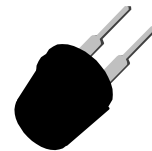
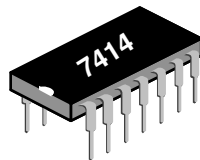
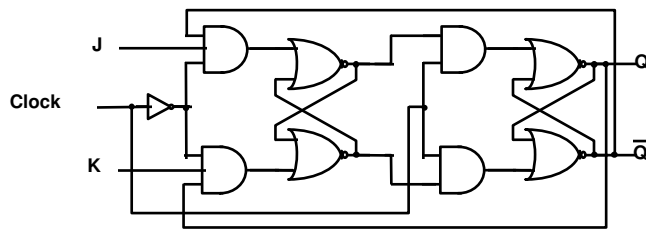
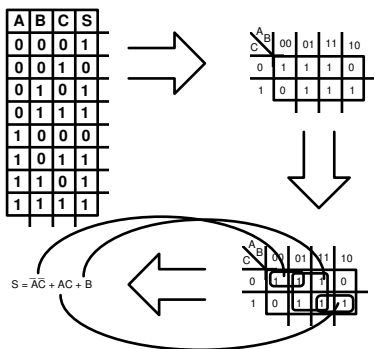
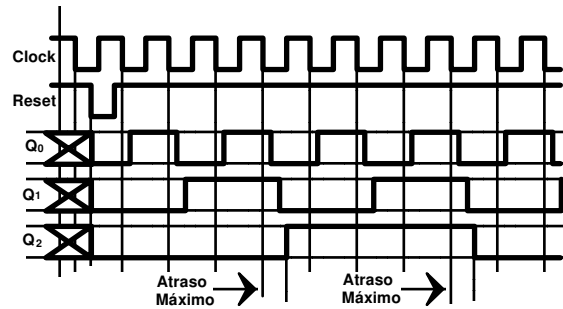
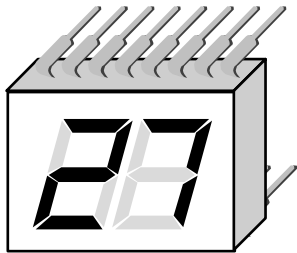


Elementos Básicos da Eletrônica Digital



Professor

Frederico Oioli de Campos

Prefácio

Esta apostila é a compilação do conteúdo das aulas de Eletrônica Digital que ministrei na ETE Júlio de Mesquita desde 1991 até 1996. Todo ano o conteúdo sofreu alteração visando a atualização e introdução de novos conceitos.

A primeira vez que ministrei esta matéria(1991), usava a preparação de aula propriamente dita para compor a lousa e os alunos a copiavam. Nos anos seguintes (1992, 1993 e 1994), usando uma cópia xerox do caderno do aluno Fábio Sandon, alterava, no próprio quadro negro e na cópia, os pontos falhos e os incrementava com novos assuntos. Em 1995, a carga horária da disciplina foi reduzida pela metade e não havia outra saída a não ser a composição de uma apostila. Em 1996, a apostila teve nova editoração, correção de erros e, finalmente em 1997, graças aos atentos olhos dos alunos Vagner Alves da Cunha e Eric Danzi Lemos mais erros foram corrigidos e a nova edição é apresentada nas páginas seguintes.

O conteúdo a ser estudado compreende os elementos básicos da Eletrônica Digital, pontos de partida elementares desta Ciência e que são muito requisitados em exames de qualificação para o mercado de trabalho para técnicos em Eletrônica.

A bibliografia usada para a elaboração de todos os textos e esquemas está relacionada a seguir e cabe a mim alertar que os apontamentos de aulas, feitos na minha graduação pela Faculdade de Engenharia Industrial, também foram uma importante fonte de pesquisa.

*Frederico Oioli de Campos
São Paulo, 13 de Setembro de 2001*

Índice:

Adjacência	22
Álgebra Booleana	5
Analógico X Digital	1
Apêndice 1 - Portas Lógicas	75
Apêndice 2 – Data Books	80
Apêndice 3 – Formas Padrão	87
Bibliografia	91
Chave anti rebote	38
Circuitos MEALY	59
Circuitos Combinacionais	27
Circuitos MOORE	54
Circuitos MOORE com Flip-Flops RS e JK	57
Circuitos Seqüenciais	50
Circuitos Seqüenciais - Elementos Básicos	35
Circuitos Seqüenciais - MOORE e MEALY	54
Codificador	31
Contador Binário Assíncrono	52
Contador Binário de Módulo Arbitrário Assíncrono	52
Contador em Anel	50
Contador em Anel Torcido	51
Conversão Analógica/Digital por modulação DELTA ou modulador PWM	72
Conversão da Base DECIMAL para a Base BINÁRIA	3
Conversão de Base BINÁRIA para a Base DECIMAL	4
Conversão de Bases	3
Conversão Digital/Analógica para seqüência de BYTES	68
Conversão Digital/Analógica por demodulação de PWM	74
Conversor A/D com comparadores de tensão	71
Conversor A/D com quantização em Bytes	71
Conversor D/A a resistor ponderado	68
Conversor D/A de escada R-2R	69
Conversores D/A e A/D	68
Decodificador	30
Demultiplexador	33
Eletrônica Digital	1
Endereçamento de um Mapa de Karnaugh	19
Enlace	22
Estado	54
Flip-Flop JK	46
Flip-Flop RS	42
Flip-Flop tipo D	44
Flip-Flop tipo T	47
Função COMPLEMENTO	6
Função E COINCIDÊNCIA	15
Função E ou AND	7
Função IGUALDADE	6
Função NE ou NAND	11
Função NOU ou NOR	11
Função OU EXCLUSIVO ou EXCLUSIVE OR	14
Função OU ou OR	8
Funções Booleanas	5
Funções de DUAS OU MAIS variáveis binárias	7
Funções de UMA variável binária	6
Funções e Portas Lógicas Especiais	14
Identidades Auxiliares	13
Irrelevância	55
Latch RS Assíncrono	35
Latch RS Síncrono	38
Latch Tipo D	40

LM 555.....	61
Mapa de Karnaugh	19
Mapa de Karnaugh de 3 Variáveis	21
Mapa de Karnaugh de 4 Variáveis	24
Mapa K de 2 Variáveis.....	20
Meio Somador	27
Multiplexador	32
Multivibrador Astável	65
Multivibrador Mono-Estável	63
Registrador de Deslocamento	50
Resumo da Álgebra de Boole, Teoremas e Identidades.....	18
Resumo das Funções e Portas Lógicas.....	17
Seqüência.....	54
Somador Completo.....	29
Somadores	27
Tabela de Estados.....	54
Temporizadores.....	61
Teorema de De Morgan	12
Transcodificador.....	31
Vantagens da ELETRÔNICA DIGITAL	2

Eletrônica Digital

Introdução

A Eletrônica é dividida em dois segmentos que, certamente todos, já ouvimos falar:

- 1) Eletrônica Analógica
- 2) Eletrônica Digital

A Disciplina **Sistemas Digitais e Microprocessadores (SDM)**, ministrada nos primeiro e segundo módulos do curso de Eletrônica da ETE Júli de Mesquita, introduz o aluno à este ramo da Eletrônica através do estudo de seus *Elementos Básicos* e da *Álgebra de Boole*.

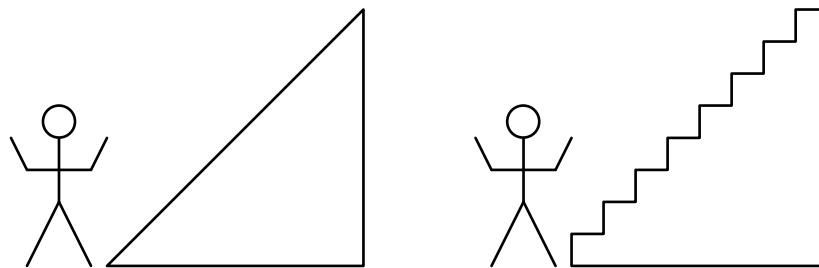
Analógico X Digital

No dia-a-dia encontramos diversos tipos de aparelhos eletrônicos que são classificadas como **DIGITAIS** ou **ANALÓGICOS**. Esta classificação fica por conta do produtor do aparelho ou então nós mesmos acabamos por classifica-los intuitivamente. Mas, afinal, quais são os parâmetros científicos usados para classificar um produto eletrônico em **ANALÓGICO** ou **DIGITAL**?

Antes de mais nada, precisamos definir as palavras ANALÓGICO e DIGITAL.

Usando de um exemplo bastante grosseiro podemos ter uma primeira idéia:

- a) Rampa X Escada



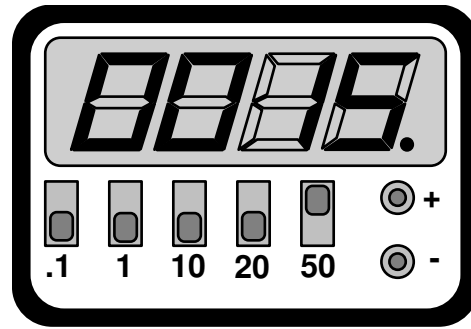
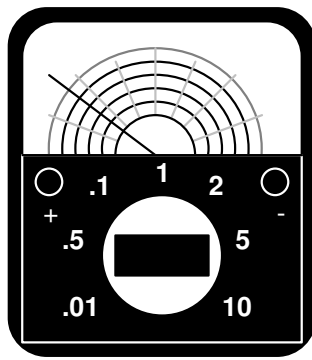
Ao analisarmos a RAMPA percebemos que se uma pessoa começar a subi-la, poderá ocupar cada uma das infinitas posições existentes entre o início e o fim, já no caso da ESCADA, a pessoa poderá estar em apenas um dos seus 8 degraus. Sendo assim, podemos dizer, com um certo receio, que a RAMPA está para o ANALÓGICO, assim como a ESCADA está para o DIGITAL.

- b) Voltímetro ANALÓGICO X Voltímetro DIGITAL

Enquanto no Voltímetro ANALÓGICO, o ponteiro pode ocupar **infinitas** posições entre o maior e o menor valor da escala, no Voltímetro DIGITAL os valores mostrados pelo *display* são discretos, isto é, existe um número **finito** de valores entre o maior e o menor valor da escala.

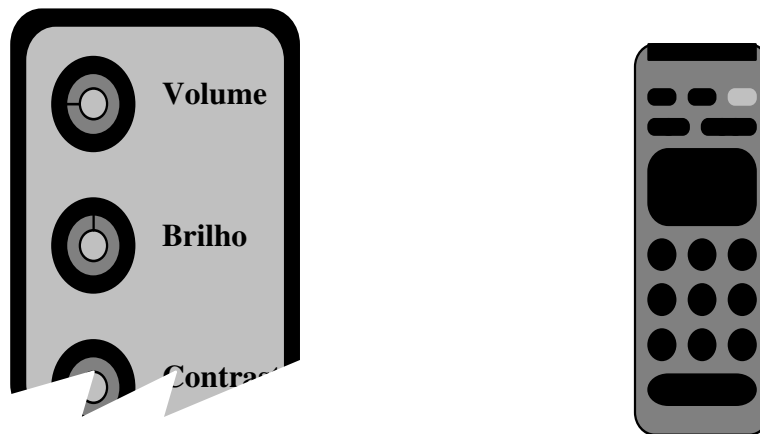
Através destes exemplos, podemos concluir que a classificação dita ANALÓGICA será dada a todo dispositivo que puder apresentar **infinitas** saídas (ou resultados) entre dois pontos preestabelecidos, em contra partida, todo dispositivo que apresentar **finitas** saídas (ou resultados) será designado de DIGITAL.

Usando termos mais científicos dizemos que um dispositivo é ANALÓGICO quando a sua saída for uma função **contínua** e que um dispositivo é DIGITAL quando a sua saída for uma função **discreta**.



No caso dos voltímetros, o processo pelo qual medimos a tensão elétrica entre dois pontos resulta em **saídas**. Porém em determinadas situações, as **entradas** que são ANALÓGICAS ou DIGITAIS:

c) Botão de Volume X Controle Remoto



Para ajustar o volume de seu televisor, usando o "botão", você terá **infinitas** posições para escolher, mas no controle remoto observamos que a intensidade do som muda em pequenos saltos e, em alguns modelos, aparece no vídeo o valor selecionado, normalmente de 0 a 50. É importante observar que você não consegue estabelecer o valor 19,5 para o volume do televisor à controle remoto, pois os saltos de valores são de 1 em 1.

Podemos dizer, então, que o televisor com "botão" tem em seu circuito de som uma **entrada** ANALÓGICA para o ajuste e que o televisor à controle remoto tem em seu circuito de som uma **entrada** DIGITAL.

Há, ainda, dispositivos com **entradas** e **saídas** ANALÓGICAS e **processamento** DIGITAL, como o *Compact Disk Player* ou CD Player, onde o som original é ANALÓGICO por natureza, a gravação é feita de forma DIGITAL, e na reprodução temos novamente o som ANALÓGICO.

Finalmente podemos dizer, com segurança, que a Eletrônica Analógica processa **sinais com funções contínuas** e a Eletrônica Digital processa **sinais com funções discretas**.

Vantagens da ELETRÔNICA DIGITAL

Como vimos nos exemplos acima, uma saída digital apresenta um número finito de valores e por isso fica muito mais simples o trabalho com estes sinais, já um dispositivo analógico, com infinitos valores, precisa de uma análise muito detalhada, para que o trabalho seja executado sem que se perca partes do sinal.

Para simplificar ainda mais o processamento de sinais digitais, foi retomada uma antiga técnica de numeração, a numeração **BINÁRIA**, que usa apenas dois símbolos para a representação de números. Como os sinais são **discretos** e portanto mensuráveis facilmente, se enumerarmos esses valores usando a numeração **BINÁRIA** teremos um *Conjunto Universo* com apenas dois elementos distintos para representarmos os sinais desejados. Isso tudo quer dizer que num dispositivo digital eletrônico teremos o processamento conjuntos finitos cujos elementos se apresentam em apenas dois valores. A esses conjuntos dá-se o nome de **BYTES** e aos seus elementos, o nome de **BITS**.

Pode ser que até esse instante esses conceitos ainda estejam confusos para você, mas no decorrer do curso as coisas se esclarecerão facilmente e de forma natural. Vamos nos concentrar agora em um ponto muito importante: a conversão de números decimais para binário e vice-versa.

Conversão de Bases

Conversão da Base DECIMAL para a Base BINÁRIA

A base de um sistema de numeração é o número de **cifras** usadas para a representação das quantidades. Em nosso dia-a-dia, usamos a base decimal para representarmos nossas quantidades como: idade, dinheiro, datas, peso, medidas, etc. As dez cifras usadas são:

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

A combinação destes símbolos nos permite infinitas representações de quantidades.

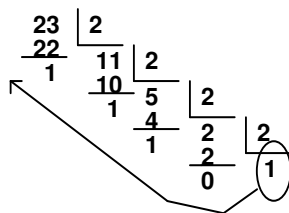
Como já foi dito, a Eletrônica Digital usa a base **BINÁRIA** para o processamento de seus sinais e por analogia podemos concluir que esta base é formada por apenas **duas cifras**:

0 e 1

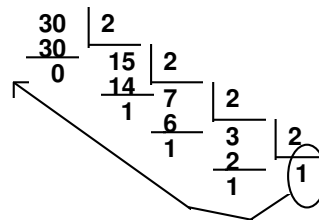
Usando apenas esses dois símbolos, também podemos representar infinitas quantidades e de forma totalmente equivalente à numeração **DECIMAL** conforme mostram os exemplos abaixo:

$$\begin{array}{lll} (5)_{10} = (101)_2 & (10)_{10} = (1010)_2 & (15)_{10} = (1111)_2 \\ (63)_{10} = (111111)_2 & (1)_{10} = (1)_2 & (1024)_{10} = (1000000000)_2 \end{array}$$

A regra básica para fazermos a conversão de **DECIMAL** para **BINÁRIO** é a *divisão sucessiva por 2*, esquematizada logo a seguir:



$$(23)_{10} = (10111)_2$$



$$(30)_{10} = (11110)_2$$

O *algoritmo* para a execução desta conversão é:

- Dividir por 2 o número que se deseja converter ;
- Se o *quociente* (resultado) for **diferente** de 1, dividir este *quociente* por 2;
- Se o novo *quociente* for diferente de 1 repetir os itens b) e c) até que o *quociente* seja igual a 1;
- O **BINÁRIO** equivalente ao **DECIMAL** é o último *quociente* colocado lado-a-lado com todos os restos das divisões, de baixo para cima.

Exercícios:

Converter os números representados em **DECIMAL** para a representação **BINÁRIA**:

- 33
- 27
- 45
- 31
- 32

Conversão de Base BINÁRIA para a Base DECIMAL

Também podemos fazer a conversão de bases de maneira inversa, isto é, a partir de um número em BINÁRIO chegamos ao seu equivalente em DECIMAL. Da mesma forma que os números DECIMAIS podem ser decompostos em múltiplos de 10 os números em BINÁRIO podem ser decompostos em múltiplos de 2:

$$\begin{aligned}(47602)_{10} &= 40000 + 7000 + 600 + 00 + 2 = \\ &= 4 \times 10^4 + 7 \times 10^3 + 6 \times 10^2 + 0 \times 10^1 + 2 \times 10^0\end{aligned}$$

$$\begin{aligned}(10010)_2 &= 10000 + 0000 + 000 + 10 + 0 = \\ &= 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = \\ &= 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = (18)_{10}\end{aligned}$$

Em ambos casos, o valor da *cifra* usada para a representação do número é multiplicado pela base do número que é elevada a $n-1$, onde n é o número de cifras que compõem o número. Observe que a na segunda linha do segundo exemplo é que ocorre a conversão da base BINÁRIA para a DECIMAL e na terceira linha temos apenas "contas" para resolver.

Exercícios:

Converter os números representados em **BINÁRIO** para a base **DECIMAL**:

- a) 1001010
- b) 101010
- c) 111101
- d) 1000000
- e) 11111

Álgebra Booleana

Introdução

Na França do século passado, um filósofo chamado **George Boole** desenvolveu uma sistemática de análise de situações bastante peculiar. Para o equacionamento e resolução de seus problemas, o filósofo analisava cada ponto envolvido na questão e os atribuía apenas **duas** hipóteses completamente opostas. Exemplos:

ACESO	APAGADO
PERTO	LONGE
CLARO	ESCURO
VERDADEIRO	FALSO
NORTE	SUL
LESTE	OESTE
SIM	NÃO
HIGHT	LOW
0	1

Um típico problema analisável pela lógica de **Boole** está descrito abaixo:

Um fazendeiro chamado Bastião tinha **dois** celeiros, um no lado **norte** da sua fazenda e outro no lado **sul**, um lobo, um bode e vários pés de couve. Bastião trabalhava duro todo dia e ainda tinha que vigiar seus pertences pois lobos apreciam os bodes e bodes apreciam pés de couve. O pobre fazendeiro caminhava, várias vezes por dia, de um celeiro a outro, com as couves dentro de uma sacola em suas costas e com uma vara bem comprida nas mãos, onde numa extremidade estava amarrado o lobo e na outra o bode.

Este problema, analisado pela lógica **booleana** teria a seguinte estrutura:

- 1) Se o lobo é deixado com o bode, na ausência de Bastião, ele vai comer o bode.
- 2) Se o bode é deixado com os pés de couve, quando Bastião estiver ausente, ele vai comer os pés de couve.
- 3) Bastião, o lobo, os pés de couve e o bode podem estar no celeiro do **norte** ou no do **sul**.

George **Boole**, em sua tese, propunha o uso de **variáveis binárias** para o equacionamento e resolução deste tipo de problema e definia essas variáveis como sendo aquelas que podem assumir apenas dois valores.

O mundo, na época de **Boole**, usava seus estudos apenas na filosofia, mas desde o surgimento da Eletrônica Digital, as regras de Boole vem sendo a base fundamental para qualquer estudo nessa área.

Na matéria Eletrônica Digital I, vamos aprender a **álgebra** que Boole criou para a resolução de problemas equacionados em variáveis **binárias** e também como construir pequenos dispositivos capazes nos solucionar problemas dinâmicos como o do fazendeiro Bastião.



Funções Booleanas

A álgebra desenvolvida por **Boole** pode ser dividida em dois grupos de funções e nós assumiremos que as variáveis envolvidas são **binárias** e podem assumir apenas os valores 0 e 1.

1) Funções de UMA variável binária

Sejam duas variáveis binárias A e Z onde Z é função de A:

$$Z = f(A)$$

Como A e Z podem assumir apenas dois valores (0 ou 1) temos apenas duas funções capazes de relaciona-las:

1.1) Função IGUALDADE

$Z = A$ (Z é igual a A), ou seja:

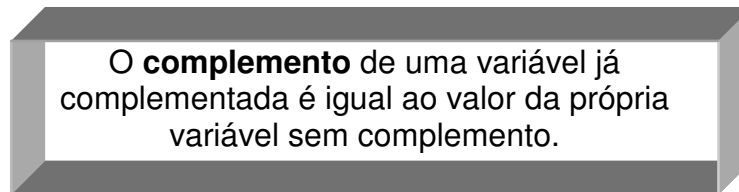
se $A = 0$, Z também é igual a 0, ou
se $A = 1$, Z também é igual a 1.

1.2) Função COMPLEMENTO

$Z = \bar{A}$ (Z é o complemento de A), ou seja:

se $A = 0$, Z é igual a 1, ou
se $A = 1$, Z é igual a 0.

A função 1.2 dá origem á primeira propriedade das funções da álgebra de Boole:



Exemplo:

$$A = 0 \Rightarrow \bar{\bar{A}} = 1 \Rightarrow A = 0$$

Exercícios:

Determine o valor de S nos casos abaixo:

a) $A = \frac{1}{B}$
 $S = \bar{B}$
 $B = \frac{C}{A}$
 $C = \bar{A}$

b) $\frac{C}{D} = 0$
 $\bar{D} = B$
 $\bar{B} = \bar{C}$
 $S = D$

2) Funções de DUAS OU MAIS variáveis binárias

Sejam n variáveis binárias A, B, C, n e Z , onde Z é função de A, B, C, n :

$$Z = f(A, B, C, n)$$

Como agora nos envolvemos com mais de uma variável, teremos um número maior de funções capazes de relacioná-las através da lógica:

2.1) Função E ou AND

$$Z = A \cdot B \quad \text{ou} \quad Z = AB$$

Z assumirá o valor 1 se, e somente se, A e B forem 1.

Exemplo:

Dados os valores das variáveis binárias A, B, C e D , calcule o valor de S .

$$A = 1 \qquad B = 1 \qquad C = 0 \qquad D = 0$$

$$\begin{aligned} E &= A \cdot B \\ F &= \overline{B} \cdot C \\ G &= C \cdot E \\ H &= G \cdot D \\ S &= \overline{H} \cdot A \end{aligned}$$

Solução:

$$\begin{aligned} - \text{se } A = 1 & \quad e \quad B = 1 \Rightarrow E = 1 \\ - \text{se } B = 1 & \Rightarrow \overline{B} = 0 \\ - \text{se } \overline{B} = 0 & \quad e \quad C = 0 \Rightarrow F = 0 \\ - \text{se } C = 0 & \quad e \quad E = 1 \Rightarrow G = 0 \\ - \text{se } G = 0 & \quad e \quad D = 0 \Rightarrow H = 0 \\ - \text{se } H = 0 & \Rightarrow \overline{H} = 1 \\ - \text{se } \overline{H} = 1 & \quad e \quad A = 1 \Rightarrow \boxed{S = 1} \end{aligned}$$

A **Função E** pode relacionar infinitas variáveis e não apenas 2 como está sugerindo a definição anterior ou mesmo o exemplo. Por este motivo temos que reavaliar a sua definição, mesmo que em nossa disciplina (Eletrônica Digital) usemos poucas vezes mais que 5 variáveis em uma mesma equação.

Seja uma função $f(A, B, C, D, n) = Z$ onde todas as variáveis se relacionam pela Função E, Z assume o valor 1 se, e somente se, todas as variáveis forem 1.

A função **E** (ou **AND**) tem as propriedades **Elemento Neutro** e **Elemento Nulo** muito parecidas com as mesmas propriedades da multiplicação, mas a função **E** não pode ser confundida com esta *operação aritmética* pois é uma *função lógica*.

2.1.1 - Elemento Neutro

$$A \bullet 1 = A$$

A função **E** aplicada entre uma variável e 1 resulta o próprio valor da variável.

2.1.2 - Elemento Nulo

$$A \bullet 0 = 0$$

A função **E** aplicada entre uma variável Binária e 0 resulta sempre 0.

2.1.3 - Elemento Complementar

$$A \bullet \bar{A} = 0$$

A função **E** aplicada entre uma variável e seu complemento resulta sempre 0.

2.1.4 - Comutativa

$$A \bullet B = B \bullet A$$

A ordem em que aplicamos a função **E** em duas variáveis não altera o resultado da equação.

2.1.5 - Associativa

$$(A \bullet B) \bullet C = A \bullet (B \bullet C)$$

Se numa equação temos várias variáveis relacionadas apenas pela função **E** podemos calcular o seu resultado sem nos preocupar com a ordem em que aplicamos a função.

Exercícios:

Determine os valores de S nos casos abaixo:

$$\begin{aligned} \text{a) } A &= 1 \\ B &= 0 \\ C &= A \bullet 1 \\ D &= F \bullet 0 \\ E &= D \bullet A \\ S &= \bar{E} \end{aligned}$$

$$\begin{aligned} \text{b) } C &= 0 \\ B &= \bar{C} \bullet 1 \\ \bar{A} &= B \bullet C \bullet 0 \\ S &= A \end{aligned}$$

2.2) Função OU ou OR

$$Z = A + B$$

Z assumirá o valor 1 se, A **ou** B **ou** ambas forem 1.

Exemplo:

Dados os valores das variáveis binárias A, B, C e D, calcule o valor de S.

$$A = 1$$

$$B = 1$$

$$C = 0$$

$$D = 0$$

$$E = A + B$$

$$F = B \cdot C$$

$$G = \overline{C} \cdot E + F$$

$$H = \overline{G} \cdot D \cdot A$$

$$S = H + A$$

Solução:

$$\begin{array}{llll} - \text{ se } A = 1 & e & B = 1 & \Rightarrow E = 1 \\ - \text{ se } B = 1 & \Rightarrow & C = 0 & \Rightarrow F = 0 \\ - \text{ se } C = 0 & , & \overline{C} = 1 & e F = 0 \Rightarrow G = 0 \\ - \text{ se } G = 0 & \Rightarrow & \overline{G} = 1 & \Rightarrow H = 0 \\ - \text{ se } \overline{G} = 1 & , & D = 0 & e A = 1 \\ - \text{ se } H = 0 & e & A = 1 & \Rightarrow \boxed{S = 1} \end{array}$$

Como podemos observar no exemplo, a **Função OU** pode relacionar mais de duas variáveis e então temos que melhorar a sua definição:

Seja uma função $f(A, B, C, D, n) = Z$ onde todas as variáveis se relacionam pela Função OU, Z assume o valor 1 se, pelo menos uma das variáveis, estiver nível lógico 1.

A função **OU** (ou **OR**) tem as propriedades **Elemento Neutro** e **Elemento Nulo** muito parecidas com as mesmas propriedades da adição, mas a função **OU** não pode ser confundida com esta *operação aritmética* pois é uma *função lógica*.

2.2.1 - Elemento Neutro

$$A + 0 = A$$

A função **OU** aplicada entre uma variável e 0 resulta no próprio valor da variável.

2.2.2 - Elemento "Nulo"

$$A + 1 = 1$$

A função **OU** aplicada entre uma variável e 1 sempre resulta 1. Observe que a palavra **"Nulo"** nos induz a pensar que o resultado da expressão será 0, mas neste caso a função resulta 1 e, portanto, devemos entender que a função se anula resultando sempre 1.

2.2.3 - Elemento Complementar

$$A + \overline{A} = 1$$

A função **OU** aplicada entre uma variável e seu complemento sempre resulta 1.

2.2.4 - Comutativa

$$A + B = B + A$$

A ordem em que aplicamos a função **OU** em duas variáveis não altera o resultado da equação.

2.2.5 - Associativa

$$(A + B) + C = A + (B + C)$$

Se numa equação temos várias variáveis relacionadas apenas pela função **OU** podemos calcular o seu resultado sem nos preocupar com a ordem em que aplicamos a função.

Além dessas propriedades que as funções **E** e **OU**, apresentam isoladamente, temos também outra propriedade quando analisamos as duas funções simultaneamente:

2.2.6 - Distributiva

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

Se podemos aplicar a propriedade distributiva entre variáveis booleanas relacionadas pelas funções **E** e **OU** podemos também colocar variáveis em evidência, quando nos for conveniente. Exemplo:

$$A \cdot B + C \cdot B + D \cdot B = B \cdot (A + C + D)$$

Exercícios:

a) Verificar se as igualdades são verdadeiras ou falsas:

a) $(A + B) \cdot (A + C) = A + BC$

b) $A + BA = B$

2) Simplificar as expressões:

a) $F = (A + B)(B + C) + BC + BA$

b) $F = (AB + AC + AD)(A + B)$

c) $F = (A + B)(C + D)(A + D)(B + C)$

d) $F = A(B(C + D) + C)$

e) $F = (A + B)(A + C)(A + D)$

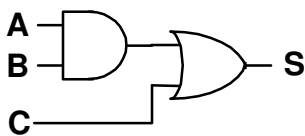
f) $F = (A + B)(A(D + C)) + AB$

Conforme discutimos anteriormente neste capítulo e mais detalhadamente no **Apêndice 1**, a Eletrônica Digital desenvolveu circuitos capazes de executarem as Funções Booleanas e também criou símbolos especiais para cada circuito. Sendo assim podemos representar equações complexas usando apenas símbolos. Exemplos:

Representação Algébrica

$$S = AB + C$$

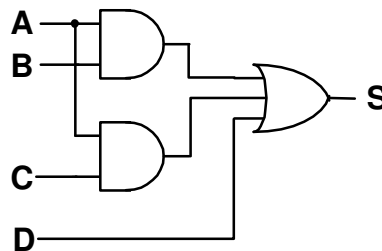
Representação Esquemática



Representação Algébrica

$$S = AB + AC + D$$

Representação Esquemática



Exercícios:

1) Representar esquematicamente as funções abaixo:

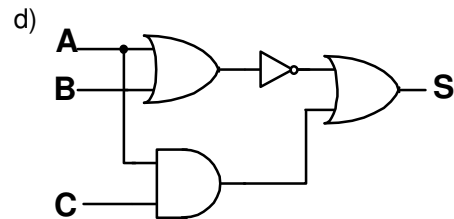
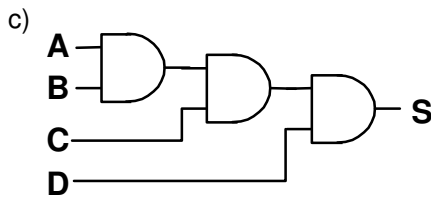
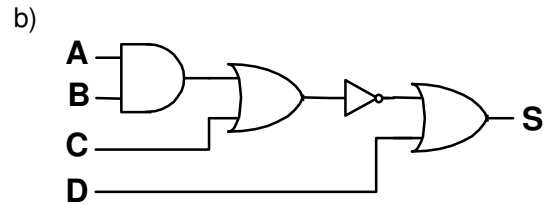
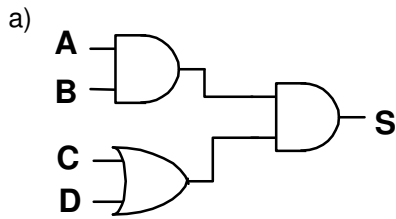
a) $S = A + BC + AC$

b) $S = (A + B)(C + D)(A + D)$

c) $S = \overline{A} + BC + D$

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

2) Representar algebricamente as funções esquematizadas abaixo:



Da mesma forma que usamos a função complemento (ou a porta inversora) para calcularmos o complemento de uma variável, podemos calcular o complemento de uma função Booleana, associando em sua saída uma porta inversora. Na verdade, na Álgebra de Boole temos as funções complementares correspondentes às funções **E** e **OU** designadas como funções independentes e com nomes próprios:

2.3) Função NE ou NAND

$$Z = \overline{A \cdot B} \quad \text{ou} \quad Z = \overline{AB}$$

Z assumirá o valor 0 se, e somente se, A e B forem 1.

Como nos casos anteriores, precisamos de uma definição mais completa para a função, ou seja, uma definição que possa garantir a sua aplicação para um número qualquer de variáveis.

Seja uma função $f(A, B, C, D, n) = Z$ onde todas as variáveis se relacionam pela Função NE, Z assume o valor 0 se, e somente se, todas as variáveis forem 1.

2.4) Função NOU ou NOR

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

Z assumirá o valor 0 se, A ou B ou ambas forem 1.

Melhorando a definição temos:

Seja uma função $f(A, B, C, D, n) = Z$ onde todas as variáveis se relacionam pela Função NOU, Z assume o valor 0 se, pelo menos uma das variáveis, estiver nível lógico 1.

Exercícios:

1) Representar esquematicamente as funções abaixo:

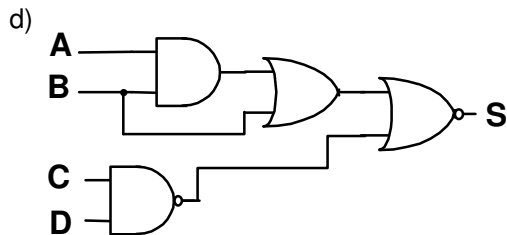
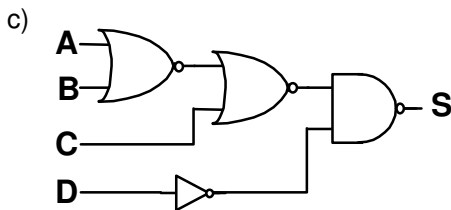
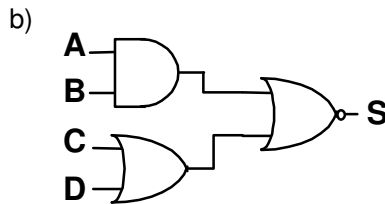
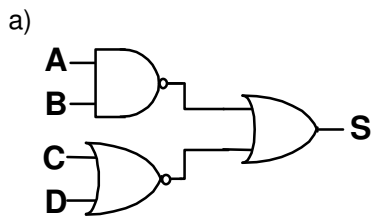
a) $S = A + \overline{B \cdot C} + B$

b) $S = \overline{A + B} \cdot \overline{C + D}$

c) $S = \overline{AB} + AC + \overline{A + D}$

d) $S = \overline{\overline{A + B} + C} + \overline{AB + AC}$

2) Representar algebricamente as funções esquematizadas abaixo:



Um outro estudioso, também da época de Boole, enunciou um teorema que nos permite transformar uma função E em uma função OU e vice-versa e, obviamente o teorema ganhou o seu nome:

3) Teorema de De Morgan

$$\overline{A \cdot B \cdot C \cdot D \cdot n} = \overline{A} + \overline{B} + \overline{C} + \overline{D} + \overline{n}$$

O complemento da função **E** aplicado à n variáveis é igual à função **OU** aplicada a essas mesmas n variáveis complementadas.

ou então:

$$\overline{A + B + C + D + n} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{n}$$

O complemento da função **OU** aplicado à n variáveis é igual à função **E** aplicada a essas mesmas n variáveis complementadas.

Exercícios:

1) Verificar se as identidades são verdadeiras ou falsas:

a) $\overline{AB} + AC = A + B$

b) $\overline{\overline{AB}} + AC = A + B$

2) Simplificar as expressões:

a) $F = \overline{\overline{A + B + C} + AC} + \overline{\overline{AB}} + \overline{\overline{BC}}$

b) $F = \overline{\overline{A \cdot B \cdot C} + A \cdot B} + \overline{\overline{A \cdot C} + B}$

c) $F = \overline{\overline{A + B + C + D} \cdot \overline{\overline{ABC}} + \overline{B}}$

d) $F = \overline{\overline{A + \overline{B}} + C + D} \cdot \overline{\overline{ABCD}}$

3) João vai ao cinema se Alice for com ele e se ele puder usar o carro da família. Entretanto, Alice decidiu ir à praia se não estiver chovendo e se a temperatura estiver acima de 26°C. O pai de João fez planos para usar o carro para visitar amigos se estiver chovendo ou se a temperatura estiver acima de 26°C.

Equacione o problema utilizando a Álgebra de Boole de maneira que esta equação seja 1 quando João pode ir ao cinema.

Vamos finalizar este capítulo com um estudo em mais duas funções de Boole aplicáveis a apenas duas variáveis. Um estudo mais detalhado sobre essas funções será feito posteriormente, quando então analisaremos as suas aplicações para um número maior de variáveis.

4) Identidades Auxiliares

Podemos ainda usar três identidades na redução de circuitos lógicos. São elas:

a) $A + AB = A$

Se colocarmos A em evidência, temos:

$$A(1 + B) = A$$

Como $1 + B = 1$, então:

$$A \cdot 1 = A, \text{ ou seja:}$$

$$A = A$$

$$b) A + \overline{AB} = A + B$$

Conforme já vimos, o complemento de um complemento não altera uma expressão, temos:

$$A + \overline{AB} = \overline{\overline{A + \overline{AB}}}$$

Reduzindo a expressão pelo teorema de De Morgan:

$$= \overline{A \cdot \overline{AB}} = \overline{A \cdot (\overline{A + B})} = \overline{A \cdot (A + B)}$$

Aplicando a Distributiva:

$$= \overline{AA + AB} = \overline{0 + AB} = \overline{AB} = \overline{A} + \overline{B} \\ = A + B$$

$$c) (A + B)(A + C) = A + BC$$

Aplicando a Distributiva:

$$= AA + AC + BA + BC \\ = A + AC + AB + BC \\ = A + A(C + B) + BC \\ = A(1 + C + B) + BC \\ = A \cdot 1 + BC \\ = A + BC$$

5) Funções e Portas Lógicas Especiais

Temos ainda duas funções lógicas e suas respectivas portas que devem ser encaradas de forma especial pois a elas não se aplicam diretamente as propriedades e teoremas estudados até agora. São elas:

5.1 Função OU EXCLUSIVO ou EXCLUSIVE OR

$$Z = A \oplus B$$

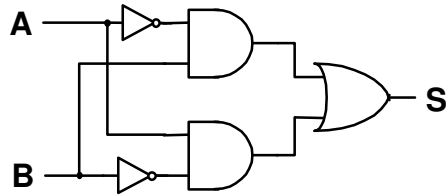
Z assumirá o valor 1 se, e somente se, A e B tiverem valores diferentes.

Podemos analisar o circuito que executa esta função a partir da associação de portas lógicas já estudadas. Isso facilita o entendimento e é absolutamente suficiente já que o nosso propósito não é o estudo da Eletrônica Digital a nível de componentes discretos e sim a nível de circuitos integrados.

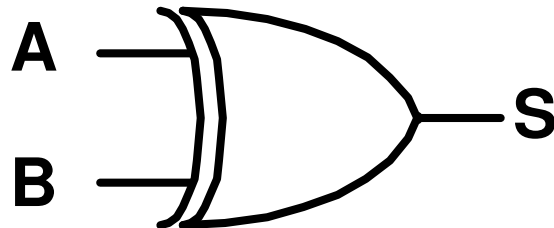
Se a função **OU EXCLUSIVO** assume o valor 1 somente quando os valores em suas entradas são diferentes temos apenas duas possibilidades para que isso aconteça (considerando que esta função está sendo aplicada em apenas duas variáveis):

$$A = 0 \text{ e } B = 1 \quad \text{ou} \quad A = 1 \text{ e } B = 0$$

Temos, então a seguinte associação capaz de executar essa função:



A simbologia usada para representar esta função é:



5.2 Função E COINCIDÊNCIA

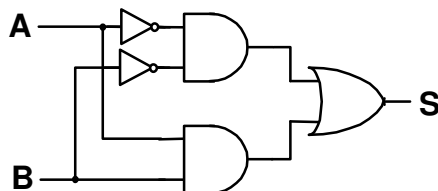
$$Z = A \odot B$$

Z assumirá o valor 1 se, e somente se, A e B tiverem valores iguais.

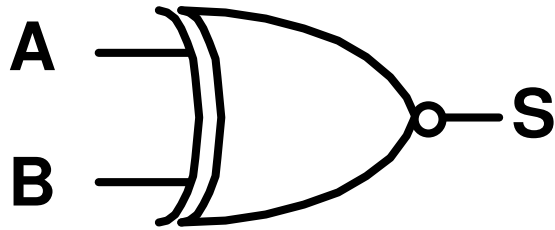
Se a função **E COINCIDÊNCIA** assume o valor 1 somente quando os valores em suas entradas são iguais temos, também, apenas duas possibilidades para que isso aconteça (considerando, também que esta função está sendo aplicada em apenas duas variáveis):

$$A = 0 \text{ e } B = 0 \quad \text{ou} \quad A = 1 \text{ e } B = 1$$

Da mesma forma que o caso anterior, vamos analisar a função **E COINCIDÊNCIA** usando a associação de portas lógicas já estudadas:



A simbologia para representar esta função é:



É muito normal e muito prático usarmos tabelas para mostrarmos os valores que uma função Booleana pode assumir, pois se considerarmos um número finito de variáveis estas tabelas terão um número finito de linhas e representarão **todos** os resultados possíveis. Para calcularmos o número de linhas para representarmos todas as situações basta usarmos a seguinte relação:

$$n_{\text{linhas}} = (2)^{n_{\text{variáveis}}}$$

Exemplo:


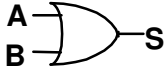





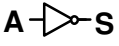
Uma tabela que represente a função **E** aplicada a duas variáveis deve ter:

$$n_{\text{linhas}} = 2^2 = 4$$

Esta tabela tem a aparência ilustrada abaixo, e recebe o nome de **Tabela Verdade** pois é capaz de representar **todas** as situações possíveis para o número especificado e variáveis:

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

Resumo das Funções e Portas Lógicas

Nome da Função	Representação Algébrica	Representação Lógica	Tabela Verdade															
E ou AND	$S = A \cdot B$		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1
A	B	S																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OU ou OR	$S = A + B$		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1
A	B	S																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NE ou NAND	$S = \overline{A \cdot B}$		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0
A	B	S																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOU ou NOR	$S = \overline{A + B}$		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0
A	B	S																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
OU EXCLUSIVO	$S = A \oplus B$		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0
A	B	S																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
E COINCIDÊNCIA	$S = A \odot B$		<table border="1"> <thead> <tr><th>A</th><th>B</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	1
A	B	S																
0	0	1																
0	1	0																
1	0	0																
1	1	1																
IGUALDADE "DRIVER"	$S = A$		<table border="1"> <thead> <tr><th>A</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table>	A	S	0	0	1	1									
A	S																	
0	0																	
1	1																	
COMPLEMENTO "INVERSOR"	$S = \bar{A}$		<table border="1"> <thead> <tr><th>A</th><th>S</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	S	0	1	1	0									
A	S																	
0	1																	
1	0																	

Mapa de Karnaugh

Introdução

No capítulo anterior vimos toda a Álgebra de Boole e também como simplificarmos as funções usando seus teoremas e propriedades. Agora estudaremos uma nova metodologia para conseguirmos fazer as mesmas simplificações ou reduções de funções lógicas. Esta nova metodologia foi criada com o intuito de tornar mais simples o nosso trabalho. Veitch e Karnaugh, foram dois estudiosos do século passado que tornaram possível a simplificações de funções lógicas por simples observação visual da tabela verdade, quando esta está transcrita em mapas especialmente criados para este procedimento.

Endereçamento de um Mapa de Karnaugh

O mapa de Karnaugh nada mais é que uma tabela verdade escrita de uma forma diferente. Ele é composto pelo um número de **células** igual ao número de **linhas** da tabela verdade e, portanto, tem 2^n células, onde n é o número de variáveis que compõem a função. Então antes de mais nada temos que saber como é que se transcreve uma tabela para um mapa de Karnaugh e também que saber como é que é este mapa.

Acredito que todos nós saibamos como jogar um jogo chamado **Batalha Naval** que tem o seguinte aspecto:

	A	B	C	D	E	F
1						
2		*				*
3		*			*	
4		*		*		
5		*				
6					*	*

Se sabemos jogar Batalha naval, sabemos que a fileira vertical composta por quatro asteriscos tem os seguintes endereços:

B2, B3, B4 e B5

Por analogia, as fileiras compostas por três asteriscos em diagonal e a fileira composta por dois asteriscos na horizontal tem, respectivamente os seguintes endereços:

D4, E3 e F2 e
E6 e F6

Se entendemos esta sistemática de **endereçamento** podemos verificar que num mapa de Karnaugh o processo é muito parecido. Observe o exemplo de um Mapa K de quatro variáveis:

A C/D \ B	00	01	11	10
00				α
01		δ		
11				
10				β

O endereço da célula α é:

A = 1, B = 0, C = 0 e D = 0

O endereço da célula β é:

A = 1, B = 0, C = 1 e D = 0

e, finalmente, o endereço da célula δ é:

A = 0, B = 1, C = 0 e D = 1

Observe a maneira particular que colocamos os valores em binário. Eles não estão na ordem que estamos acostumados a usa-lo e esta é justamente a maneira particular que caracteriza o mapa de Karnaugh.

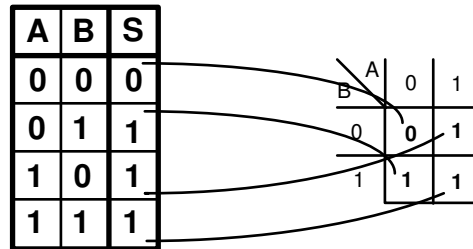
Para exemplificarmos o endereçamento de um mapa K fica mais fácil e mais claro iniciarmos com um mapa de quatro variáveis, mas didaticamente vamos estudar primeiro os mapas de 2 e 3 variáveis para então chegarmos o de 4.

Mapa K de 2 Variáveis

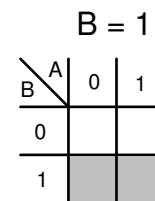
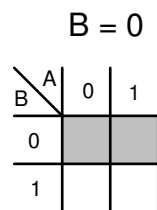
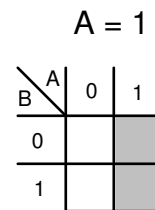
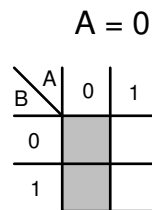
Uma mapa de Karnaugh de duas variáveis tem o seguinte aspecto e conforme a sistemática de endereçamento vista anteriormente teria a seguinte transcrição da sua respectiva tabela verdade:

Tabela Verdade

Mapa de Karnaugh



Analisando um mapa K detalhadamente podemos identificar regiões onde A é sempre 0, onde B é sempre 0, onde A é sempre 1 e onde B é sempre 1, conforme ilustrado abaixo:



Se voltarmos ao primeiro exemplo do mapa de Karnaugh de 2 variáveis podemos entender como esta metodologia funciona. Observe que as regiões em que a função tem como resultado o valor 1 são as regiões em que A = 1 ou em que B = 1 e isso nos dá a simplificação de Karnaugh, ou seja:

$$S = A + B$$

Percebemos, então que esta é a própria função **OU** e já deveríamos esperar por isso, pois a tabela verdade é a tabela da função OU.

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

B \ A	0	1
0	0	1
1	1	1

$$S = A + B$$

Vamos analisar agora o caso da função **E**. Temos a sua tabela verdade e a respectiva transcrição para o mapa de Karnaugh:

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

B \ A	0	1
0	0	0
1	0	1

Pelo mapa K observamos que única **célula** em que a função apresenta como saída o valor 1 é justamente a intersecção das regiões em que $A = 1$ e $B = 1$, então dizemos que $S = A \cdot B$.

Exercícios:

Escrever as funções representadas pelas tabelas verdade abaixo:

a)

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

b)

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

c)

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

d)

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

Mapa de Karnaugh de 3 Variáveis

Podemos analisar também funções de três variáveis através dos mapas K, e para isso basta usarmos dois mapas de duas variáveis associados convenientemente. Temos então duas formas de associá-los que são completamente equivalentes:

		A			
		B	00	01	11
C	0				
	1				

		A	
		0	1
B	C	00	01
		11	10

A partir deste instante temos que definir alguns parâmetros para prosseguirmos os nossos estudos. São eles:

1) Adjacência

Consideraremos duas células de um mapa de Karnaugh **adjacentes** se, e somente se, as variáveis que a endereçam apresentem **apenas uma** mudança de valor. Exemplos:

		A	
		0	1
B	C	α	β
		ϵ	δ
		γ	ϕ
		φ	η

As células α e β são adjacentes pois para α $A = 0$, $B = 0$ e $C = 0$ e para β , $A = 1$, $B = 0$ e $C = 0$. Percebemos então que apenas A apresentou mudança em seu valor.

As células γ e η não são adjacentes pois para γ $A = 0$, $B = 1$ e $C = 1$ e para η , $A = 1$, $B = 1$ e $C = 0$. Percebemos então que A e C apresentaram mudanças em seus valores.

Exercícios:

Dado mapa de Karnaugh anterior, indicar se as células listadas abaixo são adjacentes ou não, justificando a sua resposta:

- a) α e φ
- b) γ e ϕ
- c) η e β
- d) ϵ e φ

3) Enlace

Enlace é o agrupamento que fazemos no mapa K afim de visualizarmos as células adjacentes. De cada enlace teremos uma expressão booleana correspondente e estes nos darão o resultado do mapa que é a função simplificada. Os enlaces só podem agrupar um número de células que seja igual a uma potência de dois ou seja $1 (2^0)$, $2 (2^1)$, $4 (2^2)$, $8 (2^3)$ etc.

Um mapa de Karnaugh de 3 variáveis na sua forma horizontal pode ter apenas os seguintes enlaces:

Enlaces de 1 célula

	A	B			
C	\	00	01	11	10
0		□	□	□	□
1		□	□	□	□

Enlaces de 2 células

	A	B			
C	\	00	01	11	10
0		□	□	□	□
1		□	□	□	□

	A	B			
C	\	00	01	11	10
0		□		□	□
1		□		□	□

	A	B			
C	\	00	01	11	10
0		□	□		□
1		□	□		□

Enlaces de 4 células

	A	B			
C	\	00	01	11	10
0		□		□	□
1		□		□	□

	A	B			
C	\	00	01	11	10
0		□	□	□	
1		□	□	□	

	A	B			
C	\	00	01	11	10
0		□			
1		□			

Enlace de 8 células

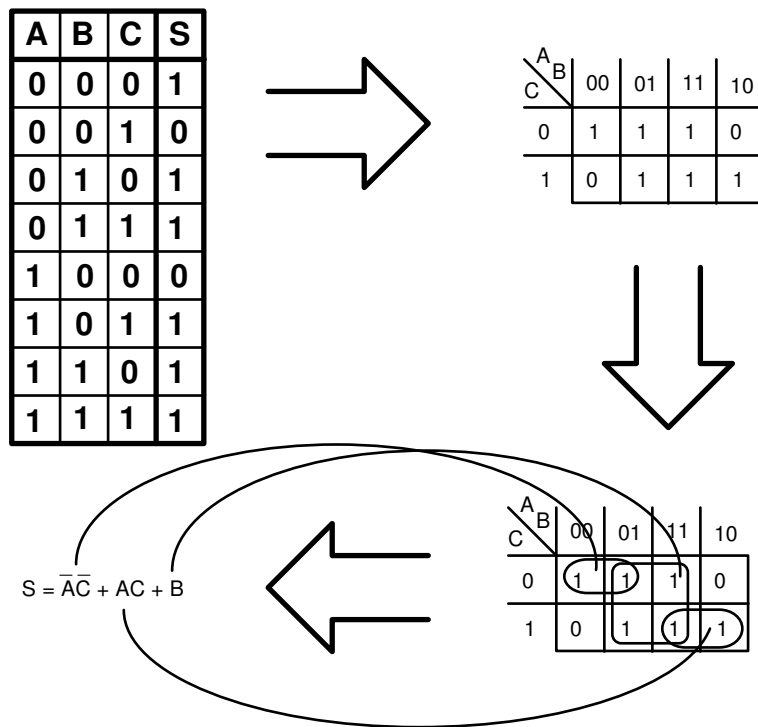
	A	B			
C	\	00	01	11	10
0		□			
1		□			

Podemos concluir então que cada enlace define uma região onde as variáveis de endereçamento apresentam uma propriedade em comum. Portanto para resolvermos um mapa de Karnaugh devemos seguir os seguintes passos:

- 1) Identificar as células cujos valores são 1
- 2) Fazermos os enlaces permitidos (observando as adjacências e o número de células do enlace)
- 3) Deduzirmos a expressão booleana para cada enlace e agruparmos essas expressões através da função **OU**.

Exemplo:

Deduzir a função booleana que representa a tabela verdade abaixo usando o mapa de Karnaugh:



Exercícios:

Deduzir as funções booleanas representadas pelas tabelas verdade a seguir:

a)

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

b)

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

c)

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

Mapa de Karnaugh de 4 Variáveis

Podemos analisar também funções de quatro variáveis através dos mapas K, e para isso basta usarmos dois mapas de três variáveis associados convenientemente.

A \ B \ C \ D	00	01	11	10
00				
01				
11				
10				

As regras de **adjacências** e de **enlaces** para o mapa de Karnaugh de 4 variáveis continuam sendo as mesmas já que estas regras valem para mapas com qualquer número de células. Por isso, neste caso não vamos analisar todos os tipos de enlaces possíveis, pois podemos correr o risco de criar vícios aos alunos que passariam a procurar esta referência ao invés de deduzir quais enlaces são válidos a partir da análise de cada

caso. Para ilustrar o procedimento da resolução segundo Karnaugh em um mapa de 4 variáveis citaremos um exemplo:

Dada a tabela verdade abaixo, deduza a função booleana utilizando o mapa K:

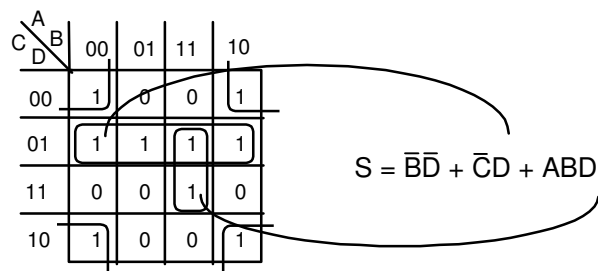
Observações importantes:

Para não cometermos erros no momento de fazermos os enlaces, devemos observar duas regras:

1) Fazer primeiro os enlaces com maior número de células, pois caso contrário corremos o risco de fazermos agrupamentos que poderiam ser substituídos por um maior.

2) Verificar se em cada enlace existe pelo menos uma célula que pertença a apenas um enlace, pois corremos o risco de fazermos enlaces redundantes, ou seja, enlaces perfeitamente dispensáveis.

A	B	C	D	S
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1



Obs.

O termo $\overline{B}\overline{D}$ é devido ao grupo formado quatro

Exercícios:

Deduzir as funções booleanas representadas pelas tabelas verdade a seguir:

a)	<table border="1"><tr><th>A</th><th>B</th><th>C</th><th>D</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	A	B	C	D	S	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1	0	0	1	0	1	0	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	1	1	0	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1
A	B	C	D	S																																																																																		
0	0	0	0	0																																																																																		
0	0	0	1	1																																																																																		
0	0	1	0	0																																																																																		
0	0	1	1	1																																																																																		
0	1	0	0	1																																																																																		
0	1	0	1	1																																																																																		
0	1	1	0	0																																																																																		
0	1	1	1	1																																																																																		
1	0	0	0	1																																																																																		
1	0	0	1	1																																																																																		
1	0	1	0	1																																																																																		
1	0	1	1	1																																																																																		
1	1	0	0	1																																																																																		
1	1	0	1	1																																																																																		
1	1	1	0	1																																																																																		
1	1	1	1	1																																																																																		
b)	<table border="1"><tr><th>A</th><th>B</th><th>C</th><th>D</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	A	B	C	D	S	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1	1	0	0	1	0	0	1	0	1	0	1	1	0	1	1	0	1	0	1	1	1	1	1	0	0	0	1	1	0	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	0	0	1	1	1	1	0
A	B	C	D	S																																																																																		
0	0	0	0	1																																																																																		
0	0	0	1	0																																																																																		
0	0	1	0	1																																																																																		
0	0	1	1	0																																																																																		
0	1	0	0	1																																																																																		
0	1	0	1	1																																																																																		
0	1	1	0	1																																																																																		
0	1	1	1	1																																																																																		
1	0	0	0	1																																																																																		
1	0	0	1	1																																																																																		
1	0	1	0	1																																																																																		
1	0	1	1	1																																																																																		
1	1	0	0	0																																																																																		
1	1	0	1	0																																																																																		
1	1	1	0	0																																																																																		
1	1	1	1	0																																																																																		
c)	<table border="1"><tr><th>A</th><th>B</th><th>C</th><th>D</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	A	B	C	D	S	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1	1	0	0	1	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1	1	1	0	1	0	0	0	0	1	0	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	0	1	1	0	1	1	1	1	1	0	0	1	1	1	1	1
A	B	C	D	S																																																																																		
0	0	0	0	1																																																																																		
0	0	0	1	0																																																																																		
0	0	1	0	1																																																																																		
0	0	1	1	0																																																																																		
0	1	0	0	1																																																																																		
0	1	0	1	0																																																																																		
0	1	1	0	1																																																																																		
0	1	1	1	0																																																																																		
1	0	0	0	0																																																																																		
1	0	0	1	1																																																																																		
1	0	1	0	1																																																																																		
1	0	1	1	1																																																																																		
1	1	0	0	0																																																																																		
1	1	0	1	1																																																																																		
1	1	1	0	0																																																																																		
1	1	1	1	1																																																																																		

Exercícios de fixação

Fazer os enlaces e deduzir as funções booleanas nos mapas de Karnaugh representados abaixo:

a)

B \ A	0	1
0	1	0
1	0	1

b)

B \ A	0	1
0	1	1
1	0	0

c)

B \ A	0	1
0	0	1
1	1	0

d)

B \ C \ A	0	1
00	0	1
01	1	1
11	1	1
10	1	0

e)

C \ B \ A	00	01	11	10
0	1	1	0	1
1	1	0	1	0

f)

B \ C \ A	0	1
00	0	1
01	0	0
11	1	0
10	1	1

g)

C \ D \ B \ A	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	1	1	0	0
10	1	1	0	0

h)

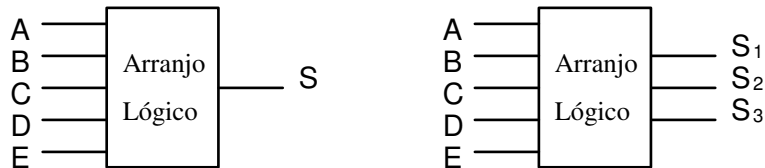
C \ D \ B \ A	00	01	11	10
00	0	1	1	1
01	1	1	1	0
11	1	1	1	0
10	0	1	1	1

i)

C \ D \ B \ A	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	0	0	1	0
10	1	1	0	1

Circuitos Combinacionais

São circuitos digitais que tem como saídas o resultado de funções lógicas aplicadas às suas entradas. Estes circuitos são formados apenas por portas lógicas e podem ter apenas uma saída, ou então várias. Em relação às suas entradas podemos concluir que deverá ter um número maior que 1, pois caso contrário teríamos uma função de apenas uma variável e desta forma estaríamos restritos às funções **igualdade** e **complemento**. Exemplos:



Estudaremos os circuitos combinacionais mais importantes que temos na Eletrônica Digital, mas não podemos nos esquecer que qualquer arranjo lógico que se enquadre na definição feita acima será um circuito combinacional. Acontece, porém que alguns deles são muito usados e sempre aparecem na mesma forma ou então com pequenas variações e por este motivo devem ter um tratamento especial. São eles:

1 - Somadores

Esses circuitos são capazes de executar a **soma aritmética** de dois números em binário. São muito utilizados em circuitos digitais que executam operações aritméticas, pois podemos reduzir todas as operações aritméticas a um conjunto de somas. Analisaremos estes circuitos em duas partes para sermos mais didáticos.

1.1 - Meio Somador

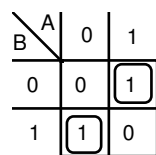
Este arranjo lógico é capaz de "calcular" a soma de dois bits. Para um melhor entendimento analise os quatro possíveis casos da soma de dois bits e veja que esta análise é fundamental para o equacionamento da função.

$$\begin{array}{r}
 + 0 \\
 0 \\
 \hline
 0
 \end{array}
 \quad
 \begin{array}{r}
 + 0 \\
 1 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 + 1 \\
 0 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 + 1 \\
 1 \\
 \hline
 10 \Leftrightarrow (2)_{10}
 \end{array}$$

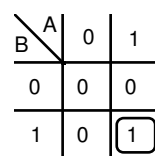
Para montarmos a tabela verdade do problema vamos chamar o primeiro número de A, o segundo de B, o resultado de S e o "vai um" de C (Carry Bit). Observe que nos três primeiros casos o Carry Bit é sempre nulo, mas no último caso ele tem o valor 1.

Uma vez montada a tabela verdade chegamos à função lógica através da resolução dos mapas de Karnaugh correspondentes, um para a saída A e outro para a saída B. E, depois, construímos o circuito com portas lógicas.

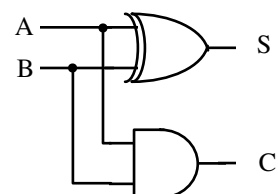
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$$S = A\bar{B} + \bar{A}B = A \oplus B$$



$$S = AB$$



Sabemos que os números em binário podem ter muito mais que 1 Bit. Vamos, então estudar um circuito que seja capaz de "calcular" a soma de número binários com mais de 1 Bit.

1.2 - Somador Completo

Analise como fazemos a soma de dois números onde cada um tenha mais que um Bit:

$$\begin{array}{r}
 \\
 1 \\
 10010111010 \\
 + 11001011100 \\
 \hline
 101100010110
 \end{array}$$

Usando o mesmo processo discutido no circuito do meio somador, podemos equacionar e chegar a um circuito capaz de "calcular" a soma aritmética de dois número quaisquer em binário, mas na verdade cada circuito será responsável pelo "cálculo" da soma de uma coluna. No caso do exemplo anterior, precisaremos de onze circuitos já que cada parcela da soma é composta por números de onze bits.

Antes de iniciarmos o projeto vamos definir o nome das variáveis que utilizaremos:

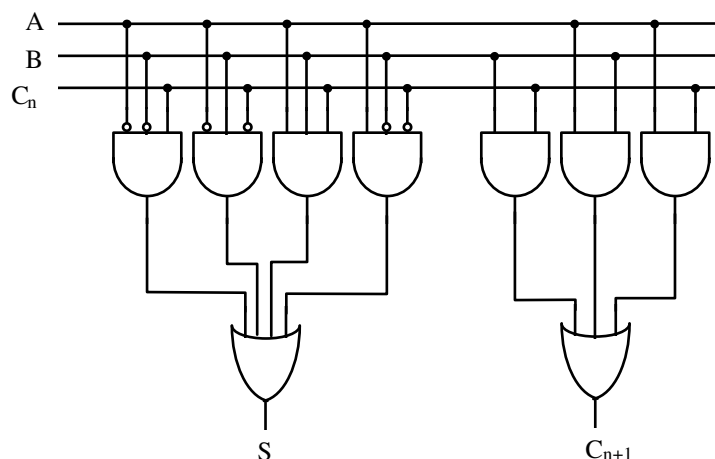
A = Bit do primeiro número C_n = "Veio um"
 B = Bit do segundo número C_{n+1} = "Vai um"

C_n	A	B	S	C_{n+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

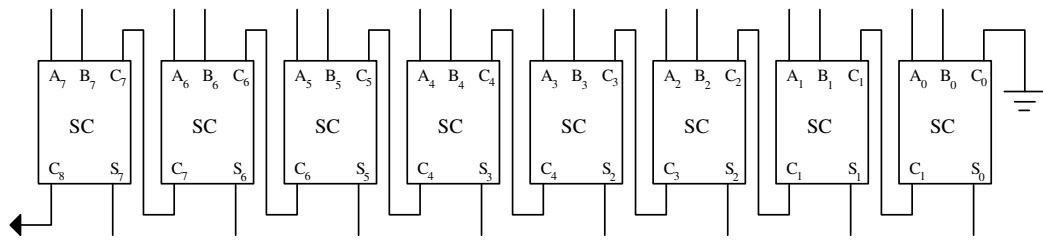
C_n	A	B	S	C_{n+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = C_n \bar{A} \bar{B} + \bar{C}_n \bar{A} B + C_n A B + \bar{C}_n A \bar{B}$$

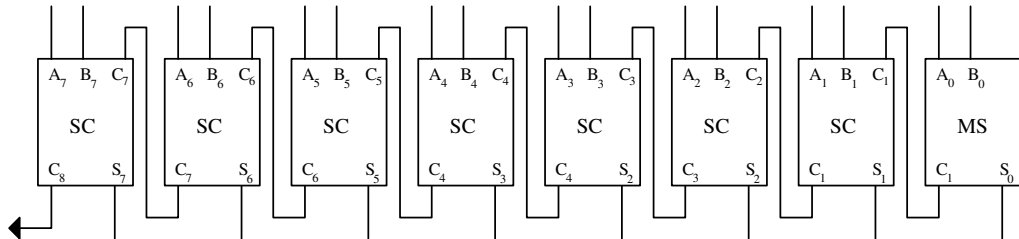
$$C_{n+1} = C_n B + AB + C_n A$$



Como podemos observar, o número de portas lógicas necessárias para a construção de um **Somador Completo** é muito grande ainda mais quando lembramos que este circuito é capaz de somar apenas dois bits. Para somarmos dois números de 8 bits cada, por exemplo, precisaremos de 8 circuitos iguais a este e isto torna inviável o desenho do circuito completo. Uma saída para este problema de representação é usarmos representações simplificadas como sugere o exemplo abaixo:



Observe que o primeiro bloco da direita tem a sua entrada C_0 aterrada, já que em uma soma de duas parcelas nunca teremos o "veio um" na primeira coluna. Sendo assim poderíamos substituir este bloco pelo bloco de um **Meio Somador** conforme mostra o exemplo seguinte:



Exercícios

Calcular a soma dos número em binário indicadas abaixo, indicando ao lado os valores correspondentes em decimal:

a)
$$\begin{array}{r} 1001 \\ +0101 \\ \hline \end{array}$$

b)
$$\begin{array}{r} 0110 \\ +0011 \\ \hline \end{array}$$

c)
$$\begin{array}{r} 1000 \\ +0111 \\ \hline \end{array}$$

d)
$$\begin{array}{r} 1110 \\ +0111 \\ \hline \end{array}$$

2 - Decodificador

Decodificador é um circuito combinacional que ativa uma saída diferente para cada código diferente colocado em suas entradas. Um exemplo de tabela verdade e projeto de circuito esta logo abaixo:

A	B	S_0	S_1	S_2	S_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

B \ A	0	1
0	1	0
1	0	0

$$S_0 = \bar{A}\bar{B}$$

B \ A	0	1
0	0	1
1	0	0

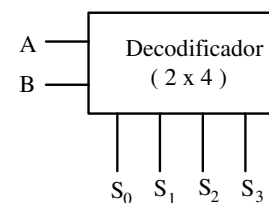
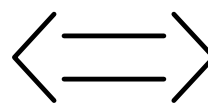
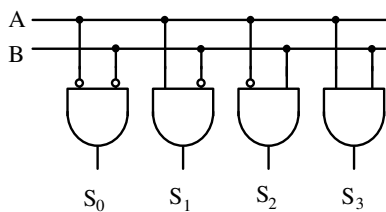
$$S_1 = A\bar{B}$$

B \ A	0	1
0	0	0
1	1	0

$$S_2 = \bar{A}B$$

B \ A	0	1
0	0	0
1	0	1

$$S_3 = AB$$



3 - Codificador

Este circuito executa a função **inversa** a do codificador ou seja produz um código diferente em suas saídas para cada entrada diferente ativada. Podemos analisar o projeto do circuito através de uma tabela verdade construída a partir da sua definição.

I_3	I_2	I_1	I_0	A	B
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

A tabela verdade pode parecer um pouco estranha pois apesar de ter quatro variáveis de entrada não tem a esperadas dezesseis linhas. O problema é que as quatro entradas só podem ser ativadas uma de cada vez e com isso temos que eliminar todas as outras combinações possíveis para elas, mas para resolvermos o circuito através dos mapas de Karnaugh teremos que ter todas as linhas. Vamos então introduzir o conceito de **irrelevância**:

Em alguns casos de circuitos combinacionais teremos situações que nunca acontecem e portanto não nos importaremos com os valores das entradas destes casos. Dizemos então que são casos **irrelevantes**, ou seja, tanto faz as entradas terem nível lógico 1 ou nível lógico zero. A grande vantagem desta situação é que para resolvermos os mapas de Karnaugh destes circuitos podemos considerar os níveis lógicos como 1 ou como 0 levando em consideração apenas nos for mais conveniente para conseguirmos um maior enlace do mapa sem nos esquecer das regras que regem esses enlaces. Analise então como fica o projeto deste codificador:

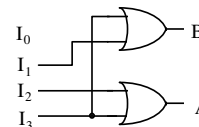
I_3	I_2	I_1	I_0	A	B
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	X	X
0	1	0	0	1	0
0	1	0	1	X	X
0	1	1	0	X	X
0	1	1	1	X	X
1	0	0	0	1	1
1	0	0	1	X	X
1	0	1	0	X	X
1	0	1	1	X	X
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

$I_2 \backslash I_3 \backslash I_1 \backslash I_0$	00	01	11	10
00	X	1	X	1
01	0	X	X	X
11	X	X	X	X
10	0	X	X	X

$$A = I_2 + I_3$$

$I_2 \backslash I_3 \backslash I_1 \backslash I_0$	00	01	11	10
00	X	0	X	1
01	0	X	X	X
11	X	X	X	X
10	1	X	X	X

$$B = I_1 + I_3$$



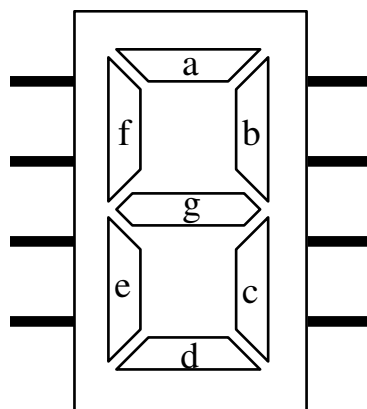
Observe que a entrada I_0 não é conectada no circuito propriamente dito e que pela lógica isto está certo, pois quando esta estiver ativada devemos ter nas saídas $A = 0$ e $B = 0$.

Um exemplo de aplicação para os codificadores e decodificadores são os teclados de computadores. Você já deve ter notado que um teclado deste tipo tem normalmente 105 teclas, mas o fio que os conecta com o gabinete da CPU é muito fino para conter 105 fios. Na verdade as teclas são codificadas através de um codificador para economizarmos em fios. Veja que um codificador com 7 saídas pode ter 128 entradas. Isso significa que podemos transmitir por uma via de 7 fios 128 valores diferentes, onde cada valor representa uma tecla. O circuito responsável pela codificação de teclados dos computadores atuais é mais complexo que este que estudamos, mas o princípio de funcionamento é o mesmo.

4 - Transcodificador

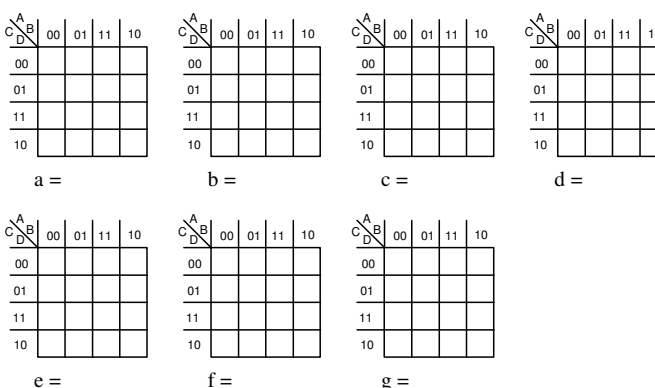
É o circuito combinacional que é capaz de transformar um código, em binário, em outro, também em binário. Como exemplo deste circuito vamos analisar o **transcodificador para display de sete segmentos** que transforma uma numeração em binário nos níveis lógicos necessários para que em um display de sete

segmentos tenhamos aceso o algarismo em decimal correspondente. Vamos primeiro analisar o display de sete segmentos:



Podemos encontrar este tipo de display com duas denominações diferentes: **anodo comum** e **catodo comum**. Isto se deve a fato de serem construídos a partir de **LEDs** e como os leds são diodos emissores de luz, também tem seus terminais denominados de anodo e catodo. Porém para simplificar as ligações dos 7 leds nesses displays os anodos ou os catodos são todos interligados. Desta forma, se o display for do tipo **catodo comum** devemos ligar este terminal ao **terra** (polo negativo da fonte) e podemos acender cada segmento aplicando um nível lógico 1 no terminal correspondente. Porém se o display for do tipo **anodo comum**, devemos ligar este terminal a **Vcc** (polo positivo da fonte) e para acender cada segmento devemos aplicar nível lógico 0 nos terminais correspondentes. Para efeito de exemplo, vamos considerar que o nosso display é do tipo **catodo comum** e portanto precisaremos construi a tabela verdade considerando que o segmento vai acender quando colocarmos nível lógico 1 em cada terminal. Temos então a seguinte tabela verdade:

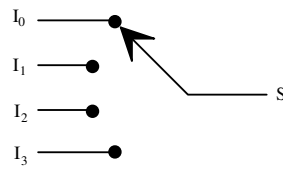
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							



Encontramos no mercado de Eletrônica este transcodificador pronto em um único circuito integrado, o que nos facilita muito montagem de circuitos digitais que exigem este dispositivo. Porém ha um vício em se chamar este dispositivo de **decodificador para display de 7 segmentos** mas o seu nome verdadeiro é **transcodificador para display de 7 segmentos**, pois transforma o código binário no código necessário para formar no display o algarismo correspondente em decimal. Podemos encontrar também no mercado o transcodificador para display de 7 segmentos para algarismos **hexadecimais** (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F). Como exercício, projete um transcodificador capaz de transformar o código em binário em algarismos hexadecimais em um display de 7 segmentos e desenhe o circuito com portas lógicas.

5 - Multiplexador

Para analisarmos este circuito vamos usar como exemplo uma chave mecânica de 1 polo e 4 posições. Analise o desenho abaixo:



Com esta chave podemos conectar 4 entradas (I_0 , I_1 , I_2 e I_3) com um única saída (S) de acordo com a seleção que fizermos girando o seu eixo. Este circuito está muito presente em nosso cotidiano, basta repararmos. Como exemplo podemos citar a chave seletora de toca-discos, rádio, cassete, CD, etc em aparelhos de som.

O multiplexador digital funciona da mesma forma e função, porem opera apenas com **sinais digitais** e a sua seleção também é feita digitalmente. Um exemplo de circuito multiplexador digital está desenhado logo abaixo:

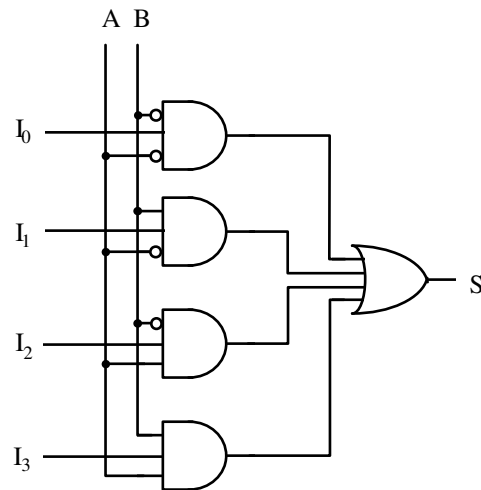
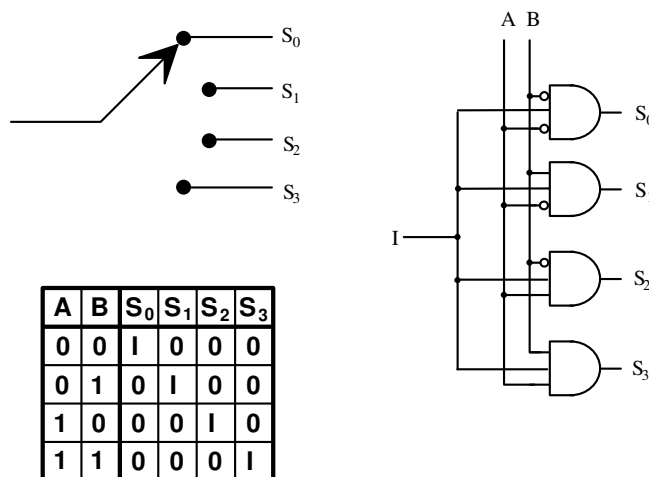


Tabela Verdade

A	B	S
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

6 - Demultiplexador

Este circuito tem a função inversa à do circuito anterior, ou seja, pode conectar uma única entrada à várias saídas de acordo com a seleção feita. A chave mecânica nos servirá novamente de exemplo. Analise o circuito abaixo onde temos uma chave mecânica e também o circuito digital que executa a função semelhante a esta chave:



A	B	S_0	S_1	S_2	S_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Chegamos, então ao final da lógica combinacional. Mesmo que não tenhamos estudado todos os circuitos combinacionais, o que seria impossível e fugiria ao objetivo da matéria Eletrônica Digital I, temos a base fundamental para o projeto e estudo de qualquer um desses circuitos, basta seguirmos os procedimentos analisados até aqui, ou seja:

- Definir função do circuito através de sentenças que possam ser transformadas em equações Booleanas e minimiza-las através da Álgebra de Boole, ou então

- Montar a tabela verdade e deduzir as equações através dos Mapas de Karnaugh

O próximo capítulo desta apostila tratará da análise e projetos de circuitos **Seqüenciais**, e por algum tempo nos afastaremos da álgebra de Boole. Porém ao final deste estudo os Mapas de Karnaugh e as funções booleanas terão fundamental importância para os projetos que passarão a ser muito mais interessantes e com aplicações práticas imediatas. Além disso tudo, teremos um visão muito mais ampla e completa sobre o funcionamento de diversos aparelhos comumente encontrados no mercado.

Circuitos Seqüenciais – Elementos Básicos

Introdução

Estudamos até agora circuitos digitais que executam funções lógicas, mas são incapazes de armazenar informações. Neste capítulo iniciamos o estudo sobre os circuitos que além de armazenarem informações também poderão executar seqüências pré-determinadas. Veja que para que um dispositivo possa executar uma seqüência é necessário que ele consiga também armazenar informações. Caso contrário não poderíamos estar falando em seqüência pré-determinada.

Esses circuitos estão bastante presentes em nosso cotidiano. Um semáforo, por exemplo é um circuito seqüencial pois acende as suas lâmpadas sempre na mesma seqüência. A grande maioria dos luminosos de lâmpadas neon também são exemplos de circuitos seqüenciais. Indo mais a fundo, o computador também é um circuito seqüencial. Veja como ele pode executar várias vezes a mesma função. É lógico que o computador tem em seu interior outros tipos de circuitos, mas a sua base é um circuito seqüencial.

Iniciaremos o nosso estudo analisando os elementos básicos dos circuitos seqüenciais e no próximo capítulo estudaremos os circuitos seqüenciais que são construído a partir destes elementos.

1) Latch RS Assíncrono

Este é o elemento mais simples e realmente básico da Eletrônica Digital Seqüencial. Todos os outros circuitos seqüenciais são baseados ou formados por associações deste dispositivo. A sua função é armazenar o valor de 1 bit por um tempo indeterminado e obviamente poderá armazenar apenas dois valores (um de cada vez) que são o nível lógico 0 e o nível lógico 1.

A palavra **Assíncrono** que faz parte de seu nome, indica que ele não tem sincronismo com nada, isto é basta aplicarmos os sinais de comandos que ele armazena um nível lógico imediatamente. Mais adiante estudaremos **Latch RS Síncrono** que terá um sinal fazendo o sincronismo do seu funcionamento.

O circuito do **Latch RS Assíncrono** é muito simples e está representado abaixo, seguido de uma análise do seu funcionamento:

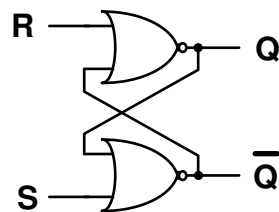


Tabela Verdade da Porta NOR

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

onde:

R = Reset (provoca o armazenamento do nível lógico zero)

S = Set (provoca o armazenamento no nível lógico um)

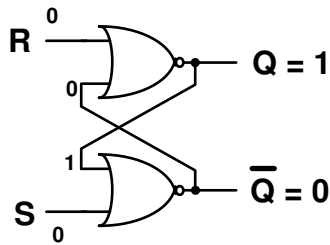
Q = Saída principal

\bar{Q} = Saída auxiliar (complemento de **Q**)

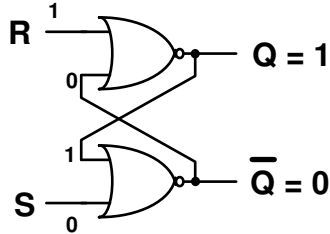
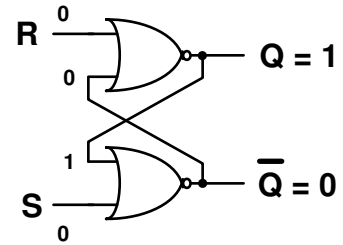
Para analisarmos o funcionamento do Latch RS Assíncrono, tomemos como exemplo o funcionamento de um interruptor de lâmpadas:

- Se não mexermos nele, a lâmpada permanece no estado em que esta (acesa ou apagada).
- Se desligarmos, a lâmpada fica apagada por um tempo indeterminado.
- Se ligarmos, a lâmpada fica acesa por um tempo indeterminado.
- Não podemos comanda-lo desejando que a lâmpada fique acesa e apagada ao mesmo tempo.

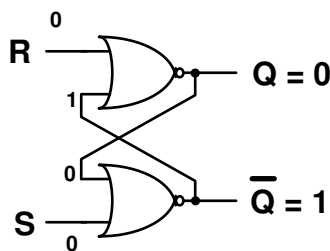
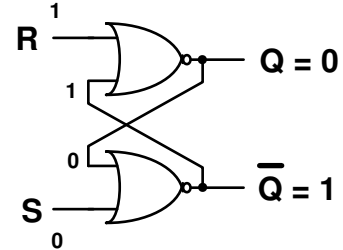
O funcionamento do Latch é exatamente o mesmo, basta substituímos a palavra acender para SET e apagar para RESET. Analise o funcionamento diretamente pelo circuito considerando que inicialmente a saída principal (Q) está em **nível lógico um** e conseqüentemente a saída complementar está em **nível lógico zero**:



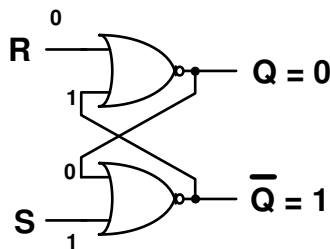
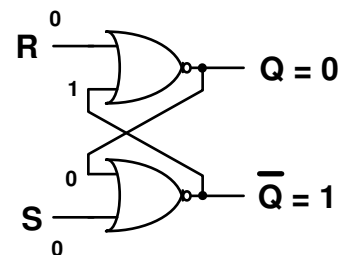
Se $R=0$ e $S=0$ então o circuito permanece no mesmo estado



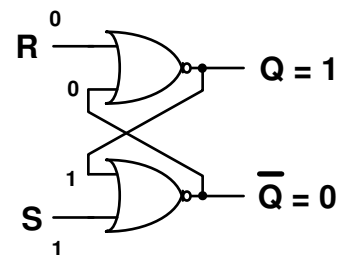
Se $R=1$ e $S=0$ então o circuito será RESETADO



Se $R=0$ e $S=0$, novamente, então o circuito permanece, novamente



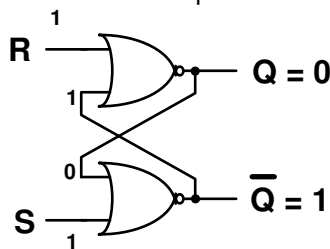
Se $R=0$ e $S=1$ então o circuito será SETADO



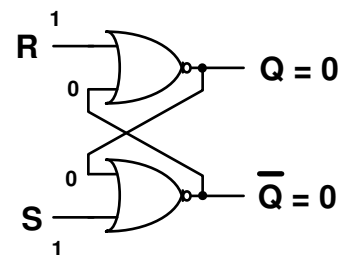
Obs.:

- Muito cuidado para não se atrapalhar com esta análise nos circuitos da esquerda. Veja que no momento em que alteramos um sinal nas entradas SET ou RESET, é a porta que recebeu esta alteração que muda o seu sinal, porém a representação deste fato é impraticável no papel.

- Se colocarmos nível lógico 1 nas duas entradas, teremos nível lógico 0 nas duas saídas e isso é um erro lógico pois as saídas são complementares. Portanto nunca podemos fazer isso. Lembre-se que é impossível acender e apagar uma lâmpada ao mesmo tempo.



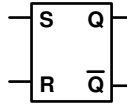
Se $R=1$ e $S=1$ então o circuito apresenta um erro lógico



Na tabela verdade de um Latch RS Assíncrono temos que representar os níveis lógicos de forma que saibamos que temos uma situação atual e iremos passar para uma próxima situação assim que ativarmos uma das entradas. Indicaremos a situação atual por Q_n e a próxima por Q_{n+1} :

R	S	Q_{n+1}	\bar{Q}_{n+1}	
0	0	Q_n	\bar{Q}_n	Permanece
0	1	1	0	Set
1	0	0	1	Reset
1	1	0	0	Não usado (erro lógico)

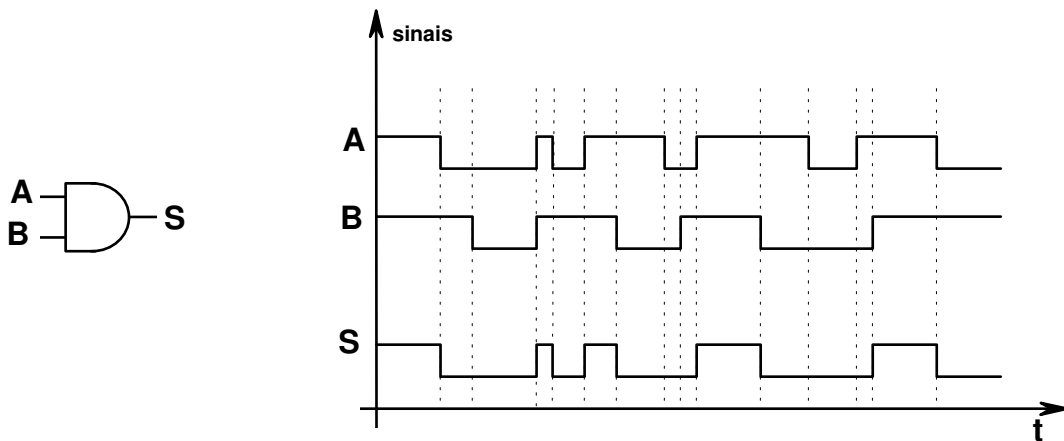
Se representássemos o circuito inteiro do latch em circuitos mais complexos, teríamos desenhos muito grandes e por esse motivo temos uma representação esquemática que facilita muito o nosso trabalho:



Observe que as posições das saídas estão invertidas em relação às entradas se compararmos com a representação lógica.

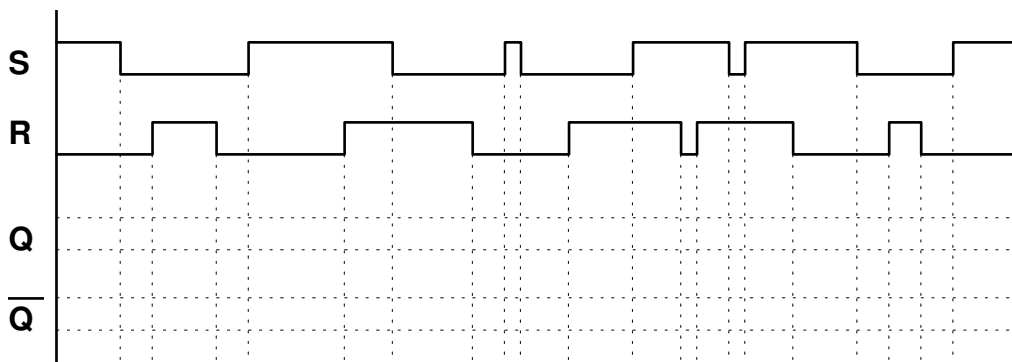
Para fazermos os exercícios vamos ter que aprender uma outra forma de representação de sinais lógicos: a Carta de Tempos. Ela é uma sobreposição de gráficos que mostram os níveis lógicos todos sincronizados em uma única linha de tempo:

Carta de tempos de uma Porta AND



Exercício

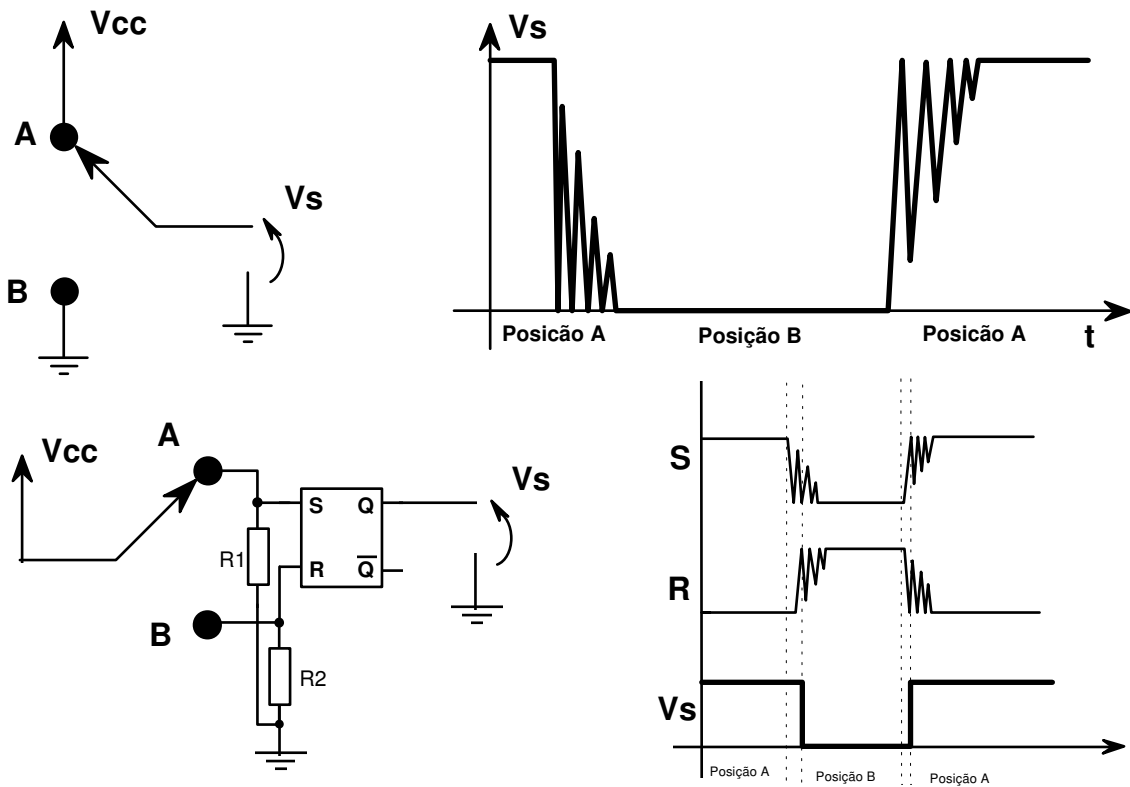
Completar a carta de tempos do Latch RS Assíncrono:



Temos diversas aplicações para os Latches RS Assíncronos. De imediato vamos ver apenas um exemplo:

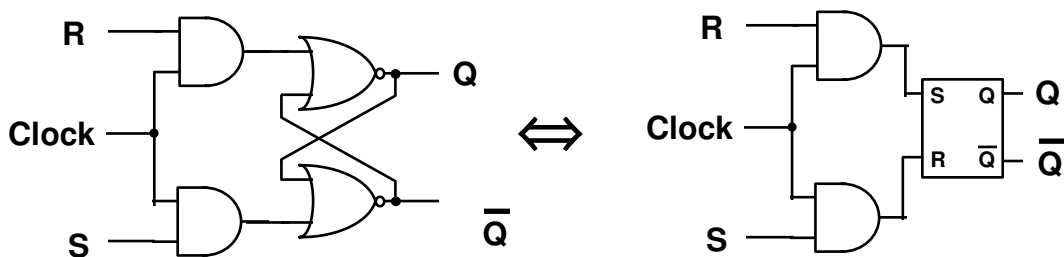
Chave anti rebote

Uma chave elétrica sem apresenta ruídos quando fazemos a comutação devido ao atrito entre os contatos. Este ruído é extremamente indesejável quando se trata de circuitos digitais ou até mesmo circuitos de audio e outros quaisquer. Associando um Latch RS Assíncrono como mostra o circuito abaixo, eliminamos completamente este problema:

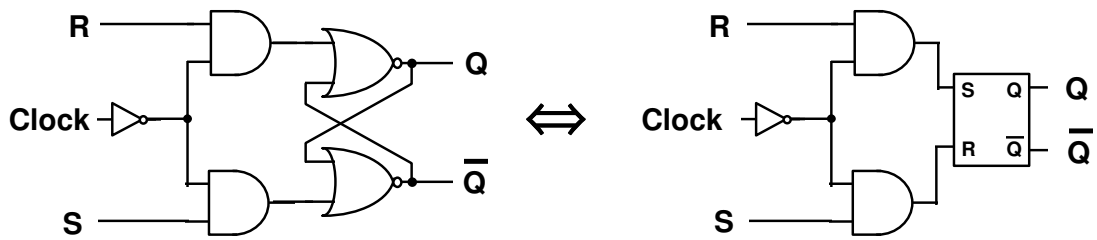


2) Latch RS Síncrono

Como foi comentado anteriormente teríamos circuitos síncronos. Este é o exemplo de um deles. Precisamos de circuitos síncronos principalmente quando temos vários deles operando em conjunto. Para que todos mudem de estado simultaneamente temos que ter um sinal de sincronismo. Obviamente o dispositivo terá mais uma entrada para este sinal que se chamará **Entrada de Clock**. Observe o circuito para entender o seu funcionamento:

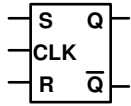


Os sinais Set e Reset somente atuarão no circuito se o sinal de clock estiver em nível lógico 1, caso contrário as portas AND garantem nível lógico em suas saídas e o latch permanece no mesmo estado. O nome completo deste circuito é **Latch RS Síncrono Sensível ao Nível Lógico 1**. Temos também um outro tipo deste circuito que é o **Latch RS Síncrono Sensível ao Nível Lógico 0**:

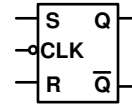


A única diferença entre esses dois circuitos é que o segundo tem um inversor na entrada de clock e isso faz com que ele só esteja habilitado para mudar de estado quando esta entrada de clock estiver em nível lógico zero.

Temos também representações esquemáticas simplificadas para os dois circuitos:



**Latch RS Síncrono
Sensível ao Nível Lógico 1**

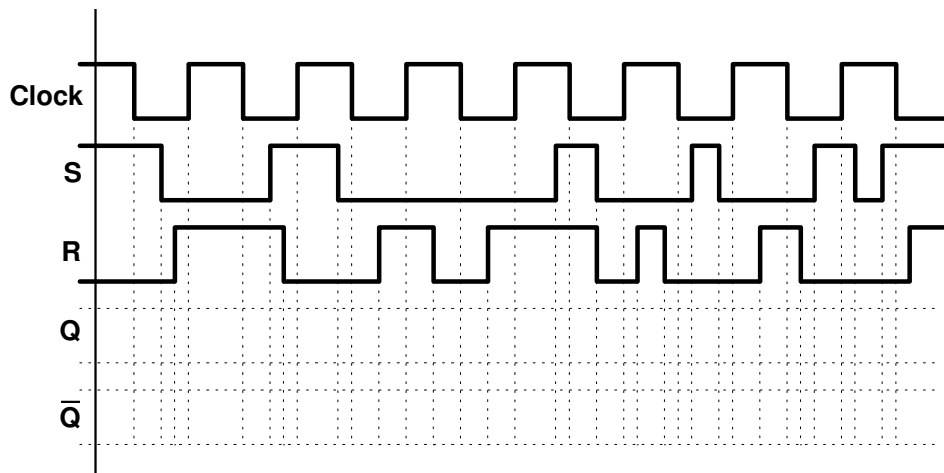


**Latch RS Síncrono
Sensível ao Nível Lógico 0**

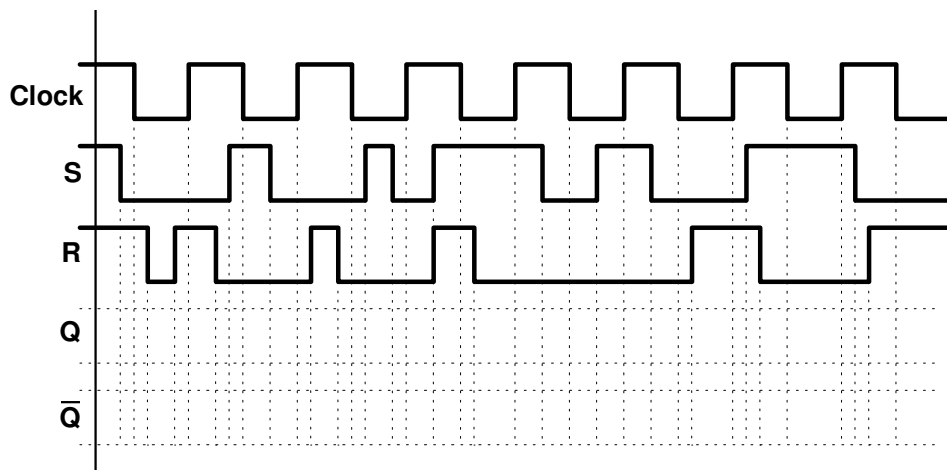
Exercícios:

Completar as cartas de tempo para os circuitos:

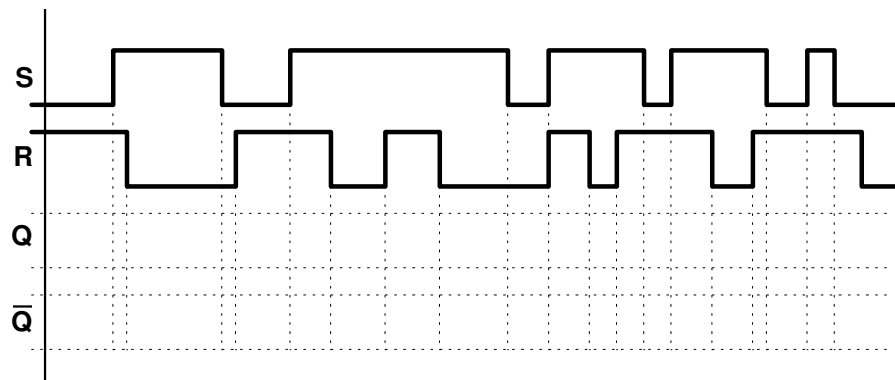
a) Latch RS Síncrono Sensível ao Nível Lógico 1



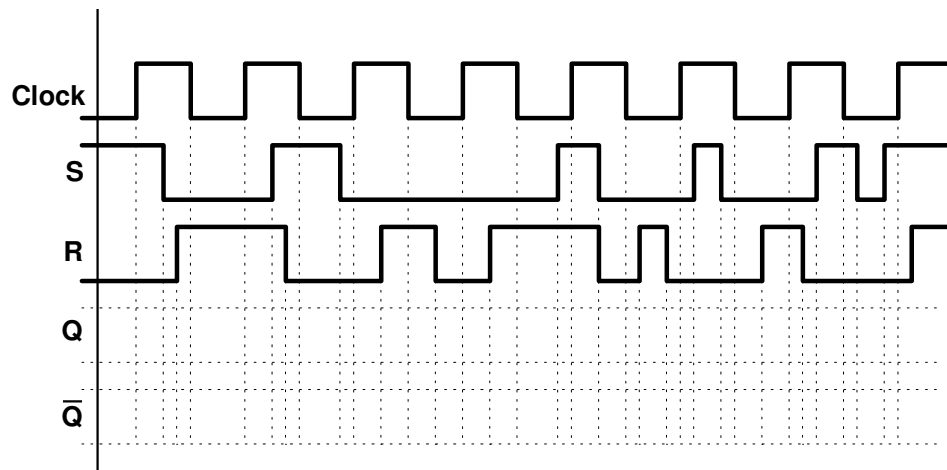
b) Latch RS Síncrono Sensível ao Nível Lógico 0



c) Latch RS Assíncrono



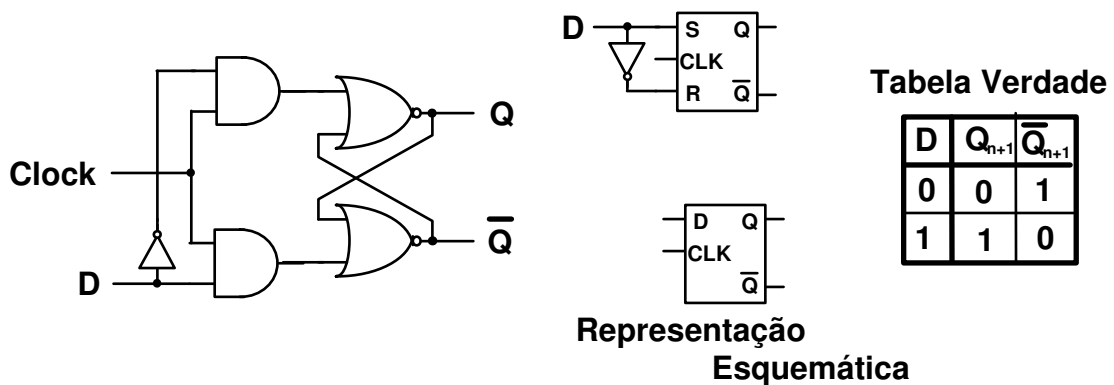
d) Latch RS Síncrono Sensível ao Nível Lógico 1



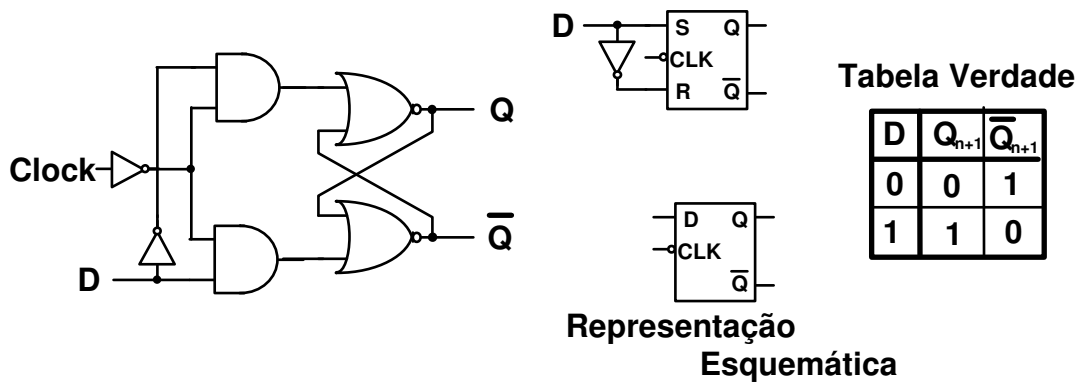
3) Latch Tipo D

Seria muito conveniente se pudéssemos mudar o estado do latch usando apenas um sinal de controle. Para conseguirmos isso basta associarmos um inversor entre as entradas do Latch RS Síncrono, conforme mostra o esquema abaixo. O circuito com esta configuração será chamado de Latch tipo D e também é apresentado em duas versões, conforme a sensibilidade do sinal de Clock.

Latch tipo D Sensível ao Nível Lógico 1



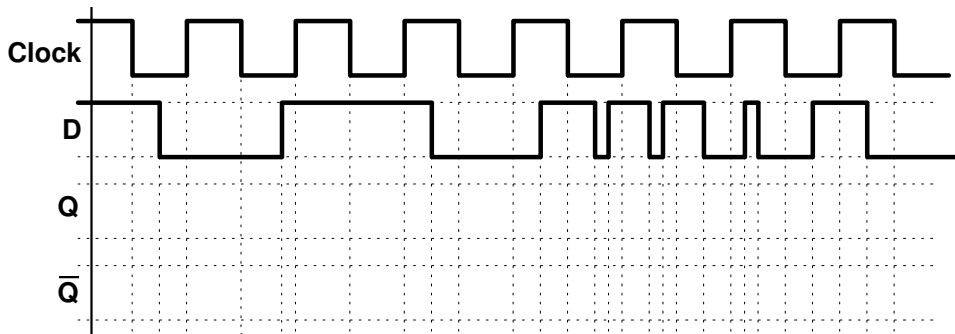
Latch tipo D Sensível ao Nível Lógico 0



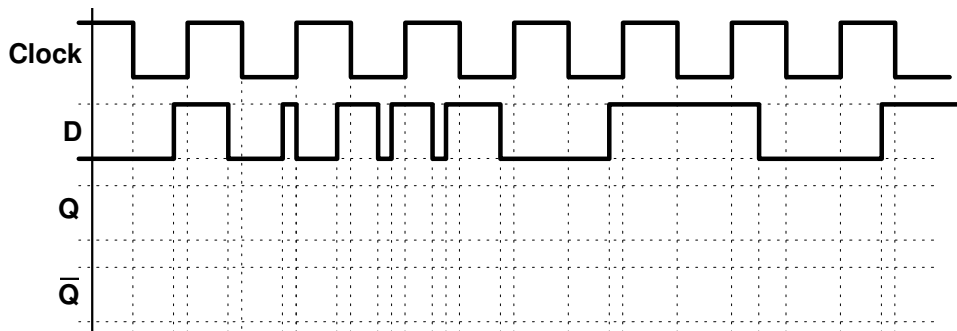
Exercícios

Completar as cartas de tempo dos dispositivos indicados:

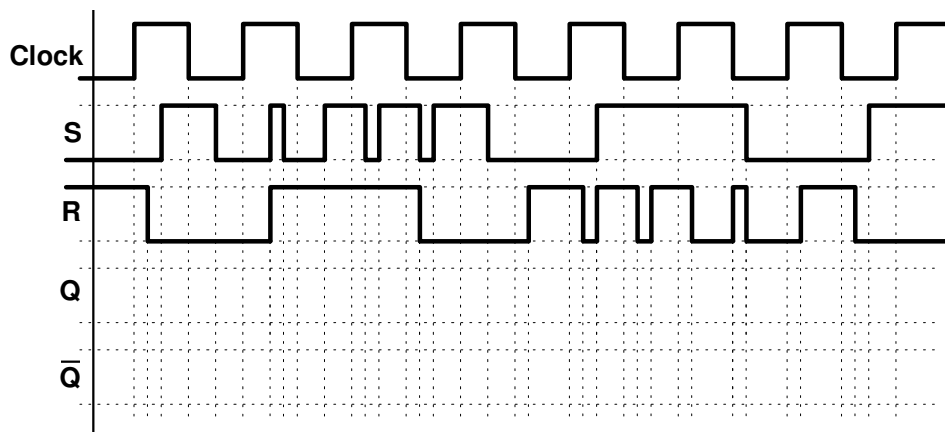
1) Latch tipo D Sensível ao Nível Lógico 1



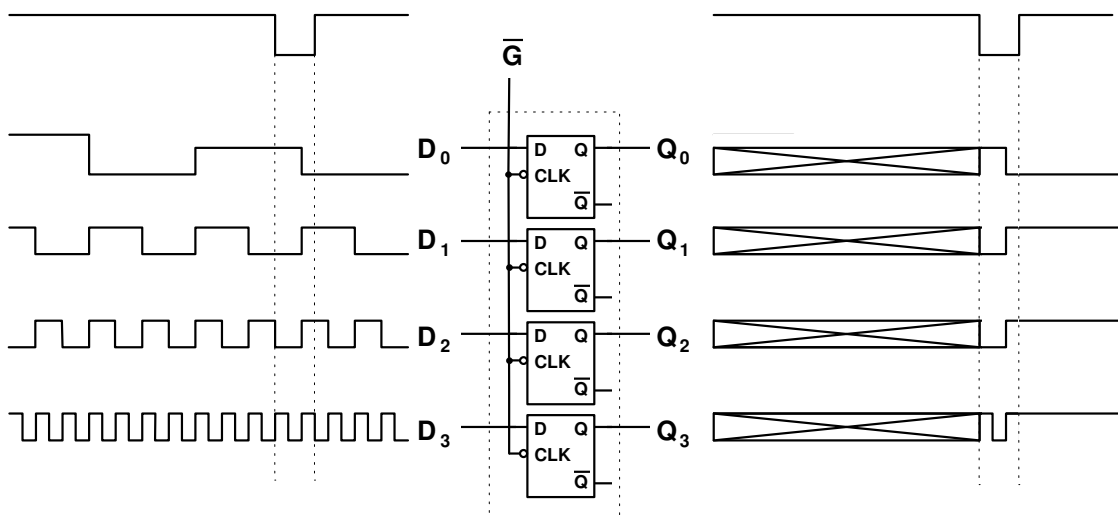
2) Latch tipo D sensível ao Nível Lógico 0



3) Latch RS Sensível ao Nível Lógico 0



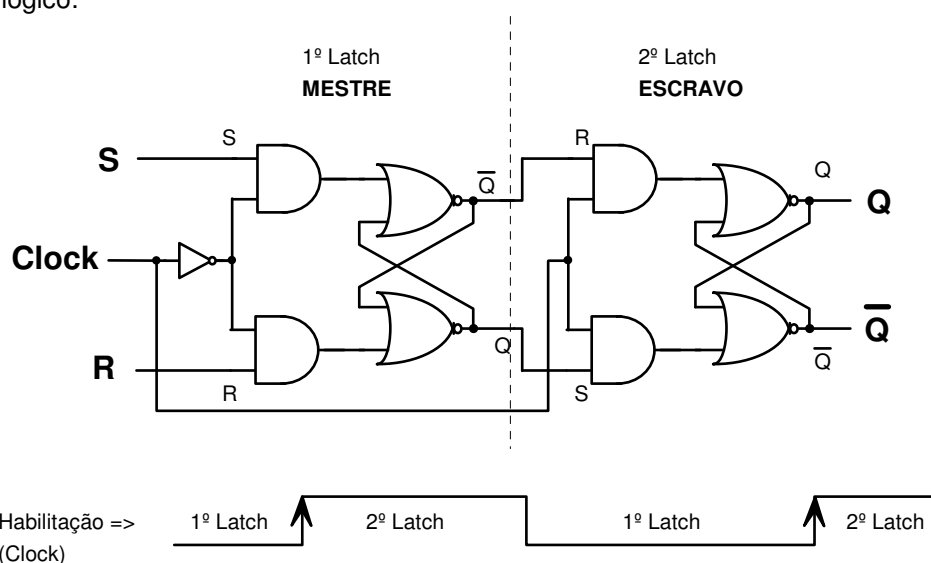
Alguns circuitos formados por vários **Latches tipo D** com todos os clocks interligados também é chamado de Latch e tem a função de armazenar o valor de um **Byte**:



O desenho acima mostra claramente que apesar do latch armazenar um byte, mantém as suas saídas alterando os níveis durante todo tempo em que há a habilitação do sinal **G**. Este fato pode causar problemas em circuitos de grande porte. Determinados circuitos não funcionam por causa deste fato e outros funcionam muito lentos, pois precisamos um longo tempo de estabilização de um byte para outro. A solução deste problema aconteceu com a associação mestre-escravo de dois latches, sendo que um deles é sensível um nível lógico e o outro sensível ao nível lógico complementar. Este circuito se chama **Flip-Flop**.

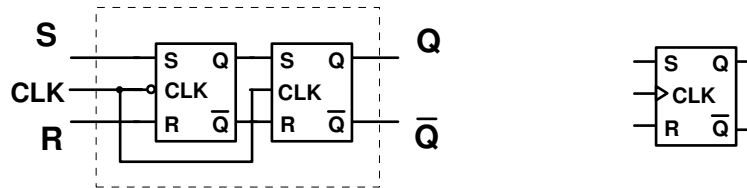
4) Flip-Flop RS

Circuito lógico:

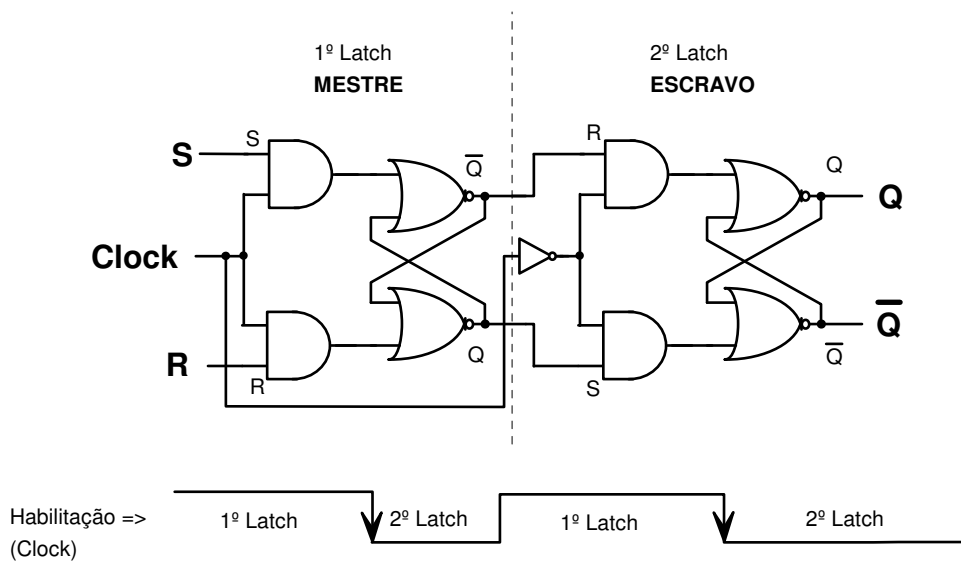


Como o segundo latch é escravo do primeiro, teremos o armazenamento de um novo dado somente quando houver a transição da habilitação (sinal de **Clock**) de nível lógico zero para nível lógico um. Esta transição de nível do sinal do clock se chama **Borda de Subida**. Observe também que o segundo latch com a posição das entradas Set e Reset invertidas em relação ao primeiro latch e, conseqüentemente o circuito final tem as posições das saídas invertidas em relação às entradas, se compararmos com a representação lógica de um latch.

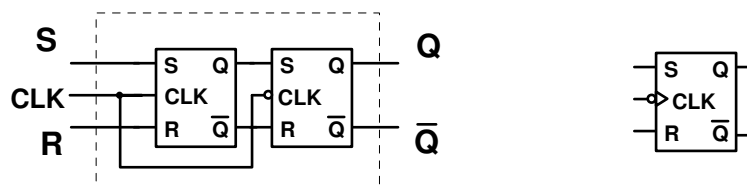
O nome completo deste circuito é **Flip-Flop RS Sensível à Borda de Subida** e também tem representação esquemática simplificada. Veja abaixo uma evolução nos esquemas até chegarmos à representação final:



Temos também a versão complementar de habilitação, ou seja, o **Flip-Flop RS Sensível à Borda de Descida**.



E a evolução de seu circuito lógico para a representação esquemática simplificada está logo abaixo:

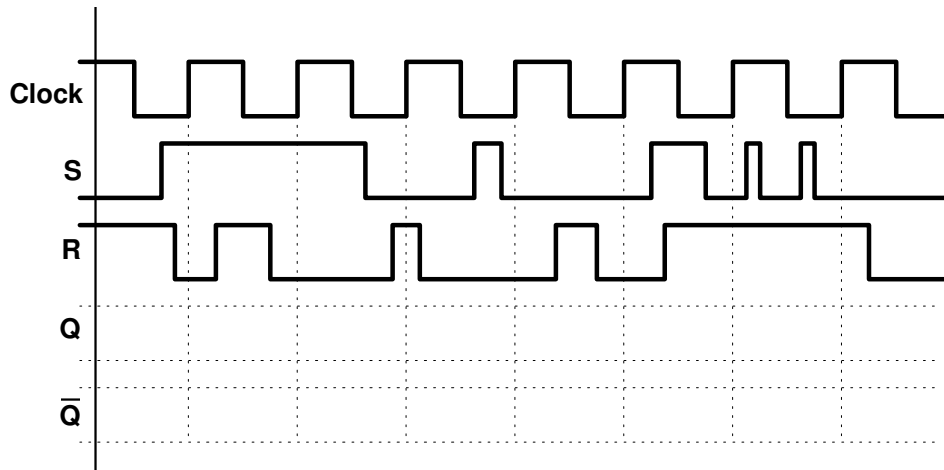


A tabela verdade destes circuitos é a mesma do Latch RS Sensível ao nível. A única diferença entre eles é que um é sensível ao nível e o outro sensível à borda do sinal de clock.

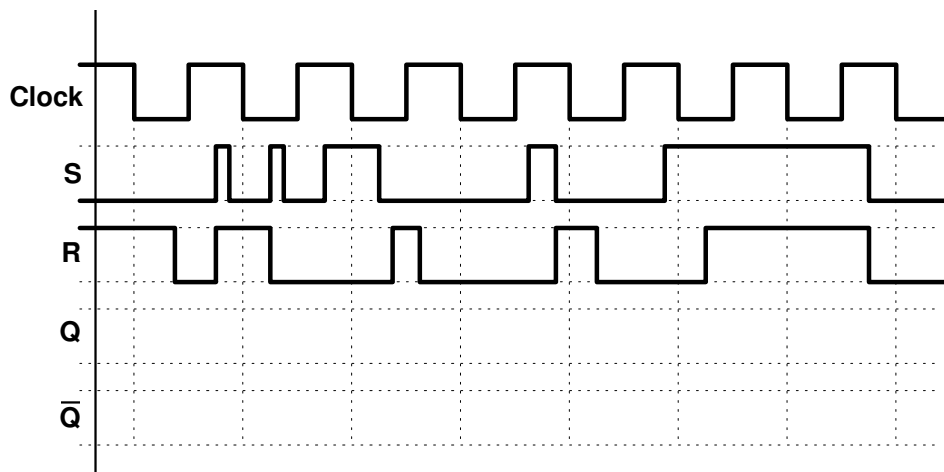
Exercícios:

Complete as cartas de tempo dos circuitos indicados:

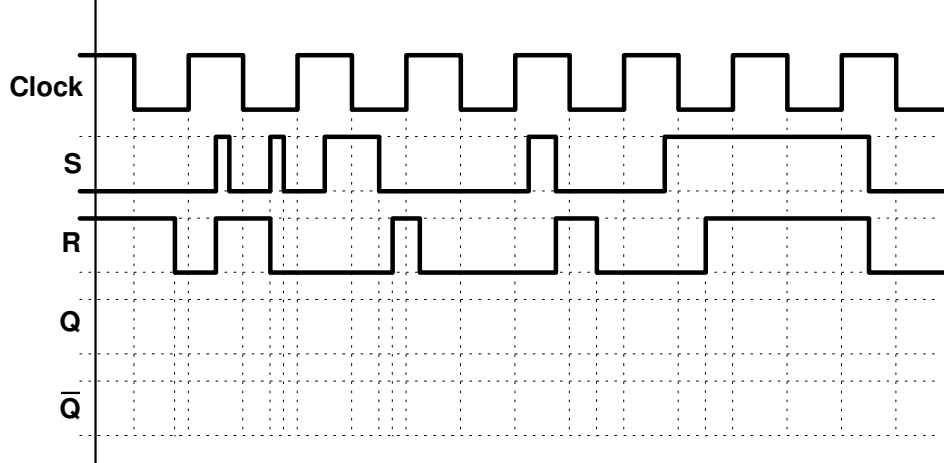
1) Flip-Flop RS Sensível às Borda de Subida



2) Flip-Flop RS Sensível à borda de descida:



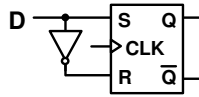
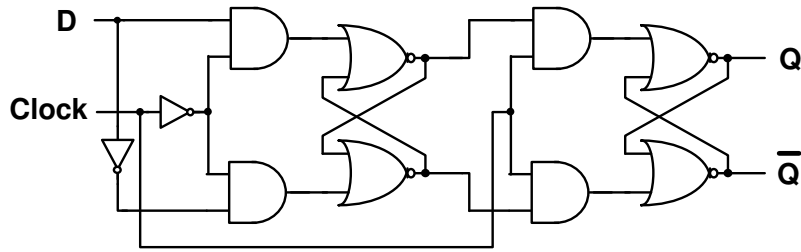
3) Latch RS Sensível ao Nível Lógico 0



5) Flip-Flop tipo D

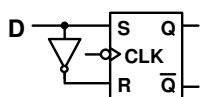
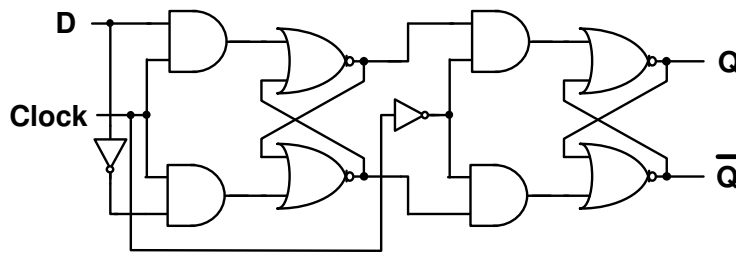
A possibilidade de podermos armazenar um bit em um Flip-Flop usando apenas um sinal de comando é tão importante o quanto era para o Latch. Usando, então, a mesma técnica podemos fazer um circuito comandado desta forma. Analise o esquema abaixo:

Flip-Flop tipo D Sensível à Borda de Subida



Representação Esquemática

Flip-Flop tipo D Sensível à Borda de Descida

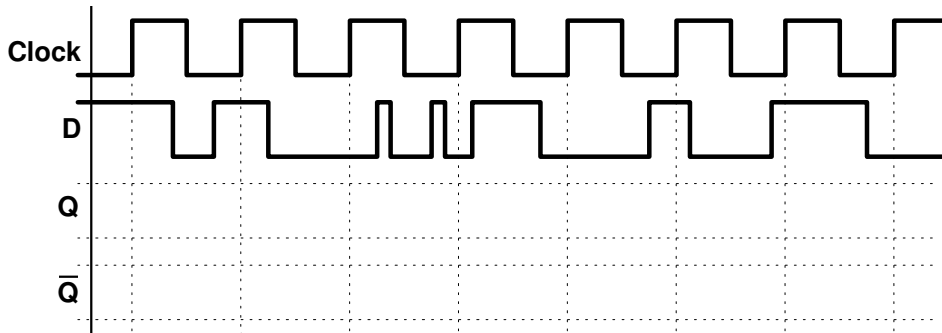


Representação Esquemática

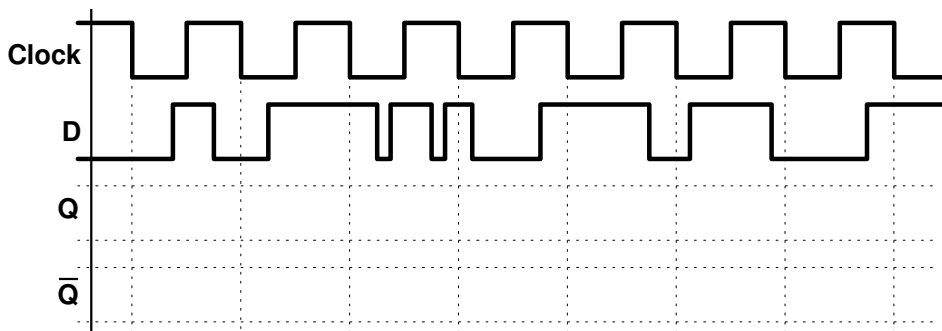
Exercícios:

Completar as cartas de tempo dos circuitos indicados:

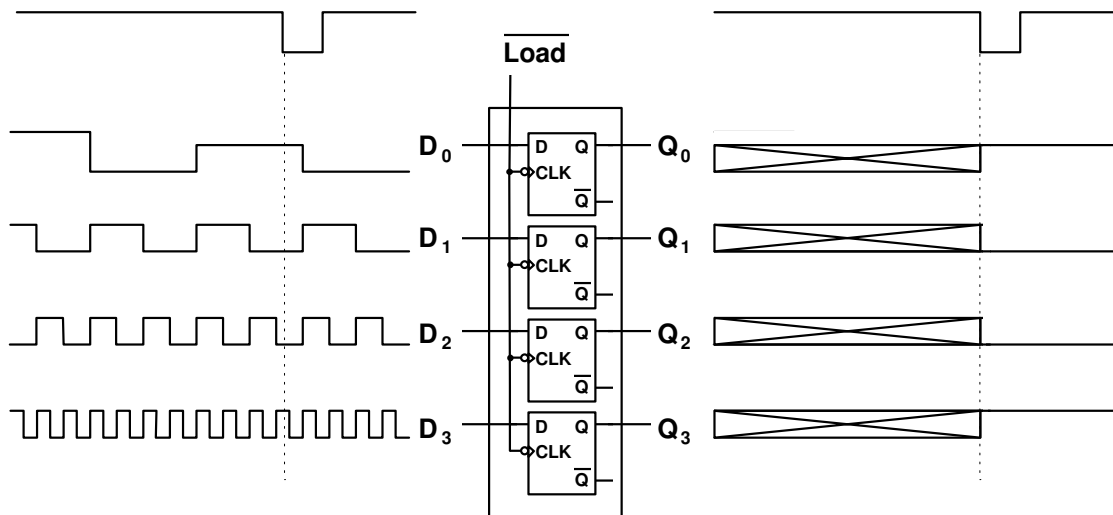
1) Flip-Flop tipo D sensível à Borda de Subida



2) Flip-Flop tipo D sensível à Borda de Descida



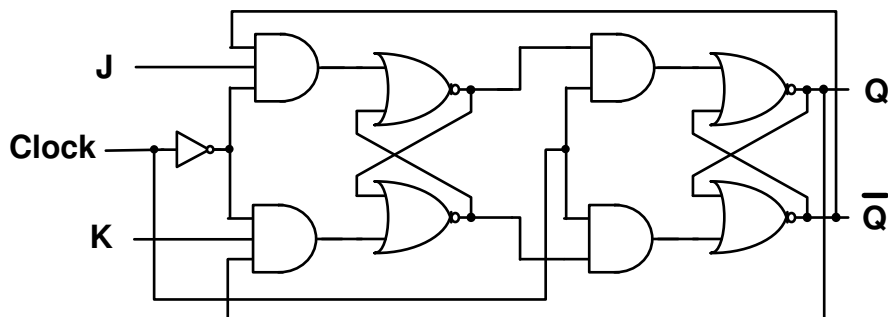
Assim como temos Latches de n bits, temos Flip-Flops de n bits, mas estes são chamados de registradores. A sua vantagem sobre o latch é a precisão e rapidez de armazenamento em relação ao sinal do clock, pondo fim nos problemas que o latch apresentava. Veja o exemplo abaixo:



É bom lembrar que em determinadas ocasiões a morosidade de armazenamento do latch não importa e sim o fato de instantaneamente estar apresentando os dados em sua saída e ter ainda a capacidade de armazená-los. Cada circuito terá uma aplicação mais adequada e portanto não existe um melhor ou pior. Tudo depende da utilização de cada um no local correto.

6) Flip-Flop JK

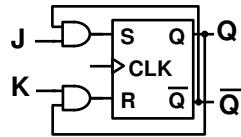
As palavras SET e RESET significam, em relação a latches e Flip-Flops, armazenar nível lógico 1 e armazenar nível lógico 0, respectivamente. Este fato nos dá margem a concluir que jamais poderíamos ativar os dois sinais simultaneamente pois não existe um nível lógico que seja 0 e 1 ao mesmo tempo. O Flip-Flop JK tem para o J a mesma função do SET e para o K a mesma do RESET e permite que ativemos os dois simultaneamente e se fizermos isso ele complementa o nível lógico que está armazenando. Analise o esquema abaixo:



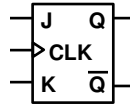
Temos a seguir a tabela verdade deste dispositivo que é muito parecida com a dos latches e Flip-Flops RS. A única diferença é que agora temos a possibilidade de usarmos todas as combinações para os sinais de comando.

K	J	Q_{n+1}	\bar{Q}_{n+1}	
0	0	Q_n	\bar{Q}_n	Permanece
0	1	1	0	Set
1	0	0	1	Reset
1	1	\bar{Q}_n	Q_n	Complementa

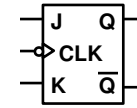
A representação esquemática do Flip-Flop JK é a mesma do Flip-Flop RS, com exceção das letras que indicam as entradas de comando:



Sensível à Borda de Subida

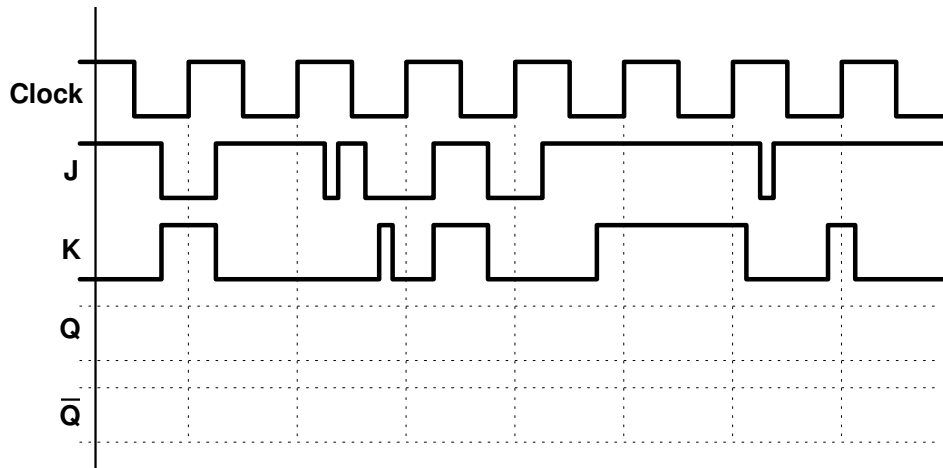


Sensível à Borda de Descida

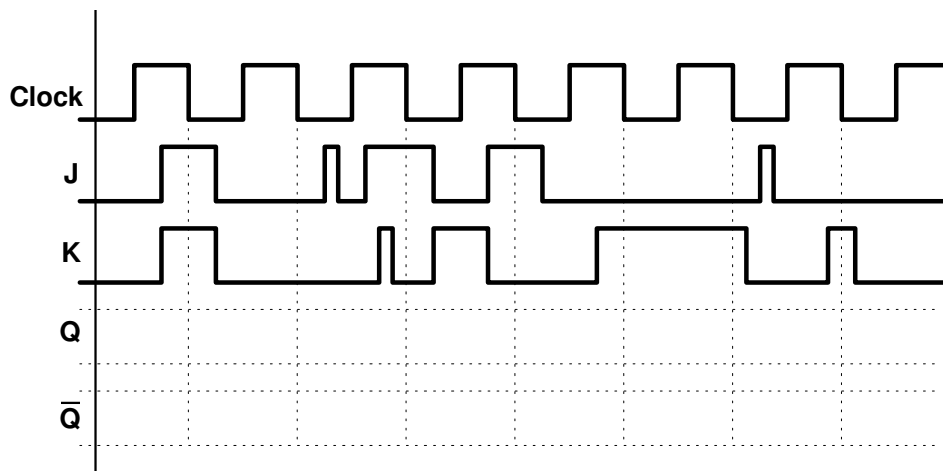


Exercícios
Completar as cartas de tempo dos circuitos citados a seguir:

1) Flip-Flop JK Sensível à Borda de Subida

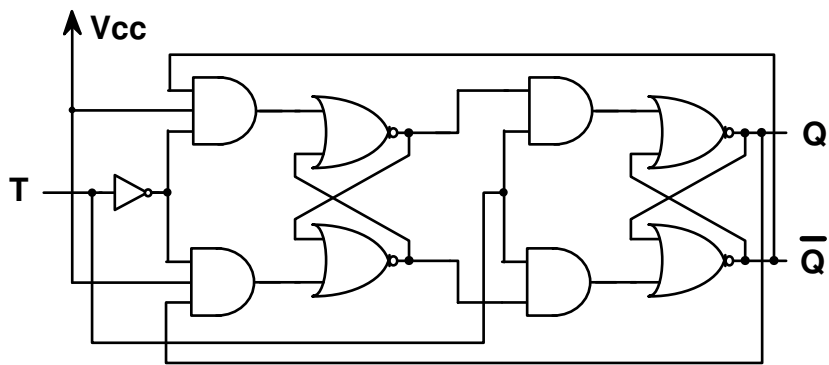


2) Flip-Flop JK Sensível à Borda de Descida

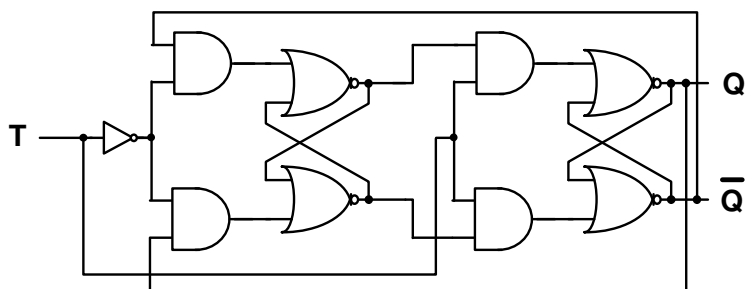


7) Flip-Flop tipo T

Esta é uma variação do Flip-Flop JK, onde ambas entradas de comando estão presas ao nível lógico 1. A entrada de Clock passa a se chamar T (toggle) e é a única entrada do circuito. A função deste dispositivo é complementar o nível lógico que está armazenando a cada borda do sinal de clock (para qual ele é sensível). Analise o circuito lógico:

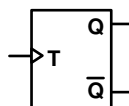


Uma outra forma de obtenção deste circuito é trocarmos as portas AND de três entradas para portas de duas entradas como mostra o circuito abaixo:

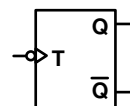


A representação esquemática do circuito está logo abaixo:

Flip-Flop tipo T sensível à Borda de Subida

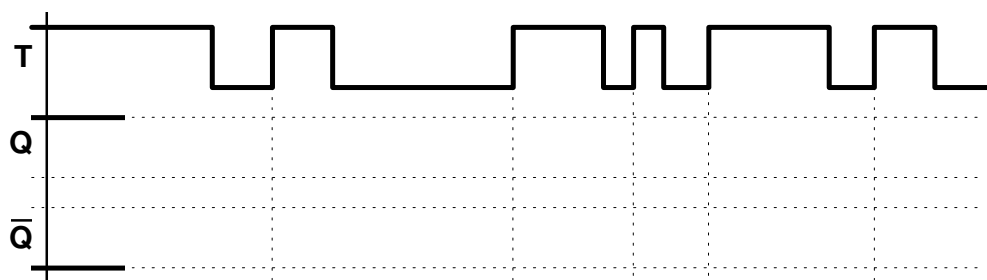


Flip-Flop tipo T sensível à Borda de Descida



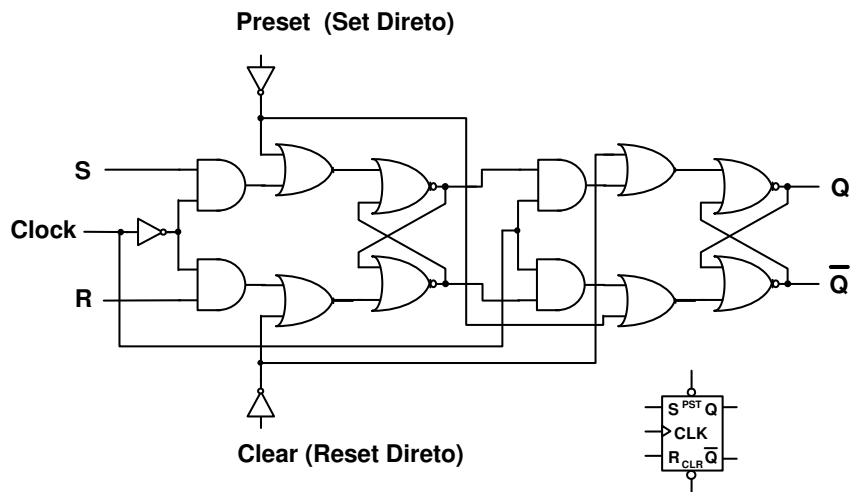
Exercício

Complete as cartas de tempo do Flip-Flop tipo T Sensível à Borda de Subida



Vimos que os Flip-Flops são sincronizados com as bordas do clock para qual são sensíveis. Este fato é muito interessante para muitos projetos mas nos obriga a esperar um período completo do clock para que ele mude de estado. Como vimos nas diversas cartas de tempo, a condição inicial de um latch ou de um Flip-Flop é indefinida e portanto para operarmos um circuito com segurança quando ele é ligado, teríamos que esperar um período completo do clock para podermos definir a sua situação inicial, se não fossem inventados os terminais de SET DIRETO e RESET DIRETO ilustrados na figura abaixo:

Flip Flop RS Sensível à Borda de Subida com Preset e Clear



Os conceitos de Preset e Clear se aplicam a todos os dispositivos **SÍNCRONOS** estudados até agora pois permitem uma mudança de estado assíncrona.

Exercício:

Desenhe as representações esquemáticas de todos os dispositivos estudados até agora, acrescentando os terminais de Preset e Clear quando for possível.

Circuitos Seqüenciais

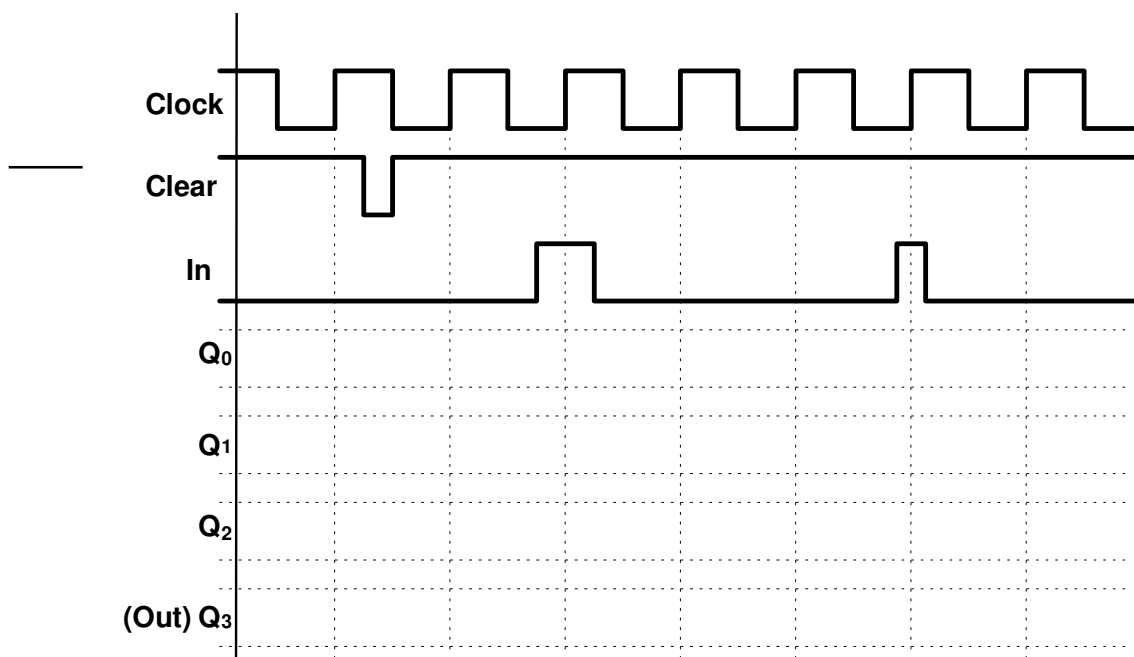
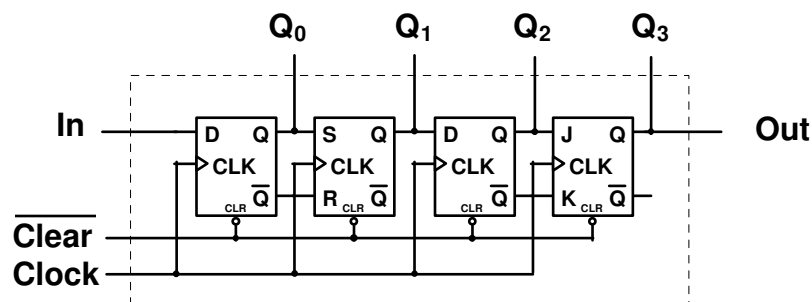
Introdução

Os circuitos seqüenciais propriamente ditos tem como elementos básicos os Flip-Flops e Latches. No capítulo anterior analisamos dois deles (Latch de n bits e Registrador), que aparentemente não nos dão a idéia de que são realmente circuitos seqüenciais, mas são. Neste capítulo os circuitos estudados serão contadores e estes sim nos induzirão a idéia de seqüência.

1) Registrador de Deslocamento

Este circuito é construído por Flip-Flops associados de maneira que o bit armazenado em um será transferido para outro a cada borda de clock, provocando assim um deslocamento dos valores armazenados.

O exemplo abaixo ilustra um registrador de deslocamento construído com os Flip-Flops tipo D, RS e JK. Na verdade esses registradores são construídos com apenas um tipo de Flip-Flop mas misturando os tipos, você poderá ver como se constrói um registrador com qualquer um deles. No caso de usarmos apenas Flip-Flops RS ou JK, temos que transformar o primeiro em um tipo D, para que o nosso dispositivo final possa ser operado com apenas uma entrada de bits.

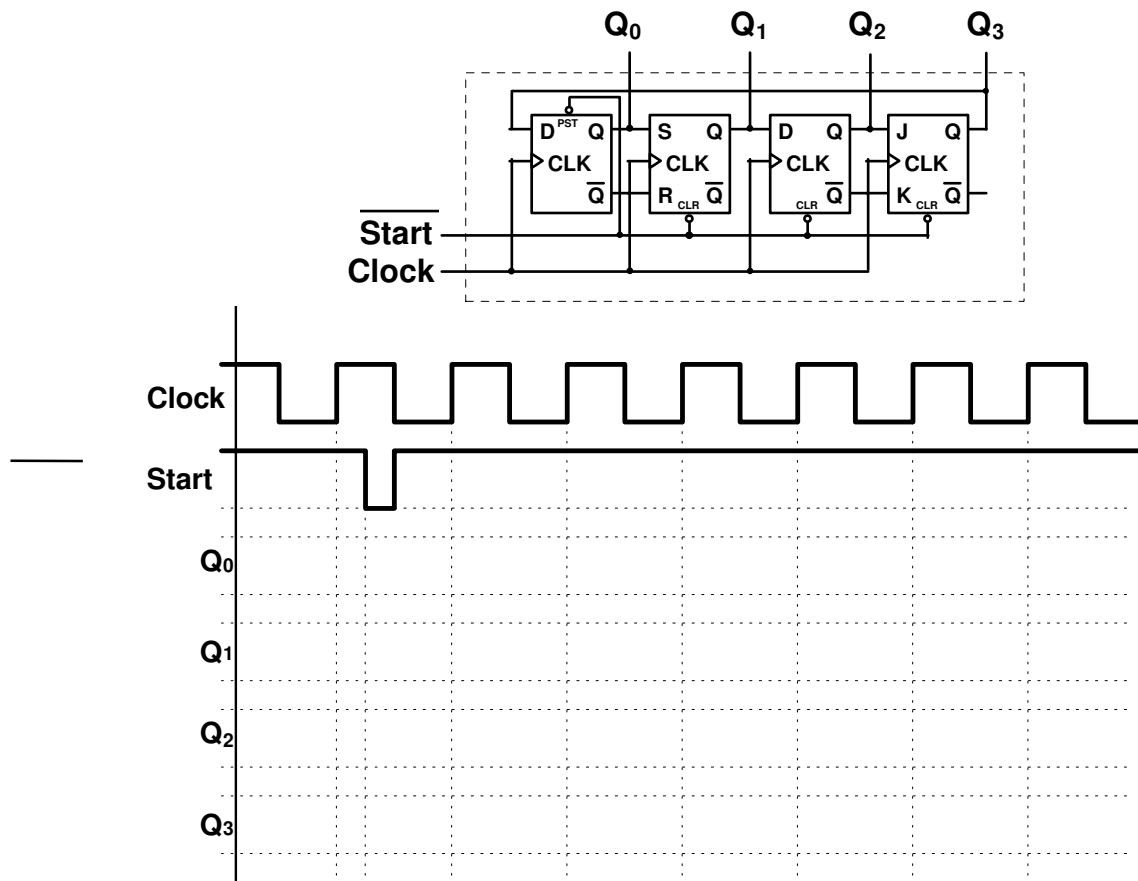


2) Contador em Anel

Um contador em anel tem como base o registrador de deslocamento. A diferença é a interligação de saída com a entrada. Desta forma os bits ficarão circulando indefinidamente neste dispositivo.

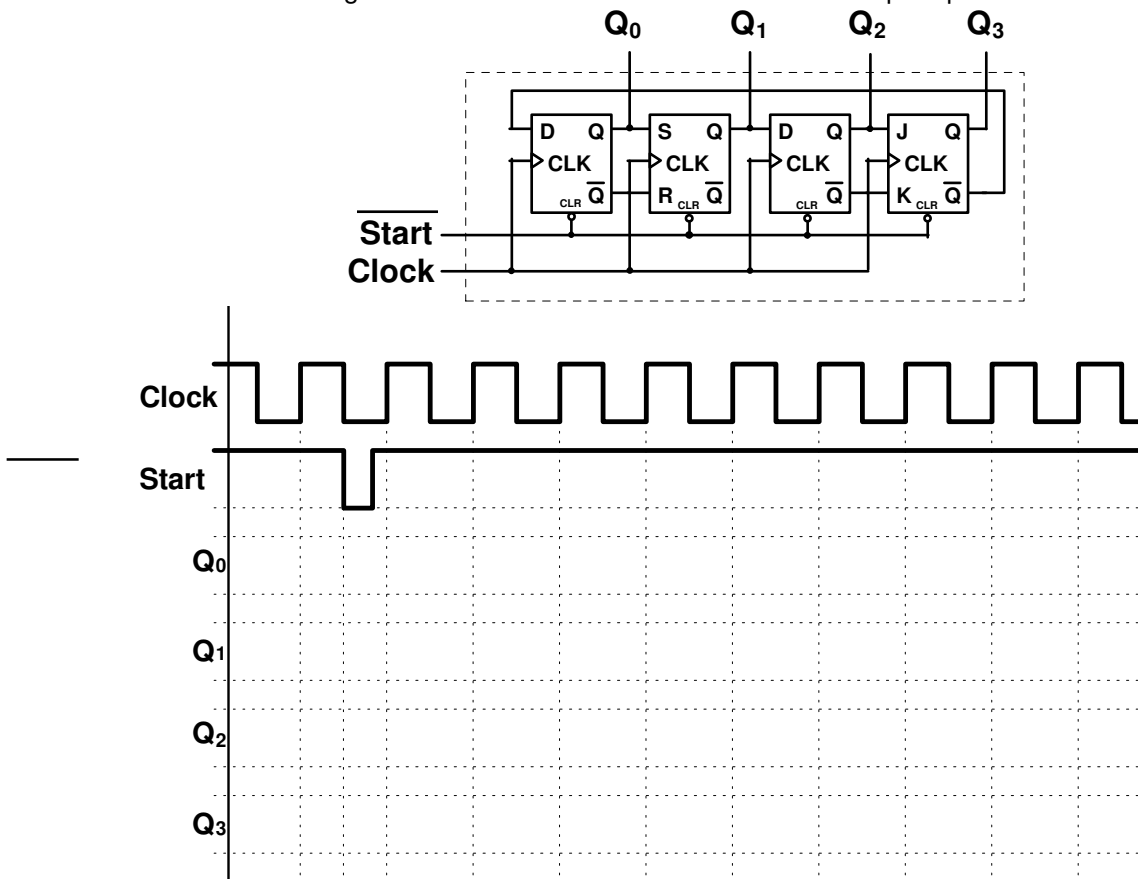
O módulo de contagem de um contador em anel é igual ao número de Flip-Flops que o compõem.

Observe a troca do nome do terminal CLEAR por START que é explicada pelo fato do primeiro Flip-Flop ser presetado quando ativarmos este terminal para garantir a circulação de apenas um bit.



3) Contador em Anel Torcido

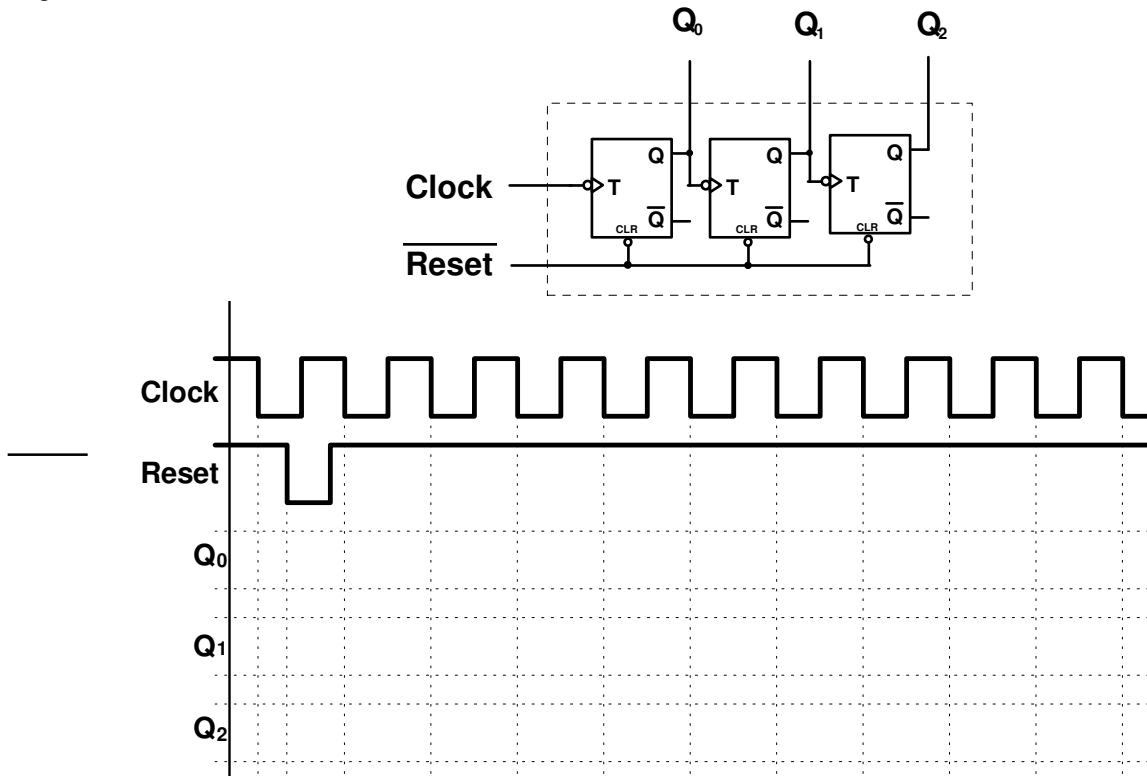
Este contador tem como base o circuito anterior, porém a realimentação é feita em modo invertido, isto é, se o último Flip-Flop estiver setado na próxima borda o primeiro estará resetado e vice-versa. O módulo de contagem deste circuito é o dobro do número de Flip-Flops.



4) Contador Binário Assíncrono

Este dispositivo é capaz de fazer a contagem binária com módulo = 2^n , onde n é o número de Flip-Flops que compõem o circuito. Os Flip-Flops são do tipo T e sensíveis à borda de descida. Se construirmos o mesmo circuito com Flip-Flops tipo T sensíveis à borda de subida a contagem será decrescente.

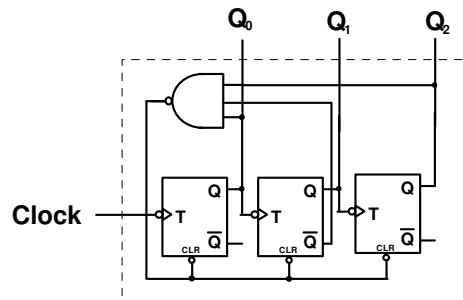
O circuito é assíncrono porque não há ligações do clock e um único sinal já que os Flip-Flops estão ligados em cascata.

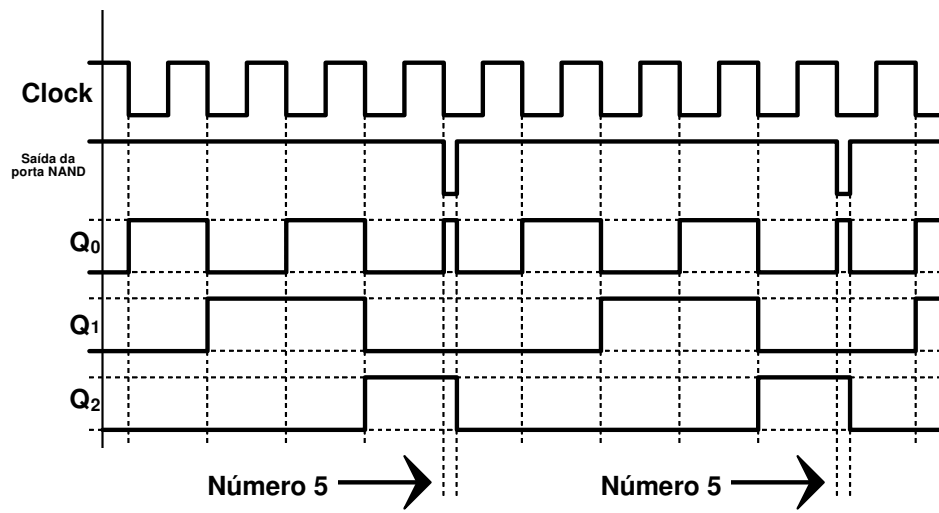


5) Contador Binário de Módulo Arbitrário Assíncrono

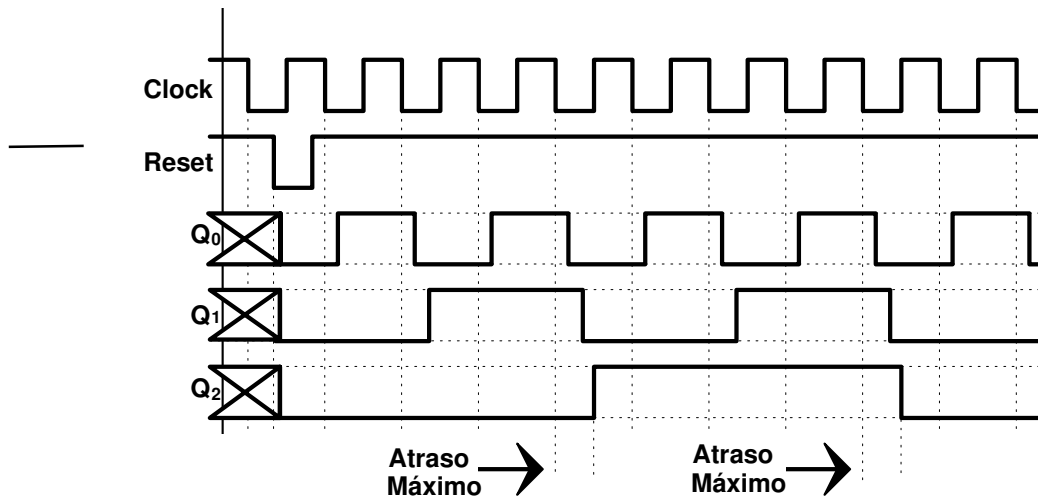
Usando como base o circuito anterior, podemos construir contadores binários com qualquer módulo de contagem, basta associarmos um arranjo lógico que seja capaz de identificar quando ultrapassarmos o último número da contagem e então resetar o circuito para que tudo comece de novo. Esta não é uma maneira muito apropriada de se construir um circuito deste tipo, pois existirá sempre um estado que não pertence ao módulo de contagem que é justamente aquele que provoca o RESET. Para ilustrar este fato, temos a carta de tempos com este problema bastante exagerado, logo após a representação do circuito do contador de 0 a 4.

Q_2	Q_1	Q_0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5*
1	1	0	6
1	1	1	7





Este problema ocorre porque os circuitos eletrônicos não são ideais. Existem atrasos na passagem e execução das operações lógicas com sinais elétricos. Para evidenciar este problema veja o que acontece se exagerarmos este atraso na carta de tempos do **Contador Binário Assíncrono** visto nesse capítulo:



Circuitos Seqüenciais - MOORE e MEALY

Introdução

Moore e Mealy foram os inventores de circuitos que nos permitem construir contadores e controladores de máquinas totalmente automáticos. Seus circuitos são construídos com apenas portas lógicas e Flip-Flops. A partir deste instante você poderá começar a idealizar os seu primeiros projetos em Eletrônica Digital que realmente tem cara de projeto.

Antes de iniciarmos a teoria sobre os circuitos de MOORE e de MEALY temos que definir alguns conceitos:

- Estado

Em Eletrônica Digital **estado** é a situação definida pelos níveis lógicos das saídas dos circuitos. O número de estados diferentes que um circuito pode ter é igual 2 elevado ao número de saídas que ele tem. Obviamente se um circuito tem saídas normais e complementares não podemos contar as duas. Ou você conta as normais ou as complementares. Exemplos:

- O Flip-Flop só pode assumir 2 estados porque tem apenas uma saída
- Um registrador de 4 saídas pode assumir 16 estados diferentes

- Seqüência

Em eletrônica Digital é uma sucessão de estados que seguem uma lógica de formação e sempre se repete.

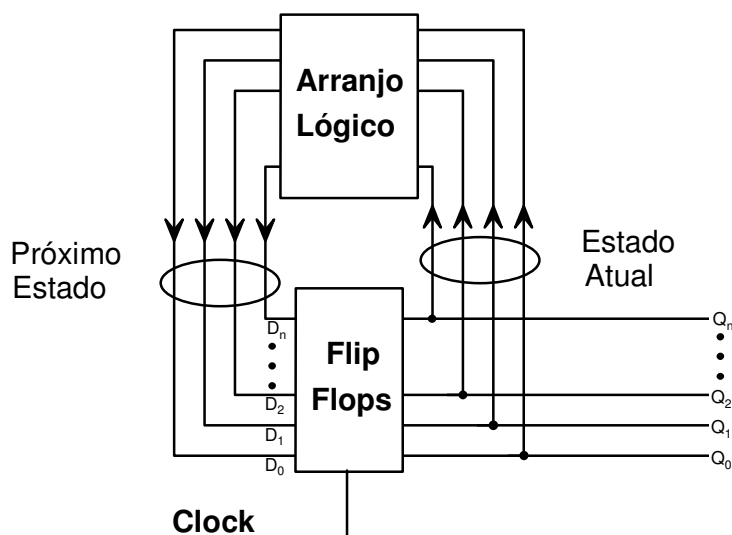
- Tabela de Estados

É a tabela usada para projetar circuitos seqüenciais. Cada linha da tabela mostra o estado atual do circuito e qual será o próximo

Circuitos MOORE

Esses circuitos são formados por portas lógicas e Flip-Flops. Os Flip-Flops são responsáveis pelo armazenamento do estado atual de uma seqüência e as portas lógicas pela geração dos níveis lógicos do próximo estado. Para cada arranjo lógico diferente teremos uma seqüência diferente.

O circuito MOORE tem a seguinte forma:



Exemplo:

Projetar um circuito com a configuração de MOORE que execute a contagem 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2 etc, em binário. Use Flip-Flops tipo D

O primeiro passo é a construção da Tabela de Estados. Como vimos na definição desta tabela em cada linha ela deverá mostrar o estado atual do circuito e também o próximo:

Estados			
Atual		Próximo	
Q ₁	Q ₀	D ₁	D ₀
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Agora projetamos um arranjo lógico que, tendo como entrada as saídas Q dos Flip-Flops (estado atual), gere o nível lógico apropriado em suas saídas (próximo estado). Usaremos para isso os mapas de Karnaugh:

Q ₀ \ Q ₁	0	1
0	0	1
1	1	0

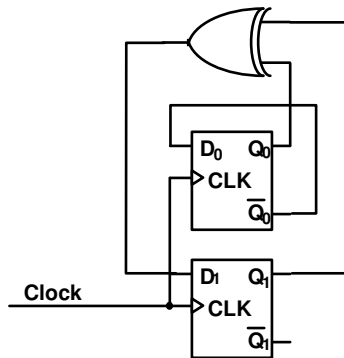
$$D_1 = Q_0 \bar{Q}_1 + \bar{Q}_0 Q_1$$

$$D_1 = Q_0 \oplus Q_1$$

Q ₀ \ Q ₁	0	1
0	1	1
1	0	0

$$D_0 = \bar{Q}_0$$

Finalmente montamos o circuito:



Exercício:

- 1) Projetar um circuito contador de 0 a 7 binário.
- 2) Projetar um circuito contador que execute a seqüência 3,4,2,6,7,0,5 e 1

Irrelevância

Como você deve ter observado nos exemplos e exercícios os contadores executavam seqüências que usavam todos os estados possíveis para cada caso: para 2 saídas tivemos 4 estados e para 3 saídas tivemos 8 estados. Existem casos em que um contador não usará todos estados possíveis para as suas saídas e aí chamaremos esses estados de **irrelevantes**. Analise o exemplo a seguir:

Contador de 3 saídas que conte apenas os números pares (0, 2, 4 e 6).

Estados						
Atual			Próximo			
Q ₂	Q ₁	Q ₀	D ₂	D ₁	D ₀	
0	0	0	0	1	0	
0	0	1	x	x	x	
0	1	0	1	0	0	
0	1	1	x	x	x	
1	0	0	1	1	0	
1	0	1	x	x	x	
1	1	0	0	0	0	
1	1	1	x	x	x	

Q ₂ \ Q ₁	00	01	11	10
0	0	1	0	1
1	x	x	x	x

$$D_2 = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1 = Q_2 \oplus Q_1$$

Q ₂ \ Q ₁	00	01	11	10
0	1	0	0	1
1	x	x	x	x

$$D_1 = \bar{Q}_1$$

Q ₂ \ Q ₁	00	01	11	10
0	0	0	0	0
1	x	x	x	x

$$D_0 = 0$$

Como os estados irrelevantes nunca estarão presentes nas saídas do contador, a letra "x" indica que os valores para o próximo estado tanto podem ser 1 como 0. Isso facilita a simplificação do mapa K pois podemos considerar o "x" 0 ou 1 desde que obtenhamos a **maior simplificação possível**.

A irrelevância ajuda por um lado mas atrapalha por outro. Imagine que ao ligarmos o contador os Flip-Flops "acordem" num estado que não pertença à seqüência. Qual seria o próximo estado?

A única forma de se descobrir é atribuir valores para os "Xs" conforme o agrupamento que fizemos:

Estados						
Atual			Próximo			
Q ₂	Q ₁	Q ₀	D ₂	D ₁	D ₀	
0	0	0	0	1	0	
0	0	1	0	1	0	
0	1	0	1	0	0	
0	1	1	1	0	0	
1	0	0	1	1	0	
1	0	1	1	1	0	
1	1	0	0	0	0	
1	1	1	0	0	0	

Q ₂ \ Q ₁	00	01	11	10
0	0	1	0	1
1	0	1	0	1

$$D_2 = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_1 = Q_2 \oplus Q_1$$

Q ₂ \ Q ₁	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$$D_1 = \bar{Q}_1$$

Q ₂ \ Q ₁	00	01	11	10
0	0	0	0	0
1	0	0	0	0

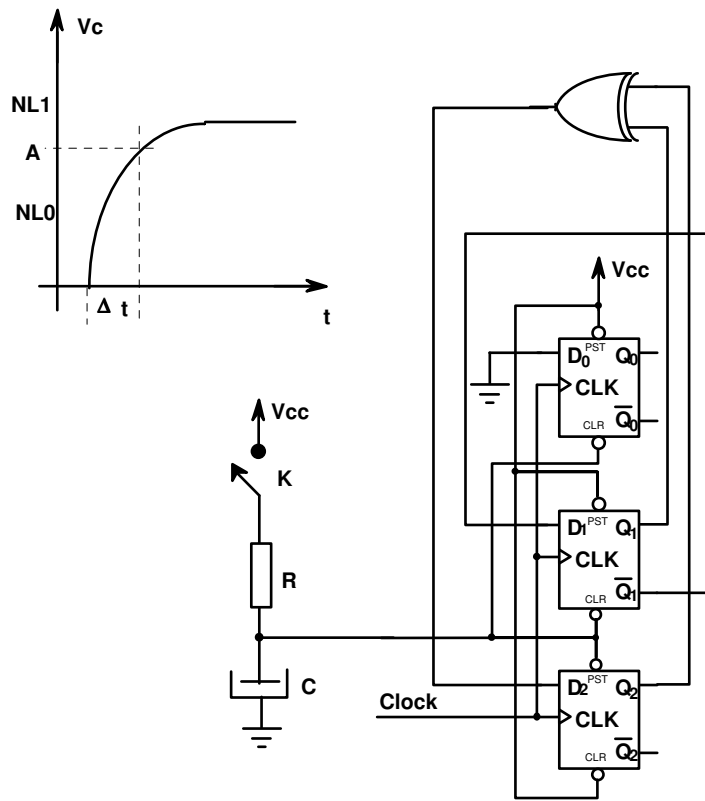
$$D_0 = 0$$

Concluimos que:

- Se "acordar" no estado 1 próximo será o 2
- Se "acordar" no estado 3 próximo será o 4
- Se "acordar" no estado 5 próximo será o 6
- Se "acordar" no estado 7 próximo será o 0

Felizmente não tivemos problema com este circuito. Caso o estado inicial seja um que não pertença a seqüência, o contador automaticamente retorna para um valor que pertença. Mas existem casos, aliás a maioria, que isso não acontece. O resultado é drástico porque teremos na saída valores indesejáveis. Além desses casos não é aconselhável permitir que um circuito tenha em sua saída um estado que não pertence a seqüência. A solução para o problema é usarmos as entradas PRESET e CLEAR, para que o circuito "acorde" no estado que quisermos.

O exemplo abaixo mostra o circuito deste último projeto "acordando" sempre no estado 4.



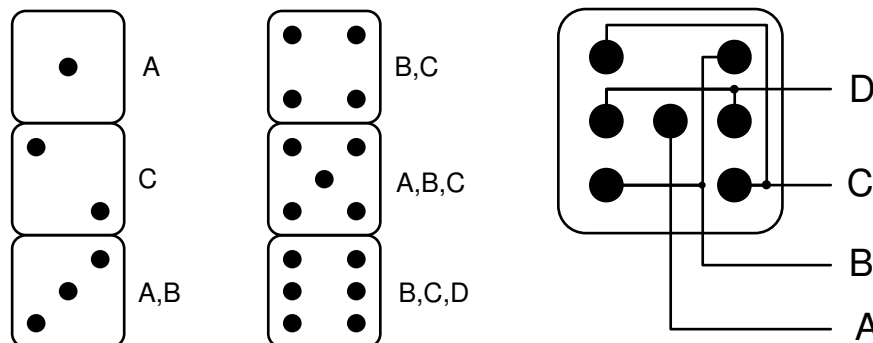
O circuito formado pelo capacitor C e pelo resistor R tem a função de manter os PRESETs dos Flip-Flops 0 e 1 e o CLEAR do Flip-Flop 2 em nível lógico 0 durante um Δt , quando ligamos o circuito. Veja o gráfico. Não se esqueça que cada Flip-Flop tem que estar conectado a V_{cc} e a GND para que funcione. No exemplo e em como nos esquemas em geral essas ligações são omitidas para simplificar o desenho.

Exercícios:

1) Projete um circuito que execute a seqüência 3, 7, 11, 5, 0, 2, 9, 13, 1 e 14, com início automático no estado 3 e verifique o que aconteceria se acidentalmente os estados que não pertencem a seqüência fossem o estado atual.

2) Projete um circuito digital pelo processo de MOORE que seja capaz de simular um DADO onde os números seriam indicados por LEDs.

Use o seguinte esquema para o painel:



Circuitos MOORE com Flip-Flops RS e JK

Até agora estudamos os circuitos seqüenciais em pequenos projetos e exercícios usando como memória de estado atual os Flip-Flops tipo D porque é mais simples para controlarmos o próximo estado, já que este tipo de Flip-Flop permite alteração de seu conteúdo através de uma única entrada de controle, a entrada D. Se passarmos a usar Flip-Flops RS ou JK teremos um aumento de 2 vezes no número de entradas de controle por Flip-Flop e conseqüentemente um aumento de 2 vezes no número de mapas de Karnaugh a serem resolvidos, mas devemos observar que os mapas terão uma resolução com circuitos lógicos muito mais

reduzidos pois esses Flip-Flops introduzirão uma série de estados irrelevantes nesses mapas. Observe a comparação dos três casos, considerando que a primeira coluna da tabela mostra o estado atual dos dispositivos, a segunda o próximo estado que desejamos e a terceira qual nível lógico devemos colocar nas entradas de controle para que o próximo estado desejado aconteça:

Atual	Próximo	Controle
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Atual	Próximo	Controle	
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Atual	Próximo	Controle	
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Vamos resolver novamente o primeiro exemplo deste capítulo usando Flip-Flops JK para observarmos se realmente existem vantagens. Este exemplo pedia um circuito seqüencial contador de 0 a 3:

Atual		Próximo		Controle			
Q_{1n}	Q_{0n}	Q_{1n+1}	Q_{0n+1}	J_1	K_1	J_0	K_0
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1

$Q_1 \backslash Q_0$	0	1
0	0	X
1	1	X

$$J_1 = Q_0$$

$Q_1 \backslash Q_0$	0	1
0	X	0
1	X	1

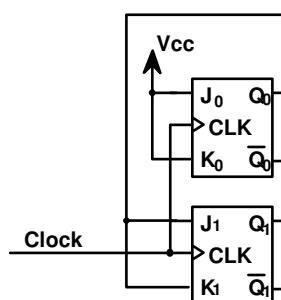
$$K_1 = Q_0$$

$Q_1 \backslash Q_0$	0	1
0	1	1
1	X	X

$$J_0 = 1$$

$Q_1 \backslash Q_0$	0	1
0	X	X
1	1	1

$$K_0 = 1$$



Exercícios:

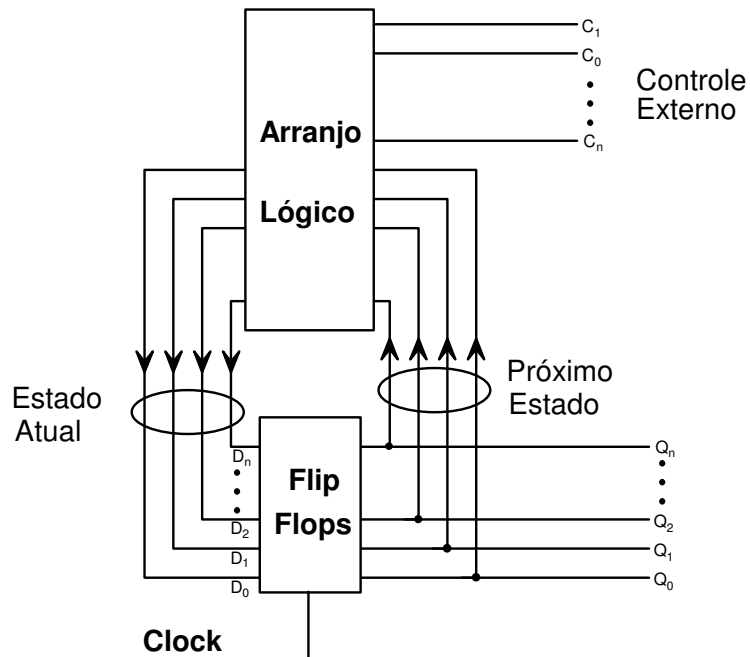
- 1) Projetar um contador de 3 saídas que conte apenas os números pares (0, 2, 4 e 6) usando Flip-Flops RS.
- 2) Reprojete o circuito que simula o DADO, usando Flip-Flops JK.

Circuitos MEALY

Os circuitos idealizados por MEALY tem basicamente a mesma estrutura dos circuitos MOORE, mas MEALY prevê entradas independentes para que um operador possa alterar o a seqüência para subseqüências

pré-definidas. Como exemplo imagine um contador que possa contar crescente e decrescente, de acordo com o nível lógico colocado em sua entrada de controle (up/down).

Forma básica dos circuitos MEALY



Exemplo:

Projetar um contador de módulo 4, que conte em ordem crescente quando uma entrada C for igual a 0 e em ordem decrescente quando esta mesma entrada C for igual a 1. Use Flip-Flops tipo D.

Estados				
Atual			Próximo	
C	Q ₁	Q ₀	D ₁	D ₀
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

C \ Q ₁ / Q ₀	00	01	11	10
0	0	1	0	1
1	1	0	1	0

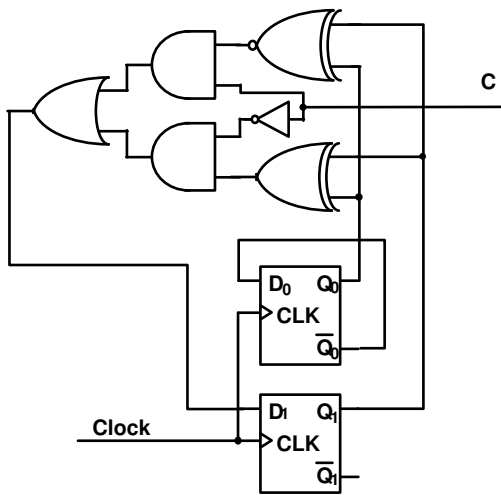
$$D_1 = \overline{C}Q_0\overline{Q_1} + \overline{C}\overline{Q_0}Q_1 + CQ_0Q_1 + C\overline{Q_0}\overline{Q_1}$$

$$D_1 = \overline{C}(Q_0\overline{Q_1} + \overline{Q_0}Q_1) + C(Q_0Q_1 + \overline{Q_0}\overline{Q_1})$$

$$D_1 = \overline{C}(Q_0 \oplus Q_1) + C(Q_0 \odot Q_1)$$

C \ Q ₁ / Q ₀	00	01	11	10
0	1	1	1	1
1	0	0	0	0

$$D_0 = \overline{Q_0}$$



Exercícios:

- 1) Desenvolva o circuito do exemplo anterior usando Flip-Flops RS
- 2) Projete um circuito contador up/down de módulo 8 com Flip-Flops JK

Temporizadores

Introdução

Temporizadores, em Eletrônica Digital, são dispositivos que contam o tempo decorrido a partir do momento em que são acionados. Existem, porém, alguns temporizadores especiais que ficam contando tempos indefinidamente, indicando com a ativação de um sinal, toda vez que a contagem chega ao fim, isto é, ele repete indefinidamente a contagem. Nesses casos podemos chamar esses dispositivos de osciladores.

Alguns exemplos de temporizadores:

- No primeiro caso, temos as clássicas **minuterias** usadas em corredores de edifícios visando a economia de energia elétrica. Esses dispositivos mantêm as luzes dos corredores acesas durante o tempo suficiente para que o morador saia do elevador e chegue ao seu apartamento. Em seguida as lâmpadas são apagadas. Hoje em dia temos ainda muitas minuterias mecânicas em operação, mas estas estão sendo substituídas por circuitos eletrônicos, devido a sua simplicidade, confiabilidade e principalmente custo. Esses temporizadores podem ser **Analógicos**, **Digitais** ou **Híbridos** (parte analógico e parte digital).

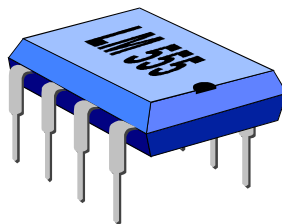
É muito importante você observar que neste exemplo, as lâmpadas só voltam a acender quando o morador ativar novamente o botão de disparo da minuteria.

- No segundo caso, temos os semáforos que estão continuamente acendendo e apagando as três lâmpadas que compõem seus painéis, com intervalos de tempo bem definidos.

Observe que neste caso, ninguém ativa nenhum botão. Os semáforos são automáticos.

Na nossa disciplina vamos estudar os dois tipos de circuitos e teremos sempre como base a utilização de um circuito integrado de grande utilização em muitos projetos: este circuito é o **LM 555**. Considerando que ele seja novidade ao estudante, vamos fazer uma análise com nível de profundidade suficiente para o bom entendimento dos projetos que usam este circuito integrado.

LM 555



Atualmente, existem vários fabricantes que produzem este circuito e de maneira geral, qualquer um deles serve para a maioria das aplicações práticas, entretanto para projetos que exigem maior confiabilidade é conveniente que se faça uma análise nas especificações dadas pelos fabricantes nos conhecidos **Data Books**.

Na nossa disciplina, estudaremos o funcionamento teórico destes dispositivos, sem nos preocuparmos com os parâmetros de Tensão e Corrente elétricas.

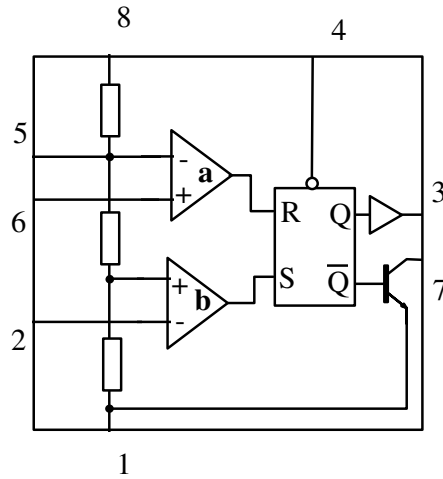
Este circuito é híbrido pois tem partes analógicas (uma rede de resistores divisores de tensão e dois amplificadores operacionais na configuração clássica de **comparadores**) e ainda uma parte digital (Latch RS Assíncrono e um transistor que estará saturado ou em corte).

Obs.

Cabe aqui uma observação muito importante relativa as nomenclaturas **Latch** e **Flip-Flop**. Esses dispositivos são muito confundidos na bibliografia disponível do mercado. Quando resolvi escrever a apostila usada no ano anterior, achei mais completa e confiável as definições de **Latch** e **Flip-Flop** que **Andrew S. Tanenbaum** faz em seu livro '**Organização Estruturada de Computadores**'. Por esse motivo, usaremos as nomenclatura que estamos acostumados. Ressalto porém que em diversas publicações o aluno encontrará a descrição de **Flip-Flop** e não **Latch**. Para maiores detalhes sugiro uma consulta na obra de **Tanenbaum** nas páginas 81 e 82 da 3ª Edição (1992).

O circuito completo do integrado é ilustrado a seguir. Observe que não há nenhum dispositivo que seja novidade para você. O novo é a forma como eles estão interligados. Mesmo assim faremos uma análise detalhada de cada uma das partes.

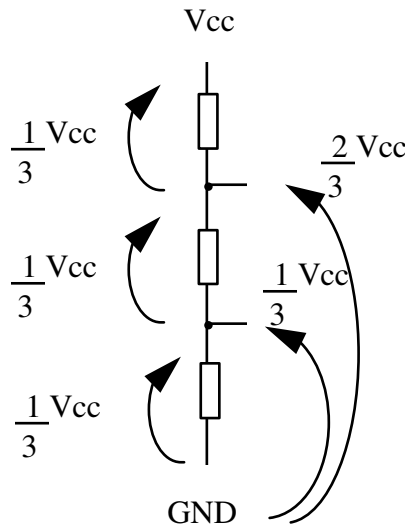
LM 555



Pinos	Descrição
1	GND
2	Disparo
3	Saída
4	Reset
5	Tensão de Controle
6	Sesor de Nível
7	Dreno
8	Vcc

Também será comum o aluno encontrar descrições diferentes para os pinos deste integrado, porém deverá perceber que elas significam a mesma coisa, ou seja, são sinônimos.

Inicialmente vamos analisar a rede de resistores presentes entre os pinos 1 e 8:



Os três resistores tem o mesmo valor e por isso temos $1/3$ da tensão aplicada entre os pinos 1 e 8 sobre cada resistor. Obviamente, em relação ao terra (GND) temos as tensões de $1/3$ de V_{cc} até o primeiro resistor e $2/3$ de V_{cc} até o segundo resistor. Observando esta rede conectada o circuito do 555, vemos que o pino 5 tem uma conexão com os resistores. Ela será útil para desbalancearmos a divisão da tensão, isto é, através deste pino podemos modificar os valores de $1/3$ e $2/3$ de V_{cc} para outros valores que esteja de acordo com nosso projeto ou até para a produção de efeitos especiais.

A segunda parte a ser analisada são os comparadores **a** e **b** que são os amplificadores operacionais. Fazendo uma breve recordação do funcionamento destes dispositivos, temos o seguinte resumo:

- se a tensão do terminal (+) for maior que a do terminal (-) temos nível lógico 1 na saída;
- se a tensão do terminal (-) for menor que a do terminal (+) temos nível lógico 1 na saída;
- nos outros casos temos nível lógico 0 na saída.

Uma boa regra para não se confundir é associar o sinal (+) com a palavra *maior* e o sinal (-) com a palavra *menor*. Sendo assim, quando os valores aplicados são coerentes, (a tensão no terminal + é maior que a tensão do terminal - ou então a tensão do terminal - é menor que a tensão do terminal +) temos nível lógico 1 na saída, caso contrário teremos nível lógico 0.

Ainda observando o desenho completo do LM 555, vemos que o comparador **a** é que ativa o terminal **reset** do latch e o comparador **b** o terminal **set** e ainda que cada comparador tem uma entrada ligada à um terminal externo (pinos 2 e 6) e a outra à rede de resistores. Além disso o terminal 5 está conectado a entrada do comparador **a** que é ligada a rede de resistores.

Analisando tudo que foi dito até agora, e ligarmos o circuito à uma fonte de tensão apropriada e deixarmos o pino 5 desconectado, podemos concluir que:

- para **setarmos** o latch temos que aplicar no pino 2 uma tensão menor que $1/3$ de V_{cc} ;
- Para **resetarmos** o latch temos que aplicar no pino 6 uma tensão maior que $2/3$ de V_{cc} .

Finalmente temos as análise dos pinos 3, que é a saída do latch com a potência amplificada por um **driver** e do pino 7 que é o **coletor** de um transistor que irá **drenar** a corrente presente neste terminal para o terra, quando o latch estiver **resetado** ($Q = 1$). Caso contrário, se comportará como um circuito aberto, isto é, não entra nem sai corrente alguma por este pino.

É muito comum encontrarmos um **inversor** ligando a saída \bar{Q} ao pino 3 e você deve entender que isso não altera nada no funcionamento do circuito, nem os valores de seus sinais.

Acredito que não haja a necessidade de comentários sobre o pino 4, mas de qualquer forma, aí está a sua função: **resetar** o latch **independentemente** dos sinais presentes nas entradas **S** e **R**, quando for aplicado sobre ele um nível **lógico 0** ou seja GND.

Terminada esta análise, vamos estudar o nosso primeiro circuito temporizador. Verifique portanto se não sobrou nenhuma dúvida sobre o funcionamento de cada uma das partes deste integrado, pois se isso acontecer você não conseguirá entender o funcionamento de um projeto que utilize este integrado.

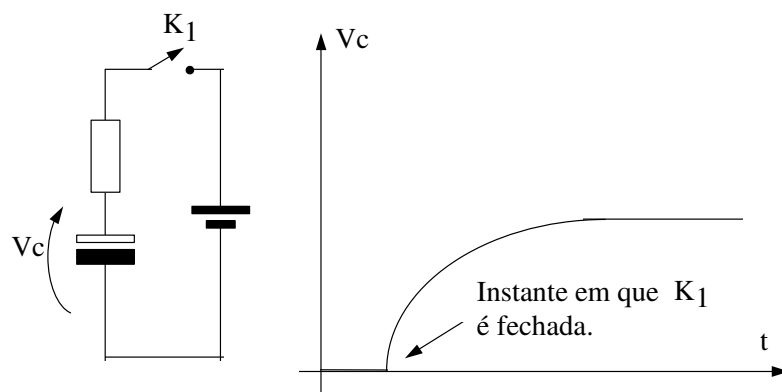
Multivibrador Mono-Estável

Este é o nome original do circuito eletrônico da minuteria. A palavra MONO quer dizer UM e portanto temos um Multivibrador com apenas um estado estável.

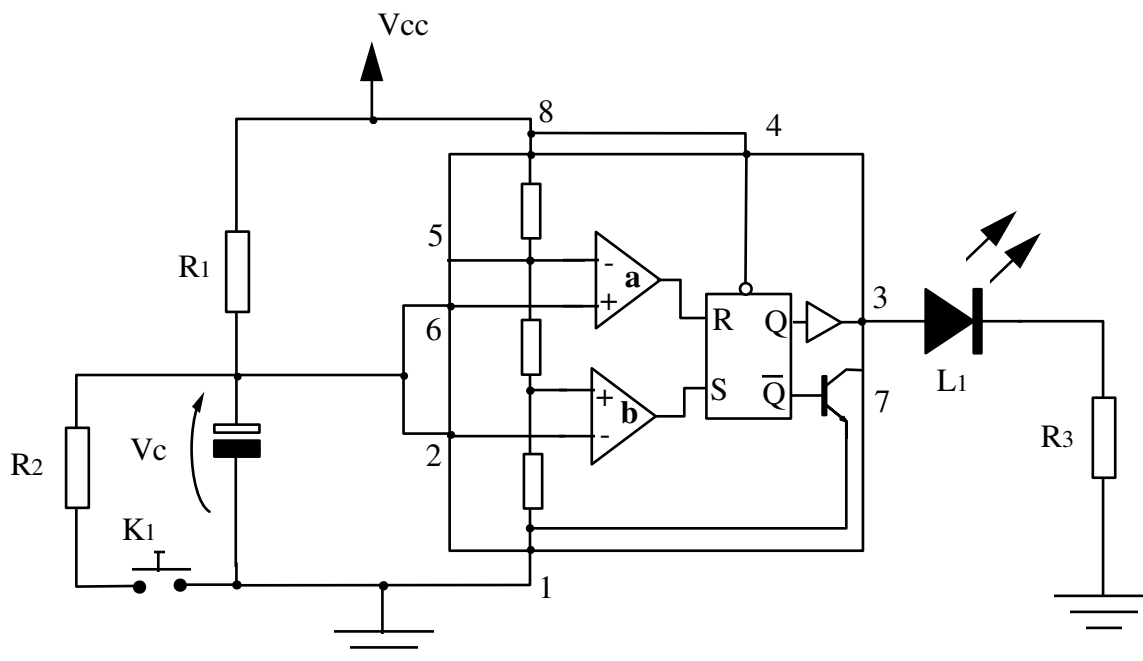
Recordando o funcionamento da minuteria, vemos que isto é verdade pois o circuito está estabilizado quando a lâmpada está apagada e ficará assim por um tempo indeterminado, se ninguém ativar a sua entrada. Quando um morador pressiona o botão para que a luz do corredor acenda, ele está na verdade desestabilizando o circuito que manterá uma (ou mais) lâmpadas acesas até que se esgote o tempo para qual o circuito foi projetado. Nesse instante a lâmpada se apaga e o circuito permanece estabilizado até que outra pessoa o desestabiliza novamente.

Tudo muito fácil e simples até agora, mas como projetar um circuito que funcione da maneira que foi descrita acima, usando o LM 555 ?

Primeiro temos que encontrar um componente eletrônico que seja capaz de nos fornecer o tempo como variável. Um componente possível é o **capacitor**, já que para que carregue com uma tensão é necessário um tempo e, obviamente, uma tensão aplicada entre seu dois terminais. Analise abaixo o gráfica de carga de um capacitor em série com um resistor:



Sabemos que a constante de tempo do circuito anterior é dada por $\tau = RC$ e portanto podemos variar o tempo necessário para a carga do capacitor modificando o valor do resistor ou do capacitor. Como o latch do CI 555 é acionado por valores de tensão definidos pela rede de resistores ($1/3$ e $2/3$ de V_{cc}) usaremos este fato para compor o nosso circuito final. Observe o esquema abaixo:



Considerando que o circuito estivesse desligado a algum tempo, portanto $V_c = 0$, e fosse ligado repentinamente. Nesse instante o LED L_1 estaria aceso pois sendo 0 a tensão aplicada nos pinos 2 e 6 teríamos um **set** no latch, provocado pelo comparador **b**. Com o passar do tempo, o capacitor vai se carregando com uma velocidade que depende do seu valor e do valor do resistor. Quando a tensão do capacitor atingir $2/3$ de V_{cc} , teremos nível lógico 1 na saída do comparador **a** e conseqüentemente o latch será **resetado**, fazendo com que o LED L_1 se apague. O circuito se estabiliza nessa situação, pois o capacitor continua se carregando até atingir o valor de V_{cc} e o latch não muda mais de estado.

Resumindo, quando ligamos o circuito, o LED estará aceso, mas logo em seguida se apaga e permanece assim por tempo indefinido.

Digamos que alguém pressione por algum tempo a chave K_1 . Analisando o circuito, percebemos que o capacitor vai se descarregar através de R_2 . Sendo assim, a tensão presente nos pinos 2 e 6 será 0 e o comparador **b** apresentará nível lógico 1 em sua saída, enquanto que o comparador **a** apresentará 0 e, portanto, o latch é setado, provocando o acendimento do LED. Quando a chave K_1 for deixada em repouso novamente, o capacitor inicia a sua carga e quando atingir novamente o valor de $2/3$ de V_{cc} , provocará o reset no latch e o circuito se estabiliza novamente.

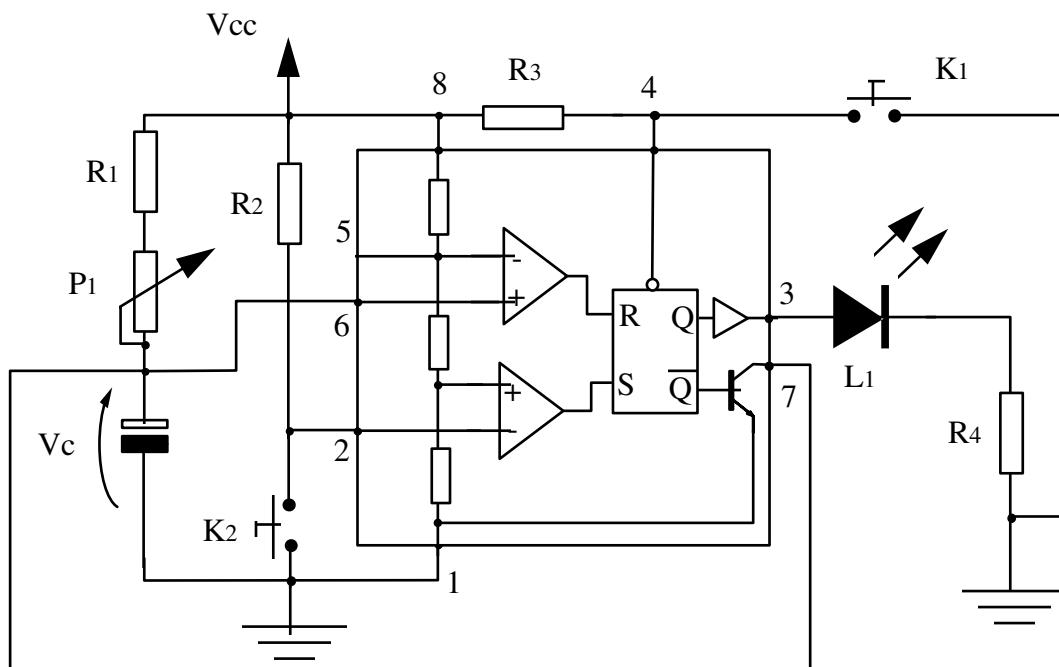
O circuito apresentado não é a única forma de se construir um temporizador com o LM 555 e além disso apresenta algumas deficiências. Ele foi escolhido por ser o mais simples de se entender.

A deficiência mais marcante é que os resistores R_1 e R_2 formam um divisor de tensão e portanto ao pressionarmos a chave K_1 , o capacitor não se descarrega completamente. Ficará com um valor de tensão igual o valor da tensão sobre R_2 . Além disso, R_1 tem que ter um valor que seja maior que o dobro de R_2 pois caso contrário a tensão do capacitor nunca será menor que $1/3$ de V_{cc} . Poderíamos retirar R_2 mas neste caso teríamos um faiscamento nos contatos da chave K_1 diminuindo muito a sua vida útil.

Outro ponto a ser notado é que a contagem do tempo inicia quando a chave K_1 é deixada em repouso e não no momento em que é pressionada.

Vejamos agora um circuito de temporização bem mais elaborado que também usa o LM 555. Note que o funcionamento do integrado é sempre o mesmo, mas as ligações externas fazem com que o projeto como um todo funcione de formas diferentes.

Analisando o circuito seguinte e considerando que ele está em repouso, isto é, o LED está apagado, concluímos que permanecerá indefinidamente neste estado pois se a saída apresenta nível lógico 0, significa que o latch está resetado e conseqüentemente que o pino 7 está fortemente conectado ao terra. Sendo assim, toda corrente que vem de V_{cc} e passa por R_1 e P_1 é desviada para o terra ao invés de carregar o capacitor. Ao pressionarmos momentaneamente a chave K_2 faremos com que o comparador **b** apresente nível lógico 1 em sua saída e conseqüentemente set o latch. Nesta situação o pino 7 se comportará como um circuito aberto, já que o transistor ligado à sua base estará em corte ($Q = 0$). O capacitor começara a se carregar com a corrente vinda de V_{cc} através de R_1 e P_1 . Quando a tensão no capacitor for ligeiramente maior que $2/3$ de V_{cc} , o comprador **a** apresente nível lógico 1 e assim o latch será resetado. Com o latch resetado, o transistor conectado ao pino 7 entra em saturação e além de descarregar imediatamente o capacitor, desvia toda corrente vinda de V_{cc} através de R_1 e P_1 para o terra estabilizando o circuito neste estado.



Os fabricantes deste integrado fornecem muitos outros exemplos de projetos que usam o LM 555 e junto com eles algumas fórmulas para cálculos teóricos. No caso deste circuito podemos calcular o valor do tempo em que o LED permanece aceso através da seguinte relação:

$$T = 1,1 (R_1 + P_1) C$$

onde:

T é o tempo em segundos

R₁ e P₁ são os valores desses componentes em MΩ

C é o valor do Capacitor em μF

Como o potenciômetro varia seu valor resistivo, seria conveniente calcular o valor de T para o máximo e mínimo valores de P₁.

Uma diferença entre este circuito e o anterior que também vale a pena comentar é que no segundo a temporização inicia no instante em que K₂ é pressionada, independentemente do tempo que ela permanece assim. Temos, além disso, uma outra chave K₁ que permite a interrupção da contagem do tempo pois força o latch a um reset, qualquer que seja os sinais aplicados em S e R.

Apenas a título de curiosidade, se montarmos o circuito com os seguintes componentes, teríamos um intervalo máximo de 27,5 e mínimo de 16,5 segundos:

$$R_1 = 1,5 \text{ M}\Omega$$

$$R_2 = 15 \text{ K}\Omega$$

$$R_3 = 15 \text{ K}\Omega$$

$$R_4 = 1 \text{ K}\Omega$$

$$P_1 = 1 \text{ M}\Omega$$

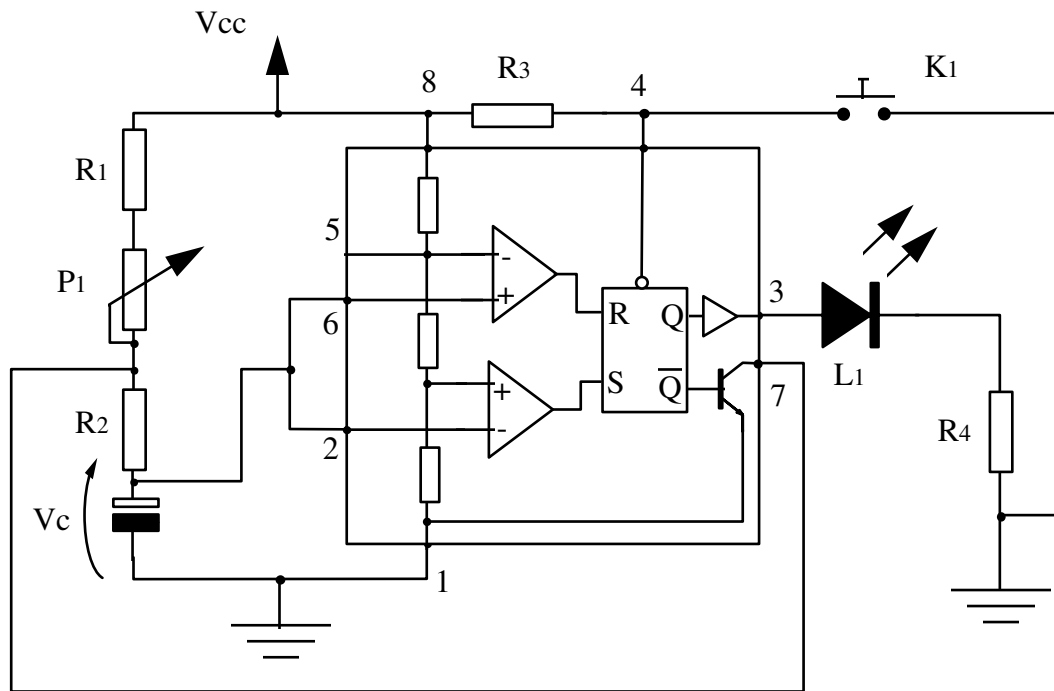
$$C = 10 \mu\text{F}$$

$$V_{cc} = 12\text{V}$$

Multivibrador Astável

Como seu próprio nome sugere, este circuito é instável, ou seja não se estabiliza. Como a sua saída é digital, concluímos que teremos ali a presença de um sinal retangular e podemos chama-lo também de oscilador.

Novamente o “coração” do projeto a ser estudado é o Circuito Integrado LM 555 e o elemento que faz a temporização é um capacitor. A diferença básica está na forma de conexão de todos os componentes. Analise o esquema abaixo:

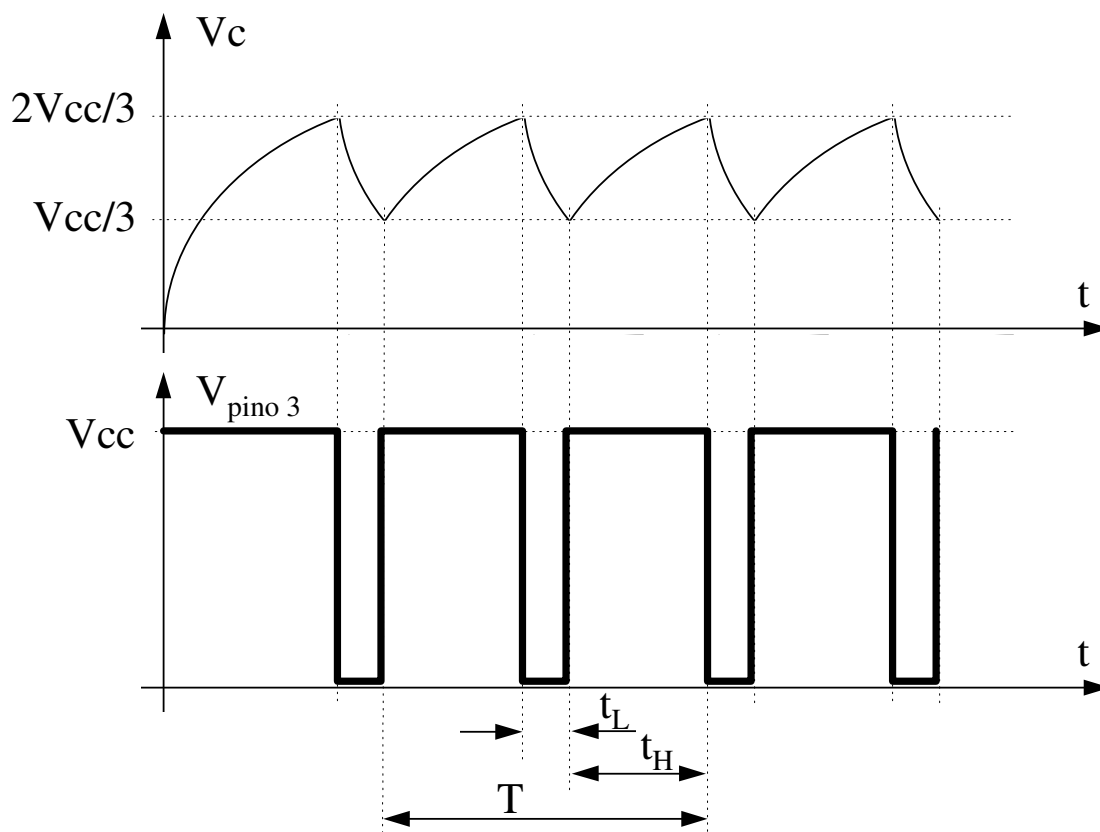


Temos agora tanto o pino 2 como o pino 6 estão conectados à tensão presente no capacitor. Isso significa que quando o capacitor estiver com uma tensão inferior à $1/3$ de V_{cc} o comparador **b** ativa a sua saída e seta o latch e quando a tensão for maior $2/3$ de V_{cc} o comparador **a** ativa a sua saída e reseta o latch. Relembrando que o pino 7 está fortemente ligado à terra (via transistor) quando o latch está resetao e que se comporta como um circuito aberto quando o latch está setado, podemos tirar algumas conclusões:

- Considerando que inicialmente o circuito estava desligado um tempo suficiente para que o capacitor perdesse toda sua carga e foi repentinamente ligado. Nessa condições o latch será setado pois temos no pino 2 uma tensão menor que $1/3$ de V_{cc} ($V_c = 0$). Sendo assim o pino 7 está aberto e então circulará uma corrente de V_{cc} para o capacitor (passando por R_1 , P_1 e R_2) e ele começa a se carregar. Observe que não há outro caminho para a corrente circular, já que a impedância de entrada de operacionais é muito grande.
- Quando o capacitor estiver carregado com uma tensão ligeiramente superior a $2/3$ de V_{cc} o comparador **a** ativa o sinal R do latch, provocando um reset. Quando isso ocorrer o pino 7 ficará fortemente ligado à terra e a corrente vinda de V_{cc} que passa por R_1 e P_1 é desviada para terra por este pino (7) pois este caminho é “mais fácil”. Além disso o capacitor que já tem um de seus terminais ligado à terra e o outro à R_2 se descarregará pois o outro terminal de R_2 também está conectado a terra via pino 7. O capacitor então começa a se descarregar.
- Quando o valor da tensão sobre o capacitor for ligeiramente inferior a $1/3$ de V_{cc} , o comparador **b** dispara o S do latch. Como o pino 7 é um circuito aberto quando o latch está setado, o capacitor vai se carregar novamente pois não existe mais outro caminho para a corrente e o ciclo está fechado, ou seja, se continuarmos a análise veremos que estamos exatamente no item analisado logo acima e portanto o circuito está oscilando.

Uma ilustração gráfica da tensão do capacitor sincronizada com o valo na saída pode facilitar muito o entendimento do funcionamento do circuito e inclusive fica claro que o primeiro instante em que o capacitor se carrega é mais longo que os outros pois ele começará do zero e não de $1/3$ de V_{cc} .

Da mesma forma que no caso anterior, os fabricantes fornecem vários esquemas de exemplos de circuitos e suas respectivas fórmulas para cálculos teóricos. Este exemplo estudado é o mais usado para este integrado e você o identificará facilmente em projetos de sirenes, pisca-pisca, osciladores, etc. Aliás é bom comentar que o circuito de clock do DIGILAB I é exatamente este. A mudança das frequências é feita pela alteração dos valores dos resistores ou do capacitor. No nosso exemplo, a mudança pode assumir diversos valores pois temos um resistor variável (P_1) em série com o resistor R_1 .



$$t_H = 0,693(R_1 + P_1 + R_2) C$$

$$t_L = 0,683 R_2 C$$

$$T = t_H + t_L$$

$$f = \frac{1}{T} = \frac{1,44}{(R_1 + 2 (P_1 + R_2)) C}$$

Da mesma forma que no caso anterior os resistores estão em $M\Omega$, o capacitor em μF , t em segundos e f em Hz.

Observe no gráfico que $t_H > t_L$, isto porque para a carga do capacitor temos R_1 , P_1 e R_2 em série, mas para a descarga temos apenas R_2 . Isto pode ser verificado no esquema e nas fórmulas. Podemos também concluir que o potenciômetro varia a frequência de oscilação e o t_H , mas o t_L não é alterado.

Temos um grande inconveniente comum aos dois circuitos estudados: **IMPRECISÃO**. Os capacitores sofrem uma influência muito grande da temperatura ambiente e mudam seus valores em quantidade significativa. É lógico que para minuterias em corredores de edifícios, pisca-piscas e outros projetos onde a precisão do tempo não é tão importante não ha nenhum problema em usar os exemplos que analisamos. Mas para construirmos um relógio digital, por exemplo, seria absolutamente catastrófico o emprego dos exemplos analisados. A saída para este problema é a utilização de circuitos mais sofisticados com osciladores à cristal.

O **crystal de quartzo** é um mineral encontrado na natureza que se comporta de maneira muito estável no diz respeito à oscilações estudaremos osciladores à cristal quando estivermos falando de geradores de clock para microprocessadores. Acredito que seria mais conveniente.

Conversores D/A e A/D

Introdução

Este é, sem sombra de dúvidas, um dos mais interessantes capítulos a ser estudado nessa Disciplina, pois teremos condições de entender realmente como acontecem alguns chamados 'milagres' da informática.

Até hoje, em Técnicas Digitais I e II trabalhamos apenas com **zeros** e **uns**, e temos a impressão que tudo funciona assim dentro da Eletrônica Digital. Isso é verdade pois todas as Máquinas Digitais realizam seus processos em bytes e bits. Mas então, como explicar o som tão perfeito e computadores equipados com Multi Mídia, ou então máquinas fotográficas que usam disquetes no lugar de filmes, ou ainda equipamentos de medidas (multímetros, balanças, etc. digitais) que transforma grandezas da natureza em números num *display*? Muito simples! Em todos os exemplos acima temos conversões **Digitais/Analógicas** ou então **Analógicas/Digitais**.

Existem muitas técnicas de conversões de sinais analógicos para digitais e 'vice-versa' e podem ser encaradas de duas formas: A conversão de um sinal analógico para uma seqüência de **bytes** ou a conversão de um sinal analógico para uma seqüência de **bits**. Cada forma tem suas vantagens e desvantagens e por isso vamos estudar todas elas (atenção: estudaremos todas as **formas** e não todos os **circuitos** disponíveis no mercado).

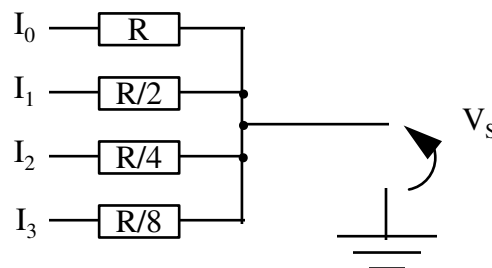
Conversão Digital/Analógica para seqüência de BYTES.

Suponha que tenhamos uma máquina digital que nos forneça uma seqüência de bytes como saída de seu processamento. Um contador binário é um exemplo muito bom deste tipo de máquina digital, pois ele apresenta um byte diferente a cada pulso de clock aplicado em sua entrada. Outro exemplo bem característico seria a saída **paralela** de um microcomputador. Ela fornece um byte diferente cada vez que recebe um sinal de **permissão para envio**. Este tipo de saída em microcomputadores é normalmente usado para a conexão de impressoras que são capazes de transformar estes byte em caracteres alfanuméricos e ainda fornecer o sinal de permissão para envio toda vez que está pronta para imprimir um novo caracter.

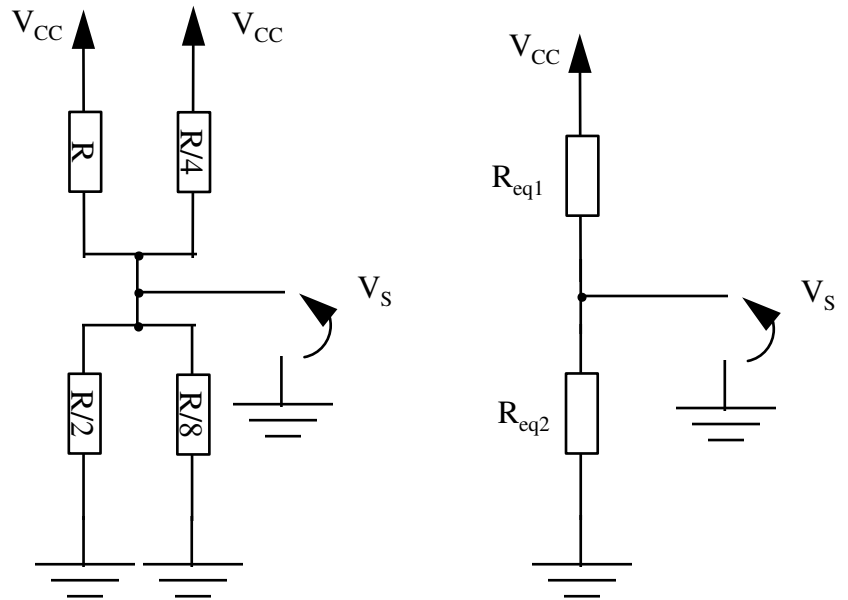
O conversor que estudaremos agora terá a função de transformar bytes diferentes em níveis diferentes de alguma grandeza elétrica (normalmente tensão ou corrente). Temos dois tipos de circuito capazes de executarem esta função:

a) Conversor D/A a resistor ponderado.

Lembrando que um bit = 0 é equivalente a uma saída ligada a GND e um bit = 1 a uma saída ligada a VCC, podemos usar o seguinte circuito para o conversor:



Se introduzirmos neste circuito o byte **0101** (equivalente a 5 em decimal) poderíamos redesenhá-lo da seguinte maneira:



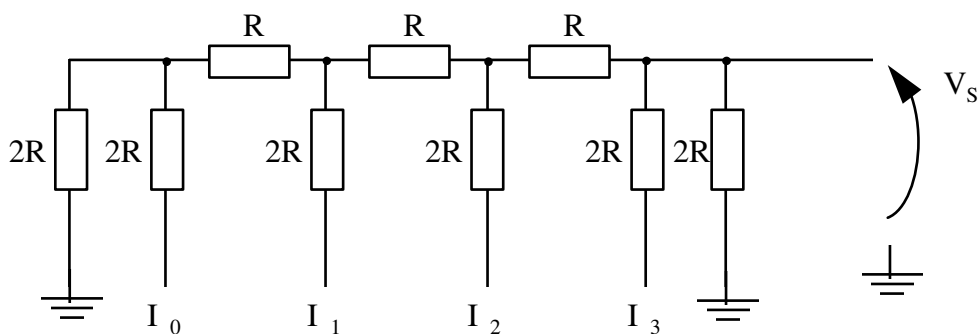
$$V_S = R_{eq2} I \quad I = \frac{V_{cc}}{R_{eq1} + R_{eq2}} \Rightarrow V_S = \frac{R_{eq2} V_{CC}}{R_{eq1} + R_{eq2}}$$

Este circuito apresenta dois inconvenientes muito grandes:

- As impedâncias de entrada e saída não são constantes para cada byte diferente introduzido na entrada.
- É muito difícil encontrar os resistores com os valores que o circuito exige. Imagine se tivermos um conversor deste tipo para 16 bits.

b) Conversor D/A de escada R-2R

O circuito:

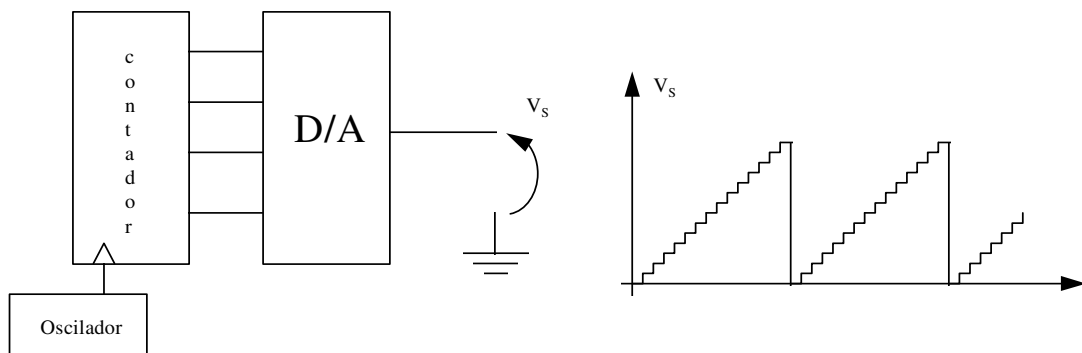


Neste caso, o cálculo da tensão V_S necessita da aplicação do teorema de Thevenin várias vezes. Se verificarmos para vários bytes de entrada notaremos que V_S muda para cada um da mesma forma que no circuito anterior, mas as impedâncias de entrada e saída permanecem constantes.

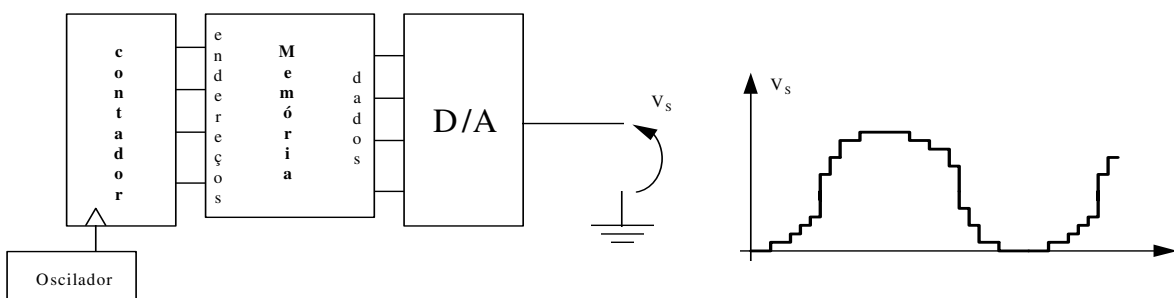
Como o que interessa para a nossa disciplina é o funcionamento do circuito e não o cálculo de seus valores, já que encontramos facilmente no mercado Circuitos Integrados prontos para o uso, com vantagens relativas a seu tamanho e custo vamos partir ao estudo de uma tabela verdade característica destes tipos de circuitos.

I_3	I_2	I_1	I_0	V_s
0	0	0	0	0
0	0	0	1	$1V_{cc}/15$
0	0	1	0	$2V_{cc}/15$
0	0	1	1	$3V_{cc}/15$
0	1	0	0	$4V_{cc}/15$
0	1	0	1	$5V_{cc}/15$
0	1	1	0	$6V_{cc}/15$
0	1	1	1	$7V_{cc}/15$
1	0	0	0	$8V_{cc}/15$
1	0	0	1	$9V_{cc}/15$
1	0	1	0	$10V_{cc}/15$
1	0	1	1	$11V_{cc}/15$
1	1	0	0	$12V_{cc}/15$
1	1	0	1	$13V_{cc}/15$
1	1	1	0	$14V_{cc}/15$
1	1	1	1	$15V_{cc}/15$

Como exemplo do funcionamento de um Conversor D/A para uma seqüência de bytes temos o circuito abaixo, onde um contador binário Hexadecimal fornece os bytes de entrada e na saída temos o sinal desenhado no gráfico ($V_s \times t$):



Sabemos que o contador gera bytes em seqüência crescente (ou decrescente), mas um computador, por exemplo, poderá gerar bytes pré definidos para conseguirmos na saída qualquer forma de onda e não apenas este tipo exemplificado acima. O gráfico seguinte ilustra uma senoide produzida por dados armazenados convenientemente em uma memória.



Como você deve ter notado a senoide deixa muito a desejar, mas observe que estamos trabalhando com um converso D/A de apenas quatro bits e isso nos dá apenas 16 patamares de tensão. Imagine se trocássemos este conversor por um outro com 16 bits de entrada. Teríamos, então na saída $65536 (2^{16})$ patamares de tensão e isso deixaria a senoide praticamente perfeita. Observe porém que a memória teria que ter também 65536 endereços (64 KBytes), o contador também teria que ter 16 bits e freqüência do oscilador teria que ser muito maior. Concluímos então que a **resolução** de um conversor Digital/Analogico está diretamente ligada ao número de bits que ele apresenta como entrada e que este fator além de elevar o seu preço encarece todo circuito de apoio ao seu funcionamento.

b.) Conversor A/D com comparadores de tensão.

Analise o circuito abaixo:

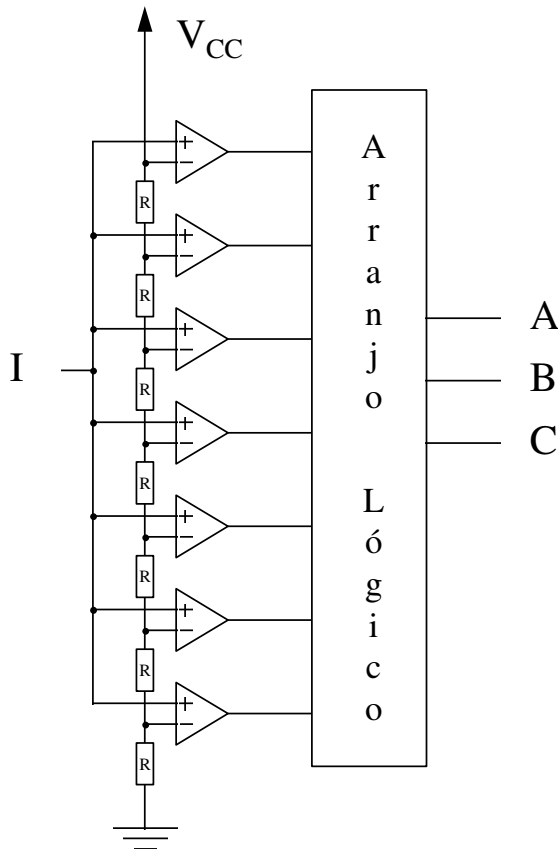


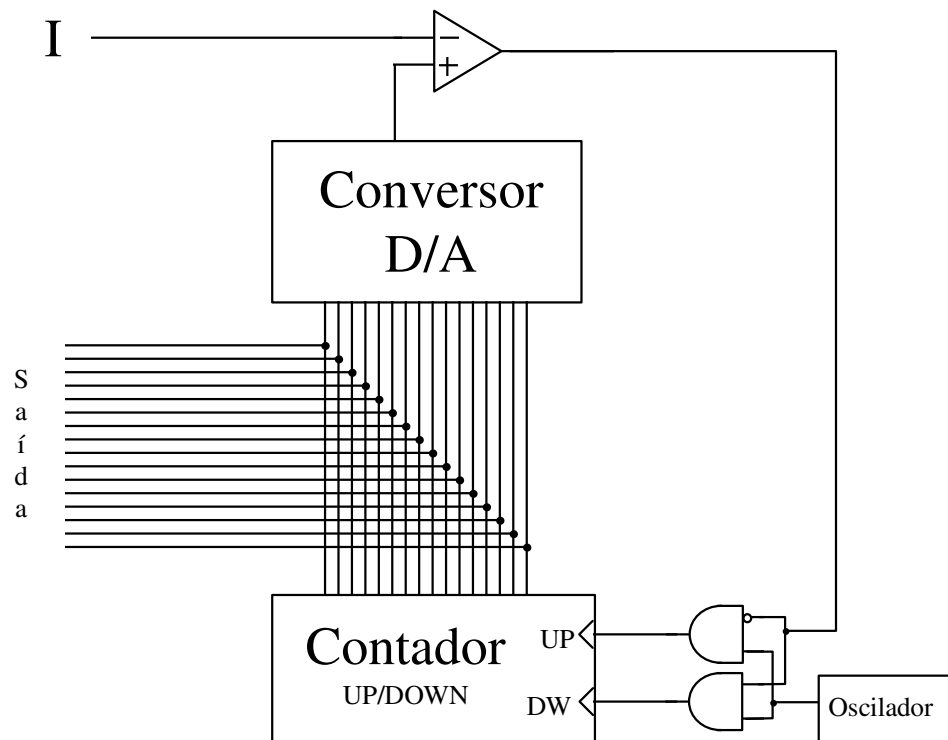
Tabela Verdade

I	C	B	A
$V_{cc}/8$	0	0	0
$2V_{cc}/8$	0	0	1
$3V_{cc}/8$	0	1	0
$4V_{cc}/8$	0	1	1
$5V_{cc}/8$	1	0	0
$6V_{cc}/8$	1	0	1
$7V_{cc}/8$	1	1	0
$8V_{cc}/8$	1	1	1

A rede composta por 8 resistores de mesmo valor (R) divide a tensão da fonte em 8 parcelas absolutamente iguais. Os comparadores detectam quando a entrada de seus terminais não inversores tiverem uma tensão maior que a de seus terminais inversores. Sendo assim quando a tensão aplicada em I, que é a tensão de se deseja converter for 0V, nenhum comparador terá a sua saída ativada pois todos eles terão nas suas entradas não inversoras uma tensão menor que a aplicada (pela rede de resistores) em suas entradas inversoras. Conforme a tensão aplicada em I for aumentando seu valor, os comparadores vão ativando suas saídas, seguindo a ordem de baixo para cima até o momento em tivermos V_{cc} na entrada, quando então todos terão suas saídas ativadas. O bloco seguinte é um arranjo lógico que tem a função de codificar as saídas dos comparadores em um código binário, isto é, se nenhum comparador estiver com sua saída ativada o Arranjo Lógico terá como saída 000; se apenas o primeiro estiver ativado o arranjo lógico produz 001 na saída; se o primeiro e o segundo estiverem ativados, o Arranjo Lógico produz 010 na saída e assim sucessivamente até o momento em que todos estiverem com suas saídas ativadas e o Arranjo Lógico com 111 na sua saída. Este circuito é sem sombra de dúvidas muito bom e faz a conversão num espaço de tempo muito pequeno, porém o seu custo é extremamente elevado. Imagine se necessitarmos de um conversor com 16 bits da saída, teremos que ter 65536 comparadores na entrada. Absolutamente inviável.

b₂) Conversor A/D com quantização em Bytes

Esta técnica de conversão reduz bastante o número de componentes no circuito, mas o tempo de conversão fica muito grande. Analise o circuito:



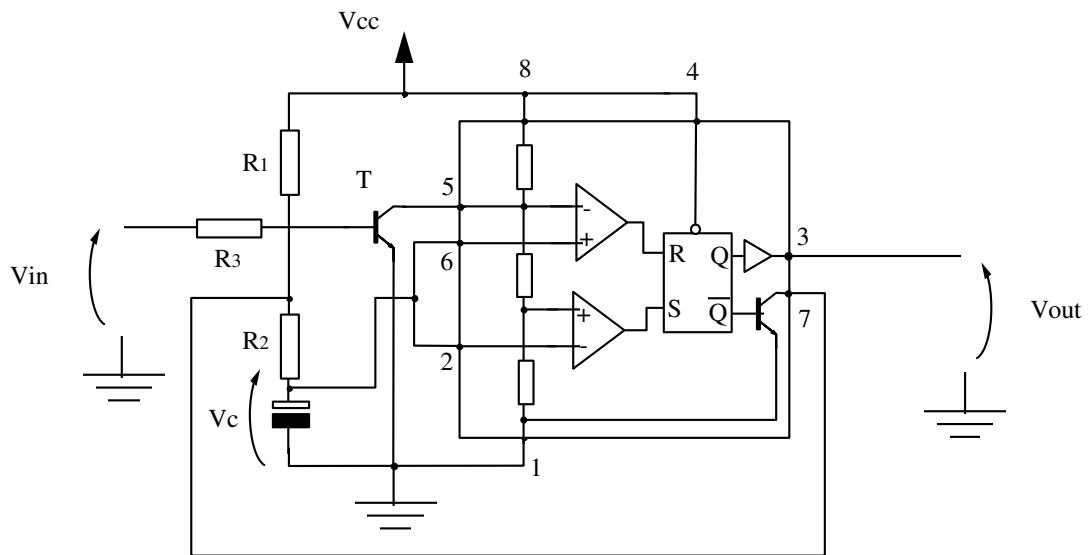
Temos circuito um contador UP/DOWN conectado a um conversor D/A. A função desses elementos é de gerar uma tensão, que pode ser crescente ou decrescente, de acordo com a entrada de clock que recebe os pulsos (bordas) do oscilador. Um comparador e um arranjo lógico, composto por duas portas AND e um inversor controlam o 'destino' do sinal de clock. Se a tensão aplicada em I for maior que a tensão gerada pelo conversor D/A, o comparador produz uma saída igual a zero e portanto o sinal de clock vindo do oscilador será aplicado na entrada UP do contador. Caso a tensão aplicada em I seja menor que a tensão gerada pelo conversor D/A o comparador produzirá uma saída igual a V_{CC} e então o sinal vindo do oscilador será aplicado na entrada DW do contador. Desta forma o contador será incrementado ou decrementado até que a tensão produzida pelo conversor D/A se iguale à tensão aplicada em I . Nesse instante temos na saída do circuito, que é na realidade a saída do contador, um byte que corresponde ao valor da tensão aplicada.

Os circuitos analisados até agora recebem um byte e produzem uma tensão correspondente ou então produzem um byte que corresponde a o valor de uma tensão. A conversão Digital para Analógico até que é feita com uma velocidade razoavelmente alta, porém a conversão Analógico para digital ou é extremamente cara (item b_1) ou extremamente lenta (item b_2). Os conversores D/A são utilizados em situações em que se deseja converter o valor de um byte para um valor de tensão ou corrente sem maiores problemas, mas os conversores A/D são úteis apenas para conversões que não exigem uma grande velocidade, como por exemplo medidores de grandezas físicas em geral (balanças, tensões, correntes, temperaturas etc.).

Para fazermos a conversão de sinais com velocidade extremamente alta, como sons, imagens etc. temos que usar uma outra técnica de conversão que é a conversão para uma seqüência de bits, isto é, o sinal produzido por um conversor A/D não é mais um byte e sim uma seqüência de bits de tamanho predefinido. O conversor D/A usado para que retornemos ao sinal original é, basicamente, um filtro passa-baixas que nos fornecerá o valor médio desta seqüência de bits. Para entendermos melhor, vamos analisar primeiro o conversor A/D.

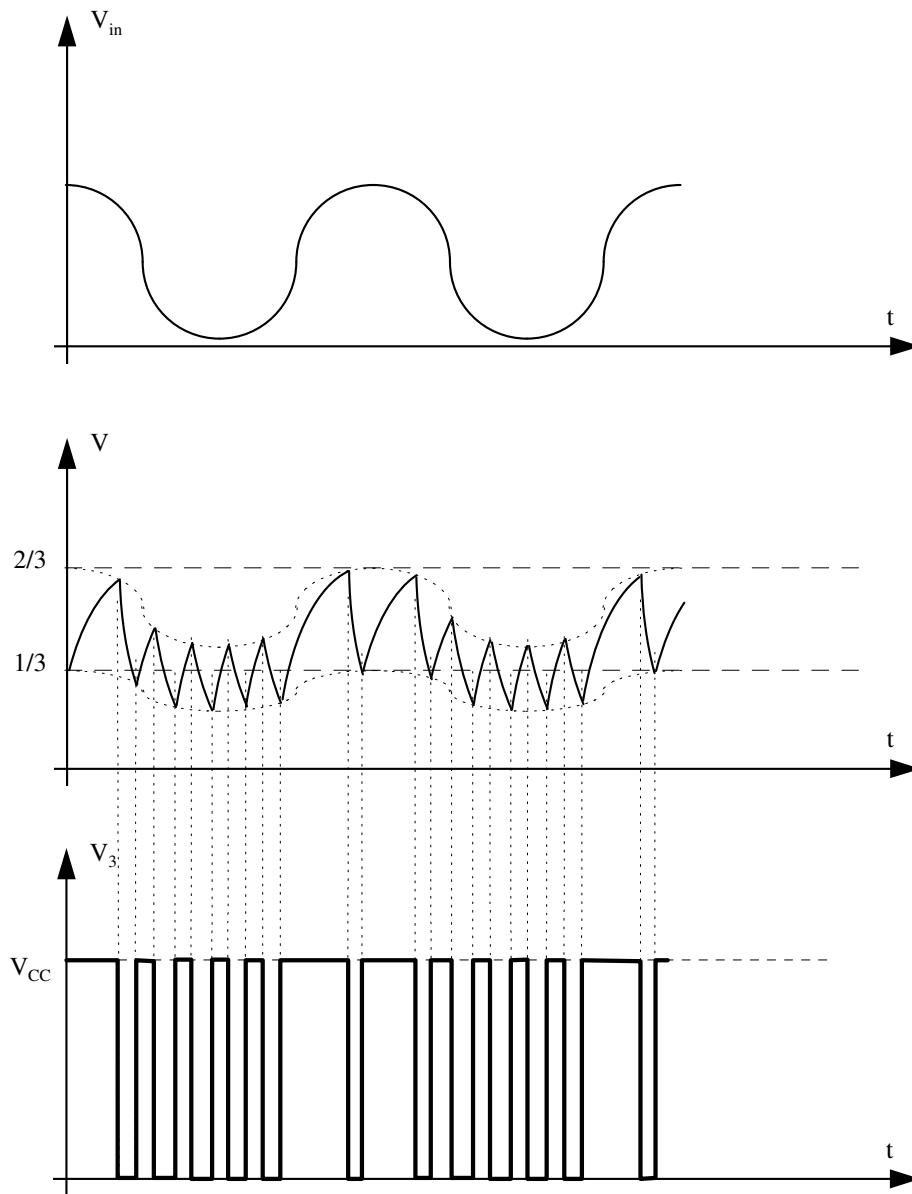
a) Conversão Analógica/Digital por modulação DELTA ou modulador PWM

Existem muitos circuitos que produzem este tipo de conversão ou modulação. Vamos analisar um bastante interessante que usa como base o circuito integrado LM 555 na sua configuração de multivibrador Astável com um controle adaptado ao seu pino 5:



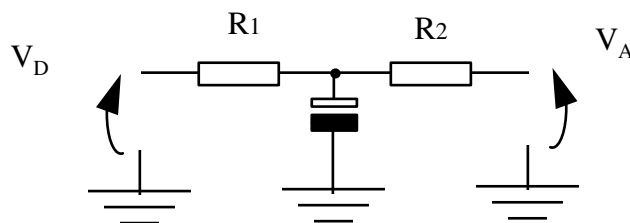
O transistor T conectado ao pino 5 do LM 555 tem a função de desbalancear a rede de resistores do integrado de acordo com a tensão aplicada e sua base. O resistor R_3 é apenas um limitador de corrente para a sua proteção. Como um transistor conduz corrente de coletor para emissor proporcionalmente à corrente aplicada em sua base, podemos dizer que ele apresenta uma resistência entre coletor e emissor inversamente proporcional à essa corrente. Não se esqueça que a corrente da base é resultado da tensão aplicada, isto é, só existe corrente se houver diferença de potencial (tensão) e ainda um caminho para que ela circule. Voltando à análise, para uma tensão relativamente alta aplicada em V_{in} , temos uma corrente também relativamente alta na base do transistor e uma corrente muito mais alta de coletor para emissor. Isso significa que a resistência entre coletor e emissor foi reduzida e portanto temos um resistor de valor baixo em paralelo com os resistores da rede do CI, alterando os valores de $1/3$ e $2/3$ de V_{CC} para valores mais baixos. Para uma tensão relativamente pequena aplicada em V_{in} , temos uma corrente relativamente baixa na base do transistor e conseqüentemente uma corrente baixa de coletor para emissor. Isso significa que o transistor se comportará como um resistor de valor muito alto em paralelo com a rede de resistores do CI e portanto as tensões $1/3$ e $2/3$ de V_{CC} praticamente não são alteradas. Como este oscilador funciona com carregando e descarregando o capacitor, quando ele atinge as tensões relativas às tensões da rede de resistores do CI, teremos na saída (pino 3) um sinal digital com variação na largura de seus pulsos em proporção ao sinal aplicado em V_{in} . Este tipo de circuito é também conhecido como modulador PWM (*Pulse Width Modulation* ou Modulador Largura de Pulso). A grande vantagem deste método é a alta velocidade de conversão que é praticamente simultânea às variações do sinal de entrada. Este tipo de conversor e alguns variantes é usado para a conversão de som do *Compact Disk Player*. Outra vantagem também muito importante e a fácil reconstituição do sinal original. Basta aplicar o sinal convertido a um filtro passa baixas, pois o valor médio do sinal digital é exatamente o sinal original. Explicando melhor, se um capacitor e um resistor forem ligados convenientemente (filtro passa baixas) para que recebam o sinal digital, teremos a carga do capacitor quando o sinal digital estiver em nível lógico 1 e a sua descarga quando este estiver em nível lógico 0. Outro ponto também interessante é que o circuito **Conversor A/D com quantização em Bytes** estudado ainda a pouco produz este mesmo tipo de modulação (ou conversão) na saída do comparador.

Os gráficos a seguir ilustram os valores das tensões importantes para o bom entendimento do conversor com o CI 555:



b) Conversão Digital/Analógica por demodulação de PWM

O circuito abaixo é um filtro passa baixas e pode demodular o sinal PWM pois fornece o valor médio deste sinal. Observe que quando a senoide está crescendo seu valor de tensão, os pulsos produzidos pelo modulador PWM vão se alargando e quando o valor de tensão da senoide está decrescendo, os pulsos do modulador PWM vão estreitando e portanto o valor médio dos pulsos nos dá a senoide como resultado.



Apêndice 1

Portas Lógicas

As grandes responsáveis pelo sucesso da Eletrônica Digital são: a simplicidade dos circuitos e a excelente *performance*. Como os circuitos operam com apenas dois níveis de sinais, fica fácil projetarmos circuitos que executem as funções de Boole. Vejamos o exemplo de alguns circuitos que executam **Funções Booleanas**:

Se a **diferença de potencial** entre *base* e *emissor* for 0v ($V_{in} = 0$), não vai haver circulação de corrente pela *base* ($i_b = 0$). Se não temos corrente na *base* do transistor, não existirá a corrente de *coletor* para *emissor* e, então, a tensão sobre o resistor será 0v ($V_{R2} = 0$). Concluímos, então que a tensão na saída será igual a V_{cc} ($V_{out} = V_{cc}$). Veja a figura 1.

Figura 1

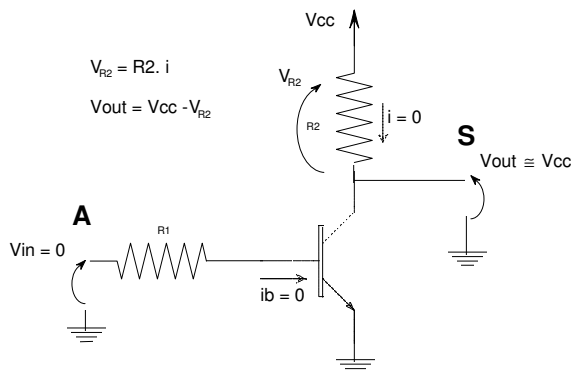
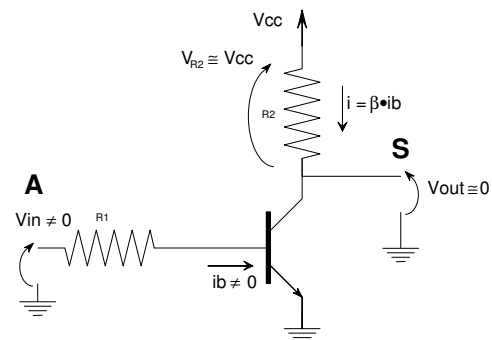


Figura 2



Na figura 2 é colocada uma tensão elétrica $V_{in} \neq 0$ na entrada do circuito. Isso provoca o aparecimento de uma corrente na *base* do transistor e então o transistor começa a conduzir uma forte corrente entre o *coletor* e *emissor*. Essa forte corrente ($i = \beta \cdot i_b$) faz surgir a tensão $V_{R2} = R2 \cdot i$ sobre o resistor $R2$. Pela lei de soma das tensões temos que:

$$V_{cc} = V_{R2} + V_{out} \Rightarrow V_{out} = V_{cc} - V_{R2}$$

$$V_{out} = V_{cc} - R2 \cdot i$$

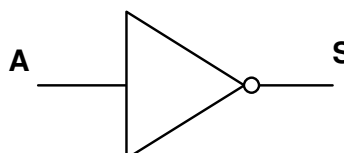
$$V_{out} = V_{cc} - R2 \cdot \beta \cdot i_b$$

Se calcularmos convenientemente os valores de $R1$ e $R2$ o circuito ilustrado das figuras 1 e 2 vai se comportar da seguinte maneira:

$$\text{se } V_{in} = 0, V_{out} \cong V_{cc}$$

$$\text{se } V_{in} = V_{cc}, V_{out} \cong 0$$

O comportamento do circuito é a própria execução da função **Complemento** da Álgebra Booleana e é chamado de **Porta Lógica Inversora** ou simplesmente **Inversor**. Como existem diversos tipos de transistores teremos vários tipos de circuitos que funcionam da mesma forma que este, mas a disciplina Eletrônica Digital I estuda apenas circuitos formados por **Portas Lógicas** e não os detalhes da sua construção. Por este motivo, temos uma simbologia própria para representar tais circuitos. Exemplo:



Inversor

Podemos, também, construir um circuito que execute a função **Igualdade** da Álgebra de Boole mesmo que isso possa nos parecer engraçado, já que a função igualdade poderia ser executada eletricamente por um simples pedaço de fio. É que as vezes precisamos dar um *ganho de potência* ao sinal elétrico que o circuito está operando e precisamos de um amplificador como, por exemplo, o circuito abaixo:

Figura 3

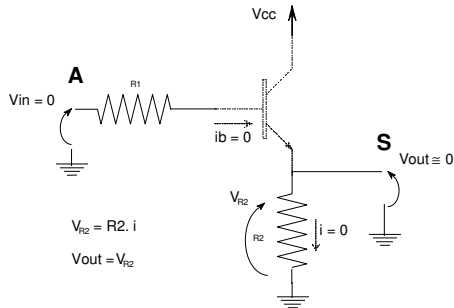
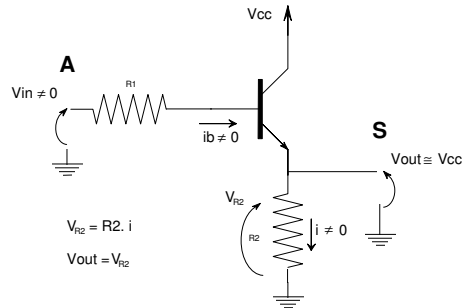


Figura 4



Se a tensão na entrada do circuito for zero ($V_{in} = 0$), a corrente de *base* será zero ($i_b = 0$). Nessas condições o transistor se encontra **aberto**, isto é, não conduz corrente entre *coletor* e *emissor*. Sendo assim, a tensão $V_{R2} = 0$ pois não circula corrente no resistor $R2$. Como podemos observar, no esquema elétrico desenhado acima, $V_{R2} = V_{out} = 0$.

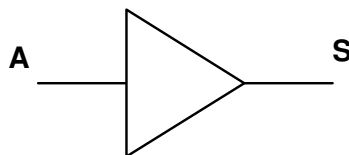
Analogamente, se a tensão na entrada do circuito for diferente de zero ($V_{in} \neq 0$), a corrente de *base* será diferente de zero ($i_b \neq 0$) e então o transistor **satura**, isto é, conduz uma forte corrente entre *coletor* e *emissor* ($i = \beta \cdot i_b + i_b$).

Se calcularmos convenientemente os valores de $R1$ e $R2$ o circuito ilustrado das figuras 3 e 4 vai se comportar da seguinte maneira:

se $V_{in} = 0$, $V_{out} \cong 0$

se $V_{in} = V_{cc}$, $V_{out} \cong V_{cc}$

Da mesma forma que o circuito das figuras 1 e 2 tem uma representação especial, o circuito das figuras 3 e 4 terá a sua:



Driver

Alguns autores chamam, erroneamente, esta porta lógica de *buffer*, mas *buffer* é *memória intermediária*. Podemos ser ainda mais ousados construindo um circuito capaz de executar a **Função E** da Álgebra Booleana:

Figura 5

Figura 6

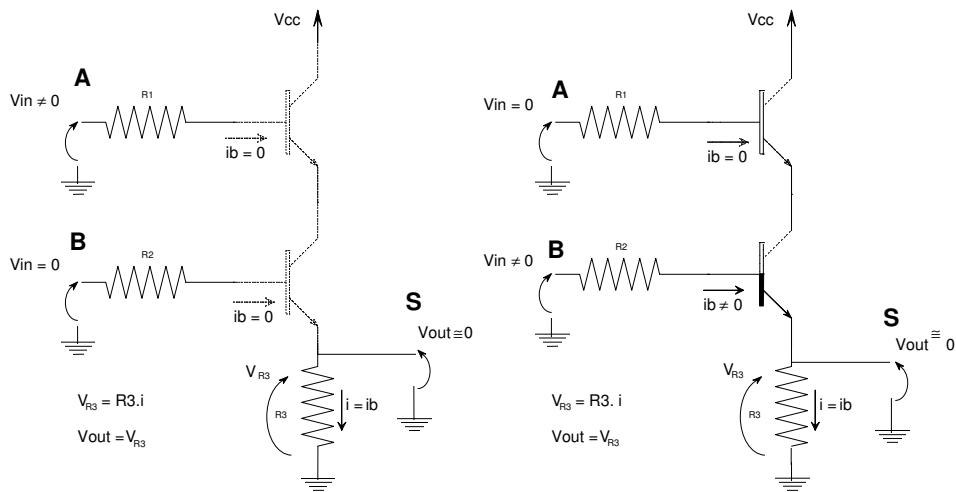
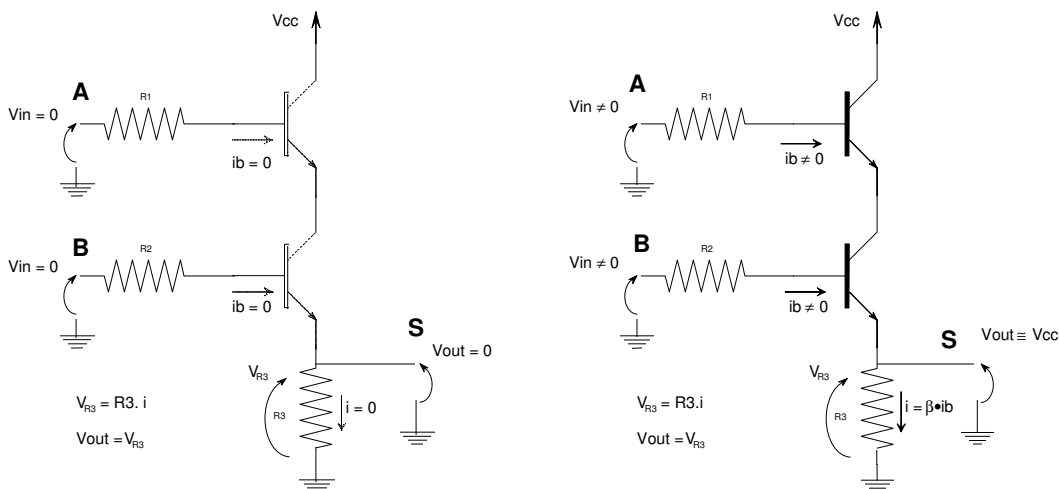
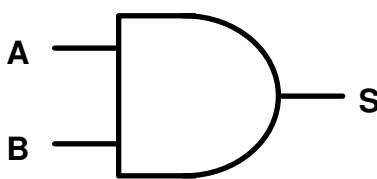


Figura 7

Figura 8



Analisando as figuras 5, 6, 7, e 8, concluímos que o circuito apresenta um nível de tensão considerável na saída somente quando ambas entradas estão com tensões diferentes de zero (Figura 8). Em todos outros casos a tensão V_{out} não atinge valores significativos, já que a corrente que atravessa o resistor na figura 6 é centena de vezes menor que a corrente que atravessa este resistor na figura 8. Da mesma forma que os outros circuitos, este também tem uma representação simbólica:



Porta E

Uma **Porta OU** seria facilmente construída com a dois diodos como indica a figura 9, mas os diodos tem uma queda de tensão característica e também uma resistência na sua *junção*. Para a maioria dos projetos, esses valores são desprezíveis, mas no nosso caso é um pouco problemático pois é muito comum associarmos as portas lógicas em *casca*, isto é, a saída de uma porta ligada a entrada de outra. Desta forma o sinal iria perdendo potência até não ficar mais confiável.

Podemos contornar este problema adicionando um transistor para dar ganho de potência, como mostra a figura 10, mas um circuito mais elaborado é mostrado na figura 11.

Figura 9

Figura 10

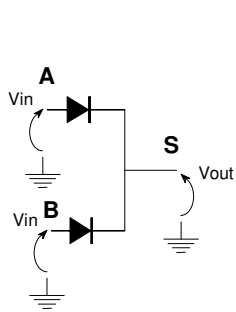


Figura 11

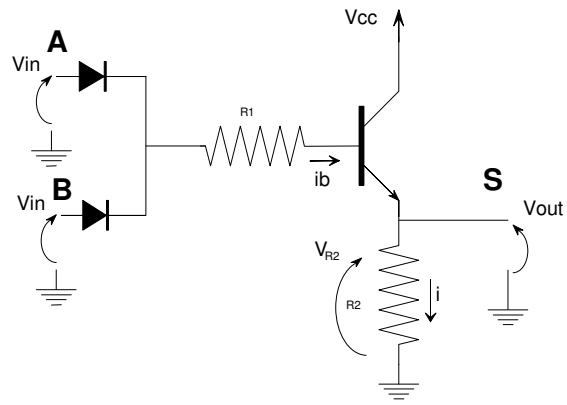


Figura 12

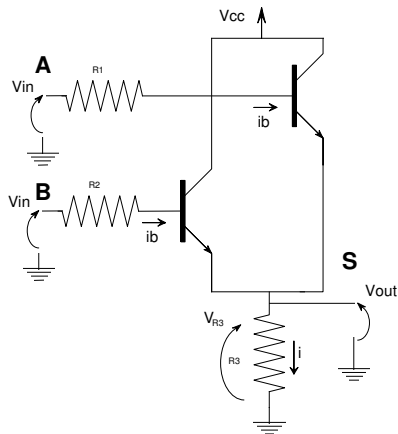
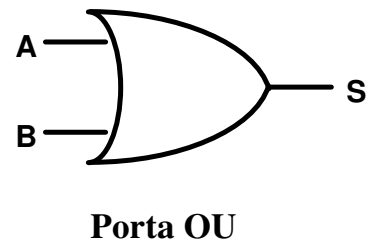


Figura 13



Podemos construir circuitos para as funções **NE** e **NOU** usando o mesmo artifício de trocar a posição do resistor em relação ao transistor (veja figuras 1 e 3).

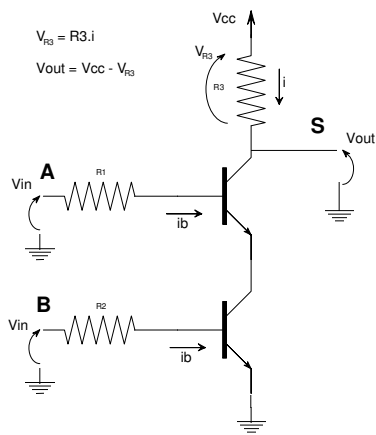


Figura 15

Figura 14

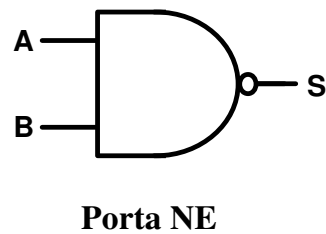
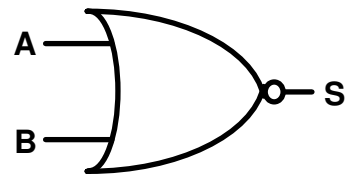
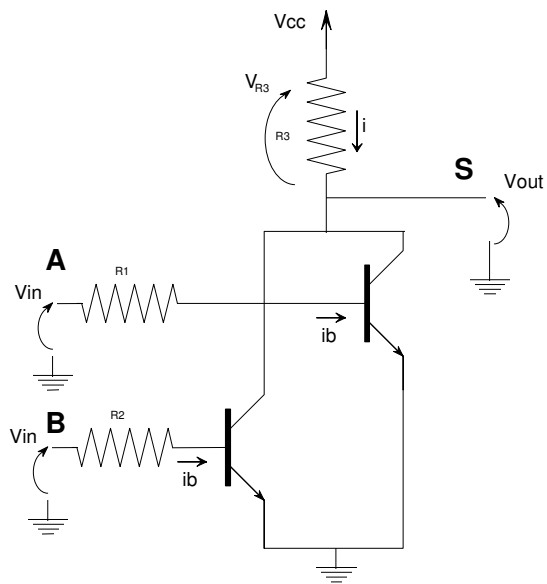


Figura 16



Porta NOU

Portas Lógicas Reais

Segue abaixo, exemplo dos catálogos do fabricante Texas Instruments™ que foram retirados do site da empresa (www.ti.com). Apenas o primeiro contém está completamente representado, tal e qual sua documentação



January 1998

CD54HC00, CD54HCT00, CD74HC00, CD74HCT00

High Speed CMOS Logic
Quad 2-Input NAND Gate

Features

- ✦ *Buffered Inputs*
- ✦ *Typical Propagation Delay: 7ns at $V_{CC} = 5V$
 $C_L = 15pF$, $T_A = 25^\circ C$*
- ✦ *Fanout (Over Temperature Range)*
 - *Standard Outputs* 10 LSTTL Loads
 - *Bus Driver Outputs* 15 LSTTL Loads
- ✦ *Wide Operating Temperature Range* ... $-55^\circ C$ to $125^\circ C$
- ✦ *Balanced Propagation Delay and Transition Times*
- ✦ *Significant Power Reduction Compared to LSTTL Logic ICs*
- ✦ *Alternate Source is Philips/Signetics*
- ✦ *HC Types*
 - 2V to 6V Operation
 - High Noise Immunity: $N_{IL} = 30\%$, $N_{IH} = 30\%$ of V_{CC} at $V_{CC} = 5V$
- ✦ *HCT Types*
 - 4.5V to 5.5V Operation
 - Direct LSTTL Input Logic Compatibility:
 $V_{IL} = 0.8V$ (Max), $V_{IH} = 2V$ (Min)
 $I_{OL} \pm 1mA$ at V_{OL} , V_{OH}
- ✦ *Related Literature*
 - CD54HC00F3A and CD54HCT00F3A Military Data Sheet, Document Number 3753

Description

The Harris CD54HC00, CD54HCT00, CD74HC00 and CD74HCT00 logic gates utilize silicon gate CMOS technology to achieve operating speeds similar to LSTTL gates with the low power consumption of standard CMOS integrated circuits. All devices have the ability to drive 10 LSTTL loads. The 74HCT logic family is functionally pin compatible with the standard 74LS logic family.

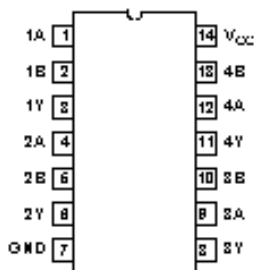
Ordering Information

PART NUMBER	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
CD74HC00E	-55 to 125	14 Ld PDIP	E14.3
CD74HCT00E	-55 to 125	14 Ld PDIP	E14.3
CD74HC00M	-55 to 125	14 Ld SOIC	M14.16
CD74HCT00M	-55 to 125	14 Ld SOIC	M14.16
CD54HC00F	-55 to 125	14 Ld CERDIP	F14.3
CD54HCT00F	-55 to 125	14 Ld CERDIP	F14.3
CD54HC00W	-55 to 125	Wafer	
CD54HCT00W	-55 to 125	Wafer	
CD54HC00H	-55 to 125	Die	
CD54HCT00H	-55 to 125	Die	

NOTE: When ordering, use the entire part number. Add the suffixes to obtain the variant in the tape and reel.

Pinout

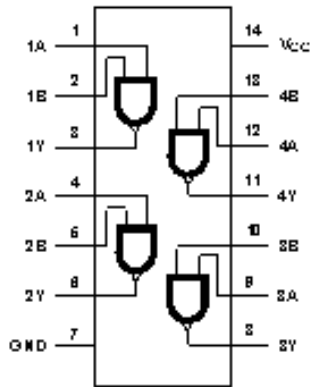
CD54HC00, CD54HCT00, CD74HC00, CD74HCT00
(PDIP/CERDIP, SOIC)
TOP VIEW



CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper IC Handling Procedures.
Copyright © Harris Corporation 1998

File Number **1464.2**

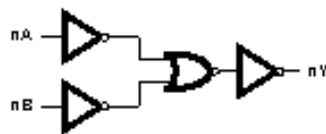
Functional Diagram



TRUTH TABLE

nA	INPUTS		OUTPUT	
	nB	nY		
L	L	L	H	
L	H	H	H	
H	L	L	H	
H	H	H	L	

Logic Symbol



CD54HC00, CD54HCT00, CD74HC00, CD74HCT00

DC Electrical Specifications (Continued)

PARAMETER	SYMBOL	TEST CONDITIONS		V_{CC} (V)	25°C			-40°C TO 85°C		-55°C TO 125°C		UNITS
		V_I (V)	I_O (mA)		MIN	TYP	MAX	MIN	MAX	MIN	MAX	
Quiescent Device Current	I_{CC}	V_{CC} or GND	0	6	-	-	2	-	20	-	40	nA
HCT TYPES												
High Level Input Voltage	V_{IH}	-	-	4.5 to 5.5	2	-	-	2	-	2	-	V
Low Level Input Voltage	V_{IL}	-	-	4.5 to 5.5	-	-	0.8	-	0.8	-	0.8	V
High Level Output Voltage CMOS Loads	V_{OH}	V_{IH} or V_{IL}	-	4.5	4.4	-	-	4.4	-	4.4	-	V
High Level Output Voltage TTL Loads			-0.02	4.5	3.98	-	-	3.84	-	3.7	-	V
Low Level Output Voltage CMOS Loads	V_{OL}	V_{IH} or V_{IL}	-4	4.5	-	-	0.1	-	0.1	-	0.1	V
Low Level Output Voltage TTL Loads			0.02	4.5	-	-	0.26	-	0.33	-	0.4	V
Input Leakage Current	I_I	V_{CC} and GND	4	5.5	-	-	±0.1	-	±1	-	±1	nA
Quiescent Device Current	I_{CC}	V_{CC} or GND	0	5.5	-	-	2	-	20	-	40	nA
Additional Quiescent Device Current Per Input Pin: 1 Unit Load	D_{CC}	V_{CC} -2.1	-	4.5 to 5.5	-	100	350	-	450	-	450	nA

NOTE:

- For data supply systems theoretical worst case ($V_I = 2.4V, V_{CC} = 5.5V$) specification is 1.8mA.

HCT Input Loading Table

INPUT	UNIT LOADS
A	1.8
B	1.1

NOTE: Unit Load is D_{CC} limit specified in DC Electrical Specifications table, e.g. 350nA max at 25°C.

Switching Specifications Input $t_r = 6ns$

PARAMETER	SYMBOL	TEST CONDITIONS	V_{CC} (V)	25°C			-40°C TO 85°C		-55°C TO 125°C		UNITS
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
HCT TYPES											
Propagation Delay, Input to Output (Figure 1)	t_{PLH}, t_{PHL}	$C_L = 50pF$	2	-	-	90	-	115	-	135	ns
			4.5	-	-	18	-	23	-	27	ns
			6	-	-	15	-	20	-	23	ns
Propagation Delay, Data Input to Output Y	t_{PLH}, t_{PHL}	$C_L = 15pF$	5	-	7	-	-	-	-	pF	

CD54HC00, CD54HCT00, CD74HC00, CD74HCT00

Switching Specifications (Input $t_r = t_f = 6ns$) (Continued)

PARAMETER	SYMBOL	TEST CONDITIONS	V _{CC} (V)	25°C			-40°C TO 85°C		-55°C TO 125°C		UNITS
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
Transition Times (Figure 1)	t _{PLH} , t _{PHL}	C _L = 50pF	2	-	-	75	-	95	18	110	ns
			4.5	-	-	15	-	19	-	22	ns
			6	-	-	13	-	16	-	19	ns
Input Capacitance	C _I	-	-	-	10	-	10	-	10	pF	
Power Dissipation Capacitance (Notes 3, 4)	C _{PD}	-	5	-	25	-	-	-	-	pF	
HCT TYPES											
Propagation Delay, Input to Output (Figure 2)	t _{PLH} , t _{PHL}	C _L = 50pF	4.5	-	-	20	-	25	-	30	ns
Propagation Delay, Data Input to Output	t _{PLH} , t _{PHL}	C _L = 15pF	5	-	8	-	-	-	-	-	pF
Transition Times (Figure 2)	t _{PLH} , t _{PHL}	C _L = 50pF	4.5	-	-	15	-	19	-	22	ns
Input Capacitance	C _I	-	-	-	10	-	10	-	10	pF	
Power Dissipation Capacitance (Notes 3, 4)	C _{PD}	-	5	-	25	-	-	-	-	pF	

NOTES:

3. C_{PD} is used to determine the dynamic power consumption per gate.

4. $P_D = V_{CC}^2 f (C_{PD} + C_I)$ where f = input frequency, C_L = output load capacitance, V_{CC} = supply voltage.

Test Circuits and Waveforms

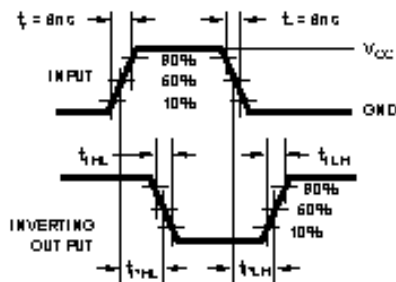


FIGURE 1. HC AND HCU TRANSITION TIMES AND PROPAGATION DELAY TIMES, COMBINATION LOGIC

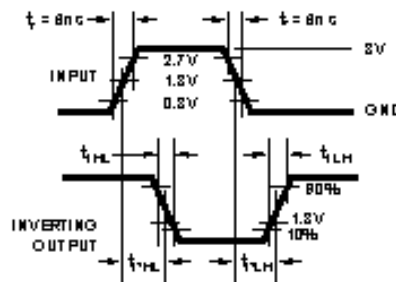


FIGURE 2. HCT TRANSITION TIMES AND PROPAGATION DELAY TIMES, COMBINATION LOGIC

INPUT LEVEL	HC TYPES	HCT TYPES
	V _{CC}	3V
V _S	50% V _{CC}	1.3V

NOTE: Transition times and propagation delay times.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Copyright © 1999, Texas Instruments Incorporated

SN54AHC02, SN74AHC02 QUADRUPLE 2-INPUT POSITIVE-NOR GATES

SC 18254H ± DECEMBER 1996 ± REVISED JANUARY 2000

EPIC[®] (Enhanced-Performance Implanted CMOS) Process

Operating Range 2-V to 5.5-V V_{CC}

Latch-Up Performance Exceeds 250 mA Per JESD 17

ESD Protection Exceeds 2000 V Per

MIL-STD-883, Method 3015; Exceeds 200 V Using Machine Model (C = 200 pF, R = 0)

Package Options Include Plastic

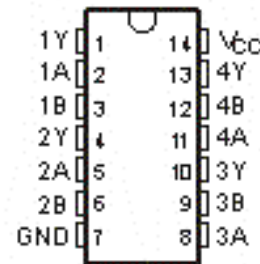
Small-Outline (D), Shrink Small-Outline (DB), Thin Very Small-Outline (DGV), Thin Shrink Small-Outline (PW), and Ceramic Flat (W) Packages, Ceramic Chip Carriers (FK), and Standard Plastic (N) and Ceramic (J) DIPs

description

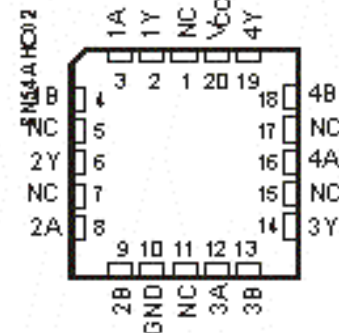
The 'AHC02 devices contain four independent 2-input NOR gates that perform the Boolean function $Y = \overline{A + B}$ or $Y = \overline{A} \cdot \overline{B}$ in positive logic.

The SN54AHC02 is characterized for operation over the full military temperature range of $\pm 55^{\circ}\text{C}$ to 125°C . The SN74AHC02 is characterized for operation from $\pm 40^{\circ}\text{C}$ to 85°C .

SN54AHC02 . . . JORGAP PACKAGE
SN74AHC02 . . . D, DB, DGV, N, OR PW PACKAGE
(TOP VIEW)



. . . FK PACKAGE
(TOP VIEW)



NC ± No Internal Connection

FUNCTION TABLE
(each gate)

INPUTS		OUTPUT
A	B	Y
H	X	L
X	H	L
L	L	H



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

EPIC is a trademark of Texas Instruments Incorporated.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



POST OFFICE BOX 655009 DALLAS, TEXAS 75265

Copyright © 2000, Texas Instruments Incorporated
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

Designed Specifically for High-Speed
Memory Decoders and Data Transmission
Systems

Incorporates Three Enable Inputs to
Simplify Cascading and/or Data Reception

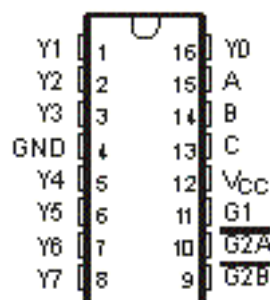
Center-Pin V_{CC} and GND Configurations
Minimize High-Speed Switching Noise

EPIC® (Enhanced-Performance Implanted
CMOS) 1- μ m Process

500-mA Typical Latch-Up Immunity at
125°C

Package Options Include Plastic
Small-Outline (D) and Thin Shrink
Small-Outline (PW) Packages, and
Standard Plastic 300-mil DIPs (N)

D, N, OR PW PACKAGE
(TOP VIEW)



description

The 74AC11138 circuit is designed to be used in high-performance memory-decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems, this decoder can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast enable circuit, the delay times of this decoder and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The conditions at the binary-select (A, B, C) inputs and the three enable ($\overline{G1}$, $\overline{G2A}$, $\overline{G2B}$) inputs select one of eight output lines. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The 74AC11138 is characterized for operation from $\pm 40^{\circ}\text{C}$ to 85°C .

FUNCTION TABLE

ENABLE INPUTS			SELECT INPUTS			OUTPUTS							
G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	H	L	H	H	H	H	H	L	H	H
H	L	L	H	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductors products and disclaimers thereto appears at the end of this data sheet.

EPIC is a trademark of Texas Instruments Incorporated.

PRODUCTION DATA information is current as of publication date.
Products conform to specifications per the terms of Texas Instruments
standard warranty. Production processing does not necessarily include
testing of all parameters.

Copyright © 1996, Texas Instruments Incorporated

 **TEXAS
INSTRUMENTS**

POST OFFICE BOX 655008 DALLAS, TEXAS 75265
POST OFFICE BOX 1442 HOUSTON, TEXAS 77251-1442

Formas-Padrão Para Funções Lógicas

As duas **Formas-Padrão** para as funções lógicas que estudaremos tem com objetivo facilitar a montagem de tabelas verdade de qualquer função. Se não tivéssemos estas **Formas-Padrão** teríamos que testar a função para todas as combinações possíveis dos valores de sua variáveis.

1) Soma Padrão de Produtos

Uma função escrita nesta forma será uma combinação de funções **E** associadas entre si através da função **OU**. Exemplos:

a) Dada a função lógica abaixo, exprimi-la como uma soma padrão de produtos:

$$F = (\bar{A} + BC)(B + CD)$$

Solução usando a lei distributiva

$$F = \bar{A}B + \bar{A}CD + BCB + BCCD$$

$$F = \bar{A}B + \bar{A}CD + BC + BCD$$

b) Dada a função lógica abaixo, exprimi-la como uma soma padrão de produtos:

$$F = (A + \overline{BC})(\overline{D + BE})$$

Solução usando o Teorema de De Morgan e a lei distributiva

$$F = (A + \bar{B} + \bar{C})(\bar{D}(\overline{BE}))$$

$$F = (A + \bar{B} + \bar{C})(\bar{D}(\bar{B} + \bar{E}))$$

$$F = (A + \bar{B} + \bar{C})(\bar{D}\bar{B} + \bar{D}\bar{E})$$

$$F = A\bar{D}\bar{B} + A\bar{D}\bar{E} + \bar{B}\bar{D}\bar{B} + \bar{B}\bar{D}\bar{E} + \bar{C}\bar{D}\bar{B} + \bar{C}\bar{D}\bar{E}$$

$$F = A\bar{B}\bar{D} + A\bar{D}\bar{E} + \bar{B}\bar{D} + \bar{B}\bar{D}\bar{E} + \bar{B}\bar{C}\bar{D} + \bar{C}\bar{D}\bar{E}$$

É sempre possível escrever uma função lógica como uma simples *soma* de termos, sendo cada termo um *produto* de variáveis, basta seguirmos as regras:

a) Se apenas variáveis individuais aparecem complementadas, precisamos apenas da lei distributiva (primeiro exemplo).

b) Se um sinal de complemento aparecer sobre uma combinação de variáveis, precisamos primeiro usar o Teorema de De Morgan até que tenhamos apenas variáveis individuais complementadas e então aplicarmos a regra citada acima (segundo exemplo).

Exercícios:

Rescrever as funções abaixo como uma soma padrão de produtos

$$a) F = (\overline{A+B})(AC+D)$$

$$b) F = (\overline{AB}+C)(\overline{CD}+F)$$

$$c) F = A(\overline{DC})$$

$$d) F = A(\overline{B+C})+BD$$

Se uma função está escrita na forma soma padrão de produtos e cada termo desta função é composto por **todas** as variáveis que fazem parte da função, dizemos que esta função é uma **Soma Padrão de Produtos Mínimos** e que cada termo é um **Minitermo**. Exemplo:

$$F = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC$$

Qualquer função booleana pode ser expressa nesta forma, basta seguirmos as seguintes regras:

- Transcrever a função na forma Soma Padrão de Produtos
- Ao identificar a ausência de uma variável em um de seus termos, multiplique-o pela soma desta variável com seu complemento.

Exemplo:

$$F = A + \overline{B}C \quad (\text{Obs. a função já está na forma Soma Padrão de Produtos})$$

$$F = A(B + \overline{B})(C + \overline{C}) + \overline{B}C(A + \overline{A})$$

$$F = A(BC + \overline{B}C + B\overline{C} + \overline{B}\overline{C}) + \overline{B}CA + \overline{B}\overline{C}\overline{A}$$

$$F = ABC + \overline{A}BC + A\overline{B}C + \overline{A}\overline{B}C + \overline{A}B\overline{C} + \overline{A}\overline{B}\overline{C} + \overline{A}BC$$

$$F = ABC + \overline{A}BC + A\overline{B}C + \overline{A}\overline{B}C + \overline{A}BC$$

A partir de uma função escrita na forma Soma Padrão de Produtos Mínimos podemos facilmente montar a tabela verdade que a define, basta considerarmos as variáveis sem complemento como 1 e as complementadas como 0. Para exemplificar montemos a tabela verdade a última função que analisamos:

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

$F = ABC + \overline{A}BC + A\overline{B}C + \overline{A}\overline{B}C + \overline{A}BC$

Exercícios:

Montar as tabelas verdade das funções abaixo transformando-as antes em Somas Padrão de Produtos Mínimos:

- a) $F = (A + B)(\bar{A} + \bar{B})$
 b) $F = (A + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C})$
 c) $F = AB(C + D)E + (C + D)AC$
 d) $F = A(B \oplus C) + B \odot C(\overline{A + B}) + \overline{ABC}$

2) Produto Padrão de Somas

Toda função lógica pode ser expressa como um *produto* de termos, onde cada termo é uma *soma* e estas *somas* são compostas por **todas** as variáveis pertinentes à função. Cada termo de uma função escrita na forma Produto Padrão de Somas será chamado de **Maxitermo**.

O processo para a obtenção desta forma padrão é exemplificado abaixo:

$$F = A + \bar{B}C$$

Escrever a função na forma Soma Padrão de Produtos Mínimos:

$$F = A(B + \bar{B})(C + \bar{C}) + \bar{B}C(A + \bar{A})$$

$$F = A(BC + \bar{B}C + B\bar{C} + \bar{B}\bar{C}) + \bar{B}CA + \bar{B}\bar{C}\bar{A}$$

$$F = ABC + \bar{A}BC + AB\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}BC + \bar{A}\bar{B}\bar{C}$$

$$F = ABC + \bar{A}BC + AB\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$$

Montar a tabela verdade correspondente e escrever a **função inversa** a partir desta tabela:

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

$$\bar{F} = \bar{A}BC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$$

Retornar à função F, "barrando" a última função:

$$\bar{F} = \bar{A}BC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$$

$$F = \overline{\bar{A}BC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}}$$

$$F = \overline{\bar{A}BC} \cdot \overline{\bar{A}\bar{B}C} \cdot \overline{\bar{A}\bar{B}\bar{C}}$$

$$F = (A + B + C)(A + \bar{B} + C)(A + \bar{B} + \bar{C})$$

Exercícios:

Rescrever as funções abaixo na forma Produto Padrão de Somas:

a) $F = \bar{A}B + A\bar{B}$

b) $F = \bar{A}\bar{B} + A\bar{B}$

c) $F = C + A\bar{B}$

d) $F = (\overline{A + B})C + AC$

Bibliografia

- IODETA, IVÁ VALEIJE, 1957
Elementos de Eletrônica *Digital*
Ivan V. Iodeta
Francisco G. Capuano
Ed. Érica - São Paulo
- TAUB, HERBERT, 1918
Circuitos Digitais e *Microprocessadores*
Herbert Taub
Ed. McGraw do Brasil
- TAUB, HERBERT, 1918
Eletrônica Digital
Herbert Taub
Donald Schilling
Ed. McGraw do Brasil
- THANENBAUN, ANDREW, 1939
Organização Estruturada de *Computadores*
Andrew Thanenbaun
Ed. Prentice Hall do Brasil