Capítulo 5

A Tecnologia de Circuitos CMOS

Everything should be made as simple as possible,
but no simpler.

Albert Einstein.

Este capítulo[1] contém uma breve introdução à tecnologia de circuitos digitais CMOS (pronuncia-se "cê-mós"), sigla para *Complementary Metal-Oxide Semiconductor*, que é usada na implementação de circuitos digitais. O método de fabricação e o comportamento dos dois dispositivos básicos da tecnologia – transistores do tipo P e do tipo N – são introduzidos na Seção 5.1. Com estes dispositivos podem ser implementados o inversor e as portas lógicas *nand* e *nor*, bem como circuitos com um terceiro "nível lógico" para além dos níveis 0 e 1. Tais circuitos são estudados na Seção 5.2. A Seção 5.3 introduz o importante tópico "tempo de propagação de sinais" e discute algumas das razões para os atrasos nos sinais introduzidos pelas portas lógicas.

Para mais detalhes veja Weste & Harris [WH10], Rabaey [RCN03], ou Sedra & Smith [SS90], que são textos com ênfase em Eletrônica. A Seção 5.1 é baseada no texto de Clark [Cla80]; as definições de *tempo de propagação* e *tempo de contaminação*, na Seção 5.4 são baseadas em material da disciplina 6.004 – *Computation Structures* do MIT, de 2013.

5.1 Semicondutores e Transistores CMOS

Um *condutor* sólido é tipicamente um metal, no qual os elétrons da camada de valência são fracamente atraídos pelo núcleo e se movem livremente entre átomos vizinhos. A cada elétron corresponde um próton, e portanto um condutor em estado natural é eletricamente neutro. Um *isolante* sólido é um material no qual os elétrons da camada de valência estão firmemente ligados ao núcleo. No que se segue, ignoramos a condução em fluídos (íons na água ou ar) ou plasma (centelhas), e isolantes líquidos ou gasosos.

Materiais *semicondutores* diferem dos condutores e dos isolantes porque em algumas circunstâncias se comportam como isolantes e em outras como condutores. Estes materiais são empregados na construção de circuitos digitais para implementar 'chaves' que podem ser controladas por sinais digitais. Tais chaves tem dois estados, ou estão fechadas – conduzindo corrente elétrica – ou estão abertas – impedindo a passagem de corrente.

[1] Versão de 18 de novembro de 2020.

O que segue é uma descrição muito simplificada da fabricação e do funcionamento de circuitos CMOS.

5.1.1 Materiais e Fabricação

O material mais frequentemente empregado na fabricação de circuitos CMOS é o silício (Si). Um bastão de silício, tipicamente com 20 a 30cm de diâmetro e um a dois metros de comprimento, é serrado em discos com 0,3 a 1 mm de espessura. O bastão, que é um monocristal de silício, é fabricado com elevadíssimo nível de pureza. O disco de silício é polido até que na sua superfície ocorra menos do que um defeito por centímetro quadrado. Por 'defeito' entende-se uma cavidade na qual faltam poucas dezenas de átomos, ou a existência, no monocristal, de átomos que não sejam de silício.

Durante a fabricação de circuitos integrados são acrescentados impurezas ao substrato de silício que agregam elétrons ao cristal – tornando-o eletricamente negativo – ou que subtraem elétrons – tornando-o eletricamente positivo. Estas impurezas, chamadas de *dopantes*, são incorporadas ao retículo cristalino pela exposição, em alta temperatura, a uma elevada concentração do dopante. Este processo é conhecido como *dopagem por difusão*.

A superfície do disco é coberta com um polímero, com certas áreas mantidas sem a cobertura. Uma 'máscara' é usada para definir quais áreas são cobertas e quais são expostas. Tais máscaras são semelhantes ao negativo de uma fotografia em preto e branco; as partes claras na máscara são expostas, e as partes escuras são encobertas durante o processo de difusão.

O disco é levado a um forno com temperatura da ordem de 800 a 1000 C, no qual é insuflado gás de fósforo (P), arsênio (As), ou de boro (B). Átomos destes elementos se difundem através do cristal de silício e se alojam na estrutura cristalina. Após a difusão, a concentração de dopantes nas áreas expostas pela máscara é da ordem de 1 átomo de dopante para 10^7 átomos de silício.

Um transistor CMOS é produzido por uma sequência de passos de fabricação. Tipicamente um passo se inicia com a exposição do polímero para formar a máscara que é usada para selecionar as regiões do disco que serão expostas, seguido de difusão com o tipo de dopante e a concentração adequados, e concluído pela remoção da máscara. Dependendo do processo de fabricação, são necessários de 100 a 400 passos de processamento, o que toma 2 a 3 semanas.

O átomo de silício possui 4 elétrons na sua camada de valência. Num retículo cristalino composto somente de silício, cada átomo se liga a exatamente quatro vizinhos e o cristal tem carga elétrica líquida igual a zero porque todos os elétrons das camadas de valência estão ligados aos seus átomos.

O átomo de fósforo possui 5 elétrons na sua camada de valência, e quando se liga a quatro átomos no retículo cristalino de silício, o quinto elétron fica fracamente ligado ao átomo de fósforo. Cada átomo de fósforo contribui com um elétron adicional ao material e a região dopada com fósforo é chamada de *semicondutor tipo N* porque a região possui um excesso de portadores de carga negativa. Fósforo é chamado de *doador* porque acrescenta elétrons ao material.

O átomo de boro possui 3 elétrons na sua camada de valência, e quando se liga ao retículo cristalino, uma das quatro ligações fica faltando. Esta ligação resulta no *buraco* deixado pelo elétron faltante, e o buraco é portanto um portador de carga positiva. Boro é chamado de *receptor* porque cada átomo de boro 'aceita' um elétron, contribuindo com uma carga positiva

ao material. A região dopada com boro é chamada de *semicondutor de tipo P* porque possui excesso de portadores de carga positiva.

Os elétrons da camada de valência de um material condutor tem alta mobilidade porque são fracamente ligados aos átomos, enquanto que num isolante os elétrons são fortemente ligados aos seus átomos e sua mobilidade é pequena. Num material semicondutor, a mobilidade dos elétrons ou dos buracos depende da concentração dos dopantes, e da aplicação de um campo elétrico para aumentar a energia dos portadores de carga. Como nos condutores, se uma diferença de potencial é aplicada sobre o material, elétrons se moverão lentamente na direção do potencial mais positivo, e buracos se moverão, ainda mais lentamente, na direção do potencial mais negativo. A mobilidade dos elétrons é aproximadamente o triplo daquela dos buracos. Removido o campo elétrico, o material volta a se comportar como um isolante.

A velocidade de propagação da luz no vácuo (c) é de 300.000 km por segundo. A velocidade com que o campo elétrico se propaga através de um meio físico, tal como um fio de cobre ou fibra ótica, é de $0,6$ a $0,7c$, que é da ordem de 20 cm por nanosegundo. Quando o meio físico sólido é submetido a um campo elétrico, a velocidade de propagação de um único elétron é de alguns milímetros por segundo. Esta é uma distinção importante, e talvez contraintuitiva: o campo eletromagnético se propaga a 20cm/ns enquanto que os portadores de carga individuais se movem a 10 mm/s.

5.1.2 Operação dos Transistores

O termo *transistor* é uma abreviatura para *transfer-resistor*, ou "resistor de transferência". Este termo define dispositivos de três terminais, nos quais a resistência entre um par de terminais pode ser controlada pelo terceiro terminal. Dependendo dos potenciais elétricos relativos entre os terminais do transistor, este se comporta como uma baixa resistência – um condutor – ou como uma alta resistência – um isolante. É este comportamento que permite que transistores sejam usados como chaves.

A Figura 5.1 mostra um corte transversal de um transistor do tipo N. O substrato contém material do tipo P e os dois poços são dopados com material do tipo N. A cada poço são ligados os terminais com a *fonte* e o *dreno* de portadores de carga do transistor. O terceiro terminal, chamado de *gate*, é isolado do substrato por uma fina camada de óxido de silício, que é um excelente isolante. Esta camada isolante tem espessura de 20Å, que é a espessura de 6 ou 7 átomos empilhados. O *gate* consiste de uma camada de metal ou de polisilício[2]. Entre os dois poços, abaixo do *gate*, se estabelece o *canal de condução* do transistor.

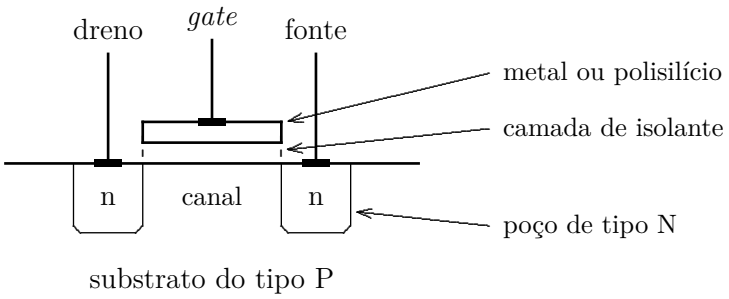


Figura 5.1: Estrutura física de um transistor CMOS tipo N.

[2]O termo '*gate*' não tem relacionamento com "porta lógica", ou "*logical gate*".

O substrato é mantido no potencial elétrico correspondente ao nível lógico 0. Se o potencial correspondente ao nível lógico 1 é aplicado no *gate*, como mostrado no lado esquerdo da Figura 5.2, este terminal fica com carga líquida positiva. No substrato, sob o *gate*, acumulam-se cargas elétricas negativas atraídas da vizinhança pelo campo elétrico provocado pela carga positiva no *gate*. O acúmulo de cargas sob o *gate* transforma o canal num condutor, e se uma diferença de potencial for aplicada entre fonte e dreno, por causa da baixa resistência no canal, uma corrente se estabelece da fonte para o dreno, como indica a seta no diagrama do lado direito da Figura 5.2. Valores típicos para o potencial que representa o nível lógico 1 são de 1,25 a 2,5V.

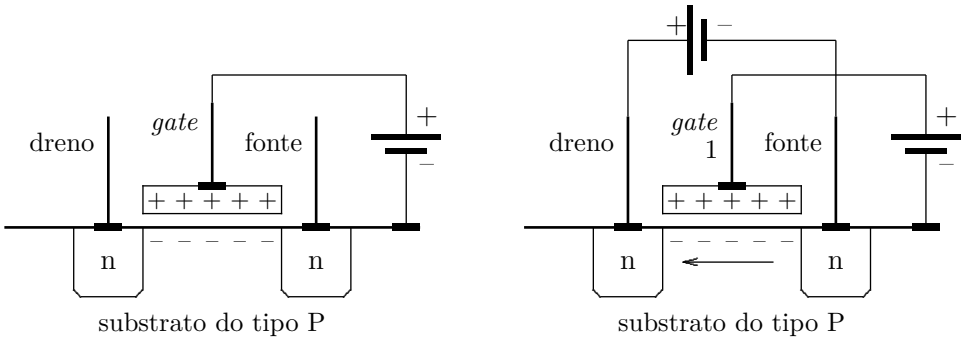


Figura 5.2: Modo de operação de um transistor CMOS tipo N.

Se o potencial do *gate* for reduzido para 0V, a corrente se interrompe porque os elétrons se afastarão do canal retornando aos seus íons de origem, aumentando a resistência do canal. Quando a tensão no *gate* corresponde ao nível lógico 1, o transistor conduz e se comporta como uma chave fechada porque há um caminho de baixa resistência entre fonte e dreno. Quando a tensão no *gate* corresponde a 0, o transistor se comporta como uma chave aberta e os terminais fonte e dreno ficam isolados.

Este transistor é chamado de *transistor tipo N* porque os portadores de carga são elétrons, com carga negativa portanto. Num *transistor tipo P*, as polaridades se invertem e os portadores de carga são buracos.

Espaço em branco proposital.

A Figura 5.3 mostra a estrutura e indica o modo de funcionamento de um transistor tipo P. O substrato é do tipo N, e os dois poços são do tipo P. O substrato é mantido no mesmo potencial que a fonte de alimentação. Se o potencial no *gate* corresponde ao nível lógico 0, este fica com carga líquida negativa, e esta carga afasta os elétrons sob o *gate* e expõe buracos que reduzem a resistência no canal. Se um potencial positivo é aplicado entre fonte e dreno, se estabelecerá um fluxo de buracos da fonte para o dreno.

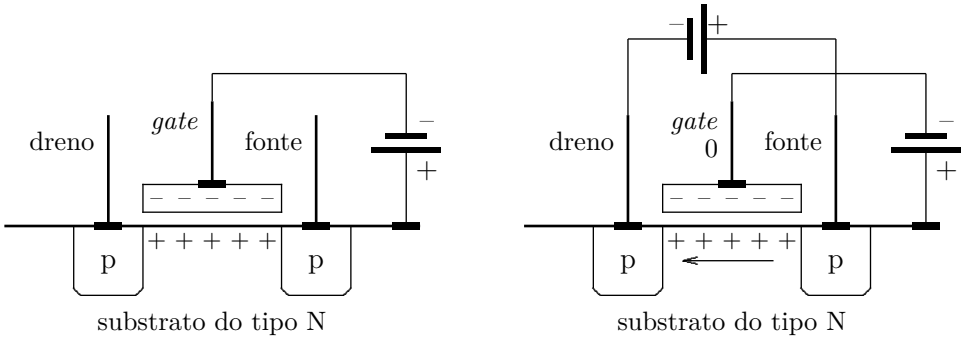


Figura 5.3: Modo de operação de um transistor CMOS tipo P.

Se a fonte de um transistor N for ligada ao nível lógico 1, no dreno o sinal elétrico será equivalente a um nível 1 'fraco' porque os portadores de carga são elétrons, e não buracos. Efeito similar ocorre com transistores P: se a fonte for ligada a um nível lógico 0, o dreno apresentará uma versão fraca de 0 porque seus portadores de carga são buracos, e não elétrons. A 'força', ou a intensidade, de um sinal é relacionada à capacidade de transportar corrente elétrica em quantidade suficiente para a correta operação dos circuitos ligados àquele sinal. No que concerne à 'força' dos sinais, transistores são simétricos e fonte e dreno se comportam da mesma forma.

Os transistores descritos acima são chamados de FETs, ou *Field Effect Transistors* (Transistor de Efeito de Campo), porque o canal somente se estabelece pelo o efeito do campo elétrico provocado pelas cargas acumuladas no *gate*. O nome "por extenso" destes transistores é MOSFETs, abreviatura para *Metal Oxide Semiconductor, Field Effect Transistor*, e o *Metal-Oxide-Semiconductor* indica a construção do transistor com um *gate* metálico sobre a camada de óxido de silício, por sua vez sobre um substrato de semicondutor.

5.2 Implementação de Portas Lógicas

Esta seção mostra como construir portas lógicas com transistores CMOS. A Seção 5.2.1 emprega chaves controladas como uma primeira abstração para os transistores e descreve os circuitos do inversor e das portas *nand* e *nor*. Na Seção 5.2.2, as chaves são substituídas por modelos lógicos para os transistores, modelos estes que permitem que se empregue a abstração de bits para descrever o comportamento dos circuitos com transistores CMOS.

5.2.1 Circuitos com chaves

Os transistores descritos na Seção 5.1.2 se comportam como chaves e por isso é interessante examinarmos a implementação de funções lógicas com chaves, antes de abordar a implementação com transistores.

Uma *chave digital* é um dispositivo de três terminais, um terminal de controle que abre ou fecha o contato entre os outros dois terminais. Uma chave *normalmente aberta*, ou tipo N, é mostrada na Figura 5.4. A chave é "normalmente aberta" porque o sinal de controle deve ser ativado – colocado no nível lógico 1 – para fechar seus contatos.

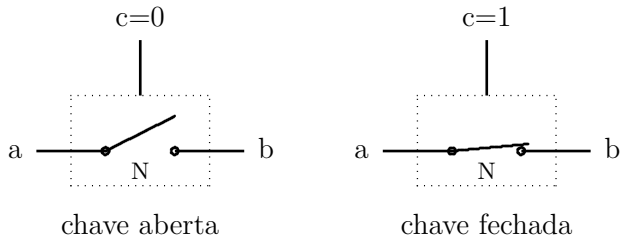


Figura 5.4: Símbolos para as chaves digitais de tipo N.

Se o terminal de controle está em 1, então os contatos se fecham, do contrário, os contatos abrem. Seu comportamento é descrito pela Equação 5.1.

$$\text{chave tipo N}: \quad (c = 1) \Rightarrow (a = b) \tag{5.1}$$

Se o sinal de controle está ativo ($c = 1$), então os níveis lógicos em a e b são iguais. Do contrário, nada se pode afirmar a respeito de seus valores.

Note que não há contato entre o terminal de controle e os terminais a e b. Chaves são uma abstração para o comportamento dos transistores: o terminal de controle c é o *gate* do transistor e os terminais a e b, ligados ao contato da chave, são a fonte e o dreno no transistor.

É conveniente usarmos chaves nas quais o sinal de controle é ativo em 0. Uma chave *normalmente fechada*, tipo P, é mostrada na Figura 5.5. Com o sinal de controle inativo – colocado no nível lógico 0 – os contatos ficam fechados. O círculo no terminal de controle indica a inversão, e seu comportamento é descrito pela Equação 5.2.

$$\text{chave tipo P}: \quad (z = 0) \Rightarrow (x = y) \tag{5.2}$$

Se o sinal de controle está ativo ($z = 0$), então os níveis lógicos em x e y são iguais. Do contrário, nada se pode afirmar a respeito de seus valores.

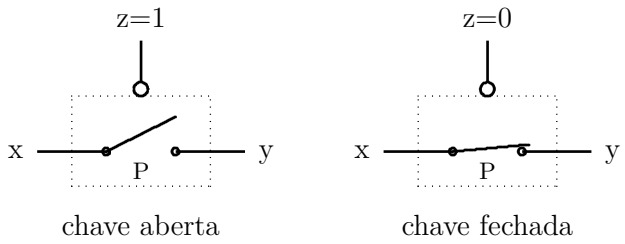


Figura 5.5: Símbolos para as chaves digitais de tipo P.

Inversor O circuito com chaves que implementa um inversor é mostrado na Figura 5.6. As linhas horizontais no topo e na base representam as ligações à fonte de alimentação; a linha no topo do diagrama é, por convenção, a ligação ao potencial mais alto (VCC) que representa uma fonte inesgotável de nível lógico 1, enquanto que a linha na base é a ligação à referência de tensão (GND, de *ground*), que representa uma fonte inesgotável de nível lógico 0. Do ponto de vista elétrico, estas são ligações a um nível lógico 1 'forte', e a um nível lógico 0 'forte'.

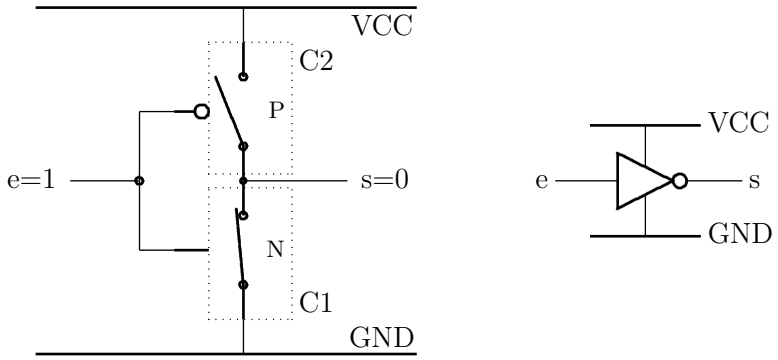


Figura 5.6: Inversor implementado com chaves.

O circuito, com duas chaves, mantém a saída s ligada a VCC (1) ou a GND (0). Se a entrada e está em 1, a chave C1 fica fechada, ligando s a 0. Se a entrada está em 0, a chave C2 fica fechada, ligando s a 1. Os sinais de controle de C1 e de C2 são complementares: o terminal de controle de C2 é ativo em 0, e o de C1 é ativo em 1. Se as duas chaves forem ativadas simultaneamente, ocorrerá um curto circuito na fonte de alimentação, através de C1 e de C2, o que pode danificar o dispositivo ou provocar o seu mau funcionamento, sendo as duas ocorrências altamente indesejáveis.

Ligação em Série A ligação de duas chaves *em série* equivale à conjunção de dois sinais, como indicado na Figura 5.7 – o sinal x é ligado ao sinal y, se ambos os terminais de controle, p e q estão em 1. Se p ou q estiver em 0, nada se pode dizer sobre os níveis de x e y. A conjunção dos sinais de controle pode ser descrita sucintamente por $[(p = 1) \wedge (q = 1)] \Rightarrow (x = y)$.

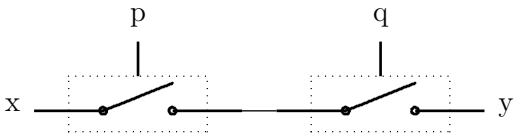


Figura 5.7: Ligação em série.

Ligação em Paralelo A ligação de duas chaves *em paralelo*, como mostra a Figura 5.8, equivale à disjunção de dois sinais. Se qualquer um dentre p ou q estiver em 1, então uma das chaves liga x a y. Se ambos p e q estiverem em 1, as duas chaves ligam x a y; se ambos estiverem em 0, as duas chaves ficam abertas e nada se pode dizer sobre os níveis de x e y. A disjunção dos sinais de controle pode ser descrita por $[(p = 1) \vee (q = 1)] \Rightarrow (x = y)$.

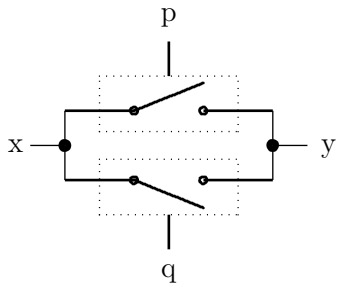


Figura 5.8: Ligação em paralelo.

Porta *nand* O circuito que implementa uma porta *nand* é mostrado na Figura 5.9. A saída s fica em 0 exatamente quando a e b estão em 1, ou $\overline{s} = a \wedge b$. Para tanto, a saída deve ser ligada a 0 por um circuito com duas chaves em série, controladas por a e b. A saída fica em 1 quando, no mínimo, um dentre a ou b é 0, ou $s = \overline{a} \vee \overline{b}$. A saída deve ser conectada a 1 pela ligação em paralelo de duas chaves controladas por $\overline{a}$ e por $\overline{b}$. O circuito da porta *nand* é composto por duas redes, uma que liga a saída à VCC, e outra que liga a saída à GND, e as equações que definem o comportamento dessas redes são equivalentes segundo o DeMorgan. Na Figura 5.9 as entradas a e b estão em 1, e portanto a saída s é 0.

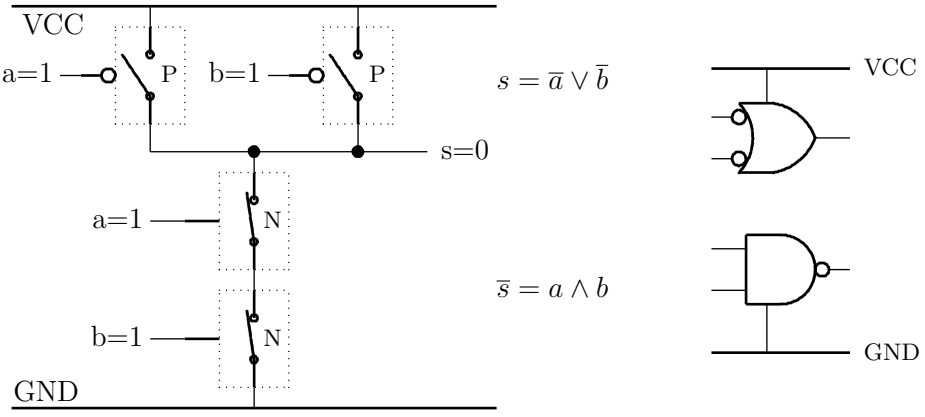


Figura 5.9: Porta *nand* implementada com chaves.

Os dois pontos escuros na linha do sinal s representam ligações nos sinais, como um ponto de solda que une dois ou mais fios. Se duas linhas se cruzam sem o "ponto de solda", então não há ligação entre elas.

Porta *nor* Uma porta *nor* pode ser implementada com chaves aplicando-se a dualidade ao projeto da porta *nand*: cada rede série é substituída por uma rede paralela, e cada rede paralela é substituída por uma rede série. A Figura 5.10 mostra uma porta *nor* implementada a partir do circuito da porta *nand*. A rede ligada em série que conecta a saída a 0 é substituída por uma rede ligada em paralelo; a rede ligada em paralelo que conecta a saída a 1 é substituída por uma rede ligada em série. Tal como na porta *nand*, as equações das duas redes são equivalentes segundo DeMorgan. Na Figura 5.10 a entrada a é 0 e b é 1, e portanto a saída s é 0.

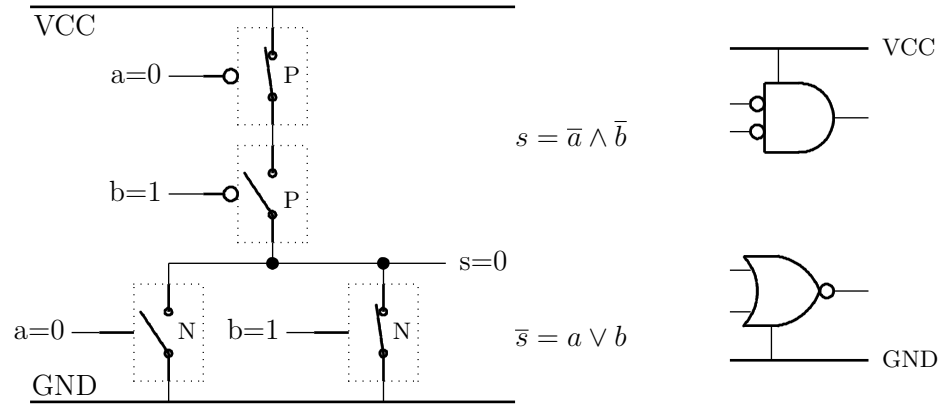


Figura 5.10: Porta *nor* implementada com chaves.

Não é coincidência que as chaves que ligam a saída à VCC tenham o terminal de controle com uma negação. A razão para tal é esclarecida na Seção 5.2.2.

5.2.2 Portas Lógicas CMOS

A Figura 5.11 mostra os símbolos usuais para os dois transistores CMOS, transistores tipo P e tipo N. Estes transistores se comportam como chaves e quando o terminal g (*gate*) está ligado ao nível lógico adequado, o nível lógico dos outros dois terminais f e d é equivalente, porque a chave está fechada.

Figura 5.11: Transistores CMOS.

Quando o *gate* está no outro nível, a chave fica aberta e nada se pode dizer quanto aos valores em f e em d, porque estes dependerão dos circuitos aos quais aqueles terminais estão ligados. O terminal f é chamado de fonte (*source*) porque este terminal é ligado à fonte de cargas elétricas, enquanto que o terminal d é chamado de dreno (*drain*) porque ele é ligado ao 'dreno' por onde as cargas elétricas escoam.

Num transistor P, os portadores da carga no canal entre os terminais fonte e dreno são os 'buracos' que resultam da falta de elétrons, sendo portanto cargas elétricas positivas. Os portadores de carga no canal entre fonte e dreno de um transistor tipo N são elétrons, com carga elétrica negativa. É por causa dos dois tipos de transistores, que são normalmente usados como "pares complementares", que o nome desta tecnologia é *Complementary*-MOS.

O círculo no *gate* do transistor tipo P indica que o nível lógico 0 reduz a resistência entre os terminais fonte e dreno, como uma chave fechada. Complementarmente, o nível lógico 1 no *gate* faz o transistor de tipo N conduzir. A Equação 5.3 define o comportamento dos transistores.

$$\begin{array}{ll} \text{transistor P} & (g = 0) \Rightarrow (f = d) \\ \text{transistor N} & (g = 1) \Rightarrow (f = d) \end{array} \tag{5.3}$$

Os circuitos CMOS são tipicamente constituídos de duas redes, uma rede que "puxa a saída para cima" (para 1, ou *pull-up*) e outra rede que "puxa a saída para baixo" (para 0, ou *pull-down*). A rede que puxa para cima é composta somente de transistores tipo P porque estes conduzem bem sinais de nível lógico 1, e por conta disso, nesta rede há uma conexão com a fonte de alimentação (VCC), que é a fonte de nível lógico 1. A rede que puxa para baixo é composta somente de transistores do tipo N porque estes conduzem bem sinais de nível 0, e esta rede é ligada à referência de tensão (GND), que é a fonte de nível lógico 0.

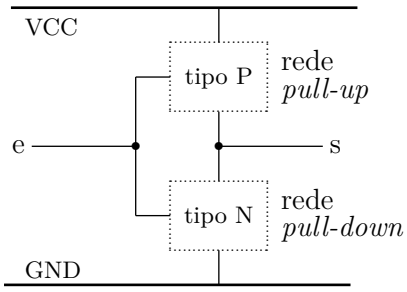


Figura 5.12: Modelo de porta CMOS com redes *pull-up* e *pull-down*.

Como o nome indica, uma *rede* é composta por fontes de alimentação (VCC e GND), por transistores e pelos fios que os interligam. Um *nó* da rede é uma ligação entre dois ou mais de seus componentes. Para simplificar a análise destes circuitos, emprega-se uma abstração para o comportamento dos fios que interligam os componentes: os fios, e portanto os nós do circuito, se comportam como *superfícies equipotenciais*. Isso significa que uma mudança de tensão em qualquer ponto de um fio se propaga instantaneamente por todos os pontos daquele fio, ou por toda a superfície coberta por aquele nó equipotencial.

Inversor

A técnica de projeto para portas lógicas com transistores CMOS é descrita empregando o inversor como exemplo. A Figura 5.13 mostra o Mapa de Karnaugh do inversor com entrada e e saída s.

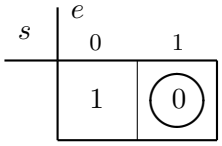


Figura 5.13: Mapa de Karnaugh para o inversor.

Para projetar a rede que puxa para baixo (*pull-down*), a função lógica que define esta rede é obtida da(s) célula(s) do mapa que estão preenchidas com 0, como indicado pelo agrupamento

da célula 1. Neste caso, a rede contém um transistor N cujos terminais fonte e dreno são GND e s respectivamente, e o *gate* é e. A rede que puxa para cima (*pull-up*) é o dual da função *pull-down*, e neste caso, $\textit{pull-up} = \overline{\textit{pull-down}} = \overline{e}$. Esta rede consiste de um transistor P cuja fonte é ligada a VCC, o dreno à s, e o *gate* é controlado por e. Note que a inversão no *gate* do transistor P corresponde à negação de e e portanto não é necessário inverter o sinal ligado ao terminal *gate*. A implementação de um inversor é mostrada na Figura 5.14. Normalmente, as ligações à fonte de alimentação são omitidas dos esquemáticos, embora estas ligações devam sempre existir.

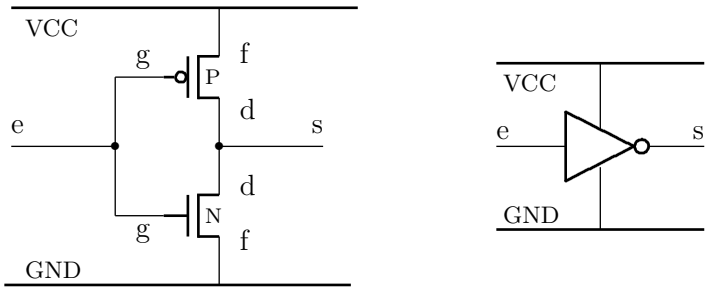


Figura 5.14: Inversor CMOS.

O sinal de entrada e é ligado ao *gate* dos dois transistores e a saída s é ligada aos seus drenos. Quando a entrada está em 1, o transistor P está 'aberto' (circuito aberto) e com uma ligação de alta resistência entre sua fonte e a saída. O transistor N está 'fechado' (circuito fechado) e portando há um caminho de baixa resistência entre a saída e GND, fazendo $s = 0$. Quando a entrada está em 0, o transistor P é uma chave fechada, ligando a saída à VCC, enquanto que o transistor N é uma chave aberta. A Figura 5.15 mostra o inversor nas duas situações. O transistor que está 'aberto' é mostrado com linhas pontilhadas, indicando que aquele transistor se comporta como um isolante.

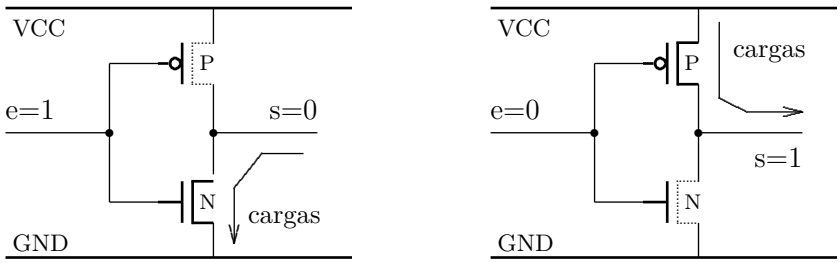


Figura 5.15: Operação do circuito inversor CMOS.

Porta *nor*

O Mapa de Karnaugh da porta *nor* é mostrado na Figura 5.16. As células preenchidas com 0 estão agrupadas e estas determinam a rede que puxa a saída para baixo: $\textit{pull-down} = b \vee a$. Esta rede é implementada com uma ligação em paralelo de dois transistores N, com os sinais a e b ligados aos seus *gates*, como mostra a Figura 5.17. A rede que puxa para cima é o dual da rede *pull-down*: $\textit{pull-up} = \overline{\textit{pull-down}} = \overline{b} \wedge \overline{a}$. Esta rede é implementada pela ligação em série de dois transistores P, com seus *gates* ligados a a e b. Lembre que as inversões da função *pull-up* estão implícitas no comportamento dos transistores P.

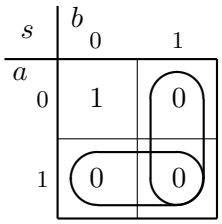


Figura 5.16: Mapa de Karnaugh para a porta *nor*.

A implementação da porta *nor* é mostrada na Figura 5.17. A rede que puxa para 1 comporta-se como uma porta *and* com as entradas complementadas, enquanto que a rede que puxa para 0 comporta-se como uma porta *nor*. O Teorema de DeMorgan garante que as "portas lógicas" destas duas redes têm comportamento equivalente. A saída é 0 se qualquer das entradas estiver em 1 – a ligação em paralelo dos transistores N equivale a $a \vee b$. A saída é 1 somente se as duas entradas forem 0 porque a ligação em série dos transistores P equivale a $\overline{a} \wedge \overline{b}$. As duas redes são necessárias por causa das características de condução dos transistores: aqueles do tipo P conduzem bem cargas positivas, enquanto que transistores do tipo N conduzem bem cargas negativas.

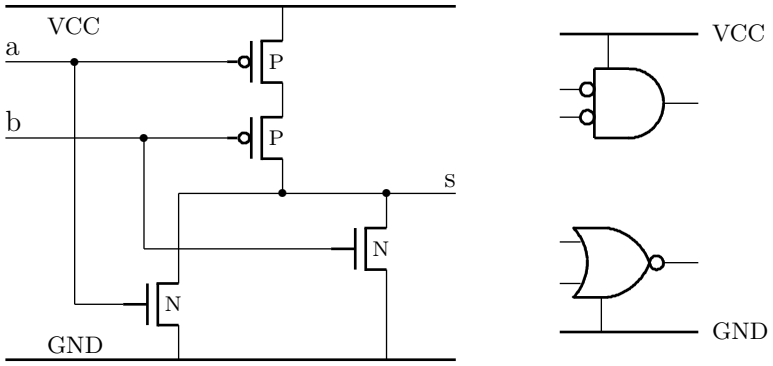


Figura 5.17: Porta *nor* implementada em CMOS.

Porta *nand*

O Mapa de Karnaugh da porta *nand* é mostrado na Figura 5.18. A rede *pull-down* é $a \wedge b$, e é portanto implementada pela ligação em série de dois transistores N. A rede que puxa a saída para cima é *pull-up* $= \overline{\textit{pull-down}} = \overline{b} \vee \overline{a}$, implementada pela ligação em paralelo de dois transistores P. Lembre que as negações em *pull-up* são implícitas ao comportamento dos transistores P.

s	b 0	1
a 0	1	1
1	1	(0)

Figura 5.18: Mapa de Karnaugh para a porta *nand*.

A Figura 5.19 mostra a implementação da porta *nand*. Os circuitos das portas *nand* e *nor* são chamados de *duais* por causa da dualidade nas ligações entre os transistores P e N. Neste caso, *dualidade* significa que uma ligação em série ($\wedge$) é substituída por uma ligação em paralelo ($\vee$), e que uma ligação em paralelo ($\vee$) é substituída por uma ligação em série ($\wedge$). Conforme a Definição 3.12, um circuito Γ é *dual* de um circuito Δ se para todas as ligações em série no circuito Γ existem ligações em paralelo correspondentes no circuito Δ, e se para as ligações em paralelo em Γ existem ligações em série em Δ.

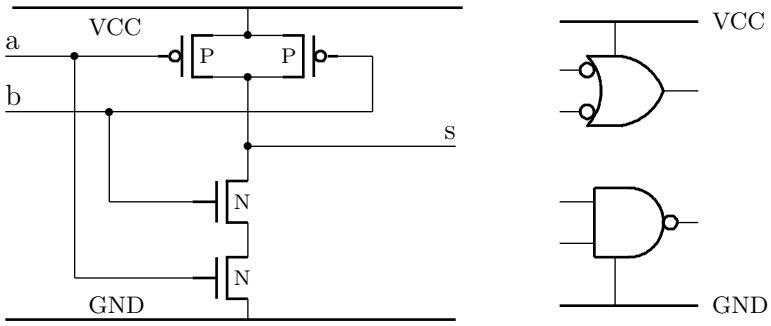


Figura 5.19: Porta *nand* implementada em CMOS.

Aspectos tecnológicos dos circuitos CMOS

A topologia de implementação de circuitos CMOS com redes *pull-up* e *pull-down* implica em que as portas CMOS sejam *sempre* inversoras. Como a rede *pull-down* é composta somente por transistores do tipo N, quando as entradas estão em 1, a saída é puxada para 0. Da mesma forma, quando as entradas da rede *pull-up* são 0, os transistores do tipo P puxam a saída para 1. Isso não chega a ser um problema porque pode-se aplicar DeMorgan para simplificar as inversões, como no circuito da Figura 4.6 (pág. 58).

A implementação de inversores, portas *nand* e portas *nor* emprega circuitos com um número mínimo de transistores. A implementação de portas *and* e *or* em CMOS implica, necessariamente, na ligação de inversores adicionais nas saídas destas portas – ou às suas entradas.

A dualidade das redes *pull-up* e *pull-down* tem um efeito colateral assaz importante: uma vez que as entradas estejam em repouso por tempo suficiente para que todos os nós da rede tenham completado as transições de 1 para 0 e de 0 para 1, uma das duas redes interrompe o fluxo de corrente entre VCC e GND, e portanto o circuito não dissipa energia. Isso significa que circuitos CMOS em repouso não dissipam energia porque não circula corrente através das resistências entre fonte e dreno dos transistores. Na Seção 5.3.3 veremos que esta afirmação não é completamente verdadeira. Se, por um erro de projeto as redes *pull-up* e *pull-down* conduzirem ao mesmo tempo, a corrente entre VCC e GND pode danificar o dispositivo, ou no mínimo, drenar a bateria mais rapidamente do que o desejável.

O comportamento dinâmico dos circuitos CMOS faz com que, durante um breve intervalo de tempo, as duas redes conduzam e portanto o circuito dissipa energia na vizinhança das transições. Considere a transição de 0 para 1 na entrada de um inversor; a transição não é instantânea e a tensão no *gate* dos transistores fica indeterminada enquanto sobe de 0V até VCC, o que faz com que o *pull-up* deixe de conduzir – sua resistência aumenta – ao mesmo tempo em que o *pull-down* começa a conduzir – sua resistência diminui. Durante a breve transição na entrada, os dois transistores conduzem e o circuito dissipa energia. Finda a transição, a dissipação tende rapidamente a zero. Voltaremos a esse assunto na Seção 5.3.3.

5.2.3 Portas Complexas e Células

Um dos grandes atrativos da tecnologia CMOS advém da possibilidade de projetar funções lógicas algo mais complexas do que portas lógicas ao interligar redes com vários transistores. Funções como a soma de dois bits e vem-um, ou o multiplexador de duas entradas, podem ser implementadas com um número de transistores que é menor do que se fossem empregadas portas lógicas individuais.

Estes circuitos são chamados de *células*, e uma vez que uma célula tenha sido otimizada para velocidade e/ou tamanho, ela pode ser replicada em circuitos ainda mais complexos. Por exemplo, um circuito que efetua a soma em 16 bits consiste de 16 réplicas da célula que computa a soma e o vai-um de três bits.

Chamamos de *portas complexas* àqueles circuitos que são algo mais complexos do que uma porta lógica. Tais circuitos são projetados da mesma forma que as portas lógicas: a partir da especificação lógica, derivamos a rede *pull-down*, e então derivamos seu dual, que é a rede *pull-up*. Vejamos alguns exemplos.

Espaço em branco proposital.

Exemplo 5.1 *Considere a função*

$$r = (a \wedge b) \vee (c \wedge d)\,.$$

A rede pull-down é derivada a partir do complemento de r, que são as células com zeros no Mapa de Karnaugh. A Equação 5.4 contém a derivação de pull-down.

$$\begin{array}{lrcll} \textit{pull-down:} & \overline{r} & = & \overline{(a \wedge b) \vee (c \wedge d)} & \text{DeMorgan} \\ & & = & \overline{(a \wedge b)} \wedge \overline{(c \wedge d)} & \text{DeMorgan} \\ & & = & (\overline{a} \vee \overline{b}) \wedge (\overline{c} \vee \overline{d}) & \end{array} \tag{5.4}$$

O circuito que puxa a saída para baixo contém duas ligações em paralelo ($\overline{a} \vee \overline{b}$) e ($\overline{c} \vee \overline{d}$), e estas duas são ligadas em série por causa do $\wedge$, como mostra a parte de baixo da Figura 5.20.

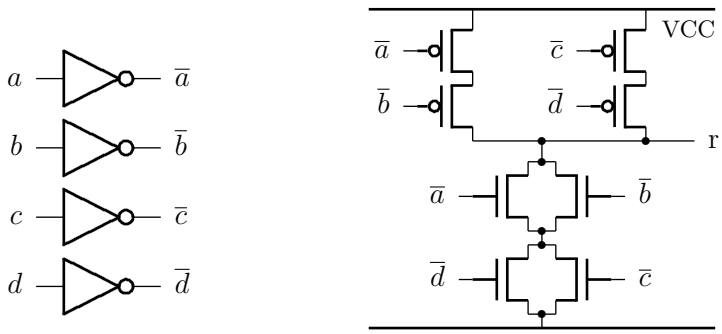


Figura 5.20: Implementação de $r = (a \wedge b) \vee (c \wedge d)$.

O circuito que puxa a saída para cima contém duas ligações em série ($\overline{a} \wedge \overline{b}$) e ($\overline{c} \wedge \overline{d}$), e estas são ligadas em paralelo ($\vee$), como mostra a parte de cima da Figura 5.20. Note que o nível lógico dos sinais nos gates dos transistores tipo P não é diferente daquele dos gates dos transistores tipo N.

A porta complexa que implementa a função r é um multiplexador de duas entradas quando $b = \overline{d}$. Dependendo do circuito à que esta célula será ligada, além dos 8 transistores, podem ser necessários quatro inversores para complementar as entradas. ◁

Exemplo 5.2 *O multiplexador da Figura 5.20 emprega oito transistores, mais os oito transistores para obter os complementos dos sinais das entradas. Se iniciarmos o projeto com o complemento de r*

$$\overline{r} = \overline{(a \wedge b) \vee (c \wedge d)}\,,$$

a rede pull-down usa variáveis de entrada sem complementar, o que resulta num circuito mais compacto. A Equação 5.5 contém a derivação da rede pull-down.

$$\begin{array}{lrcll} \textit{pull-down:} & \overline{\overline{r}} & = & \overline{\overline{(a \wedge b) \vee (c \wedge d)}} & \text{involução} \\ & & = & (a \wedge b) \vee (c \wedge d) & \end{array} \tag{5.5}$$

O circuito que puxa a saída para baixo contém duas ligações série ($a \wedge b$) e ($c \wedge d$), e estas duas são ligadas em paralelo por causa do $\vee$, como mostra a parte de baixo da Figura 5.21.

É necessário um inversor para complementar a saída $\overline{r} = s$, e esta implementação economiza seis transistores, com relação à da Figura 5.20. Por outro lado, as portas lógicas CMOS são naturalmente inversoras e é provável que os complementos dos sinais já estejam disponíveis no circuito. Qual das duas é a solução mais econômica depende do circuito à volta do multiplexador. ◁

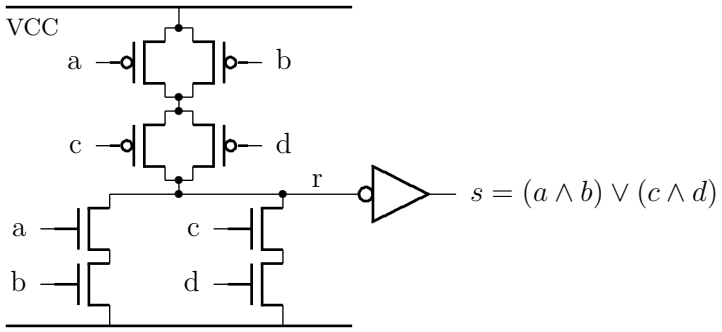


Figura 5.21: Implementação de $s = \overline{r} = (a \wedge b) \vee (c \wedge d)$**.**

São necessários 14 transistores para implementar um multiplexador com portas lógicas, quatro a mais do que no circuito da Figura 5.21. O ganho não está somente no número de transistores; esta célula pode ser implementado de forma mais compacta do que aquela com portas lógicas individuais – circuitos compactos implicam numa melhor utilização da superfície do circuito integrado, e com a redução na área, geralmente se obtém uma redução no tempo de propagação do circuito.

Exemplo 5.3 *Implementemos a função*

$$t = \overline{(a \vee b) \wedge c}\,.$$

As funções pull-down e pull-up estão indicadas na Equação 5.6. A rede que puxa a saída para baixo é composta uma ligação em paralelo $(a \vee b)$*, que é ligada em série com* c $(\wedge c)$*. A rede que puxa a saída para cima é uma ligação em paralelo, na qual um ramo é* $\overline{c}$ *e o outro a ligação em série de* $\overline{a}$ *com* $\overline{b}$ $(\overline{a} \wedge \overline{b})$*. A Figura 5.22 mostra o circuito que implementa* t*.* ◁

$$\begin{array}{lllll}
\textit{pull-down:} & \overline{t} & = & \overline{\overline{(a \vee b) \wedge c}} & \text{involução} \\
 & & = & (a \vee b) \wedge c & \\
\textit{pull-up:} & \overline{\textit{pull-down}} & = & \overline{(a \vee b) \wedge c} & \text{DeMorgan} \\
 & & = & \overline{(a \vee b)} \vee \overline{c} & \text{DeMorgan} \\
 & & = & (\overline{a} \wedge \overline{b}) \vee \overline{c} &
\end{array} \tag{5.6}$$

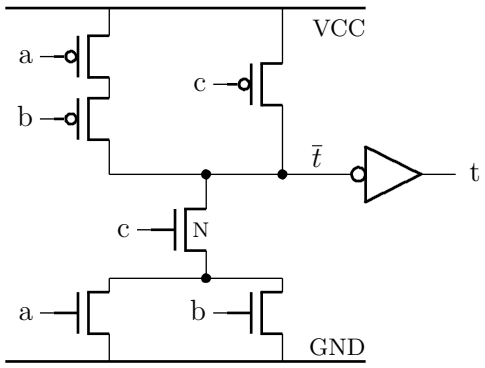


Figura 5.22: Implementação de $t = \overline{(a \vee b) \wedge c}$**.**

Exemplo 5.4 *Implementemos uma função algo mais complexa:*

$$x = \overline{a} \wedge b \wedge c \vee a \wedge \overline{b} \wedge c \vee a \wedge b \wedge \overline{c}$$

A Equação 5.7 mostra a derivação da rede pull-down.

$$\begin{array}{rcll} x & = & (\overline{a} \wedge b \wedge c) \vee (a \wedge \overline{b} \wedge c) \vee (a \wedge b \wedge \overline{c}) & \\ \overline{\overline{x}} & = & \overline{\overline{(\overline{a} \wedge b \wedge c) \vee (a \wedge \overline{b} \wedge c) \vee (a \wedge b \wedge \overline{c})}} & \text{involução} \\ \overline{x} & = & \overline{(\overline{a} \wedge b \wedge c) \vee (a \wedge \overline{b} \wedge c) \vee (a \wedge b \wedge \overline{c})} & \textit{pull-down} \end{array} \tag{5.7}$$

A rede pull-down é a ligação em paralelo de três redes (R_1, R_2, R_3), cada uma destas a ligação em série de três transistores: $R_1 = (\overline{a} \wedge b \wedge c)$, $R_2 = (a \wedge \overline{b} \wedge c)$, e $R_3 = (a \wedge b \wedge \overline{c})$. A rede pull-up é uma série de três redes, cada uma destas a ligação em paralelo de três transistores. Por exemplo, $R'_1 = (\overline{a} \vee b \vee c)$. A Figura 5.23 mostra o circuito completo. ◁

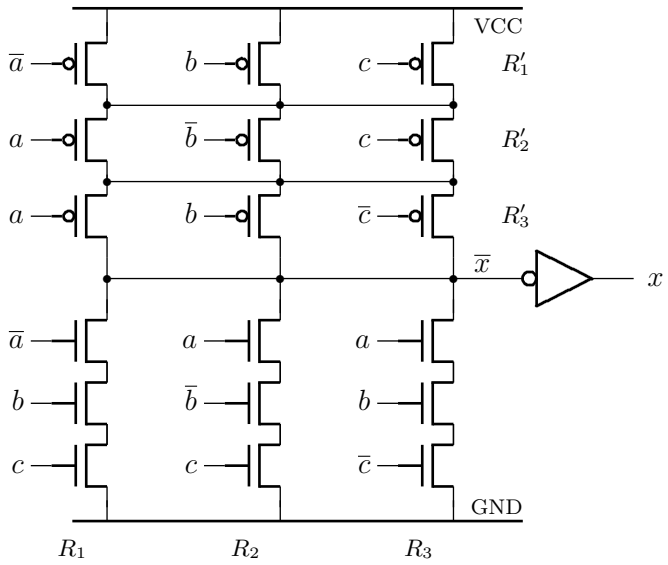


Figura 5.23: Implementação de $x = \overline{a} \wedge b \wedge c \vee a \wedge \overline{b} \wedge c \vee a \wedge b \wedge \overline{c}$**.**

Exercícios

Ex. 5.1 Projete as portas complexas que implementam as seguintes funções:
a) $m = \overline{a \wedge b \wedge c}$;
b) $n = \overline{a \vee b \vee c}$;
c) $p = \overline{a \wedge b \wedge c \wedge d}$;
d) $q = \overline{a \vee b \vee c \vee d}$;
e) $x = a \wedge \overline{b} \vee \overline{a} \wedge b$; $(\oplus)$
f) $e = a \wedge b \vee \overline{a} \wedge \overline{b}$; $(\overline{\oplus})$
g) $y = (a \wedge s) \vee (b \wedge \overline{s}) \vee (a \wedge b)$ (multiplexador bem comportado).

Ex. 5.2 Um *somador parcial* é um circuito combinacional de entradas a e b e saídas s (soma) e v (vai-um), definidas na Equação 5.8. Mostre como implementar um somador parcial em CMOS.

$$s = a \oplus b, \quad v = a \wedge b \qquad \textit{somador parcial} \tag{5.8}$$

Ex. 5.3 Um *somador completo* é um circuito combinacional com entradas a, b e vem (vem-um), e saídas s (soma) e vai (vai-um), que implementa as funções s (soma) e vai (vai-um) definidas na Equação 5.9. Mostre como implementar um somador completo em CMOS. *Pista*: a operação $\oplus$ é associativa.

$$\left.\begin{array}{rcl} s & = & a \oplus b \oplus vem \\ vai & = & a \wedge b \ \vee \ a \wedge vem \ \vee \ b \wedge vem \end{array}\right\} \quad \textit{somador completo} \tag{5.9}$$

Ex. 5.4 Um *somador completo* pode ser implementado com dois somadores parciais e uma porta *or*, como mostra a Figura 5.24. (i) Verifique que este circuito implementa a especificação para o somador completo especificado na Equação 5.9; (ii) implemente esta versão do somador completo em CMOS; e (iii) qual das versões do somador completo emprega menos transistores? *Pista*: não esqueça de que as portas lógicas CMOS são inversoras.

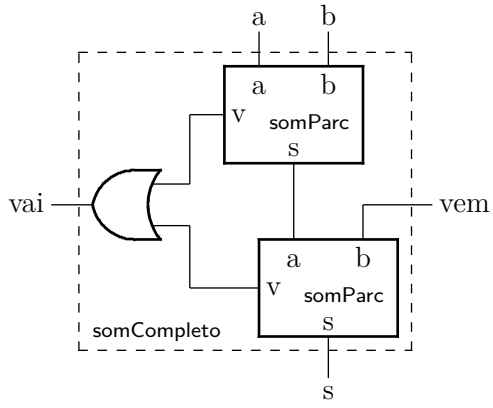


Figura 5.24: Somador completo com dois somadores parciais.

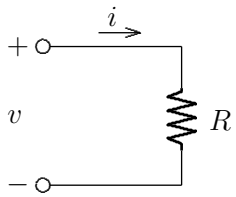
5.3 O Tempo de Propagação É Maior que Zero

Ao empregar bits como um modelo para sinais analógicos, uma característica importante foi escondida pela abstração: os sinais se propagam através dos circuitos em um tempo que não é infinitamente pequeno. Como um mínimo, no vácuo as ondas eletromagnéticas se propagam na velocidade da luz, que é de $c = 0,3 \cdot 10^9$ m/s, ou numa escala mais próxima da prática em sistemas digitais, 30 cm por nanosegundo. Em meios guiados, como em condutores metálicos, os sinais elétricos se propagam com velocidade entre $0,6$ e $0,7c$, que é de aproximadamente "um palmo por nanosegundo". Num sistema com relógio de 3,3 GHz, a mudança no valor de um bit percorre cerca de 7cm durante um ciclo do relógio.

Circuitos CMOS acrescentam atrasos significativos ao tempo de propagação de sinais, e estes atrasos decorrem da construção dos transistores, quando se considera somente os efeitos de primeira ordem, que é o caso deste texto. As Seções 5.3.1 e 5.3.2 contêm uma breve introdução ao comportamento físico dos dispositivos. A Seção 5.4 apresenta um modelo para o tempo de propagação dos circuitos. A Seção 5.3.3 discute a dissipação de energia em circuitos CMOS.

5.3.1 Um Breve Passeio Pelo Reino da Física

Todos os materiais condutores apresentam alguma resistência à passagem da corrente elétrica. A Lei de Ohm nos informa que a corrente i através de um resistor R é proporcional à tensão v aplicada aos seus terminais, como indicado na Equação 5.10, para o diagrama do circuito na Figura 5.25.



$$v = iR \qquad i = v/R \quad (5.10)$$

Figura 5.25: Lei de Ohm, tensão e corrente num resistor.

Duas superfícies metálicas isoladas, que estejam próximas uma da outra, armazenam energia no campo elétrico que existe entre as cargas elétricas opostas nas superfícies. Ao escovar o cabelo com uma escova sintética, os fios de cabelo eriçados se comportam como "placas metálicas" isoladas e carregadas: os fios ficam carregados com os elétrons removidos da escova, e como estes têm carga de mesmo sinal, os fios se repelem deixando o "cabelo em pé".

Um *capacitor* é um dispositivo que armazena energia no campo elétrico que se forma entre duas placas metálicas isoladas. A *capacitância* é medida em Farads e é a razão entre a carga líquida Q numa das placas e a tensão v entre as placas: $C = Q/v$. Se área A das placas é grande, grande é a quantidade de carga líquida para uma dada tensão; se distância D entre as placas é pequena, grande é a força entre as cargas e também a energia armazenada por unidade de carga. Portanto: $C \propto A/D$.

A corrente elétrica é a variação da carga com o tempo: $i = dQ/dt$. A carga líquida numa das placas de um capacitor é, ao longo do tempo, a integral da corrente que 'atravessa' o dispositivo:

$$Q = \int_{-\infty}^{t} i(t)\, dt \; . \tag{5.11}$$

Como as placas são isoladas, não há fluxo de cargas através do dispositivo; o que ocorre é que há ingresso de carga líquida por um terminal, e egresso da mesma quantidade de carga, com polaridade oposta, pelo outro terminal. A corrente 'através' de um capacitor é na verdade o deslocamento das cargas entra suas placas através do circuito externo ao capacitor.

A diferença de potencial, ou tensão, entre as placas de um capacitor é proporcional à corrente ao longo do tempo:

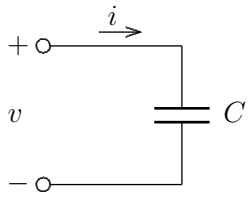
$$v = \frac{1}{C} \int_{-\infty}^{t} i(t)\, dt \;=\; \frac{1}{C}\, Q \; . \tag{5.12}$$

A corrente 'através' do capacitor é proporcional à variação de tensão entre os seus terminais:

$$i = C\, \frac{dv}{dt} \; . \tag{5.13}$$

A corrente "através do capacitor" pode ser medida em um dos seus terminais porque as placas são isoladas. Com a aplicação da diferença de potencial, a carga líquida Q é removida de uma placa e acrescida à outra, através do circuito externo.

A Figura 5.26 mostra o símbolo do capacitor e a relação entre tensão e corrente nos seus terminais, na Equação 5.14.



$$v = \frac{1}{C}\int_{-\infty}^{t} i(t)\,dt \qquad i = C\,\frac{dv}{dt} \quad (5.14)$$

Figura 5.26: Relação entre tensão e corrente num capacitor.

Mais do que os dispositivos em si, nos interessa o comportamento dinâmico de circuitos com resistores (fios) e capacitores (*gates*) de transistores. Em breve retornaremos aos transistores.

O circuito na Figura 5.27 mostra uma fonte de alimentação de tensão V ligada a um resistor R e a um capacitor C através de uma chave de duas posições. A chave esteve na posição α por um tempo longo o bastante para que a tensão no capacitor seja igual à da fonte ($v_C = V$) e portanto não circula corrente no circuito ($i = 0$).

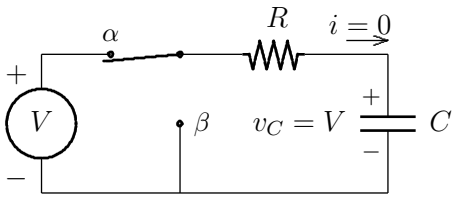


Figura 5.27: Circuito RC em repouso.

No instante t_0, a chave é movida instantaneamente para a posição β e a energia acumulada no capacitor passa a se dissipar no resistor, na forma de calor. Como este circuito é uma malha fechada, a Lei de Kirchoff garante que a soma das tensões ao longo da malha deve ser zero e portanto a tensão nos terminais do capacitor v_C é a mesma que nos terminais do resistor v_R, *viz* $v_R = v_C$, e

$$v_R + v_C = 0 \quad \text{e} \quad R\,i(t) + \frac{1}{C}\int_{t_0}^{\infty} i(t)\,dt = 0\,.$$

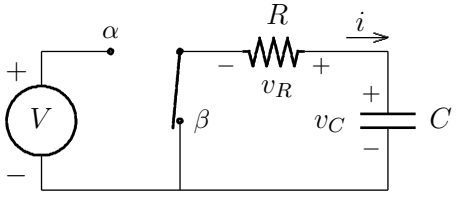


Figura 5.28: Resposta dinâmica do circuito RC.

Derivando-se esta equação com relação ao tempo, para resolver a integral da corrente, se obtém a equação diferencial

$$R\,\frac{di}{dt} + \frac{i(t)}{C} = 0$$

cuja solução, considerando-se que a tensão no capacitor é V_0 em t_0, é dada abaixo, para a

tensão e a corrente nos terminais do capacitor.

$$i_C(t) = \frac{V_0}{R}\, e^{-t/RC} \quad \text{e} \quad v_C(t) = V_0\, e^{-t/RC} \tag{5.15}$$

A Figura 5.29 mostra a curva da tensão nos terminais do capacitor (v_C) ao longo do tempo. Antes de t_0, a tensão no capacitor é a mesma da fonte. Quando a chave muda de posição, esta tensão é aplicada sobre o resistor que passa a dissipar a energia acumulada no capacitor, a uma taxa determinada pelas constantes R e C.

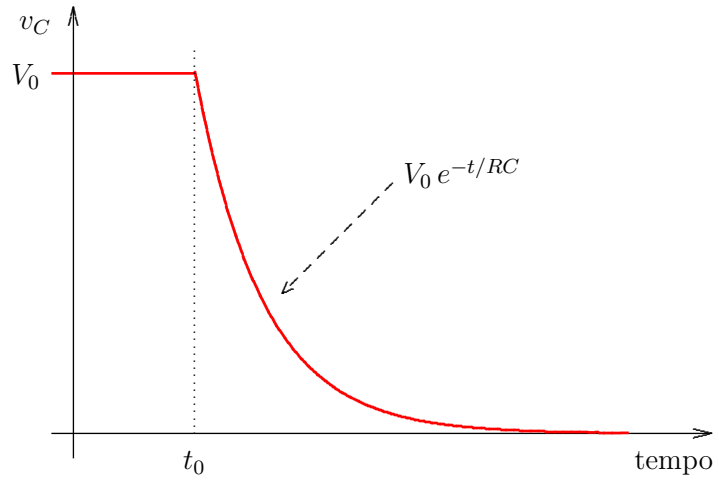


Figura 5.29: Tensão nos terminais do capacitor v_C na descarga do capacitor.

Se o produto RC é muito pequeno ($RC \to 0$), o decaimento da curva é rápido – este seria o caso de um curto-circuito nos terminais do capacitor. Se o produto RC é grande ($RC \to \infty$), então o decaimento da curva é lento – este seria o caso de uma resistência muito alta ligada aos terminais do capacitor. Com um circuito aberto, o que equivale a uma resistência infinita, um capacitor ideal manteria sua carga indefinidamente.

5.3.2 Comportamento Dinâmico de Circuitos CMOS

Considere o transistor tipo N mostrado na Figura 5.30. Para se compreender o comportamento dinâmico dos transistores, o comportamento elétrico da região que compreende *gate*, isolante e substrato pode ser aproximado àquele de um capacitor.

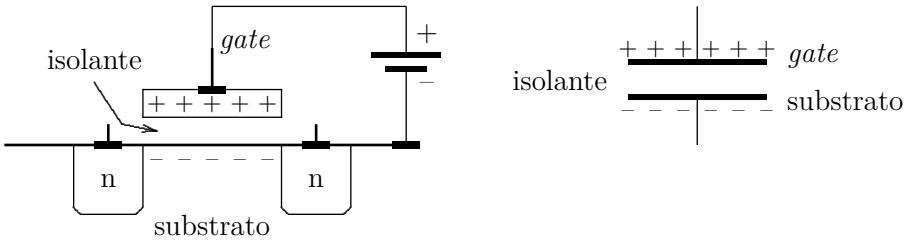


Figura 5.30: Capacitância entre *gate* e substrato.

As placas do capacitor são o *gate* e o substrato, separadas pela camada isolante. Quando há uma diferença de potencial entre as placas, o campo elétrico armazena energia, que não pode ser removida instantaneamente do dispositivo.

Num circuito CMOS típico cada entrada é ligada, no mínimo, a dois transistores, um do tipo N e outro do tipo P, e cada um dos dois *gates* contribui com uma fração da capacitância ligada à saída do circuito que produz o sinal. A Figura 5.31 mostra um circuito com dois inversores, e o circuito equivalente ao inversor I_2, como 'visto' pela saída do inversor I_1. Lembre que não circula corrente da entrada para a saída de circuitos similares ao inversor.

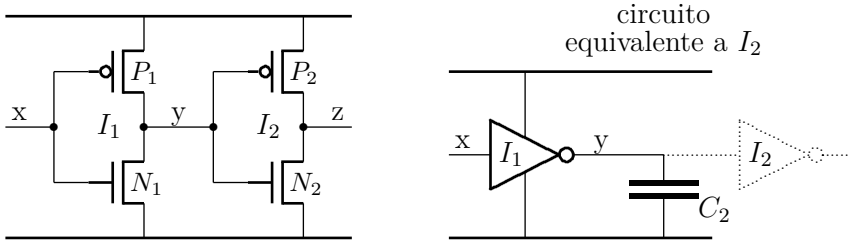


Figura 5.31: Circuito equivalente à entrada de um inversor.

Quando ocorre uma transição na entrada do inversor I_1, o capacitor ligado a sua saída deve ser carregado, com carga líquida final numa das placas ≥ 0, ou descarregado, com carga líquida final $= 0$, dependendo do valor anterior na entrada de I_1. A Figura 5.32 mostra os circuitos equivalentes nas transições de 1 para 0 e de 0 para 1.

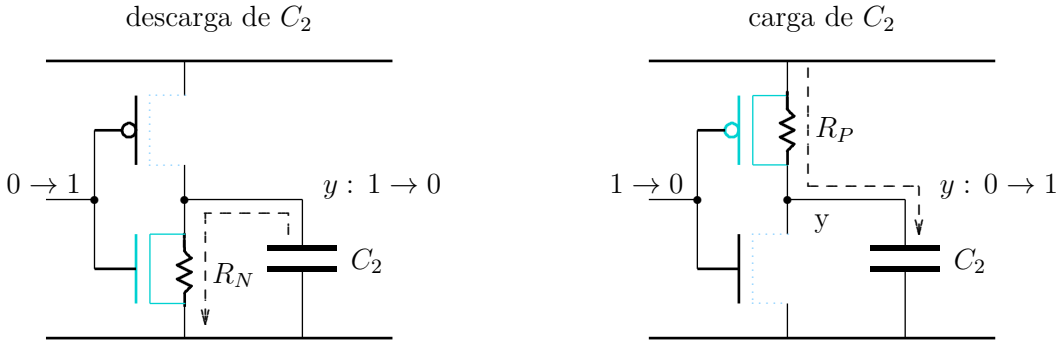


Figura 5.32: Circuitos na carga e na descarga da capacitância de saída C_2.

Antes da transição $0 \rightarrow 1$ na entrada, o capacitor estava carregado ($v_y = VCC$) porque a saída do inversor estava em 1. Logo após a transição, o transistor P fica aberto, e o transistor N conduz com uma resistência R_N. O modelo deste circuito é um resistor R_N em série com o capacitor C_2 e a equação que descreve o comportamento dinâmico da tensão no ponto y é

$$v_y(t) = (1 - e^{-t/R_N C_2})VCC\,. \tag{5.16}$$

A variação com o tempo t da tensão sobre os terminais do capacitor, na descarga, é mostrada no lado esquerdo da Figura 5.33. Na medida em que o tempo passa, a exponencial cresce e a tensão em y cai de VCC até GND. A declividade da curva é determinada pelo produto $R_N C_2$.

Na transição da 1 para 0 na entrada, o capacitor que estava inicialmente descarregado, é carregado através da resistência R_P, e a tensão em y sobe de GND a VCC com

$$v_y(t) = e^{-t/R_P C_2} VCC\,. \tag{5.17}$$

Este comportamento é mostrado no lado direito da Figura 5.33.

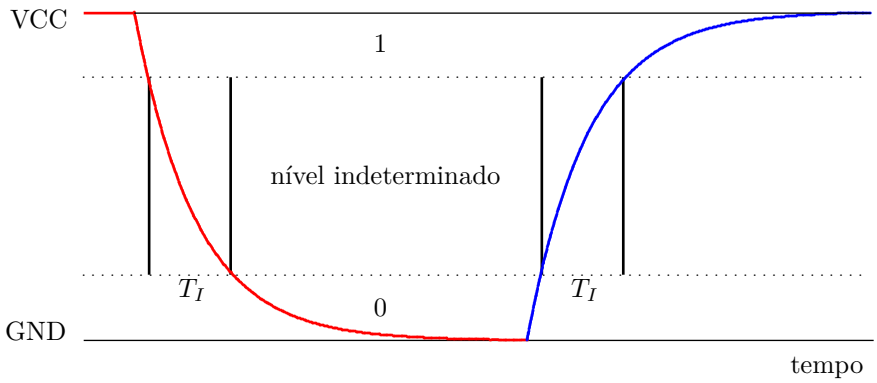


Figura 5.33: Tempo de carga e de descarga da circuito de saída.

A Figura 5.33 está dividida em três faixas horizontais: no topo está a região na qual os sinais são considerados como sendo 1, enquanto que na base está a região na qual os sinais são considerados como 0. Na região entre 0 e 1, os sinais tem um nível lógico indeterminado, que não é 0 nem 1. Exceto em situações específicas em que se deseja manter o sinal em um "terceiro estado", o nível indeterminado pode causar problemas porque a indeterminação tende a se propagar pelo circuito. Um sinal num nível indeterminado pode colocar os dois transistores de um inversor em estado de condução, fazendo com que sua saída também seja indeterminada.

O que se tenta garantir é que os sinais permaneçam indeterminados o menor tempo possível. No diagrama da Figura 5.33, o intervalo em que os sinais estão indeterminados, delimitados pelas linhas verticais, é definido como o *tempo de propagação* do inversor, denotado por T_I.

A resistência através dos transistores, R_P e R_N, não é idêntica e em geral $R_P > R_N$ e portanto a declividade das transições de $1 \rightarrow 0$ e de $0 \rightarrow 1$ é distinta. De acordo com a definição de tempo de propagação, deve-se escolher a mais longa das transições como sendo T_I.

Fan-in* e *fan-out O tempo de propagação depende da resistência entre VCC (ou GND) e a saída da porta – num *nor* de j entradas, são j transistores P em série entre VCC e a saída. O tempo de propagação também depende do número de entradas alimentadas pela saída – se a saída de uma porta lógica alimenta as entradas de três outras portas, a capacitância total é aproximadamente o triplo daquela de uma única porta, porque a capacitância de capacitores ligados em paralelo é a soma de suas capacitâncias.

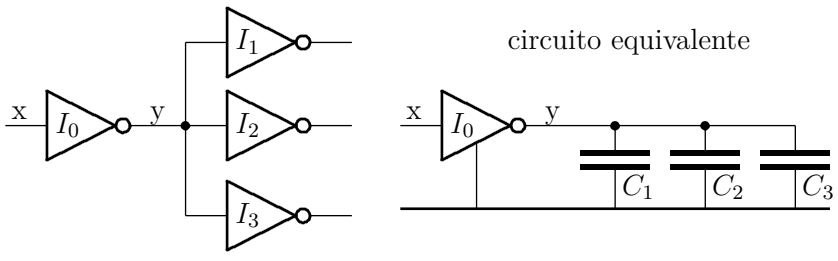


Figura 5.34: *Fan-out* e a carga capacitiva na saída de um inversor.

A quantidade de portas lógicas alimentadas pela saída de uma porta lógica é chamada de *fan-out*. O tempo de propagação de uma porta lógica aumenta com a carga capacitiva na sua

saída, que é proporcional ao seu *fan-out*. A Figura 5.34 mostra um inversor ligado a três outros inversores. O *fan-out* de I_0 é três e a carga capacitiva na sua saída é $C_y = C_1 + C_2 + C_3$.

O número de entradas de uma porta lógica é seu *fan-in*. Um inversor tem *fan-in* de 1, enquanto que uma porta de três entradas tem *fan-in* 3. A Figura 5.35 mostra o circuito equivalente para uma porta *nand* de duas entradas.

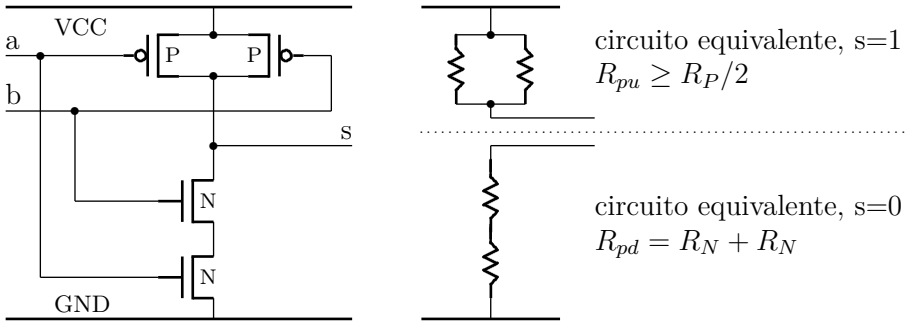


Figura 5.35: *Fan-in* e a carga resistiva na saída de uma porta *nand*.

Quando a saída é 1, a resistência na rede *pull-up*, entre VCC e a carga capacitiva da porta é $R_{pu} = R_P/2$ se as duas entradas estão em 0, ou é $R_{pu} = R_P$ se uma entrada está em 0. Com a saída em 0, as duas entradas estão em 1 e a resistência na rede *pull-down*, entre a carga capacitiva e GND, é $R_{pd} = 2R_N$.

A declividade da exponencial $e^{-t/RC}$, e portanto o tempo de propagação da porta, são afetados pelo *fan-out* ($\propto C$) e pelo *fan-in* ($\propto R$). Na medida em que R e/ou C crescem, mais longo fica o tempo de propagação da porta lógica.

A Figura 5.36 mostra uma porta *nand* de j entradas ligada a k inversores, e os circuitos equivalentes para efeitos de tempo de propagação. Nos interessa o *pior caso* do tempo de propagação, e neste circuito isso ocorre numa transição $1 \rightarrow 0$ na saída do *nand* porque então todas as j entradas estão em 1 e a resistência equivalente, da carga para GND é $R_{eq} = j \times R_N$. A capacitância equivalente, 'vista' desde a saída do *nand* é a soma da capacitância de todas portas ligadas ao fio y, e neste caso é $C_{eq} = k \times C$. Para este circuito, o tempo de propagação é determinado por

$$e^{-t/R_{eq}C_{eq}}. \tag{5.18}$$

Para uma porta *nor*, o tempo de propagação é determinado pelos transistores de tipo P, na rede que puxa a saída para 1.

O número de transistores em série é proporcional ao número de entradas numa porta lógica *nand* ou *nor*. O tempo de propagação de uma porta lógica deve ser determinado, de forma conservadora, pela maior resistência equivalente dentre todas as combinações possíveis de entradas, e por isso, o número de entradas de uma porta lógica raramente ultrapassa 4 ou 5. Funções com mais entradas são implementadas com associações de portas de até 5 entradas.

Temos abstraído o comportamento elétrico dos fios, usando superfícies equipotenciais para interligar os componentes. As ligações físicas entre componentes, se curtas, são através de polisilício, e se longas, através de fios em metal. Por 'fio' entenda-se que as ligações físicas se parecem mais com chapas retangulares longas e estreitas do que com fios cilíndricos. Por causa da geometria e da proximidade com o substrato, estes fios exibem uma capacitância que é distribuída ao longo de todo o seu comprimento. Além disso, as ligações metálicas tem resistência ôhmica que é proporcional ao comprimento e inversamente proporcional à secção

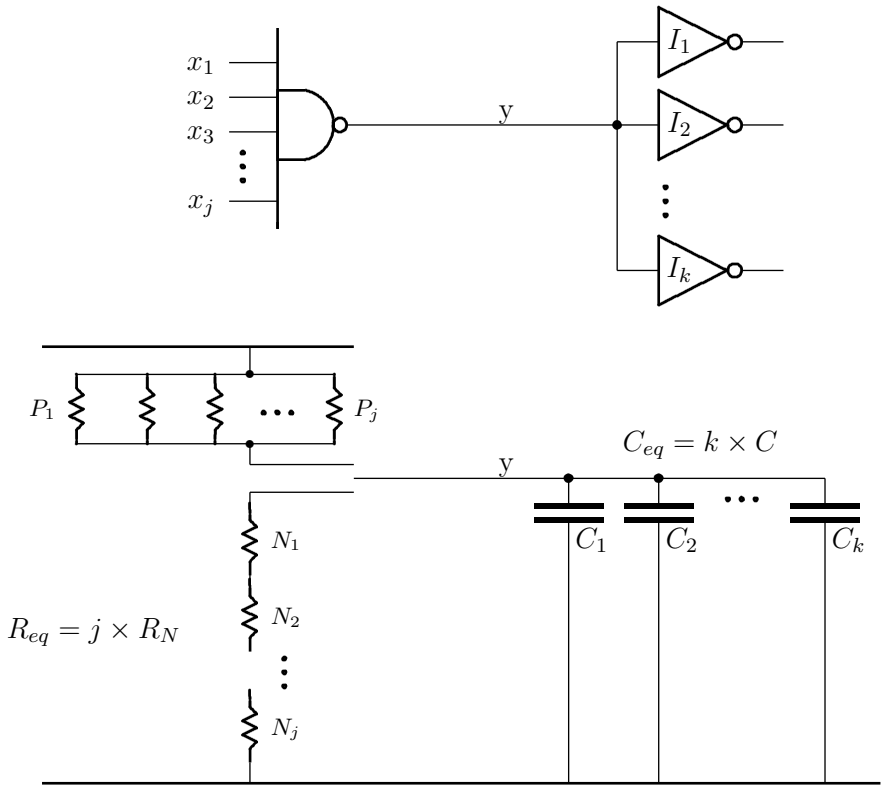


Figura 5.36: *Fan-in* e *fan-out* de uma porta *nand*.

reta transversal do condutor. Tanto a capacitância quanto a resistência dos condutores pioram o tempo de propagação dos sinais, porque elas entram como fatores aditivos no R e no C da exponencial $e^{-t/RC}$. Para minimizar estes efeitos, fios longos são segmentados e inversores são colocados ao longo do caminho para restaurar os níveis lógicos dos sinais, bem como para encurtar o tempo de propagação.

Exemplo 5.5 *Vejamos a influência do fan-out no decodificador de 6 para 64 mostrado da Figura 4.33 (pág. 80), e no Exemplo 4.19. Nos interessa o efeito da carga nas saídas das portas que geram os sinais de endereço do decodificador.*

Usando o modelo simplificado do Exercício 4.7, no qual cada entrada adicional ligada à porta piora o tempo de propagação em 10%*, temos: (i) as linhas de endereço* a_5 *e* a_4 *tem um fan-out de 4 porque cada uma delas alimenta dois inversores e duas portas* and*. Os sinais* x_i *sofrem um atraso de um fator de* $(1,1)^3 = 1,33$*; (ii) as linhas de endereço* a_3 *e* a_2 *tem um fan-out de* 4×4 *porque cada uma delas alimenta quatro* $demux$*-4, e são quatro entradas de seleção em cada* $demux$*-4. Os sinais* y_i *sofrem um atraso de um fator de* $(1,1)^{15} = 4,2$*; e (iii) as linhas de endereço* a_1 *e* a_0 *tem um fan-out de* 4×16 *porque estas alimentam 16* $demux$*-4 e seu fator de atraso é* $(1,1)^{63} = 405$*.*

Um fator de atraso de $4,6$ *não é bom mas pode se aceitável, dependendo da aplicação. Um fator de* 405 *é obviamente inaceitável. A solução empregada nos dois casos é acrescentar inversores nestes caminhos longos para reduzir o fan-out dos sinais. Por exemplo, as linhas* a_3 *e* a_2 *poderiam ser segmentadas em dois e os segmentos ligados por inversores. Nesse caso, seu fan-out cairia para a metade e o fator de atraso se reduziria para* $(1,1)^7 = 1,95$*. As linhas* a_0 *e* a_1 *devem ser segmentadas algumas vezes e possivelmente distribuídas aos seus* $demux$*-4 como uma árvore, com mais de um nível de inversores*

alimentando os demux*-4. Sem usar a distribuição em árvore, o último segmento teria um atraso ainda maior como consequência de todos os inversores intercalados justamente para reduzir o atraso.* ◁

Exemplo 5.6 *Ainda considerando o decodificador de 6 para 64 do Exemplo 4.19, vejamos a influência do fan-in. Consideremos um modelo igualmente simples para a influência do fan-in no qual cada entrada adicional numa porta lógica piora seu tempo de propagação de um fator de* 30% *se comparado com um inversor.*

O primeiro projeto emprega 64 portas and*-6 mais seis inversores. Ignorando-se os efeitos do fan-out nos inversores, o tempo de propagação de cada* and*-6 seria* $(1,3)^5 = 3,7$ *vezes o de um inversor.*

O segundo projeto do decodificador emprega uma árvore de decodificadores 1 para 2 e portanto as portas de duas entradas seriam $(1,3)^1 = 1,3$*, ou* 30% *mais lentas do que o inversor.*

O terceiro projeto emprega uma árvore de decodificadores 2 para 4 e portanto as portas and*-3 de três entradas seriam* $(1,3)^2 = 1,7$ *vezes mais lentas do que o inversor.*

O projeto com portas de largura 6 é uma árvore de um nível com um fan-out elevado nas linhas de endereço porque todas as 6 linhas (e seus complementos) são ligadas a 32 entradas. Uma árvore de distribuição dos endereços é fundamental neste caso.

O terceiro projeto é uma árvore de altura 3 e parece ser um compromisso razoável entre os custos associados à fan-in e fan-out. Uma árvore de distribuição dos endereços para reduzir os atrasos deve ser considerada se este decodificador estiver em algum caminho crítico do circuito.

O que se pode deduzir com estes modelos simplificados, poder-se-ia dizer simplórios, para as influências de fan-in e fan-out, é que o fan-out demanda mais atenção e cuidado na implementação de qualquer circuito que não seja trivial. ◁

5.3.3 Energia e Potência

A *potência* é a taxa na qual a energia é dissipada (ou transferida) num circuito elétrico e sua unidade é o *Watt*, que expressa a taxa de trabalho por unidade de tempo, *viz.* 1 watt equivale a 1 joule por segundo ($W = J/s$).

A energia elétrica é *dissipada* num resistor na forma de calor, tal como na resistência de um chuveiro elétrico. A potência de um chuveiro é um bom indicador da temperatura na qual a água sairá do regador; mantida a temperatura ambiente, quanto maior a potência do chuveiro, mais quente será a água no regador. Para uma dada tensão da rede elétrica (127 ou 220V), quanto menor a resistência, maior a corrente que a atravessa, portanto maior a potência dissipada para aquecer a água.

A potência, ou a energia dissipada, num resistor é o produto da tensão aplicada nos seus terminais e pela corrente que circula entre eles:

$$P_R = v \times i = v \times \frac{Q}{t} = R \times i^2 = \frac{v^2}{R}\,, \tag{5.19}$$

para tensão v, corrente $i = dQ/dt$, carga elétrica Q, resistência R e tempo t.

Quando uma tensão v_F é aplicada às placas de um capacitor, por um tempo suficientemente longo, se diz que o capacitor "está carregado" quando a tensão entre as placas atinge v_F, e a corrente 'através' do capacitor é $i_C = 0$. O campo elétrico entre as placas *armazena* uma

quantidade de energia E_C determinada pela Lei de Joule, que é

$$E_C = \int_{-\infty}^{t} v \times i \, dt = \int_{-\infty}^{t} v \times C \frac{dv}{dt} \, dt \tag{5.20}$$

A Figura 5.32 mostra os circuitos equivalentes de dois inversores, quando as entradas transicionam de 1 para 0 e de 0 para 1. A Figura 5.37 mostra versões simplificadas daqueles circuitos, considerando apenas o resistor equivalente ao transistor do tipo P na carga do capacitor, e o resistor equivalente ao transistor do tipo N na descarga.

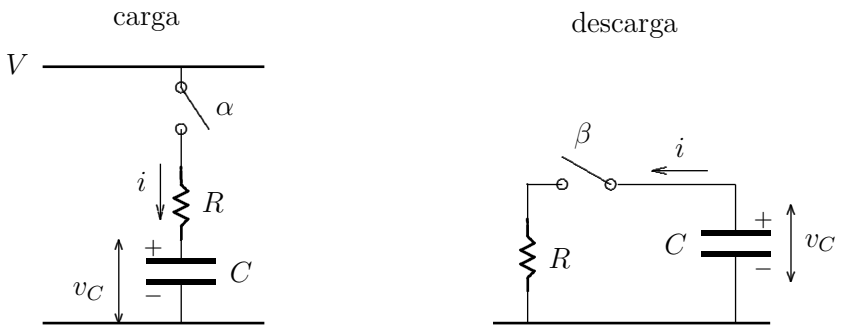


Figura 5.37: Energia acumulada num capacitor.

Supondo que o capacitor está inicialmente descarregado, e que no instante t_0 a chave α é fechada, a tensão $v_C(0)$ é 0 e a corrente através do resistor é $i(0) = V/R$. Decorrido um intervalo suficientemente longo, no instante t_1 a tensão no capacitor se iguala a da fonte de alimentação, $v_C(t_1) = V$, quando então não circula corrente através do circuito e $i(t_1) = 0$.

A energia fornecida pela fonte de alimentação E_F no processo de carga do capacitor é

$$E_F = \int v \times i \, dt = V \int_{-\infty}^{t_1} i \, \frac{dv}{dt} \, dt = VQ = CV^2 \tag{5.21}$$

sendo Q a carga transferida da fonte para o capacitor. Da Equação 5.12 temos que $Q = CV$ e portanto a energia fornecida pela fonte de alimentação é CV^2.

Se considerarmos que a tensão é 0 quando $t = -\infty$ e que a tensão é V no instante t_1, podemos fazer uma troca de variável, reescrevendo a Equação 5.20 como

$$E_C = \int_0^V C \times v \, dv = \frac{1}{2} CV^2 \tag{5.22}$$

A energia dissipada como calor no resistor é $1/2\,CV^2$, pois a fonte forneceu CV^2 joules e o capacitor armazenou $1/2CV^2$ joules.

Com o capacitor inicialmente carregado, $v_C = V$, e a chave β sendo fechada no instante t_2, inicialmente a corrente através do resistor será $i = V/R$, decrescendo exponencialmente até 0. Se a energia inicialmente armazenado no capacitor é $1/2\,CV^2$, então essa mesma quantidade de energia é dissipada no resistor.

Dissipação nas transições Até aqui usamos uma simplificação ao presumir que a resistência no canal de condução dos transistores seja ou zero ou infinita. Considere um transistor do tipo N: quando o transistor conduz, se a tensão no seu *gate* é próxima do valor do nível

lógico 1, a resistência entre fonte e dreno é baixa e o transistor está na *zona de saturação* – a corrente I_{fd} entre os terminais fonte e dreno independe da tensão v_g no *gate*. Se v_g está próxima de 0, então a resistência é elevada –o transistor está na *zona de corte* e a corrente I_{fd} é desprezível.

Durante a transição entre níveis lógicos válidos, a resistência do transistor varia de quase infinita para quase zero, e a corrente I_{fd} é proporcional a v_g. Nesta zona de operação o transistor opera como um amplificador: uma pequena variação na tensão no *gate* provoca uma grande variação na corrente I_{fd}. O efeito da resistência variável se intensifica com transições lentas nas entradas das portas lógicas e por isso os transistores são dimensionados para produzir transições rápidas e assim manter curtos os períodos em modo de amplificação.

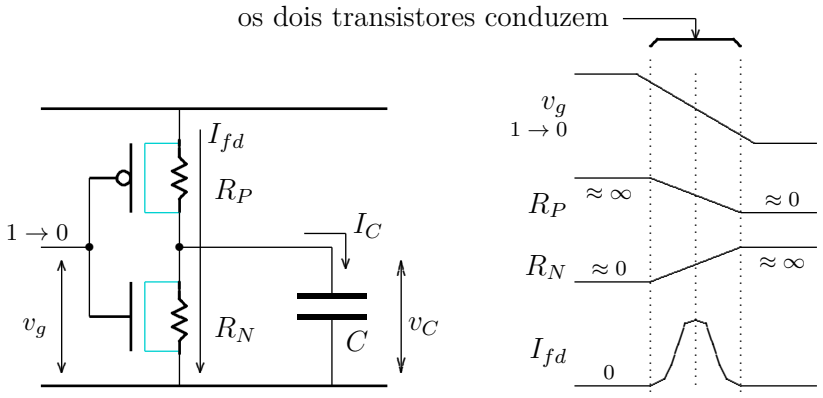


Figura 5.38: Variação dinâmica da corrente num inversor.

A Figura 5.38 mostra um inversor com sua carga capacitiva durante uma transição de 1 para 0 na sua entrada, e quatro curvas: (i) com o comportamento aproximado do sinal na entrada – tensão v_g nos *gates* dos dois transistores; (ii) com a variação na resistência do transistor P, que decresce de um valor muito grande para aproximadamente 0; (iii) com a variação na resistência do transistor N, que cresce de aproximadamente 0 para um valor muito grande; e (iv) a corrente através dos dois transistores. A dissipação de energia $(R_N + R_P)(I_{fd})^2$ ocorre durante esse pico de corrente que circula da fonte para terra.

Potência dinâmica Numa transição $0 \to 1$, um inversor dissipa $CV^2/2$ joules. A mesma quantidade de energia é dissipada numa transição $1 \to 0$. Logo, um ciclo completo na saída do inversor $(1 \to 0 \to 1)$ dissipa CV^2 joules. Se a saída de um inversor é comutada com uma frequência f então a *potência dinâmica* P_{din} dissipada é

$$P_{\text{din}} = f\, C_L\, V^2 \tag{5.23}$$

para uma capacitância de carga C_L e tensão de alimentação V. Note que a potência dissipada é proporcional à capacitância de carga (*fan-out*) e à velocidade de operação, e é proporcional *ao quadrado* da tensão de alimentação. É por isso que a tensão de alimentação foi reduzida de 5V nos anos 1980 para 1V nos anos 2010, a frequência de operação aumentou 1000 vezes, enquanto que a potência dissipada aumentou somente 30 vezes.

Nesta discussão estamos ignorando o pico de corrente (I_{fd}) durante as transições – a energia dissipada nas transições deve ser considerada em adição àquela armazenada nos capacitores, e no caso geral, a dissipação nas transições é uma ordem de magnitude menor do que aquela.

A energia dissipada por um circuito deve ser fornecida pela fonte de alimentação ou bateria – um dos maiores componentes do custo de operação de um circuito é a conta de *energia* elétrica. A energia dissipada para efetuar a computação $\mathcal{K}$ depende da quantidade de transições ($1 \to 0$ e $0 \to 1$) nos sinais do circuito que efetua $\mathcal{K}$; mantidas as demais condições, esta energia independe da frequência de operação – se a computação demorar um segundo ou uma hora, a energia fornecida pela fonte ou bateria é a mesma.

Lembre que potência é a energia dissipada por unidade de tempo: $P = E/s$ e que a frequência f de um evento é o número de ocorrências por unidade de tempo. Considerando que

$$P_{\text{din}} = f\, C_L\, V^2 \tag{5.24}$$

e que C_L e V são fixos, se empregarmos sinais lentos – sinais com frequência baixa, logo f pequeno – para efetuar a computação $\mathcal{K}$, então a potência a ser dissipada é menor, embora tenhamos que esperar mais tempo para obter o resultado desejado. Se empregarmos sinais rápidos, com f grande, então obteremos o resultado em menos tempo mas com um circuito que dissipa mais energia e portanto trabalha mais quente. Assim como num chuveiro, a potência dissipada como calor deve ser removida de alguma forma, por convecção ou resfriamento forçado. Se é necessário resfriamento forçado, então a conta de luz será maior por causa da energia dispendida pelo equipamento adicional.

Potência estática Circuitos CMOS dissipam energia mesmo quando não há transições nas saídas dos circuitos. Por causa das dimensões reduzidas, os isolantes não são perfeitos e portanto em praticamente todos os transistores ocorrem *correntes de fuga*. Para um determinado circuito integrado, a potência estática dissipada é o produto da tensão de alimentação pela corrente suprida pela fonte de alimentação se o circuito estivesse inoperante:

$$P_{\text{est}} = VCC \times I_{\text{fuga}}\,. \tag{5.25}$$

É esperado que a *potência estática* de um circuito integrado seja algo perto de 20 a 25% da sua potência dinâmica. Em circuitos de alto desempenho, com muitos transistores, e transistores pequenos, a potência estática pode chegar a 50% da potência dinâmica. Um processador que dissipa 100W quando trabalha "a plena carga" pode dissipar 50W quando estiver com carga mínima.

5.4 Modelo de Temporização

Circuitos combinacionais são definidos na Seção 4.1 como sendo circuitos (i) com entradas digitais; (ii) com saídas digitais; (iii) suas saídas produzem os valores determinados pela especificação funcional; e (iv) com as entradas estáveis, decorrido o tempo de propagação do circuito, as saídas estabilizam.

Agora que temos uma melhor compreensão do comportamento dinâmico de circuitos CMOS podemos definir a abstração para o *tempo de propagação* – intervalo desde que as entradas ficaram estáveis e válidas até que a saída fique estável – e para o *tempo de contaminação* – intervalo desde que as entradas tornaram-se inválidas até que a saída torne-se também inválida. Com essa abstração podemos projetar circuitos combinacionais e determinar, ou definir, seus parâmetros de comportamento temporal, sem nos preocuparmos com a enorme complexidade da temporização exata dos circuitos.

Antes de definirmos as abstrações, vejamos uma versão ligeiramente idealizada do comportamento real de um circuito com dois inversores. A Figura 5.39 mostra dois inversores ligados em série e o diagrama de tempo de uma transição de 1 para 0 na entrada do inversor I_1. O sinal na entrada de I_1 tem a mesma forma de onda que o sinal na sua saída, embora as constantes RC das exponenciais sejam distintas.

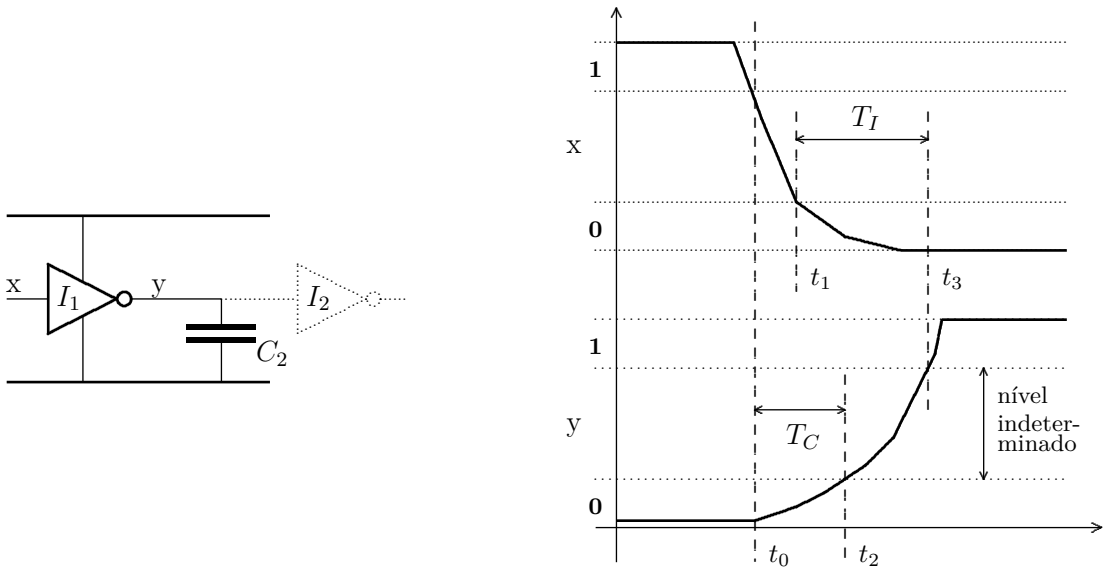


Figura 5.39: Tempo de Propagação (T_I) e Tempo de Contaminação (T_C).

O diagrama mostra, nas linhas pontilhadas horizontais, as faixas de tensão nas quais são reconhecidos os níveis lógicos 0 e 1. Após o início da transição em x, decorre um intervalo $T_C = [t_0, t_2]$ até que y atinja a tensão limite do nível 0. Este intervalo é o *tempo de contaminação* porque a saída y permanece no nível lógico anterior à transição em x – a saída permanece "contaminada pelo passado".

O intervalo desde que a tensão em x atinge o limite do nível lógico 0 até que o sinal y atravessa a faixa de nível indeterminado e ultrapassa o limite do nível lógico 1 é o *tempo de propagação* do inversor $T_I = [t_1, t_3]$. Como vimos na Seção 5.3.2, os intervalos T_I e T_C dependem do número de entradas do circuito, 1 para um inversor, (*fan-in*) e da capacitância na saída, que é C_2 neste caso (*fan-out*).

5.4.1 Tempo de propagação

Agora que entendemos o comportamento dinâmico de circuitos CMOS é possível definirmos precisamente o que entendemos pelo *tempo de propagação* de um circuito combinacional:

> *o tempo de propagação de um circuito combinacional é o* maior *intervalo entre o instante em que todas as entradas ficam estáveis, e o instante em que sua saída fica válida e estável.*

A Figura 5.40 mostra um diagrama com o comportamento temporal de um inversor. A saída y fica estável, no valor determinado pela especificação funcional do inversor, somente depois de decorrido o intervalo T_I, que é o tempo de propagação do inversor. Durante o intervalo T_I,

entre a mudança na entrada e seu reflexo na saída, o nível lógico em y é indeterminado e nada se pode afirmar quanto a y durante este intervalo. A indeterminação é indicada no diagrama pelas regiões hachuradas.

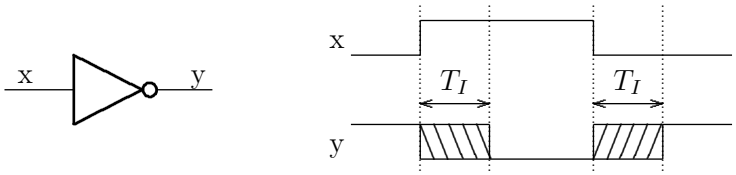


Figura 5.40: Tempo de propagação do inversor.

Lembre que o tempo de propagação de um circuito com vários dispositivos é o máximo dentre o tempo de propagação cumulativo através de todos os caminhos entre as entradas e a saída. A Figura 5.41 mostra a ligação em série de dois inversores e o diagrama de tempos dos sinais nas suas entradas e saídas.

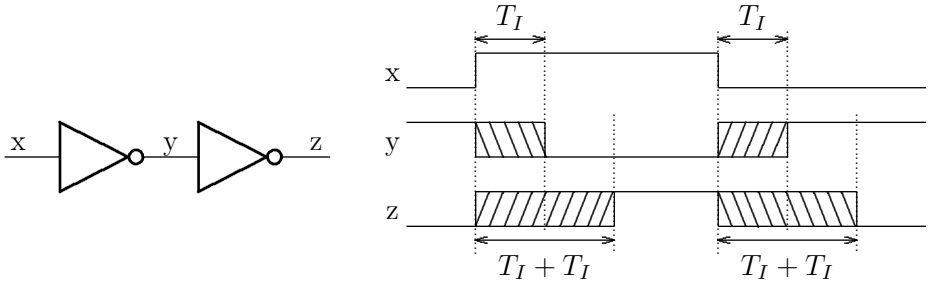


Figura 5.41: Tempo de propagação cumulativo de dois inversores.

O sinal z é uma cópia de x deslocada no tempo, e o deslocamento é de duas vezes T_I, porque o tempo de propagação do segundo inversor somente é contado a partir do instante em que o sinal y fica estável. Assim que a entrada do segundo inversor fica indefinida, sua saída fica indefinida, porque com entradas inválidas, *nada* se pode dizer sobre as saídas de circuitos combinacionais.

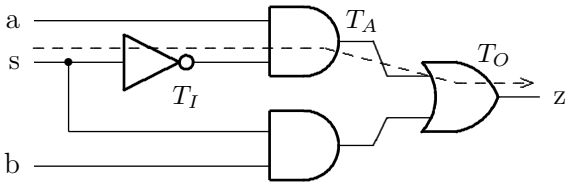


Figura 5.42: Tempo de propagação do multiplexador.

Exemplo 5.7 *O tempo de propagação de um multiplexador de duas entradas é determinado pelo caminho mais longo das entradas para a saída, e este caminho é mostrado pela linha tracejada na Figura 5.42.*

O tempo de propagação do mux-2 *é a soma dos tempos de propagação dos componentes no caminho mais longo, ou no* caminho crítico, *que neste exemplo é dado por* $T_M \geq T_I + T_A + T_O$. ◁

5.4.2 Tempo de contaminação

Além do tempo de propagação, o *tempo de contaminação* pode ser usado para especificar o comportamento temporal de um dispositivo.

> *O tempo de contaminação é o* menor *intervalo no qual uma saída permanece válida após a entrada tornar-se inválida.*

A 'contaminação' é o intervalo que decorre entre a alteração numa entrada até que a saída torne-se inválida, e essa demora resulta da carga e/ou descarga das capacitâncias nos nós internos e externos do circuito. Uma vez que uma saída é contaminada, e atinge um nível indeterminado, todos os sinais a ela ligados devem também ser considerados inválidos.

A Figura 5.43 mostra o diagrama de tempo de um inversor e indica o intervalo de contaminação e o tempo de propagação. Assim que o sinal x inicia a transição de 0 para 1, o sinal cruza a região indeterminada, tornando-se portanto inválido, mas a saída do inversor permanece válida durante T_C.

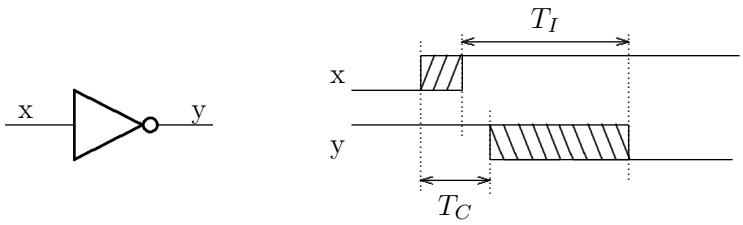


Figura 5.43: Tempo de contaminação T_C e tempo de propagação T_I.

Lembre que o tempo de propagação é o intervalo mais longo entre a entrada válida e a saída também válida, enquanto que o tempo de contaminação é o intervalo mais curto entre a entrada inválida e a saída ainda válida.

Para computar o tempo de propagação de um circuito com vários componentes escolhemos o caminho mais longo, que é aquele que acumula o *maior* atraso das entradas para a saída, porque o tempo de propagação é um limite superior para o intervalo entre entradas válidas e saídas válidas.

Para computar o tempo de contaminação de um circuito escolhemos o caminho com o *menor* valor acumulado, porque o tempo de contaminação é definido como um limite inferior na validade de uma saída com entradas inválidas.

Denotamos o tempo de propagação do circuito X como T_X, enquanto que seu tempo de contaminação é denotado por $T_{C,X}$.

Exemplo 5.8 *Considere que o tempo de contaminação das portas* and *nos circuitos da Figura 4.10 (pág. 62) seja $T_{C,A} = 50\,ps$. O tempo de contaminação da cadeia de portas é 50ps, determinado pelo caminho $x \rightsquigarrow c$, porque este caminho manteria a saída válida por 50ps após as entradas tornarem-se inválidas: $T_{\mathrm{C,cadeia}} \leqslant T_{C,A}$.*

O tempo de contaminação da árvore é de 100ps porque todos os caminhos entre entrada e saída atravessam duas portas lógicas: $T_{\mathrm{C,árvore}} \leqslant 2T_A$. ◁

Exemplo 5.9 *O tempo de contaminação do multiplexador de duas entradas é determinado pelo*

caminho mais curto das entradas para a saída, e estes caminhos são mostrados pelas linhas tracejadas na Figura 5.44. O tempo de contaminação do mux*-2 é dado por* $T_{C,M} \leqslant T_{C,A} + T_{C,O}$*, porque uma alteração nas entradas só se manifesta na saída após a mudança propagar-se através de uma das portas* and *e da porta* or*.* ◁

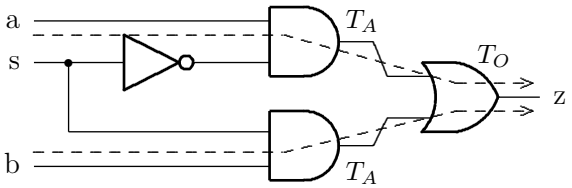


Figura 5.44: Tempo de contaminação do multiplexador.

Exemplo 5.10 *Suponha que o tempo de propagação das portas* and *e* or *sejam de 20ps, e o de um inversor, 15ps. Com estes valores, podemos computar o tempo de propagação de um multiplexador de 8 entradas, construído com sete multiplexadores de duas entradas.*

O tempo de propagação de um mux*-2 é de 55ps, e o do* mux*-8 é de 135ps, e o caminho de propagação mais longo é indicado pela linha pontilhada na figura. Somente o inversor do* mux*-2 selecionado por* s_0 *está no caminho crítico: se as entradas mudam ao mesmo tempo então as influências de* s_1 *e* s_2 *estabilizam* antes *do que a perturbação causada por* s_0*.*

O tempo de contaminação $T_{C,M}$ *é indicado pela linha tracejada e é de 40ps. Os sinais* s_0 *e* s_1 não *estão no caminho de contaminação: se as entradas mudam ao mesmo tempo, então as perturbações causadas por* s_0 *e* s_1 *se manifestam na saída* depois *que as perturbações causadas por* s_2 *no terceiro nível.* ◁

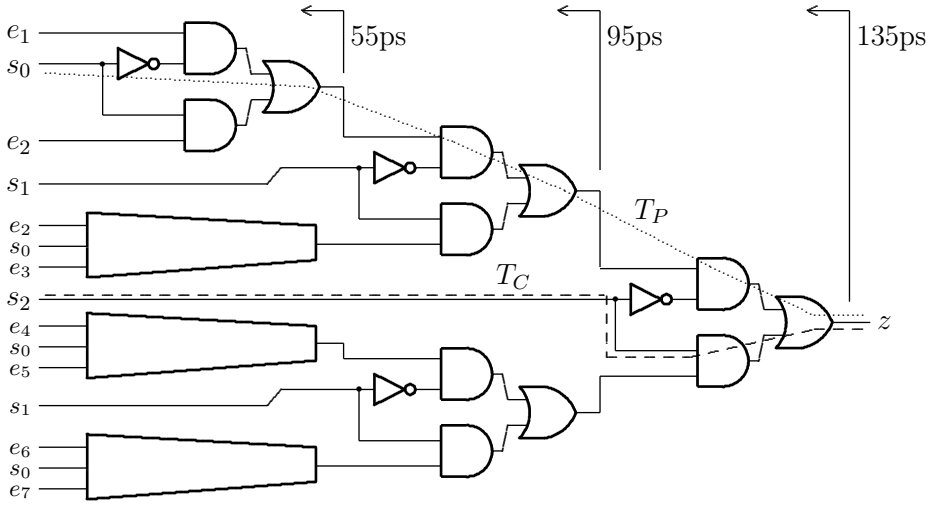


Figura 5.45: Temporização de um multiplexador de 8 entradas.

Exemplo 5.11 *Com base nos dados de temporização do Exemplo 5.10, calculemos o tempo de propagação e de contaminação do* demux*-8 da Figura 4.25. Para a temporização do* demux*-2, considere o circuito da Figura 4.24.*

O caminho mais longo atravessa o inversor do primeiro nível e o and *de cada um dos três níveis, portanto o tempo de propagação é* $T_D \geqslant T_I + 3 \times T_A = 75ps$.

O tempo de contaminação é determinado pelo tempo de contaminação do and *no terceiro nível, que é o caminho mais curto da entrada* s_0 *para as saídas* y_i*:* $T_{C,D} \leqslant T_{C,A}$. ◁

Exemplo 5.12 *Consideremos em mais detalhe o comportamento temporal de portas lógicas CMOS. O circuito da Figura 5.46 contém uma porta* nand*, seguida de um inversor, e a saída do inversor é ligada a duas portas, uma* nand *e uma* nor*. Nos interessa o comportamento dos sinais, ou nós da rede,* a, c *e* d.

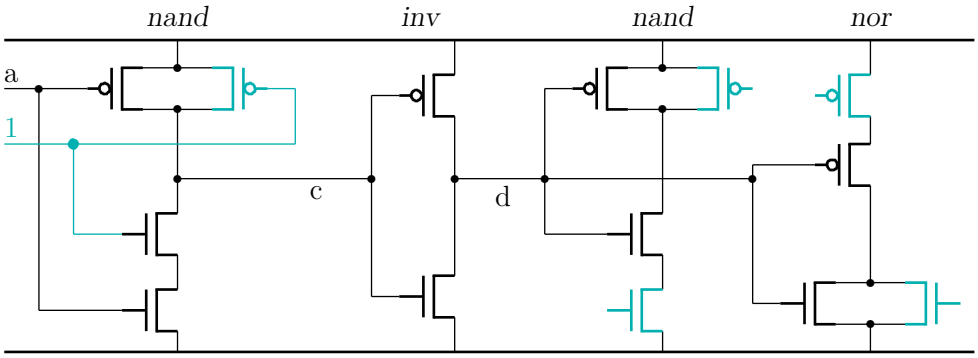


Figura 5.46: Circuito com quatro portas lógicas.

O fan-out *do primeiro* nand *é 1 porque sua saída é ligada somente ao inversor (nó* c*), enquanto que o* fan-out *do inversor é 2, porque sua saída é ligada a duas portas (nó* d*). As transições na saída do inversor são mais lentas porque a capacitância 'vista' pelo inversor é o dobro daquela 'vista' pelo primeiro* nand *– o inversor deve drenar a carga de quatro* gates*, enquanto que o* nand *drena somente a carga de dois* gates*.*

A Figura 5.47 mostra um diagrama de tempos com os sinais a, c *e* d*. A escala de tempo está um tanto exagerada para facilitar a visualização, e as exponenciais* $e^{-t/RC}$ *foram aproximadas por retas. Os intervalos em que os níveis ficam indeterminados estão assinalados pelos retângulos de cor cinza.*

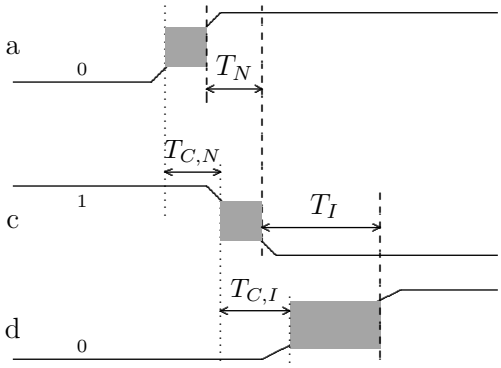


Figura 5.47: Tempo de propagação e de contaminação de quatro portas.

O intervalo de contaminação ($T_{C,N}$) do primeiro nand inicia assim que sua entrada a fica indeterminada, e persiste até que a saída do inversor (c) fique também indeterminada. O intervalo de contaminação do inversor inicia quando o sinal c fica indeterminado, e termina quando o nível em d fica indeterminado. Por causa do fan-out, $T_{C,I} > T_{C,N}$.

O tempo de propagação do primeiro nand (T_N) é contado desde que a fica determinado até que c estabiliza. O tempo de propagação do inversor (T_I) é contado desde que c estabiliza até que d estabilize. Também por causa do fan-out, $T_I > T_N$. ◁

Nos diagramas de tempo mostrados neste texto usamos duas convenções: (i) as bordas de subida e de descida são mostradas como verticais, e (ii) a não ser que mencionado explicitamente, ignoramos os efeitos de *fan-in* e *fan-out* na modelagem de temporização dos circuitos.

O diagrama de tempos mostrado no Exemplo 5.12 pode ser considerado otimista porque as mudanças nas saídas das portas lógicas ocorrem *após* os níveis dos sinais nas entradas ficarem estáveis. Esta é uma escolha arbitrária porque, em circuitos reais, a temporização depende de uma série de fatores, tais como a geometria de cada transistor e de cada fio. As capacitâncias e resistências são determinadas pelas dimensões dos dispositivos, tais como a área dos *gates* dos transistores, o comprimento e espessura dos fios e a espessura dos isolantes entre as partes condutoras dos componentes e o substrato.

As simulações detalhadas das correntes e tensões nos circuitos tomam em conta todos estes fatores, e obviamente têm custo computacional elevado porque os modelos são complexos e o número de componentes em circuitos relevantes varia de milhares a milhões. A análise dos resultados também é custosa porque envolve grandezas contínuas – tensões e correntes – que devem ser discretizadas para que uma interpretação 'digital' seja possível, e para verificar se o circuito atende à sua especificação funcional e de temporização.

5.4.3 Comportamento transitório

O tempo de contaminação é uma espécie de memória de curto prazo e este comportamento será explorado adiante. O tempo de contaminação ajuda a explicar certos comportamentos inesperados. Nos circuitos mostrados na Figura 5.48, quando se considera seu comportamento estático, o sinal c é *sempre* 0, enquanto que o sinal r é *sempre* 1, se todos os dispositivos se comportam de acordo com nossa definição para circuitos combinacionais.

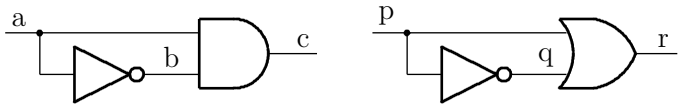


Figura 5.48: Circuitos com comportamento estático bem definido.

Contudo, se considerarmos o comportamento dinâmico destes circuitos, as afirmações da frase anterior são falsas durante um intervalo relativamente curto após as transições nas entradas. Vejamos o circuito com a porta *and*. Para simplificar, suponha que o tempo de contaminação da porta *and* e do inversor são idênticos. A Figura 5.49 mostra um diagrama de tempo que considera somente os efeitos do tempo de contaminação, e que $T_I = T_A = 0$.

No intervalo α, a entrada b é zero e a saída é aquela esperada. No intervalo β, o atraso de contaminação mantém a saída do inversor válida após a alteração na sua entrada, as duas entradas da porta *and* são 0 e sua saída permanece em 0. No intervalo γ a saída do inversor

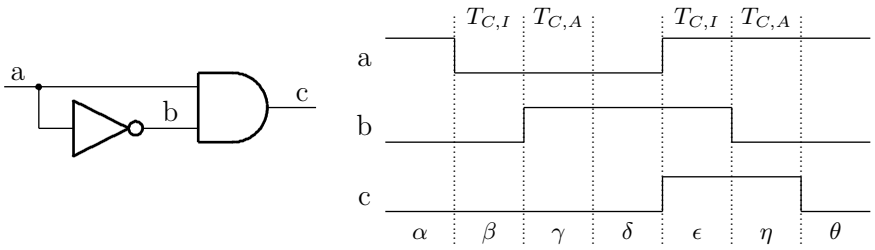


Figura 5.49: Circuito com comportamento dinâmico distinto do estático.

muda para 1 e c permanece em 0. No intervalo δ os sinais ficam estáveis. No intervalo ϵ o atraso de contaminação mantém a saída do inversor em 1, enquanto a entrada a muda para 1 – com as duas entradas da porta *and* em 1, sua a saída muda para 1 porque estamos considerando que $T_I = T_A = 0$. No intervalo η, a contaminação da porta *and* mantém sua saída em 1 durante $T_{C,A}$. No intervalo θ, a saída do *and* reflete sua entrada b em 0. O pulso espúrio em c dura $T_{C,I} + T_{C,A}$.

Exemplo 5.13 *O autor recomenda enfaticamente ao leitor que desenhe, sem demora, um diagrama de tempo similar ao da Figura 5.49 para o circuito com a porta or.* ◁

A definição de circuito combinacional que adotamos considera o seu comportamento correto *após o decurso do seu tempo de propagação*, caso sua especificação funcional seja satisfeita pelas saídas. Isso significa que o comportamento descrito no diagrama de tempos da Figura 5.49 é uma ocorrência observável e indesejada, mas quando se considera o tempo de propagação das portas lógicas, aquele circuito se enquadra na definição de circuito combinacional.

A Figura 5.50 expande o diagrama da Figura 5.49 com a inclusão do tempo de propagação, que é o intervalo mais longo necessário para a saída estabilizar, após as entradas estabilizarem. Os efeitos da contaminação estão indicados por traços espessos, e o tempo de propagação por regiões hachuradas.

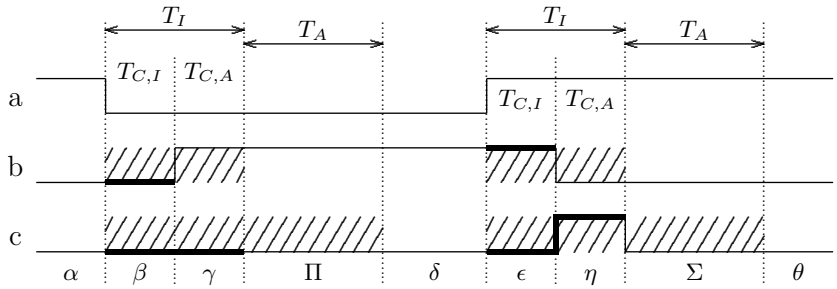


Figura 5.50: Comportamento dinâmico considerando o tempo de propagação.

Nos intervalos β e γ que se estendem por T_I segundos, a saída c é inválida porque a entrada b do *and* é inválida. No intervalo Π, b fica válido mas a saída c é inválida durante o intervalo T_A, que é o tempo de propagação da porta *and*. No intervalo δ a saída é aquela determinada pelas especificações do inversor e da porta *and*. Nos intervalos ϵ e η a saída c é inválida porque b é inválido, em que pese a ocorrência do pulso em c durante η. Decorrido o segundo T_A (Σ), a saída fica válida e correta no intervalo θ.

O sinal b sofre contaminação por causa de a nos intervalos β e ϵ. O sinal c sofre contaminação em β por causa de b e em γ por causa de a e de b. A contaminação em c, no intervalo ϵ decorre do nível em a, e no intervalo η decorre do nível (contaminado) em b.

Exemplo 5.14 *Considere o circuito da Figura 5.51, que detecta tanto as bordas de subida quanto as bordas de descida na sua entrada. A cada borda no sinal e, o xor produz um pulso em sua saída porque suas entradas estão distintas. Desenhe um diagrama de tempos como o da Figura 5.49 – somente considerando o tempo de contaminação – para certificar-se de que o comportamento é o esperado.* ◁

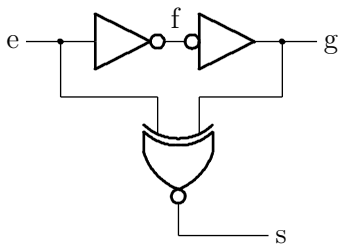


Figura 5.51: Detector de bordas com porta *xor*.

Exemplo 5.15 *Vejamos uma aplicação para o comportamento dinâmico do circuito com a porta or e dois inversores ou um buffer, como o mostrado na Figura 5.52. Para simplificar a discussão, suponha que o tempo de propagação da porta or é zero.*

Considere um pulso de curta duração, ou um espículo (glitch), de $1 \rightsquigarrow 0 \rightsquigarrow 1$, que dura uma fração do tempo de propagação do buffer T_B. O buffer atrasa o espículo no sinal q o suficiente para que a entrada p do or estabilize, de tal forma que a saída r não apresenta o espículo, e assim o pulso é 'filtrado' pelo circuito. Isso é o que mostra o lado esquerdo do diagrama de tempos. Contudo, o lado direito do diagrama apresenta dois *espículos, um reproduzido a partir do sinal original e um segundo, que é a versão atrasada pelo buffer. Melhora um lado, piora o outro.* ◁

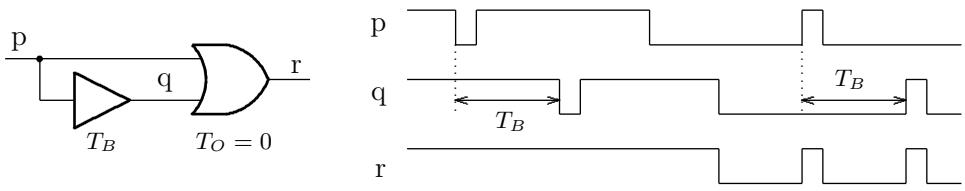


Figura 5.52: Filtro de espículos com porta *or*.

5.4.4 Implicativos Primários, Implicativos Essenciais e Temporização

O multiplexador mostrado na Figura 5.53 exibe um comportamento similar ao circuito com a porta *or* da Figura 5.48. Quando as entradas a e b estão ambas em 1, os sinais m e n refletem, a menos do tempo de propagação das portas *and*, os valores de $\overline{s}$ e s. Dependendo de como este multiplexador é empregado, o comportamento do sistema pode ser errático por causa do pulso espúrio causado pela 'corrida' entre os sinais derivados de s nas entradas da porta *or* – a versão atrasada de s em n contra a versão atrasada de $\overline{s}$ em m. A detecção destas corridas é problemática porque, durante a depuração, pode ser difícil de reproduzir as condições que provocam o pulso espúrio na saída do multiplexador.

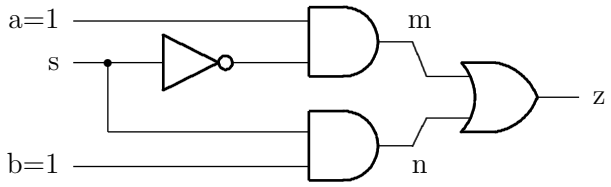


Figura 5.53: Multiplexador com pulso espúrio na saída.

A Figura 5.54 mostra o Mapa de Karnaugh para o multiplexador de duas entradas. O pulso espúrio na saída do multiplexador ocorre quando as entradas passam da configuração correspondente à célula 7 ($\langle$s,a,b$\rangle = 111$) para a configuração da célula 3 ($\langle$s,a,b$\rangle = 011$). Se um agrupamento adicional for incluído no mapa, cobrindo as células 3 e 7, o pulso indesejado não ocorrerá. Este agrupamento corresponde a $a \wedge b$ e mantém a saída em 1, independentemente do que ocorra com s. O circuito é mais complexo, com a adição de uma porta *and* e de uma entrada adicional no *or*, mas seu comportamento dinâmico é exatamente o desejado, e independe dos efeitos transitórios causados pelo tempo de contaminação.

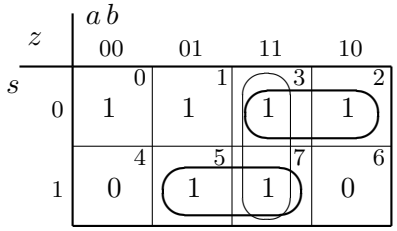


Figura 5.54: Mapa de Karnaugh do multiplexador bem comportado.

A saída do multiplexador de duas entradas pode ser descrita em termos de uma soma de produtos com os termos

$$z(s,a,b) = (\overline{s} \wedge \overline{b} \wedge a) \vee (\overline{s} \wedge b \wedge a) \vee (s \wedge b \wedge \overline{a}) \vee (s \wedge b \wedge a) \tag{5.26}$$

Do ponto de vista lógico, cada um dos termos da soma implica $z = 1$, embora somente a disjunção dos quatro seja equivalente a z. Assim,

$$(\overline{s} \wedge \overline{b} \wedge a) \Rightarrow (z = 1)$$

O mesmo vale para a versão simplificada, como a do mapa da Figura 5.54:

$$(\overline{s} \wedge a) \Rightarrow (z = 1)$$

Os termos individuais da soma de produtos são chamados de *implicativos* porque, quando as variáveis fazem o termo igual a 1, esta mesma combinação de variáveis implica em que o resultado da soma de produtos seja 1.

Todos os quatro termos da Equação 5.26 são termos implicativos. Os três termos simplificados obtidos do Mapa de Karnaugh $\langle\, (\overline{s} \wedge a), (s \wedge b), (a \wedge b) \,\rangle$ são chamados de *implicativos primários*, porque não podem ser eliminados por simplificação. Um termo da soma de produtos que não pode ser agrupado a nenhum outro, sendo portanto imprescindível para representar uma função é chamado de *implicativo essencial*. Para uma função com um certo número de mintermos, pode haver mais de um conjunto de implicativos primários que simplificam aquela função.

O mapa da Figura 5.55 mostra três implicativos essenciais que não podem ser eliminados por simplificação: um que cobre o mintermo 0, um que cobre os mintermos 3 e 7, e o que cobre os mintermos 12, 13, 14 e 15. O implicativo primário – mostrado com o traço fino – que cobre os mintermos 7 e 15 não é essencial. Este implicativo primário deve ser implementado caso a aplicação do circuito especifique um comportamento dinâmico sem corridas, que neste exemplo são transições $1 \rightsquigarrow 0$ em x. No caso geral, a saída de um circuito implementado com todos os implicativos primários é livre dos pulsos espúrios provocados por corridas.

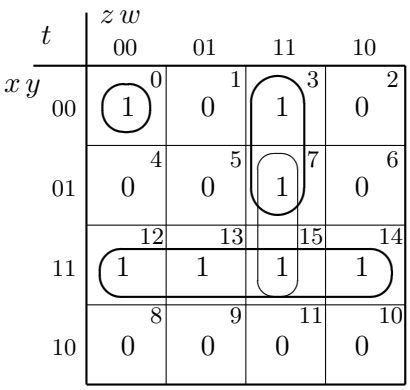


Figura 5.55: Função de quatro variáveis e seus implicativos.

5.4.5 Lógica Restauradora

O comportamento dinâmico destes circuitos é distinto daqueles previstos pela nossa abstração para dispositivos combinacionais. Nosso modelo abstrai particularidades dos circuitos CMOS que se manifestam em circuitos reais. Circuitos CMOS são projetados para 'melhorar' os sinais em suas saídas, com relação aos sinais nas suas entradas. Os níveis lógicos nas saídas das portas lógicas são "mais fortes" do que aqueles presentes em suas entradas porque as redes *pull-up* e *pull-down* puxam as saídas para VCC e para GND, que são 1 e 0 fortes. Por causa deste comportamento os circuitos CMOS são chamados de *restauradores* (*restoring logic*) porque entradas fracas resultam em saídas fortes. Veja a Figura 5.33: assim que um sinal de entrada sai da região indeterminada, a saída é puxada para 1 ou para 0, porque os transistores funcionam como amplificadores: um nível elétrico fraco numa entrada é amplificado até os níveis das fontes de 0 ou de 1.

Exercícios

Ex. 5.5 Desenhe um diagrama de tempos similar ao do Exemplo 5.15 para um filtro com uma porta *and*.

Ex. 5.6 Ainda com referência ao Exemplo 5.15, desenhe outro diagrama de tempos para um filtro que combina uma porta *or* e uma porta *and*.

Ex. 5.7 Qual é a largura máxima dos pulsos que podem ser filtrados pelos circuitos do Exemplo 5.15 e dos Exercícios 5.5 e 5.6?

Ex. 5.8 Mostre como implementar portas lógicas *and*, *or*, *xor*, *nand* e *nor* de 16 entradas empregando somente portas *nand* e *nor* de 4 ou menos entradas e inversores. Sua implementação deve minimizar o tempo de propagação. *Pistas*: (i) quais funções são associativas? (ii) árvores tem altura $\propto \log(N)$ enquanto que vetores tem comprimento $\propto N$.

5.5 ROM – *Read-Only Memory*

Memórias somente de leitura (*Read-Only Memory*, ROM) são memórias que podem ser lidas mas não podem ser alteradas enquanto o circuito está em operação normal, ao contrário das memórias de leitura e escrita (*Random Access Memory*, RAM) – cujos conteúdos podem ser alterados durante a operação normal do circuito.

Qualquer função $F : \mathbb{B}^n \mapsto \mathbb{B}$ pode ser representada como uma soma de produtos

$$F(e_0, e_1, \ldots e_n) = \bigvee [\bigwedge (e_0, e_1, \ldots e_N)\,] \,.$$

Uma função de n variáveis pode ser implementada com um multiplexador de $N = 2^n$ entradas. Cada uma das N entradas é qualificada pela conjunção de n literais – a k-ésima entrada é qualificada pela conjunção dos literais que representam o número k. A Figura 5.56 mostra a implementação da função F com um multiplexador de n entradas. Cada porta *and* seleciona, e qualifica, um dos mintermos da soma de produtos.

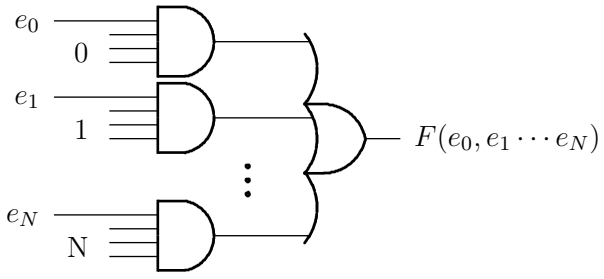


Figura 5.56: Função implementada com um multiplexador de *n* entradas.

Um circuito com a mesma organização de um multiplexador é usado para implementar uma memória ROM. Um *decodificador de linha*, similar ao conjunto de portas *and* do multiplexador, é usado para selecionar uma das linhas da matriz. Uma porta *or* é utilizada para 'coletar' os bits em 1 em cada coluna da matriz. A próxima seção mostra como implementar um decodificador de linha.

5.5.1 Decodificador de Linha

Vejamos como uma porta *or* com várias entradas pode ser implementada usando somente transistores do tipo N. Neste circuito violaremos uma das regras de projeto e usaremos um transistor do tipo N ligado à VCC – como estes transistores conduzem mal o nível lógico 1, dizemos que o sinal do dreno deste transistor é um *sinal fraco*. Quando um sinal fraco é ligado a um sinal forte, o sinal forte sempre prevalece e se sobrepõe ao nível do sinal fraco.

O circuito da Figura 5.57 mostra uma porta *or* com 4 entradas. O transistor no topo do circuito é chamado de *pull-up* porque sua função é puxar, fracamente, o sinal r para o nível

lógico 1. Se uma ou mais das entradas é 1, então o sinal r é puxado para 0 pelo(s) transistor(es) tipo N e a saída s é 1. Se todas as entradas forem 0, então o *pull-up* puxa r para 1 e portanto a saída s fica em 0.

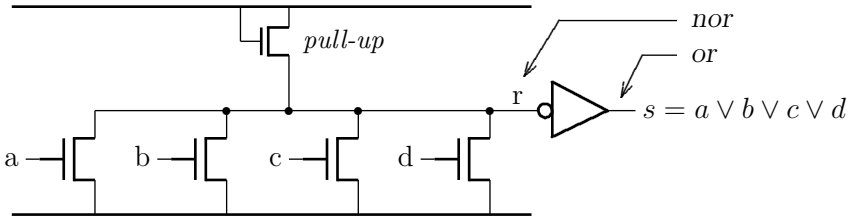


Figura 5.57: Função *or* de quatro entradas.

Necessitamos de uma porta *and* para implementar um decodificador de linhas. Se a memória é uma matriz com L linhas, então o decodificador necessita $\ell = \log_2(L)$ bits de endereço, e cada uma das linhas é selecionada pela conjunção dos bits apropriados do endereço. O circuito da Figura 5.58 mostra a função *and* implementada com a porta *nor*. O circuito dentro das linhas pontilhadas é a porta *nor* da Figura 5.57.

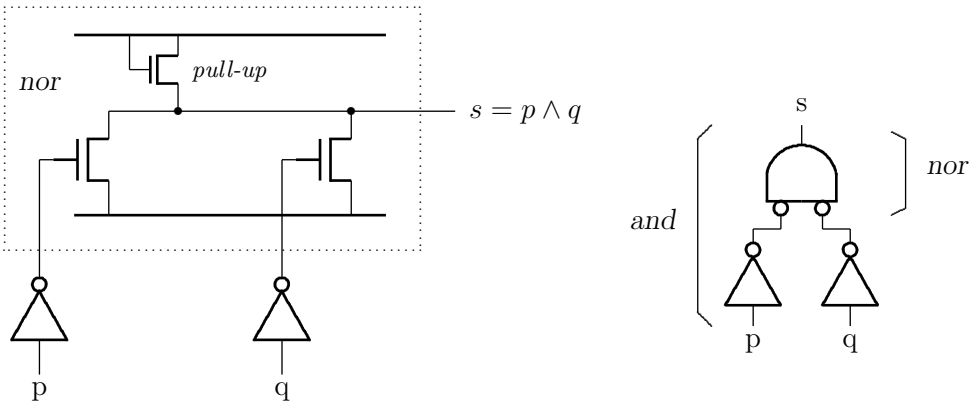


Figura 5.58: Função *and* de duas entradas.

Um decodificador de 2 bits de endereço $\langle a_1, a_0 \rangle$ para 4 linhas $\langle l_3, l_2, l_1, l_0 \rangle$ é mostrado na Figura 5.59. Nesta figura VCC é indicado como uma barra horizontal, e GND como um triângulo achatado. Cada linha do circuito corresponde a uma porta *and* de duas entradas.

A caixa pontilhada mostra a representação para diagramas de bloco. Este decodificador é uma coleção de *and*s e por isso é chamado de *and array*. Cada uma das L linhas da matriz corresponde a um mintermo e qualquer função de até ℓ variáveis pode ser implementada pela disjunção dos mintermos apropriados, como veremos no que se segue.

Espaço em branco proposital.

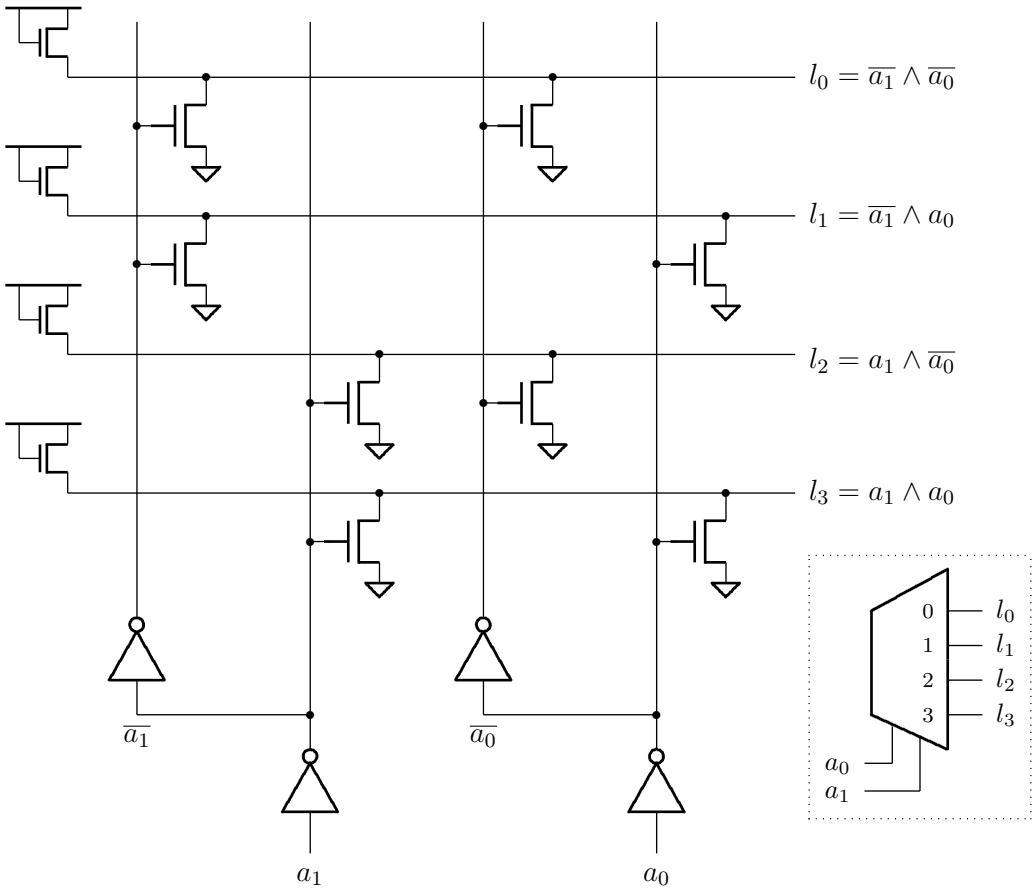


Figura 5.59: Decodificador de 2 bits para 4 linhas.

A Figura 5.60 mostra o decodificador com as entradas $\langle a_1, a_0 \rangle = 2$, selecionando a saída 2. Os transistores que não conduzem são mostrados com o canal como um linha pontilhada, e os que conduzem são mostrados com linhas contínuas. Como os dois transistores da linha l_2 estão com seus *gates* em 0, e portanto não conduzem, o *pull-up* desta linha 'puxa' l_2 para 1. Nas outras linhas há pelo menos um transistor puxando o sinal no fio horizontal para 0, como indicam as setas tracejadas.

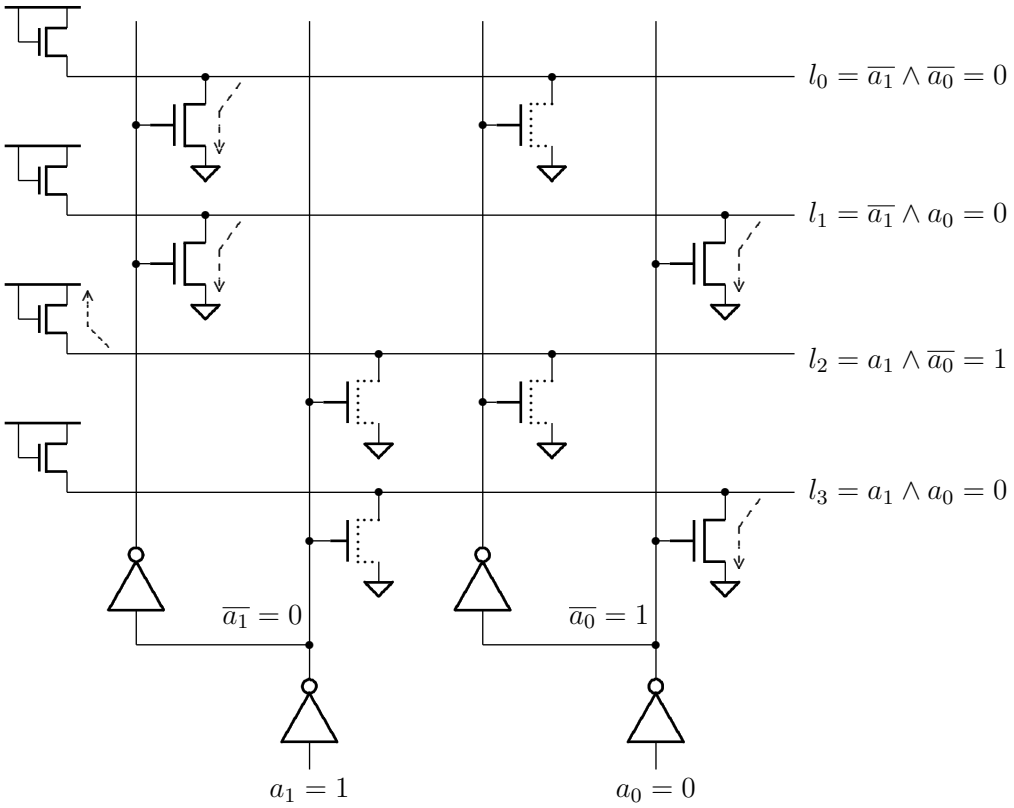


Figura 5.60: Seleção da saída 2 no *and-array* (*decod-4*).

5.5.2 Matriz de Dados da ROM

Uma memória ROM é construída como uma matriz de L linhas por C colunas, e em cada intersecção há um transistor para indicar que o bit correspondente à posição (i, j) é 1; a ausência de um transistor na intersecção (m, n) indica que aquele bit é 0.

Uma função *or* similar à da Figura 5.57 pode ser usada para implementar a disjunção dos mintermos gerados pelo decodificador. Com várias disjunções obtemos um *or array*, que combinado com um *and array*, resulta numa coleção de somas de produtos, que é justamente a matriz de memória M que estamos a construir. A Figura 5.61 mostra uma memória com $n + 1$ bits de endereço e $w + 1$ bits de dados tais que o valor de cada um dos bits de dados r_k resulta da disjunção de um certo número de mintermos

$$r_k = \bigvee l_j,\ j \in [0, m],\ k \in [0, w] \tag{5.27}$$

e cada mintermo l_j é uma conjunção de endereços ou de seus complementos

$$l_j = \bigwedge \dot{a}_i,\ i \in [0, n]\,. \tag{5.28}$$

Na Equação 5.28, $\dot{a}_i$ representa um literal que pode ser a_i ou $\overline{a_i}$.

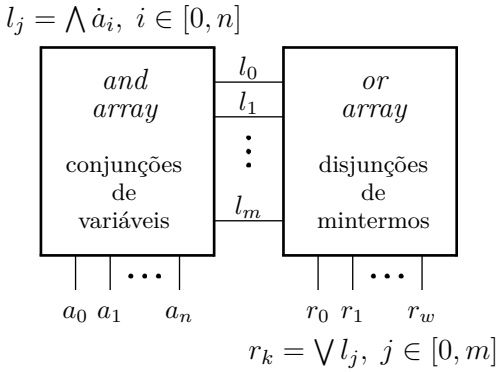


Figura 5.61: Memória ROM como soma de produtos.

A Figura 5.62 mostra o circuito de uma memória ROM com 4 bits de altura e 1 bit de largura. O decodificador ativa exatamente uma das quatro entradas do *or* – estas saídas são as linhas da matriz 4×1. Se na intersecção de uma linha com a coluna houver um transistor, quando aquela linha é selecionada o transistor conduz, r fica em 0 e a saída s em 1. Se não há um transistor na intersecção, quando a linha é selecionada, o *pull-up* mantém o sinal r em 1 e a saída s fica em 0. O inversor na saída restaura e reforça o sinal na saída da ROM.

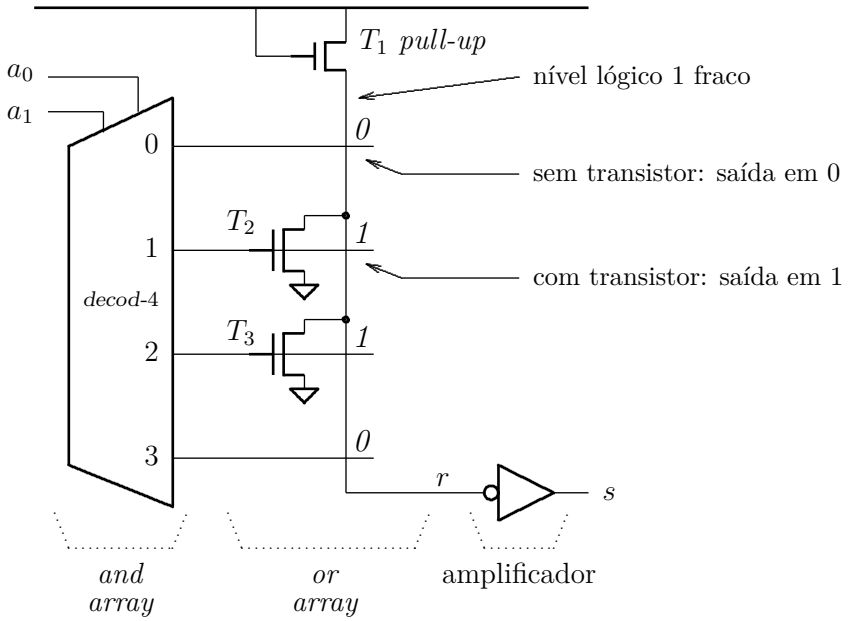


Figura 5.62: ROM 4x1.

O circuito da Figura 5.62 é uma memória ROM com 4x1 bits – *quatro palavras de um bit*. Esta nomenclatura é a usual para descrever a organização externa de memórias: altura×largura, sendo a 'altura' o número de palavras da memória, e a 'largura' o número de bits de cada palavra. Para escolher uma das quatro palavras são necessárias duas *linhas de endereço*, que são os sinais a_0 e a_1. A saída da ROM apresenta a palavra selecionada, e neste exemplo é o valor do bit selecionado por $\langle a_1, a_0 \rangle$. Qual é a função de duas variáveis, $f(a_0, a_1) = s$, implementada nesta memória ROM?

No topo da figura, o transistor T_1 é um transistor do tipo N e seus terminais *gate* e fonte estão ligados à VCC, e seu terminal dreno ligado à porta *or*. Transistores tipo N não conduzem bem o nível lógico 1 e quando não há um transistor tipo N puxando a linha de bit para zero (T_2 ou T_3), T_1 mantém um nível lógico 1 fraco. O inversor na saída garante níveis lógicos fortes para os circuitos que usam a saída da ROM. Da forma como desenhada, uma linha vertical da matriz, que é uma disjunção, é chamada de *linha de bit* (*bit line*), enquanto que as saídas do decodificador são chamadas de *linhas de palavra* (*word line*) porque selecionam uma palavra da memória. Lembre que esta matriz tem somente uma coluna.

Nas linhas 0 e 3, o cruzamento da linha de seleção com a linha de bit significa que não há contato elétrico entre estes dois sinais. A Figura 5.63 (a) mostra uma versão simplificada do diagrama da ROM 4x1. Quando a linha 0 é selecionada, não há ligação entre a saída do decodificador e a linha de bit, e portanto a saída da ROM é o complemento do 1 fraco no dreno de T_1.

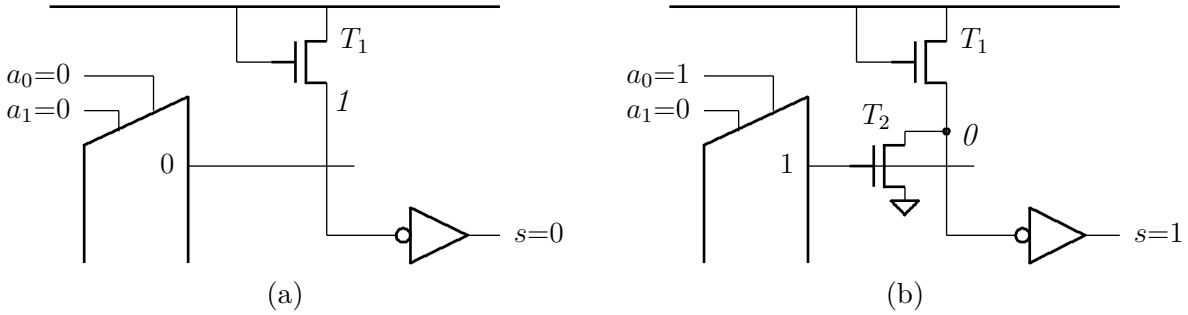


Figura 5.63: ROM: (a) bit em 0; (b) bit em 1.

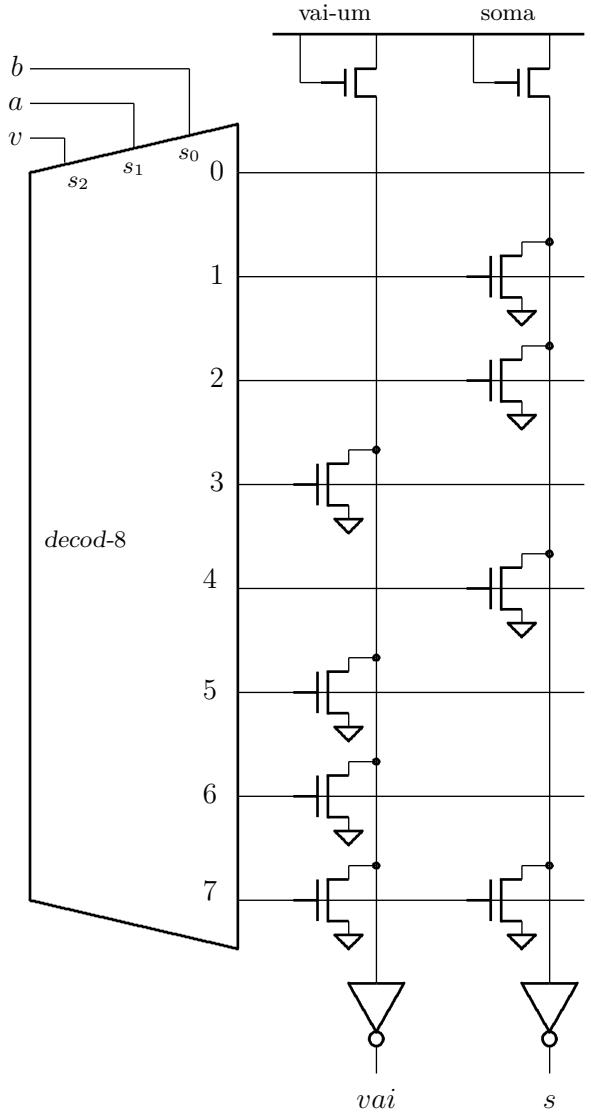
Nas linhas 1 e 2, há um transistor tipo N, com seu *gate* ligado à saída do decodificador – com o dreno ligado à linha de bit, e a fonte ligada à GND. Quando a linha 1 é selecionada, o *gate* de T_2 fica em 1 e a linha de bit é ligada à GND. Esta ligação produz um 0 'forte' na linha de bit, que se sobrepõe ao sinal fraco produzido por T_1. A Figura 5.63 (b) mostra o transistor na linha 1.

Memórias ROM permitem implementações compactas de tabelas verdade, de uma forma similar àquela apresentada na Seção 4.3.1. O circuito da Figura 5.62 implementa a tabela verdade da função *xor*: $s = a_0 \oplus a_1$.

Exemplo 5.16 *Vejamos como implementar um somador completo com uma memória ROM. A Figura 5.64 mostra a tabela verdade de um somador completo e a implementação das funções soma e vai-um com uma memória ROM de oito linhas de dois bits cada, ou uma memória ROM 8x2. Há uma correspondência exata entre as linhas da tabela e as ligações nas intersecções. Os oito produtos são implementados no decodificador, enquanto que as somas (disjunções) são implementada nas colunas.*

Considere a soma de produtos da função vai-um. O decodificador gera os oito mintermos, um por linha, enquanto que o or da coluna soma os produtos distintos de 0, que são as linhas com transistores nas intersecções. ◁

Memórias como estas permitem a implementação rápida e eficiente de funções lógicas com muitas entradas. A implementação de uma função de oito entradas com Mapas de Karnaugh e portas lógicas é um processo tedioso e complexo, enquanto que a geração de uma ROM para esta função consiste em 'copiar' a tabela verdade da função para a ROM. Veja o Exercício 5.13.



v	a	b	vai	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figura 5.64: Somador completo implementado com uma ROM 8x2.

Na medida do possível, tenta-se projetar a memória para que ela seja mais 'quadrada' do que "alta e estreita", porque linhas longas incorrem em tempo de propagação também longo. Veja o Exercício 5.11.

Memórias ROM podem ser gravadas uma única vez, durante sua fabricação, no que é chamado de "programação por máscara". Todas as intersecções entre linhas e colunas possuem transistores mas somente aquelas posições com bits em 1 tem seus *gates* ligados à linha de seleção. Isso é obtido num dos passos finais do processo de fabricação, no qual uma das máscaras da metalização só contém conexões para os *gates* que correspondem a 1s na ROM.

Memórias PROM são *Programmable ROM* se os transistores nas intersecções podem ser programados eletricamente, ao invés de somente durante a fabricação. Memórias EPROM são *Erasable Programmable ROM* e depois de programadas eletricamente podem ser apagadas, tipicamente pela exposição à luz ultravioleta. Memórias EEPROM são *Electrically-Erasable Programmable ROM* e podem ser apagadas eletricamente. Estas últimas são extremamente

úteis porque permitem a reprogramação sem que seja necessário remover o circuito integrado de memória para sua exposição a uma fonte de luz. Memórias FLASH são um tipo de EEPROM.

Exercícios

Ex. 5.9 Projete um decodificador de 3 para 8 linhas tomando por base o circuito da Figura 5.59. Compare o número de transistores da sua implementação com o de uma implementação que usa portas *and*-3 e inversores.

Ex. 5.10 Projete uma memória ROM que implementa as funções *and*, *nand*, *or*, *nor* com 4 entradas. Compare o número de transistores necessários para a implementação das portas lógicas como visto na Seção 5.2.2, com o número de transistores na ROM. Não esqueça de contar os transistores no decodificador.

Ex. 5.11 Converta o projeto da Figura 5.64 para uma memória de altura 4, usando dois multiplexadores de duas entradas.

Ex. 5.12 Discuta as vantagens e desvantagens de implementar uma memória ROM 1024x1024, como um retângulo bem mais alto do que largo, como um retângulo bem mais largo do que alto, e aproximadamente retangular. Considere todos os circuitos envolvidos, o decodificador de linha, a matriz de dados e o multiplexador de coluna.

Ex. 5.13 Escreva um programa que, dada uma função de V variáveis, produz os números das linhas da ROM nas quais devem ser ligados os transistores.

5.6 RAM – *Random Access Memory*

Memórias que permitem leitura e escrita são chamadas de *Random Access Memory*, ou RAM[3]. Estas memórias são usadas para a armazenagem de dados, e tipicamente são construídas com matrizes de armazenagem retangulares ou quadradas. O *random* é em contraposição à "memória de acesso sequencial", que era a tecnologia disponível na época em que as RAMs foram desenvolvidas.

Memórias RAM são construídas com duas tecnologias distintas. *Memórias Dinâmicas* são construídas com grande capacidade de armazenamento – CIs de 4 a 16Gbits estão disponíveis em 2016 – e baixo custo. A outra tecnologia é a das *memórias estáticas*, construídas para tempo de acesso curto, em detrimento da capacidade. As próximas seções apresentam estas tecnologias.

5.6.1 DRAM – Memória Dinâmica

Memórias RAM dinâmicas (DRAM) empregam células com somente dois transistores, donde advém a grande densidade – muitas células por unidade de área – e a grande capacidade. O 'dinamismo' destas memórias é um efeito colateral da célula minúscula: um transistor com a geometria apropriada é usado como um capacitor, e a energia contida neste capacitor representa

[3]Não custa insistir: a tradução *correta* para o Português é "memória de escrita e leitura" e não o ridículo "memória de acesso randômico".

o valor do bit armazenado na célula. Como o circuito de uma célula é pequeno, com diâmetro da ordem de 10^{-8}m, o isolamento elétrico dos componentes é frágil, a carga armazenada no capacitor escapa para a vizinhança, e o valor armazenado no capacitor se perde. Por isso, o conteúdo destas memórias deve ser 'refrescado' periodicamente. Durante o "ciclo de *refresh*", o conteúdo das células é lido e então reescrito, garantindo-se assim que os dados armazenados sejam preservados. Tipicamente, um ciclo de *refresh* deve ocorrer a cada 50 ou 60ms.

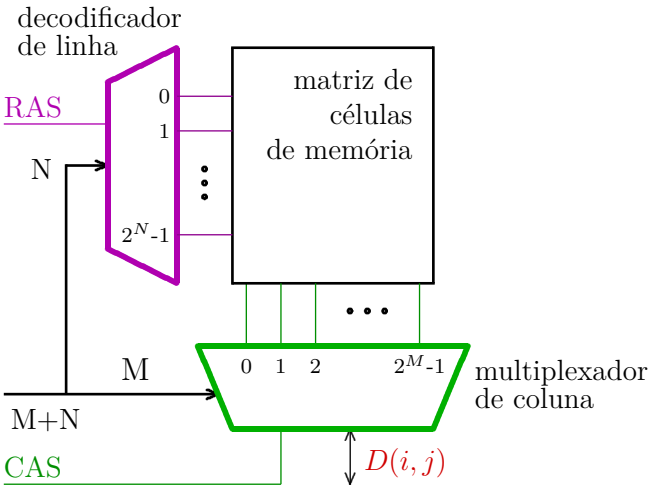


Figura 5.65: Organização de memória.

A Figura 5.65 mostra a organização de uma matriz de memória. Para uma matriz com 2^N linhas e 2^M colunas, são necessárias $N + M$ linhas de endereço para selecionar a célula de memória apontada pelo endereço indicado nos sinais N e M. Geralmente, os bits mais significativos do endereço selecionam uma das linhas da matriz.

O sinal RAS, ou *Row Address Strobe*, fica ativo para selecionar, ou ativar, uma das linhas da matriz. O sinal CAS, ou *Column Address Strobe*, fica ativo para selecionar uma das colunas.

Uma matriz de memória dinâmica é mostrada na Figura 5.66. Cada célula se liga à matriz por dois sinais, uma *linha de bit* (*bit line*), e uma *linha de palavra* (*word line*). Estes sinais selecionam uma célula e ao mesmo tempo permitem observar ou alterar seu conteúdo. Quando se usa o jargão de memória, as linhas da matriz são chamadas de *páginas*. Cuidado com o uso da palavra 'linha': esta palavra é usada (i) para os sinais de acesso às células de memória, e (ii) para a linha da matriz de dados, que é uma "página da DRAM".

Aplicando-se os números da página i e da coluna j aos endereços, se obtém acesso ao conteúdo da célula da memória com o bit $D(i, j)$. A entrada do decodificador de linha, com o número da linha, contém um registrador que armazena todos os N bits com o número da linha, e este número é capturado quando o sinal RAS fica ativo. Uma vez que o endereço da página foi armazenado, o sinal CAS é usado para selecionar uma das M colunas da matriz. Um acesso à matriz de dados ocorre em três fases: (i) o número da página é armazenado, quando o sinal RAS fica ativo; (ii) uma coluna é selecionada quando o sinal CAS fica ativo; e (iii) o conteúdo da célula é então lido da saída do multiplexador de coluna. O *controlador de memória* é o circuito que ativa os sinais certos na hora e na sequência corretas. Parece complicado, e de fato é. Em breve retornaremos ao acesso à uma célula da matriz de dados.

Uma célula de memória dinâmica consiste de um *transistor de passagem* e de um capacitor, e uma versão simplificada é mostrada na Figura 5.67. O capacitor é o *gate* de um segundo

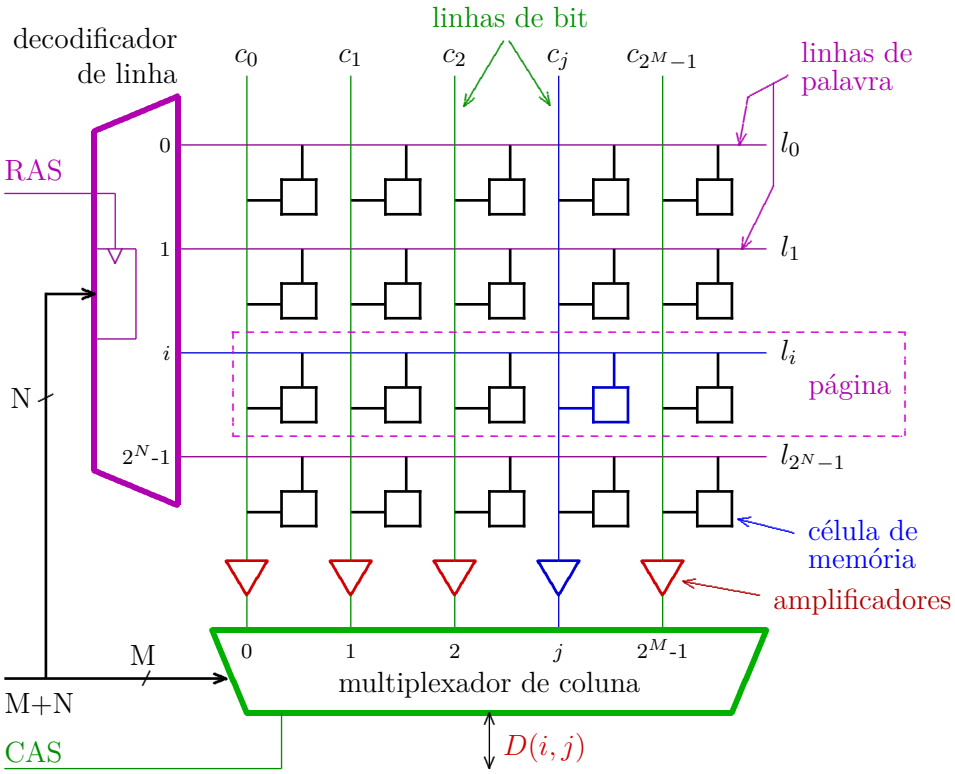


Figura 5.66: Memória DRAM com capacidade de $N \times M$ bits.

transistor, e a tensão entre as placas deste capacitor representa o bit armazenado. Quando uma página da matriz é selecionada, a linha de palavra é ativada, e o transistor de passagem conecta o terminal do capacitor à linha de bit – $(g = 1) \Rightarrow (s = d)$ – a variação de tensão na linha de bit é amostrada e amplificada, permitindo a recuperação do bit armazenado.

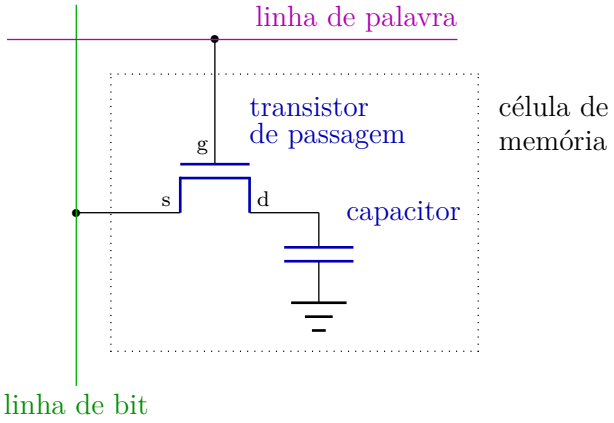


Figura 5.67: Célula de memória dinâmica.

Escrita

Numa escrita, a linha de bit é colocada no valor do bit por escrever, e então a linha de palavra é colocada em 1. O capacitor, se está carregado com uma tensão próxima de VCC, armazena o nível lógico 1, e quando está descarregado, com 0V entre as placas, armazena o nível 0. A Figura 5.68 mostra a sequência de eventos de uma escrita. O amplificador na linha de bit é usado nas leituras para restaurar o valor armazenado na célula.

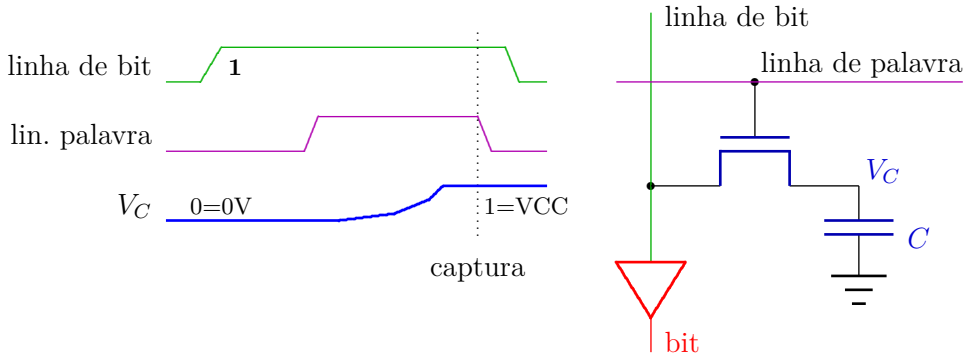


Figura 5.68: Escrita numa célula de DRAM.

Suponha que desejamos escrever 1 na célula de memória, e que o valor armazenado anteriormente era 0, e portanto o capacitor está descarregado. A linha de bit é colocada em 1, e quando a linha de palavra fica em 1, o capacitor, que estava com 0V de tensão, se carrega até VCC. Quando a linha de palavra é desativada, o novo valor é capturado porque a energia armazenada no capacitor não tem por onde escapar. Contudo, sendo frágil o isolamento, a carga escapa a energia se dissipa lentamente como calor.

Se o novo valor é igual ao anterior, a tensão armazenada sofre um reforço porque o sinal na linha de bit é mais forte do que o sinal no terminal do capacitor. Se o bit por armazenar é 0 e o valor anterior era 1, quando a linha de palavra é ativada, a tensão no capacitor cai para 0V, e assim permanece depois que a linha de palavra for desativada.

Leitura

Leituras são algo mais complexas do que escritas porque a carga armazenada nos capacitores é minúscula, da ordem de 10^6 elétrons. Na preparação para a leitura, a tensão na linha de bit é colocada exatamente em VCC/2. Quando a linha de palavra é ativada e o terminal do capacitor é ligado à linha de bit, a tensão entre as placas do capacitor gera uma pequena perturbação na tensão da linha de bit. Essa perturbação é mostrada com um traço mais espesso na linha de seleção de bit. A perturbação é amplificada e o valor original é recuperado da saída do amplificador mostrado na Figura 5.68, no sinal de nome *bit*.

As leituras são destrutivas. Para obter o valor armazenado, o capacitor é descarregado para VCC/2 se estava em 1, ou é carregado para VCC/2 se estava em 0. A descarga ocorre, e é mostrada, na fase de leitura. VCC/2 é um nível lógico indeterminado e pode contaminar o restante do circuito com valores indeterminados. Quando uma célula é lida, o valor recuperado deve ser gravado novamente, com seu nível lógico restaurado. A restauração do valor degradado é indicada como a *fase de restauração*.

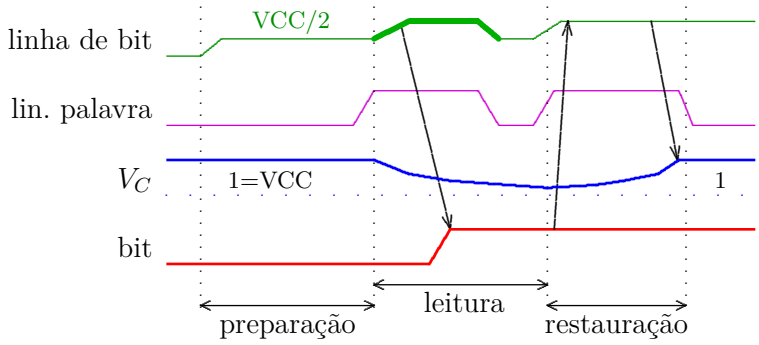


Figura 5.69: Preparação, leitura e restauração de uma célula de DRAM.

Refresh

O refrescamento da memória é efetuado com se fosse uma leitura. Veja, na Figura 5.66, que todas as células de uma linha da matriz podem ser refrescadas simultaneamente. Num ciclo de *refresh*, ao contrário de uma escrita ou leitura, todas as células de uma linha são lidas, e então gravadas novamente. O controlador de memória dinâmica deve varrer todas as linhas da DRAM, e refrescar cada linha dentro de um intervalo de 50 a 60ms.

5.6.2 SRAM – Memória Estática

Memórias estáticas empregam células maiores e mais complexas que aquelas de DRAM, sendo estas memórias menos densas e com menor capacidade. Contudo, o parâmetro de projeto mais importante para memórias estáticas é o tempo de acesso, que é um tanto menor do que nas DRAMs. Em geral, memórias estáticas são implementadas junto aos processadores, no mesmo circuito integrado, e portanto seu tempo de acesso tende a ser tão curto quanto o possível, para uma dada capacidade e tecnologia de circuitos.

Uma célula de memória estática é mostrada na Figura 5.70. Esta célula emprega 6 transistores ao invés dos dois de uma célula dinâmica. A ligação circular dos dois inversores viola a nossa definição de circuito combinacional, mas os apresentamos nesta seção para manter a continuidade do assunto.

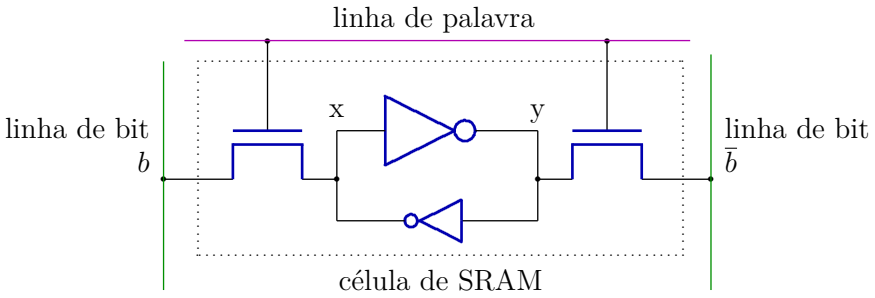


Figura 5.70: Célula de memória estática.

Os dois inversores formam um *básculo*, que é um circuito que se mantém num determinado estado se as condições externas não se alterarem e a alimentação do circuito for mantida. Quando a linha de palavra está em zero, se o sinal x está em 1, então o sinal y está em 0.

Se y está em 0, então x está em 1, e o básculo mantém um valor estável. Este valor pode ser alterado se a linha de palavra estiver em 1, e os sinais complementares em b e $\overline{b}$ forem diferentes dos valores armazenados. Os inversores têm tamanhos distintos para que, quando o valor armazenado for trocado, o inversor mais fraco perca a disputa com o inversor mais forte.

Os inversores do básculo são projetados para gerar sinais mais fracos do que os circuitos que geram os sinais b e $\overline{b}$. Numa escrita, as linhas de bit complementares devem ser colocadas nos níveis apropriados e então a linha de palavra é ativada. Se o conteúdo da célula se mantém, os valores em x e y não se alteram. Se o conteúdo da célula deve ser alterado, os valores em b e $\overline{b}$ se sobrepõem aos sinais em x e y, que acabam trocando de nível. Quando a linha de palavra é desativada, o novo valor é capturado pelo básculo.

O diagrama de blocos da memória DRAM não mostra o circuito de escrita. A Figura 5.71 mostra um diagrama de blocos de uma SRAM com mais detalhe do que aquele da DRAM. Os circuitos de decodificação de linha e de multiplexação de coluna são similares aos da DRAM. O topo do diagrama mostra o *seletor de escrita*, que escolhe qual a coluna por atualizar. Se o sinal de escrita (wr) estiver ativo, então o valor de entrada $D(i,j)$ é aplicado ao circuito de escrita da coluna selecionada.

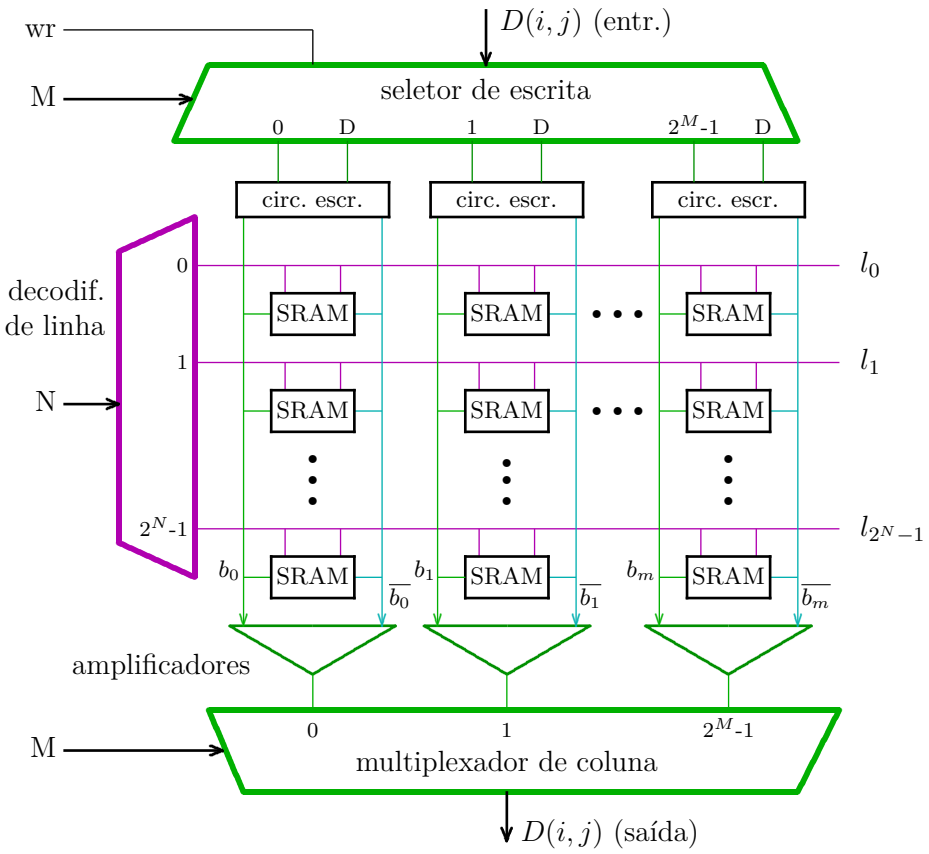


Figura 5.71: Memória SRAM com capacidade de $N \times M$ bits.

Na base do diagrama estão os *amplificadores diferenciais* que amplificam a diferença entre b e $\overline{b}$ para recuperar o valor armazenado na célula $D(i,j)$.

Numa leitura, a tensão nas linhas de bit deve ser colocada em VCC/2. Quando a linha de palavra é ativada, as saídas dos inversores provocam uma perturbação nas linhas de bit, e

o amplificador diferencial detecta a *diferença* entre as tensões em b e $\overline{b}$ e recupera o valor armazenado na célula.

Não é necessário refrescar as células após a leitura porque os básculos mantém o valor armazenado. Para acelerar o acesso ao conteúdo das células, o número da linha e o da coluna são fornecidos simultaneamente à matriz. Numa leitura, decorrido o tempo de acesso do decodificador de linha, mais o tempo para acessar o conteúdo da célula, este valor é apresentado na saída após o tempo de propagação do multiplexador de coluna.

O tempo de acesso à SRAM é menor do que o de uma DRAM porque a capacidade da memória estática é, em geral, uma fração daquela da DRAM, e portanto o tempo para decodificar a linha e multiplexar a coluna é mais curto porque estes circuitos são menores. Além disso, não há *refresh* nas SRAM, o que também reduz seu tempo médio de acesso.

5.7 Bits que não pertencem à $\mathbb{B}$

Neste capítulo introduzimos novos valores para os bits, valores estes que não pertencem à $\mathbb{B}$, e passamos a usar sinais 'fortes" e sinais 'fracos'. Como é possível conciliar a nossa lógica quase-Booleana com estes novos valores, sem aumentar exageradamente a complexidade dos modelos que empregamos para nossos circuitos?

Uma solução, relativamente simples, é adotada na linguagem VHDL, quando se empregam sinais da chamada *lógica padrão*, de tipo `std\_logic`. Sinais deste tipo podem estar num de nove níveis: não-inicializado, desconhecido-forte, 0-forte, 1-forte, alta-impedância, desconhecido-fraco, 0-fraco, 1-fraco e não-importa.

Se um sinal fraco é ligado a um sinal forte, o nível do sinal forte predomina. Um curto circuito entre 1-forte e 0-forte resulta em desconhecido-forte. Um sinal desconhecido-forte contamina todos os circuitos aos quais esteja ligado.

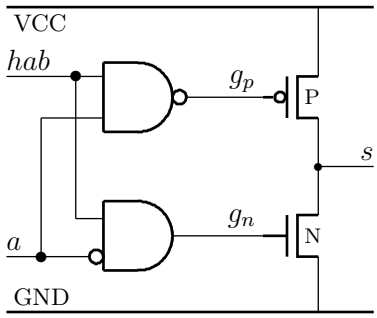
Para fins de simulação, sinais não-inicializados equivalem a desconhecido-forte porque a máquina de simulação é incapaz de adivinhar os valores iniciais desejados para os sinais.

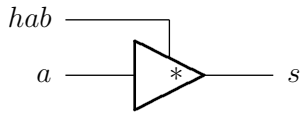
Vejamos os circuitos que operam com um *terceiro estado* que não é 0 e nem é 1, e ainda *multiplexadores analógicos* construídos com dois transistores.

5.7.1 Terceiro Estado

É possível projetar circuitos digitais com um *terceiro estado*, além dos estados 0 e 1. No terceiro estado, a saída do circuito forma um caminho que oferece alta resistência à passagem de corrente, desligando-a dos circuitos aos quais está conectada. Circuitos que operam com os três estados são chamados de *three-state* ou *tri-state*.

Estes circuitos possuem um sinal de controle que permite colocar a saída no terceiro estado com o efeito de desligá-la dos circuitos à que está conectada. A Figura 5.72 mostra o circuito interno de um *buffer three-state* – um *buffer* funciona como um amplificador que não altera o nível lógico do sinal, apenas torna-o "mais forte". O símbolo do circuito é mostrado no lado direito da figura. O asterisco junto à saída é uma indicação visual de que esta saída pode operar no terceiro estado.





hab	*a*	g_p	g_n	P	N	*s*
0	X	1	0	aberto	aberto	desligado
1	0	1	1	aberto	conduz	0
1	1	0	0	conduz	aberto	1

Figura 5.72: *Buffer three-state* CMOS.

Quando o sinal de habilitação está inativo ($hab = 0$) os níveis nos *gates* dos dois transistores ligados à saída fazem com que eles fiquem abertos, de forma a que não haja nenhum caminho de baixa resistência que ligue a saída à VCC ou à GND. Um sinal ligado a uma saída *three-state* é indeterminado, e se diz que está *flutuando* se não há um circuito que puxe o nível lógico neste sinal para 0 ou para 1.

Circuitos *three-state* são usadas para a ligação de várias saídas a um mesmo fio, formando um *barramento*, como mostra a Figura 5.73. Os circuitos que geram os sinais a_n, b_n e c_n têm suas saídas ligadas ao fio $barr_n$, e este conduz o sinal que é entrada para outros circuitos que produzem os sinais x_n e y_n.

O circuito de controle deve ser projetado para garantir que no máximo um dentre *habA*, *habB*, ou *habC*, esteja ativo a qualquer momento. Se mais de uma saída estiver ligada simultaneamente, o nível de sinal resultante em $barr_n$ pode assumir um nível lógico indeterminado, se um dos sinais estiver em 1 e o outro em 0. Note que este circuito implementa um multiplexador de três entradas.

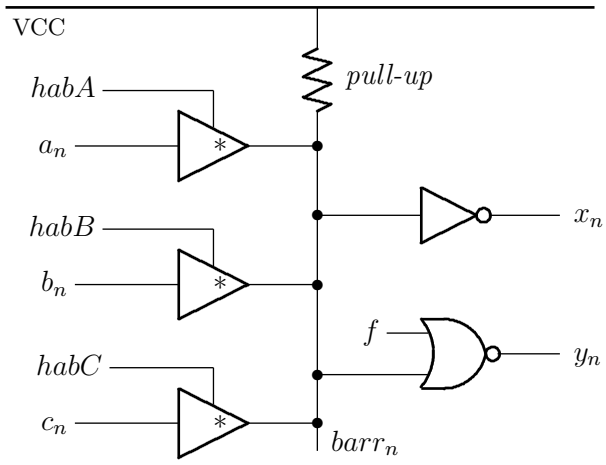


Figura 5.73: Ligação de saídas *three-state*.

Quando todas as saídas estão no terceiro estado, o nível lógico do sinal é indeterminado porque nada se pode dizer sobre ele. Para garantir que este sinal fique em um nível lógico determinado, um resistor é ligado à VCC. Este resistor é chamado de *pull-up* porque sua função é puxar o nível lógico do sinal para 1 se todas as saídas estão no terceiro estado. Quando uma das saídas é ativada e está em 0, o transistor N desta saída puxa o nível lógico para 0, sobrepondo-se ao 1-fraco do *pull-up*. O *pull-up* é chamado de "circuito passivo" porque somente atua quando

nenhum dos circuitos ativos – portas lógicas ou *buffers* – está puxando o nível lógico do sinal para 0 ou para 1.

Em geral, circuitos *three-state* são usados para interligar circuitos integrados, e raramente são usados internamente aos CIs. A temporização destes circuitos é complexa porque deve-se garantir que somente uma fonte de sinal esteja ativa a cada instante. Multiplexadores são usados internamente aos CIs porque sua construção impede curto-circuitos entre as várias fontes de sinal.

5.7.2 Portas de Transmissão

O terceiro estado é empregado por causa das características dos transistores CMOS que permitem o desligamento do circuito de saída. Outro circuito útil, também dependente do comportamento elétrico dos transistores é a *porta de transmissão* (*transmission gate*). A Equação 5.3 indica que se o *gate* de um transistor CMOS está no nível lógico ativo, então seus terminais fonte e dreno estão conectados (chave fechada). Lembre que transistores do tipo P transmitem bem o nível lógico 1 e transmitem mal o nível 0. O mesmo vale para transistores do tipo N, com os níveis trocados – o nível 0 é transmitido bem, e o nível 1 fracamente.

Uma porta de transmissão pode ser implementada com dois transistores e sinais de controle adequados, como indica a Figura 5.74. Os dois transistores são necessários para garantir que sinais dos dois níveis são transmitidos entre os terminais "de dados". Como os transistores são simétricos com relação aos níveis nos terminais fonte e dreno, este circuito permite a circulação de corrente da entrada para a saída, e da saída para a entrada, ao contrário das outras portas lógicas que estudamos. Por esta razão, estes circuitos são também chamados de *chaves analógicas*.

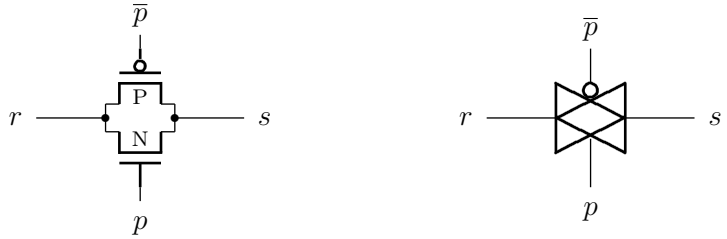


Figura 5.74: Porta de transmissão.

Quando os sinais de passagem estão ativos, $p = 1$ e $\overline{p} = 0$, os terminais r e s ficam interligados e no mesmo nível lógico. O símbolo desta porta parece com dois *buffers*, um em cada direção, justamente para indicar a capacidade de transmissão nos dois sentidos.

Portas de transmissão podem ser usadas para implementar multiplexadores e circuitos de deslocamento e rotação. A Figura 5.75 mostra um multiplexador de duas entradas para sinais com 3 bits de largura. Quando $selA = 1$, as portas de transmissão ligadas ao sinal $A{=}\langle a_0, a_1, a_2\rangle$ transmitem, enquanto que as portas ligadas ao sinal $B{=}\langle b_0, b_1, b_2\rangle$ são desligadas. Quando $selA = 0$, a situação se inverte e o sinal B é ligado a C.

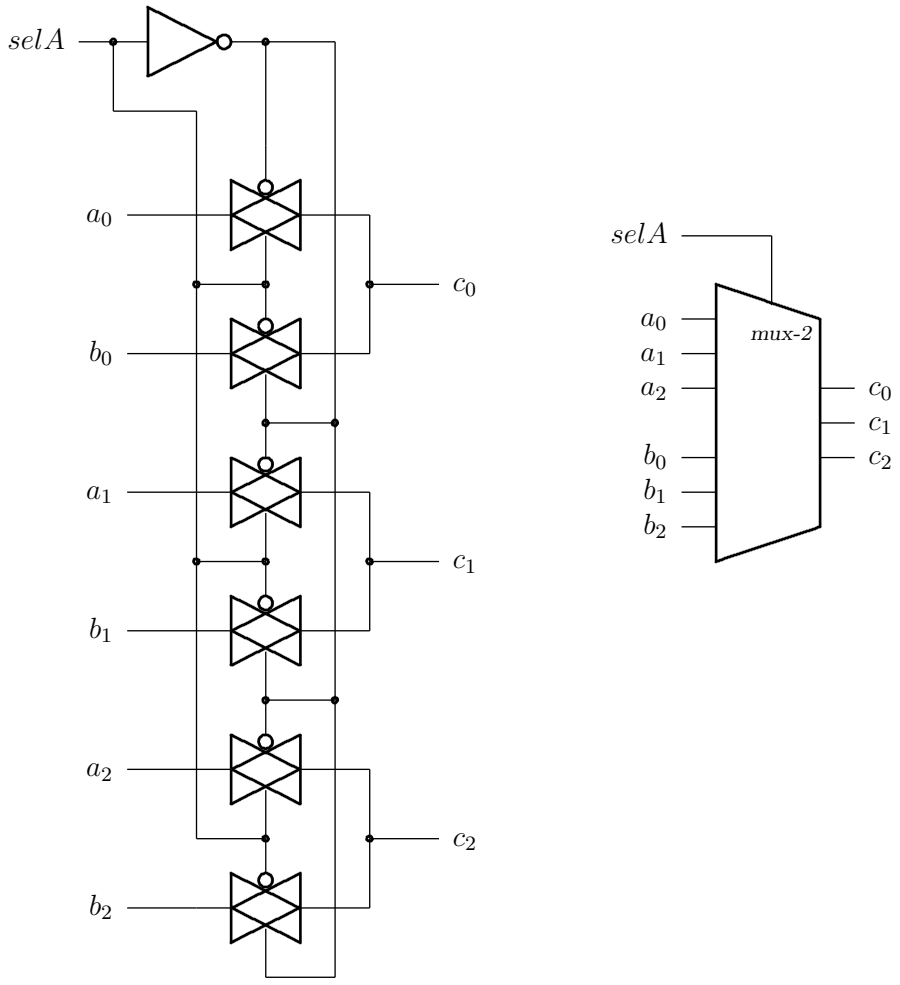


Figura 5.75: Multiplexador com três bits de largura.

Exercícios

Ex. 5.14 É possível implementar um *buffer three-state* com menos de 10 transistores? Não use portas de transmissão.

Ex. 5.15 Mostre como implementar a função lógica $\vee$ com *buffers three-state*.

Ex. 5.16 Usando a técnica da Figura 4.13, mostre como implementar um multiplexador de 8 entradas com portas de transmissão.

Ex. 5.17 Mostre como implementar um deslocador de um bit com portas de transmissão.

Índice Remissivo

W

X

Z