

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO
RELATÓRIO TÉCNICO – TEORIA DA COMPUTAÇÃO

MARIA EDUARDA PEDROSO

**TRABALHO FINAL: IMPLEMENTAÇÃO DE MÁQUINA DE TURING
EM UM HARDWARE E COMPARAÇÃO COM CODIGO EM C**

APUCARANA, 2022

MARIA EDUARDA PEDROSO

**TRABALHO FINAL: IMPLEMENTAÇÃO DE MÁQUINA DE TURING
EM UM HARDWARE E COMPARAÇÃO COM CODIGO EM C**

Relatório Técnico do Trabalho Disciplinar
apresentado como requisito parcial para
obtenção de créditos na disciplina de Teoria
da computação da Universidade Tecnológica
Federal do Paraná.

Orientador: Prof. Dr. Lucio Agostinho Rocha

APUCARANA, 2022

RESUMO

O presente trabalho tem como objetivo apresentar a implementação de uma máquina de Turing na qual dado uma linguagem com o alfabeto de 0 e 1 ela reconheça se o número é divisível por três ou não, a máquina foi implementada em um hardware, nesse caso o Arduino, apresentar os resultados e comparar com a implementação de um código feito em linguagem.

Palavras-chave: Máquina de Turing. Turing. Teoria da computação. Autômato.

SUMÁRIO

1 INTRODUÇÃO.....	5
1.1 MAQUINA DE TURING.....	5
1.2 ARDUINO E TINKERCAD.....	7
2 DESENVOLVIMENTO.....	9
3 RESULTADOS.....	15
4 CONSIDERAÇÕES FINAIS.....	17
5 LINKS EXTERNOS.....	17
REFERENCIAS.....	18

1 INTRODUÇÃO

Exploraremos neste trabalho o conceito de Máquinas de Turing. Um dos objetivos a serem cumpridos por nós será o de fornecer elementos teóricos básicos para a compreensão do que seja e de como funciona uma Máquina de Turing. Para isso, faremos uma descrição de sua natureza e de seu funcionamento buscando caracterizar quais os aspectos lógico-conceituais dos modelos das Máquinas de Turing.

1.1 MÁQUINA DE TURING

A máquina de Turing é um dispositivo teórico conhecido como máquina abstrata universal, concebida pelo matemático britânico Alan Turing muitos anos antes de existirem os modernos computadores digitais, tais como os conhecemos hoje. Por isso, é um modelo abstrato (teórico) de um computador, que se restringe apenas aos aspectos lógico-conceituais de seu funcionamento (memória, estados e transições, etc.) e não à sua implementação física. O modelo de máquina de Turing foi primeiramente idealizada por Alan Turing na década de 30, na tentativa de resolver um problema proposto por Hilbert, o bem conhecido problema da decisão. Turing acreditava que poderia dar uma solução decisiva e definitiva a esse problema. Propôs então que uma definição de “máquina de computação” (do inglês computing machine) o faria. Essa concepção de máquina atingiria o âmago da noção de procedimento efetivo – conceito muito importante hoje na I.A. e na ciência computacional. Mas Turing não tinha em mente apenas isso. Como muitos comentadores nos mostram hoje, a ideia central que Turing queria demonstrar e argumentar a favor era a de que uma Máquina de Turing poderia simular o mesmo desempenho da mente humana, em qualquer tipo de sua atividade e não apenas no sentido de calcular. Turing acreditava que a mente funcionava seguindo instruções simples, regida por um conjunto de regras mecânicas, por procedimentos puramente mecânicos. Temos que explicitar que esta máquina consiste de:

1. Uma fita que é dividida em células, uma adjacente à outra. Cada célula contém um símbolo de algum alfabeto finito. O alfabeto contém um símbolo especial branco (aqui escrito como \rightarrow) e um ou mais símbolos adicionais. Assume-se que a fita é arbitrariamente extensível para a esquerda e para a direita, isto é, a máquina

de Turing possui tanta fita quanto é necessário para a computação. Assume-se também que células que ainda não foram escritas estão preenchidas com o símbolo branco.

2. Um cabeçote, que pode ler e escrever símbolos na fita e mover-se para a esquerda e para a direita.

3. Um registrador de estados, que armazena o estado da máquina de Turing. O número de estados diferentes é sempre finito e há um estado especial denominado estado inicial com o qual o registrador de estado é inicializado.

4. Uma tabela de ação (ou função de transição) que diz à máquina que símbolo escrever, como mover o cabeçote (para esquerda e para direita) e qual será seu novo estado, dados o símbolo que ele acabou de ler na fita e o estado em que se encontra. Se não houver entrada alguma na tabela para a combinação atual de símbolo e estado então a máquina para.

. Deste modo, podemos dizer que uma máquina de Turing (com uma fita) é usualmente definida como uma Tupla $M = (Q, \Sigma, \Gamma, s, b, F, \delta)$, em que cada um desses símbolos podem ser caracterizados da seguinte forma:

- O símbolo Q representa um conjunto finito de estados;
- O símbolo Σ representa um alfabeto finito de símbolos;
- O símbolo Γ representa o alfabeto da fita (conjunto finito de símbolos);
- Os três seguintes símbolos $s \in \Sigma \times Q$ representam uma função que designa o estado inicial da máquina;
- Os três seguintes símbolos $b \in \Sigma \times Q$ designam o símbolo branco (o único símbolo que se permite ocorrer na fita infinitamente em qualquer passo durante a computação);
- F designa o conjunto dos estados finais da máquina;
- \Rightarrow é uma função parcial chamada função de transição, onde \leftarrow é o movimento para a esquerda e \rightarrow é o movimento para a direita.

Como podemos ver nessa descrição formal, uma máquina de Turing é um modelo de uma máquina abstrata, quer dizer, é apenas um modelo lógico-conceitual. Tal modelo abstrato foi projetado para realizar computações, efetuar processos computacionais. Os processos computacionais¹⁹ efetuados por uma máquina de Turing consistem em uma manipulação

ordenada/sequenciada/programada de símbolos (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519).

Designa-se por “fita” de uma máquina de Turing uma fita que pode ser dividida em casas/quadrados/células, em que cada um(a) desses(as) casas/quadrados/células estão escritos um dos símbolos do alfabeto que essa máquina possui e que é capaz de reconhecer (é claro que isso inclui a possibilidade de não poder ter nada escrito nas casas/quadrados/células da fita dessa máquina, possibilidade essa em que a casas/quadrados/células podem estarem em branco, isto é, pode-se dizer que na fita está escrito o símbolo branco). A cabeça de leitura pode estar posicionada, a cada momento do tempo, sobre uma das células da fita (BRANQUINHO; MURCHO; GOMES, 2006, p. 515-519).

A máquina de Turing pode encontrar-se num estado, representado pelos seguintes símbolos $q \in Q$, que é dito o estado da máquina em determinado instante de tempo. Bem, se o estado da máquina é (está) ativo, então a máquina se encontra numa situação que podemos dizer ser uma situação ativa. Caso contrário, pode-se dizer que a máquina se encontra em uma situação passiva (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519).

A máquina de Turing sempre poderá ser encontrada numa condição que é comum chamar de condição finita: isso quer dizer mais nada que, mesmo que a fita seja infinita (para esquerda e para a direita), a cada instante de tempo somente pode estar inscrito um símbolo próprio num número finito de casas, ou que a cada período de tempo, somente num número finito de casas é que pode estar inscrito um símbolo próprio (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519).

Deste modo, um processo computacional é executado por esse dispositivo (a máquina de Turing) que pode ler/escrever símbolos em uma fita dividida em quadrados. A cabeça de leitura/gravação move-se em qualquer direção ao longo da fita e a um quadrado por vez. O quadrado que é “lido” em cada etapa é conhecido como “quadrado ativo”. A regra que está sendo executada determina o que se convencionou chamar “estado” da máquina (POZZA, PENEDO, 2002).

1.2 ARDUINO E TINKERCAD

De acordo com as informações contidas em sua página na rede mundial de computadores, o Arduino é considerado a placa eletrônica mais usada para

prototipação no mundo. Capaz de construir avançados modelos de produtos, essa ferramenta possui o seu hardware e software distribuídos gratuitamente para a comunidade de desenvolvimento de embarcados. Criado na Itália no ano de 2005, busca ser uma plataforma de prototipagem eletrônica de fácil manuseio e de baixo custo, permitindo que qualquer pessoa interessada em projetos eletrônicos e ambientes interativos possa desfrutar dos seus recursos. Há uma alta diversidade de modelos disponíveis e o usado neste projeto foi o Arduino UNO REV3 implementado no software TinkerCad.

O Tinkercad é uma aplicação gratuita e fácil de usar para projetos 3D, componentes eletrônicos e codificação. É utilizado por professores, crianças, amadores e projetistas para imaginar, projetar e fabricar qualquer coisa.

Essa ferramenta representa os diversos componentes de forma muito realista, incluindo um ambiente de desenvolvimento para o Arduino idêntico ao real. A ferramenta está disponível online, devendo ser criada uma conta para o usar. É possível partilhar os ficheiros dos projetos entre vários utilizadores, nomeadamente entre colegas do mesmo grupo e/ou entre alunos e professores.

2 DESENVOLVIMENTO

Para esse trabalho a primeira etapa foi a escolha da Linguagem que no meu caso escolhi implementar uma máquina na qual dado um número binário conseguimos obter se ele é ou não divisível por três, após a escolha foi criado o diagrama de estados e a gramática como exemplificado abaixo, a parte matemática não será descrita nesse trabalho visto que não é o intuito do mesmo.

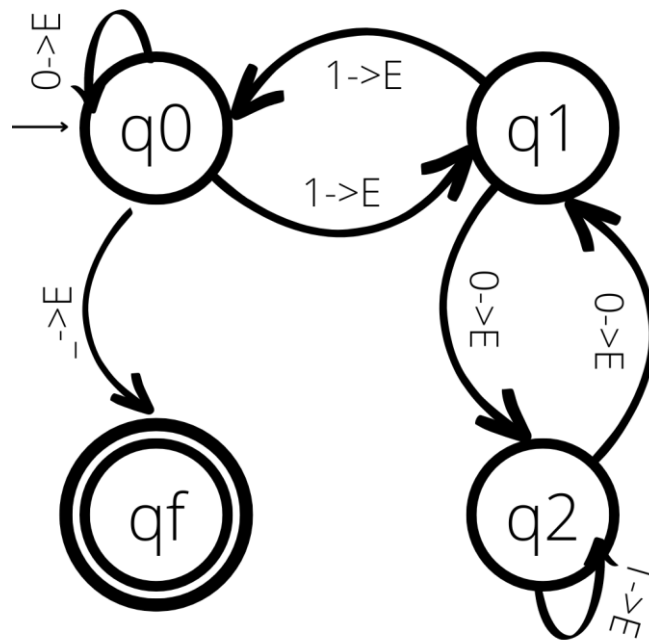


Figura 1 – Diagrama de Estados.

A formalização da máquina é apresentada abaixo, sendo a mesma formalização inicialmente elencada na introdução do trabalho.

$MT = (\{0, 1\}, \{q_0, q_1, q_2, q_f\}, \{0, 1, \Gamma\}, \delta, \{q_0\}, \{q_f\}, \{q_0, q_1, q_2\})$

$\delta \{q_0, 0$

$q_0, 0, >$

$q_0, 1$

$q_1, 1, >$

$q_1, 0$

$q_2, 0, >$

$q_1, 1$

$q_0, 1, >$

q2,0

q1,0,>

q2,1

q2,1,>

q0,_

qAccept,_,-}

Com a análise e visualização do diagrama de estados podemos notar que possuímos 3 estados que podemos elencar em cases para colocar no nosso hardware, as transições podemos notar que são bem parecidas com condições e no meu caso utilizei de ifs para transladar as mesmas. Dito isso foi se criado um circuito no TinkerCad que é uma ferramenta online de design de modelos 3D em CAD e também de simulação de circuitos elétricos analógicos e digitais, desenvolvida pela Autodesk. Por ser gratuito e fácil de usar, encontramos nele uma oportunidade de criar circuitos sem um gasto elevado ou totalmente gratuito, visto que eu não possuía todos os componentes necessários para confecção do circuito essa plataforma conseguiu suprir bem esse empecilho e comprar após a implementação apenas os componentes necessários.

O código implementado nesse hardware está descrito abaixo, no repositório do github temos o código que foi implementado no Arduino, a única diferença entre eles é o tamanho da fita que no Arduino foi preciso diminuir para 4 bits devido a quantidade de portas usadas para o display, como podemos notar nesse código todas as nossas portas são de entrada, comprovando assim o que foi dito anteriormente, utilizamos também de um display de led apenas por estética e

visualização mais clara da nossa fita e do número, na qual é relatada como um array de inteiros, ou seja nossa fita nesse caso é uma fita finita visto que possuímos uma quantidade de entradas finitas e nosso algoritmo não demanda de muitos passos vazios. O cabeçote da fita é um valor inteiro que conforme o esquema indica para ir a esquerda acontece um incremento da variável, e caso precisasse ir para a direita aconteceria um decremento do mesmo.

```
#include <Adafruit_LiquidCrystal.h>
Adafruit_LiquidCrystal lcd_1(0);
int fita[8];
int cabecote = 0;
int estado = 0;
bool continua = true;
bool divisivel = false;
bool pegarvalor = true;
void setup()
{
    Serial.begin(9600);

    pinMode(0, INPUT);
    pinMode(1, INPUT);
    pinMode(2, INPUT);
    pinMode(3, INPUT);
    pinMode(4, INPUT);
    pinMode(5, INPUT);
    pinMode(6, INPUT);
    pinMode(7, INPUT);
    pinMode(8, INPUT);
    lcd_1.begin(16, 2);
    lcd_1.print(" Maquina Turing ");
}

void loop()
{
    if (digitalRead(8) == 1)//MAQUINA LIGADA
    {
        if (continua)//FITA NÃO ACABOU AINDA
        {
            lcd_1.clear();
            if(pegarvalor){
                fita[0] = digitalRead(7);
                fita[1] = digitalRead(6);
                fita[2] = digitalRead(5);
                fita[3] = digitalRead(4);
                fita[4] = digitalRead(3);
                fita[5] = digitalRead(2);
                fita[6] = digitalRead(1);
                fita[7] = digitalRead(0);
                pegarvalor=false;
            }
            lcd_1.setCursor(0, 0);
            for (int i = 0; i < 8; i++){
                lcd_1.print(fita[i]);
            }

            Serial.print("CABECOTE ");
            Serial.println(cabecote);
            lcd_1.setCursor(0, 1);

            for (int i = 0; i < 8; i++) {
                if (i == cabecote) {
                    lcd_1.print("|");
                }
                else {
                    lcd_1.print(" ");
                }
            }
        }
    }
}
```

```

    lcd_1.print(" EST: ");
    lcd_1.print(estado);

    switch (estado)
    {
    case 0:
        if (fita[cabecote] == 1){
            estado = 1;
        }
        if (cabecote >= 8) {
            estado = 3;
            continua = false;
        }
        break;
    case 1:
        if (fita[cabecote] == 1){
            estado = 0;
        }
        if (fita[cabecote] == 0) {
            estado = 2;
        }
        if (cabecote >= 8){
            continua = false;
        }
        break;
    case 2:
        if (fita[cabecote] == 0) {
            estado = 1;
        }
        if (cabecote >= 8){
            continua = false;
        }
        break;
    }
    cabecote += 1;

    if (estado == 3) {
        continua = false;
        divisivel = true;
    }
}
else{//FITA JA ACABOU
    lcd_1.clear();
    lcd_1.setCursor(0, 0);
    for (int i = 0; i < 8; i++){//COLOCAR VALOR
        lcd_1.print(fita[i]);
    }
    if (divisivel == true) {//E DIVISIVEL
        lcd_1.setCursor(0, 1);
        lcd_1.print("DIVISIVEL");
    }
    else{//NAO DIVISIVEL
        lcd_1.setCursor(0, 1);
        lcd_1.print("NAO DIVISIVEL");
    }
}
}
else{//DESLIGADO
    lcd_1.clear();
    lcd_1.setCursor(0, 1);
    lcd_1.print(" DESLIGADO ");
    Serial.println("DESLIGADO");
    cabecote = 0;
    estado = 0;
    continua = true;
    divisivel = false;
    pegarvalor=true;
}
delay(500);
}

```

Figura 4 – Código inserido no Hardware.

Já um segundo algoritmo foi feito sem implementação em Hardware, ele segue o mesmo conceito e lógica do anterior, o mesmo foi feito em linguagem C, que é considerada uma linguagem de alto nível genérica, e pode ser usada em diversos tipos de projeto, como a criação de aplicativos, sistemas operacionais, drivers, entre outros. A escolha da mesma é devido a uma das suas grandes vantagens que é a capacidade de gerar códigos rápidos, ou seja, com um tempo de execução baixo. Além disso, a programação em C é bastante simplificada, pois sua estrutura é simples e flexível.

```
#include <stdio.h>
#define true 1
#define false 0

int main()
{
    int fita[8];
    int cabecote = 0;
    int estado = 0;
    int continua = true;
    int divisivel = false;
    int digitalRead;

    for (int i = 0; i < 8; i++) {
        printf("Digite o binario [%d]:", i);
        scanf("%d", &fita[i]);
    }
    printf(" Maquina Turing ");

    scanf("%d", &digitalRead);
    if (digitalRead == 1) // MAQUINA LIGADA
    {
        while (continua) { // FITA NÃO ACABOU AINDA
            printf(" EST: %d\n", estado);
            for (int i = 0; i < 8; i++){
                printf("%d", fita[i]);
            }
            printf("\n");

            for (int i = 0; i < 8; i++){
                if (i == cabecote){
                    printf("|");
                }
                else{
                    printf(" ");
                }
            }
            printf("\n");

            switch (estado)
            {
                case 0:
                    if (fita[cabecote] == 1){
                        estado = 1;
                    }
                    if (cabecote >= 8){
                        estado = 3;
                        continua = false;
                    }
                    break;
                case 1:
                    if (fita[cabecote] == 1){
                        estado = 0;
                    }
                    if (fita[cabecote] == 0){
                        estado = 2;
                    }
            }
        }
    }
}
```

```

        if (cabecote >= 8){
            continua = false;
        }
        break;
    case 2:
        if (fita[cabecote] == 0){
            estado = 1;
        }
        if (cabecote >= 8){
            continua = false;
        }
        break;
    }
    cabecote += 1;

    if (estado == 3){
        continua = false;
        divisivel = true;
    }
} // FITA JA ACABOU

for (int i = 0; i < 8; i++) { // COLOCAR VALOR
    printf("%d", fita[i]);
}
printf("\n");
if (divisivel == true) { // E DIVISIVEL
    printf("DIVISIVEL");
}
else{ // NAO DIVISIVEL
    printf("NAO DIVISIVEL");
}
}
else{ // DESLIGADO
    printf("DESLIGADO");
}
}

return 0;
}

```

Figura 5 – Código C.

Como pode-se notar é realmente a mesma lógica, utilizei de um compilador online para obter esses resultados, no qual traduziu a linguagem de alto nível escolhida, nesse caso linguagem C, para uma linguagem de baixo nível mais conhecida como linguagem de máquina.

3 RESULTADOS

Com as mesmas entradas trago aqui os resultados dos dois exemplos de implementação da mesma máquina de Turing, como podemos ver as duas máquinas funcionam de maneira equivalente e suas saídas são as mesmas que o resultado esperado, a única diferença é a forma de visualizar entre elas e que na implementada no hardware podemos ligar e desligar a máquina graças ao looping e a um botão de controle, já a implementada em software apenas executa uma única vez. Abaixo temos alguns exemplos dos resultados.

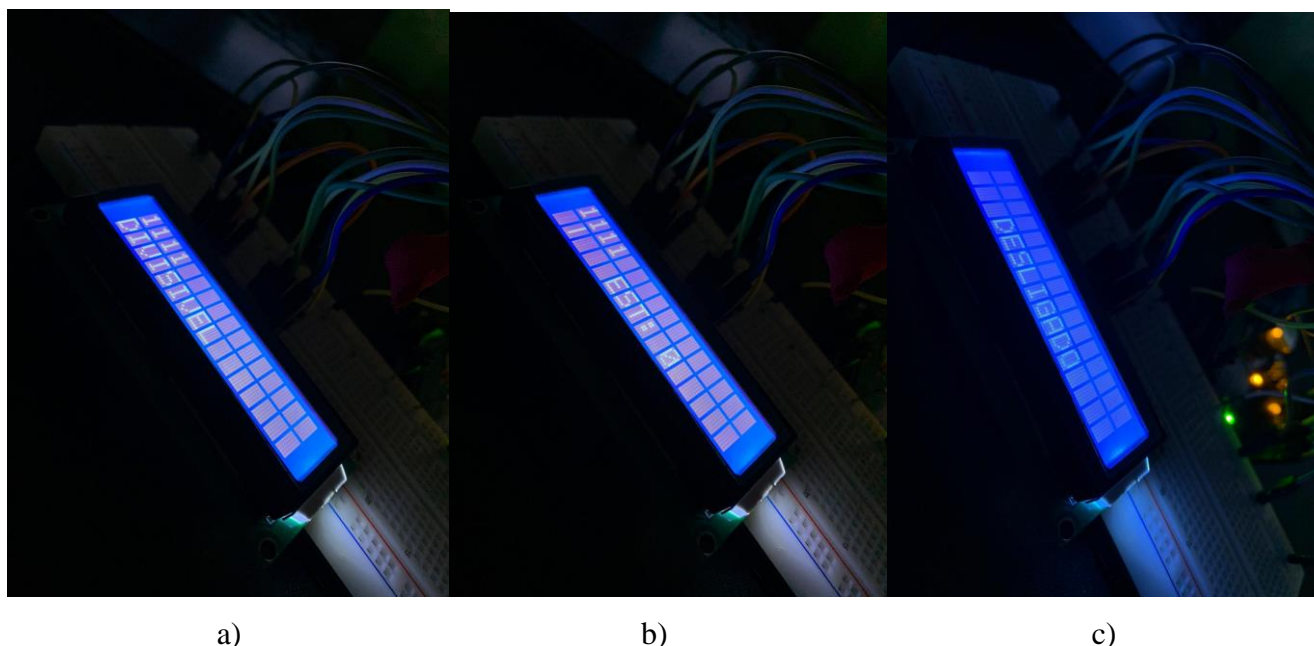


Figura 8- Saída Obtida com o Hardware implementado na prática a)Saída b) máquina ligada c) máquina desligada.

4 CONSIDERAÇÕES FINAIS

O software funcionou muito bem e conseguimos ver a máquina funcionando, o mesmo aconteceu com o código C que mesmo sem ser implementado em um hardware fez o mesmo papel e deverasmente bem.

Acredito que quando o circuito é realmente implementado com os componentes reais e não apenas no TinkerCad temos aprendido não apenas disciplinar, mas sim multidisciplinar e a implementação é bem mais interessante e consegue prender bem mais a atenção dos alunos, implementar ele apenas no software foi bem desafiador, mas com certeza a sensação de dever cumprido é maior com ele funcionando em sala com componentes reais.

O trabalho abriu bastante minha mente para o tema, fiquei bastante intrigada com a máquina e aumentei meus conhecimentos com a pesquisa, com toda certeza o intuito dele foi cumprido.

5 LINKS EXTERNOS

[GitHub:](#)

[Tinkercad:](#)

REFERÊNCIAS

EPSTEIN, Richard L.; CARNIELI, Walter. Computability: computable functions, logic, and the foundations of mathematics. London: An International Thomson Publishing Company London, 2000.

POZZA, Osvaldo Antonio; PENEDO, Sérgio. A máquina de Turing. Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de nov.2022.

WIKIPEDIA: a enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/Brasil>>. Acesso em: 22 de set. 2022.

BRANQUINHO, João; MURCHO, Desidério; GOMES, Nelson Gonçalves. Enciclopédia de termos lógico-filosóficos. São Paulo: Martins Fontes, 2006.

FARIAS, Gilberto; MEDEIROS, E. S. Introdução à Computação: 1. ed. UAB, 2013.

Arduino. About Us. Disponível em . Acesso em: 28 out. 2022. Arduino. Arduino Uno Rev3. Disponível em < <https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 28 out. 2022.

BOYLESTAD, NASHELSKY. Dispositivos Eletrônicos e Teoria de Circuitos. 8ªEdição. Prentice Hall. 2004.