ENSINO DE PROJETOS DE SISTEMAS DIGITAIS E SINAIS MISTOS COM O SUPORTE DA FERRAMENTA COMPUTACIONAL MS²PSoC

TEACHING DIGITAL AND MIXED-SIGNAL DESIGN WITH THE SUPPORT OF COMPUTATIONAL TOOL MS²PSoC

Alexandre Araujo Amaral de Almeida[1], Alexandre César Rodrigues da Silva[2]

Abstract *— This paper presents a methodology for teaching mixed-signal design with the support of the computational tool MS²PSoC. The methodology consists in deploy in hardware a mixed-signal that was developed and simulated in software. The software used for system simulation is the Matlab/Simulink and the hardware PSoC for the implementation. The MS²PSoC tool is used to translate the system developed in Matlab/Simulink to the PSoC environment. Through this translation, the Simulink mixed-signal system can be implemented in hardware PSoC. To evaluate the methodology, a line code MLT-3 was developed in the Simulink environment and implemented in hardware PSoC trough the MS²PSoC translation. The results have shown that with the MS²PSoC the student can see the implementation of the system that was simulated in Simulink. The MS²PSoC can also be used for teaching hardware description language Verilog, since this is the language used to perform the translation.*

Index Terms — Matlab/Simulink, Mixed-Signal Design, PSoC (Programmable System-on-Chip), Verilog.

INTRODUÇÃO

O aumento na complexidade de projetos em sistemas embarcados tem reflexo direto no tempo de desenvolvimento. Esta complexidade é devido à natureza dos diferentes tipos de componentes envolvidos e o aumento de funcionalidades incorporadas em um único *chip*. Portanto, estão sendo realizadas pesquisas em ferramentas computacionais que tratam estas restrições em sistemas embarcados, tais como recursos computacionais (memória e processamento), consumo de energia sem comprometer o desempenho e a geração automática de código a partir de um alto nível de abstração.

O ensino de projetos de sistemas digitais e sinais mistos em conjunto com sistemas embarcados gera interesse nos estudantes, pois é possível verificar o funcionamento do circuito desenvolvido aplicado em *hardware*. Partindo de um diagrama de estados à visualização em *hardware*, fornece ao estudante a visão completa do desenvolvimento de um projeto (descrição, análise, simulação e implementação). Neste trabalho, a descrição, análise e simulação de um projeto, são realizadas com o ambiente Matlab/Simulink. Para a implementação, utiliza-se o *hardware* PSoC (*Programmable System-on-Chip*).

O Matlab/Simulink é um ambiente computacional que oferece suporte ao desenvolvimento de projetos em alto nível de abstração. O Simulink é uma ferramenta com interface de blocos que permite o desenvolvimento e análise de sistemas digitais e sinais mistos [1]. Esta ferramenta oferece também suporte a sistemas embarcados, realizando a implementação dos modelos desenvolvidos diretamente do ambiente Simulink. Entretanto, o PSoC é um sistema embarcado que o Simulink não realiza a implementação.

O PSoC é um sistema embarcado desenvolvido pela Cypress Semiconductor que contém sistema digital e analógico no mesmo *hardware*. Com este *hardware* é possível implementar modelos de sinais mistos em um mesmo *chip* sem a necessidade de componentes externos [2]. O ambiente computacional que realiza a comunicação entre *software* e *hardware* PSoC é o PSoC Creator. Com este ambiente é possível desenvolver sistemas para serem implementados em *hardware*, porém não contém um simulador.

Explorando esta falta de suporte do ambiente Simulink para o *hardware* PSoC e a falta de um simulador no ambiente PSoC Creator, este artigo apresenta uma ferramenta computacional chamada MS²PSoC que reproduz um sistema desenvolvido no ambiente Simulink para o ambiente PSoC Creator. Através desta reprodução, automatiza-se a implementação de um modelo descrito no ambiente Simulink em *hardware* PSoC.

Alguns trabalhos auxiliaram o desenvolvimento desta ferramenta através de ideias e conhecimento. Alguns destes trabalhos serão apresentados abaixo.

Chindris e Muresan [3] desenvolveram componentes presentes no ambiente PSoC Creator no ambiente Simulink. O objetivo era o de simular estes componentes, visto que o ambiente PSoC Creator não realiza a simulação. A continuação do trabalho seria o de gerar o código C destes componentes para serem implementados em *hardware* PSoC.

[1] Alexandre Araujo Amaral de Almeida, Mestrando no Programa de Pós-Graduação de Engenharia Elétrica, UNESP, câmpus de Ilha Solteira, aamaralalmeida@gmail.com
[2] Alexandre César Rodrigues da Silva, Professor Titular da Faculdade de Engenharia, UNESP, câmpus de Ilha Solteira, acrsilva@dee.feis.unesp.br

DOI 10.14684/ICECE.9.2015.111-115
© 2015 COPEC

Lambersky [4] apresentou em seu trabalho a geração automática de código para o TMS570 MCU a partir do Simulink. O autor utilizou o Code Composer Studio com o HalCoGen para traduzir o código C gerado pelo Simulink para o microcontrolador.

Estes trabalhos despertaram o interesse para o desenvolvimento da ferramenta computacional MS²PSoC.

Tranchero e Reyneri [5] propuseram a geração automática de código para circuitos assíncronos a partir do alto nível de abstração do Simulink. Estes circuitos assíncronos são baseados no sistema embarcado FPGA. O modelo descrito no ambiente Simulink é simulado e o código VHDL (*VHSIC Hardware Description Language*) sintetizável é gerado pelo CodeSimulink. O código VHDL é usado para realizar testes em baixo nível de abstração e os resultados são comparados com os de alto nível de abstração.

Laia e Cruvinel [6] utilizaram a geração automática de código HDL do ambiente Simulink para descrever o filtro de Kalman. Como o Simulink gera o HDL de apenas alguns componentes, os autores tiveram que adaptar os parâmetros do filtro.

Estes trabalhos mostraram algumas restrições presentes na geração de código HDL do ambiente Simulink. Visto que o Simulink gera o HDL para apenas alguns componentes, a ferramenta MS²PSoC gera o próprio código HDL. Obtendo assim, uma maior biblioteca de componentes.

Grout e Silva [7] propuseram uma ferramenta de tradução que gera o código VHDL-AMS de um modelo descrito em alto nível de abstração no ambiente Simulink. A ferramenta também gera a estrutura de arquivos necessária para que o modelo seja simulado no ambiente SystemVision. Deste trabalho surgiu a ideia de a ferramenta MS²PSoC gerar a estrutura de arquivos para o ambiente PSoC Creator.

A inovação proposta neste trabalho é a de automatizar o processo de implementação, em *hardware* PSoC, um sistema digital e sinais mistos descrito em alto nível de abstração no ambiente Simulink.

FERRAMENTA MS²PSOC

A ferramenta computacional denominada MS²PSoC realiza a tradução de um modelo descrito em alto nível de abstração no ambiente Matlab/Simulink para o ambiente PSoC Creator. Esta tradução é realizada através da interpretação das informações obtidas do modelo gerado pelo ambiente Simulink. O objetivo desta tradução é a implementação do modelo descrito no ambiente Simulink em hardware PSoC.

A motivação para o desenvolvimento da ferramenta MS²PSoC se encontra na falta de um simulador no ambiente PSoC Creator e na falta de suporte para o hardware PSoC no ambiente Matlab/Simulink.

A ferramenta MS²PSoC foi desenvolvida utilizando-se da linguagem de programação C. As diferenças da ferramenta desenvolvida com relação a ferramentas de tradução convencionais são a geração da estrutura de arquivos voltada para o ambiente PSoC Creator e a compatibilidade do código Verilog e C gerados com os compiladores do ambiente PSoC Creator.

Com a ferramenta é possível verificar o funcionamento do modelo descrito e simulado no ambiente Simulink em hardware PSoC, ajudando a despertar o interesse na aprendizagem de projetos de sistemas digitais e sinais mistos.

A ferramenta MS²PSoC possui uma biblioteca de componentes previamente estabelecidos para serem utilizados no desenvolvimento do modelo. Se algum componente que não estiver presente na biblioteca for utilizado no modelo, ocorrerá um erro no momento da tradução.

A metodologia utilizada na tradução de um modelo descrito em alto nível de abstração no ambiente Matlab/Simulink para o ambiente PSoC Creator é apresentada na Figura 3.

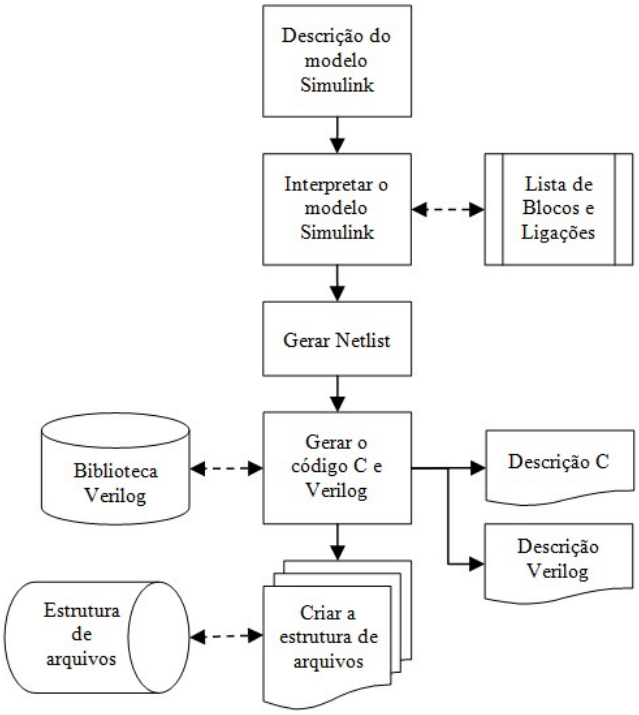

FIGURA 1
FLUXOGRAMA DA FERRAMENTA MS²PSOC.

A execução da ferramenta se inicia pela leitura do modelo Simulink. A partir desta leitura a ferramenta extrai todas as informações relevantes para a reprodução do modelo, tais como os componentes utilizados e suas respectivas ligações. Assim, gerando uma *netlist*. Com esta *netlist*, inicia-se a geração dos códigos C e Verilog, conforme o modelo descrito no Simulink.

Os componentes digitais possuem uma biblioteca com os respectivos códigos Verilog, portanto a utilização de um

componente que não esteja nesta biblioteca resultará em erro no momento da tradução. Analogamente, ocorrerá um erro na tradução caso se utilize algum componente analógico que não esteja na biblioteca da ferramenta.

Posteriormente à geração dos arquivos contendo a descrição do modelo, a estrutura de arquivos para o ambiente PSoC Creator é criada. Os arquivos e pastas gerados correspondem à estrutura mínima obrigatória para execução do ambiente. Após a geração desta estrutura de arquivos a ferramenta encerra o seu funcionamento.

Para avaliação da ferramenta computacional MS²PSoC utilizou-se um exemplo didático tratando-se da codificação MLT-3. Apresenta-se o projeto do codificador na seção a seguir.

CODIFICAÇÃO MLT-3

O MLT-3 (*Multi Level Transmission 3*) é uma linha de código usada para melhorar a eficiência na rede de transmissão FDDI (*Fiber Distributed Data Interface*) usando o padrão de par entrelaçado [8].

A codificação MLT-3 possui três níveis em que uma entrada de nível lógico alto causa a transição entre estes níveis. Para uma entrada de nível lógico baixo não ocorre mudança no nível atual. Os níveis de saída são +1, 0, -1. A alocação realizada para o desenvolvimento do modelo foi que o "01" representa o nível +1, o "00" representa o nível 0 e o "11" representa o nível -1. O diagrama de transição de estados da linha de código MLT-3, descrito pelo modelo de Mealy é apresentado na Figura 1.

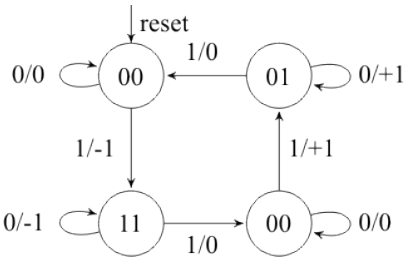

FIGURA 2
DIAGRAMA DE ESTADOS DA CODIFICAÇÃO MLT-3.

A partir do diagrama de transições de estado foi gerado um arquivo de entrada para a ferramenta computacional TABELA [9]. Esta ferramenta gera a tabela de transição de estados e realiza a minimização utilizando os elementos de memória escolhidos. Neste caso, o elemento de memória escolhido foi o flip-flop D.

Com as funções minimizadas através da ferramenta TABELA desenvolveu-se o modelo da codificação MLT-3 no ambiente Simulink, utilizando-se apenas componentes disponíveis na biblioteca da ferramenta MS²PSoC. O circuito é apresentado na Figura 2.

A simulação do circuito é apresentada na Figura 3. Nota-se que o flip-flop D utilizado é sensível à borda de descida, portanto as saídas são atualizadas na transição de nível lógico alto (1) para nível lógico baixo (0) a cada pulso do Clock.

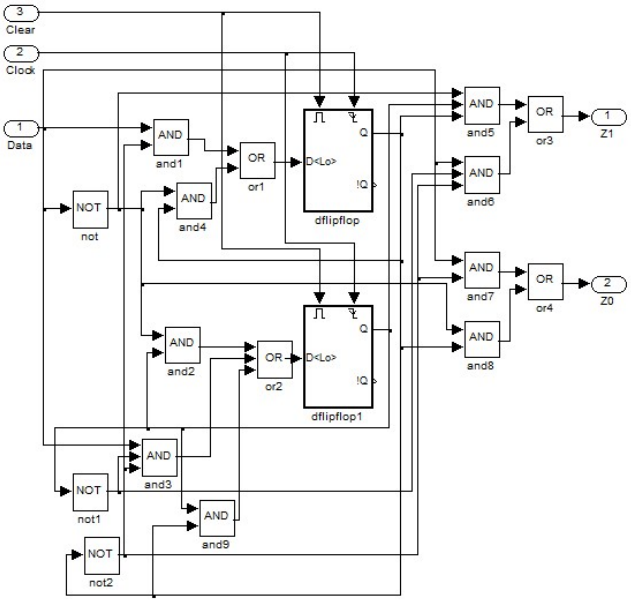

FIGURA 3
CIRCUITO DA CODIFICAÇÃO MLT-3 DESENVOLVIDO EM SIMULINK.

Através da simulação pode-se comprovar o funcionamento do circuito desenvolvido no ambiente Simulink. A cada pulso de Clock em que o sinal de entrada Data se encontra em nível lógico alto, ocorre a transição de estado do sinal de saída. E, com o sinal de entrada Data em nível lógico baixo, a saída se mantém inalterada.

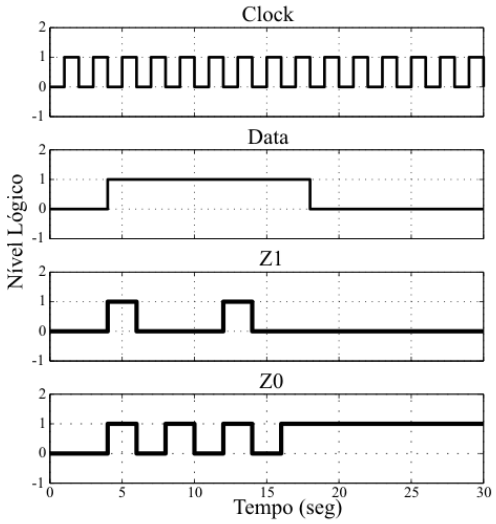

FIGURA 4
SIMULAÇÃO DO CÓDIGO MLT-3 NO AMBIENTE SIMULINK.

Para implementar o circuito desenvolvido em um hardware PSoC, traduziu-se o modelo utilizando-se a ferramenta computacional MS²PSoC. Através desta tradução pode-se verificar o funcionamento do código MLT-3 em hardware e avaliar a ferramenta desenvolvida.

AVALIAÇÃO

Para avaliar a ferramenta computacional MS²PSoC foi utilizada a codificação MLT-3. O modelo foi desenvolvido em ambiente Simulink utilizando-se somente componentes disponíveis na biblioteca da ferramenta. Em seguida o modelo foi simulado e verificado seu funcionamento.

A ferramenta MS²PSoC traduziu o modelo descrito em ambiente Simulink e gerou a estrutura de arquivos para o projeto ser aberto no ambiente PSoC Creator.

Após a tradução o projeto foi aberto no ambiente PSoC Creator e configurado os portos necessários para se realizar a implementação no hardware PSoC. O projeto foi implementado no Kit de Desenvolvimento CY8CKIT-030 do PSoC 3. Para o sinal de entrada Data e Clear foram utilizados botões de pressão e para o sinal Clock foi utilizado o componente *clock* do ambiente PSoC Creator com a frequência de 1 Hz. As saídas Z0 e Z1 foram conectadas ao osciloscópio digital (Politerm – modelo POL-15D).

Para avaliação do modelo traduzido e implementado no hardware PSoC, foi utilizado o mesmo sinal de entrada da simulação no ambiente Simulink apresentada na Figura 4. Os sinais de saída são apresentados nas Figuras 5 e 6.

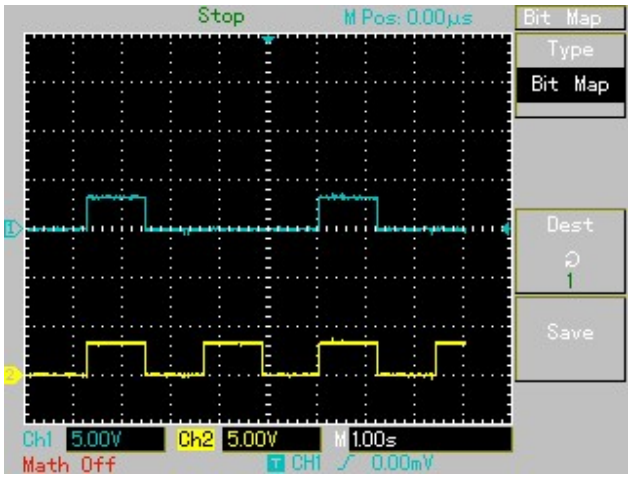

FIGURA 5
IMPLEMENTAÇÃO DA CODIFICAÇÃO MLT-3 COM SINAL DE ENTRADA ALTO.

Como projetado e previamente simulado, nota-se na Figura 5 que a cada pulso do Clock em que a entrada Data se encontra em nível lógico alto, ocorre à transição de estados do sinal de saída. Quando o sinal de entrada Data está em nível lógico baixo as saídas se mantêm inalteradas, como apresentado na Figura 6.

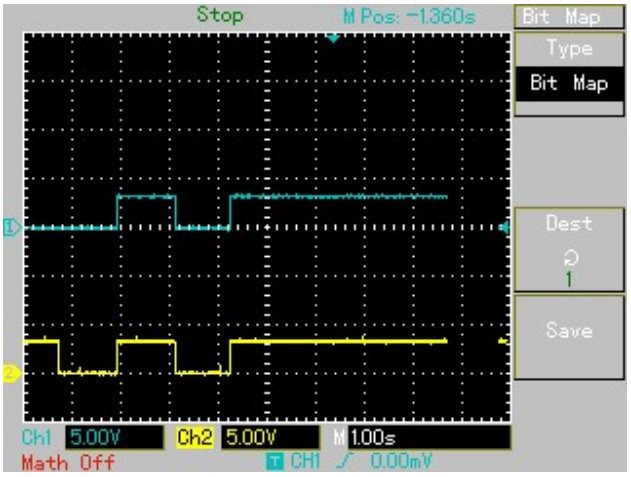

FIGURA 6
IMPLEMENTAÇÃO DA CODIFICAÇÃO MLT-3 COM SINAL DE ENTRADA BAIXO.

Nota-se que a tradução foi realizada com sucesso e os sinais obtidos através do osciloscópio confirmaram o funcionamento da ferramenta MS²PSoC. Através do uso da ferramenta foi possível verificar em hardware o funcionamento de um modelo descrito e simulado no ambiente Simulink.

Além da implementação em hardware do modelo desenvolvido, também é possível o ensino de linguagem de descrição de hardware Verilog com a ferramenta MS²PSoC. Pois a ferramenta utiliza esta linguagem para realizar a tradução do modelo descrito em Simulink. Um exemplo do código Verilog utilizado (flip-flop D) é apresentado na Figura 7.

```
module dflipflop (d, clk, clr, q, q1);
    input   d;   // 1
    input   clk; // trigger
    input   clr; // enable
    output  q;   // o1
    output  q1;  // o2

    reg     q;
    wire    q1;

    always @(negedge clk or negedge clr)
        if (clr == 1'b0) begin
            q <= 1'b0; end
        else begin
        q <= d; end

    assign q1 = ~q;

endmodule
```

FIGURA 7
EXEMPLO DO CÓDIGO VERILOG DE UM FLIP-FLOP D.

Comparando os resultados obtidos da simulação do modelo em ambiente Simulink e a implementação em hardware PSoC, comprova-se o funcionamento da ferramenta MS²PSoC. Visto o caráter didático, utilizou-se um estudo de caso com um sistema digital, baseado em portas lógicas e flip-flop D, mas o uso da ferramenta se estende para os modelos de sinais mistos.

CONCLUSÕES

Este artigo apresenta uma ferramenta computacional chamada MS²PSoC que realiza a tradução de um modelo descrito em alto nível de abstração no ambiente Matlab/Simulink para o ambiente PSoC Creator. O objetivo desta tradução é a implementação do modelo Simulink em hardware PSoC.

O Simulink é utilizado como *front-end* devido ao seu suporte a análise e simulação de sistemas digitais e sinais mistos. Como o Simulink não disponibiliza o suporte ao sistema embarcado PSoC, escolheu-se este hardware para ser utilizado como *back-end.*

A ferramenta MS²PSoC realiza a tradução do modelo desenvolvido no Simulink através da descrição deste modelo em linguagem C e linguagem de descrição de hardware Verilog. A MS²PSoC ainda gera a estrutura de arquivos necessária para que o modelo traduzido seja aberto no ambiente PSoC Creator.

Esta ferramenta pode ser utilizada para o ensino de sistemas digitais e sinais mistos visto que oferece a possibilidade para o estudante implementar em hardware PSoC o modelo projetado e simulado em ambiente Simulink. A ferramenta MS²PSoC oferece também a oportunidade do ensino da linguagem de descrição de hardware Verilog, pois é a linguagem utilizada na tradução de sistemas digitais.

Para avaliação da ferramenta foi desenvolvido e simulado a codificação MLT-3 em ambiente Simulink e posteriormente traduzido com a ferramenta MS²PSoC. Através desta tradução foi possível implementar e verificar o funcionamento do modelo descrito em hardware PSoC.

Comparando os resultados obtidos na simulação no ambiente Simulink e na implementação em hardware PSoC verifica-se o funcionamento da ferramenta computacional MS²PSoC.

AGRADECIMENTOS

Os autores registram seus agradecimentos ao Programa de Pós-Graduação da Faculdade de Engenharia de Ilha Solteira, UNESP - Universidade Estadual Paulista. Este trabalho tem o apoio financeiro do CNPq pelos processos de nº 309023/2012-2, nº 550134/2013-1 e nº 166837/2013-0.

REFERÊNCIAS

[1] MathWorks, "Documentação", 2013. [Online]. Disponível em: http://www.mathworks.com/

[2] C. S. Corporation, *PSoC Architecture TRM*, USA, 2014.

[3] G. Chindris and M. Muresan, "Deploying simulink models into system-on-chip structures", in *Electronics Technology, 2006. ISSE '06. 29th International Spring Seminar on*, May 2006, pp. 313–317.

[4] V. Lambersky, "Model based design and automated code generation from simulink targeted for tms570 MCU", in *Education and Research Conference (EDERC), 2012 5th European DSP*, Sept 2012, pp. 225–228.

[5] M. Tranchero and L. Reyneri, "Automatic generation of vhdl code for self-timed circuits from simulink specifications", in *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on*, Dec 2007, pp. 287–290.

[6] M. A. M. Laia and P. Cruvinel, "Using simulink to generate hdl code for validating an embedded kalman filter", in *Critical Embedded Systems (CBSEC), 2012 Second Brazilian Conference on*, May 2012, pp. 30–35.

[7] A. da Silva and I. Grout, "Ms2sv: Environment for translation of matlab / simulink models to vhdl-ams models", *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 9, no. 5, pp. 663–672, Sept 2011.

[8] J. Everitt, J. Parker, P. Hurst, D. Nack, K. Konda, and C. Raad, "A 10/100 mb/s cmos ethernet transceiver for 10baset, 100basetx, and 100basefx", in *Solid-State Circuits Conference, 1998. Digest of Technical Papers. 1998 IEEE International*, Feb 1998, pp. 210–211.

[9] A. C. R. da Silva, "Contribuição `a minimização e simulação de circuitos lógicos," *Tese de Mestrado*, UNICAMP, Campinas, 1989.