

Relatório Trabalho Prático de Circuitos Digitais - Etapa 1 - Portas Lógicas

Lucas Nunes Santana, João Vitor Moreira de Sousa.

Engenharia da Computação - UniEvangélica - Centro Universitário - Anápolis, GO -
Brasil

nlucas00043@gmail.com, jvmoreirasousa@hotmail.com

Resumo. Este relatório apresenta uma abordagem a respeito da teoria e lógica utilizada para o desenvolvimento do trabalho prático de Circuitos Digitais sobre a aplicação das portas lógicas.

Palavras-Chave: Portas Lógicas, Arduino, Circuitos Digitais.

1. Fundamentação Teórica

As funções lógicas podem ser encontradas em apenas 2 estados distintos:

Estado 0 (zero) e Estado 1 (um).

O estado 0 representa o não ou o falso, em outras palavras significa a ausência ou a negação de algo.

O estado 1 representa o sim ou o verdadeiro, de forma sucinta o 1 é o contrário do 0.

O Trabalho consiste em aplicar as seis portas lógicas (**AND**, **OR**, **NAND**, **NOR**, **XOR**, **XNOR**) na prática utilizando dois LED's como entrada e um como saída para demonstrar cada uma dessas portas. Antes de especificar como foram feitas essas portas no código é necessário conceituá-las para um melhor entendimento.

A porta **AND** utiliza como operador o produto lógico e sua saída é igual a 1 se todas as entradas for 1. A saída é igual a zero se pelo menos uma entrada for 0.

A porta **OR** tem como operador lógico a soma. A saída é igual a 1 se pelo menos uma das entradas for 1. A saída é igual a zero se nenhuma entrada for 1.

A porta **NAND** utiliza os operadores lógicos de produto e o de inversão. A saída é igual a 0 se todas as entradas for 1. A saída é igual a 1 se ao menos uma entrada for 0.

A porta **NOR** utiliza os operadores de soma e o de inversão. A saída é igual a 0 se pelo menos uma das entradas for 1. A saída é igual a 1 se nenhuma entrada for 1.

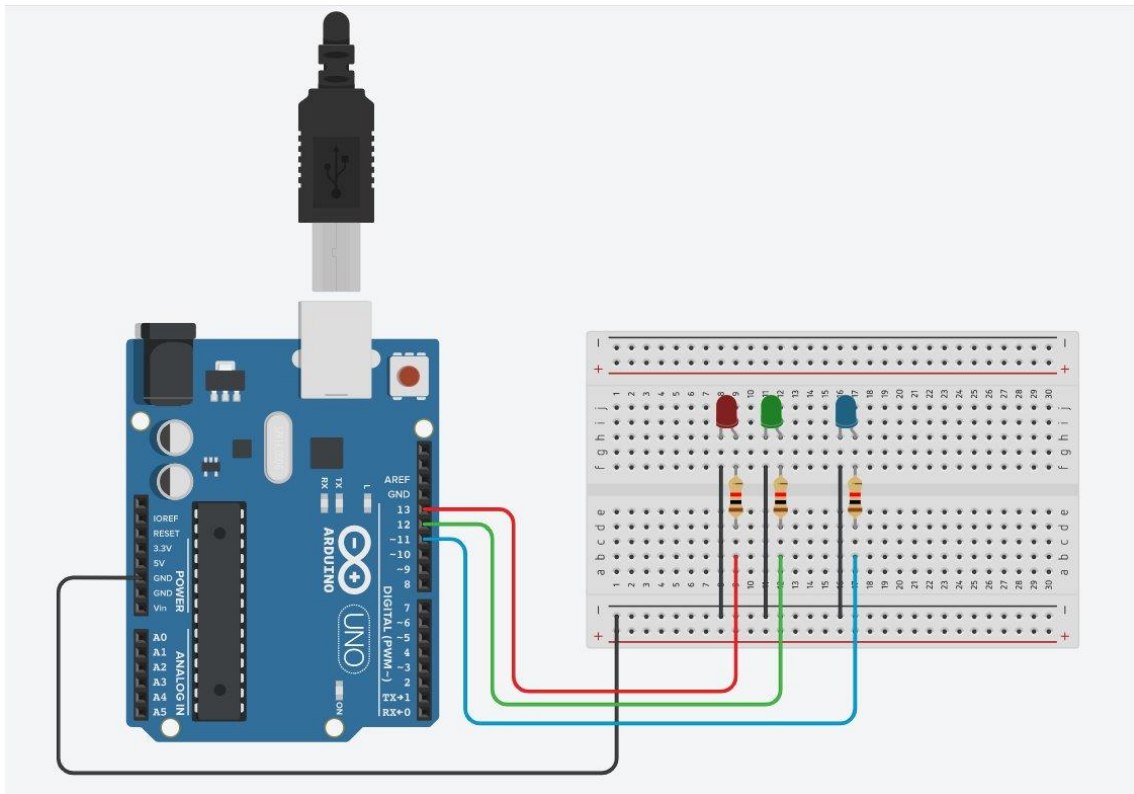
A porta **XOR** utiliza-se do operador de soma lógica, com um círculo. A saída é igual a 0 se as entradas forem iguais. A saída é igual a 1 se as entradas não forem iguais

A porta **XNOR** utiliza-se do operador de soma lógica, com um círculo e o de inversão. Tem as saídas inversas da operação XOR. A saída é igual a 1 se as entradas forem iguais. A saída é igual a 0 se se as entradas não forem iguais

2. Aplicação Prática

Como foi explicado previamente o trabalho consiste em demonstrar essas portas lógicas na prática utilizando Arduino. Para isso foi utilizada a linguagem de programação C e o simulador de projetos/circuitos denominado TinkerCad.

Primeiramente foi criado o circuito abaixo onde os LED's vermelho e verde são as entradas e o LED azul é a saída. Para que esses LED's funcionassem foram necessários os seguintes materiais: Arduino Uno R3, resistores e uma placa de ensaio pequena.



Partindo agora para a parte lógica foram utilizadas as seguintes funções ou procedimentos:

- void setup()** // Onde se define as portas de entrada e saída
- bool func_and(bool a, bool b)** // Função da porta lógica AND
- bool func_or(bool a, bool b)** // Função da porta lógica OR
- bool func_nand(bool a, bool b)** // Função da porta lógica NAND
- bool func_nor(bool a, bool b)** // Função da porta lógica NOR
- bool func_xor(bool a, bool b)** // Função da porta lógica XOR
- bool func_xnor(bool a, bool b)** // Função da porta lógica XNOR
- void loop()** // Permite repetições das variáveis

Algumas funções específicas do Arduino também foram utilizadas:

- pinMode(13,OUTPUT);** // Ativador na porta 13 do controlador, 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(2000); // Wait for 2000 millisecond(s) tempo de espera para a próxima execução.

Para a parte lógica foram utilizados os seguintes raciocínios:

-Foi criado uma lógica para permitir que os LED's recebessem o valor booleano de verdadeiro ou falso, e assim simulassem a tabela verdade para cada uma das portas lógicas.

-Foi criada uma função bool para cada porta lógica a fim de simular a sua tabela verdade correspondente.

-Foi criada uma lógica para que o circuito executasse cada uma das operações de **a** e **b** a cada 2000 ms, permitindo rodar enquanto executava as 4 opções (false e false; true e false; false e true; true e true) para cada porta.

-Também foi criado uma lógica para executar as 6 portas lógicas automaticamente, o circuito altera a execução da porta ao terminar de executar a sua tabela verdade.

Como resultado do projeto, o circuito criado demonstra a implementação de funções que mimetizam as portas lógicas fundamentais (AND, OR, NAND, NOR, XOR e XNOR) e permite a visualização através dos três LED's de saída.