

Toolbox de Circuitos Lógicos para Projeto de Sistemas Digitais

Alexandre M. Pessoa, Lígia M. C. Sousa e Rômulo N. C. Almeida

Resumo—A Lógica Digital constitui a base das memórias e dos processadores modernos, sendo largamente empregada em sistemas de comunicação e processamento de sinais. Com este vasto horizonte de aplicações, recursos computacionais foram desenvolvidos para descrever o comportamento de funções lógicas complexas e dispositivos de memória (*flip-flops*). Entretanto, *softwares* como o *Matlab* ainda carecem de ferramentas capazes de analisar e simplificar circuitos lógicos digitais. Este trabalho apresenta uma ferramenta computacional baseada em *Matlab* capaz de abordar problemas que envolvam tais sistemas.

Palavras-Chave—Lógica digital, circuitos combinacionais e sequenciais.

Abstract—The Digital Logic, which constitutes the basis of modern processors and memories, has been largely used in communication systems and signal processing. Due to this wide horizon of applications, computational resources were developed to describe the behavior of complex logic functions and memory devices (*flip-flops*). However, softwares like *Matlab* still lack of tools capable of and simplify digital logic systems. This paper presents a computational tool based on *Matlab* capable of approach issues regarding such systems.

Keywords—Digital logic, combinational and sequential circuits.

I. INTRODUÇÃO

A análise e projeto de circuitos lógicos digitais para aplicações que envolvem grandes quantidades de variáveis exigem um alto processamento computacional. Isso ocorre devido a complexidade exponencial dos algoritmos envolvidos [1]. Dessa forma, vê-se a necessidade de desenvolver ferramentas para auxiliar no projeto e análise de tais circuitos.

Este trabalho apresenta um *Matlab toolbox* que foi desenvolvido com a finalidade de projetar e analisar circuitos lógicos digitais, tendo objetivo de servir como ferramenta de apoio em cursos de graduação e em projetos que envolvam o assunto.

A seção II apresenta os recursos do *toolbox* proposto referente a circuitos combinacionais, onde será mostrado as funções e uma *GUI* (*Graphical User Interface*) que permitem manipular expressões booleanas e desenharem o circuito lógico de tais expressões. A seção III compreende a parte do *toolbox* proposto referente a circuitos sequenciais síncronos, onde será mostrado as funções e uma *GUI* que permitem projetar e analisar máquinas de estados finitos [3]. A seção IV apresenta e discute os resultados obtidos a partir do emprego do *toolbox*, e por fim, na seção V são feitas as conclusões sobre o trabalho proposto.

Alexandre Pessoa, Lígia Sousa e Rômulo Nunes, Universidade Federal do Ceará (UFC). E-mails: alexandremtspss@gmail.com, ligiacsousa@ufc.br, rnunes@dee.ufc.br

II. EMPREGO DO *Toolbox* NO PROJETO DE CIRCUITOS COMBINACIONAIS

Circuitos combinacionais consistem em um tipo de circuito lógico onde a saída do mesmo é determinada univocamente a partir da entrada [2]. Este tipo de circuito lógico é descrito por uma tabela verdade ou uma função que faz uso das três operações lógicas fundamentais: *and*, *or* e *not*. O modelo da função que representa a saída S de um circuito combinacional e que possui como entrada o conjunto de bits a_1, a_2, \dots, a_n é dado por:

$$S = f[a_1, a_2, \dots, a_n]. \quad (1)$$

De modo geral, há interesse em expressar a equação (1) utilizando o menor número possível de operações lógicas, ou seja, obter a expressão mínima da função f . Para tal fim foi desenvolvida uma função (*ffbool*) que faz uso do algoritmo de McCluskey para expressar f de forma mínima [4]. A tabela I fornece as demais funções desenvolvidas para o *toolbox*, as quais trabalham com circuitos combinacionais.

TABELA I
FUNÇÕES QUE TRABALHAM COM CIRCUITOS COMBINACIONAIS.

Funções	Descrição
ffbool	reduz expressões booleanas à forma mínima
graytobin	converte código gray para código binário
binotogray	converte código binário para código gray
tabtrue	fornece a tabela verdade de expressões booleanas
mapak	gera o mapa de Karnaugh de uma função booleana
dcircuito	desenha o circuito lógico de uma expressão booleana
system_function	auxilia no controle da janela gráfica

Com o propósito de simplificar o uso das funções da tabela I, foi desenvolvida uma *GUI* (figura 1.b) intitulada *combinacional*. O uso dessa *GUI* permite a análise e construção de circuitos combinacionais de forma mais direta e de fácil interação com usuário.

III. EMPREGO DO *Toolbox* NO PROJETO DE MÁQUINAS DE ESTADOS FINITOS

Uma máquina de estados finitos (MEF) é um tipo de circuito sequencial síncrono capaz de reter informações. Estas informações são provenientes da entrada da máquina e são armazenadas em dispositivos de memória conhecidos como *flip-flops*. Dessa forma, uma MEF é composta da junção de circuitos combinacionais com *flip-flops* onde a saída da máquina depende da entrada e do estado (informação retida)

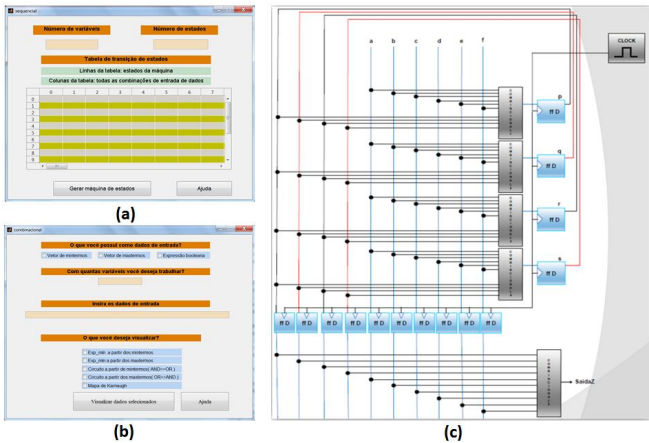


Fig. 1. (a) sequencial. (b) combinacional. (c) modelo de uma MEF

em que ela se encontra. A partir da tabela de transição de estados uma MEF é projetada construindo-se múltiplos circuitos combinacionais, onde cada circuito será responsável por controlar a entrada de um *flip-flop* [3]. Dessa forma, o *toolbox* proposto usa como modelo para uma MEF a figura 1.c, onde a entrada x , o estado s , os circuitos combinacionais c_i , ($i = 1, 2, 3, 4$) e a saída z foram definidos como:

$$x[t] = abcdef \quad (2)$$

$$s[t - 1] = pqr \quad (3)$$

$$s[t - 2] = tuvx \quad (4)$$

$$c_i[t] = f[x[t], s[t - 1]], i = 1, 2, 3, 4. \quad (5)$$

$$z[t - 1] = g[x[t - 1], s[t - 2]] \quad (6)$$

A partir das equações (4) e (6), é possível observar que a MEF proposta utiliza informações geradas a dois instantes de tempo anteriores. Dessa forma, a ferramenta implementa uma generalização do modelo de Mealy e Moore [3].

A partir da tabela de transição de estados a função *ffbool* é utilizada para obter as expressões booleanas que controlam as entradas dos *flip-flops* e a saída z da máquina. A tabela II contém a descrição das funções que trabalham com a MEF proposta.

TABELA II
FUNÇÕES USADAS NO PROJETO DE MEF'S.

Funções	Descrição
gme	fornece as expressões booleanas para os flip-flops D da MEF
equest	elimina estados redundantes em uma MEF

As funções da tabela II não fornecem uma representação gráfico para a MEF. Para isso foi utilizada uma aplicação conjunta das funções das tabelas I e II. Com este propósito foi desenvolvida uma *GUI* denominada *sequential* (figura 1.a) que permite a construção e análise de uma MEF de forma intuitiva e de fácil interação com o usuário.

IV. SIMULAÇÕES E ANÁLISE

O *toolbox* proposto foi testado em várias simulações que envolvem o projeto e análise de circuitos lógicos digitais já conhecidos. Neste trabalho são apresentados dois destes circuitos: o primeiro, figura 2.a, é um circuito combinacional conhecido como *display* de sete segmentos e o segundo, figura 2.b é uma MEF que representa a versão simplificada de um detector de início de transmissão em um sistema de comunicação, onde o início da transmissão foi definido como uma sequência de três 1's seguidos.

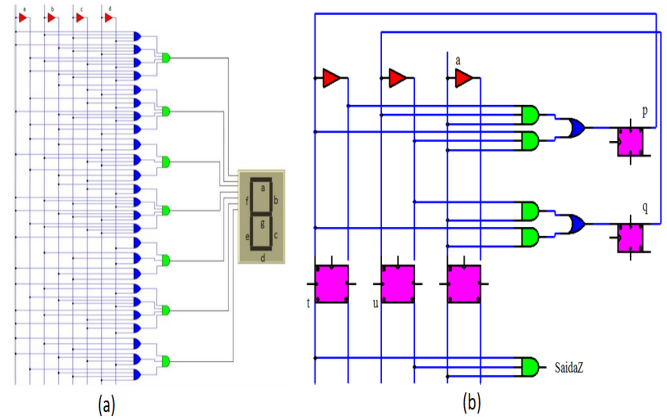


Fig. 2. (a) display 7 segmentos. (b) detector de início de transmissão.

Dos resultados obtidos foi possível observar que o *toolbox* reproduz corretamente os resultados correspondentes as implementações lógicas, confirmando a validade da ferramenta desenvolvida.

V. CONCLUSÃO

Neste trabalho, apresentou-se o desenvolvimento de um *toolbox* para a *Matlab* que visa a análise e projeto de circuitos lógicos em sistemas digitais. A partir dos resultados obtidos através das simulações, mostrou-se a eficácia da ferramenta no processo de análise e desenvolvimento de tais circuitos, podendo ser utilizada em diversas áreas de conhecimento como ferramenta didática em cursos de graduação e aplicada em projetos que envolvam o assunto.

AGRADECIMENTOS

Ao PET-UFC e aos cursos de Engenharia Elétrica e Computação da Universidade Federal do Ceará do Campus de Sobral-CE.

REFERÊNCIAS

- [1] Y. Tao, J. Cao, Y. Zhang, et al, "Using Module-level Evolvable Hardware Approach in Design of Sequential Logic Circuits", WCCI 2012 IEEE World Congress on Computational Intelligence June, 2012.
- [2] Mathyan M. Beppu, Victor R. L. do Amaral, Alexandre S. de la Vega. "Ferramenta de Auxílio Didático: Algoritmo de Quine-McCluskey em Lua.", XXXVIII Congresso Brasileiro de Educação em Engenharia (COBENGE), setembro de 2010.
- [3] S. Devadas, K. Kuertzer, "A Unified Approach to the Synthesis of Fully Testable Sequential Machines.", Transaction On Computer-Aided Design, Vol. 10. No. Jan 1991.
- [4] H. B. Min, E. S. Park, "Graph-theoretic algorithm for finding maximal supergates in combinational logic circuits", Proc.-Circuits Devices Syst., Vol. 143, No. 6, December 1996.