

Sistema de Alarme Residencial, Comercial e/ou Automobilístico

Danielle A. Oliveira¹ & André L.B. Cavalcante²

Graduanda¹ e Professor DSc².

danielle33162@upis.br, andre02592@upis.br

Dept. Informática. UPIS Faculdades Integradas.

Resumo: O artigo apresenta a aplicação da Álgebra de Boole na construção de um alarme residencial, comercial ou automotivo. Descreve axiomas e teoremas booleanos, além de apresentar os passos e conceitos necessários para a construção do alarme. O alarme é acionado quando uma porta ou janela é aberta indevidamente, e apenas é desarmado quando a senha binária do sistema é digitada. Trata-se de um artigo didático que utiliza os conceitos da lógica digital para combiná-los com a arquitetura de computadores, transformando teoria em prática.

1. INTRODUÇÃO

Este artigo descreve a construção de um sistema de alarme que pode ser utilizado em residência, comércio ou automóvel. O alarme possui uma senha de acesso que faz com que o mesmo pare de disparar quando esta é acionada. Para o desenvolvimento do mesmo é necessário o conhecimento sobre circuitos lógicos e, também, sobre simplificações de expressões booleanas. Estas simplificações podem ser feitas por meio de alguns métodos (Cavalcante, 2004). Este artigo trata da simplificação por meio do método da Álgebra de Boole.

Compreende-se tensão elétrica como sinônimo de diferença de potencial, cuja unidade de medida é o volt (V). Os circuitos digitais são projetados para produzir tensões de saída que se encontrem dentro das faixas de tensões determinadas para os níveis 0 e 1. Isso significa dizer que um circuito digital responderá da mesma forma para tensões de entrada que se encontrem dentro da faixa permitida para o nível 0 e que não fará distinção entre tensões de entrada que estejam dentro da faixa permitida para o nível 1.

Denomina-se lógica do circuito o modo como um circuito digital responde a uma entrada. Cada tipo de circuito digital obedece a um determinado conjunto de regras lógicas, por essa razão, também, são conhecidos como circuitos lógicos.

2. ALGEBRA DE BOOLE

A Álgebra de Boole foi desenvolvida no século XIX pelo matemático inglês George Boole, que a denominou originalmente de álgebra lógica (Daghlian, 1986). Deste trabalho se derivaram campos importantes da Matemática, como o cálculo de proposições e a álgebra de conjuntos.

George Boole construiu a lógica por meio de conectivos, representando as expressões por letras e ligando-as por meio desses conectivos.

A álgebra de Boole é uma ferramenta matemática relativamente simples que permite descrever as relações entre as saídas dos circuitos lógicos e as entradas como uma equação algébrica, ou ainda, como uma função booleana.

A principal diferença entre a álgebra booleana e a matemática convencional é que as variáveis booleanas podem conter apenas dois valores possíveis 0 ou 1, que muitas vezes representam o nível de tensão presente em um terminal de entrada ou saída. A Tabela 1 apresenta as várias representações para os níveis lógicos 0 e 1.

Tabela 1 – Representação dos níveis lógicos

Nível Lógico 0	Nível Lógico 1
Falso	Verdadeiro
Desligado	Ligado
Não	Sim
Chave Aberta	Chave Fechada

A Álgebra Booleana possui 3 operações básicas, AND, OR e NOT que são denominadas operações lógicas e são representadas neste artigo pelos símbolos +, · e ', respectivamente.

2.1 Axiomas Booleanos

A seguir são representados os axiomas booleanos. (B, +, ·) é uma álgebra de Boole quando e, somente quando, $\forall x, y \in B$, valem os axiomas:

- I) $x + y \in B$
- II) $x \cdot y \in B$
- III) $x + y = y + x$
- IV) $x \cdot y = y \cdot x$
- V) $x \cdot (y + z) = xy + xz$
- VI) $x + (y \cdot z) = (x + y) \cdot (x + z)$
- VII) $\exists 0 \in B / x + 0 = 0 + x = x$
- VIII) $\exists 1 \in B / x \cdot 1 = 1 \cdot x = x$
- IX) $\exists x' \in B / x' + x = 1$ e $x \cdot x' = 0$

2.2 Teoremas Booleanos

Seguem as demonstrações dos principais teoremas booleanos.

Teorema 1: (Princípio da dualidade) Os resultados dos Axiomas Booleanos permanecem válidos se substituirmos + por ·, e 0 por 1.

Dem.: A prova deste teorema segue da simetria dos axiomas em relação aos dois operadores binários.

Teorema 2: $x + x = x, \forall x \in B$

Dem.:

$$\begin{aligned} x + x &= (x + x) \cdot 1 \\ &= (x + x) \cdot (x + x') \\ &= x + (x \cdot x') \\ &= x + 0 \\ &= x \end{aligned}$$

Teorema 3: $x \cdot x = x, \forall x \in B$

Dem.:

$$\begin{aligned} x \cdot x &= (x \cdot x) + 0 \\ &= (x \cdot x) + (x \cdot x') \\ &= x \cdot (x + x') \\ &= x \cdot 1 \\ &= x \end{aligned}$$

Teorema 4: $x + 1 = 1, \forall x \in B$

Dem.:

$$\begin{aligned} x + 1 &= (x + 1) \cdot 1 \\ &= (x + 1) \cdot (x + x') \\ &= x + (1 \cdot x') \\ &= x + x' \\ &= 1 \end{aligned}$$

Teorema 5: $x \cdot 0 = 0, \forall x \in B$

Dem.: Por dualidade ao Teorema 4.

Teorema 6: $x + x \cdot y = x, \forall x, y \in B$

Dem.:

$$\begin{aligned} x + (x \cdot y) &= (x \cdot 1) + (x \cdot y) \\ &= x \cdot (1 + y) \\ &= x \cdot (y + 1) \\ &= x \cdot 1 \\ &= x \end{aligned}$$

Teorema 7: (Lei de De Morgan) $(x \cdot y)' = x' + y'$

Dem.:

$$\begin{aligned} (a \cdot b)' + a \cdot b &= (a' + b') + (a \cdot b) \\ &= (a' + b' + a) \cdot (a' + b' + b) \\ &= (1 + b') \cdot (1 + a') \\ &= 1 \cdot 1 \\ &= 1. \end{aligned}$$

3. TABELA-VERDADE E EQUIVALÊNCIA LÓGICA

Denomina-se proposição toda afirmação, verdadeira ou falsa, de sentido completo. No caso da Álgebra de Boole, as proposições representam as variáveis da expressão booleana. A tabela-verdade é um mecanismo utilizado para expor todas as possibilidades ou combinações de uma determinada quantidade de proposições, mostrando a saída de acordo com a lógica implementada nessas proposições (Salmon, 1993).

O número de combinações de entrada é igual a 2^n , onde n é a quantidade de entradas. Para a montagem da tabela-verdade utilizam-se todas as combinações possíveis de "1" ou "0" para cada proposição. Em seguida, avalia-se a saída de acordo com a expressão implementada.

Quando as duas últimas colunas de duas tabelas-verdade de expressões booleanas diferentes são idênticas denomina-se equivalência lógica.

Ao se simplificar uma expressão utilizando o método da Álgebra de Boole ou outro método, pode-se saber facilmente se está correto testando a equivalência das tabelas-verdade da expressão original e da simplificada. Se forem equivalentes, a simplificação está correta, porém quanto maior o número de proposições e operações, maior será a dificuldade em comparar tabelas-verdade.

3.1 Operação Lógica AND

A Tabela 2 mostra o que acontece quando uma operação AND é implementada. O resultado mostra que a saída será 0 em todas as ocasiões em que a entrada de uma das proposições forem 0.

Tabela 2 – Resultado da expressão $x = A \cdot B$

A	B	$x = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

A tabela-verdade mostra que x será 1 somente quando as entradas A e B forem 1. Muitas vezes o sinal \cdot é omitido, assim, essa expressão também pode ser escrita $x = AB$, não deixando assim de representar uma operação AND.

Portas lógicas são dispositivos ou circuitos lógicos que operam um ou mais sinais lógicos de entrada para produzir uma e somente uma saída, a qual é dependente da função implementada no circuito. Esta expressão pode ser representada por meio da PORTA LÓGICA AND, cujo símbolo encontra-se na Figura 1, representando uma porta lógica AND de 2 entradas (Tocci & Neal, 2003).

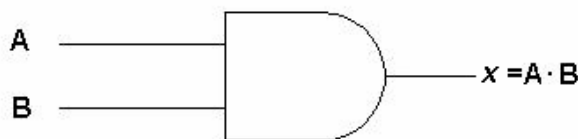


Figura 1 – Representação do símbolo para uma porta AND de duas entradas.

A saída irá ocorrer de acordo com os valores lógicos de entrada das proposições A e B, podendo ser “1” ou “0” e corresponderá com os valores existentes na tabela-verdade, mostrada anteriormente neste artigo.

3.2 Operação Lógica OR

Quando duas ou mais variáveis são implementadas por meio de uma operação lógica OR o resultado será 1 sempre que pelo menos uma das variáveis de entrada tiver nível lógico 1 (Morris & Kime, 2000). A expressão booleana para a operação OR é $x = A + B$.

A Tabela 3 mostra o comportamento da saída x de acordo com a variação das entradas A e B.

Tabela 3 – Resultado da expressão $x = A + B$

A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Sempre que uma entrada ou outra estiverem com o valor lógico 1 a saída será 1. Na operação OR só haverá saída 0 se todas as entradas forem 0.

A Figura 2 mostra o símbolo utilizado para a Porta OR, representando uma porta lógica OR de duas entradas.

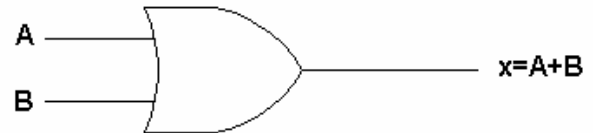


Figura 2 – Representação do símbolo para uma porta OR de duas entradas.

A saída x representa o valor lógico que resulta da expressão $x = A + B$ e este vai variar de acordo com o valor lógico das entradas, obedecendo a tabela-verdade da operação OR já demonstrada anteriormente.

3.3 Operação Lógica NOT

A operação NOT, também denominada INVERSORA é diferente das operações AND e OR porque pode ser realizada utilizando apenas uma variável de entrada. Sendo assim a expressão $x = A'$ pode ser verdadeira ou falsa de acordo com o nível lógico da variável A. Esta expressão é lida como x igual a A negado, ou x igual a não A, ou até mesmo x igual ao complemento de A.

A tabela-verdade da operação NOT é bem simples e está demonstrada na Tabela 4.

Tabela 4 – Resultado da expressão $x = A'$

A	$x = A'$
0	1
1	0

A compreensão é bastante dedutiva também, porque 0 negado 1 e 1 negado é 0.

O circuito NOT é representado por meio da porta lógica demonstrada na Figura 3.

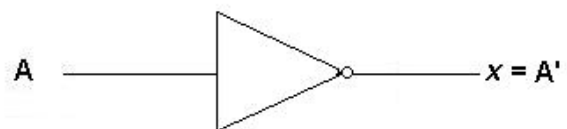


Figura 3 – Representação do símbolo para uma porta NOT.

Quando A for 1, x será 0 e, no caso de A ser 0, x será 1, obedecendo a tabela-verdade.

4. MONTAGEM DO CIRCUITO

Ao pensar em montar um sistema de alarme, o primeiro passo é definir como irá trabalhar. Em que momento vai disparar e quando o alarme é bloqueado. Isto é, ao abrir uma porta, ou uma janela, o alarme dispara e, ao digitar a senha ele é desligado.

A Tabela 5 apresenta a tabela-verdade do protótipo implementado. A tabela foi montada com cinco proposições, denominadas A, B, C, D, E, tendo então $2^5 = 32$ linhas. A proposição A, representa a porta da casa, ou a janela, enquanto as proposições B, C, D e E representam cada bit da senha.

Quando a porta está fechada o valor lógico associado a variável A é 0, e quando está aberta, é 1. Conseqüentemente, o valor lógico associado a variável S é 0 quando o alarme não foi acionado e a porta permanece fechada, e 1 em situação de perigo.

Tabela 5 – Tabela verdade do protótipo

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

A senha foi definida utilizando 4 números binários, em uma combinação de 0 ou 1. A senha escolhida foi “0110”. Isso explica porquê a saída S da linha 23 da tabela-verdade obteve valor 0, apesar da porta estar aberta.

Assim sendo, estando a proposição A com o valor lógico 0, ou seja, porta fechada, não existe a possibilidade do alarme disparar. Isto explica a saída 0 nas 16 primeiras linhas da tabela-verdade. E, quando esta proposição obtém o valor lógico 1 ou seja, porta aberta, o som é disparado. Porém, como já foi falado anteriormente, em um dos casos a senha está correta, impedindo o dispare.

5. EXPRESSÃO LÓGICA DO CIRCUITO

A expressão é encontrada por meio da coluna que representa a saída do circuito levando em consideração as saídas cujo valor lógico seja 1. Multiplica-se as 5 proposições (A, B, C, D, E) da linha cujo valor de S é igual a 1 e, soma-se com a próxima em que a saída também é 1.

Utilizando este método encontra-se a seguinte expressão:

$$S = AB'C'D'E' + AB'C'D'E + AB'C'DE' + AB'C'DE + AB'CD'E' + AB'CD'E + AB'CDE + ABC'D'E' + ABC'D'E + ABC'DE' + ABC'DE + ABCD'E' + ABCD'E + ABCDE' + ABCDE$$

O método utilizado para a simplificação do circuito foi a Álgebra de Boole, foram utilizados os axiomas e teoremas já citados anteriormente. A seguir é apresentado a simplificação do expressão S.

$$S = AB'C'D'(E' + E) + AB'C'D(E' + E) + AB'C(D'(E' + E) + DE) + ABC'D'(E' + E) + ABC'D(E' + E) + ABCD'(E' + E) + ABCD(E' + E)$$

$$S = AB'C'D' + AB'C'D + AB'CD' + AB'CE + ABC'D' + ABC'D + ABCD' + ABCD$$

$$S = AB'C'(D' + D) + AB'CD' + AB'CE + ABC'(D' + D) + ABC(D' + D)$$

$$S = AB'C' + AB'CD' + AB'CE + ABC' + ABC$$

$$S = AB'(C' + CD' + CE) + AB(C' + C)$$

$$S = AB'(C' + D' + E) + AB(C' + C)$$

$$S = AB + AC' + AD' + AE$$

Com esta nova expressão equivalente de S, diminui-se a quantidade de materiais necessários. Em um circuito desta natureza a economia com material é mínima, mas considerando um projeto maior esta se torna relevante, além da economia financeira, também, torna-se mais simples a implementação do mesmo.

6. CIRCUITOS INTEGRADOS

Durante muito tempo, os circuitos construídos a partir da Álgebra de Boole foram implementados utilizando-se dispositivos eletromecânicos como, por exemplo, os relês. Portanto, o nível de tensão correspondente a um nível lógico, poderia assumir qualquer valor dependendo apenas das características do projeto.

A partir do surgimento do transistor, procurou-se padronizar os sinais elétricos correspondentes aos níveis lógicos. Esta padronização ocasionou o surgimento das famílias de componentes digitais com características bastante distintas.

As famílias lógicas diferem basicamente pelo componente principal utilizado por cada uma nos circuitos. As famílias TTL (Transistor-Transistor Logic) e ECL (Emitter Coupled Logic) usam transistores bipolares como principal componente, enquanto, as famílias PMOS, NMOS e CMOS usam os transistores unipolares MOSFET (transistor de efeito de campo construído segundo a técnica MOS - Metal Oxide Semiconductor) como elemento principal de circuito (Tanenbaum,1993).

Atualmente a Família TTL e a CMOS são as mais usadas, sendo empregadas em uma grande quantidade de equipamentos digitais e, também, nos computadores e periféricos. No projeto em questão foram utilizados circuitos da família TTL, por isso somente estes serão abordados.

TTL significa Transistor-Transistor Logic (Lógica Transistor-Transistor). A tensão de alimentação se restringe a 5V contínuos, tendo, porém, uma faixa de tensão correspondente aos níveis lógicos 0 e 1.

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

Esta família é principalmente reconhecida pelo fato de ter duas séries que começam pelos números 54 para os componentes de uso militar e 74 para os componentes de uso comercial.

Os CIs da série TTL 74-padrão oferecem uma combinação de velocidade e potências consumidas adequadas a um grande número de aplicações. Entre os CIs desta série, pode-se encontrar uma ampla variedade de portas lógicas, flip-flops, construídos segundo a tecnologia SSI, além de registradores de deslocamento, contadores,

decodificadores, memórias e circuitos aritméticos, construídos com a tecnologia MSI (Idoeta & Capuano, 1984).

Os CIs utilizados foram o 7404 (NOT), 7408 (AND) e o 7432 (OR).

O CI 7408 bem como o 7432 são compostos por 14 pinos, destes, 2 são responsáveis pela alimentação elétrica do circuito que são o GND (0.0 V, pino 7) e o Vcc (+5.0 V, pino 14).

As Figuras 4 e 5 mostram o funcionamento interno dos CIs 7408 e 7432, cada um deles possui 4 portas lógicas de duas entradas cada uma.

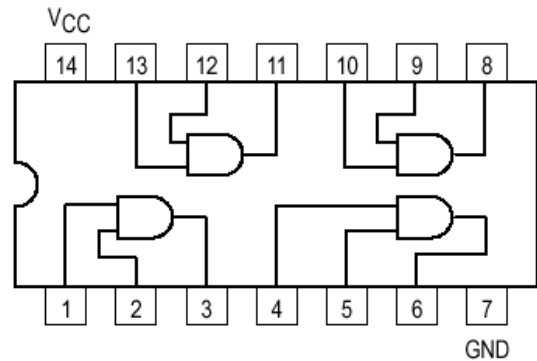


Figura 4 – CI 7408 (AND)

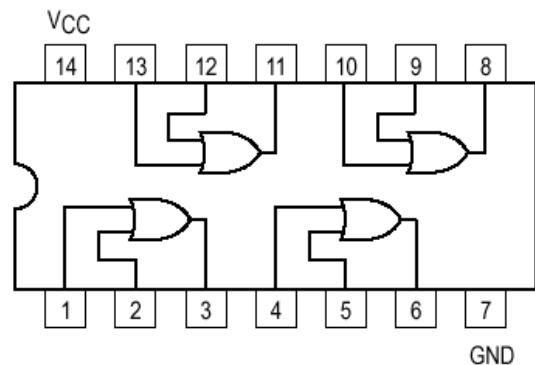


Figura 5 – CI 7432 (OR)

A Figura 6 representa o funcionamento de um CI 7404, este possui 6 inversores.

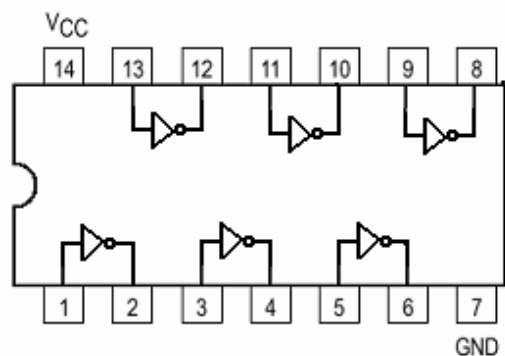


Figura 6 – CI 7404 (NOT)

Após obter todos os conhecimentos necessários para realizar o projeto torna-se extremamente fácil a implementação do mesmo.

Este trabalho foi realizado utilizando circuitos combinacionais que são aqueles em que a saída depende somente dos dados de entrada e da lógica utilizada no processamento por meio de uma ou mais portas lógicas.

Fazem parte dos circuitos combinacionais os circuitos lógicos, codificadores e decodificadores, multiplexadores e demultiplexadores, circuitos aritméticos e outros.

Todo o trabalho foi ligado por meio de uma placa denominada protoboard, esta foi alimentada por uma tensão de 5 volts. O circuito em questão utilizou apenas as portas OR, AND e NOT por meio de um dispositivo denominado circuito integrado. Um CI nada mais é do que um conjunto de portas lógicas.

Para a construção do projeto apresentado foram usados os CI's da família TTL, 7408 (AND), 7404 (NOT) e 7432 (OR).

Para a sonorização do alarme, o sinal de saída foi ligado em um transistor, este por sua vez mandou o sinal para um alto falante que se encarregou de emitir som. As proposições são ligadas ou desligadas por meio de chaves, e a senha inserida aparece em leds de 7 segmentos.

7. PROTÓTIPO

Com a expressão simplificada em mãos, foi feito o desenho do protótipo utilizando portas lógicas (Figura 7).

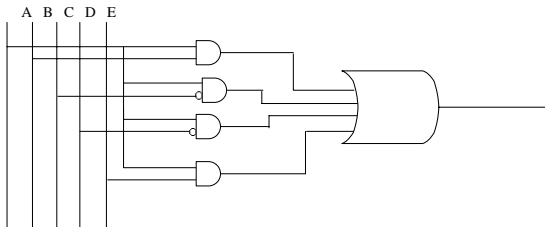


Figura 7 – Desenho do protótipo

Com a tabela verdade, a simplificação realizada e o circuito desenhado pode-se partir para a execução e montagem do protótipo.

Primeiro, foram realizados os testes, em uma protoboard do Laboratório de Hardware do Departamento de Informática da UPIS Faculdades Integradas, para em seguida, com a certeza do circuito funcionando comprar as peças e montar o sistema de alarme definitivo (Figura 8).

Para a montagem do circuito foram utilizados 5 circuitos integrados, 2 Ci's "AND", 2 Ci's "OR", e 1 "NOT".

A montagem elétrica da protoboard foi também uma das partes de execução mais complexa. Foram montadas e testadas todas as chaves e ligações para

então poder começar a montagem das entradas nos Ci's respectivos (Figura 9).

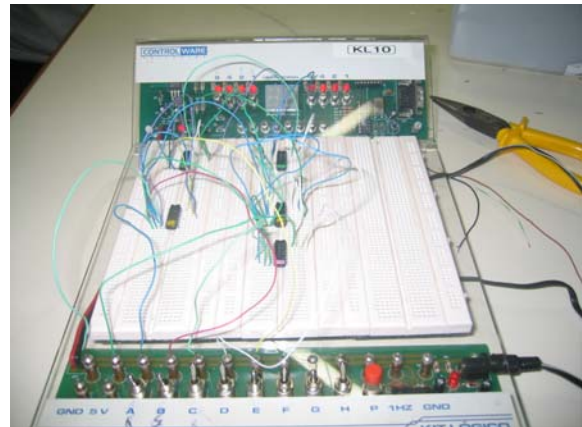


Figura 8 - Protoboard do Laboratório de Hardware da UPIS

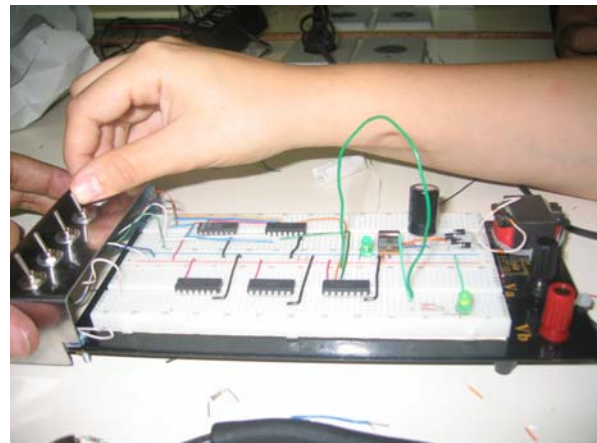


Figura 9 – Montagem do circuito do alarme

A Figura 10 apresenta a configuração final do circuito do alarme desenvolvido.

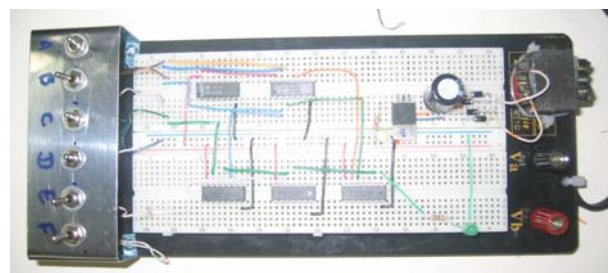


Figura 10 – Circuito do Alarme

8. CONCLUSÃO

Este artigo trata da Álgebra de Boole em uma aplicação prática por meio da construção de um sistema de alarme residencial, comercial e/ou automobilístico.

Para a construção deste projeto são necessários conhecimentos da Álgebra de Boole, os axiomas e teoremas para que se possa fazer a simplificação.

Oliveira, D.A. & Cavalcante, A.L.B. 2006 Sistema de Alarme Residencial Comercial e/ou Automobilístico. Revista Virtual (<http://www.upis.br/revistavirtual>). UPIS Faculdades Integradas, Brasília, DF, 7 p.

Também conhecimentos sobre portas lógicas, tabelas-verdade, além da necessidade de saber trabalhar em uma protoboard e circuitos integrados.

O artigo tratou destes conhecimentos mostrando a aplicação e também a importância destes para a montagem de circuitos, podendo assim aplicá-los não somente em um sistema de alarme, mas também em outros trabalhos.

O projeto apresentado ainda mostra falhas, como por exemplo, ao abrir e fechar a porta, o alarme também para de funcionar. Para que isto não aconteça deve-se utilizar, no projeto, o auxílio de um software. O alarme construído precisa de algumas melhorias, pois, ainda está no primeiro estágio.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- Cavalcante, A.L.B. Lógica Matemática, Digital e de Fuzzy. Notas de Aula. UPIS – União Pioneira de Integração Social. FATEC – Faculdade de Tecnologia. Sistemas de Informação. Brasília, DF, p. 89.
- Daghlian, J. Lógica e Álgebra de Boole. São Paulo: Atlas, 1986.
- Idoeta, I.V. & Capuano, F.G. Elementos de Eletrônica Digital. São Paulo: Erica, 1984.
- Morris, M.M & Kime, C.R. Logic and Computer Design Fundamentals. Englewood Cliffs:Prentice Hall, 2000.
- Salmon C.W. Lógica. São Paulo: Prentice-Hall do Brasil, 1993.
- Tanenbaum, S.A. Organização Estruturada de Computadores. São Paulo: Prentice-Hall do Brasil, 1993.
- Tocci, R.J. & Neal, S.W. Sistemas Digitais: Princípios e Aplicações. 8. ed. São Paulo: Prentice Hall, 2003.