

CURSO TÉCNICO DE ELETRÔNICA

ELETRÔNICA DIGITAL

2º MÓDULO

Sumário

1	Sistemas de Numeração	3
1.1	Introdução.....	3
1.1.1	Números Binários	4
2	História de Augustos de Morgam.....	7
3	História de George Boole	9
3.1	Teorema da Álgebra de Boole	12
4	Introdução aos Circuitos Lógicos.....	13
4.1	Famílias Lógicas Bipolares.....	14
4.2	Famílias Lógicas MOS.....	15
4.3	Tecnologia TTL.....	16
4.3.1	Séries da Família TTL.....	16
4.4	Classificação do processo de utilização	16
4.5	Numeração dos pinos de conexão dos CI's	17
4.6	Tipos de Família TTL.....	18
4.7	Comparação entre as Famílias Lógicas	18
4.8	Tipos de circuitos de entradas e saídas	19
4.9	Comparação entre as Famílias TTL	20
4.10	Compatibilidade	21
4.11	Eficiência	21
5	Relação de Velocidade X Consumo X Frequência	21
5.1	Compatibilidade	22
6	Parâmetros em Corrente Contínua.....	23
7	Códigos do Fabricantes.....	24
7.1	Códigos dos Circuitos Integrados.....	24
7.2	Significado dos Códigos	25
8	Normas Técnicas.....	25
9	Tecnologia CMOS	26
9.1	Característica e descrição do processo de Fabricação	26
9.2	Lógica CMOS	39
10	Resistor-Transistor-Lógico.....	40
11	Diode-Transistor-Lógico	42
12	Valores elétricos do sinal Binário.....	43
13	Circuitos Integrados.....	44
14	Portas Lógicas	45
14.1	Porta AND	45
14.2	Porta OR.....	46
14.3	Porta NAND	46
14.4	Porta NOR	47
14.5	Porta XOR	47
14.6	Porta AND	48
14.7	Tempos de subida e descida.....	54
14.8	Tempo de Propagação	55

1 Sistema de Numeração

1.1 Introdução

Na vida cotidiana o homem lida, do ponto de vista numérico, com o sistema decimal. Da mesma forma, o computador, por características físicas, lida sob ambos os aspectos com o sistema binário.

Os circuitos eletrônicos que compõem os computadores são projetados para reconhecer sinais elétricos do tipo digital, onde o algarismo 1 representa existência de tensão e o algarismo 0 representa ausência de tensão.

Sistema de Numeração: Conjunto de símbolos utilizados para a representação de quantidades e as regras que definem a forma de representação.

Tipos: Sistema Decimal, Binário, Octal, Hexadecimal

SISTEMA DECIMAL

O homem utiliza o sistema decimal para contar desde muitos anos. A base do sistema decimal é o número 10, que corresponde ao número de símbolos utilizados para representação de quantidades.

0 1 2 3 4 5 6 7 8 9

$$1992 = 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$$

SISTEMA BINÁRIO

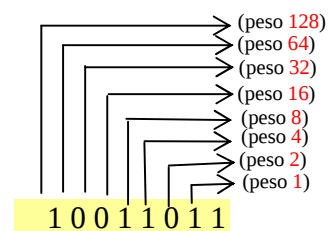
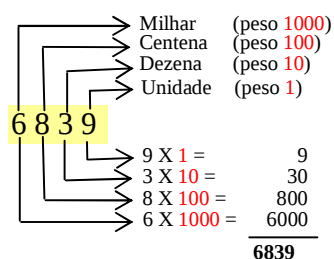
Sistema de numeração dos computadores atuais utilizado internamente pelo hardware. No sistema binário são utilizados os dígitos 1 e 0 para a representação. Cada dígito de um número representado neste sistema é denominado bit (binary digit).

1.1.1 Números Binários

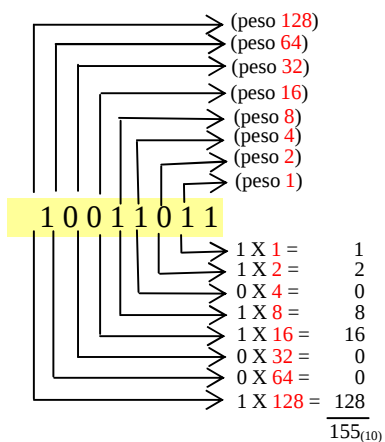
Código binário → 0 ou 1 → designa-se por **bit** (binary digit)

Com estes dois dígitos apenas pode construir-se uma base de numeração conhecida por **numeração binária** ou **base 2** e que tem a mesma arquitectura da numeração decimal (base 10) que usamos desde a escola primária.

Noção de ponderação ou peso



Exemplo da passagem de um número binário para decimal



É de notar que:

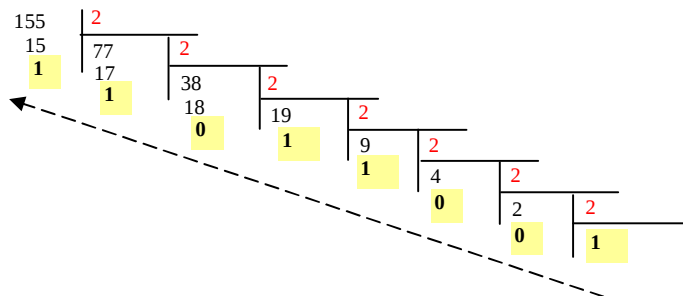
Sempre que o algarismo binário é **0**, o seu produto pelo peso correspondente, seja qual for, é sempre nulo, logo, sempre que o algarismo binário é **0** podemos ignorá-lo.

Sempre que o algarismo binário é **1**, o seu produto pelo peso correspondente tem como resultado o valor do próprio peso. Assim sendo, dispensa-se a

execução do respectivo produto.

Conclusão: $10011011_{(2)} \Leftrightarrow 155_{(10)}$

Exemplo da passagem de um número decimal para binário



Método das divisões sucessivas:

Como a base destino é a binária (base 2), vamos dividir sucessivamente o número decimal (155) por 2, até esgotar a parte inteira.

Seleccionando por ordem

inversa de ocorrência, desde o último quociente, todos os restos como indicado, temos a representação desejada:

$$155_{(10)} \Leftrightarrow 10011011_{(2)}$$

Tabela de correspondência entre um número decimal e o binário.

Decimal	Binário			
	peso			
	8	4	2	1
0				0
1				1
2			1	0
3			1	1
4		1	0	0
5		1	0	1
6		1	1	0
7		1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

Este código binário tem no entanto alguns inconvenientes, já que não permite à partida determinar qual o número de bits necessário para a representação de um número. Surge então outro código o **BCD** (**B**inary **C**oded **D**ecimal). O sistema BCD usa 4 bits para representar qualquer algarismo decimal.

Decimal	BCD			
	peso			
	8	4	2	1

0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Exemplo: $371_{(10)} \Leftrightarrow \underbrace{0011}_3 \underbrace{0111}_7 \underbrace{0001}_1_{(BCD)}$

Tal como o código binário simples o BCD só pode representar dados numéricos.

Para se poder codificar não só algarismos mas também letras maiúsculas e minúsculas, símbolos de pontuação desenvolveu-se um novo código com um total de 8 bits. Com 8 bits é possível um número de combinações $2^8 = 256$.

SISTEMA OCTAL

É o sistema de numeração cuja base é 8, ou seja, utiliza 8 símbolos para a representação de quantidade.

0 1 2 3 4 5 6 7

SISTEMA HEXADECIMAL

É o sistema cuja base é 16, utiliza 16 símbolos para a representação de quantidades.

0 1 2 3 4 5 6 7 8 9 A B C D E F

A – 10

B – 11

C – 12

D – 13

E – 14

F –15

2 História de Augustos De Morgan

Augustus De Morgan (1806 – 1871), nasceu na Índia, na região onde é hoje Madurai, Tamil, quando seu pai ali trabalhava e faleceu em Londres. Foi matemático e lógico. Formulou o que ficou conhecido como “leis de Morgan” e introduziu o termo de indução matemática. Existe uma cratera com seu nome na Lua.

Seu pai, John, foi Tenente-Coronel, que teve várias funções na Companhia das Índias Orientais. Sua mãe descendia de James Dodson, que computou uma tábua de anti -logaritmos. Augustus foi o quinto filho e perdeu a visão do olho direito, pouco após o nascimento.

A família voltou à Inglaterra quando Augustus tinha 7 meses. Seu pai faleceu quando ele tinha 10 anos.

De Morgan residiu em vários lugares do sudoeste da Inglaterra e assim Augustus freqüentou várias escolas nos seus primeiros anos. De Morgan era ativa e fervorosa na Igreja da Inglaterra (Anglicana) e desejava que seu filho se tornasse clérigo, mas nesta época o filho já mostrava uma certa rebeldia com relação a certas coisas da Igreja.

O talento de Augustus para a matemática só foi percebido quando ele tinha 14 anos. Em 1823, com 16 anos, ele ingressou no Trinity College em Cambridge. Suas horas de recreio ele passava tocando flauta, na qual se tornou exímio. Obteve o Bachelor of Arts, mas para obter o Master of Arts, ele teria que passar num exame de Teologia, ao qual se recusou, apesar de ser anglicano. (Cerca de 1875 estes testes teológicos seriam abolidos nas Universidades de Oxford e Cambridge).

Dirigiu-se então a Londres a fim de iniciar seus estudos de Direito, com a finalidade de exercer a profissão. Mas neste meio tempo, um grupo de intelectuais de mente liberal, resolveu fundar a London University (atualmente University College London), com o princípio de neutralidade religiosa.

De Morgan, então com 22 anos, foi designado Professor de Matemática. Mas sendo mal definidas as relações entre o Senado dos Professores e o Corpo dos Estudantes, estabeleceu-se uma disputa entre o Professor de Anatomia e os estudantes. E em virtude da decisão tomada pelo Conselho, vários professores se demitiram, encabeçados por De Morgan.

Mais adiante, ele foi convidado a reassumir a catedral, ali permanecendo por 30 anos.

Através da Society for the Diffusion of Useful Knowledge (Sociedade para a difusão de conhecimentos úteis), recém fundada sob a liderança de Lord Brougham, De Morgan publicou um grande trabalho sobre o Cálculo Diferencial e Integral, publicando, além disso, 712 artigos, ou seja, a sexta parte dos artigos, na Penny Cyclopaedia, editada pela mesma Society.

Em Londres, De Morgan tornou-se grande amigo de William Frend, também matemático, mas com idéias heterodoxas a respeito dos números negativos. Apesar dessa divergência em termos matemáticos, os dois convergiam a respeito de assuntos religiosos.

Em 1837 De Morgan casou-se com Sophia Elizabeth, uma filha de Frend. De Morgan foi extremamente bem sucedido como professor; para ele, a compreensão e assimilação de grandes princípios era muito mais importante que a habilidade analítica em casos particulares de princípios não bem entendidos.

Em 1837 publicou seus primeiros trabalhos de lógica formal

Em 1838 ele definiu e introduziu o conceito de “indução matemática”, num dos artigos publicados na Penny Cyclopaedia, referida acima, algo que estava sendo usado até então sem clareza e rigor.

Em 1849 ele publicou “Trigonometry and double Algebra”, no qual deu uma interpretação geométrica dos números complexos. Ele reconheceu o caráter puramente simbólico da álgebra e estava consciente de que outras álgebras, além da usual, poderiam ser criadas.

Talvez seu trabalho mais importante tenha sido “Budget of Paradoxes”, publicados inicialmente nas colunas do “Athenaeum”, um periódico; este trabalho foi revisado e melhorado pelo autor até o fim da vida, sendo publicado postumamente pela viúva.

De Morgan se correspondeu com Babbage, sendo tutor de Lady Lovelace, a qual, segundo consta, escreveu o primeiro programa de computador para Babbage.

De Morgan teve 3 filhos e 4 filhas. Seu segundo filho, George, se tornaria um matemático ilustre, sendo um dos co-fundadores da Mathematical Society em Londres; De Morgan foi o primeiro Presidente da nova Sociedade e seu filho o primeiro Secretário. Ela se tornaria, mais tarde, a London Mathematical Society. Em 1866 um clérigo Unitário foi recomendado para uma cátedra, mas foi vetado pelo Conselho. De Morgan considerou esta atitude um desrespeito à neutralidade religiosa e resignou à própria catedral, em protesto, com 60 anos de idade.

Seus alunos lhe mantiveram com 500 libras esterlinas ao ano. Mas dois anos depois seu filho George, pelo qual ele tinha tanto orgulho, faleceu e logo após uma de suas filhas também morreu.

De Morgan resistiu pouco a estes dois fatos, falecendo com 65 anos.

De Morgan tinha uma maneira de viver bastante peculiar, sendo um recluso: morando em Londres, nunca foi à Abadia de Westminster (onde são coroados os reis e rainhas da Inglaterra, com estátuas dos grandes heróis nacionais) nem à Câmara dos Comuns

(o Parlamento britânico) nem à Torre de Londres (uma visita “obrigatória” para qualquer pessoa que vá a Londres).

Ele nunca tentou se tornar “Fellow da Royal Society”, e aliás nunca ia às reuniões da dita Sociedade.

Ele se tornou grande amigo de William R. Hamilton, um irlandês, grande matemático, que inventou os “quaternions”.

Se todos os trabalhos de De Morgan fossem publicados, formariam uma pequena biblioteca.

Ele criou a “Relation Algebra”, uma álgebra binária, relacionada à Álgebra de Boole, que se tornaria muito importante para os Principia Mathematica de Bertrand Russell e A. N. Whitehead. Mais tarde, esta álgebra se tornaria também muito importante nos trabalhos, iniciados em 1940 por Alfred Tarski e seus colegas na Universidade da Califórnia.

3 História de George Boole

George Boole



Matemático Britânico, nasceu em Lincoln a 2 de Novembro de 1815. Filho de um Sapateiro, não tendo assim condições financeiras para obter um grau elevado em termos de educação. Mas a sua determinação levaria a que ultrapassasse esse obstáculo. Enquanto criança estudou na Escola Primária Lincoln, e depois numa Escola Comercial.

George Boole de início interessou-se por línguas, tendo aulas particulares de Latim com um Livreiro local. Aos doze anos de idade já conseguia traduzir um Poema Lírico em Latim, do poeta Horácio, demonstrando assim enormes capacidades. Aos 16 anos já era Professor Assistente, quatro anos mais tarde acabaria por fundar a sua própria Escola, isto em 1835. Já a algum tempo que Boole estudava matemática sozinho, embora já seu Pai o tivesse estimulado quando era novo ao dar-lhe um ensaio de construção de Instrumentos Ópticos.

Trabalhos de Laplace e Lagrange, eram alvo do estudo de Boole, através de notas que retirava. Recebeu encorajamento de Duncan Gregory, Editor de um Jornal Matemático, para estudar em Cambrige. Contudo não abandonaria seus Pais, que necessitavam dos seus cuidados. Em 1844, lançou um trabalho sobre,

a Aplicação de métodos Algébricos, para a solução de Equações Diferenciais, recebendo uma medalha de Ouro da Royal Society.

A Analise Matemática da Lógica foi outro dos trabalhos publicados em 1847, que divulgou assim as ideias que tinha da Lógica Simbólica, assim a Lógica, apresentada por Aristóteles, poderá ser apresentada por Equações Algébricas. Boole disse inclusive. "Nós não necessitamos mais de associar Lógica e Metafísica, mas sim Lógica e Matemática".

Boole tornou-se rapidamente conhecido, e o seu trabalho e ideias reconhecidos por todos os Matemáticos Britânicos e não só. 1840 foi o ano em que foi eleito para ocupar o lugar de Professor principal de Matemática na Irlanda, em Queen's College em Cork. E seria ali, que Boole iria permanecer para o resto da vida.

Uma investigação sobre as Leis do Pensamento, seria em 1854 a sua nova publicação, onde estão cimentadas as Teorias da Lógica e das Probabilidades. Ele conseguiu aquilo que é conhecido como Álgebra de Boole, pois abordou a Lógica, de forma a reduzi-la a uma Álgebra simples, inserindo Lógica em Matemática. Boole casou em 1855, com Mary Everest.

Em 1857 foi eleito membro da Royal Society, e recebeu Honras e reconhecimento das Universidades de Dublin e Oxford. Um trabalho sobre Equações Diferenciais em 1859, e em 1860 sobre cálculo de diferenças finitas, e outro sobre Métodos Gerais nas Probabilidades, foram alvo da investigação de Boole.

Publicou muitos trabalhos, e foi o primeiro a investigar a propriedade básica dos números, tal como a Propriedade Distributiva. Do seu casamento, com Mary Everest teve cinco filhas. Boole viria a falecer em 1864, com apenas 49 anos de idade vítima de Pneumonia. Hoje em dia a Álgebra de Boole, é aplicada na construção dos Computadores, sendo assim uma das razões fundamentais da revolução que os computadores estão a ter no mundo de hoje, aplica-se igualmente à pesquisa de Inteligência Artificial e na ligação dos telefones, entre muitas outras aplicações.

Boole foi e continua a ser considerado pelos colegas de profissão, e por todos aqueles que se dedicam à matemática, como tendo sido um homem genial. A lei especial da Lógica de Boole diz que x em relação a $y = x$. Para isso ser verdade, $x = 1$ ou $x = 0$. Sendo assim, a Lógica de Boole tem de utilizar um sistema Binário.

- - O matemático Britânico George Boole define uma álgebra lógica
- chamada a lógica Booleana, que possibilita mais tarde a criação
- do sistema binário e a electrónica digital;

A álgebra booleana ou álgebra de Boole

O nome Álgebra Booleana é em homenagem ao matemático inglês George Boole que em 1854, publicou um livro clássico. Uma investigação das leis do pensamento sobre as quais são baseadas as teorias matemáticas da lógica e das probabilidades. O propósito estabelecido por boole era o de realizar uma análise matemática da lógica.

A Álgebra de Boole surgiu inicialmente por ter relações com os problemas que apareceram no projeto de circuitos de chaveamento com réles em 1838, Claude E. Shannon que era assistente de pesquisa no departamento de engenharia elétrica no MIT, em uma versão de sua tese para o grau de mestre de ciências que foi publicada sob o título A Symbolic Analysis of Relay and Switching Circuits. Este artigo apresentava um método para representação de qualquer circuito consistindo de combinações de chaves e réles por um conjunto de expressões combinações matemáticas, e foi desenvolvido um cálculo para manipular estas expressões. O cálculo usado baseava-se comprovadamente na álgebra booleana.

Os postulados da álgebra de Boole são :

$$a+1 = 1$$

$$a+0 = a$$

$$a \times 1 = a$$

$$a \times 0 = 0$$

$$a \times a = a$$

Princípios da Análise Booleana:

A análise de Boole é uma secção discreta de matemática que se trata de relações de lógica em forma digital. Para usar estas relações, nós precisamos de um jogo de regras para a manipulação simbólica que nos deixara simplificar expressões complexas e resolver para desconhecidos. Originalmente, álgebra de Boole descreveu proposições cujo os resultados seriam verdadeiros ou falsos. No trabalho de computador é usado além de descrever circuitos cujo o estado pode ser qualquer um 1 (verdadeiro) ou 0 (falso).

3.1 Teorema da álgebra de Boole

Lei comutativa: (a) $A + B = B + A$ (B) $A B = B A$

Lei associativa: (a) $(A+B) + C = A (B+C) (B) (AB) C = A (BC)$

Lei Distributiva: (a) $A (B+C) = B + A C (B) A + (BC) = (A+B) (A+C)$

Lei de identidade: (a) $A + A = A (B) A A = A$

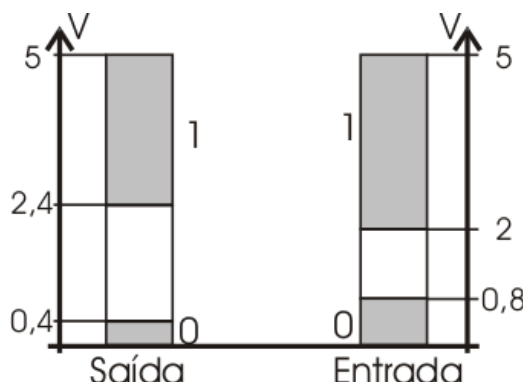
Lei de redundância (a) $A + A B = A (B) A (A+B) = A$

4 Introdução aos Circuitos Lógicos

As famílias de circuitos lógicos diferenciam-se entre si basicamente pelo tipo de dispositivo semicondutor que incorporam e o modo como se interligam os componentes internamente. Duas famílias são de especial interesse quando

se trata de circuitos lógicos: as famílias *CMOS* e *TTL*. A família *CMOS* é caracterizada pelo uso de transistores do tipo *CMOS* (*Complementary metal-oxid semiconductor*), os mais comuns em eletrônica e circuitos integrados. Os integrantes da família *TTL* (*Transistor-to-Transistor Logic*) são desenvolvidos principalmente pela Texas Company. Os circuitos de prefixo 74 são os de emprego comum e comercial, e operam na faixa de 0°C a 70°C.

A família *TTL* opera com tensão de alimentação de 5V e, portanto, todas as tensões em um sistema *TTL* estão entre 0V e 5V. Uma porta deste tipo estabelece como nível lógico mínimo de saída tensões superiores a 2,4 V, e compreende como nível lógico 1 tensões superiores a no mínimo 2V. Para o nível lógico 0, a máxima tensão de saída é 0,4 V, e o limite para interpretação como nível lógico 0 é de tensões de no máximo 0,8 V. Os níveis normais de operação são $V > 3,4V$ para nível lógico 1 e $V < 0,1 V$ para nível 0.



Níveis de tensão para circuitos TTL

No entanto, é importante notar que, além de pertencentes à família *TTL* ou *CMOS*, circuitos integrados ainda podem ser pertencentes a qualquer uma das seguintes famílias lógicas:

4.1 Famílias lógicas Bipolares

RTL – Resistor Transistor Logic – Lógica de transístor e resistência.

DTL – Díode Transistor Logic – Lógica de transístor e díodo.

TTL – Transistor Transistor Logic – Lógica transístor-transístor.

HTL – High Threshold Logic – Lógica de transístor com alto limiar.

ECL – Emitter Coupled Logic – Lógica de emissores ligados.

I²L – Integrated-Injection Logic – Lógica de injeção integrada.

Relação de todas as Famílias Lógicas

Família Lógica		Tradução	Características Gerais
RTL	Resistor Transistor Logic	Lógica que utiliza resistores e transistores.	Primeira série de circuito integrado. Baixo desempenho. Hoje não é mais fabricado.
DTL	Diode Transistor Logic	Lógica que utiliza diodos e transistores	Aperfeiçoamento do RTL. Hoje usado apenas para reposição.
TTL	Transistor Transistor Logic	Lógica que utiliza apenas transistores.	Muito importante nos circuitos digitais e minicomputadores. Utilizado em grande escala.
TTL - LS	Low power schottky	Lógica que utiliza transistores de alta velocidade e baixa	TTL aperfeiçoado. Proporciona alta velocidade e baixo consumo.
TTL - S	Super high speed	Lógica que utiliza transistores de "super alta" velocidade.	TTL de alta velocidade e alto consumo.
C MOS	Complementary Metal Oxide Silicon	Metal de óxido de silício complementar.	Baixíssimo consumo. Opera com fontes de 3 a 18 Volts.
IIL	Integrated Injection Logic	Lógica por injeção integrada.	Aperfeiçoamento do C MOS em termos de velocidade.
ECL	Emitter Coupled Logic	Lógica por emissor acoplado.	É o mais veloz de todas as séries. Pode operar em frequências de 500 MHz.
MOS	Metal Oxide Silicon	Metal de óxido de silício.	Elevado índice de integração. Pode conter até 25000 transistores. É o caso dos microprocessadores.

4.2 Famílias Lógicas

MOS

(Metal – Óxido –
Semicondutor)

CMOS – MOS de pares complementares NMOS/PMOS

NMOS – Utiliza só transístores MOS-FET canal N.

PMOS – Utiliza só transístores MOS-FET canal P.

4.3 Tecnologia TTL

4.3.1 Séries da Família TTL

Os circuitos integrados da família TTL são encontrados em duas series denominadas 74 (para uso comercial) e 54 (para uso militar).

Originalmente, os circuitos integrados foram destinados para uso militar onde tamanho, consumo e potencia e confiabilidade era preponderante.

Porém em 1964, surgiu a versão comercial de custo inferior. Respeitando-se algumas especificações, os dispositivos da série 54 são compatíveis com os da série 74.

A tecnologia TTL se apresenta em duas séries:

74XX – série comercial

54XX – série militar

Características:

Série 74XX

Temperatura: 0 a 70°C

125°C

**Alimentação: 4,75 a 5,25V
5,5V**

Série 54XX

Temperatura: -55 a

Alimentação: 4,5 a

4.4 Classificação do processo de integração

Até 1964, os dispositivos semicondutores eram encapsulados individualmente. Assim sendo, a montagem de portas lógicas e sistemas digitais era feita através de componentes discretos (resistores, diodos, transistores etc..).

O primeiro passo no desenvolvimento dos circuitos integrados foi a fabricação de uma porta lógica em uma única pastilha de silício e seu encapsulamento num embalagem própria. Verificou-se, então, que muitas portas lógicas poderiam ser fabricadas em uma única pastilha e encapsuladas em um único invólucro.

Assim sendo, em um circuito integrado, são encontrados muitos transistores, diodos e resistores sobre a mesma pastilha de silício, formando portas lógicas ou circuitos lógicos mais complexos, objetivando um menor custo, menor tamanho e maior confiabilidade.

O processo de integração pode ser classificado como:

SSI – Small Scale Integration(Integração de até 12 portas).

MSI – Médium Scale Integration.....(Integração de 13 até 99 portas)

LSI – Large Scale Integration.....(Integração de 100 a 1000 portas)

VLSI - Very Large Scale Integration....(Integração de mais de 1000 portas)

4.5 Numeração dos pinos de conexão do CI's

É feita no sentido anti-horário em relação a marca de referencia existente no CI.

Alguns integrados possuem terminais NC (No Internal Connection)

4.6 Tipos de Família TTL

Desde a introdução da primeira família (Standart) de circuitos integrados lógicos TTL, surgiram novas técnicas de projetos, bem como novos processos e tecnologias de fabricação de circuitos. Isto permitiu o surgimento de novas famílias que apresentam vantagens e desvantagens entre si.

Os principais tipos de famílias existentes no mercado e suas respectivas identificações estão mostrados na tabela abaixo:

FAMÍLIA	IDENTIFICAÇÃO
Standart	54/74
Low Power	54L / 74L
High Speed	54H / 74H
Schottky	54S / 74S
Low Power Schottky	54LS / 74LS
Advanced Schottky	54AS / 74AS
Advanced Low Power Schottky	54ALS / 74ALS

Num CI o tempo de atraso de propagação é o tempo necessário para que uma mudança nas entradas de um circuito lógico altere as suas saídas.

Quanto menos tempo de atraso de propagação, maior é velocidade de operação de um circuito lógico.

4.7 Comparação entre as Famílias Lógicas

Esta comparação pode ser resumida na tabela abaixo:

Família	Atraso de	Consumo de	Atraso X	Frequência
---------	-----------	------------	----------	------------

	Propagação	Potencia	Consumo	Máxima de Operação
	ns	mW	-	MHz
Standart	10	10	100	35
L	33	1	33	3
H	6	22	132	50
S	3	20	60	125
LS	10	2	20	45
AS	1,5	7	10,5	200
ALS	4	1	4	70

4.8 Tipos de circuitos de entradas e saídas

Uma característica dos circuitos integrados TTL é entender como nível lógico alto as entradas em aberto.

Embora o funcionamento do circuito integrado seja semelhante quando se tem um nível lógico alto aplicado as suas entradas, ou quando estas estão em aberto, esta última condição não deve ser considerada em projetos, pois um terminal em aberto aumenta a possibilidade de interferência de ruídos externos.

Além disso, quando as entradas dos dispositivos lógicos mudam de estado lentamente, vários problemas podem ser encontrados e, por isso, foram desenvolvidas outras tecnologias para os circuitos de entrada.

TIPOS DE ENTRADAS

- Convencional: **Do jeito que o sinal chega, ele é aplicado no terminal do CI, sem nenhum tratamento e isso pode causar vários problemas durante o chaveamento dos dispositivos.**
- Schmitt Trigger: **O sinal sofre um “tratamento” antes de chegar ao terminal**

do CI. Não importando o tipo de forma do sinal, ao passar pelo circuito Schmitt Trigger o sinal sai retangular.

Simbologia de um CI Schmitt Trigger:

TIPOS DE SAÍDAS:

- **Saída Totem Polem:**
- **Saída Open Collector:**
- **Saída Tri-State:**

SAÍDA TOTEM POLEM:

Esta configuração de saída possui normalmente em sua estrutura interna, um transistor conectado ao +Vcc, permitindo alimentação de cargas altamente capacitivas.

SAÍDA OPEN COLLECTOR (COLETOR ABERTO):

Configuração de saída que não possui internamente um resistor ligando o coletor de saída ao +Vcc. Este resistor, chamado de PULL UP (Rp) deve ser conectado externamente para que o dispositivo funcione corretamente.

SAIDA TRI-STATE (TRES ESTADOS):

Configuração de saída que pode fazer com que ele apresente uma alta impedância (terceiro estado) em relação à linha na qual esta conectada. Neste caso, é como se o dispositivo estivesse desligado da linha.

Isto permite conectar várias saídas em uma mesma linha de dados, possibilitando que apenas uma forneça nível lógico alto ou baixo, sem danificar os outros dispositivos.

Este tipo de saída é fundamental, pois permite interligar numa única via de dados, diversos circuitos integrados, facilitando o projeto de sistemas digitais mais complexos.

Os estados que esta configuração apresenta são:

- Estado 0 – nível lógico baixo
- Estado 1 – nível lógico alto
- Tri State – Alta impedância

4.9 Comparação entre as Famílias TTL

Para o desenvolvimento de projetos de sistemas digitais, vários fatores devem ser considerados na escolha da família e da tecnologia de fabricação de circuitos integrados a serem utilizados.

4.10 Compatibilidade

Velocidade (tempo de atraso de propagação);
Consumo de potência;
Frequência de operação;
fan-in e fan-out;
Parâmetros de tensão e corrente de entrada e saída e
tipos de circuitos de entrada e saída.

4.11 Eficiência

A eficiência dos projetos que envolvem circuitos integrados digitais é medida considerando-se principalmente a relação entre:

Velocidade;
Consumo de potência e,
Frequência de operação

Quanto maior a velocidade (menor tempo de atraso de propagação) e menor o consumo (dissipação de potência), mais eficiente é o sistema. A frequência máxima de operação, embora relacionada à velocidade, um fator importante, principalmente nos sistemas digitais que processam informações sincronizadas com pulsos de clock.

5 Relação de Velocidade X Consumo X Frequência

A tabela abaixo, mostra, para efeito de comparação, valores típicos de tempo de atraso de propagação, consumo de potência e frequência máxima de operação para os circuitos integrados de diversas famílias.

A coluna atraso x consumo serve como referência para mostrar a eficiência das diversas famílias de circuitos integrados.

Desta forma, considerando-se apenas estes aspectos, pode-se observar que família ALS é mais eficiente (atraso x consumo = 4) enquanto que a família H é a menos eficiente (Atraso x consumo = 132).

5.1 Compatibilidade

As famílias de circuitos integrados TTL, são compatíveis entre si. Entretanto, para evitar falhas operacionais, é necessária a verificação de algumas especificações como:

- Fan-in: Corresponde à carga que a entrada de um bloco lógico representa para o conjunto de saídas que está conectado a ela.
- Fan-out: É o número de blocos lógicos que pode ser conectado à saída de outro bloco lógico sem causar degradação do nível lógico.
- Tipos de circuitos de entrada e saída;
- Frequência máxima de operação

Compatibilidade entre as famílias TTL, pode ser resumida no quadro abaixo:

Família	Input Loading	Output Loading
Standart	$I_{iL} = -1,6\text{mA}$ $I_{iH} = 40\mu\text{A}$	$I_{oL} = 16\text{mA}$ $I_{oH} = -400\mu\text{A}$
L	$I_{iL} = -180\mu\text{A}$ $I_{iH} = 10\mu\text{A}$	$I_{oL} = 3,6\text{mA}$ $I_{oH} = -200\mu\text{A}$
H	$I_{iL} = -2\text{mA}$ $I_{iH} = 50\mu\text{A}$	$I_{oL} = 20\text{mA}$ $I_{oH} = -500\mu\text{A}$
S	$I_{iL} = -2\text{mA}$ $I_{iH} = 50\mu\text{A}$	$I_{oL} = 20\text{mA}$ $I_{oH} = -1\text{mA}$
LS	$I_{iL} = -400\mu\text{A}$ $I_{iH} = 20\mu\text{A}$	$I_{oL} = 8\text{mA}$ $I_{oH} = -400\mu\text{A}$

AS	$I_{iL} = -500\mu A$ $I_{iH} = 20\mu A$	$I_{oL} = 20 \text{ mA}$ $I_{oH} = -2 \text{ mA}$
ALS	$I_{iL} = -100\mu A$ $I_{iH} = 20\mu A$	$I_{oL} = 8 \text{ mA}$ $I_{oH} = -400\mu A$

NOTAS:

I_{iL} – corrente de entrada (input) quando o nível lógico for baixo.

I_{iH} – corrente de entrada (input) quando o nível lógico for alto.

I_{oL} – corrente de saída (output) quando o nível lógico for baixo.

I_{oH} – corrente de saída (output) quando o nível lógico for alto.

O fan-in é uma especificação que deve ser levada em consideração principalmente quando se fizer conexão de várias saídas OPEN COLLECTOR a uma única entrada.

Como o fan-out é um número definido para portas lógicas de mesma família, a quantidade de portas lógicas que pode ser ligada numa saída é diferente quando existe a mistura de famílias, sendo, por isso, necessário o conhecimento dos parâmetros de corrente de entrada e de saída das portas lógicas.

6 Parâmetros em Corrente Contínua

- Corrente Positiva: Corrente que entra num terminal de entrada ou saída de um dispositivo.

A corrente positiva de entrada de um dispositivo quando uma tensão em nível alto igual a tensão mínima de saída em nível alto especificada para a família, é aplicada a esta entrada.

- Corrente Negativa: Corrente que sai por um terminal de entrada ou saída de um dispositivo.

Códigos dos Fabricantes

Existem diversos fabricantes de circuitos integrados. A identificação do fabricante é possível através do seu prefixo, como mostra a tabela abaixo:

PREFIXO	FABRICANTE
SN	TEXAS
DM	NATIONAL
F	FAIRCHILD
MC	MOTOROLA
JF	PHILIPS
N	SIGNETICS
FL	SIEMENS
HD	HITACHI
MB	FUJITSU
M	MITSUBISHI
μP	NEC
TD	TOSHIBA

7 Códigos dos Fabricantes

7.1 Códigos de circuitos integrados

Cada circuito integrado recebe um código de identificação através do qual é possível conhecer, além do fabricante, várias características técnicas como:

Série;
Tipo de família;
Função lógica;
Tipo de encapsulamento;

A tabela abaixo, mostra vários códigos de identificação de circuitos integrados e seus respectivos fabricantes.

Código do Fabricante	Fabricante
----------------------	------------

SN 74 LS 195 A J	TEXAS
DM 74 L 165 A N	NATIONAL
F 9300 D C	FAIRCHILD
MC 74 91 A P	MOTOROLA
FJ H 13 1	PHILIPS
N 74 S 00 F	SIGNETICS
FL H 29 I U	SIEMENS
HD 25 48 P	HITACHI
MB 400 M	FUJITSU
M 5 32 90 P	MITSUBISHI
μP B 2000 D	NEC
TD 34 00 A P	TOSHIBA

7.2 Significado dos códigos

Os significado dos códigos de todos os componentes dos diversos códigos de identificação dos circuitos integrados são encontrados nos manuais de seus respectivos fabricantes.

Apenas para efeito de exemplificação, abaixo estão apresentados dois códigos de identificação com o significado de todos o seus componentes.

Código	SN 74	LS	08	A	N	3
Significado	1	3	4	5	6	7
2						

1 – Prefixo do Fabricante	Texas
2 – Série	Comercial
3 – Família	Low Power Schottky
4 – Função Lógica	Quad 2 – Input AND Gates
5 – Tecnologia	Série Melhorada
6 – Encapsulamento	Em linha dupla – plástico
7 – Instruções	Processamento PEP nível 3

8 Normas Técnicas

A maioria dos manuais de circuitos integrados digitais utiliza a simbologia de acordo com a norma técnica ANSI. Entretanto, existem outras normas para a apresentação dos dispositivos digitais utilizadas em projetos de empresas brasileiras (ABNT) e européias (IEEC).

O nome de uma norma técnica refere-se à sigla do órgão que a criou e possui vários códigos que diferenciam as diversas áreas de aplicação.

A tabela abaixo, apresenta quatro destas normas e seus respectivos órgãos, bem como, as principais características relativas à simbologia adotada na área de eletrônica digital.

Norma	Órgão	Características
ANSI	American National Standards Institute	Indica apenas a função lógica do dispositivo.
IEEE	Institute of Electrical and Electronic Engineers	Indica a função lógica e faz referências a outras funções e características do dispositivo.
IEC	International Electrotechnical Commission	Segue o mesmo princípio da norma IEEE. (IEC-617-12)
ABNT	Associação Brasileira de Normas Técnicas.	Segue o mesmo princípio da norma IEC. (NBR – 08767 – Símbolos Gráficos para Diagramas Lógicos)

Apesar da ABNT, tanto no Brasil como na maioria dos países do mundo, a norma mais utilizada é a ANSI, tanto por questões culturais (padrão americano) e comerciais como, também, devido à sua grande simplicidade.

9 Tecnologia CMOS

9.1 Características e descrição do processo de fabricação do CI CMOS

A tecnologia usada é CMOS 2.0 μm *n-well*, 1 camada de *Poly*, 2 camadas de metal. Nesta tecnologia podem ser construídos circuitos analógicos, digitais e híbridos e os transistores têm os seguintes parâmetros:

	NMOS	PMOS	Unidades
Comprimento mínimo da <i>gate</i>	2.0	2.0	μm
Espessura do óxido da <i>gate</i>	25	25	nm
Tensão de <i>threshold</i>	0.75	-1.10	V
Mínimo <i>pitch</i> para os metais	4.8	4.8	μm
Mínimo <i>pitch</i> para a <i>poly</i>	4.0	4.0	μm
Tensão de alimentação nominal	5	5	V
I_{ds} , correntes de saturação para $V_{ds}=5\text{V}$, $V_{gs}=5\text{V}$	4.75	1.90	mA

As características dos componentes passivos são as seguintes:

- Condensadores de precisão e altamente lineares (*metal1-poly*, *metal1-metal2*, *metal2-poly-metal1*)
- Resistências de precisão (*poly*, *Active-SN*, *Active-SP*, *metal1*, *metal2*)

O *wafer* pode ser dopado com átomos doadores e átomos receptores. Esta dopagem é feita com Fósforo (tipo n) e Boro (tipo p), respectivamente.

Quando se usa um *wafer* tipo p, o NMOS é fabricado diretamente no substrato tipo p, enquanto que os transistores de canal p, PMOS, são fabricados dentro do poço *n-well*.

O *L-Edit* permite ver todas as *layers* simultaneamente, na sequência litográfica cada *layer* é aplicada separadamente. Para realizar este processo é criada uma máscara para cada *layer*; a máscara é um lâmina de vidro, no qual é duplicado cada padrão numa das faces com uma camada fina de crômio. A máscara é transparente à luz, exceto nas regiões onde o crômio (regiões negras) não deixa passar a luz.

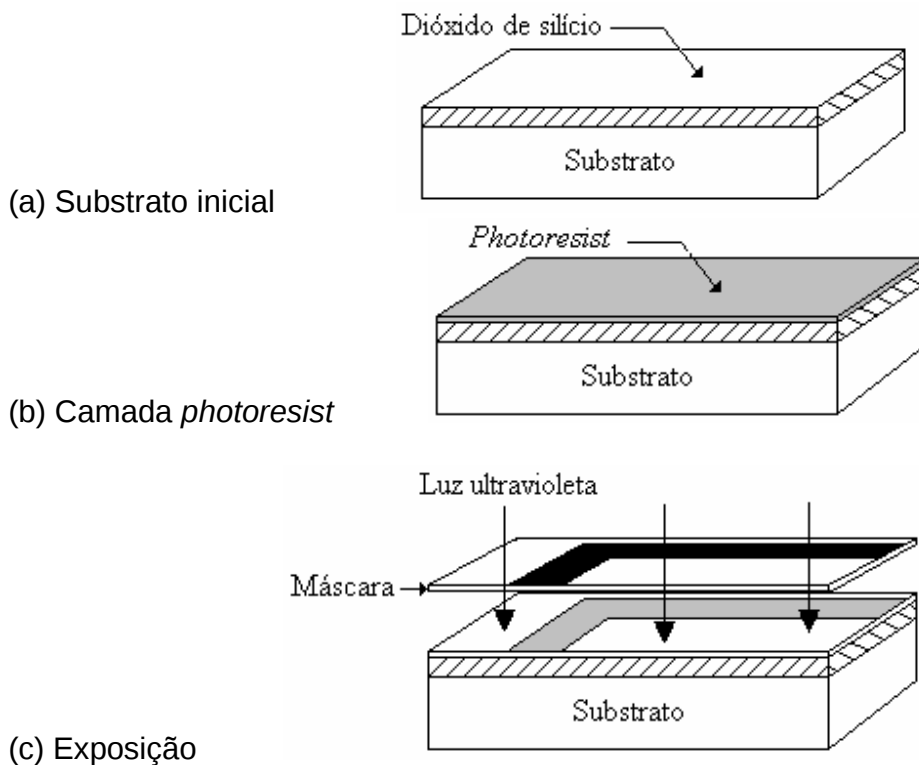
Pode ver na figura abaixo, uma sequência típica, onde a região dopada é realizada no substrato de silício. O ponto de partida é mostrado na figura (a), onde apenas é acrescentada uma camada de óxido na superfície do silício. Depois, o *wafer* é impregnado num disco giratório (*spin coating*) com *photoresist*, espalhando-se por todo o *wafer*, seguidamente submetido a uma secagem rápida no forno de forma a produzir uma superfície dura, como se vê na figura (b). O *photoresist* é um polímero orgânico sensível à luz, com propriedades similares a um qualquer filme fotográfico. A imagem na máscara é projetada opticamente na superfície do *photoresist* durante a exposição aos raios ultravioletas, como mostra a figura (c).

Em seguida, com um composto químico apropriado remove-se a área da camada de *photoresist* que não foi exposta à luz (neste caso usou-se *photoresist* negativo e os raios ultravioletas compactaram o polímero), como se pode ver na figura (d), pois para *photoresist* positivo os raios ultravioletas corroíam o polímero.

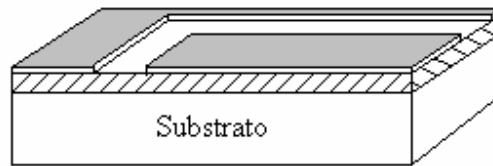
O desenho da camada *photoresist* é usado como máscara para fazer o *etching* da camada de material. No exemplo, a camada de dióxido de silício é removida usando um processo a seco *RIE* (*reactive-ion etching*). Finalmente a camada de *photoresist* é retirada. Isto pode ver-se no desenho da figura (e)

O modelo do dióxido de silício é agora usado como máscara para fazer a dopagem das zonas que ficaram tipo n. Na figura (f), a superfície do *wafer* é exposta na máquina de dopagem. As áreas do silício expostas são dopadas.

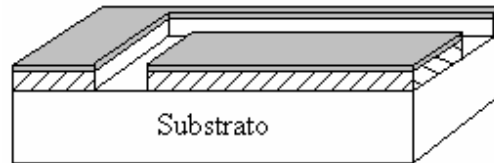
O resultado desta dopagem pode ser visto nas figuras abaixo:



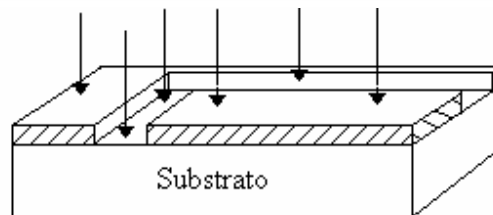
(d) *RIE*



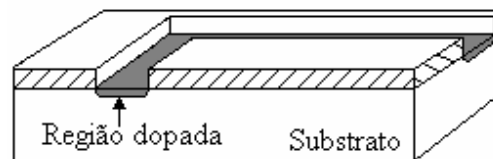
(e) *Etching*



(f) Dopagem



(g) Após a dopagem

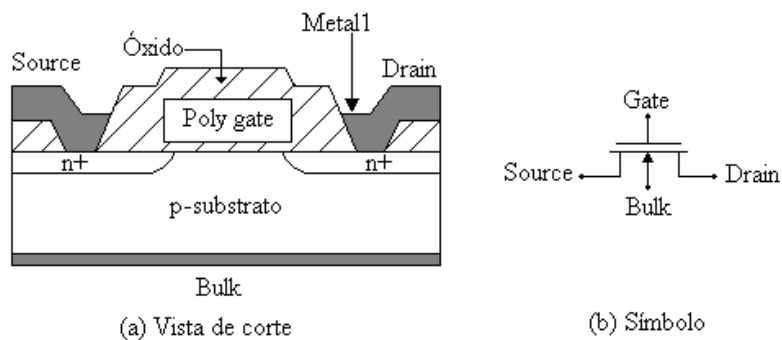


As limitações deste processo, tais como a resolução mínima e o espaço mínimo, são definidos por um conjunto de regras de desenho para o *layout* físico.

Os terminais do MOSFET são a *gate*, o *drain* e a *source*. Como mostra a figura abaixo, a ligação do substrato do tipo-p origina o *bulk electrode*. O *bulk* é ligado à tensão mais baixa do circuito para o caso do MOSFET do tipo n, normalmente ao GND (terra).

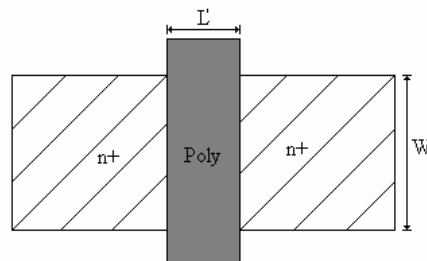
Para realizar a construção do *layout* de um MOSFET do tipo-n é necessário utilizar os elementos: *p-substrato*, *n+ do drain e source*, *dióxido de silício*, *poly (gate)*, *óxido e metal*. Em geral as camadas de óxido são implícitas no desenho, assim como o substrato. A função do *contact* é gerar uma máscara para eliminar o óxido em certas zonas, com o objectivo de permitir que o metal fique eletricamente ligado à camada n+.

A figura abaixo mostra um MOSFET do tipo N:



A construção de todas as camadas é realizada de forma como foi apresentada atrás a dopagem de zonas tipo n.

Os parâmetros básicos de geometria de um MOSFET são o comprimento de canal W e a largura L (é a distância entre as zonas n+). Estes parâmetros pode ser vistos de uma forma mais explícita se for vista de cima, conforme a figura abaixo:

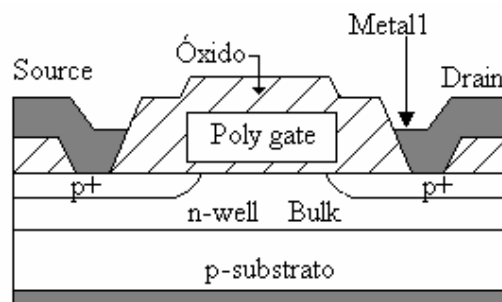


De referir que a largura de canal do MOSFET (L) é um pouco maior a largura de canal desenhada (L'), devido aos efeitos laterais de dopagem. Estes valores são calculados automaticamente, bastando definir o L efectivo, ou seja, a distância entre as regiões dopadas com n+.

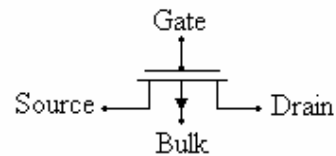
Um PMOS tem a mesma estrutura geométrica que um NMOS, mas com as polaridades invertidas, isto é, as polaridades p+ são mudadas para n+ e vice-versa.

Como o substrato é do tipo p é necessário criar uma zona tipo n para usar como *bulk* do canal do PMOS.

Como está demonstrado na figura abaixo:

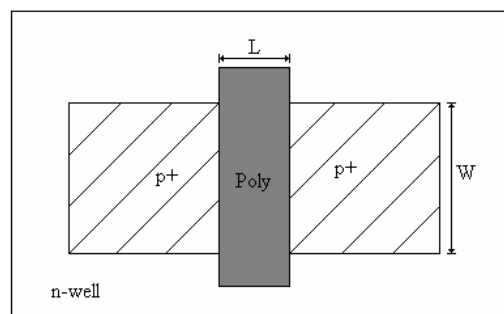


(a) Vista de corte



(b) Símbolo

Como se pode ver na vista do *layout* a camada *n-well* tem de ser desenhada, conforme a figura abaixo. A sua construção compreende os elementos: *p_substrato*, *n-well*, *p+* do *drain* e *source*, *óxido da gate*, *polisilício*, *óxido* e *metal1*.



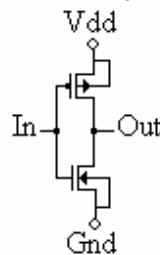
Vista de cima de um PMOS

As máscaras que são requeridas no desenho de *layout* são a *n_well*, as zonas p+, o polisilício e o metal1. As máscaras de polisilício e metal1 são

comuns no dois tipos de MOSFETs, tipo p e tipo n, e serão usadas para fazer as ligações quando estes forem integrados nos circuitos a desenhar em *layout*.

Tal como no NMOS, o PMOS apresenta como parâmetros geométricos o comprimento de canal L e a largura de canal W , onde a corrente é proporcional ao seu ratio (W/L).

Partindo destes conhecimentos, pode dizer-se que são a base para poder construir os dois tipos de MOSFETs, tipo p e tipo n, no mesmo substrato, tornando assim possível construir circuitos mais complexos. Como demonstração é apresentada a construção de um inversor, componente que é composto por um PMOS e um NMOS, como pode ver-se na Figura abaixo;

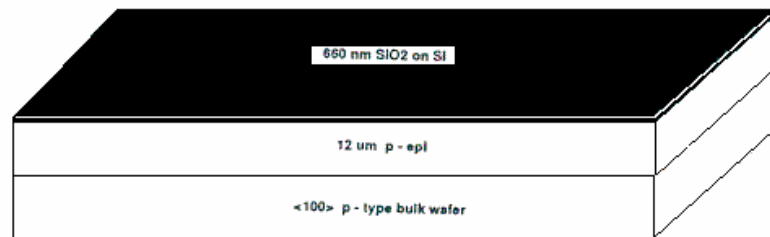


Esquema elétrico de um inversor.

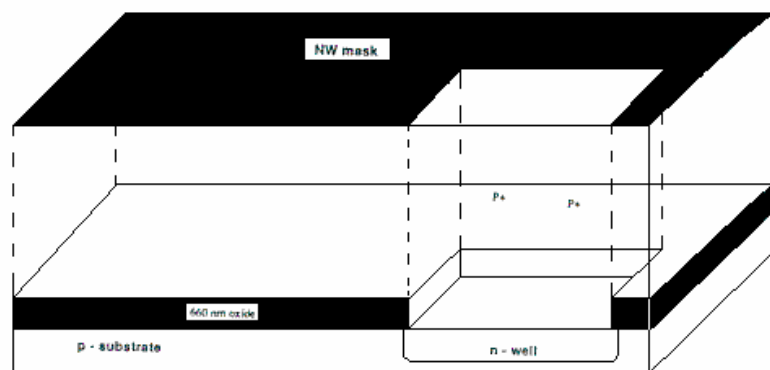
Como já foi referido atrás, utiliza-se como ponto de partida um *wafer* do tipo p como substrato..

A camada *Epi* é usada como camada base para a construção dos dispositivos. Esta camada será omitida nas figuras seguintes. Na figura abaixo, pode ver-se o passo referido (a), assim como o próximo passo, a máscara para criar a *n-well* (b).

Veja abaixo nas figuras os primeiros passos na construção e a primeira máscara, a *n-well*;



(a) Substrato e a camada de dióxido de silício



(b) Máscara n-well

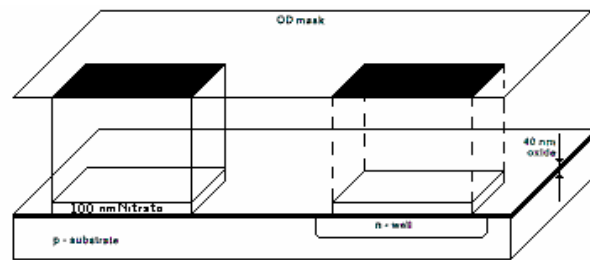
Eletricamente a zona da *n-well* deve ser mantida na zona de maior tensão do circuito, para garantir o funcionamento adequado. Normalmente é ligada à tensão de alimentação, ao Vdd.

Como a densidade de dispositivos pode ser muito elevada, é imperativo ter em atenção o problema do isolamento eléctrico de cada um deles. É neste passo do fabrico que é definida a localização de todos os transístores. O isolamento é realizado pela próxima colocação de óxido de silício.

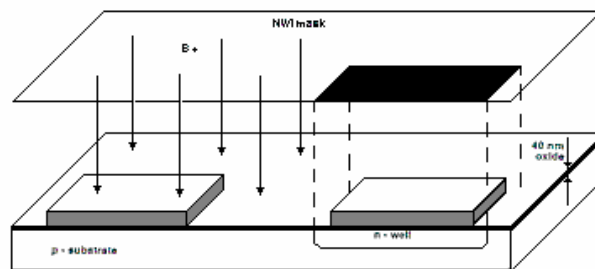
A área total do *chip* é dividida em duas grandes partes, a área activa, onde são colocados os MOSFETs, e as regiões de campo em volta da área activa, que é onde passam as linhas de ligação dos MOSFETs, como o polisilício e os metais. As áreas activas são feitas com a máscara *active* (OD) usando nitrato de silício, que é depositado numa camada fina de óxido de silício conhecida como óxido libertador de *stress* e é usado como *buffer* mecânico entre o nitrato e o silício, como se vê na figura abaixo de especificação (a).

Antes de ser acrescentado o óxido LOCOS (*Local Oxidation of silicon*) é feita uma deposição com p+ (máscara NW1), com o objectivo de aumentar a tensão de *threshold* da região de campo, para que a tensão aplicada a uma linha de ligação não crie canais de condução indesejáveis.

A terceira e quarta máscaras usada é apresentada na Figura abaixo;

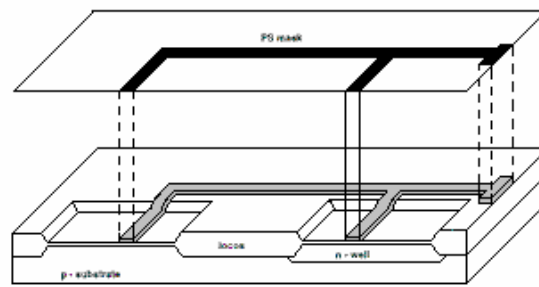


(a) Máscara OD

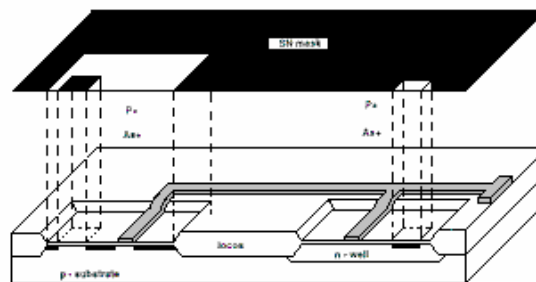


(b) Máscara NWI

Após a criação da LOCOS é removido o nitrato de silício e o *wafer* está agora em condições de ser depositado um óxido para receber por cima o polisilício, que formará as *gates* dos MOSFETs. A camada de óxido da *gate* não é aqui mostrada, mas pode ver-se a quinta e a sexta máscara da poly, na figura abaixo;



(a) Máscara da poli



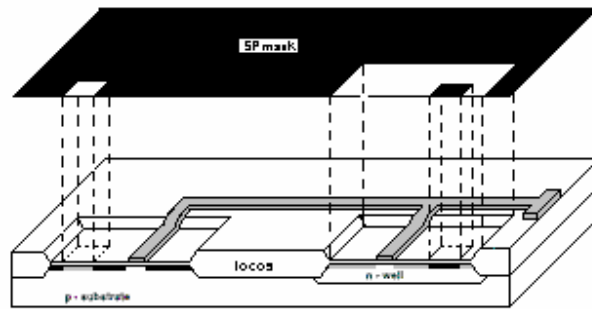
(b) Máscara SN

Para criar o NMOS utiliza-se a máscara SN, que se pode ver na figura (b) acima, onde é feita uma dopagem com doadores (como o Fósforo) para dopar as regiões da *drain* e da *source*. Na prática, esta camada não necessita de ser desenhada, pois é o inverso da camada SP, bastando para isso desenhar a camada SP.

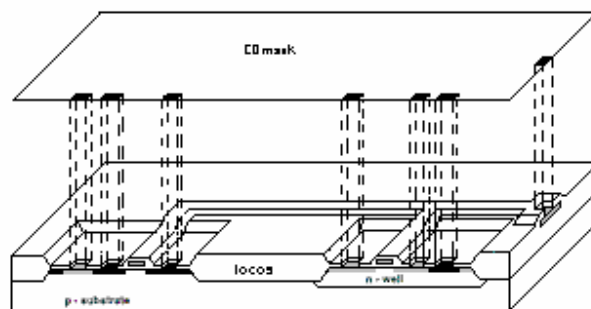
No caso contrário ao anterior, a criação de um PMOS, utiliza-se a máscara SP. Com esta é aplicada uma dopagem com Boro na superfície do *wafer*, para criar nas zonas indicadas pela máscara a *drain* e a *source* dos PMOS, como é mostrado na figura abaixo de especificação (a).

Nesta fase o *wafer* está todo coberto com um óxido que foi formado através de uma reação química e que não necessita de silício do *wafer*. É comum usar um LTO (*low-temperature oxide*) para realizar esta difusão e consequentemente as camadas de óxido que ficam por baixo das regiões dopadas não permite qualquer ato difusivo. Esta camada não foi demonstrada na sequência aqui apresentada. .

Como o *wafer* está todo isolado é necessário criar passagem para que os terminais dos MOSFETs possam ser ligados ao exterior, assim aparece a máscara dos contatos (CO), como se pode ver na Figura abaixo como sétima e oitava máscaras;

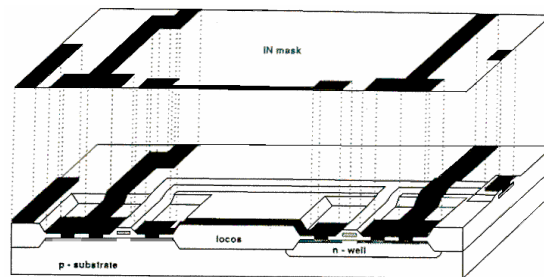


(a) Máscara SP



(b) Máscara CO

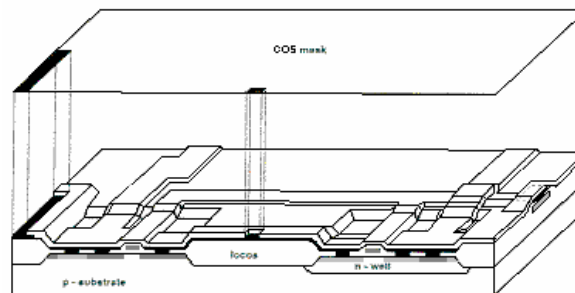
Com a camada de óxido removida no sítio dos contactos faz-se a deposição do metal1, permitindo que o metal1-active e metal1-poly fiquem eletricamente ligados. Resulta deste passo a figura abaixo;



Máscara do metal1

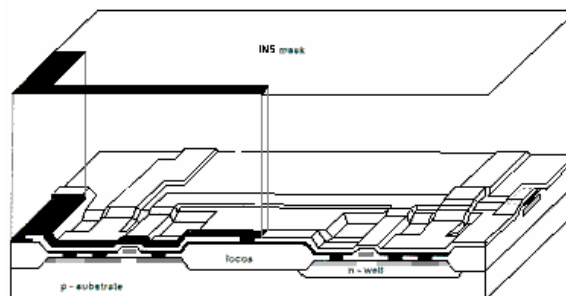
Depois de depositado o metal1 é depositado um óxido de plasma, que serve para isolar. A referida camada não é apresentada nas figuras.

A máscara COS é aplicada para fazer *etching* nas zonas dos contatos onde se irá ligar o metal2 com o metal1 ou com a *poly*, como se pode verificar na figura abaixo;



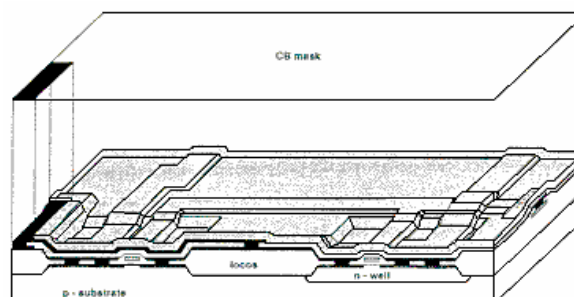
Máscara Contacto 2 (COS)

Na mesma sequência, o metal 2 é depositado quando a máscara que pode ser vista na outra figura citada abaixo;



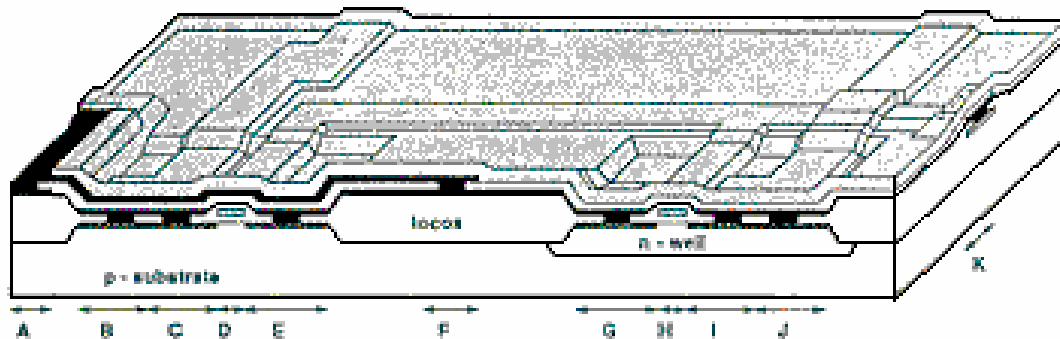
- Máscara do metal 2

O passo final é a deposição da camada de proteção sobre toda a superfície, o que implica outra máscara para deixar abertos os contactos para os *bonpads*, de forma a permitir o contacto com o exterior. A máscara chama-se *overlay* (CB), como se vê na figura que segue abaixo. Tipicamente esta camada de proteção consiste em aplicar dióxido de silício seguido por mais uma camada de nitrato de silício; o nitrato é usado porque constitui uma boa barreira contra impurezas e proteção anti-riscos.



Máscara CB

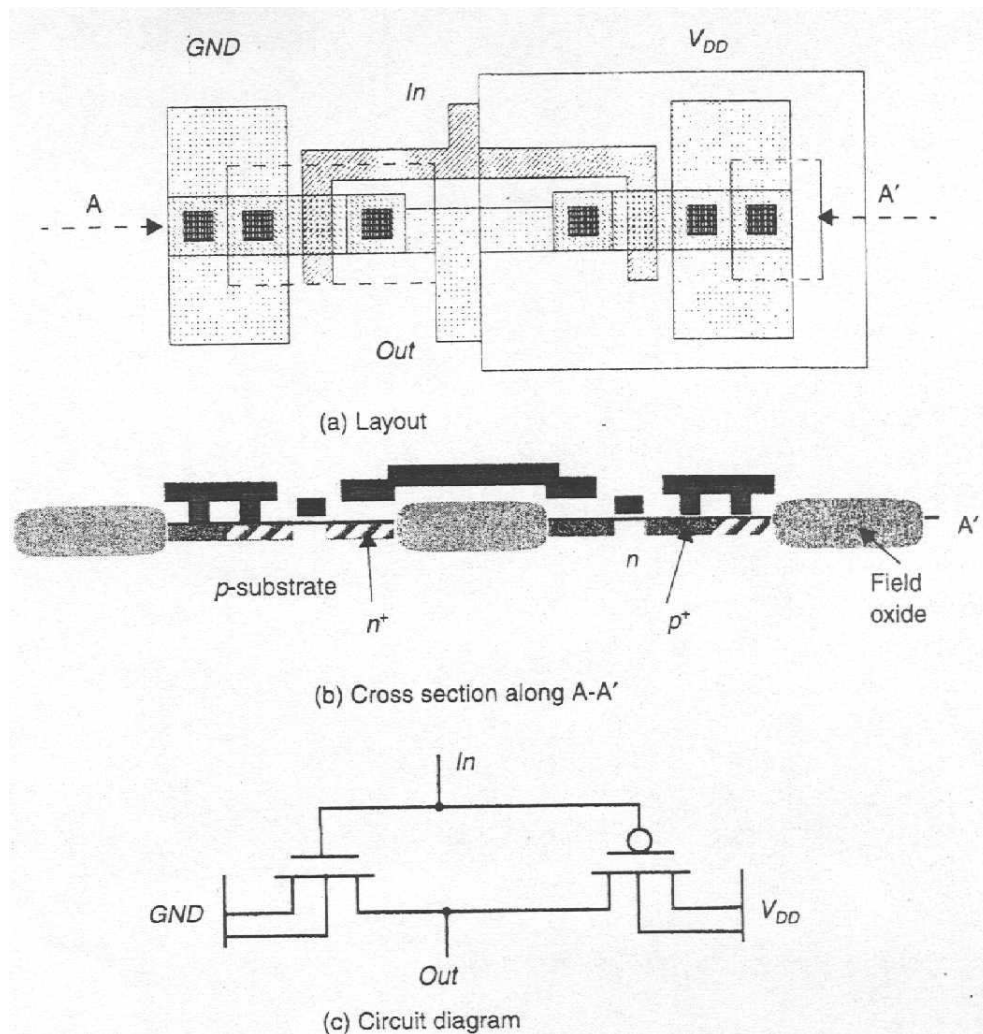
Os dispositivos realizados, no exemplo um inversor, ficam com o aspecto da figura abaixo, onde se pode identificar os locais das ligações.



Vista de corte de um Inversor completo

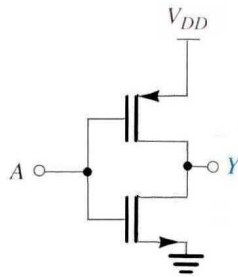
- | | |
|----------------------------|------------------------------|
| A- <i>BondPad</i> de Saída | G- <i>Drain</i> do PMOS |
| B- Contacto do Substrato | H- <i>Gate</i> do PMOS |
| C- <i>Source</i> do NMOS | I- <i>Source</i> do PMOS |
| D- <i>Gate</i> do NMOS | J- Contacto da <i>n-well</i> |
| E- <i>Drain</i> do NMOS | K- Nó de entrada |
| F- Nó de Saída | |

A sequência descrita demonstra os passos básicos deste processo, o *layout* físico do inversor, em corte e o esquema elétrico segue abaixo:



9.2 Lógica CMOS

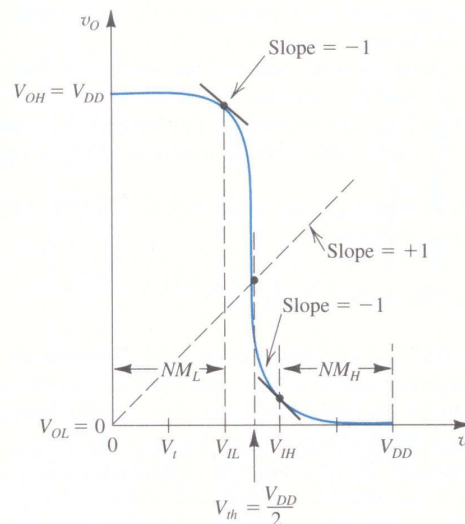
A lógica CMOS – *Complementary MOS* – tem um análise simples e pode ser facilmente entendida pelo estudo do inversor CMOS, representado na figura.



Porta lógica NOT em tecnologia CMOS.

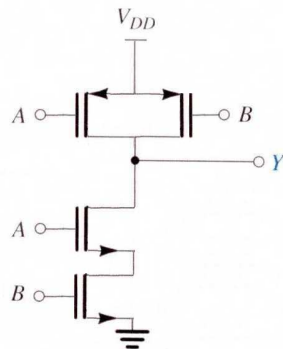
Quando a tensão no nó de entrada A tem uma tensão de entrada igual a V_{DD} - nível lógico HIGH - o transistor NMOS está na zona activa pelo que a sua resistência equivalente é baixa, enquanto que o transistor PMOS está cortado e portanto tem uma resistência equivalente muito elevada (pode considerar-se infinita), sendo assim a tensão no nó de saída Y é puxada para 0V. Por outro lado, quando a tensão de entrada é 0V o transistor PMOS está na zona activa - resistência equivalente baixa - e o transistor NMOS está cortado, e portanto a tensão no nó de saída Y é puxada para V_{DD} .

A característica $v_o(v_i)$ correspondente, representada na figura seguinte, é bastante abrupta e tem boas gamas de definição dos níveis lógicos como se referiu anteriormente. Uma das grandes vantagens da lógica CMOS é que o consumo estático é nulo pois quer o sinal esteja a HIGH quer esteja a Low um dos transistores está cortado e portanto não há corrente. Assim, só há consumo na zona intermédia da característica, que corresponde à zona em que acontecem as comutações.



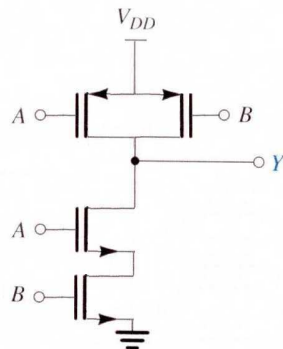
Característica $v_o(v_i)$ de um inversor em tecnologia CMOS.

Para implementar uma porta lógica NAND a partir da porta lógica NOT, é necessário acrescentar um segundo ramo de entrada NMOS em série tal como foi feito para a porta lógica RTL, e um ramo PMOS em paralelo tal é apresentado na figura seguinte.



Porta lógica NAND em tecnologia CMOS.

A complementaridade destes dois acréscimos é também referida como circuito dual, e dá o nome à família lógica. Este raciocínio pode ser também aplicado para construir outro tipo de portas, nomeadamente a porta lógica NOR (que também é uma porta lógica universal) tal como é visível na figura seguinte.

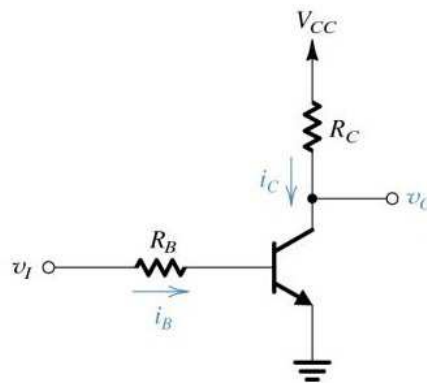


Porta lógica NOR em tecnologia CMOS.

Quando ambos os transistores estão a conduzir, ou seja, quando ambas as tensões de entrada correspondem ao nível lógico HIGH, há corrente na resistência R_c e o nível lógico de saída é Low. Quando qualquer dos dois transistor estiver cortado $i_C = 0$, a tensão de saída é igual a V_{CC} e o nível lógico na saída é HIGH, bastando para isso que uma das tensões de entrada seja Low.

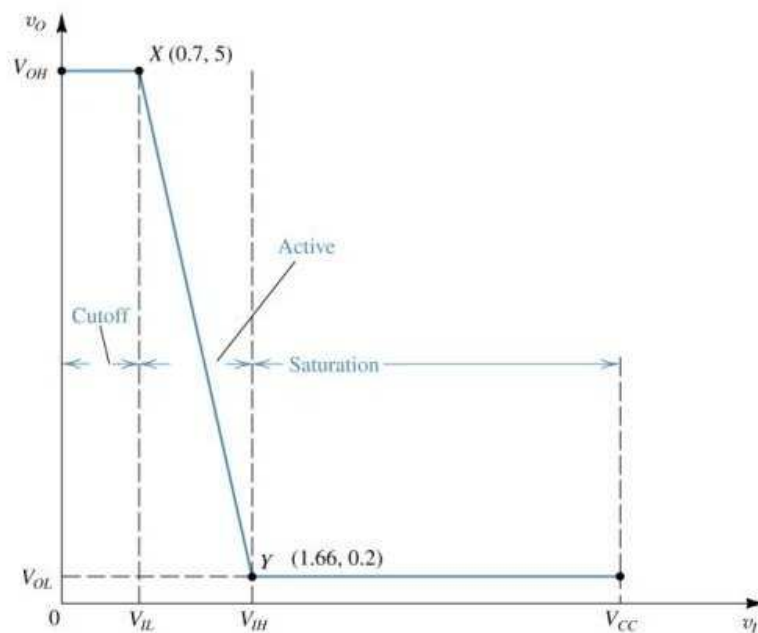
10 Resistor-Transistor-Lógico

A topologia RTL – *Resistor Transistor Logic* – pode ser considerada como uma das arquitecturas base para a realização de portas digitais. A figura seguinte representa uma porta lógica NOT.



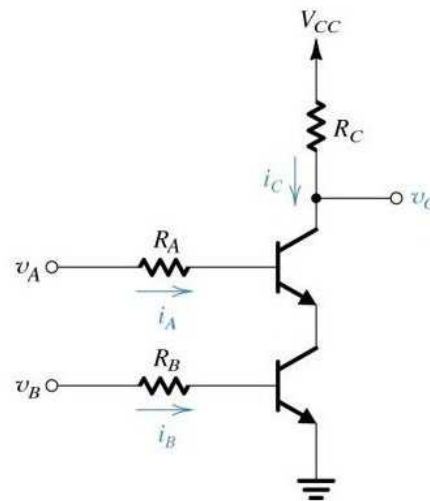
Porta lógica NOT em tecnologia RTL.

O comportamento desta porta é relativamente simples, quando a tensão de entrada tem o nível lógico HIGH o transistor está directamente polarizado e a resistência R_C é dimensionada para que o transistor esteja na zona de saturação, pelo que a tensão de saída é aproximadamente 0.2V o que corresponde ao nível lógico Low. Quando o nível lógico de entrada é Low o transistor está cortado e portanto i_C é igual a zero o que implica que a tensão de saída v_O seja V_{CC} - nível lógico HIGH. Estes dois comportamentos, juntamente com a zona intermédia em que o transistor está na zona activa, são visíveis na característica $v_O(v_I)$ representada na figura seguinte:



Característica $v_O(v_I)$ de um inversor RTL.

É relativamente simples implementar uma porta lógica NAND – que é porta lógica universal, dado que qualquer outra pode ser implementada a partir desta – com base na porta lógica NOT, basta para isso acrescentar um segundo ramo de entrada tal como é apresentado na figura seguinte.

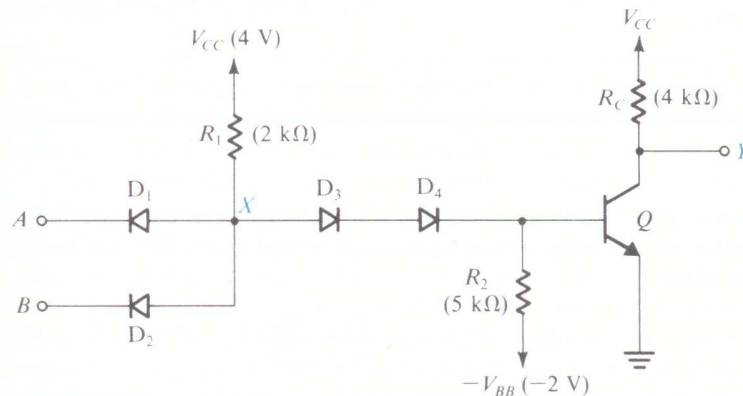


Porta lógica NAND em tecnologia RTL.

Quando ambos os transistores estão a conduzir, ou seja, quando ambas as tensões de entrada correspondem ao nível lógico HIGH, há corrente na resistência R_C e o nível lógico de saída é Low. Quando qualquer dos dois transistor estiver cortado $i_C = 0$, a tensão de saída é igual a V_{CC} e o nível lógico na saída é HIGH, bastando para isso que uma das tensões de entrada seja Low.

11 Diode-Transistor-Lógico

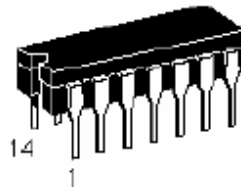
A topologia DTL – *Diode Transistor Logic* – visível na figura seguinte, apresenta tem um comportamento algo à porta lógica NAND usada em lógica RTL.



Porta lógica NAND em tecnologia DTL.

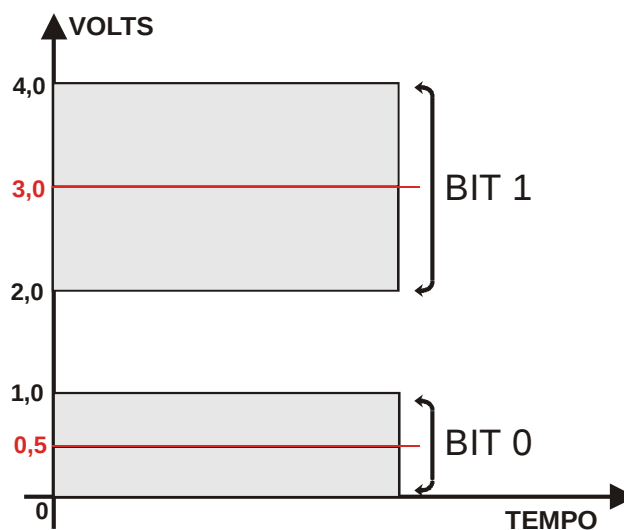
Neste caso, só quando ambos os diodos de entrada D_1 e D_2 estão cortados, é que o transistor tem a junção base-emissor polarizada directamente e impõe o nível lógico Low na saída. Caso qualquer dos dois diodos D_1 e D_2 esteja em condução, a tensão no nó X não é suficiente para polarizar o transistor - que fica cortado e portanto $i_C = 0$ – pelo que a tensão de saída é igual a V_{CC} e o nível lógico na saída é HIGH. Basta portanto que uma das tensões de entrada seja Low para que correntemente o nível lógico de saída seja Low.

PORTAS LÓGICAS



- Um computador digital é uma máquina projetada para armazenar e manipular informações representadas apenas por algarismos ou dígitos e que só podem assumir dois valores distintos, 0 e 1, razão por que é chamado *computador digital*, *sistema digital* ou simplesmente *máquina digital binária*.
- A informação binária (valores 0 e 1) é representada em um sistema digital por quantidades físicas, sinais elétricos, os quais são gerados e mantidos internamente ou recebidos de elementos externos, em dois níveis de intensidade, cada um correspondente a um valor binário (há outras formas de armazenamento de bits internamente em um computador, como campo magnético e sinais óticos).

12 Valores elétricos do sinal binário



- ❑ Circuitos digitais são formados por elementos que são conhecidos como portas (*gates*) lógicas, por permitirem ou não a passagem desses sinais, e os circuitos que contêm as portas lógicas são conhecidos como *circuitos lógicos*.
- ❑ Uma porta (*gate*) é um circuito eletrônico que recebe um ou mais sinais de entrada e produz um sinal de saída, cujo o valor é dependente do tipo de regra lógica estabelecida para a construção do referido circuito.
- ❑ Resumidamente, um computador digital é contruído, então, contendo uma infinidade de circuitos lógicos ou portas, convenientemente distribuídos e organizados, de modo que alguns servirão para armazenamento de valores, outros permitirão e controlarão o fluxo de sinais entre os componentes e outros, ainda, serão utilizados para realizar operações matemáticas.

CPU (PROCESSADOR) -----> MILHÕES DE PORTAS LÓGICAS

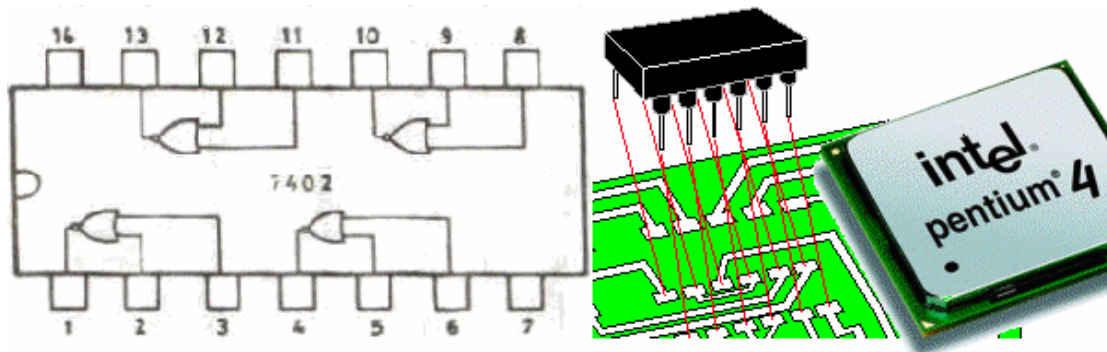
A facilidade do processamento de números binários decorre da existência de apenas dois dígitos, 0 e 1, que podem ser representados por 2 níveis de tensão (exemplo 0 = 0 volt e 1 = 5 volts).

Os símbolos representam um **bloco lógico** com uma ou mais **entradas lógicas** A, B, etc. e uma ou mais **saídas lógicas** S1, S2, etc. As entradas e saídas lógicas só assumem valores correspondentes aos níveis lógicos 0 e 1.

Um bloco lógico executa uma determinada **função lógica** para a qual foi projetado. Essa função determina os valores que as saídas assumem para cada combinação de valores das entradas. Tais relações são muitas vezes exibidas sob a forma de **tabelas de verdade**.

13 Circuitos Integrados

Dispositivo, também chamado de pastilha ou chip, que contem centenas ou milhares de transistor, diodos, resistores e capacitores interligados que compõem as portas lógicas formando um determinado circuito combinatório



SSI (Small Scale Integration): De 1 a 10 portas lógicas

MSI (Médium Scale Integration): Até 100 portas

LSI (Large Scale Integration): Até 100.000 portas

VLSI (Very Large Scale Integration): Acima de 100.000 portas
Pentium → mais de 3 milhões de portas

14 Portas Lógicas

14.1 Porta AND (E)

A porta AND combina dois sinais de entrada de forma equivalente a um circuito em série, para produzir um único sinal de saída, ou seja, ela produz uma saída 1, se todos os sinais de entrada forem 1; caso qualquer um dos sinais de entrada for 0, a porta AND produzirá um sinal de saída igual a zero.

PORTA AND (E)		$X = A \cdot B$		
A	B	A	B	X
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	1	1	1

14.2 Porta OR (OU)

A porta OR combina dois sinais de entrada de forma equivalente a um circuito em paralelo, para produzir um único sinal de saída, ou seja, ela produz uma saída 1, se qualquer um dos sinais de entrada for igual a 1; a porta OR produzirá um sinal de saída igual a zero apenas se todos os sinais de entrada forem 0.

PORTA OR (OU)		$X = A + B$		
A	B	A	B	X
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	1	1	1

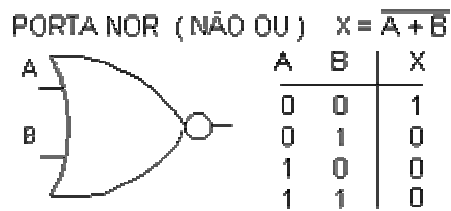
14.3 Porta NAND (Não-E)

A porta NAND equivale a uma porta AND seguida por uma porta NOT, isto é, ela produz uma saída que é o inverso da saída produzida pela porta AND.

PORTA NAND (NÃO E)		$X = \overline{A \cdot B}$		
A	B	A	B	X
0	0	0	0	1
0	1	0	1	1
1	0	1	0	1
1	1	1	1	0

14.4 Porta NOR (Não-OU)

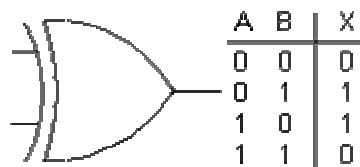
A porta NOR equivale a uma porta OR seguida por uma porta NOT, isto é, ela produz uma saída que é o inverso da saída produzida pela porta OR.



14.5 Porta XOR (OU Exclusivo)

A porta XOR compara os bits; ela produz saída 0 quando todos os bits de entrada são iguais e saída 1 quando pelo menos um dos bits de entrada é diferente dos demais.

PORTA XOR ou OU EXCLUSIVO
(EXCLUSIVE OR) $X = A \oplus B$



Disjunção exclusiva

Ou exclusivo chamada também disjunção exclusiva, conhecido geralmente por XOR ou por EOR (também XOU ou EOU), é uma operação lógica em dois operandos que resulta em um valor lógico verdadeiro se e somente se exactamente um dos operandos tem um valor verdadeiro.

Definição

A **disjunção exclusiva** (escrito como $\oplus$, $+$, ou ainda $\neq$) é uma operação sobre dois ou mais valores lógicos, tipicamente os valores de duas proposições, que produz um valor verdadeiro apenas se a quantidade de operadores verdadeiros for ímpar.

Tabela de Verdade

A tabela de verdade para A XOR E de três entradas é a seguinte:

Ou exclusivo		
A	E	A XOR E
0	0	0
0	1	1
1	0	1
1	1	0

Fórmula mais utilizada: $S = \bar{A} \times E + A \times \bar{E}$

14.6 Porta XNOR (Não OU Exclusivo)

Porta XNOR é uma porta logica também conhecida como **porta lógica coincidência** cuja operação é a inversa da porta XOR (ou-exclusivo). A operação XNOR é representada pelo símbolo ($\cdot$) e sua fórmula pode ser resumida através da fórmula $Y = A (\cdot) E$.

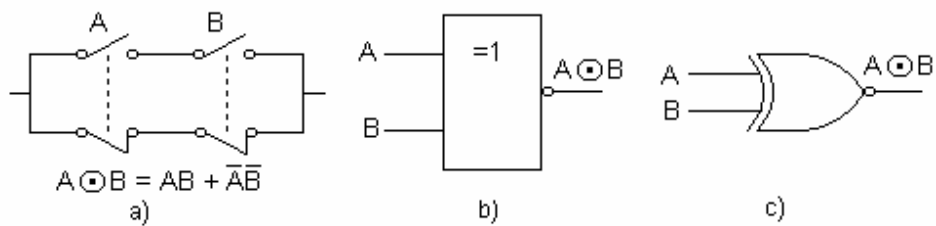
Ela recebe duas entradas (A e E) e sua tabela verdade se comporta de maneira que quando as entradas são iguais, (0 e 0) ou (1 e 1), a saída retorna 1 e quando são diferentes, (1 e 0) ou (0 e 1), retorna 0. Daí a justificativa para ela também ser conhecida como porta lógica coincidência, pois ela retorna 1(verdadeiro) apenas quando as entradas coincidem.

Tabela Verdade

A	E	A XNOR E
0	0	1
0	1	0
1	0	0
1	1	1

Para entradas iguais, saída é 1. Para entradas diferentes, saída é 0.

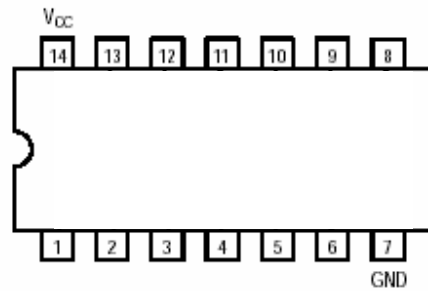
Símbolos



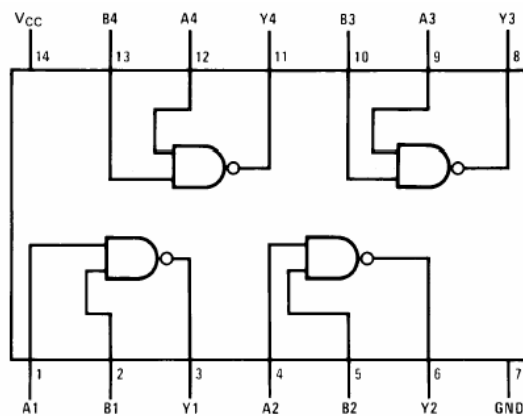
(a) circuito para porta XNOR (b) e (c) Diferentes representações simbólicas para a porta XNOR

Fórmula mais utilizada: $A \times E + \bar{A} \times \bar{E}$

Principais portas lógicas:



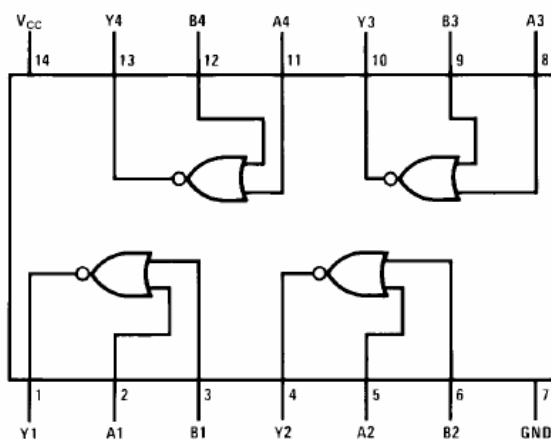
Portas Lógicas em Circuito Integrado



7400

Quad 2-input NAND Gate

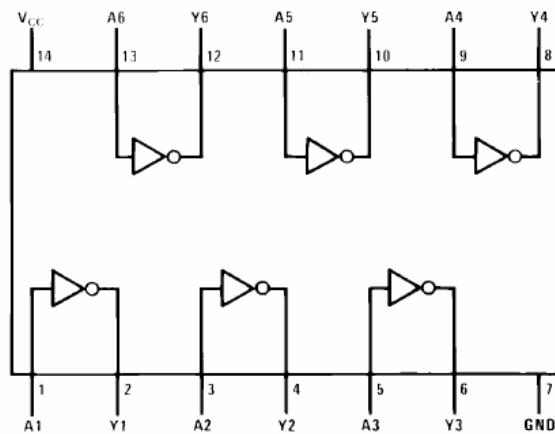
Quatro Portas NAND
de 2 entradas



7402

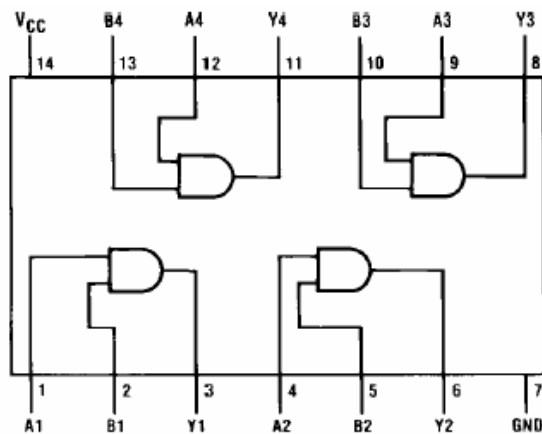
Quad 2-input NOR Gate

Quatro Portas NOR
de 2 entradas



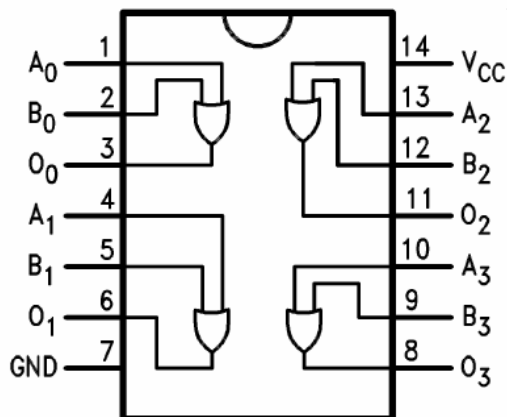
7404

Hex Inverting Gates
Seis Portas Inversoras



7408

Quad 2-input AND Gate
Quatro Portas AND
de 2 entradas



7432

Quad 2-input OR Gate
Quatro Portas OR
de 2 entradas

Exemplo de circuitos utilizando portas lógicas:

A) Uma campainha que toca (saída) se o motorista der a partida no motor do carro (entrada) sem estar com o cinto de segurança afivelado (entrada).

Se a ignição for ACIONADA (1) e o cinto estiver DESAFIVELADO (1), a campainha é ACIONADA (1). Caso contrário, a campainha não toca.

Tabela Verdade:

Ignição	Cinto	Campainha
0	0	0
0	1	0
1	0	0
1	1	1

Basta utilizar uma porta AND.

B) Detector de incêndio com vários sensores (entradas) e uma campainha para alarme (saída).

Se QUALQUER UM dos sensores for acionado (significando que um dos sensores detectou sinal de incêndio), a campainha é ACIONADA.

Tabela verdade:

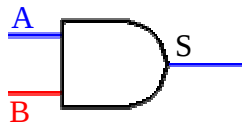
Sensor 1	Sensor 2	Campainha
0	0	0
0	1	1
1	0	1
1	1	1

Basta utilizar uma porta OR.

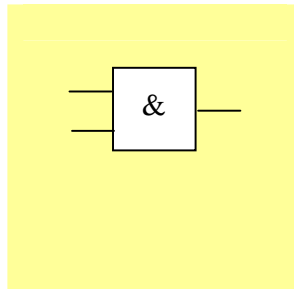
Veja abaixo as principais portas lógicas, seus símbolos (antigos e novos), a expressão da função e a tabela da verdade.

Porta lógica E (AND)

Símbolo antigo



Símbolo novo



Expressão da função

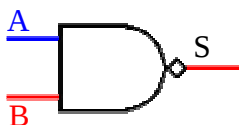
$$S = A \times B$$

Tabela de verdade da função E

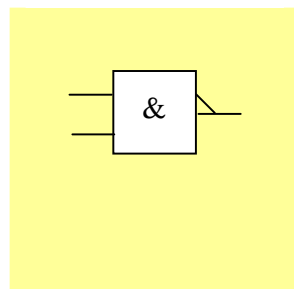
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Porta lógica NÃO E (NAND)

Símbolo antigo



Símbolo novo



Expressão da função

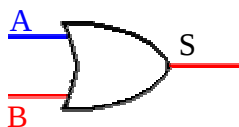
$$S = \overline{A \times B}$$

Tabela de verdade da função NÃO E

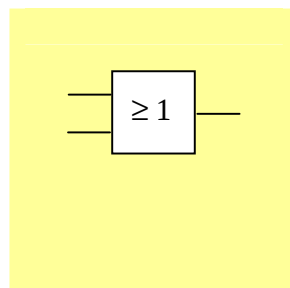
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Porta lógica OU (OR)

Símbolo antigo



Símbolo novo



Expressão da função

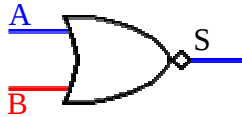
$$S = A + B$$

Tabela de verdade da função OU

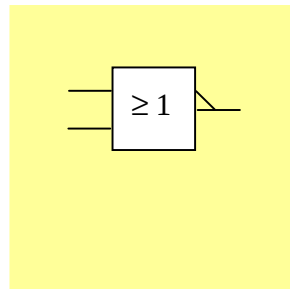
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Porta lógica NÃO OU (NOR)

Símbolo antigo



Símbolo novo



Expressão da função

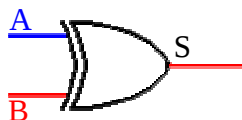
$$S = \overline{A + B}$$

Tabela de verdade da função NÃO OU

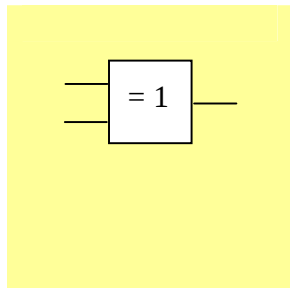
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Porta lógica OU EXCLUSIVO (XOR)

Símbolo antigo



Símbolo novo



Expressão da função

$$S = A \oplus B$$

Tabela de verdade da função OU EXCLUSIVO

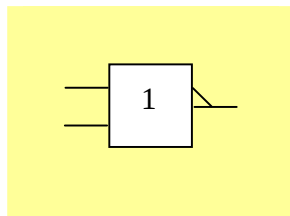
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Porta lógica NEGAÇÃO (NOT)

Símbolo antigo



Símbolo novo



Expressão da função

$$A = \overline{S}$$

Tabela de verdade da função NEGAÇÃO

A	S
1	0
0	1

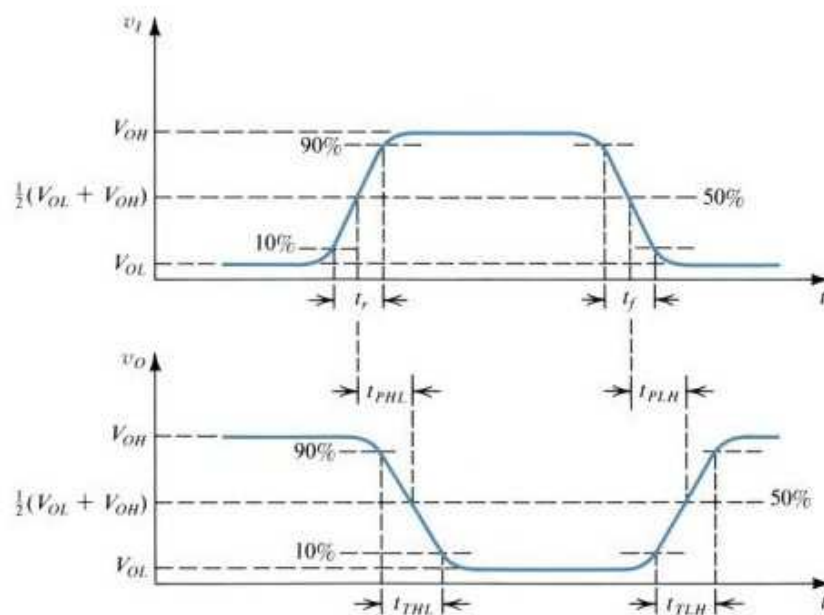
14.7 Tempos de subida e descida

O tempo que uma porta lógica demora a comutar está relacionado com os tempos de subida e descida, ou seja com os tempos de comutação de Low para High e de High e Low respectivamente. O tempo de subida (t_r) – *rise time* - é geralmente definido como o tempo que a tensão na saída da porta lógica demora a subir entre 10% e 90% do seu valor máximo. De forma complementar,

o tempo de descida (t_f) – *fall time* - é definido como o tempo que a tensão na saída demora a descer entre 90% e 10% do seu valor máximo.

14.8 Tempo de Propagação

O tempo de propagação de Low para High (t_{PLH}) e o tempo de propagação de High e Low (t_{PHL}) são definidos como o tempo que a tensão de saída demora a atingir o valor médio entre os valores máximo e mínimo da tensão de saída, desde o momento em que a entrada comuta. Ou seja, o tempo que decorre entre a definição do nível lógico de entrada e a definição do nível lógico na saída. Naturalmente, é desejável que t_{PHL} e t_{PLH} sejam iguais dado que o pior destes tempos define a velocidade máxima a que a porta lógica pode operar. A figura seguinte mostra graficamente os tempos de propagação e de comutação de Low para High e de High e Low.



Tempo de propagação e tempo de subida e descida.