

LIÇÃO 3

FAMÍLIAS DE CIRCUITOS LÓGICOS DIGITAIS

Na lição anterior conhecemos os princípios simples da Álgebra de Boole que regem o funcionamento dos circuitos lógicos digitais encontrados nos computadores e em muitos outros equipamentos. Vimos de que modo umas poucas funções simples 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 funções 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 de operações muito complexas a cada segundo.

Nesta edição veremos que tipo de circuitos são usados e como são encontrados na prática em blocos básicos que unidos podem levar a elaboração de circuitos muito complicados como os encontrados nos computadores.

O leitor irá começar a tomar contato com componentes práticos das famílias usadas na montagem dos equipamentos digitais. São estes os componentes básicos que podem ser encontrados em circuitos digitais, computadores e muitos outros.

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 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, atuando com suas mãos sobre a chave, deveria estabelecer o nível lógico de entrada, mantendo esta chave aberta ou fechada conforme desejasse 0 ou 1.

Se usarmos um transistor teremos uma vantagem importante: o transistor poderá operar com a tensão ou nível lógico produzido por uma outra função e não necessariamente por uma pessoa que acione 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, conforme a **figura 1**.

Na figura 1 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 outro 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 1.

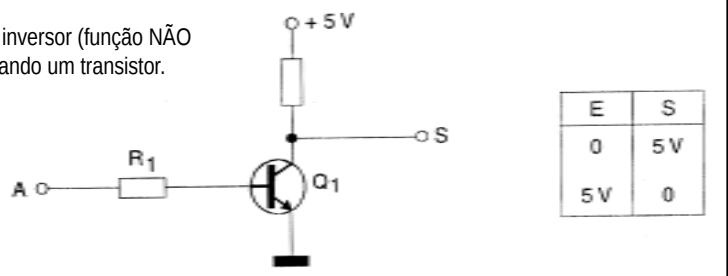
Conforme observamos na **figura 2**, outras funções podem ser conseguidas com transistores.

Isso significa que a elaboração de um circuito lógico digital capaz de realizar operações complexas usando transistores é algo que pode ser conseguido com relativa facilidade.

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.

Figura 1 - Um inversor (função NÃO ou NOT) usando um transistor.



Dessa forma, se bem que nos primeiros tempos da Eletrônica Digital cada função era montada com seus transistores, diodos e resistores na sua plaquinha para depois serem todas interligadas, este procedimento se revelou inconveniente por diversos motivos.

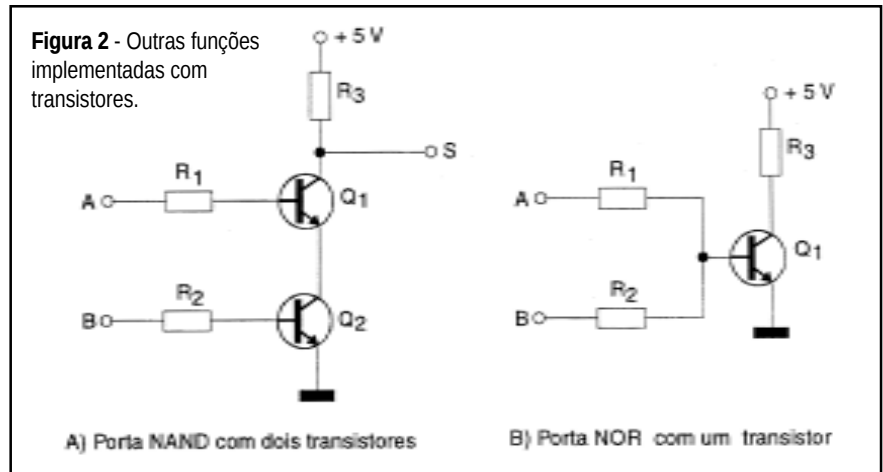
O primeiro deles é a complexidade de que o circuito adquiria se realizasse muitas funções.

O segundo, é a necessidade de padronizar o modo de funcionamento de cada circuito ou função. Seria muito importante estabelecer que todos os circuitos operassem com a mesma tensão de alimentação e fornecessem sinais que os demais pudessem reconhecer e reconhecessem os sinais gerados pelos outros.

O desenvolvimento da tecnologia dos circuitos integrados, possibilitando a colocação num único invólucro de diversos componentes já interligados, veio permitir um desenvolvimento muito rápido da Eletrônica Digital.

Foi criada então uma série de circuitos integrados que continham numa única pastilha as funções lógicas digitais mais usadas e de tal maneira projetadas que todas eram compatíveis entre si, ou seja, operavam com as mesmas tensões e reconheciam os mesmos sinais.

Figura 2 - Outras funções implementadas com transistores.



Estas séries de circuitos integrados formaram então as Famílias Lógicas, a partir das quais os projetistas tiveram facilidade em encontrar todos os blocos para montar seus equipamentos digitais.

Assim, conforme a **figura 3**, precisando montar um circuito que usasse uma porta AND duas NOR e inversores, o projetista teria disponíveis componentes compatíveis entre si contendo estas funções e de tal forma que poderiam ser interligadas das maneiras desejadas.

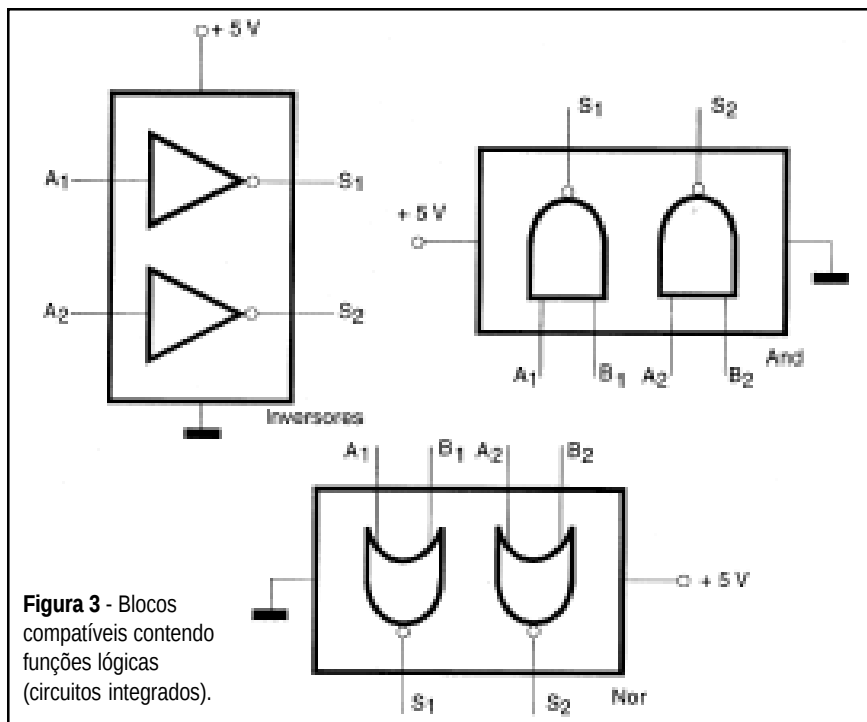
O sucesso do advento dessas famílias foi enorme, pois além do menor tamanho dos circuitos e menor consumo de energia, havia ainda a

vantagem do menor custo e obtenção de maior velocidade de operação e confiabilidade.

Diversas famílias foram criadas desde o advento dos circuitos integrados, recebendo 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
- CMOS ou Complementary Metal Oxid Semiconductor
- ECL ou Emitter Coupled Logic



Atualmente a Família TTL e a CMOS são as mais usadas, sendo empregadas em uma grande quantidade de equipamentos digitais e também nos computadores e periféricos.

3.3 - A família TTL

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

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 rapidamente associar qualquer componente que começa pelo número "74" à família TTL.

Na **figura 4** 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.

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

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.

Para os níveis lógicos serem reconhecidos devem estar dentro de faixas bem definidas.

Conforme verificamos na **figura 5**, uma porta TTL reconhecerá como nível 0 as tensões que estiverem entre 0 e 0,8 V e como 1 os que estiverem numa outra faixa entre 2,4 e 5 V.

Entre essas duas faixas existe uma região indefinida que deve ser evitada.

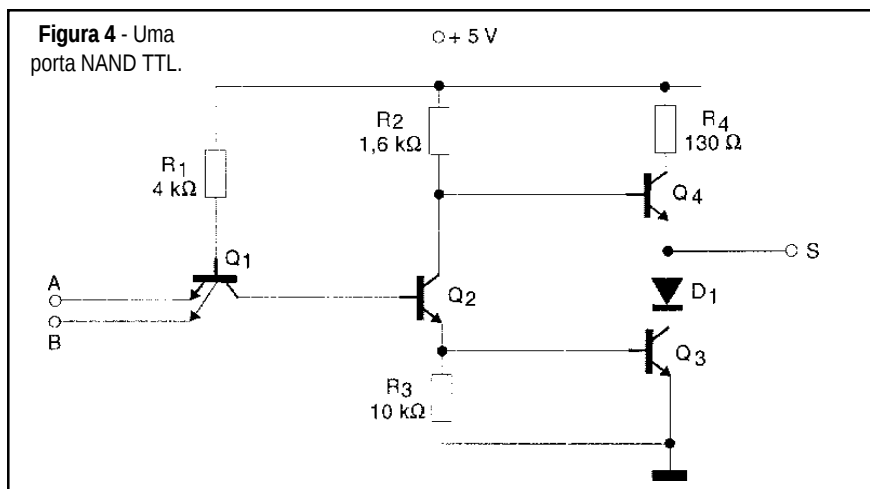
Há centenas de circuitos integrados TTL disponíveis no mercado para a realização de projetos. A maioria deles está em invólucros DIL de 14 e 16 pinos, conforme exemplos da **figura 6**.

As funções mais simples das portas disponíveis numa certa quantidade em cada integrado usam 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 colocar num integrado não apenas umas poucas portas e funções adicionais que serão estudadas futuramente como *flip-flops*, decodificadores e outros mas, também interligá-los de diversas formas e utilizá-los em aplicações específicas.

Diversas etapas no aumento da integração foram obtidas e receberam nomes que hoje são comuns quando

Figura 4 - Uma porta NAND TTL.



falamos de equipamentos digitais e computadores em geral. Temos 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 que contém de 1 a 12 portas lógicas num mesmo componente ou circuito integrado.

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 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 com mais de 1000 portas ou funções lógicas.

3.4 - Outras Características da Família TTL

Para usar corretamente os circuitos integrados TTL e mesmo saber como 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 admitidos:

- 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, **figura 7**.

Esta corrente deve ser levada em conta em qualquer projeto, pois, ela deve ser suprida pelo circuito que excitará a porta.

Quando a entrada de uma porta lógica TTL está no nível alto, figura 8, flui uma corrente no sentido oposto da ordem de 40 μ A.

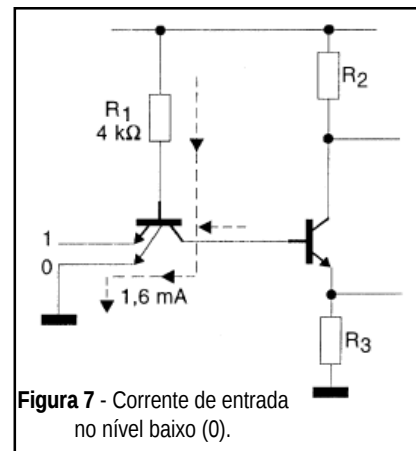
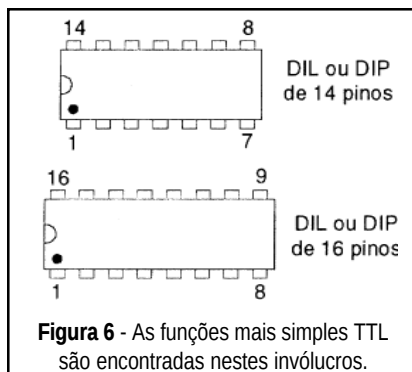
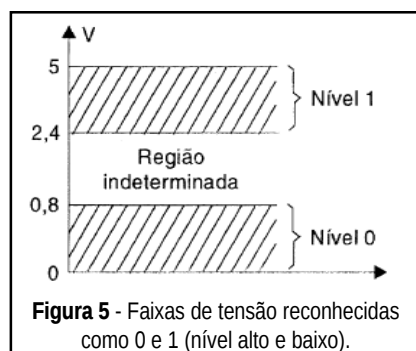
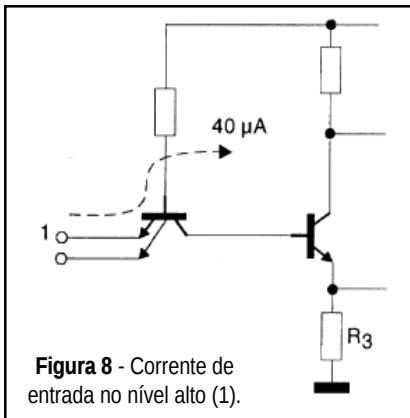


Figura 7 - Corrente de entrada no nível baixo (0).





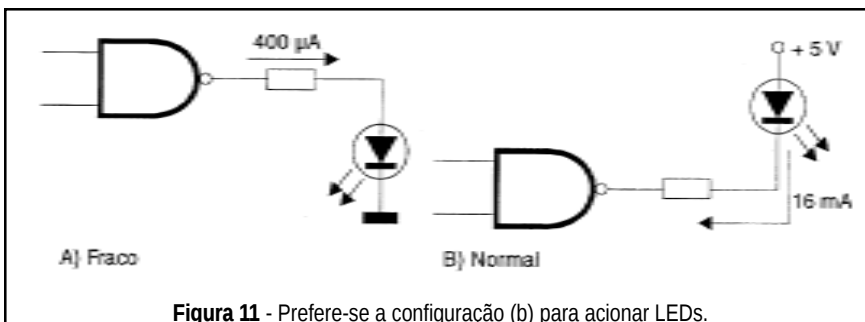
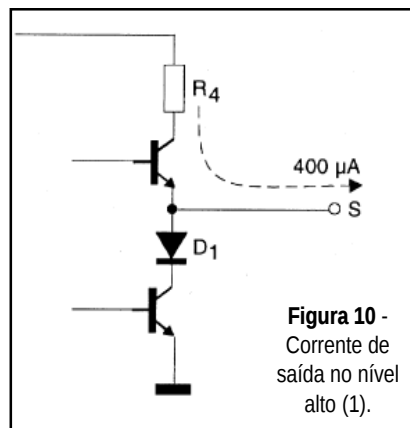
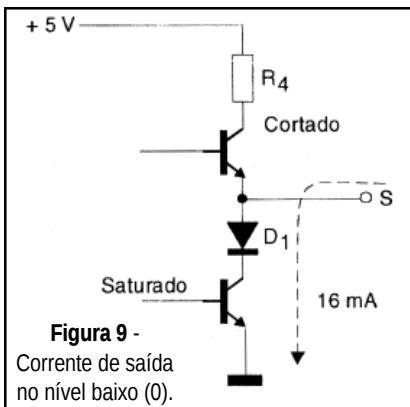
Esta corrente vai circular quando a tensão de entrada estiver com um valor superior a 2,0 V.

- 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 observamos no circuito equivalente da figura 9.

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 µA, figura 10.



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 ao nível 1.

Isso justifica o fato de que em muitas funções indicadoras, em que ligamos um LED na saída, 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 a figura 11.

- Fan In e Fan Out

Estes são termos técnicos que especificam características de extrema importância quando usamos circuitos integrados da família TTL.

A saída de uma porta não precisa estar obrigatoriamente ligada a uma entrada de outra porta. A mesma saída pode ser usada para excitar diversas portas.

Como a entrada de cada porta precisa de uma certa corrente e a saída da porta que irá excitar tem uma capacidade limitada de fornecimento ou de drenar a corrente, é preciso estabelecer um limite para a quantidade de portas que podem ser excitadas, veja o exemplo da figura 12.

Assim, levando em conta as correntes nos níveis 1 e 0 das entradas

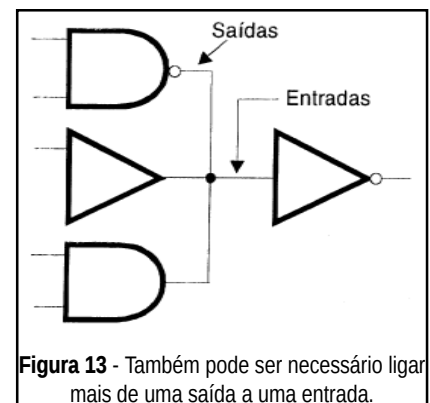
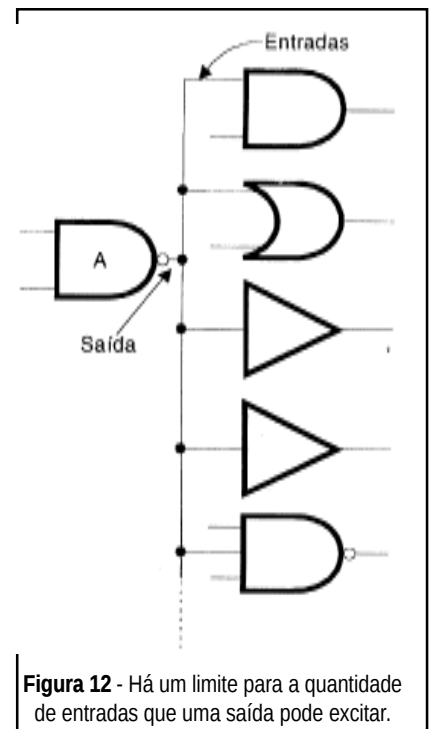
e saídas, definimos o *FAN OUT* como o número máximo de entradas que podemos ligar a uma saída TTL.

Para os componentes da família TTL normal ou *Standard* que estamos estudando, o *FAN OUT* é 10.

Por outro lado, também pode ocorrer que na entrada de uma função lógica TTL precisemos ligar mais de uma saída TTL.

Considerando novamente que circulem correntes nestas ligações e que os circuitos têm capacidades limitadas de condução, precisamos saber até que quantidade de ligações podemos fazer.

Desta forma o *FAN-IN* indica a quantidade máxima de saídas que podemos ligar a uma entrada, figura 13.



- Velocidade

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

No caso específico dos circuitos TTL, temos de considerar a própria configuração das portas que apresentam indutâncias e capacitâncias parasitas que influem na sua velocidade de operação.

Assim, levando em conta a configuração típica de uma porta, conforme observamos no circuito da **figura 14**, veremos que se for estabelecida uma transição muito rápida da tensão de entrada, a tensão no circuito não subirá com a mesma velocidade.

Este sinal terá antes de carregar as capacitâncias parasitas existentes de modo que a tensão de entrada suba gradualmente, demorando um certo tempo que deve ser considerado.

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

O resultado disso é que para os circuitos integrados TTL existe um retardo entre o instante em que o sinal passa do nível 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.

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 de um inversor.

Mostramos esses dois tempos na **figura 15**, eles são muito importantes nas especificações dos circuitos

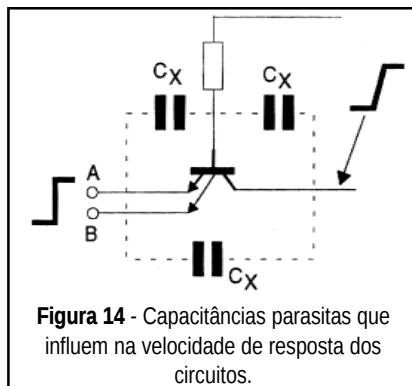


Figura 14 - Capacitâncias parasitas que influem na velocidade de resposta dos circuitos.

TTL, principalmente quando trabalhamos com o projeto de dispositivos muito rápidos. Basicamente podemos adiantar para o leitor 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.

Os primeiros circuitos TTL que foram desenvolvidos logo se mostraram inapropriados para certas aplicações.

3.5 - Subfamílias TTL

Os primeiros circuitos TTL que foram desenvolvidos logo se mostraram inapropriados para certas aplicações, quando é necessária maior velocidade, ou menor consumo de energia ou ainda os dois fatores reunidos.

Isso fez com que, mantendo as características originais de compatibilidade entre os circuitos e mantendo as mesmas funções básicas, fossem criadas sub-famílias que tivessem uma característica adicional diferenciada.

Assim, a partir da família original denominada "Standard" surgiram diversas subfamílias. Para diferenciar essas subfamílias, foram adicionadas ao número que identifica o componente (depois do 54 ou 74 com que todos começam), uma ou duas letras.

Temos então a seguinte tabela de subfamílias e da família TTL standard:

Indicação: 54/74

Família/Subfamília: Standard

Característica: nenhuma

Indicação: 54L/74L

Família/Subfamília: Low Power

Característica: Baixo consumo

Indicação: 54H/74H

Família/Subfamília: High Speed

Característica: Alta velocidade

Indicação: 54S/74S

Família/Subfamília: Schottky

Característica: nenhuma

Indicação: 54LS/74LS

Família/Subfamília: Low Power Schottky

Característica: nenhuma

A versão *standard* apresenta componentes com o custo mais baixo e também dispõe da maior quantidade de funções disponíveis.

No entanto, a versão LS se adapta mais aos circuitos de computadores, pois tem a mesma velocidade dos componentes da família *Standard* com muito menor consumo.

Algumas características podem ser comparadas, para que os leitores verifiquem as diferenças existentes.

- Velocidade

A velocidade de operação de uma função TTL normalmente é especificada pelo tempo que o sinal demora para propagar através do circuito. Em uma linguagem mais simples, trata-se do tempo entre o instante em que aplicamos os níveis lógicos na entrada e o instante em que obtemos a resposta, conforme verificamos através da forma de onda que vimos na **figura 15**.

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

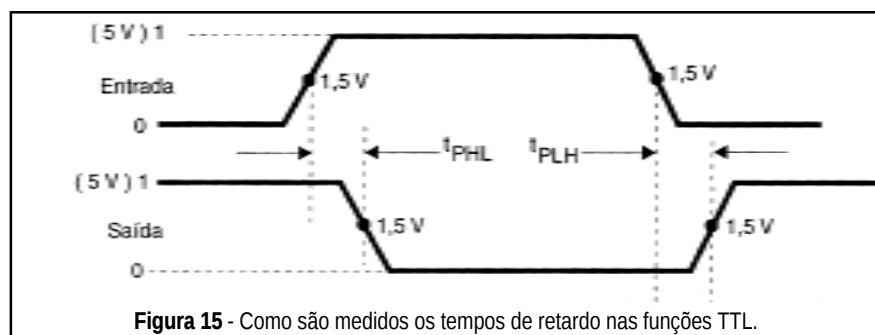
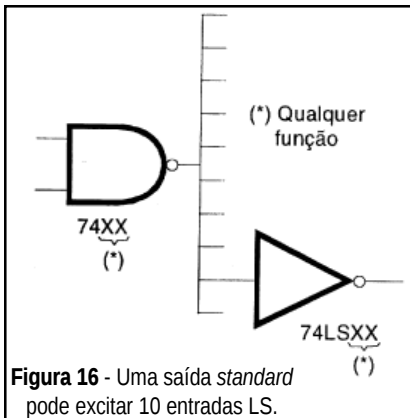


Figura 15 - Como são medidos os tempos de retardo nas funções TTL.



Assim, temos:

Família/Subfamília: TTL Standart
Tempo de programação (ns): 10

Família/Subfamília: Low Power
Tempo de programação (ns): 33

Família/Subfamília: Low Power Schottky
Tempo de programação (ns): 10

Família/Subfamília: High Speed
Tempo de programação (ns): 6

Família/Subfamília: Schottky
Tempo de programação (ns): 3

- Dissipação

Outro ponto importante no projeto de circuitos digitais é a potência consumida e portanto, dissipada na forma de calor. 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: *Standard*
Dissipação por Gate (mW): 10

Família/SubFamília: *Low Power*
Dissipação por Gate (mW): 1

Família/SubFamília: *Low Power Schottky*
Dissipação por Gate (mW): 2

Família/SubFamília: *High Speed*
Dissipação por Gate (mW): 22

Família/Subfamília: *Schottky*
Dissipação por Gate (mW): 20

O leitor já deve ter percebido um problema importante: quando aumentamos a velocidade, o consumo também aumenta. O projetista deve portanto, ser cuidadoso em escolher a sub-família que una as duas características na medida certa de sua precisão, incluindo o preço.

3.6 - Compatibilidade entre as subfamílias

Um ponto importante que deve ser levado em conta quando trabalhamos com a família *Standard* e as subfamílias 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 V.

Devemos observar, e com muito cuidado, que as correntes que circulam nas entradas e saídas dos componentes das diversas subfamílias são completamente diferentes, logo, 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 elaborar uma tabela, a partir das características dos componentes, em que a quantidade máxima de entradas de determinada subfamília possa ser ligada na saída de outra subfamília.

Esta tabela é dada a seguir:

Saída

	74L	74	74LS	74H	74S
74L	20	40	40	50	100
74LS	2,5	10	51	2,5	12,5

Entrada

	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*) pode excitar convenientemente 10 entradas 74LS (*Low Power Schottky*).

Na figura 16 mostramos como isso pode ser feito.

3.7 - Open Collector e Totem-Pole

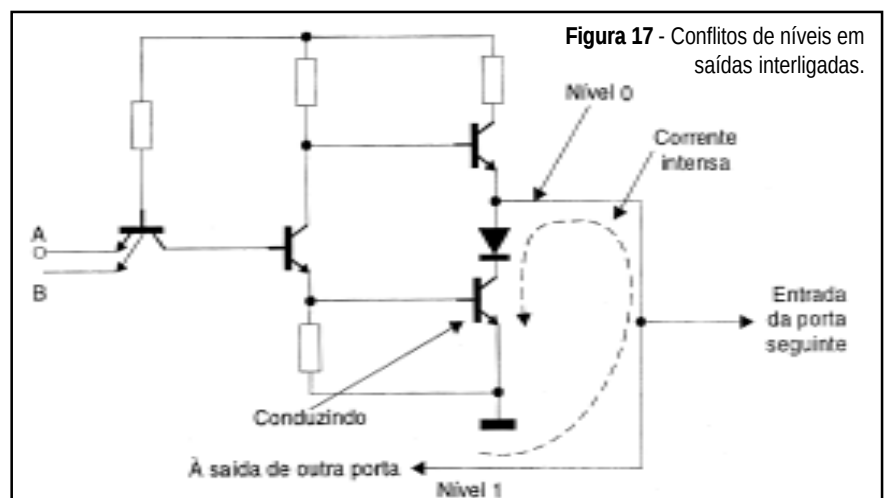
Os circuitos comuns TTL estudados até agora e que têm a configuração mostrada na figura 14 são denominados *Totem Pole*.

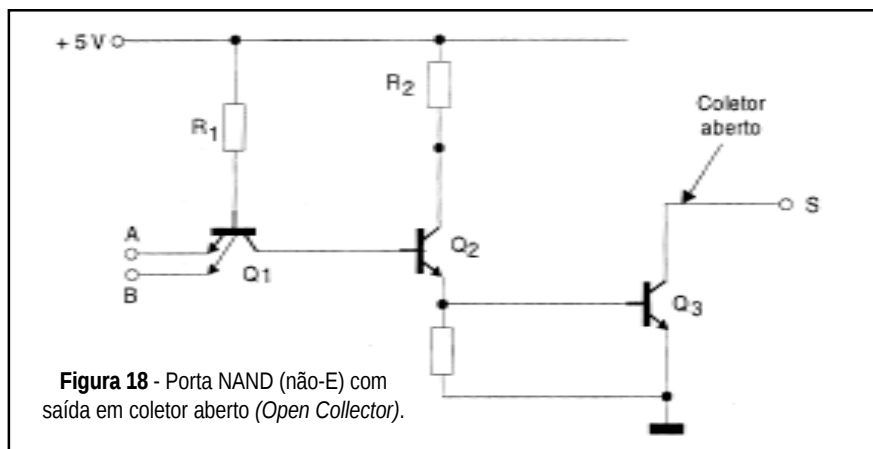
Nestes circuitos temos uma configuração em que um ou outro transistor conduz a corrente, conforme o nível estabelecido na saída seja 0 ou 1.

Este tipo de circuito apresenta um inconveniente se ligarmos duas portas em paralelo, conforme a figura 17.

Se uma das portas tiver sua saída indo ao nível alto (1) ao mesmo tempo que a outra vai ao nível baixo (0), um curto-circuito é estabelecido na saída e pode causar sua queima.

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





No entanto, existe uma possibilidade de elaborar circuitos em que as saídas de portas sejam interligadas. Isso é conseguido com a configuração denominada *Open Collector* mostrada na figura 18.

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 Ω 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.

A vantagem desta configuração está na possibilidade de interligarmos portas diferentes num mesmo ponto, figura 19.

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.

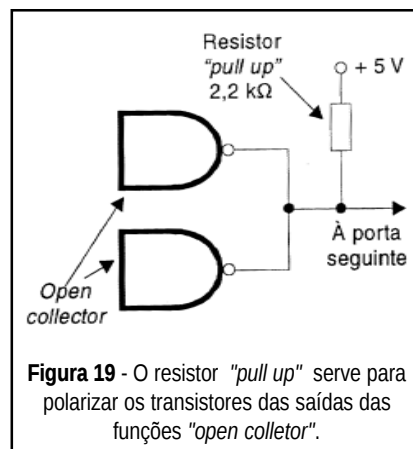
3.8 - Tri-State

Tri-state significa terceiro estado e é uma configuração que também pode ser encontrada em alguns circuitos integrados TTL, principalmente usados em Informática. Na figura 20 temos um circuito típico de uma porta NAND *tri-state* que vai servir como exemplo. Podem existir aplicações em que duas portas tenham suas saídas ligadas num mesmo circuito, figura 21.

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 deve ocorrer.

O que deve ocorrer é que quando uma porta estiver enviando seus sinais, a outra porta 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. Isso é conseguido através de uma entrada de controle denominada "*habilitação*" em inglês "*enable*" abreviada por EN.

Assim, quando EN está no nível 0, no circuito da figura 20, o transistor não conduz e nada acontece no circuito que funciona normalmente.

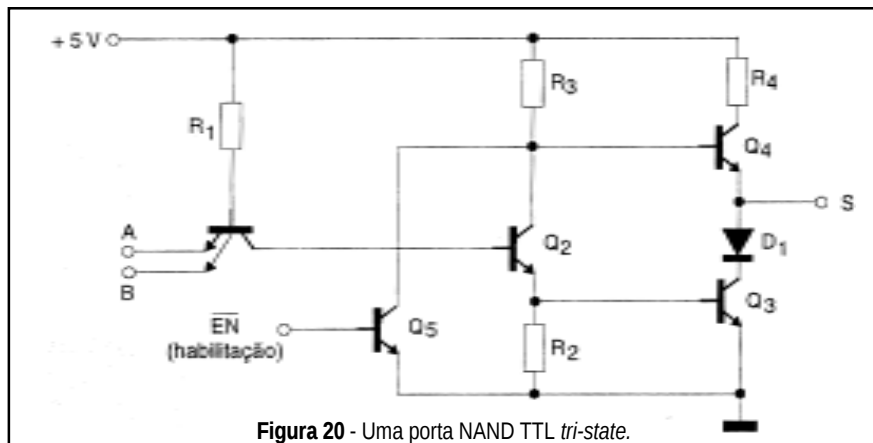
No entanto, se EN for levada ao nível 1, o transistor satura, levando ao corte, ou seja, os dois passam a se comportar como circuitos abertos, independentemente dos sinais de entrada. Na saída Y teremos então um estado de alta impedância.

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

Na figura 22 temos um exemplo de aplicação em que são usados circuitos *tri-state*. Uma unidade de processamento de um computador envia e recebe dados para/de diversos periféricos usando uma única linha (*bus*). Todos os circuitos ligados a estas linhas devem ter saídas do tipo *tri-state*.



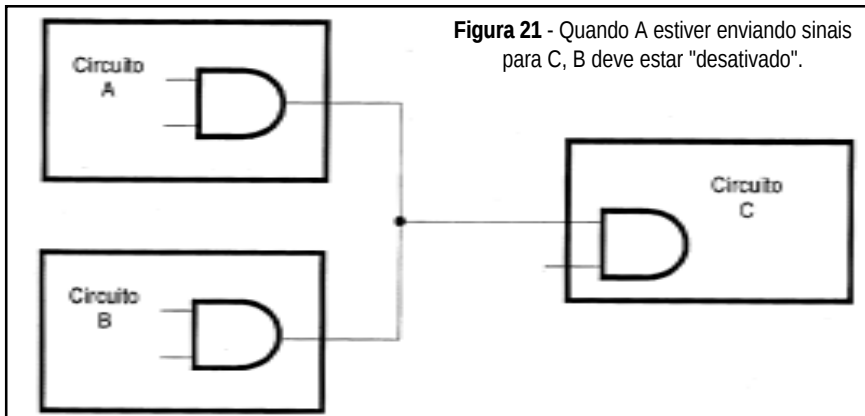


Figura 21 - Quando A estiver enviando sinais para C, B deve estar "desativado".

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 contêm grande quantidade de funções, mais de 1 000, usados principalmente nos modernos computadores são denominados:

- a) SSI
- b) MSI
- c) LSI
- d) VLSI

4. Um circuito integrado 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 configuração de saída não podemos ligar duas portas 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

Respostas da lição nº 2

- 1 - b)
- 2 - b)
- 3 - a)
- 4 - a)
- 5 - a)
- 6 - d)
- 7 - c)

Figura 22 - Na troca de dados entre diversas interfaces deve-se usar componentes com saídas *tri-state*.

