

Nomes: Felipe Carvalho Coronado Oropeza, Kevin Bezerra Soares, Albert Vinicios De Oliveira Macedo, Giovani Sant'Ana Ferreira Pedroso Vaz e Angelo R R Santos

Turma: 1 ADS manhã

Teoria das Tabelas-Verdade

A tabela-verdade é uma ferramenta fundamental na lógica proposicional e na álgebra booleana, utilizada para determinar o valor lógico de expressões compostas a partir de proposições simples. Ela permite visualizar todas as possíveis combinações de valores verdadeiros (V) ou falsos (F) das proposições envolvidas e o resultado dessas combinações através de operadores lógicos.

Conceitos Básicos

Proposição: Uma declaração que pode ser classificada como verdadeira ou falsa, mas não ambas simultaneamente.

Operadores Lógicos: Símbolos ou palavras que conectam proposições para formar expressões mais complexas. Os principais operadores são:

Negação (\neg): Inverte o valor lógico de uma proposição.

Conjunção (\wedge): É verdadeira apenas se ambas as proposições forem verdadeiras.

Disjunção (\vee): É verdadeira se pelo menos uma das proposições for verdadeira.

Condicional (\rightarrow): É falsa somente quando a primeira proposição é verdadeira e a segunda é falsa.

Bicondicional (\leftrightarrow): É verdadeira quando ambas as proposições têm o mesmo valor lógico.

Construção de uma Tabela-Verdade

1. Identificar as Proposições Simples: Determine todas as proposições básicas envolvidas na expressão.

2. Calcular as Combinações: Para n proposições, existem (2^n) combinações possíveis de valores verdadeiros e falsos.
3. Preencher a Tabela: Crie colunas para cada proposição simples e para cada subexpressão intermediária, terminando com a expressão completa.
4. Aplicar os Operadores: Utilize as regras dos operadores lógicos para determinar o valor de cada linha na tabela.

Exemplo

Considere as proposições P e Q , e a expressão lógica $P \vee Q$.

P	Q	$P \vee Q$
V	V	V
V	F	V
F	V	V
F	F	F

A tabela mostra que a disjunção $P \vee Q$ é falsa apenas quando ambas as proposições são falsas

Aplicações das Tabelas-Verdade

Validação de Argumentos Lógicos: Determinar se um argumento é válido verificando se, em todas as situações em que as premissas são verdadeiras, a conclusão também é verdadeira.

Simplificação de Expressões Lógicas: Identificar redundâncias ou simplificar circuitos lógicos em engenharia elétrica e computação.

Programação e Circuitos Digitais: Projeto e análise de circuitos digitais, onde os níveis de tensão correspondem a valores lógicos.

Importância na Computação e Matemática

As tabelas-verdade são essenciais na computação, especialmente na arquitetura de computadores e na programação, pois os computadores operam com valores binários (0 e 1), correspondentes aos valores lógicos falso e verdadeiro. Elas permitem o entendimento e a implementação de portas lógicas e circuitos digitais que formam a base do hardware computacional.

Conclusão teoria tabelas-verdade

A teoria das tabelas-verdade é um componente crucial no estudo da lógica, proporcionando uma base para a compreensão de sistemas complexos de raciocínio e cálculo. Ao dominar o uso das tabelas-verdade, é possível analisar e construir argumentos lógicos sólidos, simplificar sistemas booleanos e aprofundar o entendimento dos fundamentos da computação.

Criação de circuitos a partir de uma tabela verdade

A construção de um circuito lógico combinatório a partir de uma tabela verdade é um processo fundamental na eletrônica digital. Um circuito lógico combinatório é aquele no qual a saída depende apenas dos valores atuais das entradas, sem memória de estados anteriores. O processo de construção pode ser descrito nos seguintes passos teóricos:

Entendimento da Tabela Verdade

A tabela verdade é uma representação tabular que descreve a relação entre todas as combinações possíveis de entradas e suas correspondentes saídas para um sistema lógico. Cada linha da tabela corresponde a uma combinação diferente de valores de entrada (0 ou 1), e a coluna de saída mostra o resultado esperado para aquela combinação.

Exemplo de tabela verdade para três variáveis de entrada (A, B, C):

A	B	C	Saída
---	---	---	-------

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

Expressão Booleana a partir da Tabela Verdade:

A partir da tabela verdade, podemos deduzir uma função booleana que descreve o comportamento do circuito. Isso pode ser feito através de somas de produtos (também conhecido como forma canônica disjuntiva), onde cada linha que resulta em uma saída 1 é representada por um produto (*AND*) das entradas correspondentes, e esses produtos são somados (*OR*).

Para cada linha onde a saída é 1, criamos uma expressão booleana. Se a entrada for 0, usamos o complemento da variável.

No exemplo acima, as saídas 1 ocorrem nas linhas 2, 4, 5, 7 e 8. As expressões correspondentes são:

Linha 2: $\neg A. \neg B.C$

Linha 4: $\neg A.B.C$

Linha 5: $A. \neg B. \neg C$

Linha 7: $A.B. \neg C$

Linha 8: $A.B.C$

A expressão booleana final é a soma desses termos:

$$\text{Saída} = (\neg A. \neg B.C) + (\neg A.B.C) + (A. \neg B. \neg C) + (A.B. \neg C) + (A.B.C)$$

Simplificação da Função Booleana

Para otimizar o circuito, é importante simplificar a expressão booleana. Isso pode ser feito utilizando técnicas como o mapa de Karnaugh (K-map) ou as leis de De Morgan e outras identidades algébricas da álgebra booleana. A simplificação reduz o número de portas lógicas necessárias.

Implementação com Portas Lógicas

Uma vez obtida a expressão booleana, o próximo passo é converter essa expressão em um circuito lógico utilizando portas lógicas básicas como *AND*, *OR* e *NOT*. Cada operação booleana é mapeada para a respectiva porta

$AND (.) \rightarrow$ Porta AND

$OR (+) \rightarrow$ Porta OR

$NOT (\neg) \rightarrow$ Porta NOT

A implementação do exemplo acima utilizaria 5 portas AND, 1 porta OR (para somar as saídas das ANDs), e portas NOT onde necessário para inverter as variáveis.

Desenho do Circuito Lógico

Com as portas lógicas identificadas, desenha-se o circuito ligando as entradas às portas e interconectando-as conforme a expressão booleana. Cada conjunto de portas *AND*, *OR* e *NOT* deve ser conectado para realizar as operações descritas pela função booleana.

Verificação

Após o circuito ser construído, ele deve ser verificado para garantir que a saída para cada combinação de entradas corresponda à tabela verdade original. Isso pode ser feito simulando o circuito ou construindo-o fisicamente.

Conclusão Criação de circuitos a partir de uma tabela verdade

A construção de um circuito lógico combinatório a partir de uma tabela verdade envolve a interpretação da tabela para derivar uma expressão booleana, a simplificação dessa

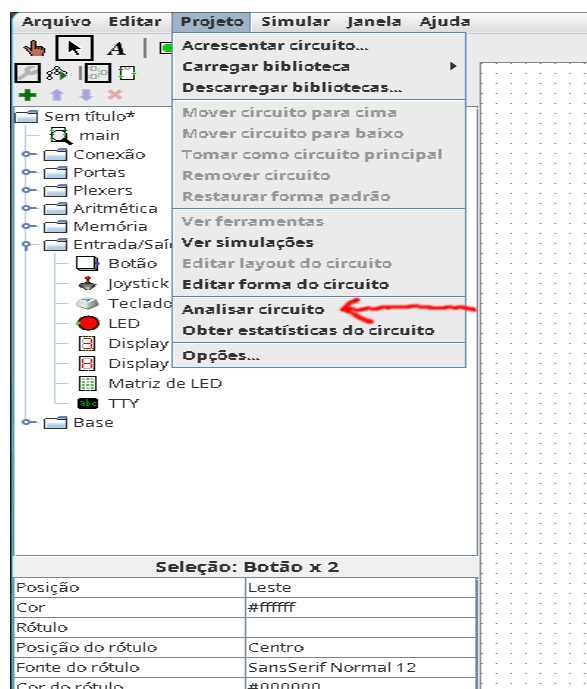
expressão, e a implementação da expressão com portas lógicas. Esse processo é central para o design de circuitos digitais em diversos dispositivos eletrônicos.

Criação do circuito no logisim

Para a criação do circuito, primeiro é necessário definir sua funcionalidade. O circuito apresentado aqui tem como função acender os LEDs de um display de 7 segmentos, indicando qual botão foi pressionado (b1, b2, b3... b9). Caso dois botões sejam pressionados simultaneamente, o display se desligará. Com a funcionalidade definida, podemos iniciar a criação do circuito.

Criação

Para facilitar o desenvolvimento, o Logisim possui a função de criar um circuito com base em uma tabela verdade. Para acessar esse menu, vá até o canto superior esquerdo, clique em 'Projeto' e, em seguida, em 'Analisar Circuito', como mostrado na imagem na página seguinte.



Será aberto um menu onde podemos definir para o Logisim o funcionamento do circuito. A primeira etapa é fazer a definição das entradas. Na aba de entradas, podemos adicionar as entradas, bastando apenas dar um nome a elas e, depois, pressionar o botão 'Acrescentar'. Repita esse procedimento para todos os 10 botões (0 a 9). Para adicionar as saídas, o processo é o mesmo: vá até a aba de saídas, dê um nome para a saída e pressione o botão 'Acrescentar'.

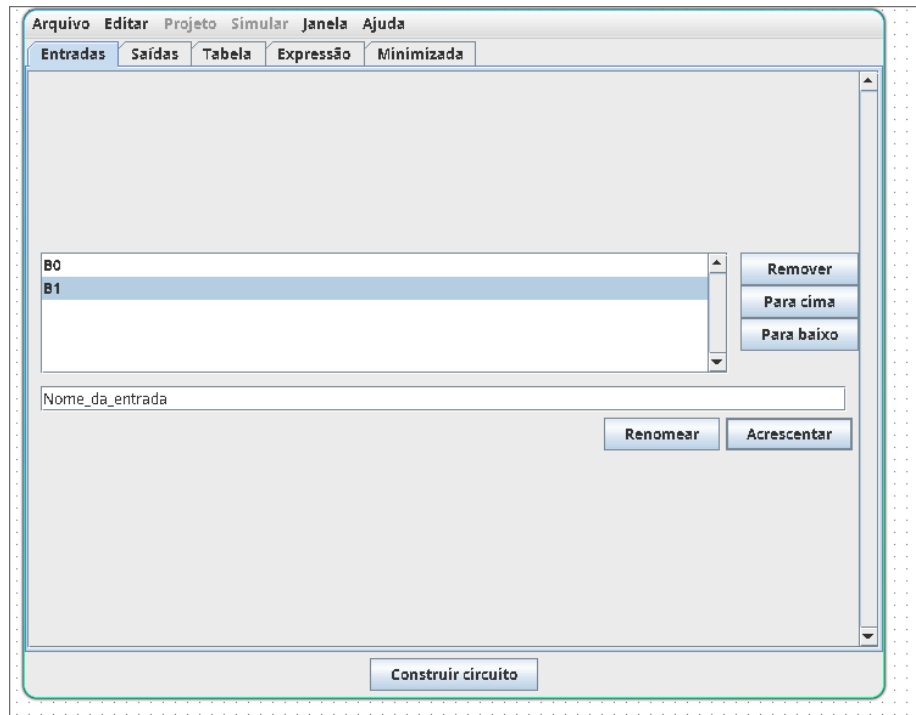


Tabela verdade

A próxima etapa é a construção da tabela verdade. Vá para a aba 'Tabela'. O primeiro passo é modificar os valores de todas as saídas para zero. A maneira mais rápida de fazer isso é: coloque todos os valores da primeira linha em zero (basta clicar no 'X'), depois copie essa linha e cole na linha abaixo. Selecione as duas linhas, copie e cole novamente. Depois, selecione as quatro linhas, copie e cole, e assim sucessivamente.

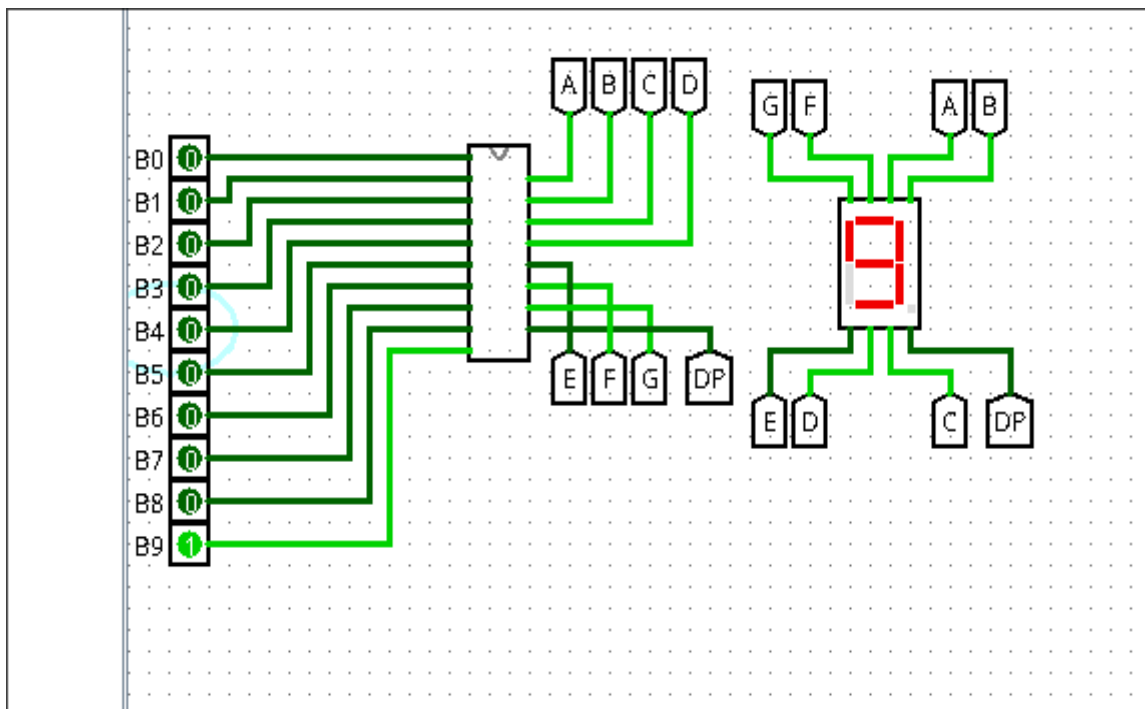
Após ter colocado todos os valores da tabela em zero, podemos alterar as linhas correspondentes aos botões pressionados. À esquerda estão os valores das entradas, e devemos modificar apenas as linhas onde apenas uma entrada está ativa. Na parte superior, estão os botões que correspondem aos dígitos. Preencha os valores da esquerda usando a tabela abaixo como guia.

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

Após isso, pressione o botão 'Criar Circuito', dê um nome ao circuito, e pronto: o circuito está concluído.

Finalizando

Selecione o circuito principal e adicione o circuito que você acabou de criar. Adicione os botões e o display conforme mostrado abaixo, e o circuito estará finalizado



Conclusão

A construção de um circuito lógico combinatório a partir de uma tabela verdade envolve a interpretação da tabela para derivar uma expressão booleana, a simplificação dessa expressão e a implementação da expressão com portas lógicas. Esse processo é central para o design de circuitos digitais em diversos dispositivos eletrônicos. A teoria das tabelas-verdade, sendo um componente crucial no estudo da lógica, proporciona uma base sólida para a compreensão de sistemas complexos de raciocínio e cálculo. Ao dominar o uso das tabelas-verdade, é possível analisar e construir argumentos lógicos robustos, simplificar sistemas booleanos e aprofundar o entendimento dos fundamentos da computação, contribuindo assim para o desenvolvimento de sistemas digitais eficientes.