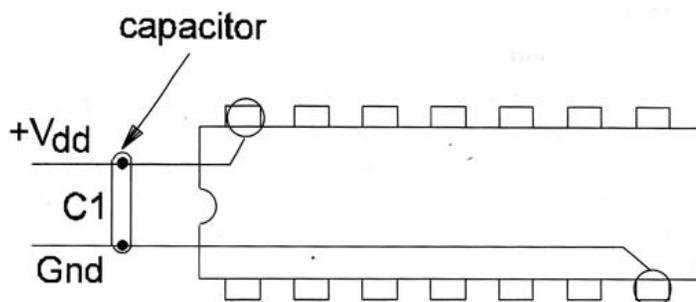


Figura 104 – Desacoplando a alimentação do circuito integrado CMOS

4.8 – Interfaceando

Interfacear é interligar dois circuitos de modo que um possa transferir sinais para o outro. Nas aplicações que envolvam circuitos eletrônicos digitais, o interfaceamento não é apenas um procedimento comum, mas necessário principalmente quando se deseja transmitir dados de um equipamento para outro.

No entanto, o problema principal que pode ocorrer num caso de interfaceamento é que os equipamentos interligados operem com tecnologias diferentes. Mesmo dentro de um equipamento podemos encontrar setores com tecnologias diferentes.

Como as duas principais tecnologias encontradas nos equipamentos digitais são a CMOS e TTL é importante que o leitor saiba como proceder quando tiver de interfaceá-las.

Assim, mesmo tendo uma faixa de tensões ampla e características diferentes dos circuitos integrados TTL, existe a possibilidade de se interfacear os circuitos CMOS com os TTL tanto para enviar sinais como para receber sinais.

Por este motivo, existem diversas possibilidades de interfaceamento entre circuitos digitais TTL e circuitos digitais CMOS que analisaremos a seguir.

O leitor deve estar atento para essas técnicas de interfaceamento, pois elas consistem num ponto bastante crítico dos projetos que envolvam as duas tecnologias e mesmo circuitos analógicos como os que fazem uso de amplificadores operacionais.

Analisemos os diversos casos:

a) A saída TTL deve excitar a entrada CMOS (alimentação de 5 V para os dois).

Se os dois circuitos operarem com uma tensão de alimentação de 5 Volts não há problema e a interligação pode ser direta.

Como as entradas CMOS têm uma impedância muito alta (não exigindo praticamente corrente alguma) da saída TTL não existe perigo do circuito CMOS “carregar” a saída TTL.

No entanto, existe um problema a ser considerado: as entradas CMOS só reconhecem como nível 1 uma tensão de pelo menos 3,5 volts enquanto que no nível alto, a tensão mínima que o TTL pode fornecer nestas condições é de 3,3 volts.

Isso significa que é preciso assegurar que a entrada CMOS reconheça o nível alto TTL o que é conseguido com a adição de um resistor externo de pull-up, conforme mostra a figura 105.

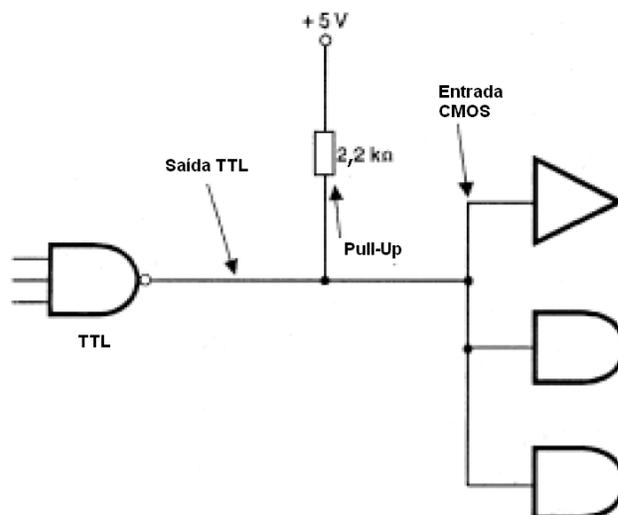


Figura 105 – Interfaceando TTL com CMOS

Este resistor de 2,2 k ohms é ligado ao positivo da alimentação de 5 volts.

b) TTL para CMOS com Tensões Diferentes

Se o circuito CMOS a ser excitado por um TTL for alimentado com tensão maior que 5 volts, por exemplo, 12 volts, deve ser usado um circuito intermediário de casamento de características.

Na figura 106 mostramos o circuito a ser usado.

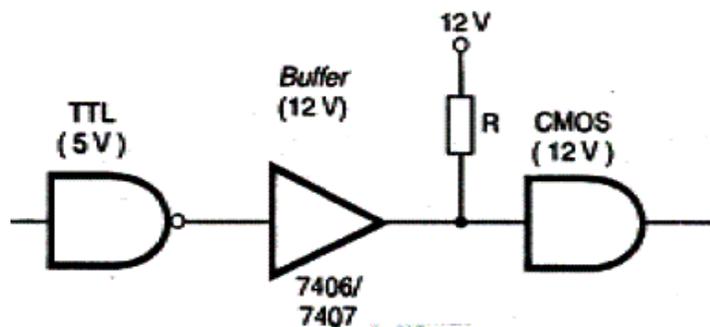


Figura 106- Interfaceando TTL com CMOS de tensão diferente

Lembramos que o transistor, na configuração de emissor comum, atua como um inversor de nível lógico. Assim, o nível lógico aplicado na entrada do circuito integrado CMOS é o inverso da saída TTL.

c) TTL (coletor aberto para CMOS) - tensões iguais.

Devemos usar um transistor driver de comutação, como o 2N2222 e dois resistores, sendo o de 10k, pull-up, conforme mostra a figura 107.

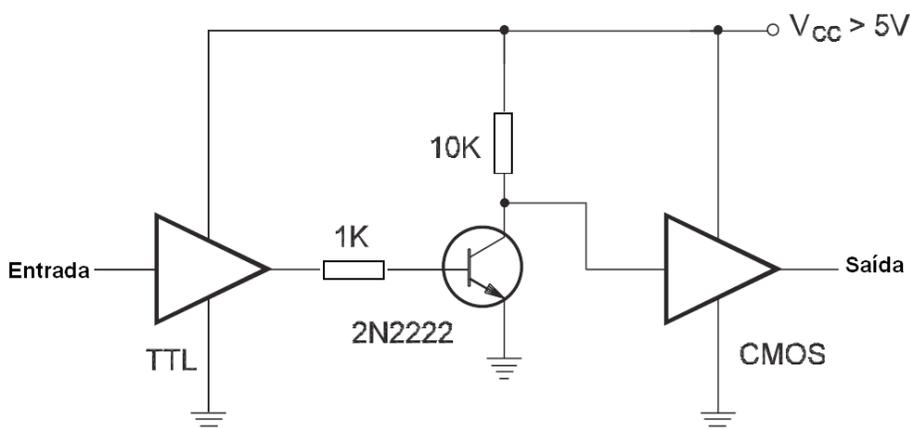


Figura 107 – TTL coletor aberto para CMOS

A tensão de alimentação do CMOS deve ser maior do que 5 V.

d) TTL coletor aberto para CMOS com tensões diferentes

Nesse caso, basta colocar um resistor de pull-up de 10 k ohms no circuito de saída do TTL com coletor aberto, conforme mostra a figura 108.

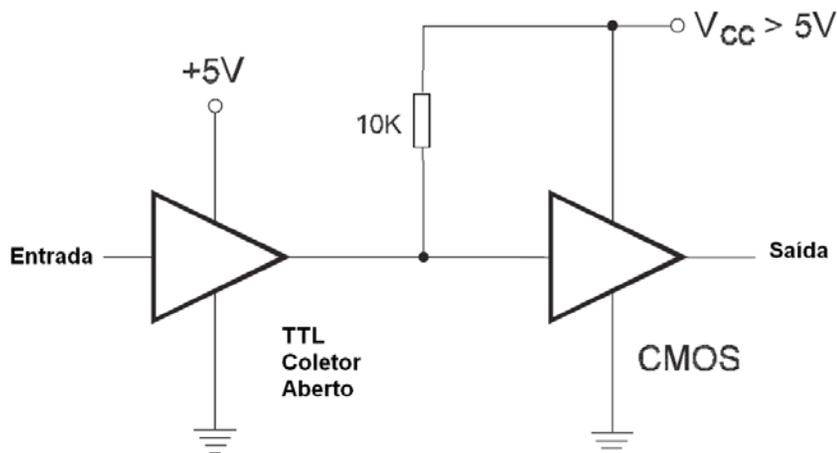


Figura 108 – TTL coletor aberto para CMOS com 5 V

e) CMOS excitando uma entrada TTL - alimentação de 5 V para os dois.

Neste caso, devemos considerar que uma saída CMOS no nível baixo pode drenar uma corrente de aproximadamente 0,5 mA e no estado alto, a mesma intensidade.

Para polarizar a entrada TTL usamos um resistor de 1 k à terra conforme mostra a figura 109.

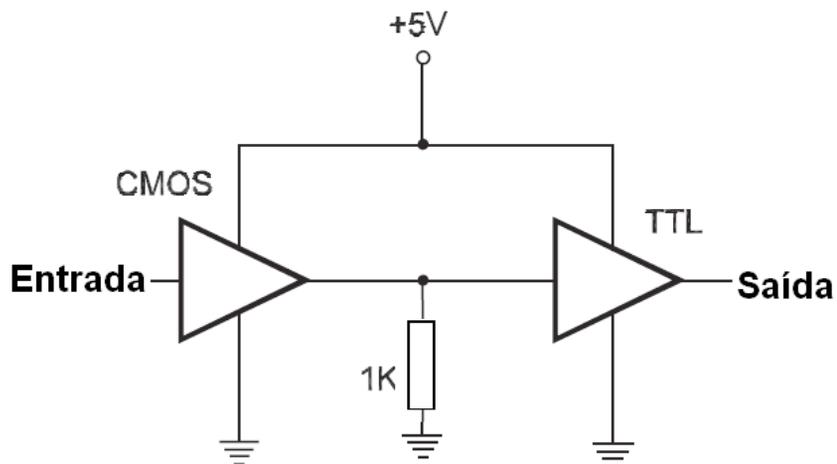


Figura 109 – CMOS excitando TTL, ambos com 5 V

f) CMOS para TTL - tensões diferentes

Neste caso, é preciso usar uma etapa de adaptação intermediária com um transistor que funciona como inversor. Qualquer transistor de comutação pode ser usado com essa finalidade, conforme mostra a figura 110.

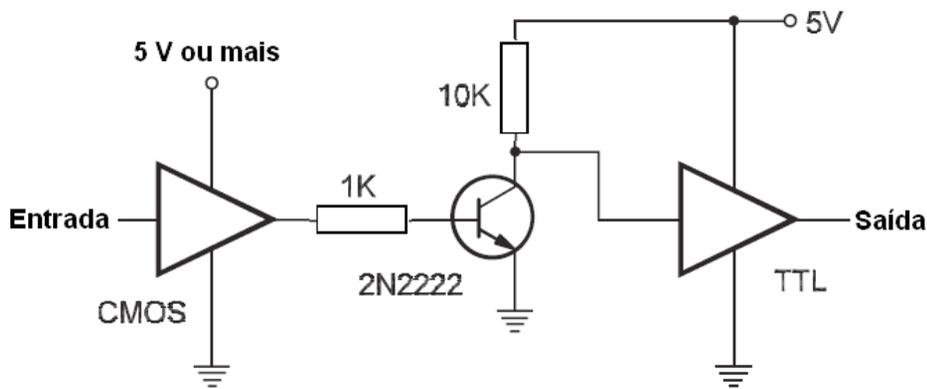


Figura 110 – CMOS para TTL com tensões diferentes

g) CMOS para TTL com um Buffer - Tensões diferentes.

Na figura 111 mostramos o modo de se fazer o interfaceamento de uma saída CMOS com alimentação maior do que 5 V com uma entrada TTL. Para essa finalidade usamos um “buffer” CMOS alimentado por 5 V. O buffer recomendado é o 4049.

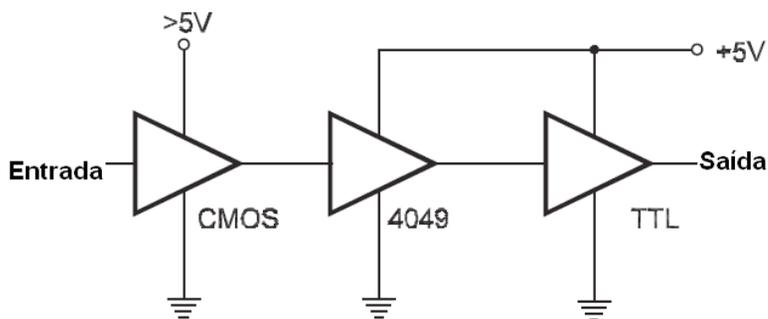


Figura 111 – CMOS para TTL com buffer

h) Amplificador Operacional Para CMOS - mesma tensão

Na figura 112 mostramos o modo de se acoplar uma saída de amplificador operacional à entrada de uma função CMOS. Observe que, nesse circuito, o amplificador operacional não está usando fonte simétrica.

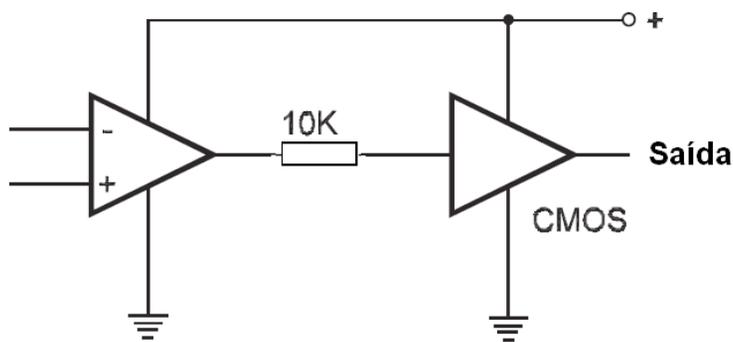


Figura 112 – Operacional para CMOS

i) TTL a CMOS com Acoplador Óptico

Quando são usadas tecnologias diferentes num projeto, como CMOS e TTL, pode ser importante garantir a passagem do sinal de uma etapa a outra, mas com o mais completo isolamento. Isso pode ser conseguido com facilidade através de um acoplador óptico.

Na figura 113 mostramos como isso pode ser feito, para transferência de sinais TTL para uma entrada CMOS de qualquer tensão de alimentação.

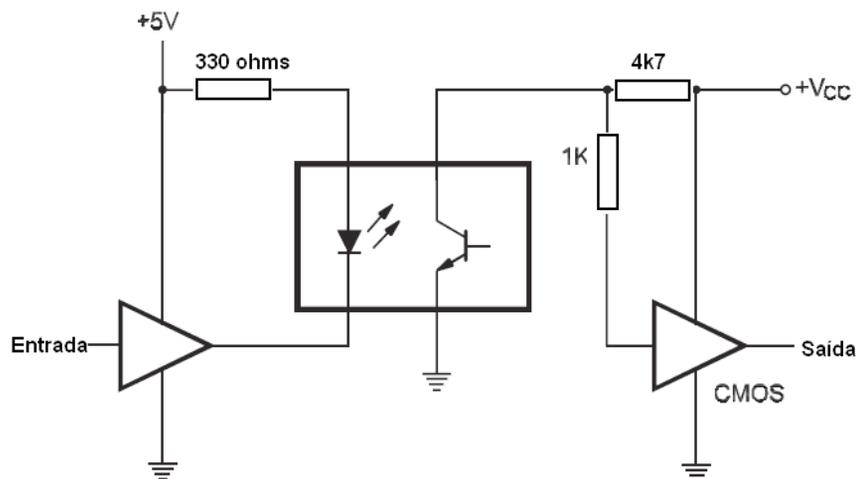


Figura 113 – Usando acoplador óptico

4.9 – Fontes de alimentação

Os circuitos integrados TTL precisam de uma tensão contínua na faixa de 4,5 a 5,5 V para poderem funcionar e são bastante sensíveis a flutuações da tensão de alimentação. Basta que a tensão saia dessa faixa que, de um lado teremos o perigo do não funcionamento e do outro a queima do dispositivo.

Já os circuitos CMOS são muito menos sensíveis a isso, pois podem operar numa faixa mais larga de tensões, conforme vimos. Os circuitos integrados CMOS podem operar satisfatoriamente com tensões de 3 a 15 V.

Isto facilita bastante o projeto das fontes e até permite a alimentação direta a partir de pilhas ou baterias, sem a necessidade de circuitos reguladores críticos e estáveis, que podem significar não só encarecimento de um projeto, como também contribuir com uma boa parcela do consumo de energia.

Desta forma, as fontes de alimentação dos circuitos que usam tecnologia CMOS podem ser muito mais simples não sendo raros os casos em que os circuitos reguladores de tensão são omitidos, ou mesmo tipos simplificados escolhidos.

Veja que, o fato dos circuitos integrados CMOS funcionarem perfeitamente com tensões comuns como 3, 6, 9 e 12 volts, que são facilmente obtidas de pilhas e baterias, os torna ideal para aplicações em que este tipo de fonte é usada.

O baixo consumo, por outro lado, é outro fator que deve ser considerado, já que pilhas comuns, mesmo as de menor capacidade podem alimentar com facilidade projetos que usem muitos circuitos integrados CMOS por longos intervalos de tempo.

Na verdade, o consumo de energia dos circuitos CMOS é tão baixo que em alguns casos, até mesmo fontes alternativas de energia de baixa potência, como células solares, podem ser usadas.

Damos a seguir uma série de circuitos de fontes, que também servem para TTL, se programadas para 5 V, e que podem ter diversas tensões, todas baseadas nos CIs da série 7800 de 1 A.

A série de circuitos integrados 78XX onde o XX é substituído por um número que indica a tensão de saída, consiste em reguladores de tensão positiva com corrente de até 1 ampère de saída e que são apresentados em invólucro TO-220 conforme mostra a figura 114.

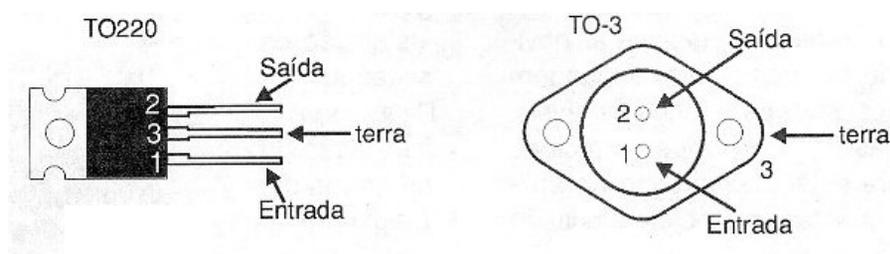


Figura 114 – Os circuitos integrados 78xx

Diversos são os fabricantes que possuem os circuitos integrados desta série em sua linha de produtos e as tensões de saída podem variar sensivelmente de um para outros. No entanto, os valores básicos para estas tensões, que são dados pelos dois últimos algarismos do tipo do componente são:

7805 = 5 volts	7812 = 12 volts
7806 = 6 volts	7815 = 15 volts
7808 = 8 volts	7818 = 18 volts
7885 = 8,5 volts	7824 = 24 volts

A tensão máxima de entrada para os tipos de 5 a 18 volts é de 35 volts. Para o tipo de 24 volts a tensão de entrada máxima é de 40 volts.

De qualquer modo, para um bom funcionamento a tensão de entrada deve ser no mínimo 2 volts mais alta que a tensão que se deseja na saída.

Os circuitos integrados da série 78XX possuem proteção interna contra curto-circuito na saída e não necessitam de qualquer componente externo.

Damos a seguir as principais características do 7805 que serve de base para avaliação dos demais tipos da série:

Série 79xx

Existe uma série equivalente de reguladores de 1 A formada por reguladores negativos de corrente, ou seja, são ligados na linha negativa da alimentação. Cuidado nesta série o terminal do meio não é o terra!

7805 - Características

	min.	tip.	max.	
Tensão de saída	4,8	5,0	5,2	volts
Regulagem de linha	-	3	50	mV
Regulagem de carga	-	15	50	mV
Corrente quiescente	-	4,2	6,0	mA
Rejeição de ripple	60	70	-	dB
Resistência de saída	-	17	-	mOhms

Observe que o radiador de calor deve ser dimensionado em função da diferença que existe entre a tensão de entrada e a tensão de saída, já que, quanto maior ela for mais calor o componente deve dissipar.

Damos, a seguir, diversos circuitos práticos envolvendo os circuitos integrados da série 78XX. O XX depois do 78 indica que o mesmo circuito pode ser usado para qualquer tensão na faixa de 5 a 18 volts com a escolha do componente apropriado.

Na figura 115 temos a aplicação imediata num regulador positivo de 1 ampère para tensões de 5 a 24 volts com corrente de saída de até 1 ampère.

Capacitores

Os capacitores de entrada e de saída deste circuito variam conforme o fabricante, existindo séries em que na saída são utilizados capacitores de valores bastante baixos e outras em que os valores devem ser maiores.

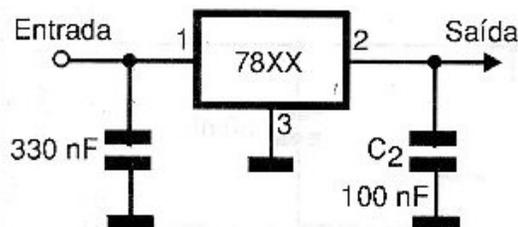


Figura 115 – Circuito básico com o regulador 78xx

O capacitor de 330 nF desacopla a entrada do estabilizador enquanto que o de 100 nF, que deve ser cerâmico de boa qualidade, tem por finalidade evitar oscilações em altas frequências e também desacopla a saída.

Utilizando um amplificador operacional 741 podemos tornar variável a tensão de saída de um regulador 7805, obtendo com isso uma fonte de 7 a 30 volts. A tensão de entrada deve ser de 35 volts e o potenciômetro de 10 k ohms deve ser linear. Os capacitores de desacoplamento devem ser cerâmicos de boa qualidade.

Este circuito é mostrado na figura 116, e na sua entrada devemos aplicar uma tensão contínua não regulada, porém com boa filtragem.

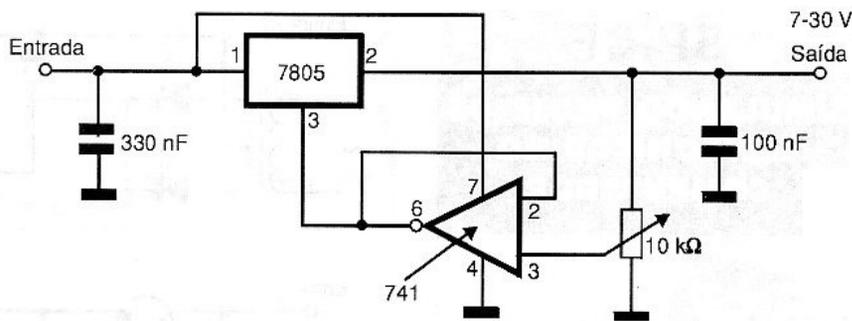


Figura 116 – Fonte de 7 a 30 V com operacional 741

Lembre-se de que a tensão máxima dos circuitos integrados CMOS é de 15 V.

Para se obter corrente maior do que 1 ampère, podemos usar um booster, conforme o mostrado na figura 117. O transistor pode ser substituído por equivalentes com correntes de coletor na faixa de 5 a 10 ampères para se obter uma fonte de 2 a 5 ampères de corrente de saída.

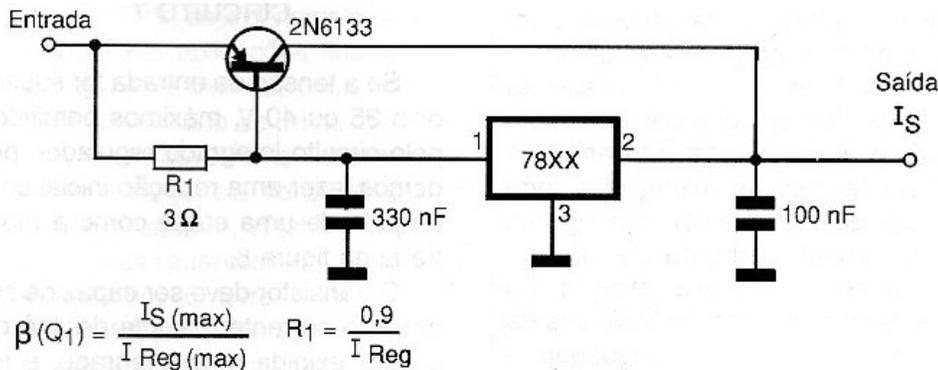


Figura 117 – Para obter correntes maiores do que 1 A pode ser usado este circuito.

As fórmulas que permitem dimensionar os diversos elementos do circuito são dadas junto ao diagrama.

Temos na figura 118 basicamente a mesma configuração do circuito anterior mas com o acréscimo de um sistema de proteção contra curto-circuitos na saída.

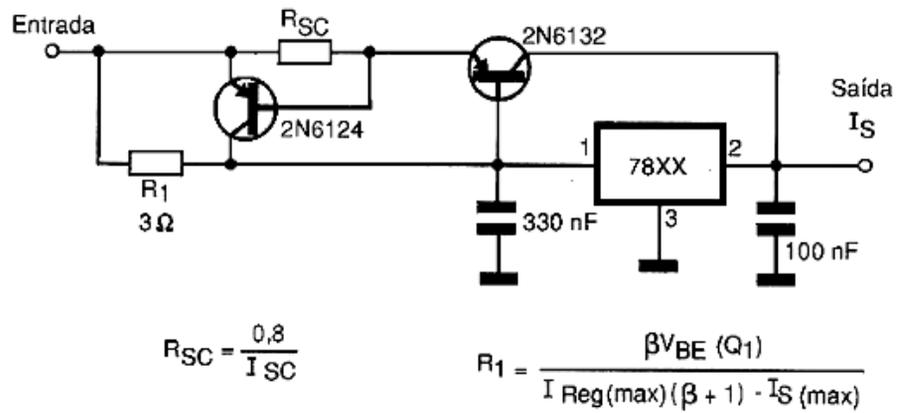


Figura 118 – Fonte de alta corrente com proteção contra curto-circuitos

O transistor Q1 deve conduzir quando a queda de tensão em RSC e for maior do que 0,6 volts ocorrendo então o corte da polarização de base do transistor de potência. Os valores dos componentes são dados pelas fórmulas junto ao próprio diagrama.

O circuito integrado fixará o valor da tensão de saída, observando-se que existe uma queda de tensão da ordem de 0,6 volts no transistor e que deve ser considerada.

O circuito mostrado na figura 119 consiste num regulador positivo que funciona aqui como regulador negativo. Temos então uma fonte de tensão negativa.

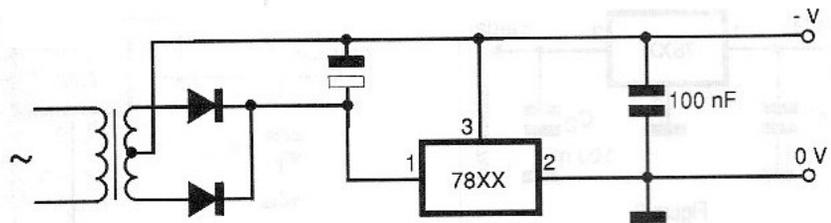


Figura 119 – Fonte com regulagem negativa usando o 78xx

O capacitor de filtro deve ser dimensionado de acordo com a tensão e a corrente de saída assim como o nível de ripple exigida para a aplicação.

Os diodos e transformador devem também ser dimensionados de modo a fornecer na entrada do circuito integrado, pelo menos 5 volts a mais do que o valor da tensão exigida na saída.

Se a tensão de entrada for superior a 35 ou 40 volts, máximos admitidos pelo circuito integrado regulador podemos fazer uma redução inicial com a ajuda de uma etapa como a mostrada na figura 120.

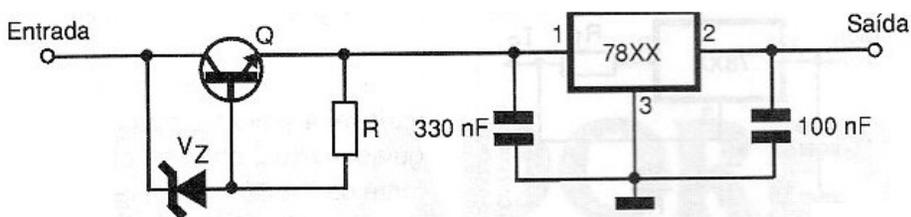


Figura 120 – Circuito para tensões de entrada elevadas

O transistor deve ser capaz de suportar a corrente máxima de 1 ampère de coletor exigida pelo integrado, e ter uma especificação de tensão máxima entre coletor e emissor de acordo com a queda de tensão que deve proporcionar no circuito.

O diodo zener, por outro lado, precisa ter uma potência de acordo com a exigida pelo circuito. O resistor R estabiliza a corrente do diodo zener de modo que, no mínimo, não ocorram variações da tensão aplicada ao integrado.

O processo mais simples de se obter uma queda de tensão de entrada para um regulador da série 78XX quando a corrente de carga deve ser constante é o mostrado na figura 121.

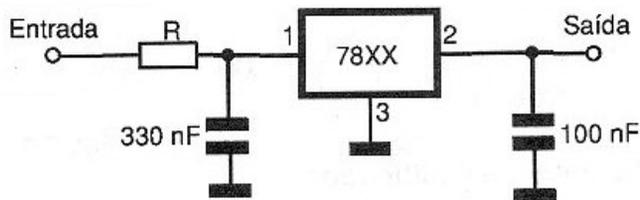


Figura 121 – Obtendo uma queda de tensão na entrada com a ajuda de um resistor

O resistor é calculado de modo a fornecer a queda de tensão exigida conforme os máximos admitidos pelo integrado. Podemos calculá-lo por:

$$R = (V_i - V_x) / I$$

Onde:

V_i é a tensão de entrada do circuito (volt)

V_x é a tensão de entrada do circuito integrado (máximo de 40 V para os de 24 V e 35 V para os de 5 a 18 V)

I é a intensidade da corrente de carga (A)

Veja que, desprezamos a corrente exigida pelo próprio circuito integrado regulador de tensão, já que ela é bastante baixa.

A dissipação do resistor será dada por:

$$P = (V_i - V_x) \times I$$

Onde as grandezas são as mesmas da fórmula anterior exceto:

P é a potência que deve ser expressa em watts.

Termos em Inglês

Conforme temos salientados nas lições anteriores, muitos termos de eletrônica digital são mantidos na forma original na literatura técnica em português. No entanto, é interessante saber o significado deles.

- Standard – normal ou padrão
- Advanced – avançado
- High Power – alta potência
- Low Power – baixa potência
- High speed – alta velocidade
- Delay – retardo
- Leak – fuga
- Rise – subir
- Fall – descer, cair

Termos para pesquisa

- Tecnologia CMOS
- Circuitos CMOS
- MOSFETs
- Lógica CMOS
- ESD
- Desacoplamento

QUESTIONÁRIO

1. O elemento de controle do sinal de um transistor de efeito de campo é denominado:

- a) base
- b) dreno
- c) comporta
- d) canal

2. O elemento de controle de um MOS FET é separado do canal por que tipo de material?

- a) Uma junção PN
- b) Um substrato condutor
- c) Uma camada de material isolante
- d) Um terminal de cobre

3. Num inversor CMOS encontramos na etapa de saída:

- a) dois FETs de canal N
- b) dois FETs de canal P
- c) Um par de transistores bipolares
- d) Um FET de canal N e outro de canal P

4. A faixa de tensões de alimentação dos circuitos integrados CMOS tem valores entre:

- a) 4,5 e 5,5 V
- b) 3 e 15 V
- c) 0 e 6 V
- d) 5 e 18 V

5. O perigo maior do manuseio dos circuitos integrados CMOS se deve a:

- a) descargas estáticas
- b) aquecimento da pastilha semicondutora
- c) perigo de quebra dos terminais
- d) contaminação radioativa

6. O que se deve fazer com as entradas não usadas de um circuito integrados CMOS.

- a) devemos cortá-las
- b) devemos aterrâ-las
- c) devemos ligá-las a um nível lógico apropriado
- d) ligar a um resistor de 100 kohms



» Combinando Funções Lógicas (Lógica Combinacional)

Nos dois capítulos anteriores estudamos as famílias lógicas CMOS e TTL, analisando suas características elétricas principais e a maneira como os componentes são fabricados e, além disso, também vendo alguns circuitos típicos. Neste capítulo continuamos a estudar as funções lógicas, agora de uma forma mais completa. Devemos analisar o que acontece quando juntamos diversas funções lógicas, prevenindo que tipo de sinal teremos em suas saídas. Os circuitos complexos, como os usados nos computadores, robôs, equipamentos industriais e de telecomunicações, se aproveitam das operações complicadas que muitas portas lógicas podem realizar em conjunto. Assim, é de fundamental importância para nosso estudo saber analisar estas funções. Esse é justamente o assunto deste capítulo. Este capítulo conta com os seguintes itens:

Itens:

- 5.1 – As tabelas verdade
- 5.2 – Lógica combinacional
- 5.3 – Como projetar um circuito combinacional
- 5.4 – Simplificando e minimizando
- 5.5 – Diagramas da Karnaugh

Objetivos:

- Entender melhor como funcionam as tabelas-verdade
- Saber como analisar um circuito em que diversas funções são combinadas
- Como projetar um circuito que utilize diversas funções combinadas
- Entender como podemos simplificar um circuito com diversas funções
- Analisar circuito combinados usando os diagramas de Karnaugh

5.1 - As tabelas verdade

Os diversos sinais de entrada aplicados a uma função lógica, com todas as suas combinações possíveis, e a saída correspondente, podem ser colocados numa tabela. Já tivemos uma noção disso nas primeiras lições.

Nas colunas de entradas colocamos todas as combinações possíveis de níveis lógicos que as entradas podem assumir. Na coluna correspondente à saída colocamos os valores que esta saída assume em função dos níveis lógicos correspondentes na entrada.

Esta tabela, conforme já vimos, é chamada “tabela verdade” ou “truth table”, se adotarmos o termo em inglês. Estudamos, desta forma, que a tabela verdade para uma função AND de duas entradas será representada conforme mostra a figura 122.



Figura 122 – Representação da função AND (E)

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

Veja que nas colunas de entrada (A e B), para termos todas as combinações possíveis, fazemos o equivalente a uma contagem binária de 0 a 3, já que:

$$00 = 0$$

$$01 = 1$$

$$10 = 2$$

$$11 = 3$$

Para uma tabela verdade feita para uma porta AND de 3 entradas teremos:

A	B	C	S
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

Neste caso, as combinações de níveis lógicos na entrada, na realidade, correspondem à uma contagem binária de 0 a 7 já que:

000 = 0
001 = 1
010 = 2
011 = 3
100 = 4
101 = 5
110 = 6
111 = 7

Numeração Binária

Se o leitor tem ainda dificuldades em entender a numeração binária, será importante voltar às primeiras lições deste curso, antes de seguir em frente.

Observe então que, para uma função de 2 entradas, temos 4 combinações possíveis de sinais de entrada. Para uma função de 3 entradas, teremos 8 combinações e para uma função de 4 entradas, 16 combinações e assim por diante.

O conhecimento da contagem binária facilita bastante a elaboração de tabelas verdade quando todas as combinações possíveis de níveis lógicos em 2, 3 ou 4 entradas devam ser analisadas.

Assim, uma vez que o leitor conheça o comportamento das principais funções, sabendo o que ocorre na saída de cada uma, quando temos determinadas entradas e sabendo elaborar tabelas verdade, fica fácil combinar funções e saber o que acontece em suas saídas.

Na verdade, o ponto principal das informações técnicas dos circuitos lógicos digitais é a sua tabela verdade.

Também, devemos observar que, além das combinações dos níveis lógicos das entradas, também podemos encontrar colunas que levam em conta os níveis lógicos de entradas especiais como, por exemplo, uma entrada de habilitação (EN), uma entrada que leve o dispositivo a um terceiro estado ou ainda que cause uma transição do nível de sinal.

Nesse caso, como já vimos, independentemente do que ocorra nas entradas, a saída estará num estado de alta impedância. As tabelas verdade podem incluir este terceiro estado do dispositivo.

5.2 - Lógica Combinacional

Denominamos “lógica combinacional” a lógica que rege o funcionamento de diversas funções mais simples quando elas são combinadas, ou reunidas num circuito mais complexo.

Para entender como ela funciona, vamos partir de um exemplo simples de lógica combinacional usando tabelas verdade para saber, por exemplo, o que ocorre na sua saída, com o circuito da figura 123.

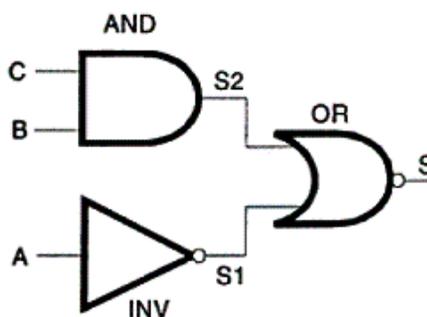


Figura 123 – Um circuito simples com três funções lógicas

Este circuito faz uso de uma porta AND, um inversor e uma porta OR. O resultado desta configuração é uma função combinacional que tem três entradas e uma saída.

Para elaborar a tabela verdade para este circuito, e assim determinarmos todas as saídas possíveis em função das entradas, devemos levar em conta que ele é formado por duas etapas.

Na primeira etapa temos a porta AND e o inversor, enquanto que na segunda etapa temos a porta OR. Isso significa que as saídas dos circuitos da primeira etapa, que chamaremos de S1 e S2, são as entradas da segunda etapa.

Temos então de levar em conta estas saídas na elaboração da tabela verdade que terá no seu topo as seguintes variáveis:

A B C S1 S2 S

A, B e C são as entradas dos circuitos. S1 e S2 são pontos intermediários do circuito, ou saídas intermediárias, que precisam ser analisadas para se obter S que é a saída final do circuito.

Começamos por colocar em A, B e C todas as suas condições possíveis, ou todas as combinações de níveis lógicos que podem ser aplicadas ao circuito:

A	B	C	S1	S2	S
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Veja que, como temos 3 entradas, as combinações possíveis para os níveis lógicos de entrada são 8. Os níveis aplicados equivalem, portanto, à contagem binária de 0 a 7.

O passo seguinte é colocar os valores possíveis de S1 que correspondem à saída do inversor. Sabemos que a tabela verdade para o inversor é:

A	S
0	1
1	0

Ora, como no nosso caso A é a entrada do inversor, e S1 é sua saída, podemos partir para a determinação de toda a coluna S1 simplesmente invertendo os valores da coluna A, da seguinte forma:

A	B	C	S1	S2	S
0	0	0	1		
0	0	1	1		
0	1	0	1		
0	1	1	1		
1	0	0	0		
1	0	1	0		
1	1	0	0		
1	1	1	0		

Para encontrar os valores da coluna S2, observamos que ela corresponde à tabela verdade da função AND, onde as entradas correspondentes são B e C e a saída é S2. Aplicamos, portanto, a função AND de entradas B e C, colocando os resultados na coluna S2.

B	C	S2
0	0	0
0	1	0
1	0	0
1	1	1

Como temos duas entradas, os resultados obtidos são 4, mas observamos que na tabela principal, as quatro combinações de entrada são repetidas duas vezes.

Temos então:

A	B	C	S1	S2	S
0	0	0	1	0	
0	0	1	1	0	
0	1	0	1	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	1	0	0	
1	1	0	0	0	
1	1	1	0	1	

Finalmente, levando em conta que S1 e S2 são entradas de uma porta OR de duas entradas, cuja saída é S e tem por tabela verdade a mostrada abaixo, podemos elaborar a coluna final de saídas (S):

S1	S2	S
0	0	0
0	1	1
1	0	1
1	1	1

Nesse caso, também, a combinação de 4 entradas é repetida duas vezes.

A tabela final fica então:

A	B	C	S1	S2	S
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

Aplicações práticas

Evidentemente, as aplicações para todos os tipos de funções existem, dependendo apenas do que o leitor está projetando. Pequenos automatismos, alarmes, fechaduras de código, são exemplos de circuitos onde combinações simples, como a dada como exemplo, podem ser usadas.

Trata-se de uma função bastante interessante que pode ser definida como “a que fornece uma saída alta somente quando a entrada A estiver no nível baixo, não importando as demais entradas ou ainda quando as três entradas estiverem no nível alto”.

Evidentemente, não sabemos em que aplicação prática o leitor pode precisar de tal função, mas fica o exemplo de como ela pode ser obtida e como podemos prever o seu funcionamento para todas as entradas possíveis, usando tabelas verdade.

5.3 - Como Projetar Um Circuito Combinacional

O problema de se saber o que acontece com a saída de um circuito formado por muitas funções lógicas, quando suas entradas recebem diversas combinações de sinais, não é o mais importante para o projetista de equipamentos digitais.

Na verdade, muito mais importante que este procedimento é justamente fazer o contrário, ou seja, projetar um circuito que, em função de determinados sinais de entrada forneça exatamente na saída o que se deseja.

O projeto de um circuito que tenha uma determinada função envolve um procedimento de síntese em algumas etapas.

Na primeira etapa deve ser definido o problema, estabelecendo-se exatamente qual é a função a ser executada, ou seja, quais as entradas e quais as saídas.

Numa segunda etapa coloca-se o problema numa tabela verdade ou ainda na forma de equações lógicas. O procedimento que abordaremos neste curso será basicamente o da obtenção das funções a partir das tabelas verdade e das equações lógicas.

Finalmente numa terceira etapa obtemos o circuito que exerça as funções desejadas.

Na terceira etapa, temos um ponto importante a ser considerado que consiste na minimização do circuito, já que na maioria dos casos pode-se implementar a mesma função de muitas formas diferentes como atesta o circuito simples mostrado na figura 124.

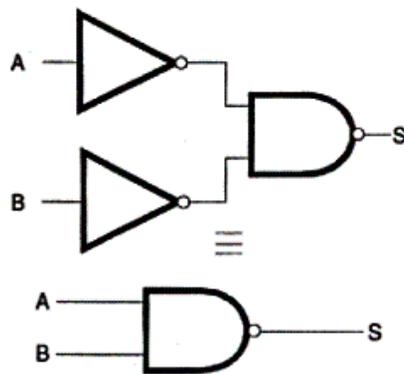


Figura 124 – Três funções combinadas que podem ser substituídas por uma única

Veja que podemos ter o mesmo circuito com quantidades de portas diferentes, o que significa que na prática deve-se levar este fato em conta. Não é apenas o número de portas que determinará a configuração final, mas sim seu custo e a eventual utilização em outras partes do circuito.

Por exemplo, se o circuito já estiver usando dois inversores dos seis disponíveis num circuito integrado e a nossa função tiver uma solução um pouco maior, mas que use estes inversores será interessante adotá-la para aproveitar os inversores ociosos.

Outro ponto a ser considerado é que hoje existem circuitos lógicos digitais que consistem em funções numa certa quantidade e que podem ser interligadas por programação.

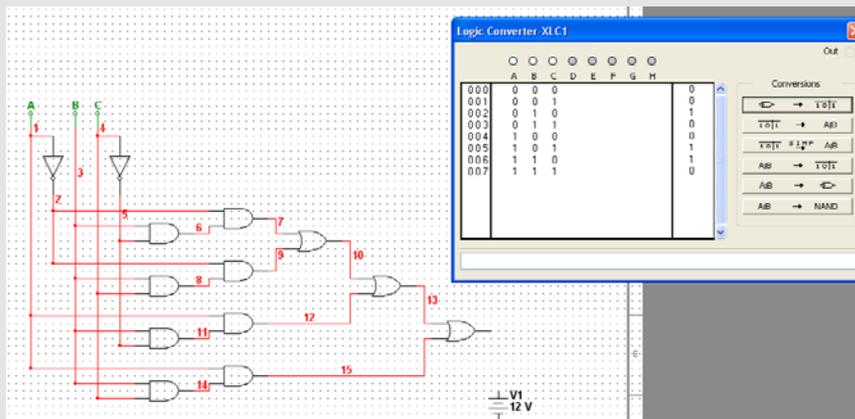
Assim, um mesmo circuito, só precisa ser programado de determinada forma para exercer as funções desejadas, interligando-se, através desse programa as portas que vão ser usadas.

A seguir daremos um exemplo de como podemos obter os circuitos a partir de uma tabela verdade.

Pedimos ao leitor especial atenção para esses procedimentos, que apesar de simples exigem cuidados, pois eles consistem numa poderosa ferramenta de trabalho para o projetista de circuitos digitais.

Simulação em Computador

Existem programas de computador, como o NI Multisim da National Instruments que incorporam ferramentas de análise de circuitos lógicas capazes de fazer também a simplificação. Na figura abaixo temos a tela do analisador lógico deste programa. Basta colocar a tabela verdade que ele “desenha” o melhor circuito que implemente a função que ela descreve.



Tela do Multisim 11 que utiliza o conversor lógico

a) Passo 1 - Determinação das equações lógicas

Lembramos que para as funções estudadas temos as seguintes representações:

* Função E (AND)

$$Y=A.B$$

* Função Não-E (NAND)

$$Y=\overline{A.B}$$

* Função OU (OR)

$$Y=A+B$$

* Função Não-OU (NOR)

$$Y=\overline{A+B}$$

* Função Não (NOT) ou inversor

$$Y=\overline{A}$$

* Função ou exclusivo (Exclusive OR)

$$Y=A(+)B$$

Vamos tomar como exemplo a tabela verdade abaixo para se determinar a função lógica correspondente:

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

Nesse exemplo, A, B e C são as entradas e Y a saída.

Indicamos os números das linhas na última coluna de modo a facilitar as explicações que daremos a seguir.

Conforme podemos ver temos saídas no nível 0 para as linhas 0, 3, 5 e 6 enquanto que para as linhas 1, 2, 4 e 7 temos saída 1.

Isso quer dizer que teremos a função OU para as linhas cuja saída é 1 podem ser encaradas como operações OR com tabelas que teriam 1 na saída apenas nas linhas 1, 2, 4 e 7, conforme mostrado a seguir:

A B C Y	A B C S1	A B C S2	A B C S3	A B C S4
0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0 0 1 1	0 0 0 1	0 0 1 0	0 0 1 0	0 0 1 0
0 1 0 1	0 1 0 0	0 1 0 1	0 1 0 0	0 1 0 0
0 1 1 0	= 0 1 1 0	+ 0 1 1 0	+ 0 1 1 0	+ 0 1 1 0
1 0 0 1	1 0 0 0	1 0 0 0	1 0 0 1	1 0 0 0
1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0
1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0
1 1 1 1	1 1 1 0	1 1 1 0	1 1 1 0	1 1 1 1

Isso nos permite escrever as equações lógicas para cada uma das quatro tabelas da seguinte forma:

$$S1 = \bar{A} \cdot \bar{B} \cdot C \text{ que corresponde a } A = 0, B = 0 \text{ e } C = 1$$

$$S2 = \bar{A} \cdot B \cdot C \text{ que corresponde a } A = 0, B = 1 \text{ e } C = 0$$

$$S3 = \bar{A} \cdot \bar{B} \cdot C \text{ que corresponde a } A = 1, B = 0 \text{ e } C = 0$$

$$S4 = A \cdot B \cdot C \text{ que corresponde a } A = 1, B = 1 \text{ e } C = 1$$

Como a saída S é a combinação das quatro funções temos:

$$S = S1 + S2 + S3 + S4$$

Notações

É interessante observar que a álgebra Booleana encontra aplicações tanto na eletrônica como na matemática e na lógica e que essas três ciências usam notações diferentes para representar suas operações.

Assim, podemos citar como exemplo as Leis da Tautologia que têm três representações diferentes:

a) Eletrônica

$$a + a = a$$

$$a \times a = a$$

b) Notação Matemática (Teoria dos Conjuntos)

$$a \cup a = a$$

$$a \cap a = a$$

c) Lógica

$$a \vee a = a$$

$$a \wedge a = a$$

Dificuldades

Se o leitor tem alguma dificuldade em entender o que fizemos será interessante voltar às lições anteriores em que falamos das funções lógicas básicas.

Substituindo pelos valores encontrados temos:

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

Esta é então a função lógica que representa a tabela verdade que propusemos como parte inicial do problema e para a qual devemos encontrar um circuito equivalente.

Passo 2 - Implementação dos Circuitos Combinacionais

Conforme estudamos em capítulos anteriores, é possível usar as portas NAND e NOR como blocos lógicos universais a partir dos quais é possível elaborar qualquer outra função ou mesmo funções mais complexas.

É por esse motivo que os circuitos integrados que contém essas funções são os mais comuns em todas as famílias lógicas.

Para exemplificar como isso pode ser feito, vamos tomar como exemplo uma função um pouco mais simples do que aquela que obtivemos no passo anterior.

Tomemos como exemplo a expressão:

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

Podemos tentar implementá-la usando portas NAND, e eventualmente inversores, já que a barra sobre cada letra indica sua negativa, conforme já estudamos.

A operação (.) pode ser realizada utilizando-se uma porta NAND que ligada a um inversor nos fornece uma porta AND.

Assim, conforme mostra a figura 125 podemos implementar $A.B.C$ usando uma porta NAND de 3 entradas e um inversor.

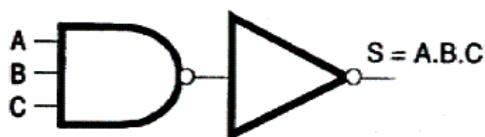


Figura 125 – Função $A.B.C$ implementada

A operação $A.B.C$ pode ser implementada conforme mostra a figura 126.

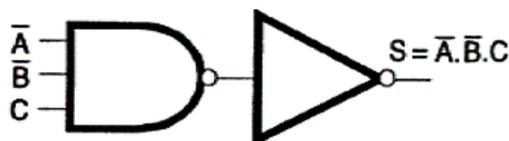


Figura 126 – Implementação da função $\bar{A}.\bar{B}.C$

A soma (+) pode ser implementada com uma porta AND ligada a dois inversores conforme mostra a figura 127.

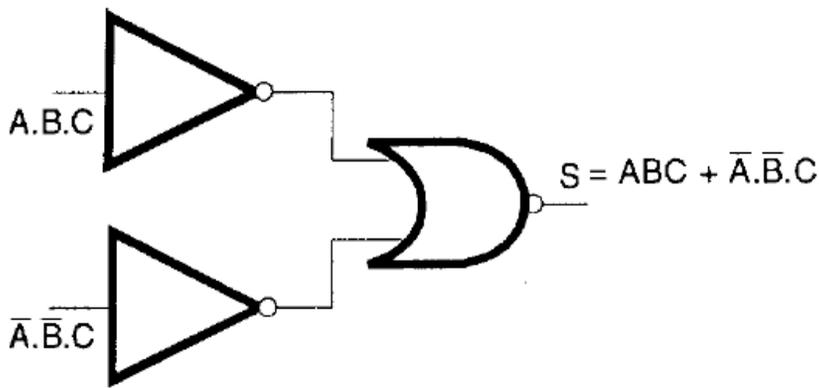


Figura 127 – Implementação da soma de duas funções

Combinando os três circuitos podemos chegar à configuração final desejada, mostrada na figura 128.

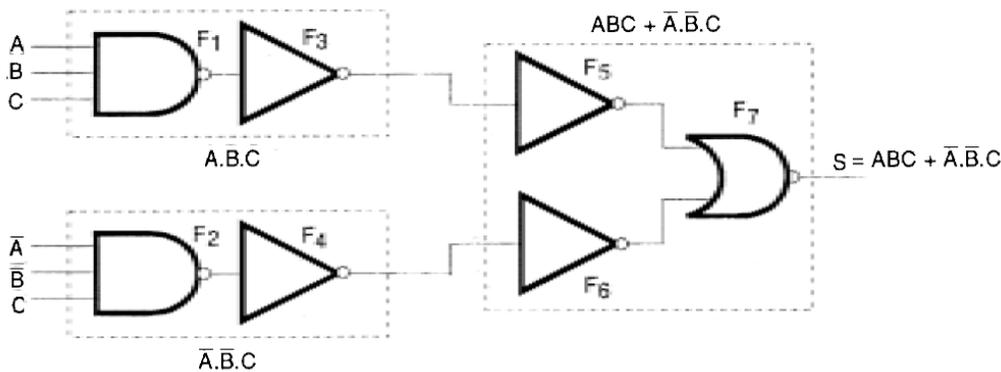


Figura 128 – Configuração quase final

Veja que a inversão da inversão usada no circuito interior nos leva ao circuito original. Isso significa que podemos simplificar a configuração, eliminando as duplas inversões em série.

Isso nos leva à configuração final do circuito mostrada na figura 129.

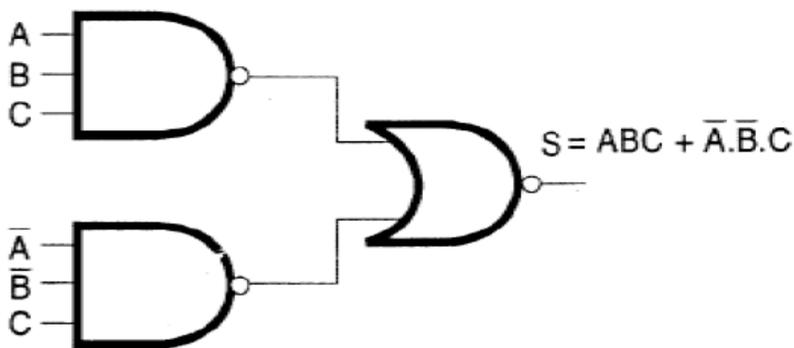


Figura 129 – Configuração final, eliminando-se os inversores

Veja ainda que quando temos uma expressão formada pela soma de produtos, podemos usar portas NAND sem a necessidade de inversores, bastando apenas lembrar duas propriedades:

- As combinações de entrada podem ser aplicadas à portas NAND.
- As saídas das portas NAND podem ser aplicadas à entrada de uma segunda porta NAND obtendo-se na saída a função desejada.

Vamos agora fazer uma tentativa para implementar uma função usando portas NOR, o que será escolhido quando tivermos um produto de somas.

Tomemos como exemplo a função:

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

As somas podem ser obtidas facilmente a partir de portas NOR com as saídas aplicadas a um inversor. A negação de NOR é OR. O circuito equivalente para três entradas é mostrado na figura 130.

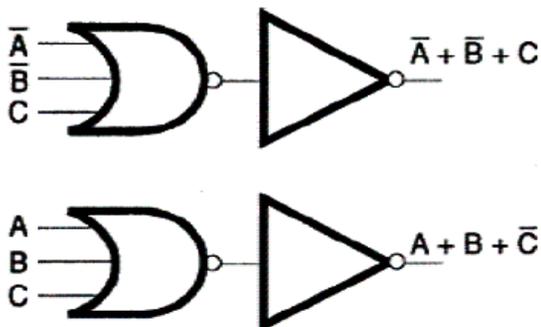


Figura 130 – Implementação da função soma

O produto das duas somas é obtido com dois inversores, aplicando os sinais a outra porta OR, ou seja, a outra configuração NOR.

Como nas duas linhas de sinais temos inversores em série, e como o inverso do inverso de um nível lógico é ele mesmo, podemos simplificar o circuito eliminando todos os inversores.

Isso nos permite chegar à configuração final que é mostrada na figura 131.

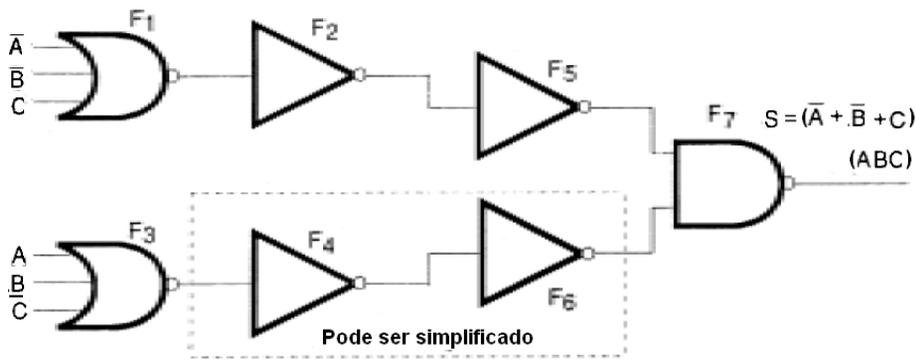


Figura 131 – Outra configuração com simplificação

Assim, se quisermos implementar uma função que consiste num produto de somas, basta seguir dois procedimentos básicos:

- Aplicar as entradas correspondentes a cada soma a uma porta OR que pode ser obtida associando-se uma porta NOR a um inversor.
- Aplicar as saídas obtidas nas funções que devem ser multiplicadas a inversores que são ligados às entradas de uma porta OR final, também obtida com a associação de um inversor a uma porta NOR.
- Como os inversores em série se anulam, eles podem ser eliminados e o circuito pode ser implementado utilizando-se apenas portas NOR.

Veja o leitor que podemos resolver o problema de implementar circuitos combinacionais, reduzindo as funções a produtos de somas ou ainda a soma de produtos, casos em que podemos trabalhar com funções NAND ou NOR, conforme mostra a figura 132.

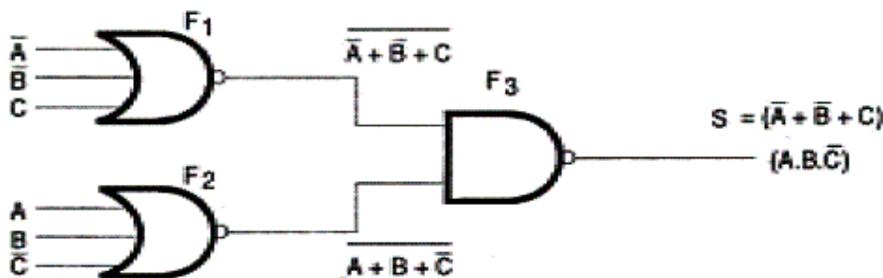


Figura 132- Resultado final da implementação

Como as duas soluções levam aos mesmos resultados, num projeto prático, é interessante analisar as configurações obtidas para um problema nos dois casos. Adota-se então a solução que utilizar menos circuitos, ou ainda que for mais conveniente, por exemplo, aproveitando portas ociosas de um circuito integrado já utilizado no mesmo projeto com outras finalidades.

Evidentemente, ao se utilizar as portas contidas nos circuitos integrados reais, o leitor deve estar atento às características elétricas, além de outras.

CMOS ou TTL

O que foi estudado vale para qualquer tipo de circuito integrado, não importando sua família lógica.

Prática

Para os leitores que desejarem ir além, a prática com a realização de exercícios é importante, assim como a montagem em matriz ou mesmo simulação em computadores são recursos interessantes para se aprender mais.

O que vimos, portanto, é apenas um exemplo do modo como podemos proceder para implementar funções complexas usando portas.

O grau de complexidade dessas funções pode aumentar muito, caso em que o trabalho “manual” descrito se torna difícil. Existem softwares que ajudam a projetar tais circuitos, os quais devem ser procurados para os que pretendem fazer tais tipos de projetos.

5.4 – Simplificando e Minimizando

Uma consequência da possibilidade de se construir funções complexas a partir de portas básicas como OR e AND (OU e E), é poder otimizar um projeto aproveitando-se poucos tipos de circuitos integrados básicos.

Assim, se tivermos uma função que seja obtida utilizando-se portas AND e OR como a mostrada na figura 133, ela tem o inconveniente de precisar de dois tipos diferentes de circuitos integrados.

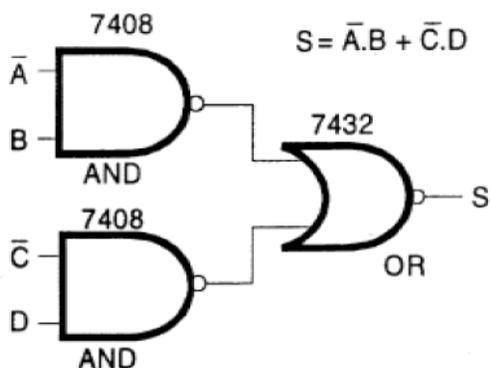


Figura 133 – Esta função utiliza dois tipos de circuitos integrados

Se quisermos esta função com circuitos TTL, por exemplo, aproveitaremos três das três portas de 3 entradas de um circuito 7411, e também precisaremos aproveitar uma das quatro portas OR de duas entradas de um circuito integrado 7432.

Evidentemente, estaremos usando dois circuitos integrados desperdiçando 1/3 de um e 3/4 do outro.

Podemos simplificar consideravelmente este circuito se usarmos apenas portas NAND com a configuração equivalente mostrada na figura 134.

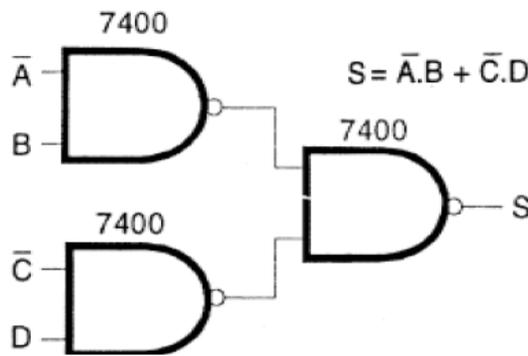


Figura 134 - A mesma função com um único tipo de porta

Este circuito, que tem a mesma função do anterior, usa as três portas de um circuito integrado 7410. O resultado final é que usamos apenas um circuito integrado, que é totalmente aproveitado, sem nenhuma parte ociosa.

Veja que é importante considerar essa possibilidade num projeto comercial onde o número de componentes, e o seu custo, são fatores importantes para sua implementação.

5.5 – Diagramas de Karnaugh

Um processo bastante interessante de se representar uma tabela verdade, e a partir dela se obter uma simplificação dos circuitos utilizados para sua implementação, é o que faz uso dos chamados diagramas ou mapas de Karnaugh.

O diagrama de Karnaugh consiste numa tabela retangular com número de quadros que corresponde a 2 elevado ao expoente N, onde N é o número de variáveis do circuito.

Cada variável lógica ocupa no gráfico metade da sua extensão, e seu complemento ocupa a outra metade.

Na figura 135 mostramos o modo como são elaborados os diagramas de Karnaugh para 1, 2 e 3 variáveis, com as expressões lógicas correspondentes a cada caso.

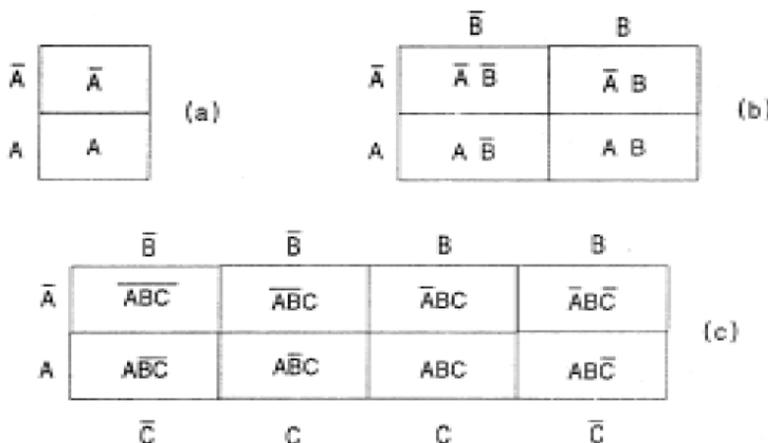


Figura 135 - Diagramas de Karnaugh para 1, 2 e 3 variáveis

Karnaugh

Maurice Karnaugh, físico americano e engenheiro de telecomunicações criador do mapa ou diagrama que leva seu nome.



Maurice Karnaugh (1924-)

Veja que, para 1 variável, temos 2 casas; para 2 variáveis, 4 casas e para 3 variáveis, 8 casas.

As expressões lógicas, associadas a cada quadrinho, são obtidas de uma forma muito semelhante a usada no conhecido joguinho de “batalha naval” onde a posição de cada “tiro” é dada por duas coordenadas, uma correspondente as linhas e outra às colunas.

Na figura 136 mostramos, como exemplo, de que modo um diagrama de Karnaugh de 4 variáveis pode ser obtido com a inclusão dentro de cada quadro da expressão correspondente.

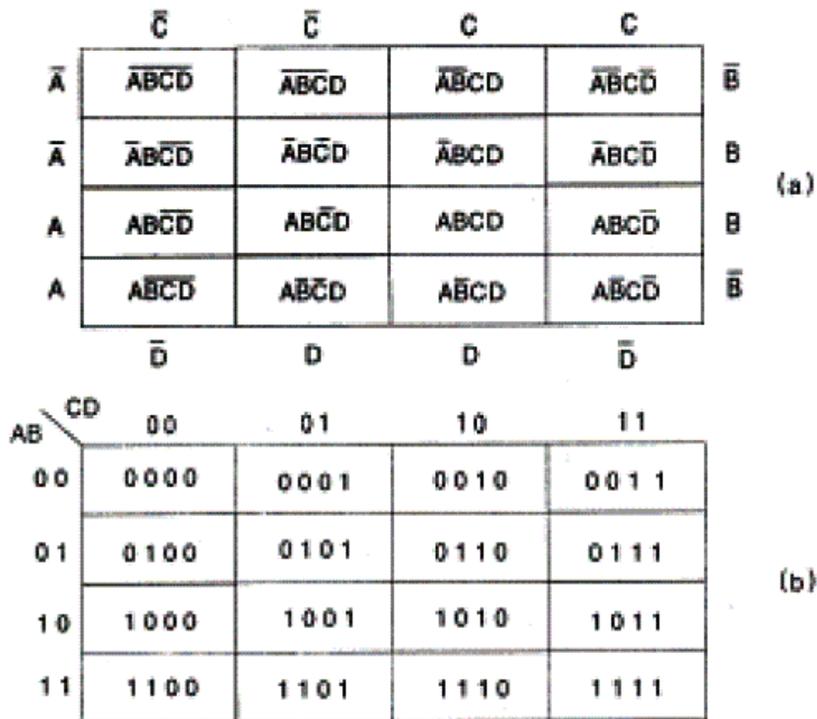


Figura 136 – Diagramas de Karnaugh e Veitch

No diagrama (b) da figura 123 os quadros foram preenchidos com os valores 0 e 1 correspondentes às entradas. Este diagrama é chamado também de diagrama de Veitch.

Uma observação importante que pode ser feita em relação a esta representação por 0 e 1, é que cada quadro difere do adjacente em apenas um dígito. Dizemos que são adjacentes os termos que estão à direita e a esquerda de cada quadro e, também, os que estão acima e abaixo. Também são adjacentes os que estiverem na mesma fila, mas um na primeira coluna e outro na última.

Na figura 137 temos um mapa com a identificação das adjacências.

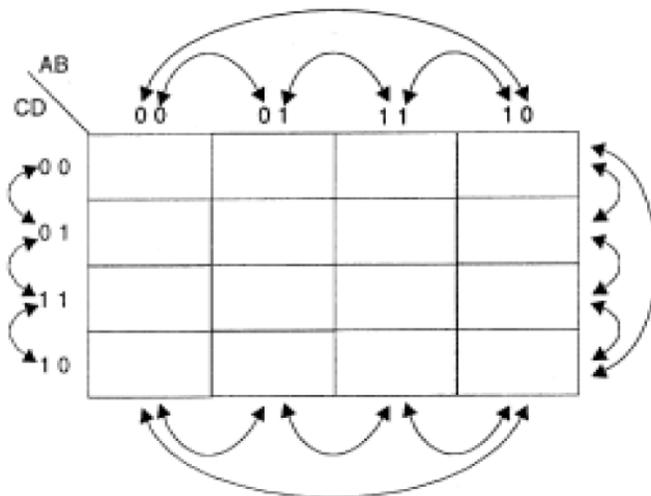


Figura 137 – Adjacências para o diagrama ou mapa de Karnaugh

Assim, para podermos fazer as simplificações, o que fazemos é plotar a tabela verdade da função que desejamos implementar num mapa de Karnaugh. Com isso, será possível identificar melhor as adjacências, e assim fazer as simplificações que levam ao uso mínimo de funções lógicas na implementação de um circuito.

Para que o leitor entenda como “funciona” o mapa de Karnaugh numa simplificação de uma função, vamos tomar como exemplo uma função relativamente simples, função que é dada pela seguinte tabela verdade:

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

Desejamos expressar essa tabela como uma soma de produtos, o que significa que os valores adjacentes que devemos procurar na tabela, são os “1”. Se fossemos expressar esta função como o produto de uma soma, os valores considerados seriam os “0”, e o procedimento final seria o mesmo.

Construímos então o Diagrama de Karnaugh para esta tabela conforme mostra a figura 138.

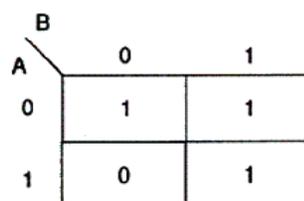


Figura 138 – A tabela verdade é plotada no mapa de Karnaugh

Batalha naval

Veja que no cruzamento da linha A=0 com a B=0 temos a casa com preenchida com o valor 1, que é o valor obtido na tabela verdade para as entradas 00.

A partir deste diagrama, nosso próximo passo consiste em se tentar fazer simplificações que possam levar a circuitos mais simples na implementação.

A idéia é então agrupar os termos adjacentes, que sejam iguais, havendo para isso diversas possibilidades que são mostradas na figura 139.

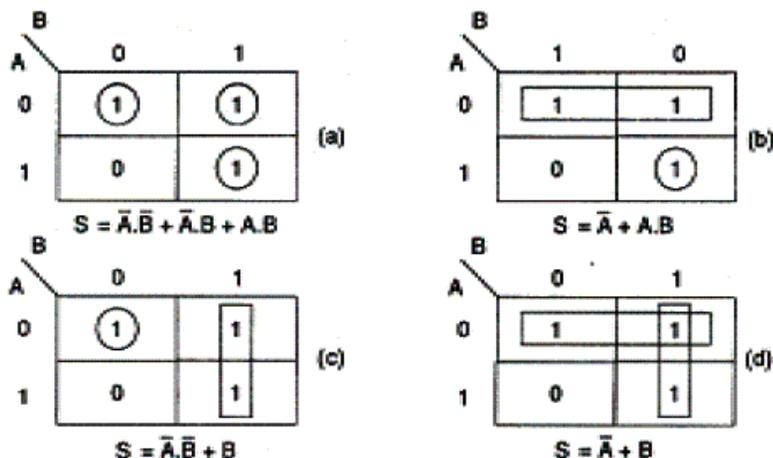


Figura 139 – Agrupamentos possíveis para termos adjacentes iguais

A primeira possibilidade que é mostrada em (a) nos leva a uma soma de três produtos, cada qual obtido pela intersecção da linha com a coluna em que está o “1” correspondente.

Assim, o primeiro está na coluna que intercepta A=0 com B=0. Ora, o valor zero na indexação indica inversão, portanto, isso significa que o primeiro fator de nosso produto será:

$$\bar{A} \cdot \bar{B}$$

O segundo “1” a ser considerado está na coluna A = 1 e B = 0 portanto temos A invertido e B sem inversão, o que nos leva ao segundo fator de nosso produto:

$$A \cdot \bar{B}$$

Finalmente, o terceiro “1” a ser considerado está na linha A = 1 e B = 1 o que significa um fator com A, multiplicado por B sem inversões ou:

$$A \cdot B$$

Como devemos expressar a função na forma de uma soma de produtos fazemos:

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

Para o segundo caso (b), temos uma simplificação maior já que agrupamos os dois “1” da primeira linha de modo que podemos adotar para ele:

$$\bar{A}$$

Para o outro valor “1” que está na casa que corresponde à intersecção de $A = 1$ com $B = 1$ vale a soma (sem inversão):

$$A + B$$

A expressão final na forma de um produto de somas serentão:

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

Da mesma forma, chegamos a simplificação (b) que é a que permite a obtenção de uma expressão mais simples, pois conseguimos juntar três casas adjacentes.

Raciocinando da mesma forma chegamos à expressão:

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

O procedimento que vimos como exemplo, envolveu uma função simples, com apenas duas variáveis de entrada. No entanto, exatamente o mesmo procedimento é válido para qualquer número de variáveis.

Os leitores interessados num aprofundamento devem procurar treinar os procedimentos indicados trabalhando com funções cada vez mais complexas.

Obras mais avançadas, que tratam de lógica digital, devem ser consultadas para os que desejarem ir além, já que a finalidade desse curso é uma introdução à eletrônica digital.

Outros métodos de simplificação existem como o Método Tabular de Quine-Mac Cluskey que são mais apropriados para a simplificação de equações de mais de 5 variáveis. Este método é recomendado para número elevado de variáveis, já que o método de Karnaugh torna-se muito complicado para se usar com mais de 5 variáveis.

Conclusão

Não é finalidade deste livro um aprofundamento maior no assunto, e certo treino se faz necessário, para que o leitor domine as técnicas envolvidas.

Assim, para os leitores interessados em se aprofundar no tema, sugerimos a procura de literatura complementar mais completa.

O que mostramos aos leitores é que existem procedimentos lógicos que permitem trabalhar com as funções de modo a se chegar aos circuitos práticos e depois fazer sua implementação.

Simulações

Programas de simulação como o Multisim, geram as equações lógicas a partir de uma tabela verdade automaticamente e ainda desenham o circuito. Se o leitor pretende se aprofundar no assunto, é interessante tomar contato com este software da National Instruments.

Conhecimento de lógica

Para entender bem o que foi visto neste capítulo e partir para um aprofundamento maior, o leitor precisa ter um excelente conhecimento de lógica digital.

Assim, a partir de uma tabela verdade que tenha qualquer combinação de entradas que nos leve a qualquer combinação de saída, pode ser elaborada na prática com funções básicas (NOR e NAND) e isso não exige que se “quebre a cabeça”.

Conhecendo os procedimentos para se resumir tudo em produto de somas e soma de produtos e também o uso dos mapas de Karnaugh para simplificação pode-se obter configurações simples que facilitam qualquer projeto.

Em suma, a partir de qualquer combinação de entradas e saídas pode-se obter o circuito que as realize a partir de funções lógicas comuns, usando os procedimentos que descrevemos nessa lição.

Também é importante observar que, com poucas funções básicas, podemos ter qualquer combinação possível de comportamentos para o circuito implementado.

Termos em inglês

Como temos citado nos capítulos anteriores, na documentação técnica em português que trata de eletrônica digital, muitos termos são mantidos na forma original em inglês, mesmo existindo uma tradução, caso das funções lógicas. Assim, damos alguns deles a seguir:

Combinational – combinacional

Diagram – diagrama

Truth table – tabela verdade

Logic equation – equação lógica

Design – projeto, projetar

Reducing – reduzindo

Adjacent - adjacente

Termos para pesquisa:

- Diagramas ou mapas de Karnaugh
- Operações lógicas
- Lógica combinacional
- Simplificação de circuitos lógicos
- Diagramas de Veitch
- Tabelas verdade
- Analisador lógico
- Multisim

QUESTIONÁRIO

1. Os valores combinados de todas as entradas e a saída correspondente podem ser colocados numa tabela que é denominada:

- a) Mapa de Karnaugh
- b) Diagrama de Veitch
- c) Tabela verdade
- d) Produto de somas

2. A tabela verdade abaixo corresponde à que função:

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

- a) AND (E)
- b) NAND (Não-E)
- c) OR (OU)
- d) NOR (Não-OU)

3. Qualquer circuito lógico pode ser implementado utilizando-se que funções básicas?

- a) NAND e inversores
- b) NAND e NOR
- c) OR e Inversores
- d) AND e Inversores

4. Para se implementar um circuito que corresponda a uma função dada por uma soma de produtos usamos que funções lógicas?

- a) Portas NAND
- b) Inversores
- c) Portas OR
- d) Não é possível fazer isso

5. Se numa implementação lógica precisarmos usar inversores em série o que podemos fazer com eles?

- a) Ligá-los à portas AND
- b) Colocá-los em paralelo
- c) Inverter suas saídas
- d) Eliminá-los



» Os Elementos Biestáveis

No capítulo anterior analisamos os modos segundo os quais podemos saber o que acontece quando combinamos funções lógicas. Vimos, na oportunidade, os procedimentos que podem ser utilizados para se implementar um circuito a partir de uma tabela verdade ou ainda da expressão da função lógica. Este capítulo de nosso estudo de eletrônica digital foi denominado lógica combinacional. No entanto, as funções lógicas não consistem nos únicos blocos básicos que são usados nos projetos de circuitos digitais. Além dessas funções básicas, existem outras e um grupo especial delas, que executa tarefas de relevante importância nos equipamentos digitais, é o formado pelos elementos biestáveis também chamados de básculas ou flip-flops. Nesta lição veremos como funcionam estes elementos, os seus tipos e onde eles podem ser usados.

Itens:

- 6.1 – Os Flip-Flops
- 6.2 – Flip-flop R-S
- 6.3 – Flip-flop R-S com clock e mestre-escravo
- 6.4 – Flip-Flop J-K com mestre e escravo
- 6.5 – Flip-flop tipo D
- 6.6 – Flip-flop tipo T
- 6.7 – Transformando flip-flops
- 6.8 – Nos equipamentos digitais
- 6.9 – Nos equipamentos antigos

Objetivos:

- Entender como funcionam os flip-flops
- Reconhecer os diversos tipos de flip-flops e como funcionam
- Saber como transformar os diversos tipos de flip-flops
- Ter uma idéia de como eles são utilizados nos equipamentos digitais
- Entender como os flip-flops podem ser usados como memórias e contadores

Bit

O bit é a unidade de informação, corresponde a um 1 ou a um 0.

6.1 - Os Flip-Flops

O termo flip-flop, bscula ou multivibrador biestvel  associado a um tipo de circuito que, apresentando apenas dois estados possveis, se torna compatvel com algumas aplicaes de sequenciamento digital.

Por esse motivo, os flip-flops consistem em dispositivos de grande importncia na implementao de projetos de eletrnica digital, merecendo uma lio completa  parte.

Estes circuitos podem ser usados para armazenar bits de informao, para fazer o sequenciamento de informaes digitais e para fazer a diviso de freqncia de sinais digitais, alm de muitas outras funes que ficaro claras a partir desse momento, no decorrer deste curso.

Analisemos ento como funcionam os flip-flops ou multivibradores biestveis.

Os flip-flops so elementos de circuito que podem apresentar em seu funcionamento apenas dois estados estveis. No existem estados intermedirios entre estes dois estados.

Podemos compar-los a uma gangorra em que existem apenas dois estados estveis possveis: quando o lado A est no nvel baixo ou lado B est no alto e quando o lado A est no alto o lado B est no baixo. As situaes intermedirias so instveis e no se mantm seno durante as transies ou mudanas de estado, conforme mostra a figura 140.

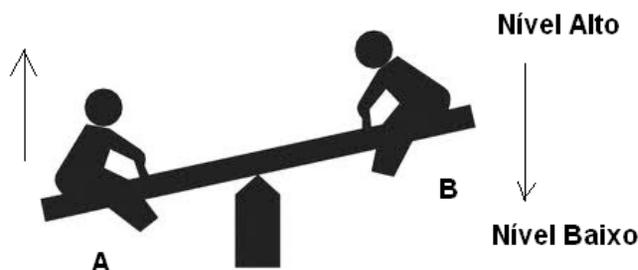
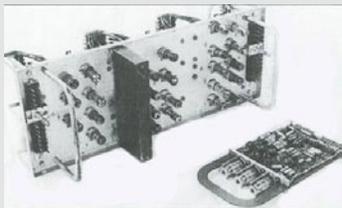


Figura 140 – Numa gangorra quando um lado est no nvel alto, o outro, obrigatoriamente, est no nvel baixo

Flip-Flops a vlvulas

Nos primeiros computadores que existiram, os flip-flops eram implementados com vlvulas, funcionando como unidades de memria. Cada flip-flop continha duas vlvulas, armazenava um nico bit, e consumindo uma enorme potncia eltrica.



Mdulos de flip-flops de vlvulas de um computador da dcada de 1950

No caso especfico de um flip-flop eletrnico, implementado com componentes como transistores, a aplicao de um sinal de entrada pode causar uma mudana de um estado para outro, e como a qualquer momento podemos saber qual  o estado em que ele se encontra, podemos considerar este circuito como uma unidade de memria capaz de armazenar um bit.

Por essa capacidade de armazenamento, o flip-flop  usado como elemento bsico de diversos tipos de memrias.

Existem diversos tipos de flip-flops que podem ser encontrados nos circuitos digitais.

Posteriormente, foram implementados flip-flops com transistores discretos e o passo final na sua evolução, foi a sua integração como elementos de circuitos integrados de famílias lógicas como a TTL, CMOS ou ainda em memórias de diversos tipos.

Para nós, em especial, interessa conhecer os tipos principais de flip-flops que podem ser encontrados nos circuitos integrados das famílias digitais principais, ou seja, TTL e CMOS e que analisaremos a partir de agora.

6.2 – Flip-flop R-S

O flip-flop R-S (de Reset e Set) – Este circuito tem sua configuração com transistores mostrada na figura 141, funcionando da seguinte maneira:

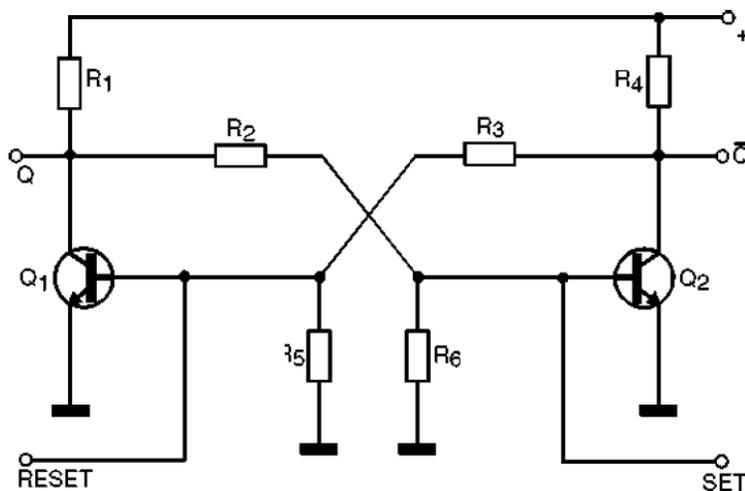


Figura 141 – Um flip-flop R-S com transistores NPN

Quando alimentamos o circuito, dadas as mínimas diferenças que podem existir entre as características dos dois transistores, um deles conduzirá mais do que o outro. Supondo que este transistor seja Q1 o que ocorre é que há uma queda de tensão no seu coletor que reduz, em consequência, a corrente que polariza a base de Q2 via R2.

Nestas condições, a tensão do coletor de Q2 se mantém alta, realimentando a base de Q1 via R3 e a situação final do circuito é estabelecida: Q1 satura e Q2 fica no corte. O flip-flop encontra seu estado estável inicial.

O flip-flop R-S tem duas saídas, representadas por Q e /Q, assim na condição inicial estável com Q1 conduzindo Q estará no nível baixo (0) e /Q estará no nível alto (1).

O processo que leva o flip-flop a este estado inicial, pronto para funcionar é muito rápido, não demorando mais do que alguns micros-

Consumo

Para que os leitores tenham uma idéia, uma válvula comum precisa de 6 V x 500 mA, apenas para seu filamento, o que resulta em 3 W de consumo, colocando-se mais 2 W para o circuito de placa. Como cada flip-flop precisa de duas válvulas, o consumo seria de 10 W para cada um. O ENIAC tinha aproximadamente 10.000 flip-flops, o que dá um consumo de 100 kW ou de aproximadamente uma vila com 200 casas, imagine qual seria o consumo de um processador moderno com 100 milhões de flip-flops!

Set e Reset

Os termos Reset e Set são muitos usados em eletrônica digital e informática, sendo até já incorporados ao nosso idioma. Frases que usam os verbos “resetar” ou “setar” são comuns, tanto no linguajar diário dos profissionais, como em publicações técnicas. Uma tradução melhor desses termos seria “fixar”, para “setar” e “reiniciar”, para “resetar”. No entanto, usaremos aqui os termos tradicionais set e reset e seus derivados como “setar” e “resetar”.

segundos, ou mesmo nanossegundos dependendo dos componentes usados.

Quando o flip-flop se encontra na situação indicada, com $Q=0$ e $\bar{Q}=1$ dizemos que ele se encontra “setado” ou ainda fixado (o termo “armado” também pode ser usado).

A mudança de estado do flip-flop pode ser obtida aplicando-se um sinal na entrada conveniente. Como usamos transistores NPN, para comutar o flip-flop, temos de fazer conduzir por um instante o transistor que está cortado, ou seja, devemos aplicar um pulso positivo na entrada correspondente.

Assim, estando o flip-flop na condição indicada e desejarmos mudar o estado, aplicamos o pulso na entrada SET. O transistor Q2 conduz por um instante, realimentando via R3 a base de Q1 que é cortado. Com o corte, a tensão na base de Q2 sobe via polarização de R2, e mesmo que o pulso de disparo desapareça, o circuito se mantém no novo estado graças à realimentação.

Sua saída Q vai ao nível (1) e a saída \bar{Q} vai ao nível (0).

Para trocar novamente de estado o flip-flop R-S, aplicamos um pulso positivo na entrada RESET levando Q1 a saturação e Q2 ao corte, situação que se firma mesmo depois de desaparecido o pulso graças à realimentação proporcionada pelos resistores.

Veja que um pulso aplicado à entrada SET, o que corresponde a um bit 1, faz com que a saída Q que estava em 0 passe a 1 armazenando este bit. O flip-flop funciona realmente como uma memória para este bit.

Da mesma forma como utilizamos transistores bipolares NPN para obter um flip-flop, podemos também empregar outros tipos de componentes em configurações semelhantes. Podemos, por exemplo, elaborar flip-flops usando transistores PNP caso em que a polaridade dos sinais de disparo vai ser invertida.

Da mesma forma, podemos usar transistores de efeito de campo (FET), tanto de canal N como canal P (bipolares ou JFETs) como também transistores de efeito de campo MOS, com os dois tipos de canal (N ou P). O que vai mudar em cada caso é o sentido de circulação das correntes e as polaridades dos sinais aplicados.

Conforme já falamos, e veremos em detalhes no último item desta lição, os flip-flops também podem ser feitos com válvulas e na realidade os primeiros que existiram eram justamente montados com estes componentes. Naquela época não existiam transistores (bipolares ou FETs) e nem circuitos integrados.

Os flip-flops podem ser elaborados com portas lógicas e o R-S que estudamos pode ser facilmente obtido a partir de duas portas NAND de duas entradas, conforme mostra a figura 142.

Todos ressetados

Nas aplicações em que sejam utilizados muitos flip-flop é importante que, ao se ligar o circuito, todos estejam no mesmo estado, ou seja, com um mesmo “lado” conduzindo. Para isso, podem ser agregadas entradas no circuito que ressetem ou setem o circuito da forma desejada.

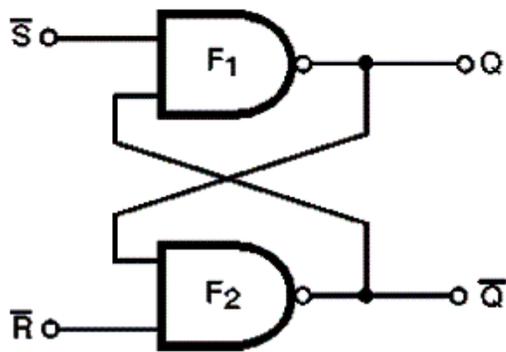


Figura 142 – Flip-flop com duas portas NAND

Levando em conta as tabelas verdade das portas NAND, vemos que a saída da primeira porta realimenta a segunda e vice-versa, garantindo assim a manutenção dos estados obtidos quando o flip-flop comuta.

No entanto, a comutação deste circuito ocorre quando as entradas passam do nível alto para o baixo, ou seja, de 1 para 0. Esta condição é indicada pelos símbolos \bar{R} e \bar{S} nas entradas.

O leitor pode então perceber que, quando as entradas estão ambas no nível baixo, o flip-flop se mantém no estado em que foi colocado por ser ligado ou por uma comutação anterior.

Por outro lado, se as entradas forem levadas simultaneamente ao nível alto, o flip-flop vai para um estado indeterminado que deve ser evitado. Na prática, a aplicação de níveis altos (1) nas duas entradas pode destruir o dispositivo.

O diagrama de tempos da figura 143 mostra o que ocorre no funcionamento de um flip-flop por etapas que podem ser analisadas da seguinte forma.

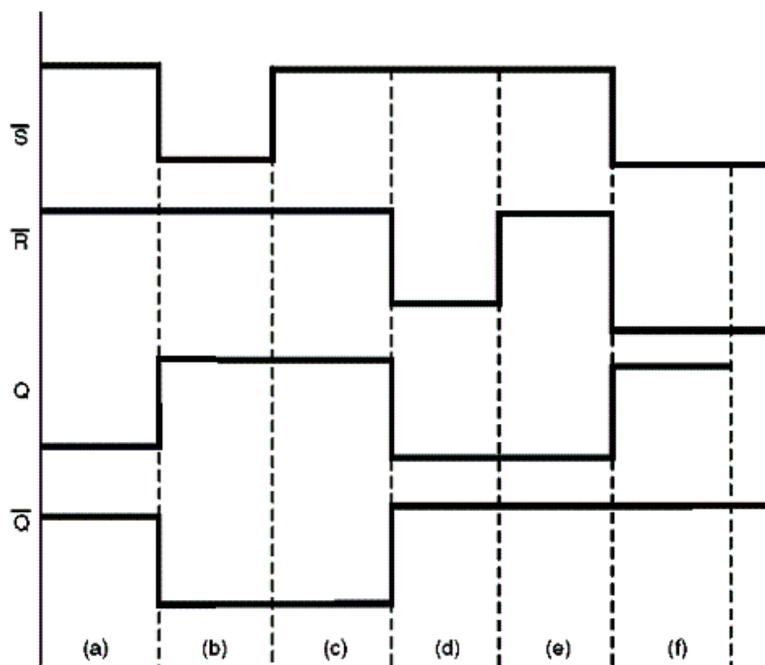


Figura 143 – Em (f) temos uma condição não permitida com R e S no nível baixo e Q e /Q no nível alto.

Barra

A barra que encontramos nas especificações de entradas e saídas indica que ela está ativa, ou se torna ativa, no nível baixo. Assim, o /R indica que para ressetar devemos levar essa entrada ao nível baixo. O R, simplesmente, indicaria que é preciso levar a entrada ao nível alto para ressetar.

Na figura temos as seguintes condições do flip-flop a serem consideradas:

- a) Flip-flop ressetado
- b) /S vai ao nível baixo e o flip-flop é setado
- c) /S vai ao nível alto e o flip-flop permanece setado
- d) /R vai ao nível baixo e o flip-flop é ressetado
- e) /R volta ao nível alto e o flip-flop permanece ressetado

Tudo isso pode ser representado por uma tabela verdade, da mesma forma que fazemos com as funções lógicas. Nesta tabela temos alguns novos símbolos com os quais o leitor deve começar a se familiarizar e que são amplamente usados em eletrônica digital, a saber:

- a) Primeira possibilidade

Q_{n-1} = representa o estado da saída Q ANTES da aplicação dos sinais.

Q_n = representa o estado da saída Q DEPOIS da aplicação dos sinais.

- b) Segunda possibilidade

Q = representa o estado da saída Q ANTES da aplicação dos sinais.

Q_{n+1} = representa o estado da saída Q DEPOIS da aplicação dos sinais.

Obs: em lugar de n em alguns livros encontramos a letra t.

Os dois tipos de representação são usados.

Nas colunas e linhas em que são colocados os níveis lógicos 0 e 1, quando aparece o termo Q_n ou $/Q_n$ isso indica que a saída vai para um estado indeterminado.

A tabela verdade do flip-flop R-S com portas NAND fica então:

R	S	Q_{n+1}	$/Q_{n+1}$
0	0	1	1
0	1	0	1
1	0	1	0
1	1	Q_n	$/Q_n$

Condições proibidas

Em lógica digital existem muitos casos em que ocorrem condições proibidas. Estas condições podem levar o circuito a um estado indeterminado ou mesmo causar a queima de um dispositivo. É preciso muita atenção para se evitar estas condições ao se realizar o projeto de um circuito digital.

Para obtermos um flip-flop R-S também podemos usar portas NOR, conforme mostra a figura 144.

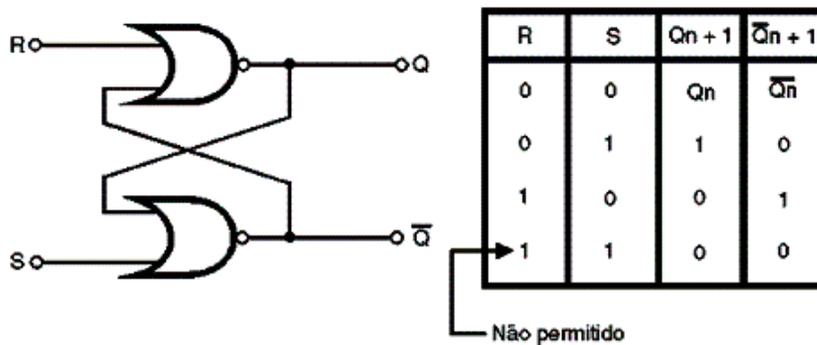


Figura 144 – Flip-flop R-S com portas NOR

Na figura 145 temos os símbolos adotados para representar este tipo de flip-flop.

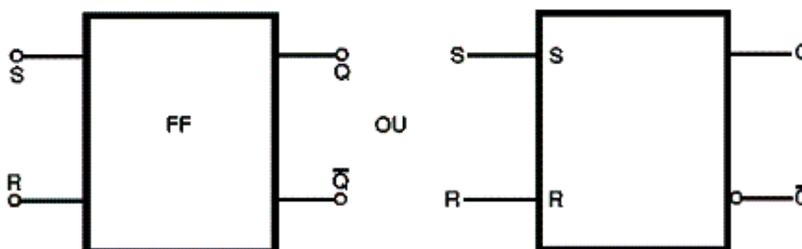


Figura 145 – Símbolos do flip-flop R-S

Este circuito também é chamado de R-S NOR LATCH, da mesma forma que o circuito anterior é denominado R-S NAND LATCH.

A palavra “latch” significa trava, o que nos sugere que se trata de um circuito que “trava” numa certa posição a partir dos comandos R e S e que usa portas NAND ou NOR. O termo latch, em inglês pode, portanto, também ser associado a flip-flops.

Latch ou trava

Este termo também é encontrado em outros tipos de circuitos, por exemplo, de eletrônica analógica, referindo-se a alarmes que “travam” uma vez disparado, por exemplo. Alarme com “latch” ou circuito com “latch” é um circuito que trava, depois de disparado como, por exemplo, os que usam SCRs.

6.2.1 – Repiques

Quando fechamos um interruptor, ou ainda um sensor é acionado, o estabelecimento da corrente no circuito não é imediato. Os contatos mecânicos tendem a oscilar, causando assim pulsos de variação da tensão ou da corrente, denominados repiques, ou usando o termo inglês “bounce” (balanço),

Esses repiques fazem com que o circuito digital que deva receber o comando do interruptor ou sensor interprete o sinal de entrada como mais de um pulso, conforme mostra a figura 146.

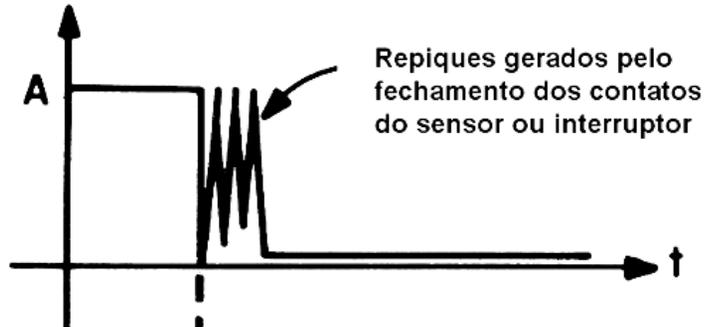


Figura 146 – O repique

Neste caso, o fechamento do sensor é interpretado como uma sequência de três pulsos. Se o circuito tiver de contar pulsos, a contagem será de 3, e não apenas um fechamento.

Clock

Eis mais um termo técnico que é comum usar na forma original em inglês: clock. Mesmo, na literatura técnica em português, encontramos clock em lugar de “relógio”, que seria a palavra correta, se feita a tradução. Clock normalmente indica o circuito ou sinal que sincroniza o funcionamento de um equipamento digital, ou seja, determina sua velocidade de operação.

6.3 – Flip-flops RS com Clock Mestre-Escravo

Este circuito, chamado de flip-flop R-S controlado por clock e mestre escravo, encontra uma gama de aplicações muito grande nos circuitos digitais mais complexos, já que eles são sempre comandados por um clock, ou seja, são circuitos lógicos sincronizados.

O uso de um circuito de controle (mestre) que determina quando o flip-flop (escravo) muda de estado é importante para permitir que as mudanças de estado do flip-flop só ocorram em instantes bem determinados. Usando portas NAND podemos inicialmente implementar um flip-flop R-S controlado por clock (Master-Slave) conforme mostra a figura 147.

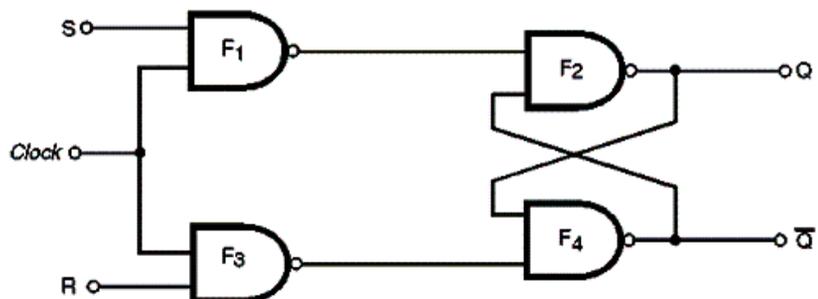


Figura 147 – Flip-flop R-S com entrada de clock

Analisemos seu funcionamento:

Partindo da situação em que a entrada de clock (relógio) esteja no nível baixo, as saídas Q e /Q permanecerão no estado inicial em que se encontravam e insensíveis a qualquer variação que ocorra nas entradas S e R (set e reset).

Quando a entrada de clock for levada ao nível 1, o circuito passa a responder aos sinais das entradas R e S.

No entanto, conforme mostra o diagrama de tempos da figura 148, este circuito tem um inconveniente.

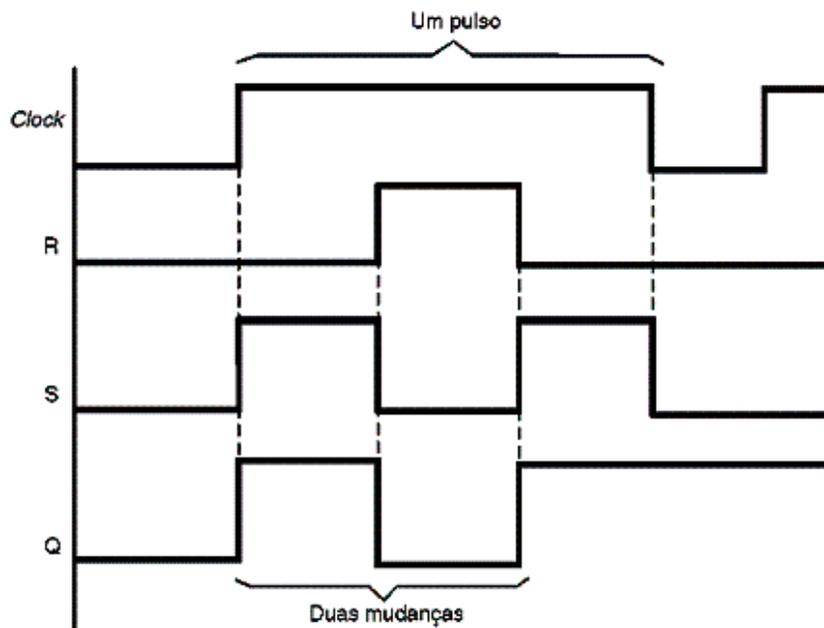


Figura 148 – Diagrama de tempos para o flip-flop R-S com clock

Como as saídas acompanham as entradas, durante o tempo em que o clock as habilita, estas saídas podem mudar de estado mais de uma vez, voltando assim ao estado inicial, o que não é desejado de forma alguma, pois levaria o circuito a uma instabilidade de funcionamento.

Um modo de contornar este problema consiste na utilização de duas etapas numa configuração mais complexa, que é mostrada na figura 149.

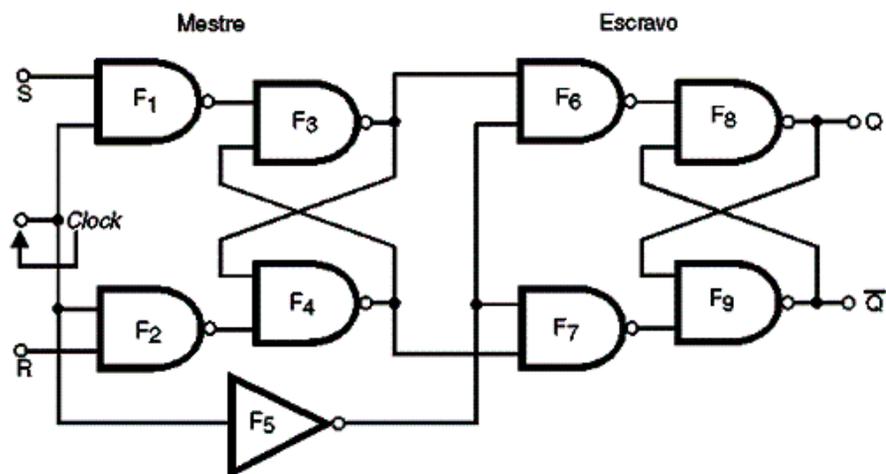


Figura 149 – Circuito mais elaborado para o flip-flop R-S com clock

Obtem-se então uma configuração mais elaborada em que se isola a etapa de saída que o flip-flop propriamente dito e a etapa de controle, em que temos os comandos set e reset.

Este circuito é denominado Flip-Flop R-S Mestre-Escravo ou Flip-Flop R-S Master-Slave podendo fazer uso, tanto de portas NAND, como um inversor cuja finalidade é inverter o pulso de clock (F5).

Isolando as duas etapas, as mudanças de sinal que ocorrem na primeira etapa não afetam a segunda etapa, a não ser quando isso é necessário.

O que ocorre neste caso é que, quando a entrada de clock for ao nível 1, o flip-flop mestre mudará de estado, mas o flip-flop escravo permanecerá insensível, mantendo seu estado.

Quando a entrada de clock passar para o nível lógico 0, a saída do flip-flop mestre será levada para o escravo.

Isso significa que o flip-flop, em seu todo, não é sensível ao nível do sinal de clock, ou seja, se ele é 0 ou 1, mas sim à sua transição. As saídas Q e /Q só vão mudar de estado no instante em que ocorrer a transição, do sinal de clock do nível alto para o nível baixo.

Com esta configuração é possível garantir que só vai ocorrer uma mudança de estado na presença de um pulso de clock.

Os flip-flops que funcionam desta forma são denominados “Edge Triggered” ou “Disparados pela Borda”, referindo-se ao pulso de disparo.

O termo “borda” se refere a transição de um sinal, ou seja, ao trecho da curva que representa a tensão em que ela varia entre o nível 0 e 1 (borda de subida), ou do nível 1 ao 0 (borda de descida). Também é comum usar o termo “frente” para indicar estas variações.

Assim, também chamamos “frente positiva” ao trecho em que ocorre a variação do sinal de 0 a 1 e fonte negativa ao trecho que temos a variação de 1 a 0.

Se a mudança de estado ou disparo (gatilhamento) ocorre quando o sinal de clock passa de 0 para 1, os flip-flops são denominados

“positive edge-triggered”(disparados pela borda ou frente positiva), enquanto que, se o disparo ocorre quando o clock vai do nível 1 para 0, ou seja, na queda do nível lógico, os flip-flops são chamados de “negative edge-triggered” (disparados pela borda ou frente negativa).

A maioria dos flip-flops deste tipo disponíveis na forma de circuito integrado é do tipo acionado pela transição negativa do sinal de clock.

Na figura 150 temos uma representação mais comum para este tipo de flip-flop, em que temos os circuitos internos, para o caso do sinal de clock ser aplicado diretamente ao flip-flop mestre (M).

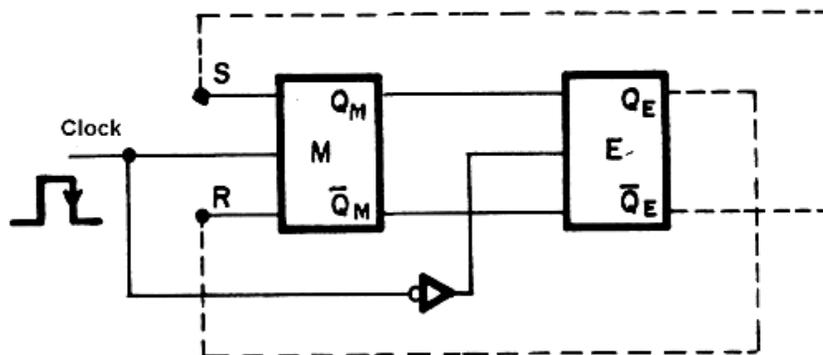


Figura 150 – Circuito interno comum em que os flip-flops têm símbolos próprios em lugar das portas NAND.

No entanto, na simbologia mais usual, temos a representação mostrada na figura 151 em que se leva em conta apenas as entradas e saídas.

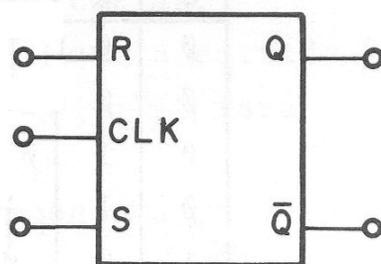


Figura 151 – Símbolo para o flip-flop R-S

Quando este tipo de flip-flop é usado num projeto de maior velocidade, é muito importante levar em conta os tempos em que todo o processo ocorre.

Assim, partindo do diagrama de tempos da figura 152, vemos que a saída do flip-flop só completa sua mudança de estado depois de certo tempo, a partir do instante em que o pulso de clock foi aplicado.

Sua ativação ocorre quando \overline{PR} estiver em 0 e \overline{CLR} em 1, no caso indicado, pois a / sobre a identificação indica que ela está ativa no nível baixo.

A outra entrada denominada CLEAR, ou apagamento, tem por função levar as saídas aos estados $Q=0$ e $\overline{Q}=1$ independentemente do que estiver ocorrendo nas demais entradas.

Na figura 154 temos a representação comum para este tipo de flip-flop.

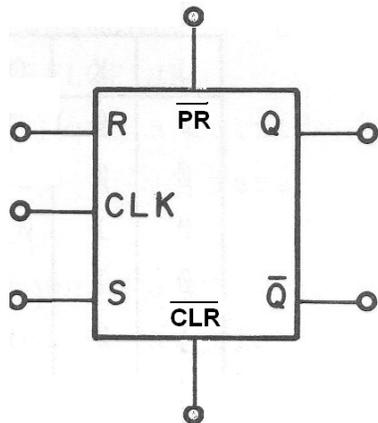


Figura 154 – Flip-Flop R-S Mestre-Escravo com Preset e Clear

É importante observar que estas duas entradas não podem ser ativadas ao mesmo tempo, pois isso levaria o circuito a um estado indeterminado que inclusive pode causar problemas aos seus componentes.

Já alertamos que é comum que em funções mais complexas, como os flip-flops, existam “situações proibidas”. Existem combinações de níveis lógicos que não podem estar presentes ao mesmo tempo em determinadas entradas, pois podem levar o circuito a um estado indefinido, ou seja, em que não é possível prever, o que vai ocorrer com a saída. Um exemplo simples é o de aplicarmos ao mesmo tempo níveis de disparo nas entradas Set e Reset de um flip-flop.

Voltando ao nosso exemplo, a tabela verdade para esse circuito nos mostra três novos símbolos que normalmente são usados em eletrônica digital.

X representa uma condição irrelevante, ou seja, qualquer que ela seja não haverá influência no que ocorre na saída.

A seta para cima indica a transição do nível baixo para o nível do sinal na entrada ou saída representadas enquanto que a seta apontando para baixo indica uma transição do nível alto para o nível do sinal correspondente.

Todos no mesmo estado

Conforme já salientamos, é importante que num circuito, quando ele seja ligado, ou em determinado instante em que ele seja solicitado, todos os flip-flops partam da mesma condição. Por exemplo, num contador, precisamos que o estado inicial seja 0000. Isso é garantido pelo uso das entradas Preset e Clear.

/CLR	/PR	R	S	CLK	Qn+1	/Qn+1
0	0	x	x	x	1	1
0	1	x	x	x	0	1
1	0	x	x	x	1	0
1	1	0	0	↓	Qn	/Qn
1	1	0	1	↓	1	0
1	1	1	0	↓	0	1
1	1	1	1	↓	1	1

X- não importa
 Qn e /Qn – não muda

6.4 - O flip-flop J-K Mestre-Escravo

Mesmo usando duas etapas, o flip-flop que vimos ainda pode apresentar alguns problemas de funcionamento quando usado em determinadas aplicações. Um aperfeiçoamento desse circuito é que veremos a seguir.

O flip-flop J-K mestre-escravo ou “master-slave”, como as demais funções que estudamos, pode ser implementado por funções lógicas comuns adquirindo a configuração básica mostrada na figura 155.

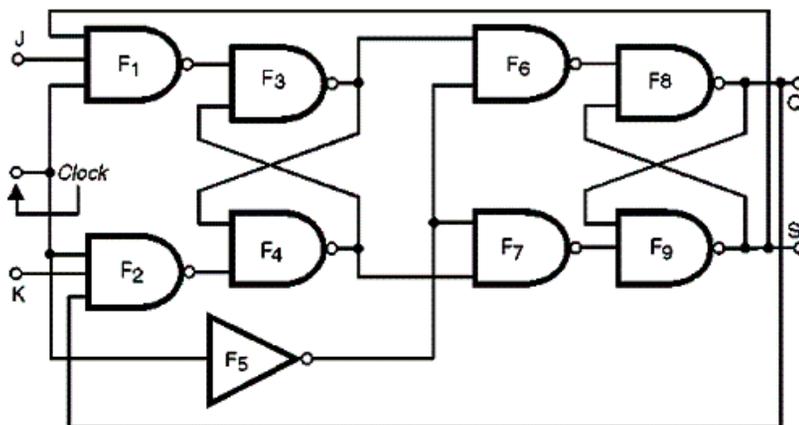


Figura 155- Flip flop J-K Mestre-Escravo

Um problema que observamos nos flip-flops R-S é que temos uma situação “proibida” que ocorre quando as entradas R e S vão ao nível alto ao mesmo tempo, o que pode levar o circuito a um estado indeterminado, conforme já alertamos.

Esta situação ocorre principalmente nas aplicações em computação e controle quando uma parte do sinal de saída é usada para realimentar a entrada. Nestas condições podem ocorrer as situações de conflito, com a produção de oscilações indesejadas e mesmo resultados imprevisíveis nos circuitos que devem ser excitados.

Esta situação pode ser contornada com a utilização de uma nova configuração que é justamente a do flip-flop J-K, e que analisaremos

a seguir. Esta configuração, por suas vantagens em determinados circuitos é a mais utilizada nas aplicações práticas.

Partimos então das entradas desse tipo de flip-flop. Podemos ter quatro combinações possíveis para os sinais aplicados nas entradas J e K, conforme mostra a tabela abaixo.

J	K
0	0
1	0
0	1
1	1

Analisemos cada uma delas:

a) $J = 0$ e $K = 0$

Quando a entrada de clock (CLK) passa por uma transição negativa, do sinal o flip-flop mantém sua condição original não mudando de estado.

b) $J = 1$ e $K = 0$

Quando a entrada de clock (CLK) passa por uma transição negativa, o flip-flop é “setado”. Se já estiver setado ele permanece nesta condição.

c) $J = 0$ e $K = 1$

Quando a entrada de clock (CLK) passa por uma transição negativa, o flip-flop é “ressetado”. Se já estiver nesta condição ele permanece.

d) $J = 1$ e $K = 1$

Nessa condição, ao receber uma transição negativa na entrada de clock (CLK), o flip-flop muda de estado (TOGGLE) (ver quadro). Se estiver setado ele resseta e se estiver ressetado ele é setado.

Podemos elaborar a tabela verdade da figura 156 para indicar o que ocorre com este flip-flop.

Toggle

Este é outro termo técnico muito usado em eletrônica digital e que, frequentemente, é mantido na forma inglesa. Ele significa “complementar” o estado, ou seja: as saídas que estiverem no nível 1 vão ao 0 e as que estiverem no 0 vão ao 1.

Zero cortado
 É comum em informática e eletrônica digital o uso do zero cortado para diferenciar da letra O.

CLK	J	K	Q_{n+1}	\bar{Q}_{n+1}	
↓	∅	∅	Q_n	\bar{Q}_n	Não muda
↓	∅	1	∅	1	Reseta
↓	1	∅	1	∅	Seta
↓	1	1	\bar{Q}_n	Q_n	Complementa (Toggle)

Figura 156 - Tabela verdade para o flip-flop J-K Mestre-Escravo

Observe o uso das setas para indicar as transições de sinal na entrada de clock, as quais comandam o funcionamento desse tipo de circuito.

Da mesma forma que nas outras configurações que estudamos, podemos também incluir as entradas de PRESET e CLEAR neste circuito, que ficará da maneira mostrada na figura 157.

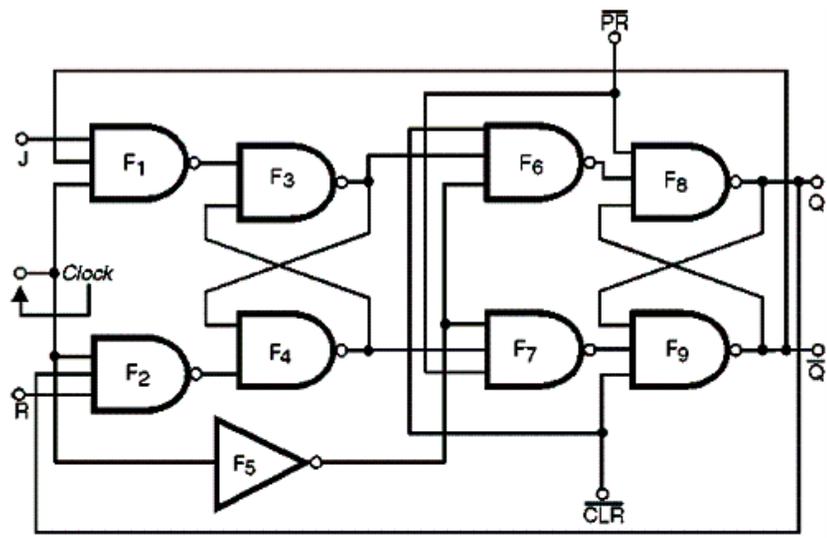


Figura 157 – Circuito final do flip-flop J-K Mestre-Escravo

Uma tabela verdade incluindo as entradas de PRESET (PR) e CLEAR (CLR) é mostrada abaixo.

$\overline{\text{CLR}}$	$\overline{\text{PR}}$	J	K	CLK	Q_{n+1}	\overline{Q}_{n+1}
\emptyset	\emptyset	X	X	X	1	1
\emptyset	1	X	X	X	\emptyset	1
1	\emptyset	X	X	X	1	\emptyset
1	1	\emptyset	\emptyset	↓	Q_n	\overline{Q}_n
1	1	\emptyset	1	↓	\emptyset	1
1	1	1	\emptyset	↓	1	\emptyset
1	1	1	1	↓	\overline{Q}_n	Q_n

Tabela verdade para o flip-flop J-K Mestre-escravo, incluindo as entradas Preset e Clear

Uma maneira melhor de analisarmos o funcionamento deste circuito é através de um diagrama de tempos, em que observamos as formas de onda nos diversos pontos de entrada e saída. Este diagrama de tempos para o flip-flop J-K é mostrado na figura 158.

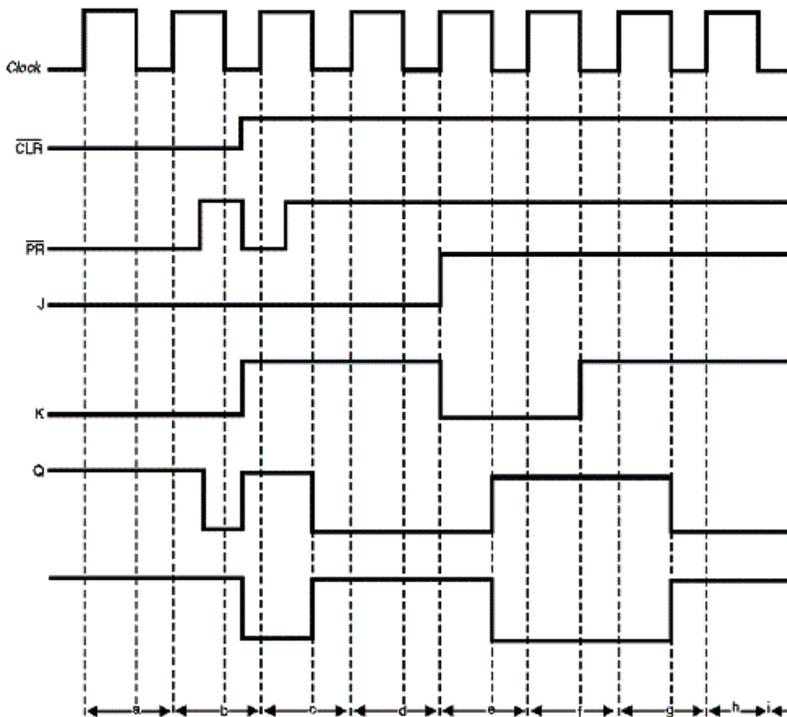


Figura 158 – Diagrama de tempos para o flip-flop J-K Mestre-Escravo com preset e clear

Analisemos alguns trechos importantes deste diagrama. mostrando o que acontece:

a) Nesse instante, CLR e PR estão no nível baixo, Q e /Q estão no nível alto, que é uma condição não permitida.

b) Aplica-se então o sinal PR que, indo ao nível alto, faz com que o flip-flop seja ressetado.

c) A aplicação de um pulso na entrada CLR que vai ao nível alto, e a ida de PR ao nível baixo, faz agora com que o flip-flop seja setado.

d) CLR e PR são mantidos no nível alto a partir deste instante. Com $J = 0$ neste trecho e K indo ao nível alto, o flip-flop será ressetado na próxima transição negativa do sinal de clock.

e) Ainda com CLR e PR no nível alto (esta condição se manterá daqui por diante), e a saída $J = 0$ e $k = 1$, o flip-flop permanece ressetado.

f) Com $J = 1$ e $K = 0$, o flip-flop é setado na transição seguinte do pulso de clock.

g) Com $J = 1$ e $K = 0$ não ocorrem mudanças de estado.

h) Com $J = 1$ e $K = 1$ na transição seguinte do pulso de clock o flip-flop muda de estado (complementa ou “toggle”). Se estiver ressetado, como neste caso, ele é setado.

i) Mantendo $j = 1$ e $K = 1$ com nova transição do pulso de clock, o flip-flop muda de estado outra vez, ou seja, complementa.

Veja que quando as entradas J e K estão no nível alto, o circuito se comporta como um disparador, mudando de estado a cada transição negativa do pulso de clock.

Conforme o leitor verá nos próximos capítulos, na maioria das aplicações digitais, o funcionamento é dinâmico, com os níveis de entrada e saídas das diversas funções mudando constantemente. Isso significa que, em muitas aplicações, a análise do circuito não se limita a saber o que ocorre com uma saída quando determinadas entradas apresentam determinados níveis.

Pode ser necessário analisar a dinâmica do funcionamento do circuito e, nesse caso, os diagramas de tempo são muito mais apropriados.

O símbolo usual para este tipo de flip-flop é o mostrado na figura 159.

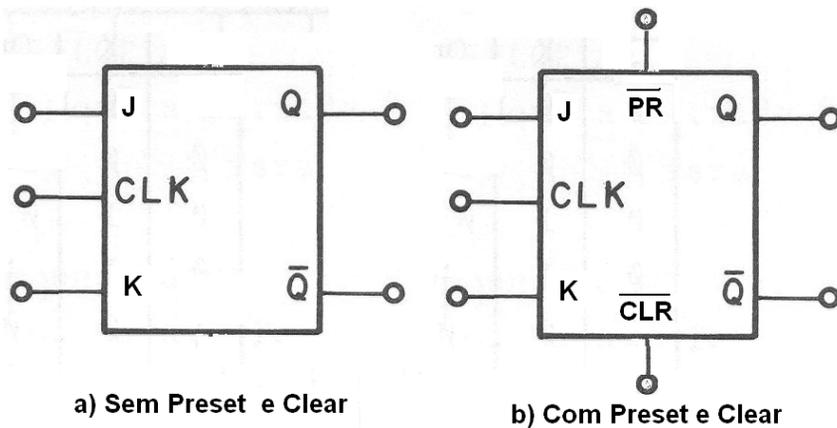


Figura 159 – Símbolos para os flip-flops tipo J-K

6.5 - O flip-flop tipo D

Os flip-flops tipo D também encontram uma vasta gama de aplicações práticas. Na figura 160 temos o símbolo adotado para representar esse tipo de flip-flop.

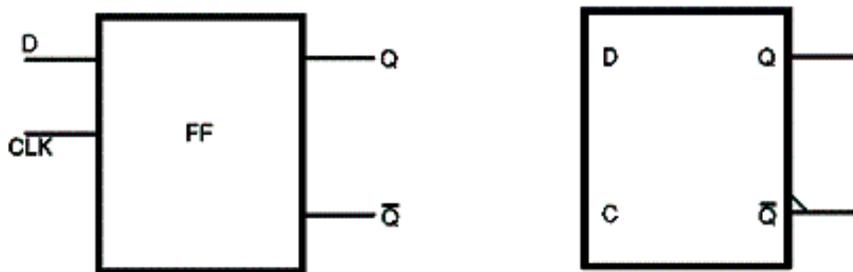


Figura 160 – Símbolos do flip-flop tipo D

O flip-flop tipo D possui uma única entrada que comanda todo o seu funcionamento. Esta entrada é que lhe dá nome. Denominada “Data” (dados), ela é abreviada por D e daí o nome do dispositivo.

Esse flip-flop opera de uma maneira muito simples: no pulso de clock ele assume o estado da entrada, conforme podemos ver pela sua tabela verdade:

D	Q _{n+1}
0	0
1	1

Pela sua simplicidade de funcionamento ele funciona muito bem em aplicações como divisores de frequência, contadores de pulsos, etc.

6.6 – Flip-flop tipo T

O nome deste flip-flop vem de “Toggle” (T) ou complementação, e seu símbolo é mostrado na figura 161.

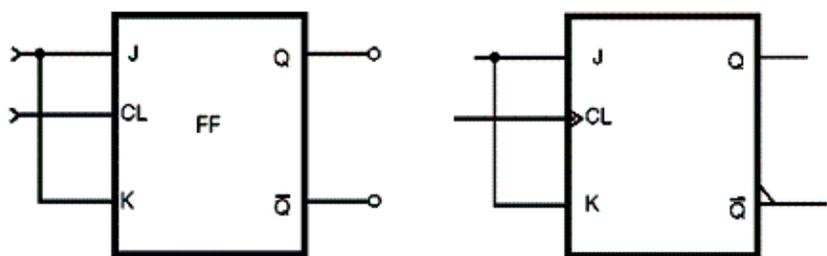


Figura 161 – O flip-flop tipo T

O que esse circuito faz pode ser entendido facilmente pelo diagrama de tempos mostrado na figura 162.

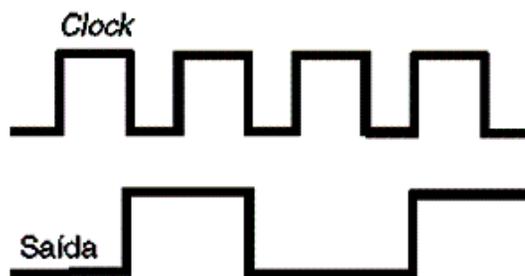


Figura 162 – Diagrama de tempos do flip-flop tipo T

Quando a entrada T desse circuito está no nível baixo, o flip-flop se mantém em seu estado anterior, mesmo com a aplicação do pulso de clock. No entanto, quando a entrada T está no nível alto, o flip-flop muda de estado. Se estava setado ele resseta, e se estava ressetado, ele seta.

Este comportamento significa na realidade a divisão da frequência do clock por dois. Em outras palavras, esse circuito se comporta como um divisor de frequência, encontrando aplicações práticas bastante importantes em eletrônica digital.

Um exemplo de aplicação é dado na figura 163 em que associamos diversos flip-flops do tipo T em série de modo que, passando por cada um a frequência do sinal de entrada, temos a sua divisão por 2.

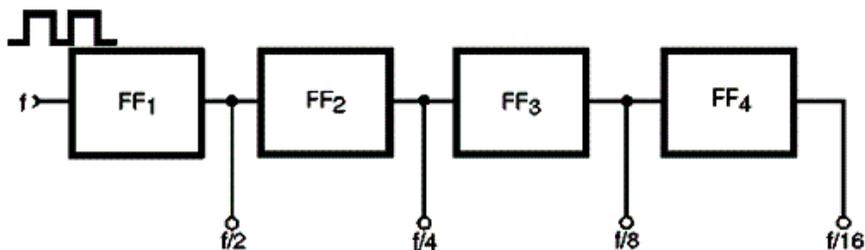


Figura 163 – Divisão de frequência com flip-flops

Usando 4 flip-flops, podemos dividir a frequência de um sinal de entrada por valores que sejam potências de 2, como: 2, 4, 8 e 16.

Este tipo de divisor de frequência é muito usado, a ponto de

existirem circuitos integrados que possuem seqüências de mais de 10 flip-flops ligados desta forma.

Na prática não temos os flip-flops tipo T como componentes prontos para uso. O que se pode fazer é obter este flip-flop a partir de outros e isso será visto no item seguinte.

6.7 – Transformando Flip-Flops

Da mesma maneira como podemos obter qualquer função lógica complexa a partir de funções simples, o que foi visto em lições anteriores, também podemos “brincar” com os flip-flops, obtendo outros tipos a partir de um tipo básico.

Assim, usando um flip-flops R-S ou J-K, que são comuns, e algumas portas lógicas, podemos obter flip-flops de outros tipos.

Na figura 164 temos algumas conversões que podem ser feitas utilizando-se flip-flops do tipo R-S.

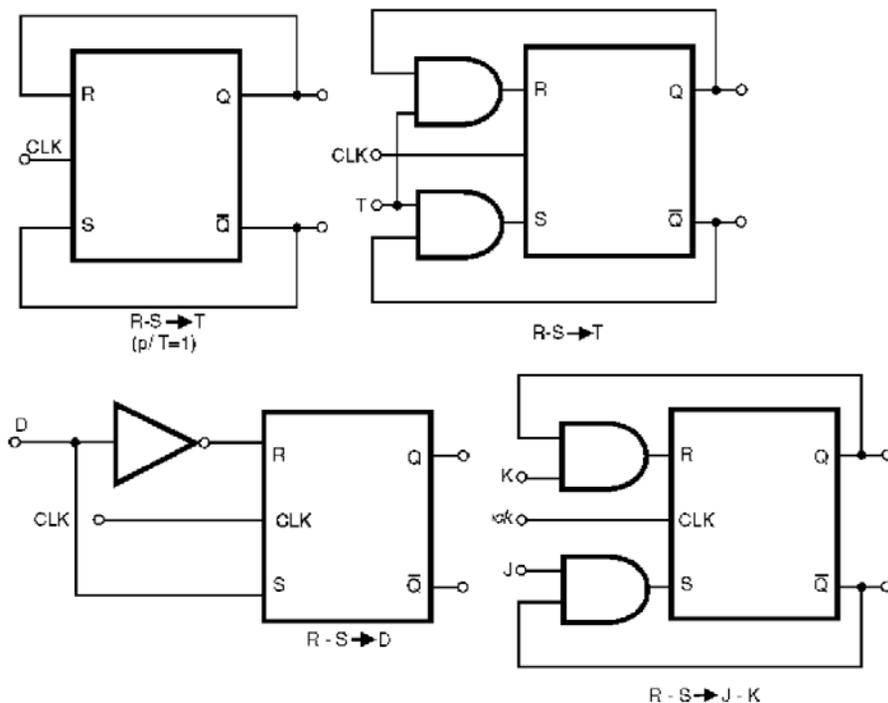


Figura 164 – Transformações de flip-flops

O modo de funcionamento de cada um pode ser facilmente entendido se o leitor tentar associar as tabelas verdades dos flip-flops que foram estudados nesta lição às tabelas verdade das portas agregadas, considerando os sinais de realimentação.

Assim, começamos pela figura 165, onde temos o modo de se obter flip-flops tipo D e T a partir de flip-flops do tipo J-K.

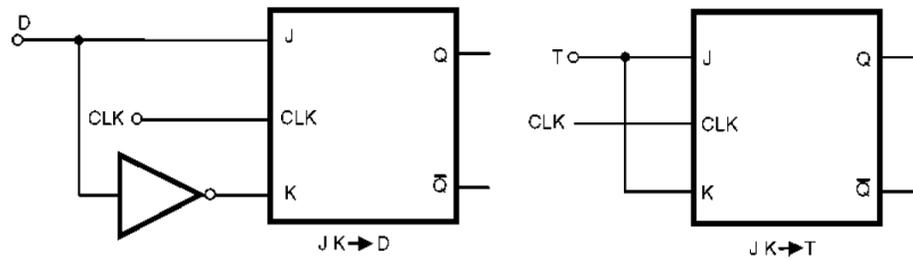


Figura 165 – Obtendo flip-flops tipo T e D a partir de flip-flops tipo J-K

Veja que a simples conexão da entrada K ao J no flip-flop do tipo J-K o transforma em um flip-flop tipo T. Esta possibilidade é muito interessante, já que flip-flops J-K são disponíveis em tecnologia tanto TTL como CMOS e podem ser usados em circuitos divisores de frequência.

Quando falarmos dos tipos específicos de circuitos integrados de flip-flops das famílias TTL e CMOS, o leitor terá informações sobre diversos tipos de flip-flops que podem ser usados em projetos práticos.

Na verdade, já utilizamos esta configuração em diversos de nossos projetos práticos publicados em livros, revistas e no nosso site.

Finalmente temos outras duas transformações importantes de flip-flops que são mostradas na figura 166.

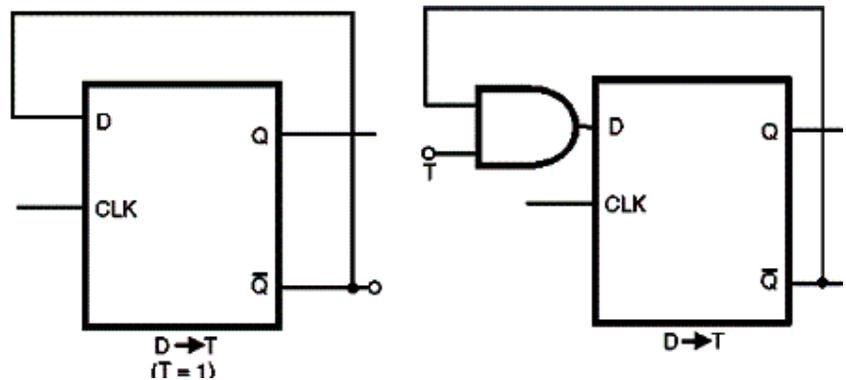


Figura 166 - Transformando flip-flops tipo D em T

No primeiro caso temos uma transformação de um flip-flop tipo D em flip-flop tipo T, bastando para isso que a saída complementar /Q seja ligada à entrada D, realimentando o circuito.

A segunda transformação, que leva um flip-flop tipo D a funcionar como tipo T, exige o emprego de uma porta AND adicional na realimentação do sinal que é retirado da saída complementar /Q.

Veja que é muito importante saber transformar flip-flops, assim como as funções lógicas. Muitos circuitos integrados possuem mais de um flip-flop disponível no mesmo invólucro. Assim, se precisarmos de um flip-flop do tipo D e de um tipo T, podemos usar um circuito que tenha dois do tipo D e transformar um deles e um do tipo T.

O resultado é que, com um componente apenas, obtemos as duas funções, com economia de custo e espaço na placa de circuito impresso.

6.8 – Nos equipamentos digitais

Encontramos os flip-flops nos computadores, equipamentos de telecomunicações, controles industriais, equipamentos de eletrônica embarcada, robôs e muitos outros, como elementos fundamentais de muitos circuitos.

Uma das aplicações mais importantes do flip-flop é na divisão de frequência de clocks. Conforme o leitor sabe, existem setores de um equipamento digital que devem operar com velocidades menores que a fornecida pelo clock principal.

É o caso dos barramentos de microcontroladores e microprocessadores e controles diversos onde são ligadas as placas de expansão, aquisição de dados e controle, modems, as saídas de dados das portas paralela e serial, etc.

Assim, em lugar de se usar um clock para cada frequência desejada, o que se faz, empregar um clock único e dividir sua frequência conforme as exigências de frequências mais baixas, conforme mostra a figura 167.

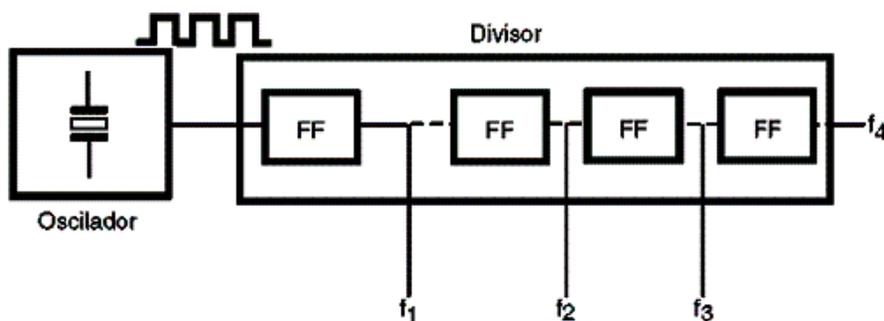


Figura 167- Obtendo diversas frequências de um oscilador único

No caso de muitos equipamentos digitais, tanto o próprio clock como a sequência de flip-flops divisores pode ser obtida num único circuito integrado.

Um ponto importante que deve ser levado em conta, e que estudaremos nos capítulos futuros, é que podemos ligar os flip-flops em conjunto com outras funções, de modo que a frequência possa ser dividida por qualquer número, e não somente por potências de 2 como 2, 4, 8, 16, 32, 64, 128, etc. Como isso é feito, será estudado mais adiante.

Outra aplicação importante é como célula de memória. Oito flip-flops ligados lado a lado podem armazenar um byte inteiro. Cada flip-flop armazena um bit.

Existem diversas memórias internas, usadas em equipamentos

Flip-flops prontos

Podemos encontrar praticamente todos os tipos de flip-flops que vimos na forma de circuitos integrados, tanto em tecnologia CMOS como TTL.

Memórias

As memórias formadas por grande quantidade de flip-flops organizados de tal forma que cada um armazene um bit são comuns em muitas aplicações. Os flip-flops nestes casos, tanto podem usar configurações com transistores comuns como com FETs.

digitais que nada mais do que flip-flops organizados em matrizes, contendo grande quantidade deles, e que podem ser habilitados tanto para a leitura de dados como para introdução (gravação de dados).

Existem ainda muitas outras funções importantes que podem ser implementadas a partir de flip-flops e que serão estudadas nos capítulos seguintes.

6.9. Os flip-flops antigos

A configuração do flip-flop não é nova. Na verdade foi em 1919 que dois pesquisadores americanos chamados Eccles e Jordan apresentaram o primeiro circuito de flip-flop usando válvulas, conforme mostra a figura 168.

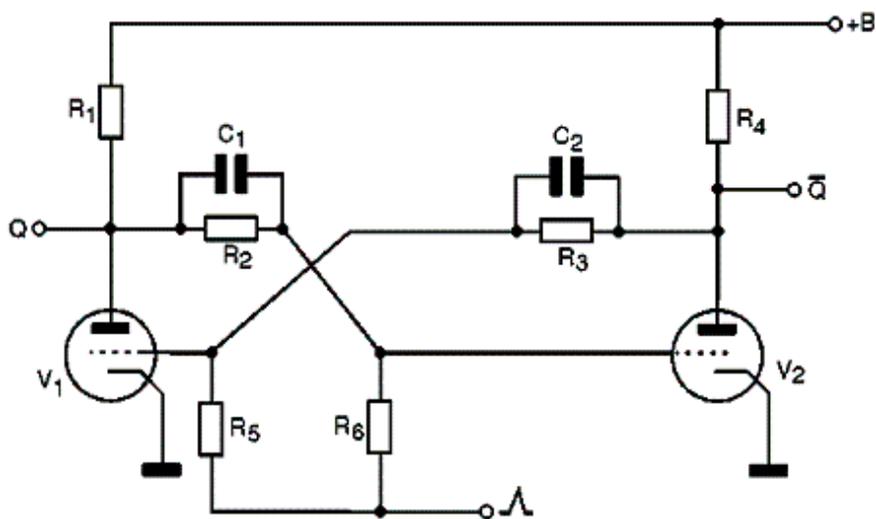


Figura 168 – Um flip-flop valvulado

Por este motivo, muitos profissionais “da velha guarda” ainda chamam os flip-flops de “Basculas ou Circuitos Eccles-Jordan” quando referem-se a esta configuração.

Em 1930 os físicos já usavam estes circuitos ligados em série para dividir a contagem dos pulsos de contadores Geiger de radiação de modo a obterem valores menores mais facilmente totalizados nas pesquisas.

Um contador binário usando uma lâmpada neon ligada às válvulas foi desenvolvido usando estes flip-flops já em 1940, mas, foi somente depois desta época que os primeiros computadores digitais passaram a usar estes circuitos de uma forma mais intensa, até o advento do transistor e depois dos circuitos integrados.

Eccles-Jordan

William Henry Eccles
(1875-1968)

Físico especialista em telecomunicações que juntamente com F.W. Jordan inventou o flip-flop ou bascula com válvulas.

Termos em inglês:

Mais termos desta lição são usados na forma original em inglês, mesmo em documentação em português. No entanto, temos alguns que possuem tradução e merecem destaque:

Reset – rearmar, ressetar
Set – armar, setar
Clear – limpar
Preset – pré-armar
Latch – trava
Master – mestre
Slave - escravo
Edge – borda
Trigger – disparar
Triggered- disparado
Toggle - complementa
Not allowed – não permitida

Termos para pesquisa:

- Básculas
- Flip-flops
- Eccles-Jordan
- Astável
- Biestável
- Memórias

QUESTIONÁRIO

1. Os elementos biestáveis de um circuito:
 - a) Possuem apenas um estado estável
 - b) Possuem dois estados instáveis
 - c) Possuem dois estados estáveis
 - d) Possuem um número indeterminado de estado estáveis

2. Usado como elemento de memória, um flip-flop pode armazenar:
 - a) 1 bit
 - b) 1 byte
 - c) meio byte
 - d) 2 bits

3. Um flip-flop R-S “setado” apresenta que níveis lógicos em suas saídas?
 - a) $Q=0$ e $\bar{Q}=0$
 - b) $Q=0$ e $\bar{Q}=1$
 - c) $Q=1$ e $\bar{Q}=0$
 - d) $Q=1$ e $\bar{Q}=1$

4. Os flip-flops “negative edge-triggered” mudam de estado quando:
 - a) O pulso clock vai do nível baixo para o nível alto
 - b) O pulso de clock vai do nível alto para o nível baixo
 - c) O pulso de clock estabiliza no nível baixo
 - d) O pulso de clock estabiliza no nível alto

5. Para que um flip-flop J-K Mestre escravo tenha a condição “toggle” quais são os níveis lógicos que devem ser colocados na entrada J e K?
 - a) $J = 0$ e $K = 0$
 - b) $J = 0$ e $K = 1$
 - c) $J = 1$ e $K = 0$
 - d) $J = 1$ e $K = 1$

6. Quatro flip-flops do tipo T ligados um após o outro (em série) recebem uma frequência de 1 600 Hz na sua entrada. Qual é a frequência obtida na saída do último flip-flop (o sinal deve ser retangular).
 - a) 800 Hz
 - b) 400 Hz
 - c) 200 Hz
 - d) 100 Hz



» Os Flip-Flops e Funções Integradas em Circuitos Integrados

No capítulo anterior aprendemos como funcionam os principais tipos de flip-flops, verificando que, dependendo dos recursos que cada um tem, eles podem ser empregados de diversas formas. Também vimos as entradas que estes dispositivos podem conter para poder melhorar seu desempenho em determinadas aplicações como, por exemplo, nos computadores, equipamentos industriais, controle, instrumentação, automação industrial, telecomunicações e mecatrônica. Também estudamos nos primeiros capítulos do curso as funções lógicas que são usadas em diversos circuitos. Tudo isso nos leva a necessidade de contarmos com estas funções na forma de circuitos integrados. De fato, existem muitos circuitos integrados, tanto TTL como CMOS, contendo flip-flops dos tipos estudados além de todas as funções lógicas (portas e inversores e amplificadores) e até mesmo alguns contendo funções especiais de grande interesse. Será justamente deles que falaremos neste capítulo. Este capítulo consta então dos seguintes itens:

Itens:

- 7.1 – Os flip-flops TTL
- 7.2 - Os flip-flops CMOS
- 7.3 – Funções lógicas TTL
- 7.4 – Funções lógicas CMOS
- 7.5 - A Função Tri-state expansível do 4048

Objetivos:

- Conhecer os flip-flops principais da família TTL.
- Conhecer os flip-flops principais da família CMOS.
- Saber quais são as funções lógicas disponíveis nas famílias TTL e CMOS.
- Analisar a função tri-state do 4048 mostrando como ela pode ser usada.

7.1 - Os flip-flops TTL

A família de circuitos integrados digitais TTL, e também as suas diversas sub-famílias, contam com uma grande quantidade de flip-flops que podem ser usados numa infinidade de aplicações práticas.

A diferença de cada tipo de circuito integrado não está apenas no tipo de flip-flop que contém como também nos seus recursos e na sua quantidade.

Também devemos observar que um fator importante na escolha de um flip-flop para uma determinada aplicação, é a sua velocidade. Para as diversas famílias TTL podemos especificar as máximas velocidades médias dos seus flip-flops da seguinte forma:

- Standard (74) - 35 MHz
- Low Power (74L) - 3 MHz
- Low Power Shottky (74LS) - 45 MHz
- High Speed (74H) - 50 MHz
- Shottky (74S) - 125 MHz

Lembramos os leitores que, para usar bem os flip-flops TTL, é preciso observar alguns cuidados como, por exemplo, manter sempre as entradas CLEAR e PRESET em níveis definidos. Deixando estas entradas abertas podem ocorrer instabilidades de funcionamento.

O nível em que elas devem ser deixadas, ou seja, sua conexão no Vcc ou 0 V depende da aplicação.

Também é importante observar que, à medida que as velocidades de operação se tornam maiores, mais crítico se torna o projeto das placas de circuito impresso em que estes circuitos integrados são usados.

Técnicas especiais de desacoplamento da alimentação, posicionamento das trilhas devem ser conhecidas por todos que pretendem trabalhar com tais circuitos integrados.

Analisemos as características de alguns dos flip-flops TTL mais usados nos projetos práticos:

a) 7473 – Duplo flip-flop J-K com Clear

Num único invólucro de 14 pinos Dual in Line temos 2 flip-flops do tipo J-K com entrada de Clear. A pinagem deste circuito integrado é mostrada na figura 169. Também existem versões do mesmo circuito integrado para montagem em superfície.

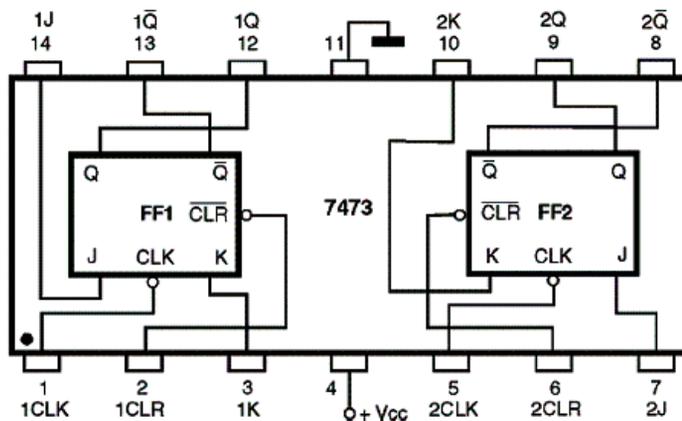


Figura 169 – Pinagem do 7473

Os flip-flops desse CI são sensíveis ao nível de clock (Level Triggered), com entrada de Clear assíncrono. O funcionamento dos flip-flops deste circuito integrado pode melhor ser entendido pela tabela-verdade dada na figura 170.

$\overline{\text{CLR}}$	CLK	J	K	Q_{n+1}	\overline{Q}_{n+1}
\emptyset	X	\emptyset	\emptyset	\emptyset	1
1		\emptyset	\emptyset	Q_n	\overline{Q}_n
1		\emptyset	1	\emptyset	1
1		1	\emptyset	1	\emptyset
1		1	1	\overline{Q}_n	Q_n

X = não importa

Figura 170 – Tabela verdade do 7473

Nesta tabela, o símbolo que tem a forma de um pulso de sinal representa um pulso de clock positivo aplicado à entrada correspondente. Esse tipo de representação é muito comum em muitos manuais técnicos. O leitor deve se familiarizar com seu significado.

Observe que quando J e K estão aterradas o clock não tem efeito sobre o circuito. Na operação normal, a entrada Clear deve ser mantida no nível alto. Se a entrada Clear for aterrada o flip-flop resseta.

A frequência máxima de operação destes flip-flops, para a série normal, é de 20 MHz com um consumo por circuito integrado da ordem de 20 mA.

b) 7474 – Duplo flip-flop tipo D com Preset e Clear

Os flip-flops contidos no invólucro DIL de 14 pinos deste circuito integrado disparam com a transição positiva do sinal de clock (Positive-Edge Triggered). A pinagem deste circuito integrado é mostrada na figura 171.

Notação

Conforme já explicamos, logo no início do curso, existem diversas notações adotadas para indicar terminais e características dos circuitos integrados digitais. Assim, neste caso vemos que as saídas podem tanto ser indicadas por Q_n e \overline{Q}_n , Q_{n+1} e \overline{Q}_{n+1} como ainda Q e \overline{Q} . Adotaremos todas, pois conforme a figura ou tabela for obtida de manuais de fabricantes diferentes, as notações também podem variar.

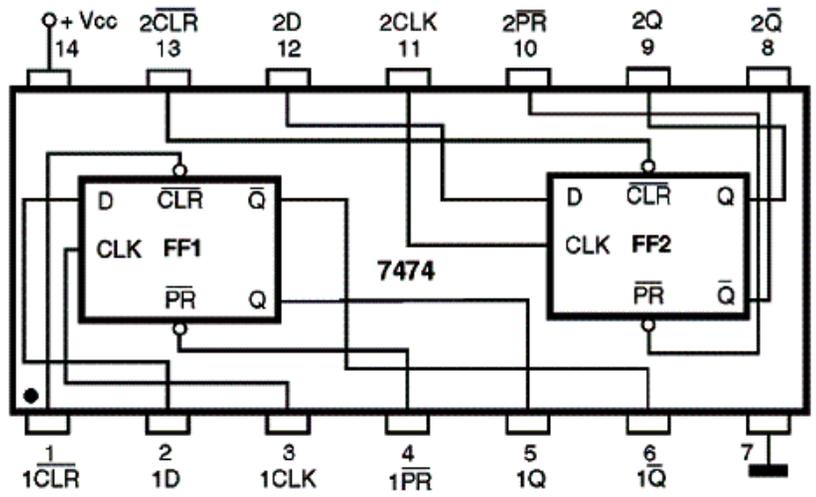


Figura 171 – Duplo flip-flop tipo D TTL 7474

Também podemos encontrar versões para montagem em superfície, com a mesma pinagem.

A tabela verdade que mostra o funcionamento dos flip-flops deste circuito integrado é dada na figura 172.

$\overline{\text{CLR}}$	$\overline{\text{PR}}$	D	CLK	Q_{n+1}	\overline{Q}_{n+1}
\emptyset	\emptyset	X	X	1	1
\emptyset	1	X	X	\emptyset	1
1	\emptyset	X	X	1	\emptyset
1	1	\emptyset	\uparrow	\emptyset	1
1	1	1	\uparrow	1	\emptyset

(*) = não permitido

x = não importa

Figura 172 – Tabela verdade para o 7474

Pela tabela, vemos que a condição em que as entradas Clear e Preset estão simultaneamente ativas não deve ser usada, pois teremos uma condição não permitida para os flip-flops.

Esse é um dos casos que citamos, em que existem combinações de entrada proibidas e que, portanto, devem ser evitadas.

A frequência máxima de operação deste circuito integrado, na versão standard, é de 25 MHz e o consumo é da ordem de 17 mA.

Lembramos que existem outras versões que são mais rápidas, as quais foram estudadas nos capítulos anteriores. Também é importante observar que todos os componentes desta linha podem ser encontrados em invólucros SMD. Na figura 173 o 74LTX74MTC da Fairchild em invólucro SMD.



Figura 173 – Versão do 7474 em invólucro SMD

c) 7475 – Quatro latches tipo D

Os latches biestáveis, conforme já explicamos, são como chaves que armazenam uma informação digital presente em sua entrada. A aplicação mais comum é justamente como memória o que significa que cada circuito integrado 7475 pode armazenar 4 bits de informação.

A pinagem deste circuito integrado é mostrada na figura 174.

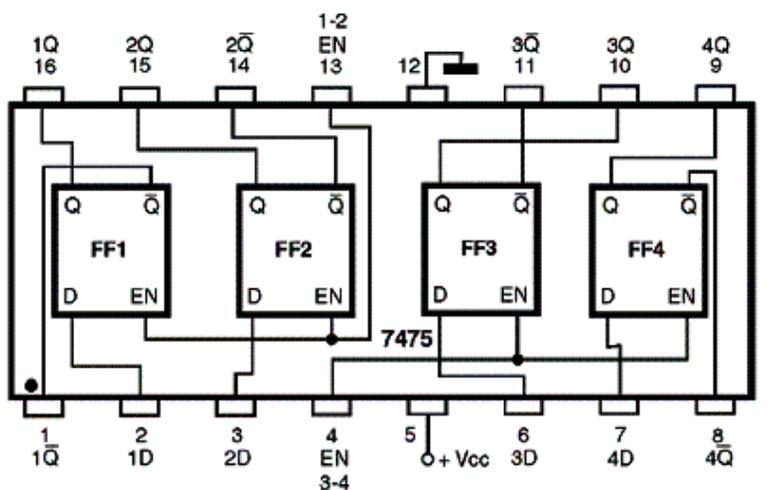


Figura 174 – 7575 – Quatro latches tipo D (flip-flops tipo D)

Quando o circuito é habilitado, o que é conseguido levando-se a linha “ENABLE” ao nível alto, as saídas Q e /Q seguem a entrada D.

Os latches contidos nesse circuito integrado são do tipo “transparente”, o que significa que, se as entradas forem modificadas, as saídas também se alteram no mesmo instante.

Quando a entrada “ENABLE” for levada ao nível baixo, as saídas não respondem aos sinais de entrada D.

Veja que o LATCH armazena a informação que estava na entrada D imediatamente antes da ocorrência de uma transição do nível alto para o nível baixo da linha de habilitação (Nível 1 para o nível 0).

O funcionamento de cada flip-flop do 7475 pode ser colocado na tabela verdade da figura 175.

Independência

Nos circuitos integrados que possuam mais de uma função, como portas, flip-flops, etc., normalmente elas são independentes tendo apenas a alimentação em comum.

EN	D	Q _{n+1}	\bar{Q}_{n+1}
∅	X	Q _n	\bar{Q}_n
1	∅	∅	1
1	1	1	∅

Figura 175 – Tabela verdade para 7475

Este circuito integrado não serve para aplicações onde se deseja mudanças de estado a cada pulso de clock. Dizemos que este circuito não pode ser usado como um registrador de deslocamento (shift-register), função que será estudada nos próximos capítulos do volume 2.

O tempo de propagação do sinal é da ordem de 24 ns e o consumo típico por circuito integrado é de 32 mA, para os componentes da série normal ou standard.

d) 7476 – Dois flip-flops J-K com Preset e Clear

Os dois flip-flops deste circuito integrado têm funcionamento independente e disparam com a mudança do nível do sinal de clock (level triggered).

O invólucro é DIL de 16 pinos conforme mostra a figura 176, mas existem versões com invólucros para montagem em superfície (SMD).

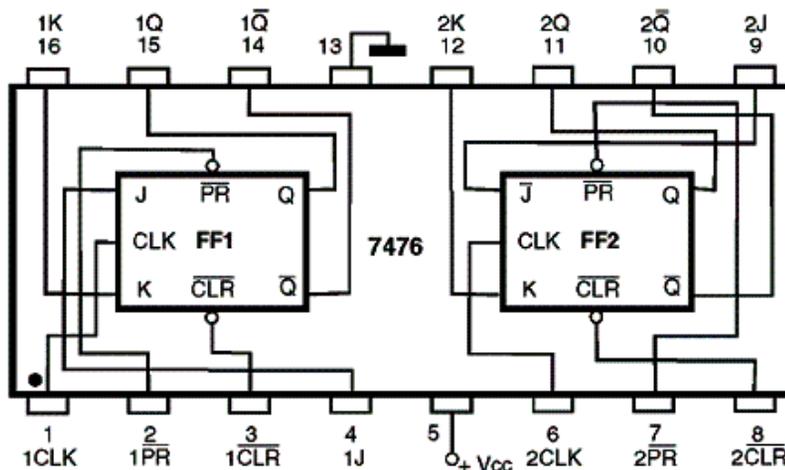


Figura 176 – 7476 – Dois flip-flops J-K com Preset e Clear

O funcionamento de cada um dos flip-flops pode ser melhor analisado através da tabela verdade que é mostrada na figura 177.

CLR	PR	J	K	CLK	Q _{n+1}	Q̄ _{n+1}
∅	∅	X	X	X	1	1
∅	1	X	X	X	∅	1
1	∅	X	X	X	1	∅
1	1	∅	∅		Q _n	Q̄ _n
1	1	∅	1		∅	1
1	1	1	∅		1	∅
1	1	1	1		Q̄ _n	Q _n

(*) não permitido X = não importa

Figura 177- Tabela verdade para o 7476

Observe o símbolo adotado para representar um pulso de clock.

Da mesma forma que nos demais circuitos integrados desta série, as entradas CLEAR E PRESET devem ser mantidas em níveis lógicos definidos para que não ocorra o funcionamento errático do circuito.

As condições em que temos ambas as saídas no nível 1 são condições proibidas, conforme já vimos devendo ser evitadas no uso do componente.

Também observamos pela tabela verdade que não se pode ativar as duas entradas de CLOCK E CLEAR ao mesmo tempo, pois isso levaria os flip-flops a uma condição não permitida. Trata-se de outra condição proibida para o componente indicado.

Um ponto interessante, que deve ser observado neste circuito integrado, é a pinagem diferente, já que normalmente nos circuitos desta série a alimentação positiva é sempre nos pino 14 ou 16 e a negativa no de pino 7 ou 8 quando os invólucros são de 14 ou 16 pinos.

Na verdade, o projetista deve estar atento para este fato, pois não é apenas esse componente que não segue essa regra. Outros componentes podem ter alimentações feitas por pinos diferentes.

A frequência máxima de operação destes flip-flops para a série normal é de 20 MHz e o consumo é de 20 mA.

e) 74174 – Seis flip-flops tipo D com Clear

Este circuito integrado contém seis flip-flops do tipo D que são disparados na transição positiva do sinal de clock. A entrada de CLEAR é comum a todos os flip-flops. O invólucro é de 16 pinos com a identificação feita segundo mostra a figura 178.

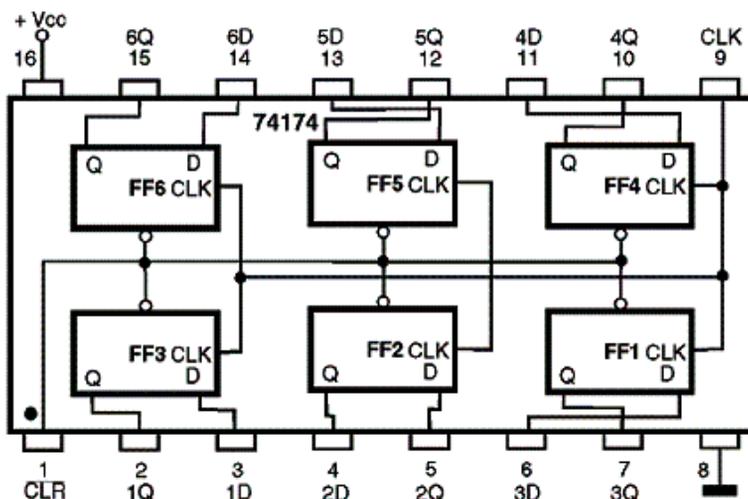


Figura 178 – Pinagem do 74174 – Seis flip-flops do tipo D com Clear

A tabela verdade que descreve o funcionamento de cada flip-flop deste circuito integrado é mostrada na figura 179.

CLR	D	CLK	Q _{n+1}
∅	X	X	∅
1	∅	↑	∅
1	1	↑	1
1	X	∅	Q _n

X = não importa

Figura 179 – Tabela verdade para 74174

Observe que nestes flip-flops temos acesso a apenas uma das saídas, o que significa que as saídas complementares, não podem ser usadas.

Na verdade, a maioria das aplicações em que são usados flip-flops aproveita apenas uma das saídas. Assim, não há nesses casos a necessidade de termos acesso à saída complementar. Este acesso, sendo eliminado, pode ajudar não só na integração de maior número de flip-flops no mesmo invólucro, como também no uso de menos pinos.

A frequência máxima de cada flip-flop da série standard (comum) é de 35 MHz com um consumo típico de 45 mA por circuito integrado.

f) 74273 – Oito flip-flops tipo D com Clear

Este circuito é semelhante ao anterior, com a diferença de que existem oito em lugar de seis flip-flops tipo D. Como cada flip-flop pode operar com um bit, esta configuração se torna ideal para aplicações em computadores já que pode operar com 8 bits que correspondem a um byte.

A pinagem do circuito integrado 74273 é mostrada na figura 180.

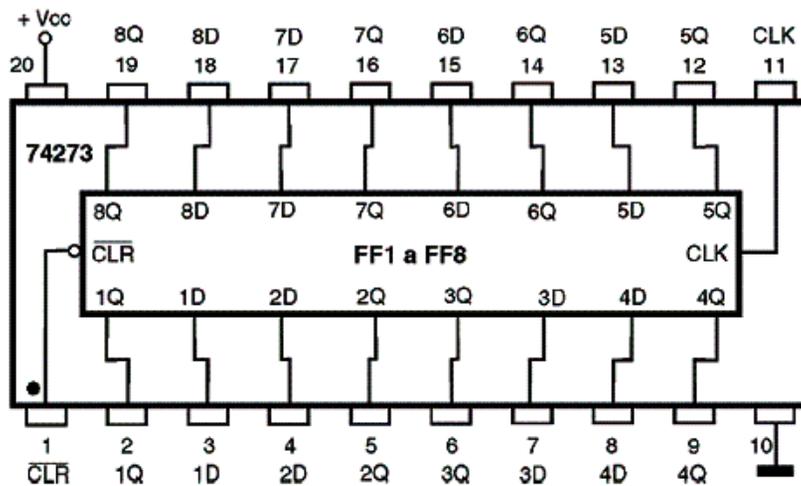


Figura 180 – 74273 – Oito flip-flops tipo D com Clear

A tabela verdade para cada flip-flop é a mesma do circuito integrado anterior e que foi mostrada na figura 180.

A frequência máxima de operação para os circuitos integrados deste tipo da série normal é de 30 MHz, com um consumo de 62 mA para cada um.

Veja que o invólucro usado é Dual In Line (DIL) de 20 pinos e que a entrada de CLEAR é comum a todos os integrados. Em outras palavras, um pulso na entrada de CLEAR, zera todos os flip-flops deste circuito integrado ao mesmo tempo.

Também observamos que não existe acesso às saídas complementares de cada flip-flop, e que podem ser obtidas versões para SMD.

g) 74LS373 – Latch octal transparente tipo D

O tipo LS (Low Power Schottky) é importante em certas aplicações, já que se trata de circuito compatível com as portas paralelas dos computadores, e também com microprocessadores usados em controles e automação industrial. Isso significa que ele pode ser excitado diretamente pelos níveis lógicos existentes num PC ou num microprocessador.

Como cada circuito integrado 74LS373 contém 8 latches com saídas tri-state, ele pode ser usado para trabalhar com um byte inteiro, sem problemas. Cada flip-flop armazena um bit.

A pinagem deste circuito integrado é mostrada na figura 181.

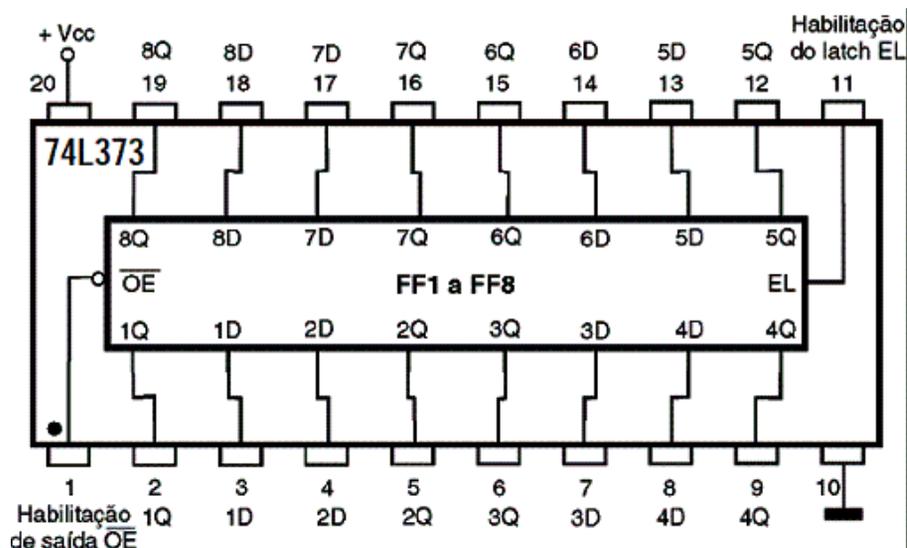


Figura 181 – 74L373 – Latch octal transparente tipo D

Quando a entrada /OE está no nível alto (1) as saídas de todos os flip-flops vão para o estado de alta impedância. Isso significa que estas saídas podem ser ligadas a um barramento comum a outros circuitos integrados sem o problema de conflitos que podem carregar os circuitos causando problemas de funcionamento, conforme já estudamos nos capítulos iniciais deste curso.

A precaução principal que se deve tomar quando estas saídas são ligadas a um barramento, é que elas não sejam inadvertidamente ativadas quando o barramento estiver transmitindo sinais, pois podem ocorrer conflitos de níveis lógicos.

Quando a entrada /OE está ativada, o que é feito levando-a ao nível baixo (0), o estado das saídas vai depender da entrada EL. Se EL for colocada no nível alto (1), o latch estará aberto e com isso ele se encontra no estado “transparente”. O que estiver na entrada D vai passar pelo circuito e aparecer na saída Q.

Se EL estiver no nível baixo (0) a saída Q não mais responde ao que ocorre nas entradas D. Nestas condições dizemos que o latch está fechado e a saída Q será o conteúdo das entradas D que foi armazenado imediatamente antes da transição das entradas EL do nível alto para o nível baixo.

Em outras palavras, podemos dizer que os flip-flops são gatilhados na transição negativa da entrada EL armazenando a informação que se lhes seja aplicada à entrada.

A tabela verdade da figura 182 mostra o que ocorre em cada flip-flop deste circuito integrado.

\overline{OE}	D	CLK	Q_{n+1}
1	X	X	alta-Z
\emptyset	\emptyset	\uparrow	\emptyset
\emptyset	1	\uparrow	1
\emptyset	X	1	Q_n

Figura 182 – Tabela verdade para o 74L373

Observe a condição de alta impedância obtida com /OE no nível alto.

A frequência máxima de operação para os latches deste circuito integrado é de 50 MHz com um consumo de 24 mA.

h) 74LS374 – Oito flip-flops tipo D com saídas Tri-State

O circuito integrado TTL 74LS374, apresentado em invólucro DIL de 20 pinos, contém 8 flip-flops do tipo D que são disparados na transição positiva do sinal de clock. As saídas são tri-state e a pinagem é mostrada na figura 183.

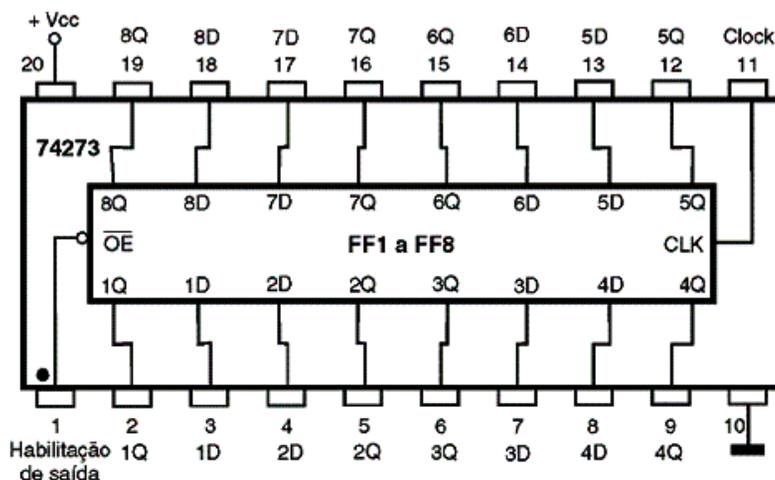


Figura 183 – 74LS374 – Oito flip-flops tipo D com saídas Tri-State

Quando a entrada /OE está no nível alto, as saídas de todos os flip-flops vão para o estado de alta impedância.

Veja que neste circuito integrado, também não temos acesso as saídas complementares de cada flop-flop.

A tabela verdade que descreve o funcionamento de cada flip-flop, é mostrada na figura 184.

**Entrada e saídas
não usadas**

Veja nos capítulos anteriores os procedimentos comuns para se evitar que ruídos ocorram pela captação de sinais pelas entradas deixadas em aberto.

\overline{OE}	D	CLK	Q_{n+1}
1	X	X	alta-Z
\emptyset	\emptyset	\uparrow	\emptyset
\emptyset	1	\uparrow	1
\emptyset	X	1	Q_n

Figura 184 – Tabela verdade do 74LS374

A frequência máxima de operação deste circuito integrado é de 50 MHz, com um consumo típico de 27 mA.

Observe que temos um clock comum para todos os flip-flops.

7.2 – Os flip-flops CMOS

A família de circuitos integrados CMOS também possui uma grande quantidade de circuitos integrados contendo flip-flops.

Se bem que os princípios de funcionamento destes flip-flops sejam os mesmos, as características elétricas são diferentes. Assim, tensão de alimentação, sensibilidade aos ruídos, velocidade, etc. são diferentes, o que deve ser levado em conta em qualquer projeto.

Uma recomendação importante relativa ao uso destes flip-flops, assim como das demais funções CMOS, é que as entradas não usadas, pela sua sensibilidade devida a alta impedância nunca devem ser mantidas abertas quando houver possibilidade de captação de ruídos.

Nos flip-flops CMOS, diferentemente dos TTL, as entradas assíncronas são ativadas no nível alto, o que significa que devem ser mantidas no nível baixo para a operação normal.

Isso significa que, nas aplicações práticas os pinos correspondentes devem ser aterrados.

a) 4013 – Dois flip-flops tipo D com Preset e Clear

O circuito 4013 é um dos mais populares da série CMOS, sendo empregado numa infinidade de projetos práticos.

Os dois flip-flops contidos neste circuito integrado são disparados na transição positiva do sinal de clock. O invólucro é o DIL de 14 pinos mostrado na figura 185.

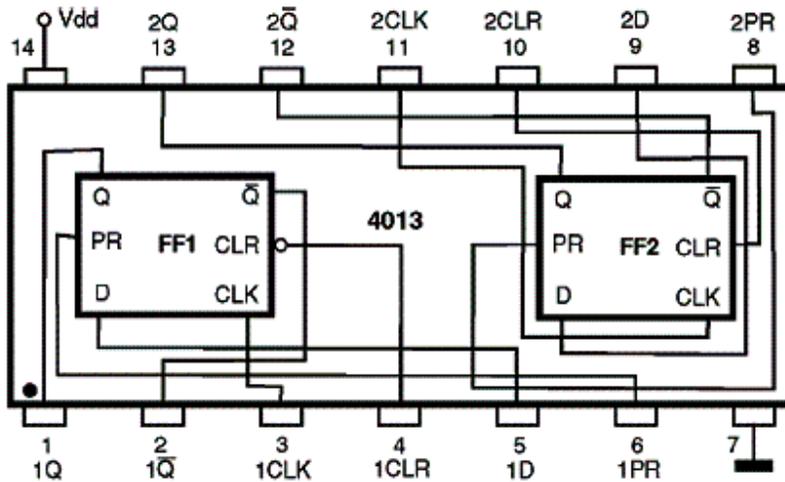


Figura 185 – 4013 – Dois flip-flops tipo D

A tabela verdade para este circuito integrado é mostrada na figura 186.

CLR	PR	D	CLK	Q _{n+1}	Q̄ _{n+1}
∅	∅	∅	↑	∅	1
∅	∅	1	↑	1	∅
1	∅	X	X	∅	1
∅	1	X	X	1	∅
1	1	X	X	1	1

X = não importa

Figura 186 – Tabela verdade do 4013

Pela tabela verdade, vemos que as entradas CLEAR E PRESET são ativas no nível alto, mas que somente uma delas pode estar nesta condição de cada vez. Se as duas entradas PRESET e CLEAR forem colocadas no nível alto ao mesmo tempo, o flip-flop vai para uma condição não permitida.

A informação presente na entrada D é transferida para a saída quando as entradas assíncronas PRESET E CLEAR estão inativas.

É importante observar que a velocidade de operação dos circuitos CMOS depende da tensão de alimentação, como já estudamos nas lições anteriores.

Nos manuais de circuitos integrados CMOS os leitores poderão encontrar tabelas que dão os diversos tempos de propagação dos sinais e as frequências de operação em função desta tensão de alimentação.

A tabela abaixo dá as principais características desse componente:

Características	Condições (Vdd)	Valores
Corrente drenada/fornecida (tip)	5 V	0,88 mA
	10 V	2,25 mA
	15 V	8,8 mA
Frequência máxima de clock	5 V	5 MHz
	10 V	12,5 MHz
	15 V	15,5 MHz
Corrente quiescente (max)	5 V	1,0 μ A
	10 V	2,0 μ A
	15 V	4,0 μ A
Faixa de tensões de alimentação	3 V a 15 V	

b) 4027 – Duplo Flip-Flop J-K com Preset e Clear

Neste circuito integrado encontramos dois flip-flops tipo J-K com entradas de PRESET E CLEAR. O invólucro é DIL de 16 pinos, mostrado na figura 187.

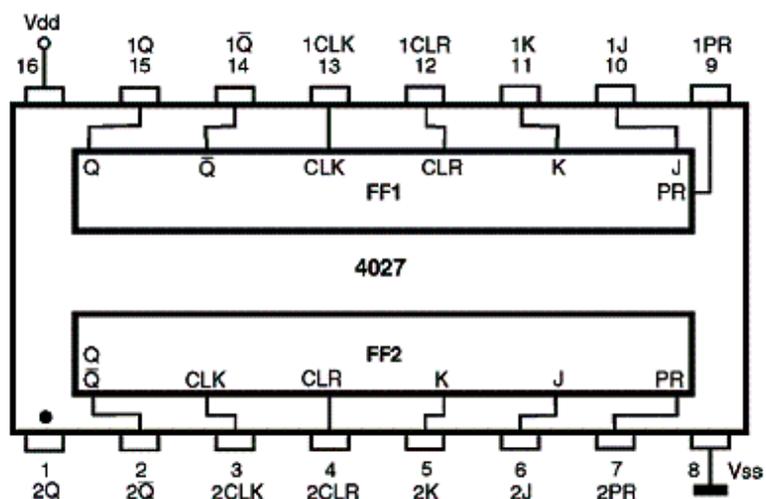


Figura 187 – 4027 – Dois flip-flops J-K com Preset e Clear

As entradas PRESET e CLEAR são independentes para cada flip-flop. A tabela verdade para cada flip-flop é mostrada na figura 188.

CLR	PR	J	K	CLK	Q _{n+1}	\bar{Q}_{n+1}
∅	∅	∅	∅	↑	Q _n	\bar{Q}_n
∅	∅	∅	1	↑	∅	1
∅	∅	1	∅	↑	1	∅
∅	∅	1	1	↑	\bar{Q}_n	Q _n
1	∅	X	X	X	∅	1
∅	1	X	X	X	1	∅
1	1	X	X	X	1	1

(*) não permitido X = não importa

Figura 188 – Tabela verdade para o 4027

Observe que temos acesso tanto as saídas normais como complementares de cada um dos flip-flops, e que as saídas CLEAR E PRESET estão ativas no nível alto. No entanto, como nos demais flip-flops, estas saídas não podem ser ativadas ao mesmo tempo, pois levariam os flip-flops a uma condição não permitida.

A tabela dada a seguir fornece as principais características elétricas do 4027.

Características	Condições (Vdd)	Valor
Corrente drenada/ fornecida (tip)	5 V	0,88 mA
	10 V	2,25 mA
	15 V	8,8 mA
Frequência máxi- ma de clock	5 V	6MHz
	10 V	12,5 MHz
	15 V	15,5 MHz
Corrente quiescen- te (max)	5 V	1 µA
	10 V	2 µA
	15 V	4 µA
Faixa de tensões de alimentação	3 V a 15 V	

c) 4043 – Quatro Flip-Flops R-S (Lógica NOR)

O circuito integrado 4043 contém quatro flip-flops R-S independentes com saídas tri-state. O invólucro DIL de 16 pinos é mostrado na figura 189.

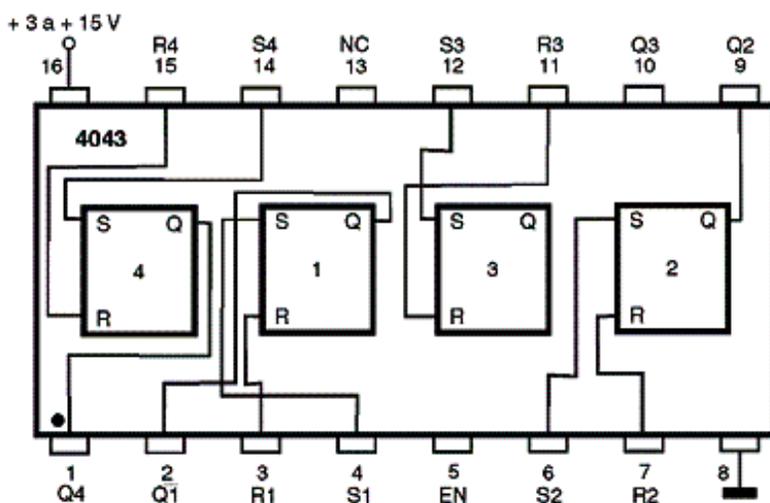


Figura 189 – 4043 – Quatro flip-flops R-S

Em cada um dos flip-flops as entradas SET e RESET podem normalmente ficar no nível baixo. Se a entrada SET for levada ao nível alto, a saída permanece no nível alto. Se a entrada RESET for levada ao nível alto a saída permanece no nível baixo. As duas saídas não podem ser levadas ao mesmo tempo ao nível alto, pois isso representa um estado não permitido.

Lembramos que os estados não permitidos podem ser perigosos para a integridade do componente levando-o a uma dissipação maior.

As saídas vão ao estado de alta impedância quando as entradas EN (habilitação ou ENABLE) são levadas ao nível baixo. Quando o nível da entrada EN é alto as saídas são conectadas aos flip-flops transferindo seus estados para os circuitos externos.

Como estes circuitos não usam clocks, não devem ser ligados em cascata para formar contadores ou shift-registers.

A tabela dada na figura 190 fornece as principais características elétricas do 4043.

Características	Condições (Vdd)	Valor
Corrente drenada/fornecida (tip)	5 V	1,0/0,4 mA
	10 V	2,6/1,0 mA
	15 V	6,8/3,0 mA
Tempo de propagação (S ou R para Q) (tip)	5 V	175 ns
	10 V	75 ns
	15 V	60 ns
Corrente quiescente (max)	5 V	1,5 µA
	10 V	3,0 µA
	15 V	4,0 µA
Faixa de tensões de alimentação	3 V a 15 V	

Figura 190 – Características do 4043

Observe que esse dispositivo tem capacidades diferentes de fornecimento e drenagem de corrente em suas saídas

c) 40174 – Seis fli-flops tipo D

O circuito integrado CMOS 40174 contém seis flip-flops tipo D, disparados pela transição positiva do sinal de clock. Apenas uma das saídas de cada flip-flop é acessível externamente e o CLEAR é comum a todos eles.

O invólucro é DIL de 16 pinos com a pinagem mostrada na figura 191.

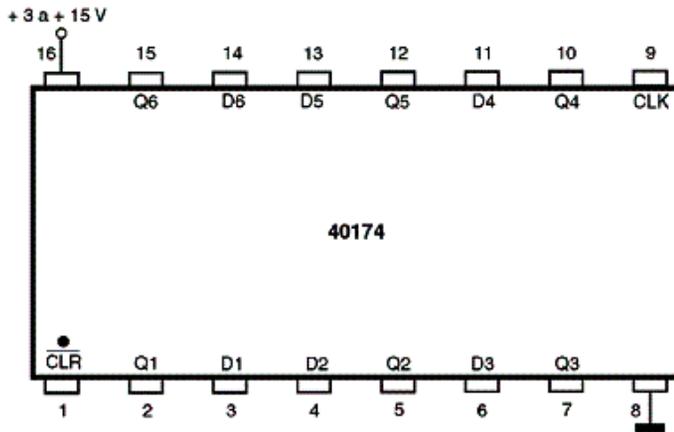


Figura 191 – 40174 – Seis flip-flops tipo D

Todos os flip-flops são controlados por uma entrada comum de clock.

A tabela verdade para cada flip-flop deste circuito integrado é mostrada na figura 192.

CLR	CLK	D	Q _{n+1}	\bar{Q}_{n+1}
∅	X	X	∅	1
1	↑	1	1	∅
1	↑	∅	∅	1
↑	1	X	Q _n	\bar{Q}_n
1	∅	X	Q _n	\bar{Q}_n

x = não importa

Figura 192 – Tabela verdade para o 40174

Na tabela da figura 193 damos as características elétricas principais desse circuito integrado.

Características	Condições (Vdd)	Valor
Corrente drenada/ fornecida (tip)	5 V	0,88 mA
	10 V	2,25 mA
	15 V	8,8 mA
Frequência máxi- ma de clock	5 V	3,5 MHz
	10 V	10 MHz
	15 V	12 MHz
Corrente quiescen- te (max)	5 V	1,0 μ A
	10 V	2,0 μ A
	15 V	4,0 μ A
Faixa de tensões de alimentação	3 V a 15 V	

Figura 193 – Características do 40174

É interessante observar que este dispositivo é equivalente ao TTL 74174.

d) 40175 – Quatro flip-flops tipo D

Trata-se de um circuito integrado que contém quatro flip-flops semelhantes ao anterior, com a diferença de que as duas saídas (normal e complementar) de cada flip-flop podem ser acessadas. O invólucro deste circuito integrado é mostrado na figura 194.

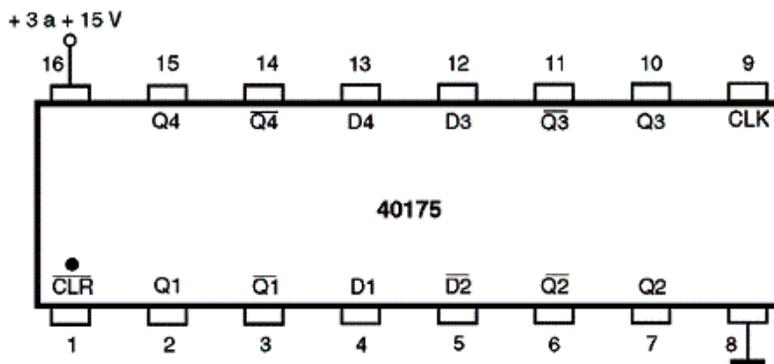


Figura 194 – 40175 – Quatro flip-flops tipo D

A tabela verdade para cada circuito integrado é a mesma do 40174.

As características elétricas do 40175 são as mesmas do 40174. Esse circuito integrado é equivalente em função ao TTL 74192.

7.3 – Funções lógicas TTL

Podemos contar com uma boa quantidade de circuitos integrados contendo as principais funções lógicas em tecnologia TTL. Damos a seguir alguns dos mais importantes, já que para obter informações sobre a totalidade das que são disponíveis será interessante para

o leitor contar com um manual TTL ou usar os mecanismos de busca da internet.

a) 7400 - Quatro Portas NAND de duas entradas

Num invólucro DIL de 14 pinos contamos com quatro portas NAND de duas entradas de funcionamento independente. A pinagem deste circuito integrado é mostrada na figura 195.

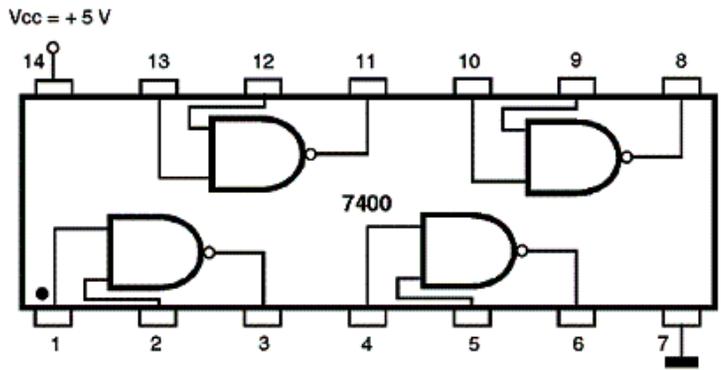


Figura 195 – 7400 – Quatro portas NAND de duas entradas

O consumo médio por circuito integrado é da ordem de 12 mA.

Lembramos que a pinagem para os equivalentes das subfamílias é a mesma, e que também se mantém para os invólucros SMD.

b) 7402 - Quatro Portas NOR de duas entradas

Esse circuito integrado em invólucro DIL de 14 pinos tem a pinagem mostrada na figura 196, e cada unidade exige uma corrente de 12 mA.

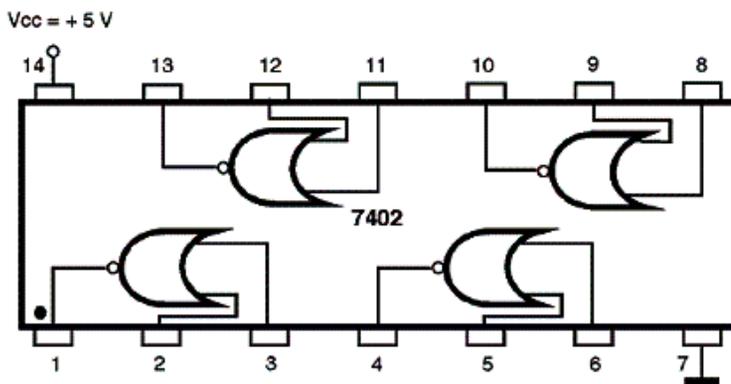


Figura 196 – 7402 – Quatro portas NOR de duas entradas

As quatro portas contidas nesse circuito integrado podem ser usadas de forma independente.

Manuais CMOS e TTL

Todo praticante de eletrônica deve ter manuais TTL e CMOS, quer seja na forma impressa como digital (E-books), pois sua consulta será frequente, quando se trabalhar com eletrônica digital.

c) 7404 - Seis Inversores (Hex Inverter)

Os seis inversores deste circuito integrado podem ser usados de forma independente. A pinagem é mostrada na figura 197.

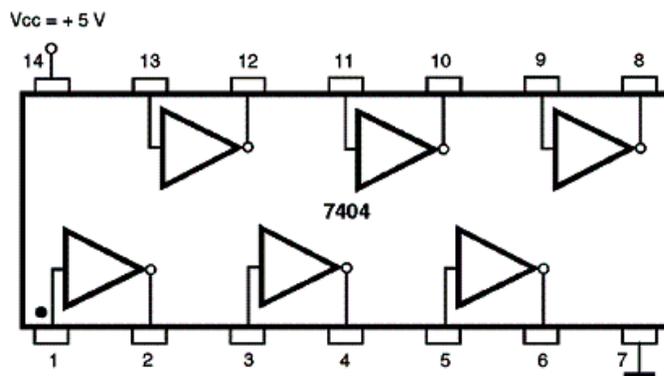


Figura 197 – 7404 – Seis inversores

As saídas dos inversores do 7404 podem drenar até 30 mA no nível baixo e isso com alimentação de até 30 V. Lembramos, entretanto, que mesmo com o circuito de saída sendo alimentado por 30 V, o circuito integrado deve manter sua alimentação em 5 V.

d) 7408 - Quatro Portas AND de duas entradas

Este circuito integrado tem a pinagem mostrada na figura 198 e cada unidade exige uma corrente de 16 mA.

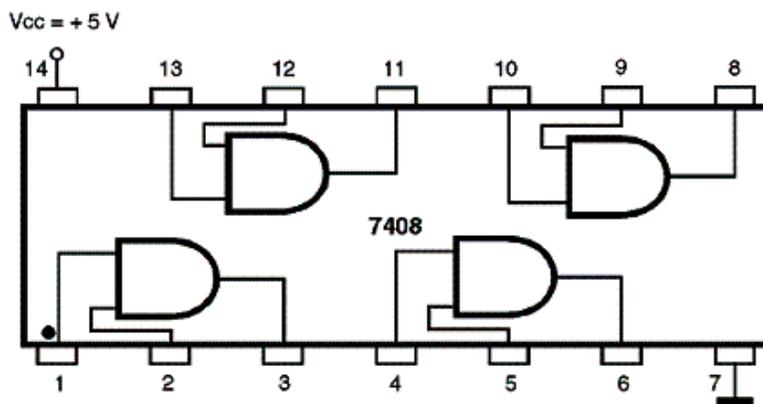


Figura 198 – 7408 – Quatro portas AND de duas entradas

As portas contidas no 7408 podem ser usadas de forma independente.

e) 7410 - Três portas NAND de três entradas

Cada uma das três portas NAND deste circuito integrado pode ser usada de forma independente. A corrente exigida pelo circuito é de 6 mA. A figura 199 mostra sua pinagem.

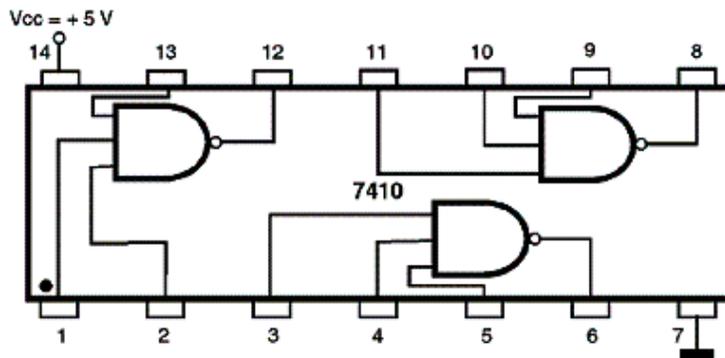
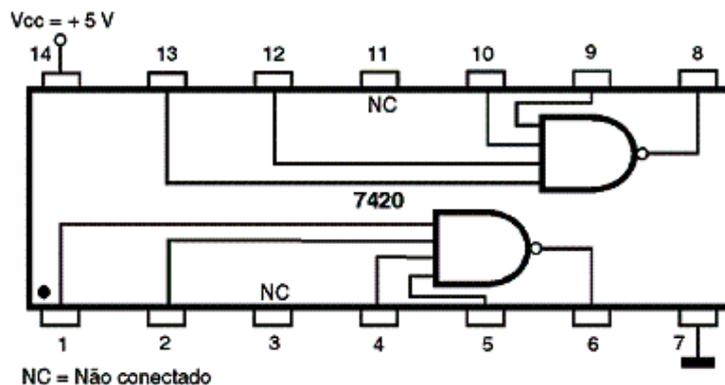


Figura 199- 7410 – Três portas NAND de três entradas

Essa pinagem também é válida para os equivalentes das outras subfamílias.

f) 7420 - Duas portas NAND de quatro entradas

Este circuito integrado contém duas portas NAND de quatro entradas que podem ser usadas de forma independente. A pinagem é mostrada na figura 200.



NC = Não conectado

Figura 200- 7420 – Duas portas NAND de quatro entradas

O consumo por unidade é de aproximadamente 4 mA.

A pinagem para a mesma função das subfamílias é a mesma.

Também observamos que, quando uma das entradas não é usada, ela pode ser colocada no nível apropriado para ser desabilitada.

g) 7432 - Quatro portas OR de duas entradas

As portas OR do circuito integrado TTL 7432 podem ser usadas de modo independente e a corrente total exigida é da ordem de 19 mA. A pinagem é mostrada na figura 201.

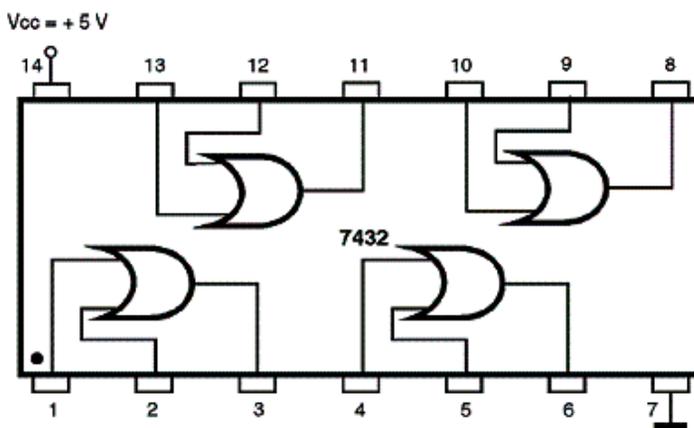


Figura 201 – 7432 - Quatro portas OR de duas entradas

A pinagem para os componentes das subfamílias se mantém, assim como das versões SMD.

h) 7486 - Quatro Portas OR-Exclusivo

As portas OU-exclusivo ou Exclusive OR deste circuito integrado podem ser usadas de forma independente. O consumo é de 30 mA e a pinagem é mostrada na figura 202.

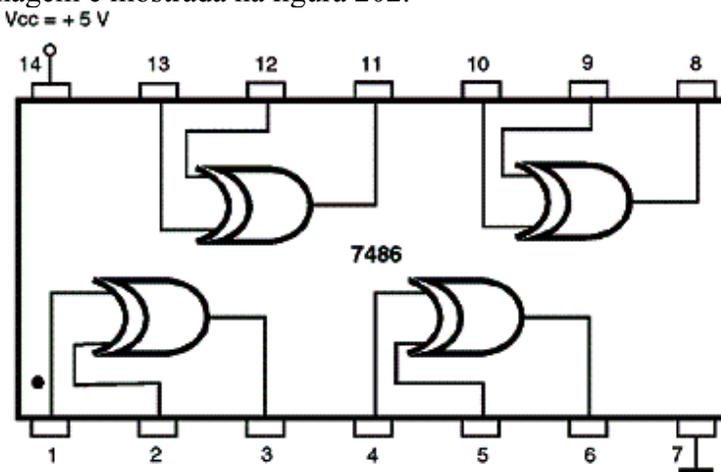


Figura 202 – 7486 - Quatro portas Exclusive-OR ou OU-exclusivo

Para os mesmos componentes das subfamílias a pinagem se mantém.

7.4 – Funções lógicas CMOS

Também podemos contar com uma boa quantidade de circuitos integrados CMOS contendo funções lógicas. Evidentemente, não temos espaço para colocar todas estas funções nesta lição, sendo recomendado ao leitor que consulte um manual CMOS, visite algum site de fabricante na Internet ou o site do autor deste livro.

b) 4011 - Quatro portas NAND de duas entradas

Em invólucro DIL de 14 pinos encontramos quatro portas NOR de duas entradas de funcionamento independente. O invólucro com a identificação dos terminais é mostrado na figura 204.

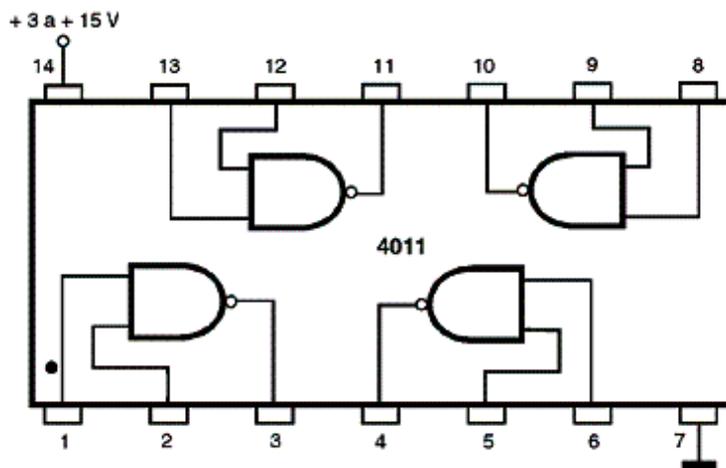


Figura 204 – 4011 – Quatro portas NAND de duas entradas

Na tabela seguinte, temos as características elétricas desse circuito integrado.

Característica	Condições (Vdd)	Valor
Corrente drenada/ fornecida (ip)	5 V	0,88 mA
	10 V	2,25 mA
	15 V	8,8 mA
Tempo de propagação (tip)	5 V	120 ns
	10 V	50 ns
	15 V	35 ns
Corrente quiescente máxima	5 V	0,4 µA
	10 V	0,5 µA
	15 V	0,6 µA
Faixa de tensões de alimentação	3 a 15 V	
Características do 4011		

Se as entradas de cada uma das portas forem interligadas, obtemos inversores lógicos.

c) 4012 - Duas portas NAND de quatro entradas

As quatro portas NOR de duas entradas deste circuito integrado podem ser usadas de forma independente. A identificação dos terminais deste circuito integrado é mostrada na figura 205.

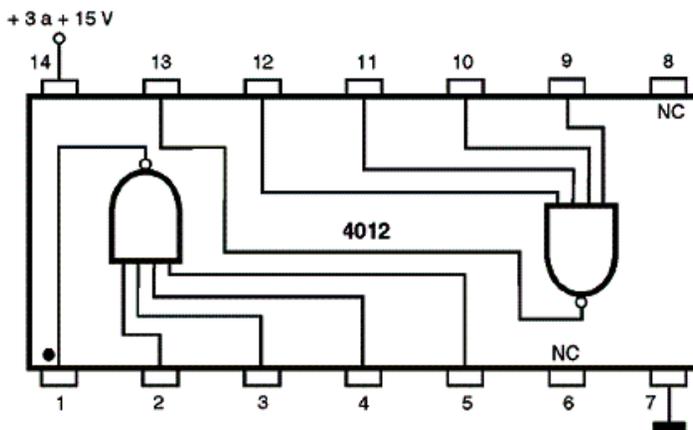


Figura 205 – 4012 – Duas portas NAND de quatro entradas

Na tabela seguinte temos as características elétricas do 4012.

Característica	Condições (Vdd)	Valor
Corrente drenada/ fornecida (tip)	5 V	0,88 mA
	10 V	2,25 mA
	15 V	8,8 mA
Tempo de propaga- ção (tip)	5 V	120 ns
	10 V	50 ns
	15 V	45 ns
Corrente quiescen- te máxima	5 V	0,25 μ A
	10 V	0,5 μ A
	15 V	1,0 μ A
Faixa de tensões de alimentação	3 a 15 V	
Características do 4012		

Para obter uma função NAND de 3 entradas, a partir das portas existentes nesse circuito integrado, basta ligar a porta não usada ao nível alto.

d) 4023 - Três portas NAND de 3 três entradas

As três portas NAND deste circuito integrado podem ser usadas de maneira independente. A pinagem é mostrada na figura 206.

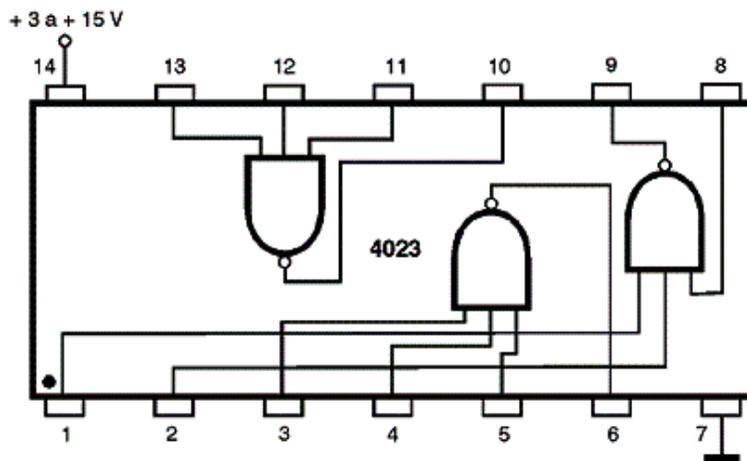


Figura 206 – Três portas NAND de três entradas

A tabela seguinte fornece as principais características elétricas desse circuito integrado.

Característica	Condições (Vdd)	Valor
Corrente drenada/ fornecida (tip)	5 V	0,5/0,5 mA
	10 V	21,0/0,6 mA
	15 V	-
Tempo de propaga- ção (tip)	5 V	60 ns
	10 V	25 ns
	15 V	-
Corrente quiescen- te máxima	5 V	0,05 µA
	10 V	0,1 µA
	15 V	-
Faixa de tensões de alimentação	3 a 15 V	
Características do 4023		

Para usar qualquer das portas como uma função NAND de duas entradas, basta ligar a remanescente ao nível lógico “1”. Para usar como inversor, basta interligar as três entradas de uma porta.

e) 4025 - Três portas NOR de três entradas

Encontramos neste circuito integrado três funções NOR que podem ser usadas de forma independente. A pinagem é mostrada na figura 207.

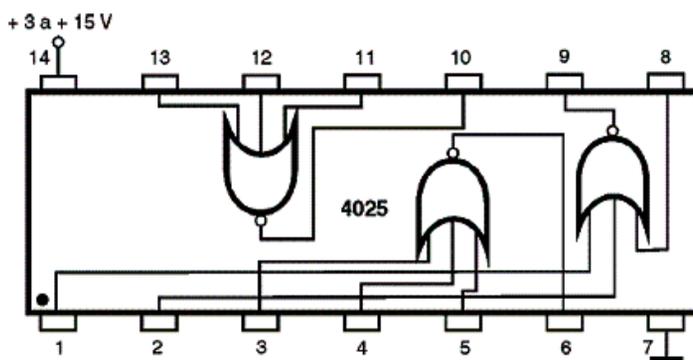


Figura 207 – 4025 – Três portas NOR de três entradas

Na tabela seguinte temos as principais características elétricas desse circuito integrado.

Característica	Condições (Vdd)	Valor
Corrente drenada/ fornecida (tip)	5 V	0,88 mA
	10 V	2,25 mA
	15 V	8,8 mA
Tempo de propagação (tip)	5 V	130 ns
	10 V	60 ns
	15 V	40 ns
Corrente quiescente máxima	5 V	0,25 μ A
	10 V	0,5 μ A
	15 V	1,0 μ A
Faixa de tensões de alimentação	3 a 15 V	
Características do 4025		

Cada porta pode ser usada como função E (AND) de duas entradas, se a porta restante for aterrada.

7.5 - A Função Tri-State Expansível do 4048

O circuito integrado 4048 tem características muito interessantes para projetos CMOS envolvendo funções lógicas. Conforme estudamos, usando combinações apropriadas de funções simples podemos obter qualquer outra função mais complexa. É justamente isso que faz o 4048 que tem a pinagem mostrada na figura 208.

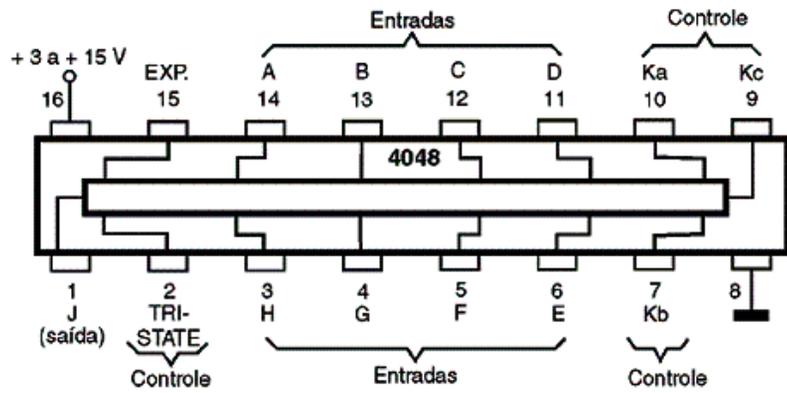


Figura 208 – Pinagem do 4048

Este circuito possui 8 entradas de sinal e uma saída de sinal além de três entradas de “programação”. Dependendo dos níveis lógicos aplicados nas entradas de programação, o circuito se comporta como uma função diferente. Assim, ele pode se tornar funções NOR, OR, NAND ou AND com 8 entradas ou ainda de forma combinada, realizando ao mesmo tempo funções de portas OR e AND, cada uma de 4 entradas além de outras que são mostradas em detalhes na figura 209.

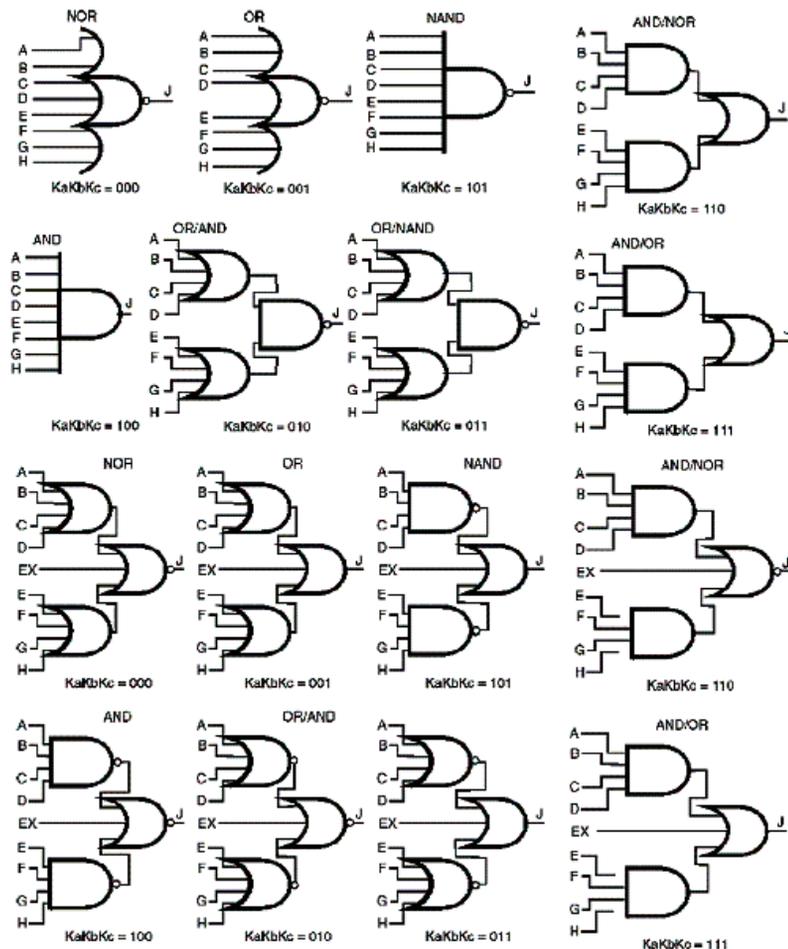


Figura 209- As funções do 4048

Assim, por exemplo, se colocarmos todas as três entradas de programação no nível alto (K_a, K_b e $K_c = 111$) o circuito se comporta como duas portas AND de quatro entradas, ligadas uma porta OR de duas entradas.

Veja então que esta interessante função pode servir de “coringa” em muitos projetos, pois pode assumir a operação de diversas combinações de outros circuitos integrados CMOS.

Internamente o 4048 é bastante complexo, contendo 32 funções independentes que são programadas pelos níveis lógicos aplicados às entradas correspondentes.

As características elétricas do 4048 são dadas na tabela a seguir:

Característica	Condições (Vdd)	Valor
Corrente drenada/ fornecida (tip)	5 V	4,0 mA
	10 V	11 mA
	15 V	23 mA
Tempo de propagação (tip)	5 V	425 ns
	10 V	200 ns
	15 V	180 ns
Corrente quiescente máxima	5 V	0,01 μ A
	10 V	0,01 μ A
	15 V	0,01 μ A
Faixa de tensões de alimentação	3 a 15 V	
Características do 4048		

Quantidade de circuitos integrados

Os circuitos integrados que vimos são apenas os principais. As séries CMOS e TTL contém centenas de circuitos integrados, cujos dados podem ser obtidos nos datasheets, manuais e mesmo nos mecanismos de busca da Internet.

Dois fatos relevantes chamam a atenção de quem vai usar essa função em algum projeto.

O primeiro deles é a alta capacidade de fornecimento ou drenagem de corrente das saídas, que é bem maior do que os demais tipos dessa série.

O segundo fato é a baixa velocidade de operação, com um tempo relativamente longo que o circuito precisa para processar o sinal, dado pelo tempo de propagação.

Termos em Inglês:

Da mesma forma que nas outras lições, ressaltamos que existem muitos termos que, mesmo na literatura técnica em português, são usados na forma original em inglês. No entanto, muitos têm traduções como:

Inverter – inversor

Gates – portas

Type – tipo

Latches – trava

Divider – divisor

Common - comum

Termos para pesquisa:

- Funções lógicas
- Sub-famílias TTL
- Interfaceamento digital
- Lógica combinacional
- Subfamílias CMOS

QUESTIONÁRIO

1. Qual é o conjunto de funções que o leitor provavelmente não encontrará na forma de um circuito integrado TTL ou CMOS?

- a) Seis portas AND de 3 entradas
- b) Seis inversores
- c) Quatro portas AND de duas entradas
- d) Quatro portas Exclusive OR

2. As quatro portas NAND de um circuito integrado TTL 7400 têm:

- a) Alimentação independente
- b) Quatro entradas
- c) Funcionamento independente
- d) Reset comum

3. Os flip-flops do circuito integrado 4027 são:

- a) Do tipo R-S
- b) Do tipo D
- c) Do tipo J-K
- d) Do tipo T

4. Um “latch” como o circuito TTL 7475 é usado para:

- a) Contagem binária
- b) Divisão de frequência
- c) Operação como porta AND
- d) Armazenamento de informação digital

5. Qual é a condição proibida nos Flip-flops CMOS e TTL?

- a) Entradas J e K ligadas em paralelo
- b) Preset e Clear ao mesmo tempo ativos
- c) Preset e Clear ao mesmo tempo desativados
- d) Saídas ligadas às entradas D ou Clear



» RESPOSTAS

» FONTES PARA INTEGRADOS TTL

a) Respostas para as questões dos capítulos

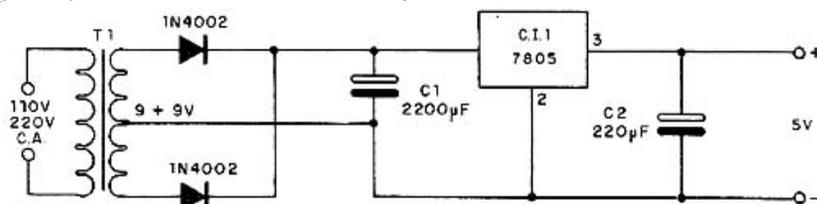
- 1) 1-d, 2-a, 3-c, 4-b, 5-a, 6-c, 7-c
- 2) 1-b, 2-b, 3-c, 4-a, 5-d, 6-c
- 3) 1-a, 2-c, 3-d, 4-b, 5-d, 6-c, 7-d
- 4) 1-c, 2-c, 3-d, 4-b, 5-a, 6-c
- 5) 1-c, 2-b, 3-b, 4-a, 5-d
- 6) 1-c, 2-a, 3-c, 4-b, 5-d, 6-d
- 7) 1-a, 2-c, 3-c, 4-d, 5-b

A.1 - Fontes para integrados TTL

Conforme estudamos, os circuitos integrados TTL funcionam com uma tensão fixa de 5 V. Assim, para sua alimentação devem ser utilizadas fontes apropriadas, cuja corrente deve ser dimensionada de acordo com o consumo total do circuito. A seguir damos três circuitos de fontes especialmente projetadas para alimentar circuitos integrados TTL.

Fonte TTL de 5 V x 1 A

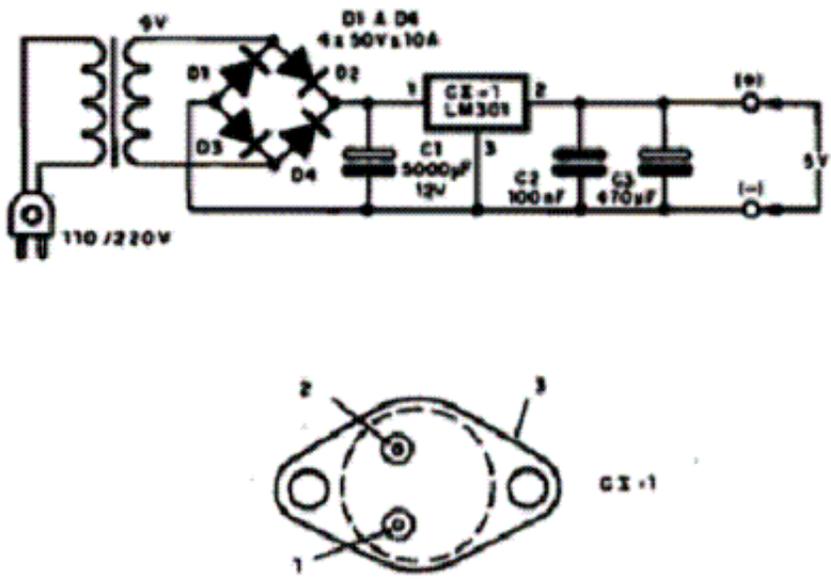
Esta é uma fonte tradicional para alimentação de circuitos TTL até 1 A de consumo. O circuito integrado 7805 deve ser dotado de dissipador de calor. Existe uma versão do 7805 em invólucro SOT-54 sem dissipador, para correntes até 200 mA. A corrente do secundário do transformador deve ser de acordo com a corrente da fonte na sua aplicação. Os diodos admitem equivalentes.



Este circuito está no livro Circuitos e Informações volume II, disponível para download gratuito no site.

Fonte TTL 5 V x 5 – LM301

Esta fonte de alimentação, indicada para circuitos TTL fornece uma tensão de 5 V com corrente até 5 A. O circuito integrado regulador de tensão positivo deve ser montado num bom radiador de calor. O transformador deve ter corrente de 5 A e os diodos devem ser de tipos capazes de conduzir a corrente exigida pelos circuitos alimentados. Observe a pinagem do LM301, com o terminal de terra na carcaça.



Fonte de 5 V x 500 mA

Esta fonte até 500 mA é indicada para a alimentação de circuitos integrados TTL. O transistor deve ser dotado de radiador de calor e pode ser usado um transformador menor se a corrente da carga for menor. O resistor de 22 ohms deve ser de pelo menos 2 W de dissipação.

