**Uni Evangélica- Centro Universitário de Anápolis**

**Prof.: Alexandre Tannus**

**Disciplina: Circuitos Digitais**

**RELATÓRIO**

**Etapa 2 -Display de LED - TinkerCAD**

**Acadêmicos: Charlley Junior Jabbar**

**Mateus Correia Bezzan**

**Sumário**

[1. INTRODUÇÃO 3](#_Toc8728583)

[2. FUNDAMENTAÇÃO TEÓRICA 4](#_Toc8728584)

[3. Desenvolvimento do Projeto 10](#_Toc8728585)

[CONSIDERAÇÕES FINAIS 15](#_Toc8728586)

[REFERENCIAS 16](#_Toc8728587)

# INTRODUÇÃO

Os circuitos digitais através dos diversos avanços tecnológicos atingiram o seu auge durante a **“era eletrônica”**, onde todos as soluções eram atingidas por meio de sistemas analógicos, também conhecidos como sistemas lineares. (IDOETA & CAPUANO).

Tudo começou através dos estudos da obra intitulada ***An Investigation of the Laws of Thought***, criada pelo matemático inglês **George Boole** ( 1815-1864 ) , obra está que apresentava um sistema matemático de análise lógica conhecido como **Álgebra de Boole.** (IDOETA & CAPUANO)

Apenas em 1938, o engenheiro americano **Claude Elwood Shannon** utilizou as teorias da **álgebra de boole** para a solução de problemas de circuitos de telefonia com relés, tendo publicado um trabalho denominado ***Symbolic Analysis of Relay and Switching*** (IDOETA & CAPUANO)*,* pragmaticamente incluindo na área da tecnologia o campo da eletrônica digital para o mundo.

Os estudos de Boole foram passados por diversos outros estudiosos que no final os postulados de Boole deram origem às principais **funções lógicas** sendo as variáveis e as expressões envolvidas uma derivação da **álgebra de Boole** denominada como **Booleanas,** afirma Idoeta como as funções lógicas e, ou, não , ne e nou acabaram vinda para a realidade da sociedade. Funções lógicas estas que saõ encontradas apenas em 2 estados distintos:

* **O estado 0 (zero) e**
* **O estado 1 (um)**

O estado **0** representará o **não, falso**, chave desligada, interruptor desligado, ausência de tensão, aparelho desligado, em outras palavras significa a ausência ou a negação de algo.

O estado 1 representa **sim, verdadeiro,** chave ligada, presença de tensão, chave ligada, de forma sucinta o 1 é o contrário do 0 , onde tudo está presente e se resume à um mero **sim ou verdadeiro.**

O fator crucial para a identificação dos objetos no mundo real é a imagem, ou seja o que reflete a luz e é reproduzida no nossos cérebros afim da criação de um julgamento arbitrário do resultado.

# FUNDAMENTAÇÃO TEÓRICA

1. **Álgebra de Boole**

As variáveis de Boole podem ser representadas por letras, estás que estão sujeitas à assumir valores entre 0 e 1.

De acordo com Idoeta e Capuano (pág. 89) , a expressão de Boole consiste na representação de uma sentença matemática composta por termos que assumem variáveis booleanas e os resultados podem ser entre 0 e 1.

1. **Os Postulados de Boole**

Boole escreveu 3 tipos de postulados estes que podem interagir entre si ou não, dessa forma Boole foi capaz de distinguir quais os tipos de operações que eram necessárias a serem realizadas. Os postulados da complementação, da adição e da multiplicação da álgebra de Boole e cada uma com a sua identidade resultante definida.

1. **Postulado da Complementação e suas identidades**

Este postulado tem como objetivo definir as regras da complementação na álgebra de Boole. Chamaremos de **A’** o complemento de A:

**Figura 1.0 – Bloco lógico que executa a complementação é o Inversor**

**1̊ Se A = 0 🡪 A’ = 1**

**2̊ Se A = 1 🡪 A’ = 0**

Dessa forma como consequência podemos concluir que:

**A = A**

1. **Postulado da Adição e suas identidades**

Este postulado representa as regras da adição dentro da Álgebra de Boole:

Representado pela figura 1.1 o bloco lógico da adição é o ***OR*** ou **OU** :

**Figura 1.1 Bloco lógico do Postulado da Adição**

**1̊ 0 + 0 = 0**

**2̊ 0 + 1 = 1**

**3̊ 1 + 0 = 1**

**4̊ 1 + 1 = 1**

Em síntese dos postulados temos então que as seguintes identidades representam a adição :

**A + 0 = A**

**A + 1 = 1**

**A + A = A**

**A + A = 1**

Caso houver a presença de ao menos um A , o resultado sempre será A.

Notamos que a Soma de 1 + qualquer variável sempre será 1;

Notamos que a soma de uma variável por ela mesma sempre será ela mesma;

E por fim notamos que ao somar a sua variável por seu complemento teremos o resultado 1.

1. **Postulado da Multiplicação e suas identidades.**

Este postulado representa as regras da multiplicação dentro da Álgebra de Boole:

Representado pela figura 1.2 o bloco lógico da multiplicação é o ***AND*** ou **E** :

**Figura 1.2 Bloco Lógico do Postulado da multiplicação**

**1̊ 0 . 0 = 0**

**2̊ 0 . 1 = 0**

**3̊ 1 . 0 = 0**

**4̊ 1 . 1 = 1**

Dessa forma podemos concluir as seguintes identidades que representam o postulado da multiplicação:

**A . 0 = 0**

**A . 1 = A**

**A . A = A**

**A . A = 0**

Observa-se que todo número multiplicado por 0 será 0;

Notamos que o resultado das expressões sempre será a própria variável A ;

Notamos que o resultado das expressões serão sempre iguais a A;

Notamos para que ambos os valores possíveis que a variável possa assumir, o valor resultante será sempre 0.

1. **Propriedades**

O objetivo das propriedades é apresentação das ferramentas para o manuseio das expressões matemáticas obtidas ao realizar o uso dos postulados, dessa forma apresentando os possíveis manuseios para as simplificações das expressões algébricas.(IDOETA & CAPUANO).

**Propriedades Comutativas**

Propriedade válida tanto para a adição quanto para a multiplicação .

**Adição: A + B = B + A**

**Multiplicação : A . B = B . A**

**Propriedades Associativas**

Semelhante ao anterior a mesma é válida tanto para os postulados de adição e multiplicação temos a propriedade associativas que visam a interligação das variáveis.

**Adição: A + (B + C) = (A + B) + C** logo teremos que essa expressão de ambos os lados é igual à **A + B + C.**

**Multiplicação: A . (B . C) = (A . B) + C** logo teremos que essa expressão de ambos os lados é igual à **A . B . C**

**Propriedades Distributivas**

A propriedade distributiva é a técnica onde realiza a multiplicação de um conjunto de associações ou comutativas, sendo também conhecida informalmente como a a técnica do chuveirinho , na qual um valor representando o macro da expressão multiplica uma outra expressão interna a própria expressão diretamente cada valores, ou seja :

**A ( B+C) = A.B + A.C**

Dessa forma auxiliando tanto na simplificação quanto na distribuição dos valores.

1. **Teorema de Morgan**

O teorema de Morgan representa um fator que ocorre com as correlações entre os postulados onde o complemento do produto é igual à soma dos complementos**:**

**(A . B ) = A + B**

Consequentemente temos então que da mesma forma que o produto dos complementos reagem a soma dos complementos resultará no produto de cada complemento:

**(A + B ) =A . B**

1. **Simplificação de circuitos lógicos variáveis e Expressões na Álgebra de Boole**

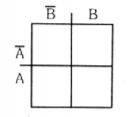
Com os conceitos dos postulados de ***Boole*** e as suas expressões booleanas, somos capazes de realizar simplificações das expressões consequentemente os circuitos digitais são reduzidos uma vez que cada redução reduz a quantidade de componentes necessários para realizar a mesma função.

De acordo com Idoeta e Capuano , ambos afirmam que há 2 meios para a realização da simplificação uma onde a pessoa realiza o uso da Álgebra de ***Boole***, o segundo meio é por meio de uma técnica “ gráfica” que são os mapas de ***Veitch- Karnaugh,*** como será representado no tópico de desenvolvimento do projeto de um contador hexadecimal .

1. **Mapas de Veitch – Karnaugh – Simplificando expressões lógicas.**

O mapa de Karnaugh é representado de acordo com a necessidade das entradas, o mesmo é viável até atingir o valor de 4 variáveis de entradas a partir disso o mesmo passa a ser muito complexo e de difícil compreensão.

Observa-se a figura 1.3 , representando 2 variáveis:

****

**Figura 1.3 Mapa de Karnaugh de duas variáveis.**

na figura 1.4 representamos todas as regiões que são possíveis ser observadas no mapa, as mesmas varrem as diversas possibilidades de combinações entre as duas variáveis..

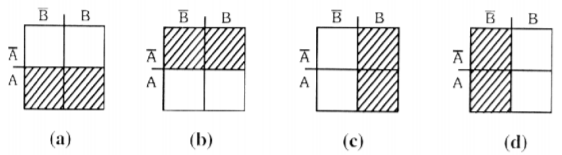


Figura 1.4 Regiões do Mapa de Karnaugh

(a) Região onde A = 1;

(b) Região onde A = 0 ( A = 1);

(c) Região onde B = 1;

(d) Região onde B = 0 (B = 1).

Portanto com 2 variáveis somos capazes de obter 4 combinações, estas que são as influenciadores das futuras expressões extraídas:

|  |  |
| --- | --- |
| A B | |
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

**Tabela 1.0 Regiões cobertas pelo mapa de duas entradas**

Porém a versão que foi usada para o desenvolvimento foi uma adaptação deste mapa diretamente com o código binário onde facilita a análise dos mapas resultantes e preenchimento por meio de uma tabela verdade conforme a figura de um mapa criado por meio de 4 entradas :

CD

A

AB

00

10

11

01

00

01

11

10

**Figura 1.5 Mapa Adaptado de Karnaugh**

O mapa adaptado segue o funcionamento de acordo com as saídas resultantes da tabela verdade , no caso da figura 1.5 os dados que iriam preencher as linhas e colunas do mesmo são as saídas de “A” .

# Desenvolvimento do Projeto

O projeto consiste em recriar um contador hexadecimal onde realizará a contagem de 0 à F qual as mesmas deveram estar implementadas no **Arduino** (hardware) que aceita a linguagem de programação C para micro controladores. Circuito este implementado que se chama sketch que apresentará obrigatoriamente duas funções : **setup() e loop() .**

O projeto está dividido em 3 etapas onde serão apresentadas os desafios a serem trilhados na implementação do microcontrolador fazendo uso tanto do conhecimento em *hardware* quanto de *firmware*.

O projeto foi desenvolvido após a realização das tabelas verdades e os mapas de *Veitch-Karnaugh* onde foi desenvolvidas as saídas das entrada fornecidas por um interruptor binário de 4 *switchs* ,

Tabela Verdade produzida pelo projeto :

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| a | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| b | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| c | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| d | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| e | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| f | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

**Tabela 1.1 Tabela Verdade do Projeto de Display de Led de números**

Após o desenvolvimento da tabela verdade conforme, o data sheet do display de led a seguir , teremos então a representação das saídas da tabela verdade no mapa de Karnaugh:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (saída) segA | 00 | 01 | 11 | 10 |
| 00 | **1** | **0** | **1** | **1** |
| 01 | **0** | **1** | **1** | **1** |
| 11 | **1** | **0** | **1** | **1** |
| 10 | **1** | **1** | **0** | **1** |

**Tabela 1.2 Mapa de Karnaugh da saída “a”**

Após a análise do mapa de Karnaugh temos então uma expressão matemática que será justamente a saída lógica do código.

Expressão Matemática :

**segA = AC + BC + CD + ABD + AD + ABC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (saída) segB | 00 | 01 | 11 | 10 |
| 00 | **1** | **1** | **1** | **1** |
| 01 | **1** | **0** | **1** | **0** |
| 11 | **0** | **1** | **0** | **0** |
| 10 | **1** | **1** | **0** | **1** |

**Tabela 1.3 Mapa de Karnaugh da saída “b”**

segB = ACD + BC + ACD + AB + ACD + BCD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (saída) segC | 00 | 01 | 11 | 10 |
| 00 | **1** | **1** | **1** | **0** |
| 01 | **1** | **1** | **1** | **1** |
| 11 | **1** | **1** | **1** | **1** |
| 10 | **0** | **1** | **0** | **0** |

**Tabela 1.4 Mapa de Karnaugh da saída “c”**

**segC = AC + AD + B + BCD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (saída) segD | 00 | 01 | 11 | 10 |
| 00 | **1** | **0** | **1** | **1** |
| 01 | **1** | **1** | **0** | **1** |
| 11 | **1** | **1** | **0** | **1** |
| 10 | **1** | **1** | **1** | **0** |

**Tabela 1.5 Mapa de Karnaugh da saída “d”**

SegD = AD + BCD + BC + CD + BD + AC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (saída) segE | 00 | 01 | 11 | 10 |
| 00 | **1** | **0** | **0** | **1** |
| 01 | **0** | **0** | **0** | **1** |
| 11 | **1** | **1** | **1** | **1** |
| 10 | **1** | **0** | **1** | **1** |

**Tabela 1.6 Mapa de Karnaugh da saída “e”**

segE = CD + ABD + AB + AC + AD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (saída) segF | 00 | 01 | 11 | 10 |
| 00 | **1** | **0** | **0** | **0** |
| 01 | **1** | **1** | **0** | **1** |
| 11 | **1** | **0** | **1** | **1** |
| 10 | **1** | **1** | **1** | **1** |

**Tabela 1.7 Mapa de Karnaugh da saída “f”**

segF = CD + AB + AC + ABC + BD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (saída) segG | 00 | 01 | 11 | 10 |
| 00 | **0** | **0** | **1** | **1** |
| 01 | **1** | **1** | **0** | **1** |
| 11 | **0** | **1** | **1** | **1** |
| 10 | **1** | **1** | **1** | **1** |

**Tabela 1.8 Mapa de Karnaugh da saída “g”**

segG = CD + AB + ABC + AD + BC .

Os materiais utilizados para o desenvolvimento do sketch – etapa 1 foram de acordo com a **Tabela 1.6** :

|  |  |
| --- | --- |
| **Material** | Imagem |
| **Arduino** |  |
| **Interruptor** |  |
| **Resistores** |  |
| **Placa de ensaio pequena** |  |
| **Display de Números** |  |

**Tabela 1.9 Materiais para construção do projeto.**

1. **Funções / procedimentos utilizados no Código fonte para a criação das portas lógicas.**

**Void setup() //** momento no qual são definidas as portas de entrada e saída de dados.

**Void loop() //** serve para realizar as repetições das variáveis contadoras e as variáveis de entrada para manter o sistema em funcionamento.

**Função acende“ N ” (bool A , bool B , bool C , bool D)**

void acendeA(bool A, bool B, bool C, bool D) // sendo responsáveis em realizar os procedimentos lógicos resultantes dos mapas de Karnaugh.

Trecho do código da condição de acender os leds do display referente à saída “A” do mesmo.

1. **Funções específicas do Arduino:**

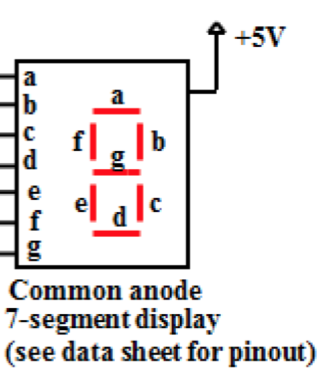
**pinMode(segA,OUTPUT); //** Ativador na porta 0 do controlador local onde os dados de envio deveram realizar a saída e serem enviados vice e versa.

**digitalWrite(segA, isAceso); //** Enviar dado de saída para a porta 0 do microcontrolador acendendo a LED como resultado de envio.

**digitalRead(botA);** // 1 Realzia a leitura da entrada no interruptor de valor 1 , aderindo valores de 0 ou 1 para o mesmo.

1. **Display Sheet Arduino –**

Este foi o *template* utilizado para configurar o *display* de *led* , segue de acordo com a figura 1.6 , a representação de quais saídas se referem à cada led :



**Figura 1.6 *Display Sheet* Arduíno.**

# CONSIDERAÇÕES FINAIS

O recurso tem por objetivo, através do arduino esclarecer os contextos desejados. Sua presença tem desenvolvido na melhoria e automatização de processos , fazendo com que a tecnologia seja usada ao favor da humanidade. O Arduino em junção com as funções lógicas no mundo real é a representação da ascensão do progresso onde o mesmo é capaz de atingir a verossimilhança dos processos que realizamos manualmente em nossa sociedade. Em suma, o Arduino e as portas lógicas nos microcontroladores é apropriado para realizações de trabalhos simples que são realizados manualmente porém de forma automatizada e com maior confiabilidade.

# REFERENCIAS

Idoeta, I.V. & Capuano, F.G.; **Elementos de Eletrônica Digital**, 12ª. edição, Érica, 1987.

Mendelson, E. ; **Álgebra booleana e circuitos de chaveamento**, McGraw-Hill, 1977.

Karnaugh, M. **"The Map Method for Synthesis of Combinational Logic Circuits". Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics.** 1953.