

Desmistificando a Eletrônica Digital para Iniciantes

Paulo Brites

PARA ALUNOS ÁREA VIP CLUBE APRENDA ELETRÔNICA C PAULO BRITES

**CÓPIA EXCLUSIVA PARA ALUNOS VIP DO
CLUBE APRENDA ELETRÔNICA COM PAULO BRITES**

Descomplicando a Eletrônica Digital para Iniciantes

Descomplicando a Eletrônica Digital para iniciantes

Paulo Brites

Este material é parte de um futuro livro e está sendo fornecido gratuitamente para uso exclusivo dos alunos da área VIP do Clube Aprenda Eletrônica com Paulo Brites.

2019

Descomplicando a Eletrônica Digital para Iniciantes

A quem se destina este livro

A maneira como os livros nacionais clássicos tratam a Eletrônica Digital não me agrada e acho que quem está se iniciando no assunto tem a mesma impressão que eu.

Tentar estudar por estes livros é um convite a desistir.

Quando eu fiz o curso técnico a eletrônica digital chamava-se técnica de pulsos e era feita com válvulas!

De repente, 10 anos mais tarde, 1978, fui fazer um treinamento sobre eletrônica digital na Ericsson em São Paulo por conta da Embratel onde eu trabalhava na época.

Os circuitos integrados já estavam começando a chegar.

Foram três semanas bem “puxadas” de cursos. Tive que estudar muito.

Voltei empolgado com o assunto e comecei a “comer” todos os livros sobre eletrônica digital que conseguia mandar vir lá nos “isteites”.

Sempre fui autodidata, gosto de estudar sozinho, às vezes, é mais demorado, mas se houver dedicação, quando se aprende, se aprende e para sempre.

Não me limitava a ler, mas também a praticar, montando circuitos e analisando como eles funcionavam.

Aliás, isto é o segredo para se aprender, praticar bastante.

Naquele eu ano eu tinha publicado o meu primeiro artigo na Revista Antenna e logo comecei a produzir artigos sobre minhas experiências com eletrônica digital (sem válvulas!).

Alguns artigos meus foram copiados e indo para numa publicação no Uruguai chamada Eletronica al día.

Descomplicando a Eletrônica Digital para Iniciantes

Contatei o editor e comecei a escrever para eles um cursinho de introdução a eletrônica digital.

De repente o contato com editor no Uruguai cessou. Ele não respondia mais as minhas cartas. Sim, a comunicação era por cartas, datilografadas, acreditem!

Talvez ele tenha “sumido”, não sei. Era a época das ditaduras geradas em Washington e implantadas na América do Sul. Tudo na melhor das intenções, para impedir que o comunismo tomasse conta do mundo.

Mas, voltando a eletrônica, cada vez mais ia percebendo que deveria existir uma outra forma de explicar as coisas de modo que todas as pessoas pudessem entender e não apenas alguns privilegiados (que talvez nem entendessem tanto assim do que falavam).

Ministrei vários cursos, escrevi várias apostilas e artigos, mas ficou tudo espalhado por aí.

Chegou a hora de botar a casa em ordem.

Escrever um livro sobre eletrônica digital que não faça as pessoas desistirem após terem lido o primeiro parágrafo sempre esteve nos meus planos.

Pretensão demais?

Sim, sou pretencioso e ousado. Sou revolucionário. Sigo as palavras do Papa Francisco e as ideias de Galileu que dizia que não existe ninguém que não possa aprender alguma coisa nova desde que tenha foça de vontade e encontre alguém esteja disposto a lhe ensinar.

Este livro complementa o Eletrônica para Estudantes, Hobistas e Inventores que publiquei em 2015.

Espero que lhe seja útil e que perceba que a eletrônica digital não é complicada como muitos pensam.

O autor.

Descomplicando a Eletrônica Digital para Iniciantes



Não pratique pirataria

Se você adquiriu uma cópia não autorizada deste livro você está praticando pirataria de acordo com a Lei de Direitos Autorais nº 9.610 de 19 de fevereiro de 1998.

Distribuir cópias em papel ou em meios digitais deste livro sem autorização por escrito do autor, além de ser contra a lei (o que pode lhe ensejar um processo judicial), prejudicará todo o trabalho que o autor devotou para elaborar o material, portanto peço-lhe que, mais por uma questão de consciência do que legal, não o faça.

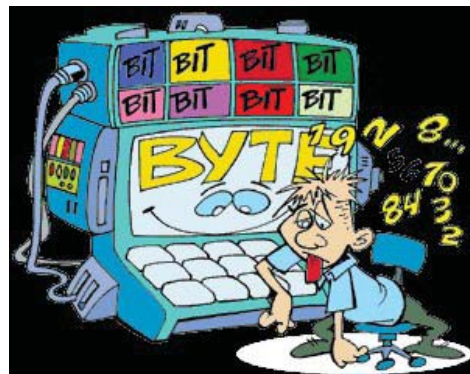
O Autor

Capítulo 1

Parafraseando a fase bíblica “faça-se a luz” eu digo “façam-se os bits” e assim, nasceu a eletrônica digital.

Este é um capítulo de leitura obrigatória.

Descomplicando a Eletrônica Digital para Iniciantes



Que “bichos” são estes?

Separar **bits** e **bytes** da Infomática e da Eletrônica Digital é quase, ou melhor, é impossível.

Como dizem por aí, “*está tudo junto e misturado*”.

Aliás, a Informática só “existe” por causa da Eletrônica Digital que, no “passado” chamava-se Técnica de Pulsos, mas isto é outra história.

Quando pensamos em Eletrônica logo nos vem a cabeça pecinhas como resistores, capacitores e os tais dos “chips”, não é mesmo?



Mas para estudar Eletrônica Digital devemos começar entendendo o que são **bits** e **bytes**.

Começemos pelo significado da palavra, ou melhor, sigla **bit**:

binary digit

que traduzindo significa **dígito binário**.

E daí, mas afinal o que é bit ou dígito binário?

Segundo o Dicionário de informática e Internet de Marcia Regina Saway, “**bit é um caractere simples em um número binário**”.

É bem provável que a definição acima tenha confundido sua cabeça mais do que esclarecido, pois agora ainda apareceu um tal de “número binário” além dos bits.

Calma, não há motivo para desespero. Prometo que tudo se esclarecerá, mas precisamos estudar e entender o que são **sistemas de numeração** o que não é nenhum bicho de sete cabeças.

Descomplicando a Eletrônica Digital para Iniciantes



Sistemas de Numeração

O sistema de numeração atual, usado universalmente pelos “humanos”, é chamado de **sistema decimal** e tem como **base** o número **10**.

Você seria capaz de “desconfiar” porque?

Acho que a figura abaixo vai lhe dar uma dica.



Afinal quem nunca contou pelos dedos, não é mesmo?

Mas o que isto tem a ver com a Eletrônica Digital?

A Eletrônica Digital, diferentemente da Analógica ou Linear, trabalha com o conceito de **estados lógicos** que utiliza apenas duas condições: ligado (*on*) ou desligado (*off*).

E onde entra o sistema de numeração nesta história?

Bem, se só podemos trabalhar com dois estados, já dá pra perceber que não podemos utilizar os dez dedos, ou seja, um sistema de numeração decimal e, então teremos usar um sistema que só use “dois dedos” e por isso, tem que ser um sistema de numeração binário (bi quer dizer dois).

Para que possamos expressar matematicamente dois **estados lógicos** iremos utilizar apenas os dígitos 0 e 1.

Reparou que eu tive o cuidado de sublinhar a palavra dígitos?

Fiz isto para chamar a atenção, pois ela tem um significado diferente da palavra número.

Dígito, que vem do latim ***digitus***, significa dedo e representa cada símbolo de um sistema de numeração em um determinado número.

Descomplicando a Eletrônica Digital para Iniciantes

decimal utiliza dez dígitos que são:
0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9.

O número 10, por exemplo, é uma “combinação” de dois dígitos: o “1” (um) e o “0” (zero).

Coloquei o **número** dez como exemplo, mas podemos escrever **qualquer número**, no sistema decimal, **usando** apenas os seus **dez dígitos** de 0 a 9.

Já vimos que na Eletrônica Digital iremos trabalhar com os números binários que só utilizam dois dígitos para representar qualquer número.

Quais os dígitos você escolheria para esta representação?

Acho que não vale a pena querer reinventar a roda numa hora dessas. Eu escolheria **zero** e **um** e, “por coincidência”, foi a mesma coisa que os cientistas fizeram.

Então, fica combinado assim:

- se o circuito está desligado (*off*) representamos esta condição pelo dígito binário 0 (zero), mas

- se o circuito está ligado (*on*) utilizaremos o dígito binário 1 (um) e não se discute mais isto.

E os bits e bytes, onde entram?
Lembra da **definição de bit** dada

lá atrás: “**um caractere simples em um número binário**”?

Pois bem, este “**caractere simples**” será o **zero** e o **um**, como já vimos.

Hora de fazer uma pergunta:

- quantos e quais são os **números decimais** que poderíamos escrever utilizando-se apenas os dois bits (0 e 1) do sistema binário?

Podemos escrever apenas quatro **números decimais** usando apenas os dois bits do sistema binário que serão representados por: **00, 01, 10 e 11**.

Você saberia dizer quais os **números decimais** estes números binários representam?

Espero, ansiosamente, que você tenha respondido: 0, 1, 2 e 3.

A primeira vista isto parece um pouco confuso e, mal comparando, lembra um pouco como aprender a ver as horas em um relógio “de ponteiro”. Lembra como era difícil? Mas você aprendeu, não foi!

Coragem. Não desanime!

Então quer dizer que tanto faz escrever 10 ou 2, assim como, 11 ou 3 que é a mesma coisa?

Descomplicando a Eletrônica Digital para Iniciantes

Calma que eu já explico.

Os “números” 10 e 11 do exemplo **não** estão no sistema decimal e sim no binário.

Uma coisa é uma coisa, outra coisa é outra coisa!

A questão é que, desde criancinha, nós aprendemos a trabalhar com a base 10, ou seja, o sistema de numeração decimal e nunca paramos para pensar que poderiam existir outras possibilidades como a base 2, ou sistema binário, que estamos vendo agora.

Quando entramos no mundo digital temos que conviver com os dois sistemas - o decimal, “dos humanos” e o binário, “da máquina”.

Então, os binários 10 e 11 correspondem respectivamente aos decimais 2 e 3.

Você notou que “combinando” apenas os dois bits (0 e 1) dos binários nós só poderíamos contar até quatro.

Em outras palavras, com dois dígitos que no sistema binário chamamos de bits só podemos expressar quatro “valores”, que convenhamos não serve para nada.

Mas, pensando bem, com o sistema de base 10 seria mesma coisa se utilizássemos apenas seus dez dígitos que vão de 0 a 9 e, portanto nunca iríamos escrever “números” maiores que nove.

E aí os hindus tiveram a grande sacada de “inventar” um sistema de numeração que além de decimal é também **posicional**, ou seja, **cada dígito** (de 0 a 9) vai ter um “valor diferente” de acordo com a **posição que ocupar no número**.

É este **sistema decimal posicional** que usamos, também chamado de números arábicos (isso mesmo os hindus inventaram e o árabes ficaram com a fama).

Vamos destrinchar isto melhor.

Por exemplo, no número 2438, o dígito dois deixa de valer dois e passa a valer dois mil (duas vezes mil) porque ele está ocupando a posição que chamaremos de “classe” dos milhares.

Da mesma forma, o quatro passa a valer quatrocentos (quatro vezes cem) porque sua posição é a da classe que chamaremos de classe das centenas.

Quanto ao três, do exemplo, ele vale trinta (três vezes dez) porque está na classe das dezenas e o oito

Descomplicando a Eletrônica Digital para Iniciantes

vale oito mesmo porque está na classe das unidades.

Conclusão,
 $2438 = 2000 + 400 + 30 + 8$.

Mais adiante veremos que estas mesmas ideias podem ser usadas num sistema de base dois como o binário da Eletrônica Digital.

Entretanto, será necessário introduzir algumas “contas” que para alguns pode ser novidade, mas acreditem em mim, são extramamente importantes para aprender a trabalhar com os números binários.

Começemos com a base dez que já faz parte da nossa vida desde criancinha e por isso, vai ficar mais fácil de entender.

Peguemos como exemplo o número 4253 que pode ser escrito da forma mostrada no final da página

A primeira coisa que você precisa observar é que o expoente da base, no caso 10, corresponde à classe a qual o dígito pertence.

Vamos entender melhor isso.

$$\begin{aligned} 4523 &= 4000 + 200 + 50 + 3 = 4 \times 1000 + 2 \times 100 + 5 \times 10 + 3 \times 1 \\ &= 4 \times 10^3 + 2 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 \end{aligned}$$

Expoente, se você não sabe, é o numerozinho que está “em cima da base” e que significa que **a base deve ser multiplicada por ela mesma a quantidade de vezes indicada pelo expoente**.

Por exemplo, $10^3 = 10 \times 10 \times 10$.

Se o expoente é 1, a gente nem escreve porque $10^1 = 10$ mesmo.

E se o expoente for zero (classe das unidades)?

Neste caso tem uma regra matemática que diz que **qualquer número com expoente zero é igual a um**. Eu não vou demonstrar isto aqui, mas é verdade. Acredite em mim, eu não minto!

E se a base fosse 2, como ficaria?

Muito simples, vamos ver.

$$2^0 = 1 \text{ (“por definição”)}$$

$$2^1 = 2$$

$$2^2 = 4 \text{ (} 2 \times 2 \text{)}$$

$$2^3 = 8 \text{ (} 2 \times 2 \times 2 \text{)}$$

$$2^4 = 16 \text{ (} 2 \times 2 \times 2 \times 2 \text{)}$$

Descomplicando a Eletrônica Digital para Iniciantes

$$2^5 = 32 (2 \times 2 \times 2 \times 2 \times 2)$$

$$2^6 = 64 (2 \times 2 \times 2 \times 2 \times 2 \times 2)$$

$$2^7 = 128 (2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2)$$

Deixo para o leitor continuar a tarefa de prosseguir até 2^{10} .

Vimos lá atrás que utilizando apenas os dois bits (0 e 1) dos binários só conseguimos escrever quatro números binários que correspondem aos decimais 0, 1, 2 e 3.

Entretanto, se usarmos com os binários o mesmo conceito de posição ou classe que os hindus usaram para os decimais poderemos escrever o número que quisermos.

Por exemplo, observe o número abaixo escrito em binário

1 1 0 1 0 0 1

E se eu lhe disser que este número corresponde ao decimal 105, você acredita?

Pois vou lhe mostrar que sim e de um forma muito fácil.

Parace “estranho”, mas é a mesma coisa que fazemos com os decimais embora nem percebamos porque já estamos acostumados com eles.

Existem apenas duas diferenças entre trabalhar com decimais e binários:

1) Usamos como base o 2 no lugar do 10

2) Em vez de usar os dígitos 0 a 9, usam-se apenas os dígitos 0 e 1 que nós chamamos de bits.

Se fizermos as contas mostradas no final da página iremos encontrar 105 que é o correspondente decimal do binário 1 1 0 1 0 1 1.

Deixo para você descobrir a qual decimal corresponde o binário

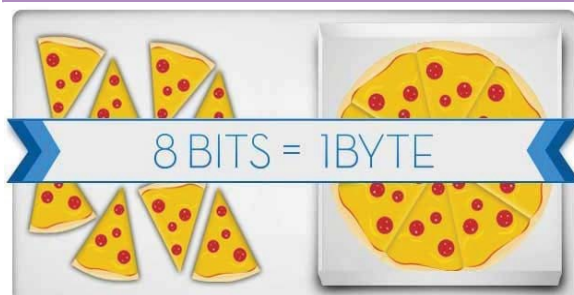
1 1 0 1 1 0 1 0 0 1

A resposta está na página **XX**, mas não vale colar.

No exemplo que produziu o decimal 105 foram usados sete bits no total, enquanto no exercício proposto termos dez bits.

Escreva assim: $1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

Descomplicando a Eletrônica Digital para Iniciantes



E os bytes, o que são?

Um termo muito usado em Informática e em Eletrônica Digital é o **byte** (leia-se bait), mas o que este termo significa afinal?

Vimos que com os dois bits 0 e 1 só podemos escrever o correspondente a quatro números decimais, o que não serve para nada, não é mesmo?

Mas não precisamos ficar restritos a usar apenas dois bits.

Poderíamos usar três bits e neste caso iríamos conseguir escrever oito números como vemos na tabela a seguir.

Costuma-se chamar de **octal** um número **binário formado por três bits**.

OCTAL	DECIMAL
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Que tal acrescentarmos mais um bit?

Você seria capaz de responder quantos decimais conseguiremos escrever com quatro bit?

Muito simples, é só fazer 2^4 e descobrimos que dá 16, portanto com quatro bits escreve-se dezesseis números que vai do zero ao quinze. Acompanhe na tabela.

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

Descomplicando a Eletrônica Digital para Iniciantes

Números binários formados por quatro bits são chamados de **haxadecimais**.

Um conjunto de quatro bits é chamado de *nibble* (leia-se “nibôu”).

Aqui surge uma pequena complicação.

Quando trabalhamos com quatro bits teremos que arranjar uma maneira “diferente” para escrever os decimais ou melhor os hexadecimais a partir do nove em diante, pois se escrevermos 10 poderá surgir uma dúvida se estamos falando do binário 1 - 0 ou do decimal dez.

A solução adotada universalmente para resolver esta complicação foi usar letras em lugar de números a partir do dez decimal.

Então a tabela “oficial” dos hexadecimais para os binários de quatro bits vai ficar da forma como vemos ao lado.

E os **bytes**, cadê eles?

O termo **byte** vem de **binary** **term** é usado para designar números binários de oito bits.

Você seria capaz de dizer quantos decimais podemos escrever com oito bits?

HEXA-DECIMAL	DECIMAL
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Muito simples, é só ver quanto dá 2^8 .

Você deve ter achado 256, portanto com **um byte (oito bits)** podemos escrever desde 0 até 255.

Você deve estar percebendo que está começando a ficar trabalhoso escrever números com oito bits, agora imagina se fossem dezesseis ou trinta e dois e por aí vai.

Para resolver esta trabalheira, a partir de oito bits costuma-se “dividir” o binário em grupos de quatro bits (*nibble*) como vemos nos exemplos a seguir, tendo-se o cuidado de informar que estamos

Descomplicando a Eletrônica Digital para Iniciantes

trabalhando com hexadecimais.

como o do exemplo abaixo

Veja só. Em vez de escrever

0101 1111 1011 0011.

1	0	1	1	0	0	0	1
---	---	---	---	---	---	---	---

iremos escrever apenas

B	1
----------	----------

A grande vantagem de trabalharmos com hexadecimais é a “economia” que fazemos na hora da escrita.

Imagine um número de 32 bits

Ele pode ser reescrito usando-se o formato hexadecimal da seguinte forma 4 F B 3 o que, sem dúvida, fica bem mais simples para nós humanos.

Por enquanto, estes conhecimentos são suficientes para podermos passar para o próximo capítulo.

Capítulo 2

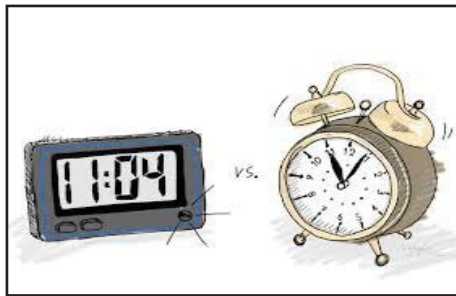
O pessoal do marketing adora dizer que é “digital”.

Mas será que eles sabem o estão dizendo?

E você prefere analógico ou digital?

Descubra a diferença ao ler este capítulo e surpreenda- se.

Descomplicando a Eletrônica Digital para Iniciantes



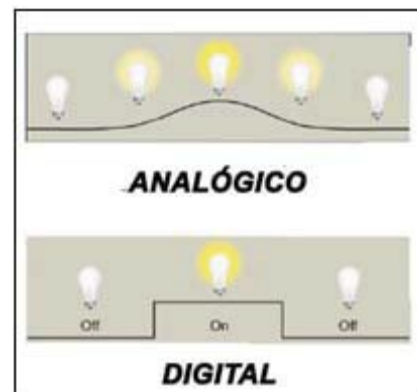
Analógico, Digital & etc

Neste capítulo veremos como estas coisas teóricas, binários, bit e bytes vão virar circuitos eletrônicos que deve ser a sua dúvida.

Começemos examinando as duas figuras ao lado.

O que é preciso ser notado é que o sinal digital não varia suavemente como analógico.

Entretanto, antes de tratar de sinais analógicos ou digitais precisamos estudar um pouco duas “coisinhas” que são a **lógica** e **álgebra de Boole** (leia-se *bule*, que não é o de fazer café).



Antes que você se desespere por causa da palavra “álgebra” que apareceu no parágrafo anterior peço-lhe que não pense em desistir porque lhe garanto que é muito fácil entender essa “coisa”.

Tudo Começou na Grécia no século IV AC

De uma maneira bem superficial podemos dizer que a lógica é uma representação metódica do raciocínio, por isso costuma ser chamada de **raciocínio lógico** (coisa que muita gente não tem, nenhum dos dois).

Aliás, todas as matemáticas se apoiam na lógica, que a meu deveria fazer parte da grade escolar desde os primeiros anos e certamente tornaria a compreensão da matemática bem mais “humana”.

Voltando à lógica, atribuem-se as primeiras manifestações do pensamento lógico ao filósofo grego Aristóteles, lá pelo século IV AC (eu ainda não tinha nascido!).

O primeiro princípio da lógica aristotélica é admitir que só existem duas possibilidades: falso ou verdadeiro (pensando bem não existe meia verdade, nem uma mentirinha).

Descomplicando a Eletrônica Digital para Iniciantes

Que tal associarmos esta ideia de Aristóteles aos números binários?

Poderíamos dizer, por exemplo, que o **verdadeiro vale 1** e o **falso vale 0**, ou poderia ser ao contrário, uma mera questão de convenção (que não tem nenhuma lógica!).

O segundo princípio de Aristóteles é admitir que tudo pode ser expresso por prOposições (não confunda com prEposições que é outra coisa).

Vamos destrinchar esta história de prOposição com um exemplo.

Se dissermos: João é brasileiro ou João gosta de futebol temos duas proposições e chamaremos cada uma de **proposição simples**.

Mas se dissermos: João é brasileiro **E** gosta de futebol, agora temos uma proposição **composta** que foi formada a partir de duas proposições simples unidas pelo **E**.

Deixemos de lado por um momento estas ideias e vamos dar um salto do século IV para o século XVIII e ir até a Inglaterra “visitar” o senhor George Simon **Boole** (1815-1864) que certamente você já desconfiou que foi o inventor da **Álgebra de Boole**.

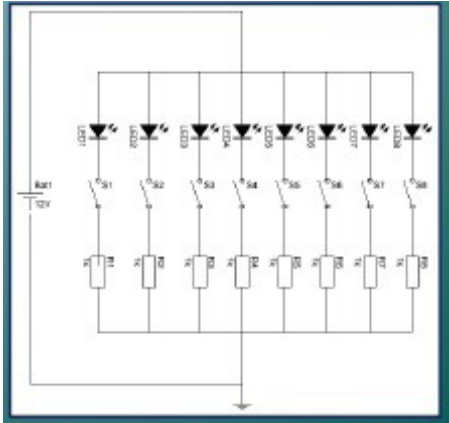
Este senhor teve a ideia de

juntar a lógica de Aristóteles com os números binários para construir uma “álgebra” que recebeu o seu nome (muito justo, não é?).

Veja na tabela abaixo como é fácil.

LÓGICA	BOOLE	ELETÔNICA
VERDADEIRO	1	ON
FALSO	0	OFF

Se você ainda não percebeu como podemos transformar o falso ou verdadeiro de Aristóteles ou o 1 e 0 binário de Boole em um circuito eletrônico a figura abaixo vai lhe dar uma ajudinha.



O funcionamento do circuito é bem simples.

Se a chave estiver fechada o led correspondente acende e se estiver aberta fica apagado.

Percebeu a analogia entre o circuito e a tabela?

Um exemplo vai ajudar a clarear as coisas.

Descomplicando a Eletrônica Digital para Iniciantes

Suponhamos as chaves S1, S2 e S6 estão fechadas, enquanto S3, S4, S5, S7 e S8 estão abertas.



Pela nossa convenção em que consideramos 1 = *on* e 0 = *off* esta sequência de oito (1 byte) leds acesos e apagados corresponde ao binário 1 1 0 0 0 1 0 0 que é a mesma coisa que o hexadecimal C 4.



Se você está achando estranho utilizar este monte de chaves mecânicas para “construir” um circuito digital, fique sabendo (se não sabe) que tudo começou assim e que estas chaves eram feitas com relés que eram acionados por válvulas eletrônica.

Dá para imaginar o barulhão que fazia um “circuito digital”, além da quantidade de energia que era consumida.

Não se preocupe que não sou tão saudosista assim e, portanto não iremos estudar circuitos digitais valvulados, pois afinal os transistores já foram inventados e é com eles que iremos dar os primeiros passos na Eletrônica Digital.

A situação dos leds acesos e apagados é mostrada na figura abaixo.

Quando um led está aceso temos uma ddp entre seus terminais e quando está apagado a ddp é zero.

Vamos representar estas ddp por uma forma de onda.

Vou supor que você sabe como funciona um transistor e, por isso irei fazer apenas uma breve revisão sobre a aplicação dos **transistores** sendo usados **como chaves lógicas** para substituir as chaves mecânicas do nosso circuito de leds que é o que interessa no momento.

Antes que alguém queira argumentar que seria melhor começar a trabalhar logo com circuitos integrados informo que é importantíssimo começar com os transistores e entender bem o funcionamento deles como chaves lógicas.

Afinal pra que usar um canhão para matar formigas se podemos dar apenas um peteleco nelas!

Descomplicando a Eletrônica Digital para Iniciantes

O transistor bipolar como Chave lógica

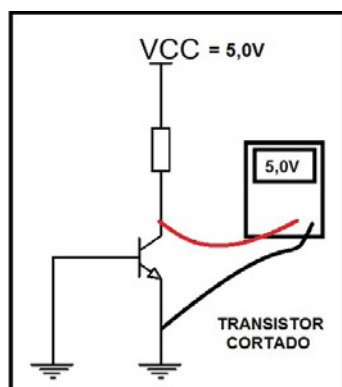
Os transistores bipolares, mais conhecidos simplesmente como **transistores**, sejam eles NPN ou PNP podem ser usados tanto como amplificadores ou como chaves.

A aplicação deles como amplificadores fica no âmbito da Eletrônica Linear, enquanto a Eletrônica Digital utiliza a opção de chave eletrônica.

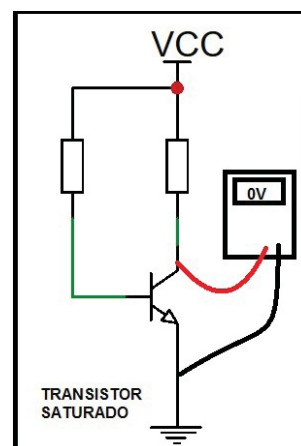
Tudo dependerá basicamente da tensão de polarização entre base e emissor para que o transistor se comporte como amplificador ou chave.

Quando o **transistor** está operando **como chave** definem-se duas condições: **cortado ou saturado**.

Se o transistor está **cortado** não há corrente no coletor e, portanto praticamente **toda a tensão da fonte aparece entre coletor e terra** já que o emissor está ligado à terra.



Se o transistor está **saturado** a **corrente no coletor é máxima** e, portanto há uma grande queda de tensão no resistor de coletor e assim, a **tensão entre coletor e emissor será quase nula**.



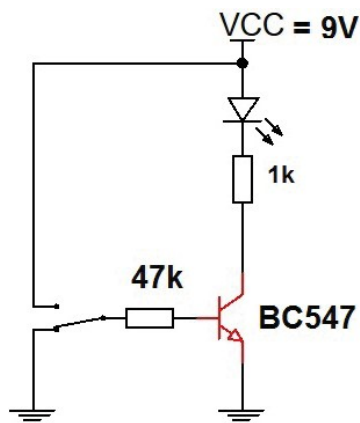
Teoria na prática

Nada melhor que praticarmos para entender a teoria, então mãos à obra, ou melhor, ao circuito.

Você vai precisar de:

- 1 protoboard
- 1 bateria de 9V com clip
- 1 led (qualquer)
- 1 transistor BC547
- 1 resistor 4k7ohms x 1/8 watts
- 1 resistor 47kohms x 1/8 watts e
- 1 multímetro digital.

Descomplicando a Eletrônica Digital para Iniciantes



Monte o circuito ao lado no protoboard e faça a medição da queda tensão entre coletor e emissor (terra) nos dois casos: cortado (led apagado) e saturado (led aceso).

Não é necessário usar uma chave propriamente dita, basta mudar a posição do resistor de 47kohms ligado à base.



Nesta foto temos o transistor cortado com a tensão entre coletor e emissor (ligado à terra) próxima da tensão da alimentação que é uma bateria de 9V.

Embora na foto não dê para perceber, mas neste caso o led está apagado.



Aqui acima temos o transistor saturado e o led aceso.

Observe que a tensão coletor-emissor é praticamente nula.

Se você não tem “intimidade” com circuitos é importante fazer esta experiência para consolidar os conceitos teóricos.

Descomplicando a Eletrônica Digital para Iniciantes

E a tal álgebra de Boole, onde entra?

Talvez você ainda esteja um pouco confuso para entender como os “uns e zeros” da Álgebra de Boole vão se “juntar” a um circuito eletrônico.

Isso é normal, eu também tive estas dúvidas quando comecei a estudar Eletrônica Digital e o pior é que não tinha ninguém para me explicar.

No circuito mostrado anteriormente, onde o transistor é utilizado como uma chave eletrônica, duas coisas merecem destaque.

Quando ligamos a base à terra a tensão de entrada é 0V, mas a saída se aproximou da tensão da fonte.

Vamos dizer isto de outro modo: **quando a entrada está “baixa” a saída fica “alta”** ou na linguagem do nosso amigo Boole, quando a entrada é 0 (lógico) a saída é 1 (lógico).

No nosso exemplo o 1 (lógico) correspondeu a 7,89V (bem próximo dos 9V da fonte).

Quando ligamos o resistor de 47Kohms da base à fonte a tensão no coletor se aproximou de 0V.

Aqui também vamos dizer isto de outra maneira: **quando a entrada está “alta” a saída fica “baixa”** ou na linguagem de Boole, quando a entrada é 1 (lógico) a saída é 0 (lógico).

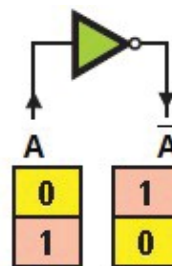
Desta maneira transformamos a teoria da Álgebra de Boole num circuito eletrônico. Fácil, não é?

Para finalizar este capítulo vamos apresentar logo o **circuito inversor**.

Podemos extrair uma característica interessante (e óbvia) do nosso circuito de um transistor e dois resistores que usamos para **inverter** a entrada:

“entrada baixa (0) produz saída alta (1), enquanto entrada alta (1) produz saída baixa (0)”.

Em Eletrônica Digital isto chama-se **circuito inversor**, por razões óbvias e representamos por:



A bolinha no vértice do triângulo representa a inversão e a barra em cima do A também.

Capítulo 1

Portas servem para deixar ou não as pessoas passarem de um lado para outro;

A portas lógicas da eletrônica digital, também embora de um modo “um pouco” diferente.

Não acredita?

Então leia o próximo capítulo.

*Eu disse **LEIA!***

Descomplicando a Eletrônica Digital para Iniciantes



As Portas Lógicas

Not, And, Or, Nand, Nor e etc...

Portas Lógicas é um tema muito importante na Eletrônica Digital, eu diria até que é a base de tudo.

Na vida real uma porta serve para entrar e sair de algum lugar e dependendo de estar aberta ou fechada proibir a entrada ou a saída.

De alguma forma as Portas Lógicas têm uma semelhança com esta ideia como veremos no decorrer do capítulo.

Veremos também a inter conexão entre bits, Portas Lógicas, Lógica e Álgebra de Boole.

No capítulo 2 nós já vimos uma Porta Lógica que é a **porta inversora** também conhecida simplesmente como **NOT**.

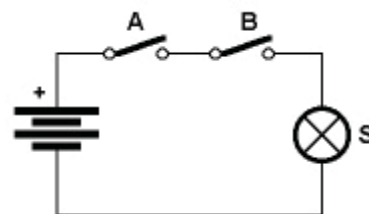
O termo NOT (não) é bem oportuno neste caso porque a porta inversora NEGA a entrada, uma vez que o 1 vira 0 ou vice-versa, o 0 vira 1, como ficou demonstrado no capítulo 2.

Voltemos por um instante à lógica de Arístóteles e re-examinemos a prOposição: “João é brasileiro **E** João gosta de futebol”.

Esta proposição é composta por duas proposições simples que estão ligadas pelo E (ou AND em inglês).

A proposição composta só se tornará verdadeira se cada uma das proposições simples que a formam forem verdadeiras.

Vamos montar um circuito lógico que represente a proposição composta?



Consideremos que a proposição “João é brasileiro” seja representada pela chave A, enquanto “João gosta de futebol” seja a chave B.

Descomplicando a Eletrônica Digital para Iniciantes

Colequemos isto em outros termos. Cada uma das proposições que serão representadas pelas chaves A e B passarão a ser chamadas de **entradas** e a lâmpada será a **saída**.

Assim, a lâmpada que representa a saída só acenderá se a chave A estiver fechada **E** a chave B também estiver fechada tornando a proposição verdadeira.

Então, se João **NÃO** é brasileiro, por exemplo, a chave A deverá ficar aberta e a proposição composta será falsa o que corresponde a lâmpada apagada.

Vamos fazer duas tabelas que representem este circuito lógico que é a porta AND para as quatro situações possíveis de cada proposição:

- João é brasileiro (verdade)
- João **não** é brasileiro (falso)
- João gosta de futebol (verdade)
- João **não** gosta de futebol (falso).

Numa tabela trabalharemos com os conceitos de falso e verdadeiro da lógica aristotélica e na outra com os bits 1 (verdade) e 0 (falso).

As chaves A e B serão as entradas e lâmpada será a saída (esta é a função de uma porta, permitir entrar e sair).

ENTRADAS		SAÍDA
A	B	
V	F	F
F	V	F
F	F	F
V	V	V

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

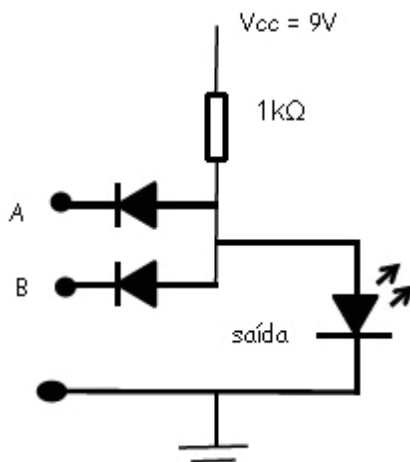
Estas tabelas costumam ser chamadas “tabela verdade” e a que acabamos de mostrar corresponde à Porta Lógica **AND** (“E” em português).

Repare que para proposição composta ser verdadeira é preciso que as duas proposições simples sejam **simultaneamente** verdadeiras.

Em outras palavras, se João for brasileiro, mas não gostar de futebol (como eu!) a segunda proposição fica sendo falsa o que corresponde a chave B aberta e, portanto a lâmpada não acenderá.

Proponho que você construa uma Porta AND de duas entradas utilizando dois diodos (quaisquer), um resistor de 1kohm, um led e uma bateria de 9V.

Descomplicando a Eletrônica Digital para Iniciantes



Observe que ligando à terra o cátodo de qualquer um dos dois diodos que representam as chaves A e B o led apagará, mas deixando os cátodos sem ligação alguma, o que corresponde a um nível alto o led permanecerá aceso, cumprindo-se assim as condições da tabela verdade de uma porta AND.

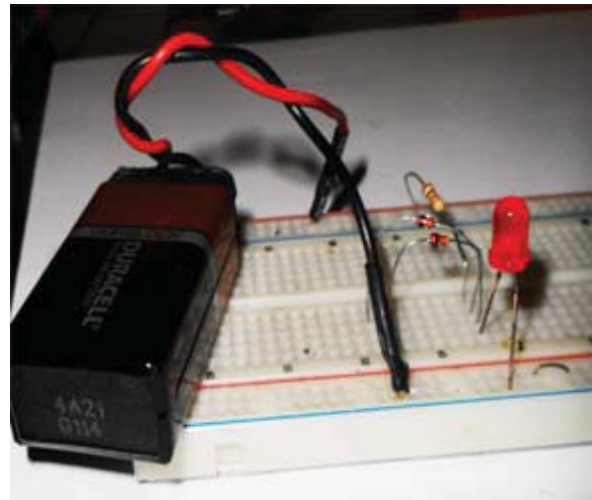
E onde entra a Álgebra de Boole nisto ?

Boole teve a feliz ideia de associar os princípios da lógica aos mesmos princípios utilizados nas operações algébricas (ou aritméticas) básicas como a adição e a multiplicação.

Ele percebeu que a operação lógica AND corresponde a operação algébrica multiplicação se utilizarmos o sistema binário, pois $1 \times 1 = 1$, enquanto

$$1 \times 0 = 0 \times 1 = 0 \times 0 = 0.$$

Usando a “linguagem” da Álgebra de Boole e analisando a tabela verdade podemos escrever a seguinte conclusão importante:



Esta é a maneira mais simples de se construir “eletronicamente” uma porta AND de duas entradas (poderia ser com n entradas) e que é bastante utilizada na prática, desmistificando uma ideia que muitas pessoas têm de que Eletrônica Digital só é feita com circuitos integrados.

A saída de uma porta AND será alta (1) quando **TODAS as entradas forem altas.**

Mas onde se usa isto na prática, deve ser o que você gostaria de perguntar.

As aplicações de uma porta AND só dependem da necessidade e da criatividade do projetista.

Por exemplo, para uma impressora começar a imprimir ela precisa reconhecer que tem papel na bandeja e que tem cartuchos instalados. Certo?

Descomplicando a Eletrônica Digital para Iniciantes

Então estes locais precisam ser monitorados por sensores e ambos terão que enviar um nível lógico a uma porta AND dentro do micro controlador da impressora que tomará as providencias para ela começar a imprimir.

Citei uma impressora como exemplo, mas poderia ser qualquer outro equipamento.

Nem sempre a porta lógica está embutida no micro controlador. Ela pode ser feita externamente e sua saída ser levada ao micro controlador ou outro circuito qualquer que, em função do nível lógico recebido, tomará a decisão.

Antes de prosseguir estudando outras portas lógicas gostaria de fazer um comentário.

Uma coisa interessante na Eletrônica Digital é que, a meu ver, a torna mais fácil de estudar em comparação com a Eletrônica Linear é que, na maioria das vezes, não precisaremos trabalhar com componentes discretos como diodos, transistores, resistores e etc.

Na Eletrônica Digital trabalharemos com blocos lógicos.

Por exemplo, uma porta AND de duas entradas como a que

acabamos de estudar será representada apenas pelo símbolo mostrado abaixo junto com a sua respectiva tabela verdade.

Não precisamos saber o que tem dentro da “caixa preta” representada pelo símbolo mostrado, nem tão pouco como o circuito que está lá dentro funciona e sim entender a tabela verdade.



O que é **obrigatório** saber é o que está no quadro abaixo.

A saída de uma porta AND será alta (1) quando TODAS as entradas forem altas.

Repare que eu disse **TODAS** as entradas.

Do ponto de vista da reparação de um equipamento, se encontramos a saída de uma porta AND em nível baixo é porque pelo menos uma de suas entradas também deve estar em nível baixo.

O que se deve fazer a seguir é

Descomplicando a Eletrônica Digital para Iniciantes

verificar cada uma das entradas e se constatar que todas estão em nível alto você acabou de descobrir que esta porta está defeituosa ou há algum componente “pendurado” em sua saída que está em curto

(um capacitor, por exemplo).

Viu como é mais fácil que num circuito linear. Você não precisa ficar testando componentes um por um.

Mais uma Porta Lógica: OR

Vamos pegar nossa proposição composta usada anteriormente e trocar o E por OU (em inglês é OR).

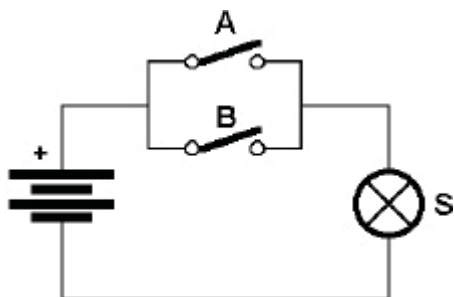
Ela vai ficar assim:

João é brasileiro **OU** João gosta de futebol.

Neste caso para que a proposição composta se torne verdadeira basta que uma das duas proposições simples que a formam seja verdadeira.

E como seria um circuito que a representaria?

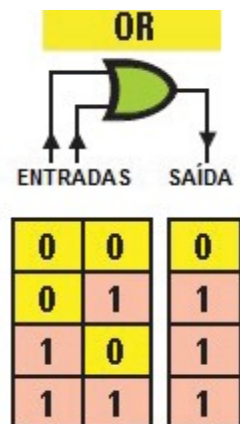
O que você acha deste aí embaixo?



Percebeu a diferença na posição das chaves em relação a porta AND? (lembra do jogo dos 7 erros?).

As chaves agora estão em paralelo, antes estavam em série e assim, mesmo que uma das chaves esteja aberta (0) a lâmpada ainda acenderá.

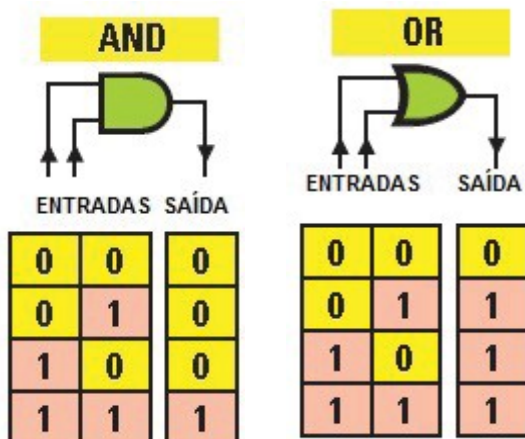
Da mesma forma que fizemos com a porta AND iremos utilizar um símbolo para representar a porta OR como vemos abaixo junto com a respectiva tabela verdade.



Mais uma vez convido-o ao joguinho dos 7 erros. Compare os símbolos das portas OR e AND e veja qual a diferença. Aproveite também e analise a diferença entre as duas tabelas verdade.

Isso é tão importante que eu vou repetir uma ao lado da outra para lhe ajudar.

Descomplicando a Eletrônica Digital para Iniciantes



Comparação entre as portas AND e OR.

Observou que a diferença nos símbolos é que na porta AND temos uma linha reta na entrada e na porta OR a linha é curva?

Sobre a tabela verdade podemos escrever uma conclusão parecida com aquela que fizemos para a porta AND.

A saída de uma porta OR será baixa (0) quando TODAS as entradas forem baixas.

Não para por aí, vem aí o Exclusive-OR

Se você parar para pensar (sei que nem todo mundo gosta de fazer isso!) a porta OR que nós estudamos tem “embutida” uma característica da porta AND, pois se João for brasileiro E também gostar de futebol a proposição ainda continua válida e isso, sob o ponto de vista da lógica, não é muito bom e pode trazer confusão.

Para resolver este probleminha criou-se a porta Exclusive OR (**ou exclusivo**) que costuma ser identificada como XOR.

Neste caso a proposição composta só será considerada verdadeira se apenas uma e somente uma das proposições simples for verdadeira.

O **ou-exclusivo** se aproxima mais do **ou** usado na linguagem,

pois quando dissemos, por exemplo, “não sei se vou ao teatro **ou** ao cinema” não estamos cogitando a possibilidade de ir aos dois lugares.

Vejamos a simbologia e a tabela verdade da porta **XOR**.



Nunca é demais lembrar que você deve praticar aquele joguinho e observar a diferença entre a simbologia da porta OR e da XOR, bem como entre as respectivas tabelas verdade.

Descomplicando a Eletrônica Digital para Iniciantes

Teoria na prática

Para consolidar o que foi estudado sobre a porta lógica XOR vamos apresentar um circuito com chaves que permita se obter as condições de sua tabela verdade.

Observe o circuito ao lado e veja se ele cumpre a tabela verdade da porta XOR.

O próximo passo será construir uma porta OR com diodos, da mesma forma que fizemos com a porta AND.

Se as entradas A e B forem deixadas sem ligação (abertas) equivale a nível lógico alto ou 1.

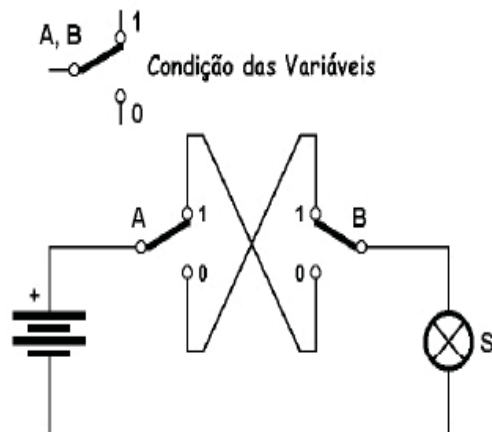
Para que o led da saída apague, indicando 0, precisamos ligar A e B à terra simultaneamente.

De Volta a Álgebra de Boole

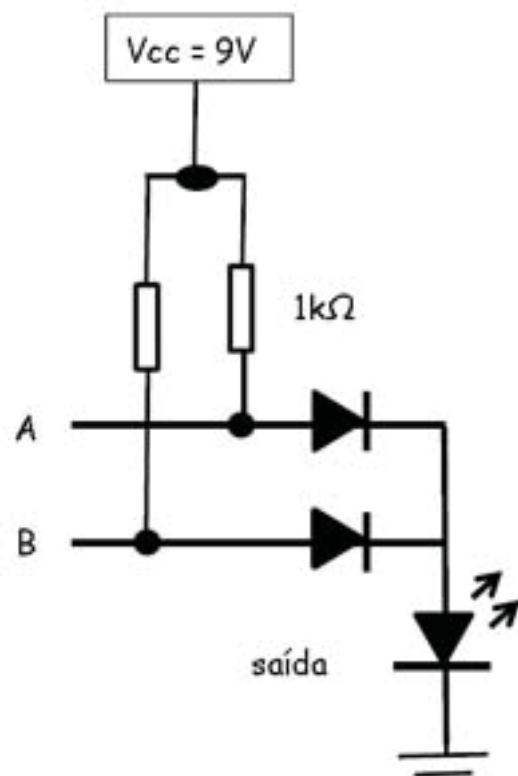
Lembra que nós vimos lá atrás que o Boole associou a porta AND com a operação de multiplicação de números binários?

Pois bem, de forma similar ele também associou a porta OR a adição de binários.

Veja os resultados e compare com a tabela verdade da porta OR.



Porta XOR (ou-exclusivo) construída com chaves mecânicas.



Porta XOR (ou-exclusivo) construída com diodos.

Descomplicando a Eletrônica Digital para Iniciantes

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

Ops! $1 + 1$ não é igual a 2?



Sim, se estivessem trabalhando na base 10, mas aqui estamos na base 2 e nela não existe o dígito 2.

Então como fazemos esta conta?

Embora num primeiro momento não seja muito importante saber isto, vou matar a sua curiosidade.

Lembra quantos números podemos escrever com apenas dois bits (0 e 1)?

Vou reativar sua memória, está lá na **página 15** do capítulo 1. Acompanhe na tabela abaixo.

BINÁRIO	DECIMAL
00	0
01	1
10	2
11	3

A operação é semelhante a que fazemos na base 10. Quando somamos, por exemplo, $9 + 1$ encontramos “zero e vai 1” que

será escrito como 10.

Aqui é a mesma coisa, $1 + 1$ dá “zero e vai 1” que nós dá 10 (binário) e que corresponde ao 2 decimal.

Mas como eu disse você não precisa se aprofundar nisto neste momento.

A maioria dos livros de Eletrônica Digital, senão todos, gasta um tempo imenso com este assunto que eu considero enfadonho e de pouco interesse para quem quer estudar eletrônica propriamente dita.

Eu não vou seguir a mesma receita, vou ser “revolucionário” (como pediu o Papa Francisco) e quebrar a tradição me preocupando mais com os circuitos e do que com a matemática.

Entretanto, advirto-lhe, se você pretende se preparar para concursos na área de eletrônica terá que estudar mais a fundo a aritmética binária porque questões sobre ela são “figurinhas carimbadas” em todos eles.

Feitas estas observações retornemos as “nossas” portas lógicas.

Até aqui estudamos quatro tipos de portas, a saber: NOT, AND, OR e XOR, mas ainda faltam mais três para completar o “time” das portas lógicas que nos permitirão construir TUDO que quisermos em Eletrônica Digital.

Estas três “novas” portas que estudaremos serão: **NAND**, **NOR** e **XNOR** onde **N** que apareceu em cada uma delas significa NOT e nada mais é do que a negação da AND, OR e XOR.

Descomplicando a Eletrônica Digital para Iniciantes

As simbologias serão as mesmas apresentadas até aqui com o acréscimo de uma bolinha na saída para indicar a negação da mesma forma que foi feita com a porta NOT.

A seguir nós veremos as simbologias e tabelas verdade destas três portas, mas porém com um andendo a mais. as portas NAND, e NOR serão apresentadas com três entradas em vez de duas como fizemos anteriormente.

Fiz isto para que você não fique pensando que não podem existir portas com mais de duas entradas, entretanto há uma exceção: **as portas XOR e XNOR só podem ser construídas com duas entradas.**

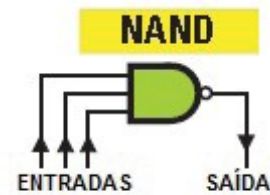
Façamos algumas reflexões sobre as tabelas verdade destas duas portas comparando com as tabelas das portas AND e OR de duas entradas apresentadas anteriormente.

Relembremos que para uma porta AND independente do número de portas que ela tenha vale

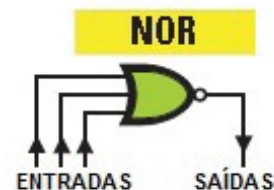
A saída de uma porta AND será alta (1) quando **TODAS as entradas forem altas.**

Mas como a porta NAND é a negação da porta AND, então quando **TODAS** as entradas forem altas a saída deverá ser ... baixa.

Um raciocínio similar pode ser feito para as demais portas e você deve perder um tempinho fazendo estas observações.



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

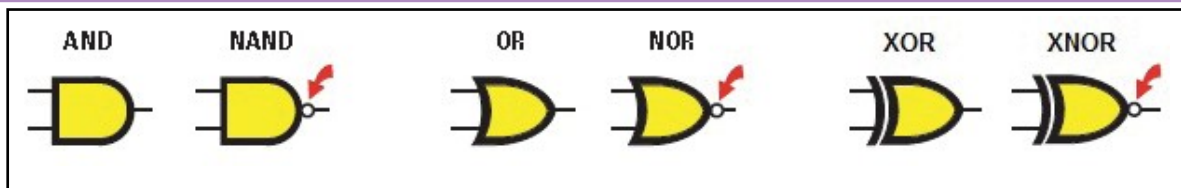


0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Dever de Casa

Completar as tabelas verdade de cada uma das portas mostradas na próxima página.

Descomplicando a Eletrônica Digital para Iniciantes

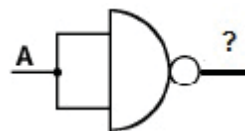


ENTRADAS			SAÍDA	
			AND	NAND
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

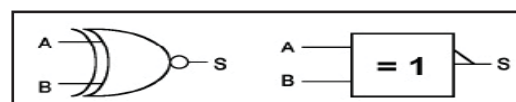
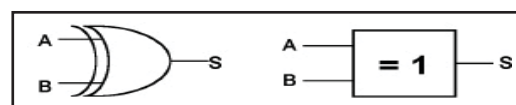
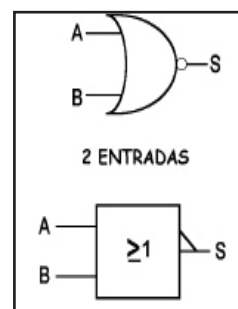
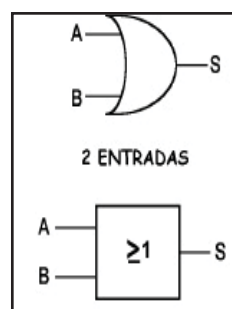
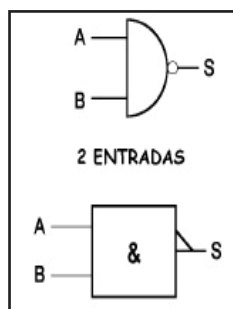
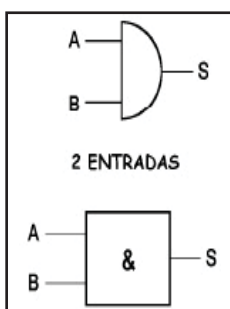
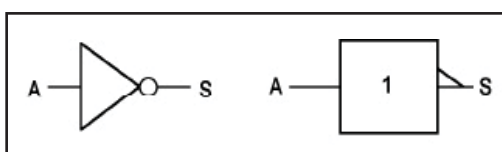
ENTRADAS			SAÍDA	
			OR	NOR
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

ENTRADAS		SAÍDA	
		XOR	XNOR
0	0		
0	1		
1	0		
1	1		

O que acontecerá se ligarmos todas as entradas de uma porta NAND em uma só, como neste exemplo de uma porta NAND de duas entradas?



Outras simbologias para as Portas Lógicas

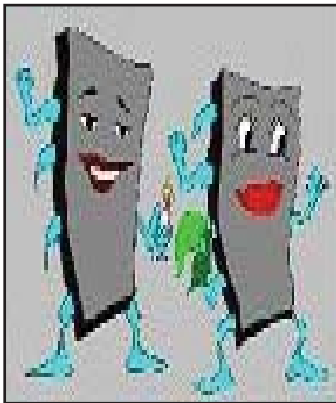


Capítulo 4

No dicionário Aurélio temos várias definições para a palavra família. A mais popular é “conjunto de todos os parentes de uma mesma pessoa”.

Uma outra é “grupo de animais, vegetais, minerais que têm caracteres comuns”

Esta definição é a que mesma se aplica as famílias lógicas como você verá ao ler o capítulo a seguir.



Circuitos Integrados & Famílias Lógicas

Nos capítulos anteriores tratamos de uma parte da Eletrônica Digital cujo conhecimento é fundamental e indispensável para qualquer um que queira trabalhar na área de Eletrônica, quer seja como projetista ou reparador.

Todos os conceitos até aqui abordados precisam estar bem consolidados, pois formam a base de toda a Eletrônica Digital qualquer que seja a evolução da tecnologia.

As sete portas lógicas estudadas sempre estarão presentes mesmo que estejam embutidas em um “chipão” e não se tenha acesso diretamente a elas.

Entretanto, nem só de micro controladores vive a Eletrônica e muitas vezes recorreremos a circuitos integrados (CI's) que realizam individualmente as funções das portas lógicas e, por sua vez, estes CIs são construídos com transistores, diodos e resistores.

De acordo com a tecnologia

utilizada para fabricação destes CI's eles são agrupados no que se costuma chamar de **famílias de circuitos lógicos** ou simplesmente **famílias lógicas**.

Mostrei no capítulo 2 como fazer uma porta AND ou OR usando apenas diodos e resistores.

Todavia este tipo de construção, até onde eu sei, não parece ter sido fabricada em forma de circuito integrado, sendo utilizada apenas com componentes discretos.

As primeiras famílias lógicas propriamente ditas foram as RTL (Resistor Transistor Logic) e DTL (Diode Transistor Logic) completamente obsoletas e seus CI's só serão encontrados em equipamentos muito antigos (década de 60), por isso não serão estudadas aqui.

Em 1962 a Texas Instruments introduziu no mercado a família TTL - Transistor Transistor Logic.

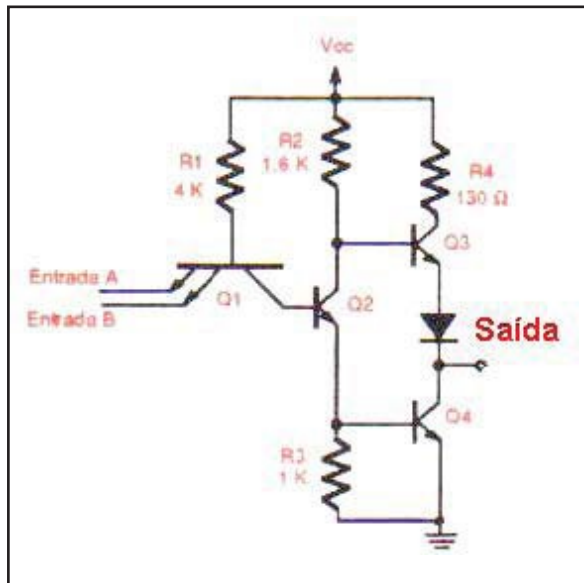
Descomplicando a Eletrônica Digital para Iniciantes

Eu arriscaria dizer que os TTL's constituem um divisor de águas na Eletrônica Digital.

A partir deles, milhares de projetos digitais começaram a surgir. Foi uma verdadeira "febre".

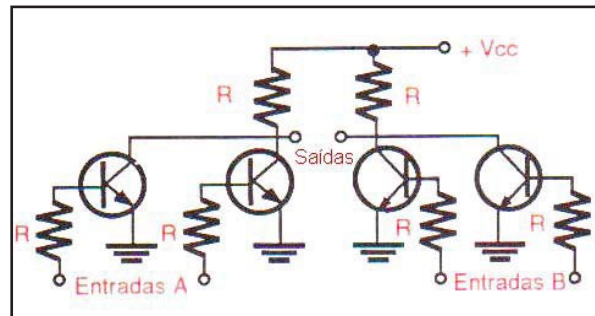
A principal característica dos TTL's é a entrada constituída por transistores bipolares com dois ou mais emissores.

Abaixo podemos ver o circuito de uma porta NAND da família TTL.

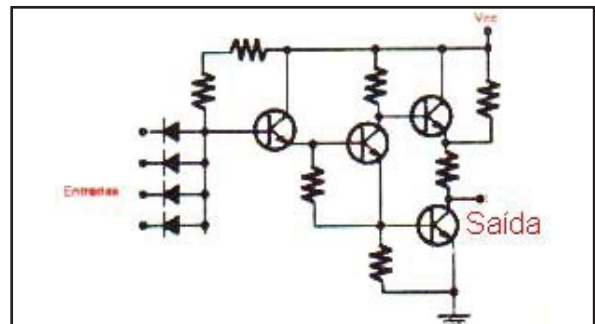


Você pode estar achando contraditório eu apresentar o "interior" do circuito de uma porta lógica como esta, uma vez que eu disse lá atrás que uma das vantagens da Eletrônica Digital era que não precisaríamos nos preocupar com isto.

E não precisaremos mesmo, este circuito foi apresentado apenas



Porta RTL - Obsoleta



Porta DTL - Obsoleta

como mera curiosidade.

Na prática teremos apenas que saber qual a função de cada pino do CI e as características elétricas que serão estudadas neste capítulo.

Os CI's da família TTL introduzidos no mercado inicialmente pela Texas lá pelo idos de 1962 hoje são fabricados por outras empresas.

A Texas criou dois grupos de CI's para a família TTL cuja nomenclatura começa com 54 ou 74.

Os CI's de prefixo 54 são para aplicações militares, portanto eletricamente mais robustos.

Já os CI's da família 74 são de

Descomplicando a Eletrônica Digital para Iniciantes

uso comercial e embora ainda hoje, como eu disse, sejam encontrados para comprar não são mais os preferidos para novos projetos.

Entretanto, como os CI's da família 74 reinaram absolutos por muitos anos é importante que estudemos suas características, uma vez que a tecnologia TTL, ou **padrão TTL**, como se costuma dizer, ainda está presente em muitos circuitos integrados atuais.

Uma das desvantagens dos primitivos TTL's era o excessivo consumo de corrente, item que já foi superado graças ao avanço na tecnologia dos semicondutores.

Esta evolução trouxe uma outra família de circuitos lógicos, introduzida no mercado pela RCA em 1968 e conhecida pela sigla CMOS que são as iniciais de Complementary-MOS donde se conclui que seus integrados, em vez de utilizarem os transistores bipolares dos TTL's, utilizam transistores MOS (Metal Oxide Semiconductor).

A principal vantagem dos CMOS sobre os TTL's era um consumo de energia bem menor a despeito de ter uma velocidade de processamento mais baixa, um problema que também foi superado ao longo do tempo.

O que é o "padrão TTL" ?

A primeira coisa que você precisa **obrigatoriamente** saber é que os integrados da família TTL, da série 74, **só podem ser alimentados com 5 volts** admitindo uma tolerância de + ou - 5% neste valor, ou seja, podem trabalhar com tensões de alimentação entre 4,75 e 5,25V.

Tensão de alimentação abaixo de 4,75V fará com a porta não funcione corretamente, enquanto acima de 5,25V levará o CI a destruição.

Este é um ponto importante para o qual não só o projetista deve

estar atento, mas principalmente o reparador.

Vale ressaltar que é fundamental que o multímetro que está sendo utilizado para medições seja confiável em face da pequena tolerância admitida no valor da tensão de alimentação dos CI's.

Entretanto, não é só com a tensão de alimentação que devemos nos preocupar. Outra questão importante trata dos níveis de tensão que o padrão TTL aceita como níveis lógicos 1 e 0.

Você já aprendeu que os

Descomplicando a Eletrônica Digital para Iniciantes

chamados bits representados pelos binários 1 e 0 só existem do ponto de vista teórico da álgebra de Boole, mas na prática o circuito trabalha mesmo é com tensões.

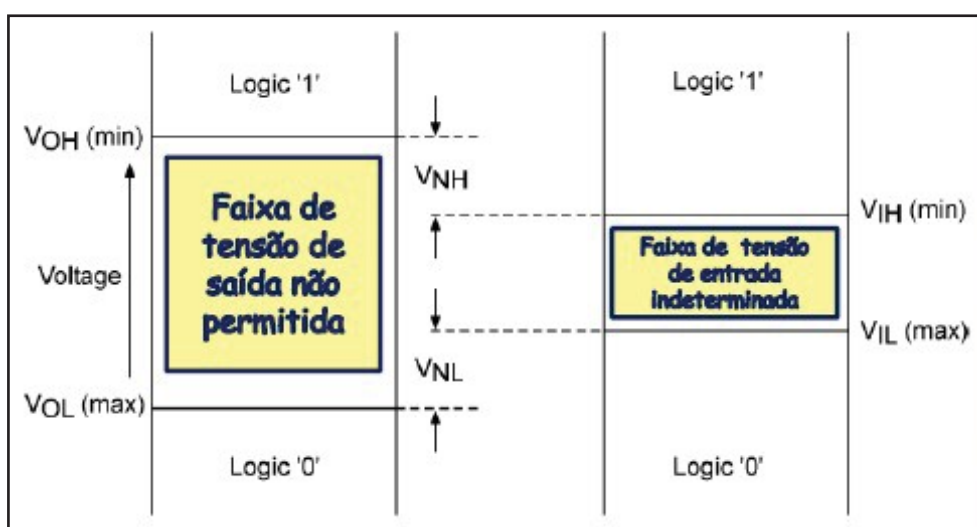
Para o projetista que está desenvolvendo a lógica do seu circuito é muito mais fácil trabalhar com bits. Cabe, porém a quem vai transformar o projeto em realidade saber quais os níveis de tensão que seus CI's irão interpretar como

os tais bits.

Por outro lado o reparador precisa saber quais são os valores destas tensões para que possa constatar se o circuito está funcionando corretamente ou não.

Anomenclatura padronizada para estas tensões que encontramos nos livros e *data sheets* é mostrada na tabela abaixo.

$V_{IH(min)}$	Tensão de Entrada Nível Alto
$V_{IL(max)}$	Tensão de Entrada Nível Baixo
$V_{OH(min)}$	Tensão de Saída Nível Alto
$V_{OL(max)}$	Tensão de Saída Nível Baixo
I_{OL}	Corrente de Saída em Nível Alto
I_{OH}	Corrente de Saída em Nível Baixo
I_{IH}	Corrente de Entrada em Nível Alto
I_{IL}	Corrente de Entrada em Nível Baixo



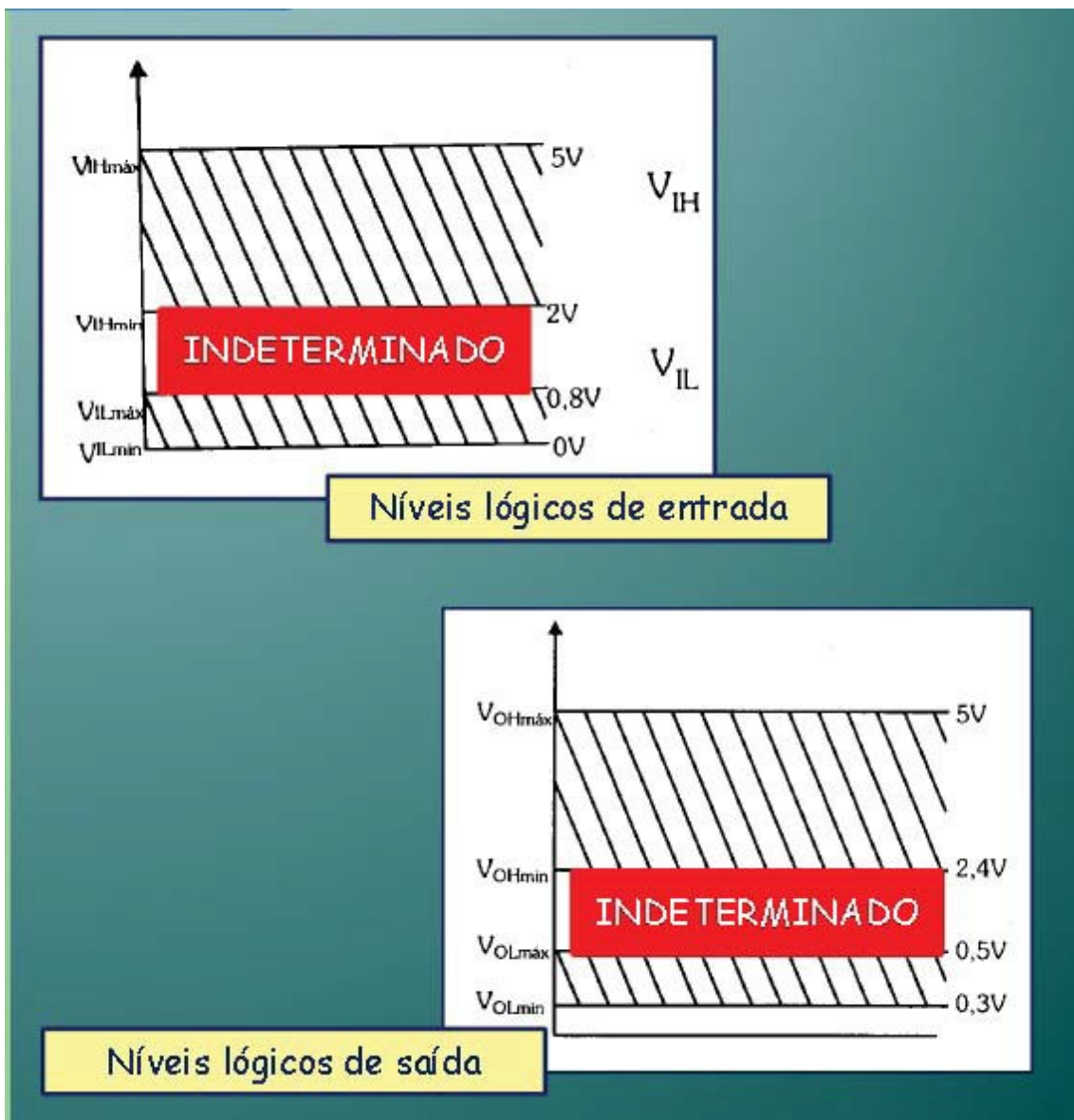
Descomplicando a Eletrônica Digital para Iniciantes

No caso dos CI's da família TTL é **importantíssimo** saber quais são os valores máximos e mínimos de tensão de entrada e saída que serão interpretados como níveis lógicos corretos.

No gráfico abaixo temos estes valores e você deverá estar sempre atento a eles, principalmente quando

estiver reparando um equipamento com CI's que trabalham com o padrão TTL.

Observe que há uma faixa de valores denominada **INDETERMINADO** que significa que se tivermos alguma tensão dentro de uma destas faixas o circuito não funcionará corretamente.



Descomplicando a Eletrônica Digital para Iniciantes

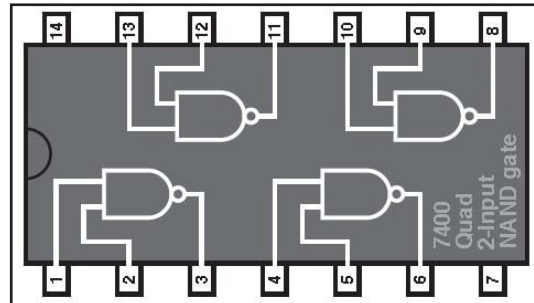
Teoria na prática

O objetivo desta experiência é que você comece a se familiarizar com a utilização de circuitos integrados e como até aqui só falamos da família TTL vou escolher um dos seus representantes mais comuns, o SN7400N, que é um conjunto de quatro portas NAND de duas entradas cada uma.

Embora não seja uma regra geral, na maioria dos CI's da família TTL a alimentação é feita pelos pinos 7 (GND) e 14 ($V_{cc} = 5V$).

A primeira experiência será para comprovar a tabela verdade e para tal utilizaremos um LED na saída e duas chaves nas duas entradas.

No esquema vemos dois resistores de 10kohms em cada entrada (pinos 1 e 2) e os mesmos foram ligados ao barramento de V_{cc}

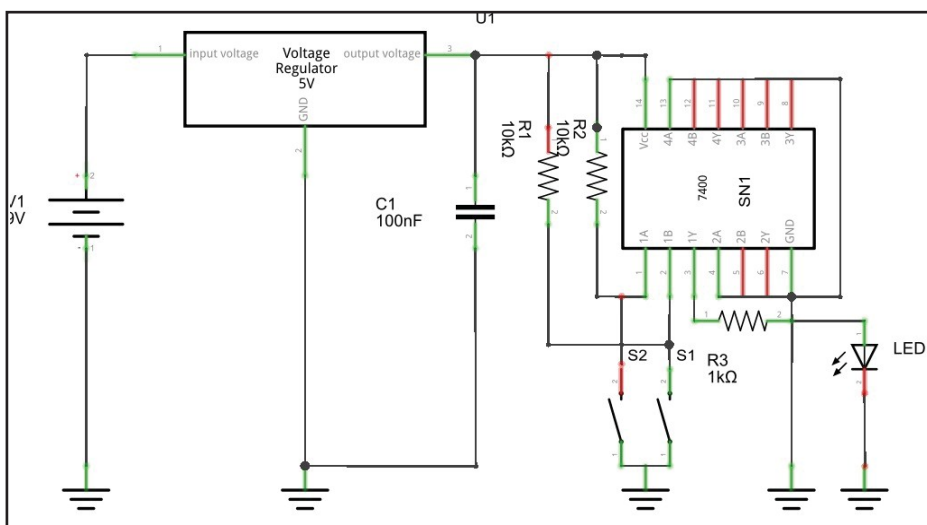


Chanfro que indica o início da contagem dos pinos.



A finalidade destes resistores é não deixar que as portas fiquem “abertas” e sem nenhuma referência de tensão o que pode causar instabilidade no circuito. Eles são chamados resistores de **pull up**.

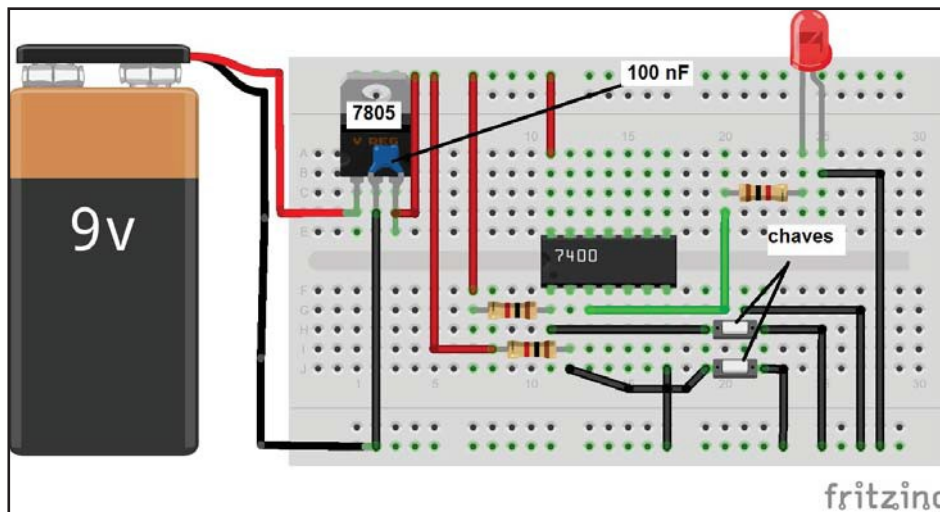
Para alimentação do circuito vamos utilizar uma bateria de 9V com um regulador de tensão 7805 para se conseguir a tensão correta para alimentar o CI.



A montagem será feita num *protoboard* como se vê na página seguinte.

No lugar das chaves pode-se fazer a conexão com fios.

Descomplicando a Eletrônica Digital para Iniciantes



Antes de passarmos para outro assunto precisamos fazer um comentário.

Você notou que no esquema da página anterior todas as entradas e saídas não utilizadas foram ligadas à terra?

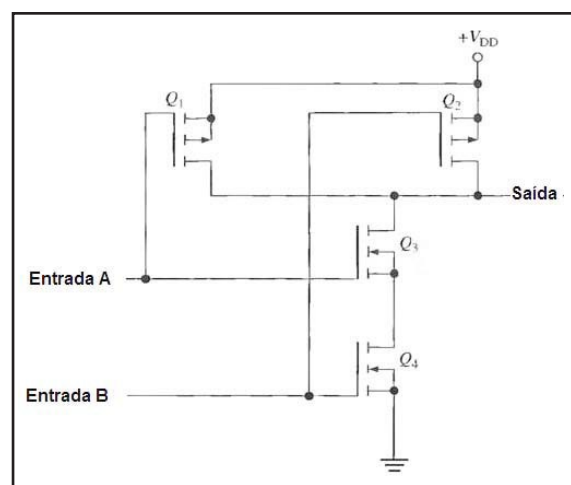
Este é um cuidado que deve ser tomado quando se constrói um circuito para evitar que ruídos espúrios possam prejudicar o funcionamento do memo.

Na nossa montagem do *protoboard* não fizemos isto porque neste caso a finalidade é apenas fazer com que você se familiarize com as portas.

E a família CMOS ?

Como já foi mencionado lá atrás a principal diferença entre a família TTL e a CMOS é que esta usa tecnologia de transistores MOS em lugar dos bipolares o que significa menor consumo de energia.

Na figura ao lado podemos ver, apenas como curiosidade, como é a construção de uma porta NAND CMOS, mas não se assuste porque não precisaremos entender como o circuito funciona.



Descomplicando a Eletrônica Digital para Iniciantes

Entretanto, não é apenas a tecnologia de construção que muda e o que importa mais para o técnico é saber qual a tensão de alimentação e as tensões dos níveis lógicos.

Uma grande vantagem dos CMOS sobre os TTL's é a flexibilidade no valor da tensão de alimentação que não precisa mais ser os rígidos 5 volts dos TTL's.

Os CMOS podem trabalhar com alimentação entre 3 e 15 volts (em alguns casos pode chegar a 18V) o que, sem dúvida, aumenta a flexibilidade do projeto.

Antes de prosseguir vale chamar a atenção para um fato pouco divulgado, mais de grande importância, principalmente, para o reparador.

Embora os CMOS tenham uma nomenclatura padronizada como CD 4xxx, às vezes, diferentemente dos TTL's, podem apresentar variações dos parâmetros de um fabricante para outro.

Sendo assim, ao substituir um CMOS numa reparação **talvez** seja necessário fazê-lo por outro que além de ter o mesmo código (é claro) seja **do mesmo fabricante**.

Os níveis lógicos do CMOS

Da mesma forma que fizemos com os TTL's precisamos saber quais os níveis de tensão nos CMOS que serão considerados com níveis lógicos alto (high) e baixo (low) ou, respectivamente bits 1 e 0.

Uma vez que a tensão de alimentação de um CMOS pode variar entre 3 e 15V, isso implica que os valores de tensão dos níveis lógicos fiquem atrelados a alimentação diferentemente do que acontece com os TTL's.

No caso dos CMOS utiliza-se a seguinte regra:

1) Nível lógico high (H) entre 2/3 do valor de V_{DD} até, no máximo o valor de V_{DD} .

(2) Nível lógico low (L) entre 0V e 1/3 do V_{DD} .

Vamos entender melhor isto através de um exemplo.

Suponhamos que a tensão de alimentação escolhida para um circuito com portas CMOS seja 12V.

Como 2/3 de 12 é igual a 8 e 1/3 é igual a 4, os valores de tensão para nível alto deverão ficar entre 8 e 12V e para nível baixo entre 0 e 4V.

Descomplicando a Eletrônica Digital para Iniciantes

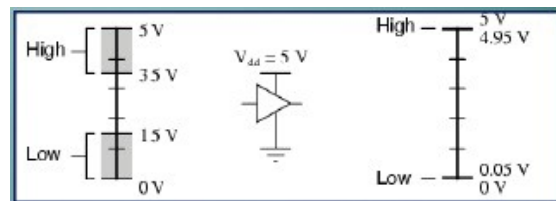
Na prática estes valores aceitam uma pequena variação de 0,05V.

Assim, no nosso exemplo, o valor máximo de tensão de nível alto seria 11,95V e o valor mínimo para nível baixo será 0,05V.

Vejamos uma situação em que fossem utilizadas simultaneamente as duas tecnologias, TTL e CMOS.

Neste caso deveríamos utilizar 5V para alimentação de todos os CI's por causa dos TTL's.

A figura abaixo mostra como ficariam os níveis lógicos para os CMOS.



O CMOS na Informática

Mesmo que você não seja “técnico” de informática certamente já ouviu falar de CMOS quando se refere a computador.

Realmente esta associação do “nosso” CMOS com a do computador procede.

A designação de CMOS na informática é referente a uma pequena memória RAM (mais a frente estudaremos o que é isso) construída com a tecnologia MOS.

Antigamente esta memória gravava alguns dados de configuração do BIOS (*Basic Input/Output System*).

Atualmente a principal função desta memória é o armazenamento da hora e da data que são atualizados pelo RTC (Real Time Clock).



Por se tratar de uma memória RAM, ela é volátil e por isso, necessita de uma bateria para alimentá-la quando o computador é desligado.

Quando esta bateria está “fraca”, toda vez que ligamos o computador notamos que o relógio está desatualizado.



Descomplicando a Eletrônica Digital para Iniciantes

Este material para por aqui por enquanto e foi entregue a você **gratuitamente** por que é aluno da área VIP do Clube Aprenda Eletrônica com Paulo Brites. Peço que não o passe adiante porque ele fará parte de um futuro e-book de Eletrônica Digital que estou escrevendo.

Muito obrigado