**Análise e Síntese de Circuitos Combinacionais de uma Única Saída**

# Adriano Soares Rodrigues e Matheus Vidal de Menezes

Instituto Tecnológico de Aeronáutica (ITA)

São José dos Campos, São Paulo, Brasil.

{sadrianorod, matheusvidaldemenezes}@gmail.com

1. **Introdução**

No início do século XX, John Ambrose Fleming criou a primeira válvula termiônica, um dispositivo que alavancou a eletrônica e o rumo do desenvolvimento tecnológico como possibilitar a criação do rádio e inclusive de televisores. As válvulas termiônicas consistem basicamente por um invólucro de vidro, metal ou cerâmica a vácuo ou algum tipo de gás, com vários elementos metálicos internos chamados de anodo, catodo, grade e filamento conforme retratado na Figura 1.



Figura 1. Exemplo de válvula termiônica com filamento de 12V e base de 7 pinos.

Após o advento das válvulas termiónicas, foram desenvolvidos os transístores, representados na Figura 2, que são basicamente semicondutores capazes de desempenhar a mesma função de uma válvula, *i.e.*, controlar o fluxo de corrente, entretanto sem dissipar tanto calor, sendo menor e mais econômico. Por conta do sucesso da substituição das válvulas, estas tiveram sua produção em larga escala interrompidas.



Figura 2. Réplica do primeiro transístor, inventado no Bell Labs, 23 de dezembro, 1947 (Foto: Reprodução/Wikipédia).

É importante salientar também que, com o progresso tecnológico dos transístores, criou-se a famosa família de circuitos integrados TTL, *Transistor-Transistor Logic*, responsáveis pelo desenvolvimento de portas lógicas, que, na verdade, são circuitos transistorizados, tecnologia essencial para o surgimento dos computadores pessoais (*Personal Computer - PC)* de hoje.

1. **Objetivo**

Diante desse breve contexto histórico, o objetivo da primeira prática laboratorial de EEA-21 Circuitos Digitais mostra-se de grande importância. Isso, porque trata da familiarização das portas lógicas estudadas teoricamente e do aprendizado quanto a utilização destas em simulações, via o *software* *Quartus*® *13.01*, de circuitos básicos como detector de Fibonacci, bem como aqueles a base de multiplexadores.

1. **Tarefas**

**Problema 4.1)**

Entrada : 4 bits 🡪 O intervalo de valores possíveis é dado por [0, 15]

Pelo enunciado, temos

Em seguida, obtém-se a seguinte Tabela Verdade abaixo para esta questão.

Tabela 1. Tabela lógica, ou tabela verdade, referente ao problema 4.1 que possui X0, X1, X2 e X3 como entradas e F como saída onde se o número inserido pertencer ao conjunto de Fibonacci retorna 1, caso contrário retorna 0.



Por meio dos resultados obtidos por F, montamos o mapa de Karnaugh mostrado na Tabela 2 com o intuito de simplificar a expressão algébrica lógica dada pela Eq. 1.

Tabela 2. Mapa de Karnaugh onde as linhas com dois dígitos representam X1X0 e as colunas representam X3X2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X3X2**  **X1X0** | **00** | **01** | **11** | **10** |
| **00** | 0 | 0 | 0 | **1** |
| **01** | **1** | **1** | **1** | 0 |
| **11** | **1** | 0 | 0 | 0 |
| **10** | **1** | 0 | 0 | 0 |

(1)

Assim, com o auxílio do software *Quartus*® *13.01*, obtemos o circuito digital dado pela Figura 3.

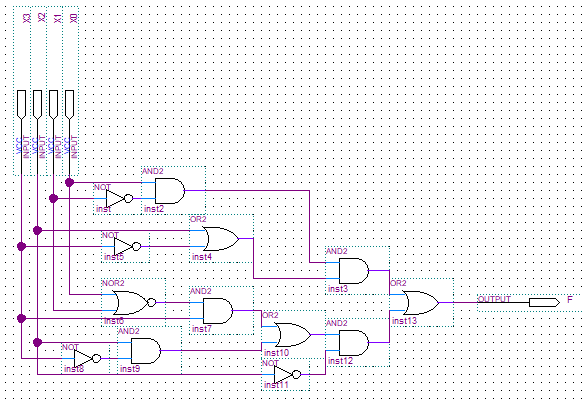


Figura 3. Descreve o circuito que exerce a função de detector de números pertencentes ao conjunto de Fibonacci no intervalo de 4 bits, em decimais de [0; 15], indicando 1 se pertence e 0 no caso negativo.

Após a compilação do sistema, obtemos o diagrama de temporização para o período dos oito valores possíveis, nota-se que atende a tabela verdade descrita acima concluindo-se que está de acordo com o esperado.

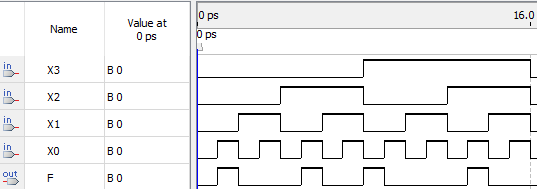


Figura 4. Diagrama de temporização para o circuito estudado e descrito na Figura 3.

**Problema 4.2)**



Para determinarmos a expressão lógica simplificada do circuito da Figura 5 lançaremos mão do método algébrico, com o auxílio dos Teoremas de De Morgan.

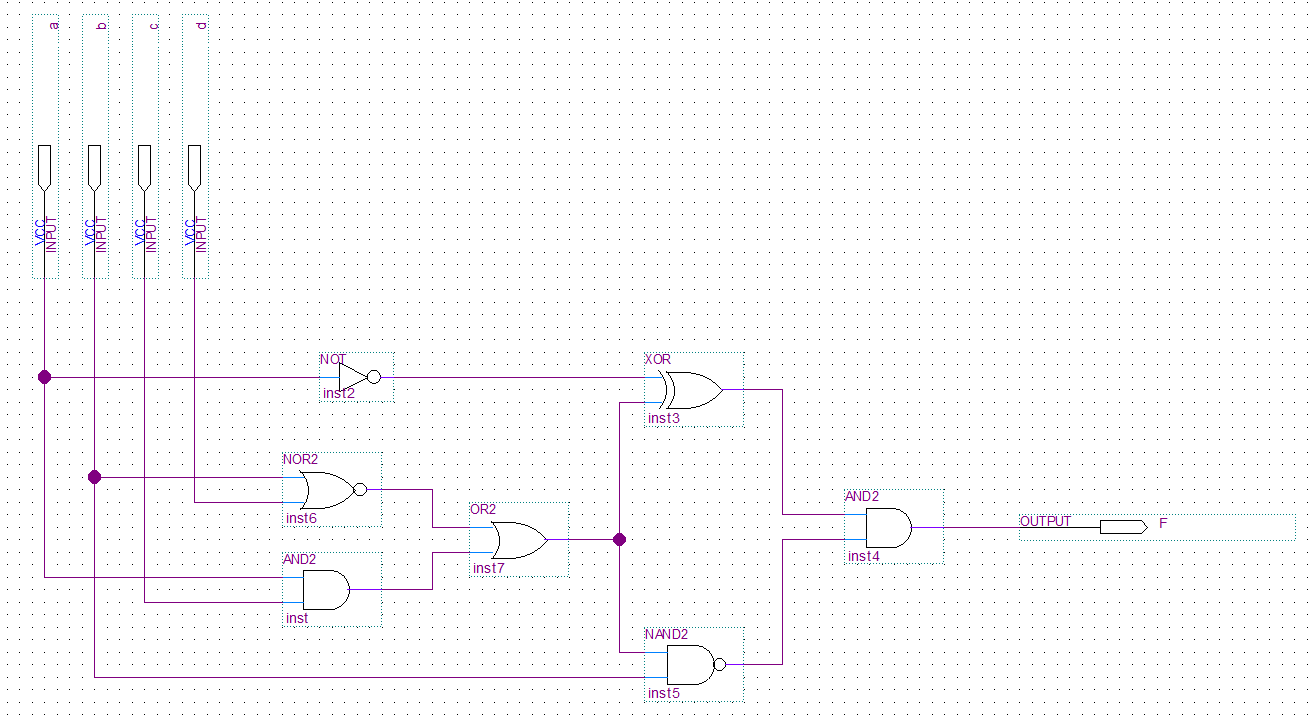


Figura 5. Circuito do problema 4.2 a ser simulado e simplificado.

Observe na Figura que o resultado da saída é dado por X



Após a compilação do sistema, obtemos o diagrama de temporização, conforme a Figura 5. Nota-se que atende a tabela verdade descrita acima concluindo-se que está de acordo com o esperado.

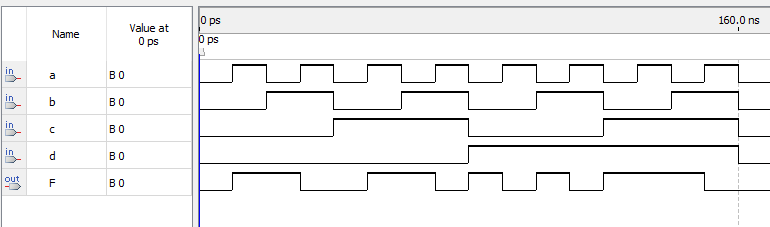


Figura 6. Diagrama de Temporização para o circuito dado no problema 4.2.







**Problema 4.3)**

Por meio do Diagrama de Temporização dado no problema, conseguimos montar a tabela verdade representada pela Tabela 3.

Tabela 3. Tabela lógica, ou tabela verdade, referente ao problema 4.3 que possui A, B e C como entradas e F como saída, obedecendo o Diagrama de Temporização.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | **F** |
| 0 | 0 | 0 | **0** |
| 1 | 0 | 0 | **1** |
| 0 | 1 | 0 | **0** |
| 1 | 1 | 0 | **1** |
| 0 | 0 | 1 | **0** |
| 1 | 0 | 1 | **0** |
| 0 | 1 | 1 | **1** |
| 1 | 1 | 1 | **1** |

A partir desta tabela, conseguimos montar o mapa de Karnaugh abaixo com o intuito de simplificar a expressão algébrica lógica. A representação matemática é a Eq. 2.

Tabela 4. Mapa de Karnaugh onde as linhas representam a entrada A e as colunas representam BC, respectivamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BC**    **A** | **00** | **01** | **11** | **10** |
| **0** | 0 | 0 | **1** | 0 |
| **1** | **1** | 0 | **1** | **1** |

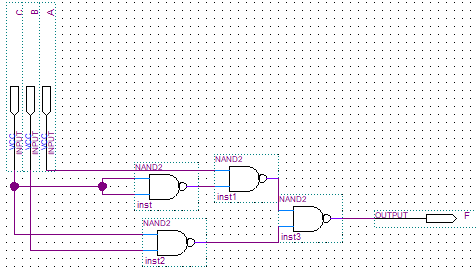
(2)

Mas como foi pedido em uma composição de apenas portas NAND de duas entradas, podemos manipular a Eq. 2, da seguinte maneira: aplicamos o operador NOT duas vezes na Eq. 2, obtendo-se a Eq. 3:

(3)  
Como e aplicando os Teoremas de De Morgan, obtemos um resultado conforme a Eq. 4.

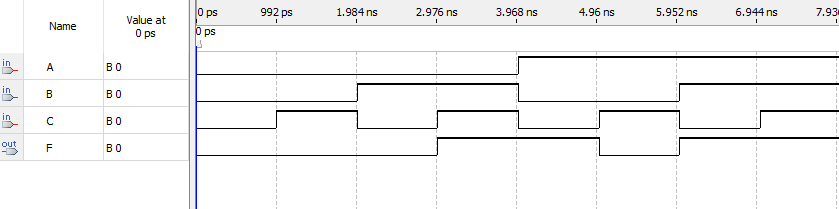
(4)

A simplificação representada na Eq. 4 é a resposta que servirá de base para a montagem do circuito digital no *software* Quartus®, dado pela Figura XX abaixo.



*Figura XX. Circuito Digital que representa os valores fornecidos na Tabela do problema 4.3.*

Após tal análise, temos o Diagrama de Temporização gerado pela simulação, dado pela Figura XX.



*Figura XX. Diagrama de Temporização obtido pela simulação do circuito apresentado.*

Ao compará-lo com o diagrama fornecido pelo problema, percebe-se que o do enunciado possuía um intervalo de deslocamento temporal, isso ocorre devido ao fato de as portas lógicas serem sistemas físicos feitos de transistores que necessitam de carregamento e descarregamento para realizar suas funções de controle de fluxo de corrente. Assim, esse intervalo de tempo intrínseco gera o intervalo temporal existente no gráfico dado, como era de se esperar. Em suma, como na simulação via *software* os componentes são considerados ideais, o *gap* observado na prática não ocorre na modelagem estudada.

**Problema 4.4)**

**Problema 4.5)**

1. **Conclusão**

Do experimento realizado, foi possível, embora toda simplicidade em sua metodologia, conseguiu obter resultados coerentes com a teoria de circuitos digitais

1. **Referências**

[1]

[2]

[3]

**Anexo A: Códigos Utilizados**

**A.1 Código Utilizado pelo Arduino para Emissão de uma Determinada Frequência pelo *Buzzer*:**

void setup() {

  pinMode(2, OUTPUT);

}

void loop() {

  tone(2, 1000);

}

**A.2 Código MATLAB que Tratou os Dados Experimentais:**

**A.2.1 grafOndasTubo.m**

%% parametros iniciais

f = 1000;

omega = 2\*pi\*f;

g = 9.81;

lambda = 340/f;

%% criacao da curva teorica 1s

t = 0:0.01:1;

y = cos(0.09\*f\*t + 0.09\*f\*t.^2).\*cos(0.09\*f\*t.^2);

dados = [t', y'];

save('dadosTeoricos\_1s.txt', 'dados', '-ascii');

%% criacao da curva teorica 12.5s

t = 0:0.01:12.5;

y = cos(0.09\*f\*t + 0.09\*f\*t.^2).\*cos(0.09\*f\*t.^2);

dados = [t', y'];

save('dadosTeoricos\_12.5s.txt', 'dados', '-ascii');

%% dados para calculo do erro

t = linspace(0, 12.4691, 2149);

y = cos(0.09\*f\*t + 0.09\*f\*t.^2).\*cos(0.09\*f\*t.^2);

dados = [t', y'];

save('dadosTeoricos\_erro.txt', 'dados', '-ascii');

%% grafico da curva teórica

figure(2)

plot(t,y)

**A.2.2 tratarDados.m**

tic;

A = load('sample-data.txt');

%% inicio do tratamento dos datos

% cada secao divide a quantidade de dados por 2

% isso eh feito para facilitar a vizualizacao dos graficos

%%

j = 1;

for i = 1:length(A)

    if(mod(i, 2) == 1)

        B(j, :) = A(i, :);

        j = j+1;

    end

end

%%

k = 1;

for i = 1:length(B)

    if(mod(i, 2) == 1)

        C(k, :) = B(i, :);

        k = k+1;

    end

end

%%

p = 1;

for i = 1:length(C)

    if(mod(i, 2) == 1)

        D(p, :) = C(i, :);

        p = p+1;

    end

end

%%

p = 1;

for i = 1:length(D)

    if(mod(i, 2) == 1)

        E(p, :) = D(i, :);

        p = p+1;

    end

end

%%

p = 1;

for i = 1:length(E)

    if(mod(i, 2) == 1)

        F(p, :) = E(i, :);

        p = p+1;

    end

end

%%

p = 1;

for i = 1:length(F)

    if(mod(i, 2) == 1)

        G(p, :) = F(i, :);

        p = p+1;

    end

end

%%

j = 1;

clear A;

for i = 1:length(F)

    if(mod(i, 2) == 1)

        A(j, :) = F(i, :);

        j = j+1;

    end

end

%%

k = 1;

clear B;

for i = 1:length(A)

    if(mod(i, 2) == 1)

        B(k, :) = A(i, :);

        k = k+1;

    end

end

%%

p = 1;

clear C;

for i = 1:length(B)

    if(mod(i, 2) == 1)

        C(p, :) = B(i, :);

        p = p+1;

    end

end

%%

p = 1;

clear D;

for i = 1:length(C)

    if(mod(i, 2) == 1)

        D(p, :) = C(i, :);

        p = p+1;

    end

end

%% imprime o tempo gasto, por curiosidade

t = toc;

sprintf('tempo gasto = %.4fs', t)

%% criacao dos graficos

% para 1s:

teoria\_1s = load('dadosTeoricos\_1s.txt');

figure(1)

plot(teoria\_1s(:, 1), teoria\_1s(:, 2), 'linewidth', 1.3);

hold on; grid on;

plot(C(1:174, 1), C(1:174, 2), 'linewidth', 1.3);

title('Comportamento das Curvas do Experimento até 1s');

xlabel('Tempo (s)', 'FontWeight', 'bold');

ylabel('Amplitude', 'FontWeight', 'bold');

axis([0, 1.01, -1, 1.01]);

legend('Curva teórica', 'Curva experimental');

hold off;

print -dpng -r300 images\curvaExp\_1s.png

% para 12.5s:

teoria\_12s = load('dadosTeoricos\_12.5s.txt');

figure(2)

plot(teoria\_12s(:, 1), teoria\_12s(:, 2));

hold on; grid on;

plot(C(:, 1), C(:, 2));

title('Comportamento das Curvas do Experimento Completo');

xlabel('Tempo (s)', 'FontWeight', 'bold');

ylabel('Amplitude', 'FontWeight', 'bold');

axis([0, 12.5, -1, 1.01]);

legend('Curva teórica', 'Curva experimental');

hold off;

print -dpng -r300 images\curvaExp\_12.5s.png

%%

% erros relativos

teoria\_erro = load('dadosTeoricos\_erro.txt');

errel(:) = abs(( teoria\_erro(:, 2) - C(:, 2) )./teoria\_erro(:, 2));

% tratamento de hampel para remover outliers

% (usou-se os 10 pontos em cada lado do ponto em questao)

% o ponto que desvia em mais de 15 unidades da media

hampelErrel = hampel(errel, 10, 15);

sumErrel = zeros(length(errel), 1);

sumErrel(1) = hampelErrel(1);

errel(1) = 0;

for i = 2:length(hampelErrel)

    sumErrel(i) = sumErrel(i-1) + hampelErrel(i);

end

mediaErros = zeros(length(sumErrel), 1);

for i = 1:length(sumErrel)

    mediaErros(i) = sumErrel(i)/i;

end

figure(3)

plot(C(:, 1), sumErrel, '.', 'linewidth', 1.3)

grid on;

title('Soma dos Erros Relativos na Amplitude');

xlabel('Tempo (s)', 'FontWeight', 'bold');

ylabel('Soma dos Erros Relativos', 'FontWeight', 'bold');

axis([0, 12.5, 0, 4200]);

print -dpng -r300 images\sumErrosRelativos.png

figure(4)

hold on; grid on;

plot(C(:, 1), hampelErrel, 'r.', 'linewidth', 1.3)

plot(C(:, 1), mediaErros, 'b', 'linewidth', 1.5)

title('Erros Relativos na Amplitude com Relação ao Modelo Teórico');

xlabel('Tempo (s)', 'FontWeight', 'bold');

ylabel('Erro Relativo', 'FontWeight', 'bold');

axis([0, 12.5, -2, 20]);

legend('Erro relativo a cada ponto', 'Média estatística dos erros');

hold off;

print -dpng -r300 images\errosRelativos.png

figure(5)

plot(A(:, 1), A(:, 2))

grid on;

title('Curva Experimental com Alta Densidade de Pontos');

xlabel('Tempo (s)', 'FontWeight', 'bold');

ylabel('Amplitude', 'FontWeight', 'bold');

axis([0, 12.5, -1, 1.01]);

%print -dpng -r300 images\curvaExp\_muitosDados\_12.5s.png

**A.3 Código no Wolfram Mathematica da Simulação do Modelo Teórico:**

(\*Gráfico do experimento\*)

f=1000; (\*A fonte sonora será de 400 Hz\*)

g=9.81; (\*Aceleração da gravidade em m/s^2\*)

L=0.5;(\*Altura inicial da coluna de água (m)\*)

t0=0; (\*Tempo inicial\*)

tfinal= Sqrt[2\*L/g]; (\*Tempo total de simulação\*)

y=Cos[90\*t+90\*t^2]\*Cos[90\*t^2];

Plot[y,{t,t0,tfinal}]

(\*Gráfico Interativo\*)

tfim=Sqrt[2\*L/g];

(\*tfim=0.32s utilizando L (comprimento inicial da coluna de água)=0.5 m\*)

Manipulate[Plot[Cos[2\*3.14\*f\*t+(3.14/2)\*(2\*g\*f\*(t^2)/340+1)]\*Cos[(3.14/2)\*(2\*g\*f\*(t^2)/340+1)],{t,0,tfim}],{f,200,2000}]