

Uni Evangélica- Centro Universitário de Anápolis

Prof.: Alexandre Tannus
Disciplina: Circuitos Digitais

RELATÓRIO

Etapa 1 -Portas Lógicas - TinkerCAD

Acadêmicos: Charlley Junior Jabbar
Mateus Correia Bezzan

Sumário

1. INTRODUÇÃO	3
2. FUNDAMENTAÇÃO TEÓRICA	4
3. Desenvolvimento do Projeto	8
CONSIDERAÇÕES FINAIS	10
REFERENCIAS.....	10

1. INTRODUÇÃO

Os circuitos digitais através dos diversos avanços tecnológicos atingiram o seu auge durante a “**era eletrônica**”, onde todas 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 esta 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 vindo para a realidade da sociedade. Funções lógicas estas que são 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**.

2. FUNDAMENTAÇÃO TEÓRICA

a) Função E ou AND.

A função E, é aquela que executa a **multiplicação** de 2 ou mais variáveis booleanas. A sua representação algébrica é realizada como: $S = A \cdot B$ ou $S = A \text{ e } B$. Também é conhecida pelo termo em inglês **AND**.

A sua tabela verdade de uma função **E ou AND**:

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

1.0 Tabela Verdade E ou And

Como podemos observar a **tabela 1.1**, apenas 2 resultados são encontrados 0 indicando o não ou falso e 1 indicando sim ou verdadeiro.

Representação da porta lógica **E ou AND** em um circuito digital:

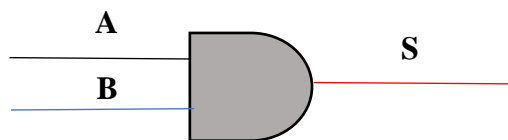


Figura 1.0. Porta lógica E ou AND

O número de situações possíveis irá depender da quantidade de entrada de dados, seguindo a função de crescimento 2^n , neste contexto então caso N seja 3 então haverá 8 linhas na tabela lógica de situações de 0 ou 1.

b) Função OU ou Or.

A função **OU** é àquela que assume valor 1 quando uma ou mais variáveis da entrada forem iguais a 1 e assume valor 0 se, e somente se, todas as variáveis de entrada forem iguais a 0. Sua representação algébrica para 2 variáveis de entrada é $S=A+B$, onde se lê $S = A$ **ou** B . (IDOETA & CAPUANO). O termo **OR**, também utilizado, é derivado do inglês.

A sua **tabela verdade** de uma função **Ou** ou **OR**:

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

1.1 Tabela Verdade OU ou OR

Como podemos observar a **Tabela 1.2** apresenta um valor verdadeiro em todo momento no qual ocorrer algum tipo de entrada verdadeira, ou seja sempre que ocorrer alguma entrada no valor de 1 será verdadeiro o resultado independente da quantidade advinda de 0 , continuará sendo verdadeira.

Representação da porta lógica **OU** ou **OR** em um circuito digital:

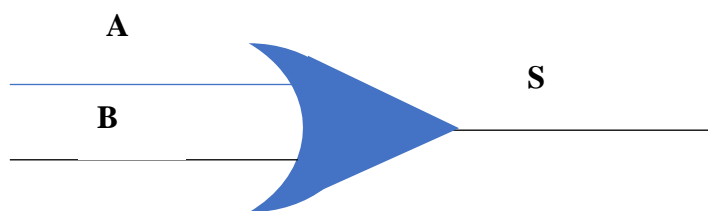


Figura 1.2 Porta lógica OU ou OR

c) **Função OU EXCLUSIVO** ou **EXOR**.

A função que ele executa, como o próprio nome diz, consiste em fornecer **1 à saída quando as variáveis de entrada forem diferentes entre si**. Com esta pequena apresentação podemos montar sua tabela da verdade e, obter pelo mesmo processo visto até aqui, sua expressão característica e, posteriormente, esquematizar o circuito (IDOETA & CAPUANO). O **OU EXCLUSIVO** é representado da seguinte forma algebricamente :

$$A \oplus B = S$$

A sua **tabela verdade** de uma função **Ou EXCLUSIVO** ou **EXOR**:

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

1.2 Tabela Verdade OU EXCLUSIVO ou EXOR

Da mesma tabela é possível observar a sua expressão característica

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

A sua representação em portas lógicas é bem semelhante à do **OU** , porem possui um detalhe que diferencia , mesmo conceito que é levado em sua tabela lógica que um detalhe diferencia o **OU EXCLUSIVO** do **Ou simples**.

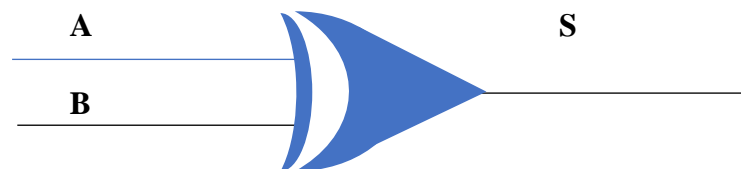


Figura 1.3 Porta Lógica OU EXCLUSIVO ou EXOR

d) Função Negação (~ , N , -)

A função Negação é uma das consequências dos postulados de Boole afirma Mendelson, a mesma é responsável em inverter os resultados , podendo ser confundida com a função inversora.

De forma simples a função negação serve para negar o valor de entrada , consequentemente invertendo a saída . Observe a **Figura 1.4** , onde a tabela verdade do E (*AND*) possui a sua saída negada assim então resultando na função NE (*NAND*)

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

$$A \cdot B = S$$

$$A \cdot B = \bar{S}$$

Figura 1.4 Porta Lógica E em comparação à NE

Este caso em especial acontece com as demais funções **lógicas** **OU** , **OU EXCLUSIVO** dessa forma criando um total de 7 funções lógicas incluindo a Negação como parte das mesmas.

A seguir as outras representações das tabelas verdades “negadas” , **NOU** , **N OU EXCLUSIVO (NXOR)**:

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

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

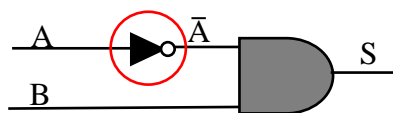
1.4 Tabela Verdade NOU ou NOR

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

$$\overline{A \oplus B} = S$$

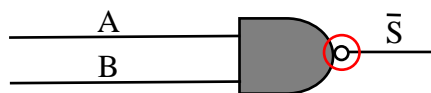
1.5 Tabela Verdade N OU EXCLUSIVO ou NXOR

A função negação possui representações em circuitos digitais , representações estas que a sua posição define se a mesma é utilizada durante a entrada ou saída de um dado ou informação de acordo com as **Figura 1.5** e **Figura1.6** :



$$\bar{A}.B = S$$

Figura 1.5
Negação durante a entrada



$$A.B = \bar{S}$$

Figura 1.6
Negação durante a saída

3. Desenvolvimento do Projeto

O projeto consiste em recriar as portas lógicas **AND** , **OR** , **XOR** , **NAND** , **NOR** , e **XNOR** , ao 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*.

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




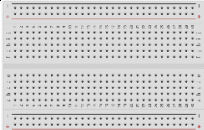
Material	Imagem
Arduino	
LEDs	
Resistores	
Placa de ensaio <u>pequena</u>	

Tabela 1.6 Materiais para construção do projeto.

a) 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 AND (int A, int B) // Função lógica AND

Void OR (int A, int B) // Função lógica OR

Void NOT (int A, int I) // Função lógica NOT

Void NAND (int A , int B , int I) // Função lógica NAND

Void NOR (int A, int B , int I) // Função lógica NOR

Void XOR (int A, int B , int I) // Função lógica XOR

Void XNOR (int A, int B , int I) // Função lógica XNOR

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

b) Funções específicas do Arduino:

pinMode(13,OUTPUT); // Ativador na porta 13 do controlador local onde os dados de envio deveram realizar a saída e serem enviados vice e versa.

digitalWrite(11, HIGH); // Enviar dado de saída para a porta 11 do microcontrolador acendendo a LED como resultado de envio.

digitalWrite(11,LOW); // Ao contrário do HIGH o LOW é responsável em apagar ou manter-se apagado o resultado lógico ao qual foi operado o microcontrolador.

delay(1000); // *Wait for 1000 millisecond(s)* (Prolongar Tempo de espera para a próxima repetição, contado em milissegundos.)

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**, 12^a. edição, Érica, 1987.
E. Mendelson; **Álgebra booleana e circuitos de chaveamento**, McGraw-Hill, 1977.