



## AMBIENTE DIDÁTICO PARA O ENSINO DE MÁQUINAS DE ESTADOS FINITOS

**Alexandre César Rodrigues da Silva** – acrsilva@dee.feis.unesp.br

UNESP – Universidade Estadual Paulista, FEIS – Faculdade de Engenharia de Ilha Solteira, DEE – Departamento de Engenharia Elétrica

Av. Brasil, 56, Centro

15.385-000 – Ilha Solteira - SP

**Tiago da Silva Almeida** – tiagoalmeida@aluno.feis.unesp.br

UNESP – Universidade Estadual Paulista, FEIS – Faculdade de Engenharia de Ilha Solteira, DEE – Departamento de Engenharia Elétrica

Av. Brasil, 56, Centro

15.385-000 – Ilha Solteira – SP

**Silvano Renato Rossi** – srossi@fio.unicen.edu.ar

Universidad Nacional del Centro de La Provincia de Buenos Aires

Av. Del Valle 5737, B7400JWI, Olivarria, Argentina

**Resumo:** Neste trabalho é proposta uma metodologia para o ensino de máquinas de estados finitos. Para tanto, foi desenvolvida uma ferramenta denominada  $SF^2Tab^+VHDL$ , capaz de traduzir máquinas de estados finitos descritas pelo modelo de Mealy, modeladas em ambiente Stateflow<sup>®</sup>, que traduz o modelo em linguagem VHDL e em tabela de transição de estados correspondente. Com essa tradução é possível a interação com outras ferramentas já desenvolvidas, possibilitando um estudo comparativo entre diferentes formas de análise de um mesmo circuito e entre diferentes níveis de abstração da linguagem VHDL. Com isso, é obtida uma metodologia para o ensino de projeto de circuitos lógicos, em particular máquinas sequenciais, criando um ambiente de síntese que pode ser utilizado para fins didáticos.

**Palavras-chave:** Síntese, VHDL, máquina de estados finitos

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)

## 1 INTRODUÇÃO

O ensino de máquinas de estados finitos sempre foi considerado um grande desafio em decorrência da falta de ferramentas para simular o comportamento destes autômatos. Algumas metodologias têm utilizado Rede de Petri cuja transcrição emprega o ambiente PIPE (Dias, 2007), pois pode-se avaliar o comportamento do modelo antes da implementação. Entretanto, essa metodologia leva a um novo desafio que é o aprendizado de Rede de Petri, que não é uma tarefa trivial.

Nesse contexto surge a necessidade de se criar uma metodologia de ensino para auxiliar o aluno no aprendizado sobre máquinas de estados finitos empregando os conceitos utilizados tradicionalmente, ou seja, o diagrama de transição de estados.

Nesse sentido, foi desenvolvido nesse trabalho um software denominado SF<sup>2</sup>Tab<sup>+</sup>VHDL (*Stateflow<sup>®</sup> to TABELA or VHDL*) capaz de gerar diferentes descrições em linguagem VHDL (*VHSIC Hardware Description Language*) de modelos de máquinas de estados finitos descritas no ambiente Stateflow<sup>®</sup>. O programa SF<sup>2</sup>Tab<sup>+</sup>VHDL traduz um diagrama de transição de estados descrito pelo modelo de Mealy e gera o código VHDL no âmbito comportamental correspondente ou em uma tabela de transição de estados. A tabela de transição de estados servirá como entrada para o programa TABELA que realizará a otimização das funções booleanas de controle e de saída dessa determinada máquina de estados finitos.

## 2 MÁQUINAS SEQUENCIAIS

Máquinas de estados finitos são representações abstratas do circuito sequencial e ocupa um importante papel no ensino da eletrônica digital básica. Seu comportamento pode ser descrito como uma sequência de eventos que acontecem em instantes de tempo discreto. Vários são os exemplos de máquinas de estados finitos empregadas no nosso dia a dia como, por exemplo, codificadores, semáforos, etc.

Um circuito sequencial possui um número finito de entradas, constituindo o conjunto  $N$  das variáveis de entrada, ou seja,  $N = \{N_1, N_2, \dots, N_n\}$ . Possui, também, um conjunto finito  $M$  de saída, sendo  $M = \{M_1, M_2, \dots, M_m\}$ .

O valor contido em cada elemento de memória forma o conjunto  $K$  das variáveis de estado,  $K = \{K_1, K_2, \dots, K_k\}$ , e definem o estado atual da máquina. As saídas e as funções de transições internas que geram o conjunto  $S = \{S_1, S_2, \dots, S_s\}$  (próximo estado) dependem das entradas  $N$  e dos estados atuais  $K$  da máquina e são definidas através dos circuitos combinacionais. As máquinas sequenciais podem ser representadas pela seguinte equação:

$$K(t+1) = f[K(t), N(t)] \quad (1)$$

onde  $f$  é a função de transição de estados. O valor da saída  $M(t)$  é obtido através de duas formas:

$$M(t) = g[K(t)] \quad (2)$$

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)

ou

$$M(t) = g[K(t), N(t)] \quad (3)$$

onde  $g$  é a função de saída.

Uma máquina com propriedades descritas nas equações (1) e (2) é denominada *modelo de Moore* e uma máquina descrita através das equações (1) e (3) é denominada *modelo de Mealy* (SILVA, 1989).

Nesse trabalho considerou-se somente as máquinas descritas através do modelo de Mealy, onde a saída é determinada através do estado atual e da entrada da máquina.

Uma máquina de estados finitos pode ser descrita através de um diagrama de transição de estados. O diagrama de estados é um grafo orientado onde cada estado representa um nó e de cada nó emanam  $p$  arcos orientados correspondendo às transições de estado. Cada arco orientado é rotulado com a entrada que determina a transição entre os estados e saída gerada. Apresenta-se na “Figura 1” o diagrama de transição de estados de um circuito detector de três (0) consecutivos descrito através da máquina de Mealy (SILVA, 1989).

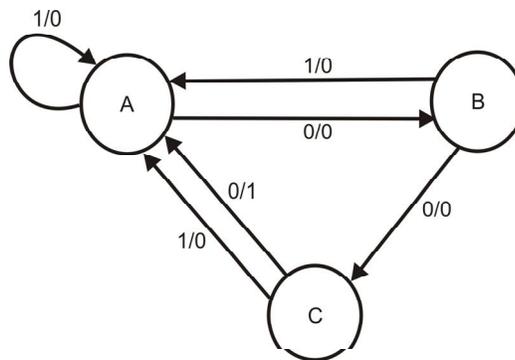


Figura 1. Diagrama de transição de estados de uma máquina de estados descrito pelo modelo de Mealy.

Utilizando elementos de memória projeta-se um circuito que opera de acordo com o comportamento descrito pelo diagrama de transição de estado.

A sistematização do ensino de projeto de circuitos digitais é extremamente conveniente, pois auxilia na prática de ensino, mostrando as tarefas cansativas e sujeitas a erros, permitindo uma análise crítica das soluções possíveis.

### 3 METODOLOGIA DE TRABALHO

#### 3.1 O programa TABELA

O programa TABELA foi desenvolvido para sintetizar máquinas de estados finitos e foi implementado em linguagem Pascal por pesquisadores da UNICAMP (Universidade Estadual de Campinas). O programa monta uma tabela de transição de estados armazenando-a no arquivo de saída. A partir desta tabela são obtidos os

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)

mintermos e os “*don't care states*” das funções de transição internas de todos os flip-flops e das saídas do circuito. Utilizando o algoritmo de minimização de funções booleana de Quine-McCluskey, estas funções são obtidas em suas fórmulas mínimas. O algoritmo de Quine-McCluskey é um método clássico que possui duas fases: a obtenção dos implicantes primos e a cobertura irredundante, onde a partir do conjunto de implicantes primos são obtidos os implicantes essenciais para a realização da função. Na “Figura 2” apresenta-se o diagrama funcional do programa TABELA.

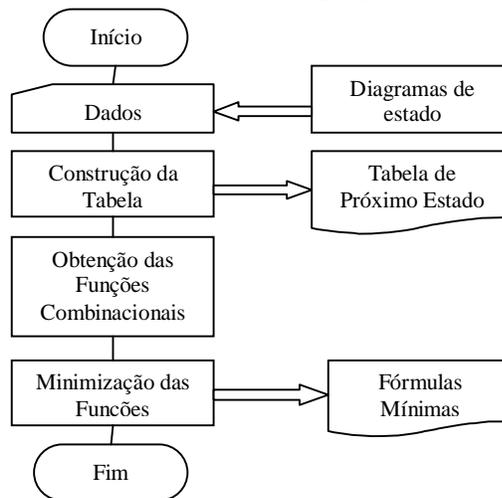


Figura 2. Diagrama de blocos do programa TABELA (SILVA, 1989).

Na “Figura 3” apresenta-se como deve ser o arquivo de entrada para o TABELA utilizando o mesmo exemplo para o circuito apresentado no “Figura 1”, que também será utilizado para a validação da metodologia de ensino proposta.

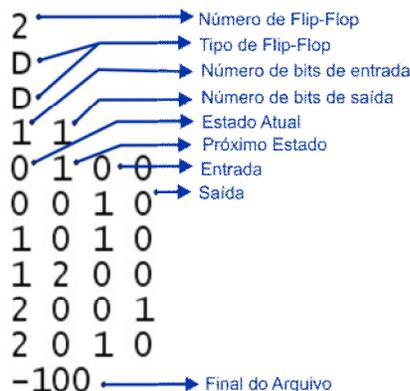


Figura 3. Arquivo de entrada para o programa TABELA

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)



O programa TABELA gera a tabela de transição de estados de uma máquina sequencial a partir de seu diagrama de estados e minimiza as funções de transições internas correspondentes aos elementos de memória utilizados e as funções de saída do circuito. Os dados solicitados pelo programa são: nome do dispositivo de saída de resultados (não deve conter a extensão), número de flip-flops, tipo de cada um deles (D ou JK), número de variáveis de entrada, número de variáveis de saída, tabela de próximo estado, forma: estado atual, próximo estado, entrada e saída. O final da descrição é representado pela notação “-100” (SILVA, 1989).

### 3.2 O programa SF<sup>2</sup>Tab<sup>+</sup>VHDL

Para auxiliar no ensino de projetos de máquinas de estados finitos desenvolveu-se uma metodologia e uma nova ferramenta de tradução denominada SF<sup>2</sup>Tab<sup>+</sup>VHDL (*Stateflow<sup>®</sup> to TABELA or VHDL*). O SF<sup>2</sup>Tab<sup>+</sup>VHDL realiza a tradução de máquinas de estados finitos descritas em alto nível de abstração no ambiente Matlab<sup>®</sup> / Simulink<sup>®</sup> / Stateflow<sup>®</sup> para a tabela de transição de estados, para que posteriormente seja feita a minimização do circuito pelo TABELA, e para a descrição comportamental correspondente em VHDL. Como a descrição em VHDL independe da tecnologia, as ferramentas de síntese comerciais se encarregam de sintetizar na tecnologia alvo, destacando-se que nesse trabalho utilizou-se o ambiente Quartus II e o FPGA Cyclone II da Altera, modelo EP2C20F484C7.

O ambiente Stateflow<sup>®</sup> é considerado uma extensão do Simulink<sup>®</sup>, que é utilizado para a modelagem e simulação de máquinas de estados finitos em diferentes modos de descrição e análise. Por fazer parte de um ambiente bastante difundido no meio acadêmico e de pesquisa (KARRIS, 2007), esse ambiente foi escolhido como alvo da tradução a ser realizada. Bem como a linguagem VHDL foi escolhida, por ser bastante difundida no meio acadêmico. Maiores detalhes sobre as linguagens de descrição de hardware podem ser encontrados em (D'AMORE, 2005).

O programa SF<sup>2</sup>Tab<sup>+</sup>VHDL possui uma interface gráfica simplificada que permite ao aluno selecionar qual arquivo deseja gerar, a tabela de transição de estados, a descrição através da VHDL comportamental, ou ambos.

O programa possui um menu chamado Ferramentas com as seguintes opções: Nova Tradução, utilizado para abrir o arquivo com a máquina de estados modelado em ambiente Stateflow<sup>®</sup> o modelo a ser traduzido; Traduzir, que realiza a tradução do modelo; e Sair, que finaliza a execução do programa.

O programa realiza a leitura do arquivo contendo a máquina de estados descrita através do modelo de Mealy, em ambiente Stateflow<sup>®</sup>. Em seguida, é localizada toda a estrutura da máquina de estados contida no arquivo, armazenando de forma temporária essa informação. A partir daí, o programa faz a localização de todos os estados e de todas as transições e, também, armazena essas informações temporariamente. Após a localização dos estados é feita uma lista de todos os estados presentes na máquina, através da localização das transições é feita uma lista das transições, das condições de entrada e das saídas existentes. Quando o aluno seleciona gerar o arquivo de entrada

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)

para o programa TABELA, o programa exibe uma mensagem perguntando qual elemento de memória (flip-flop) será utilizado, ou seja, o tipo de elemento de memória é uma entrada manual. Já na tradução para VHDL correspondente está mensagem não aparece.

Na “Figura 4” é apresentado o diagrama funcional do programa SF<sup>2</sup>Tab<sup>+</sup>VHDL com todos os passos necessários para realizar a tradução.

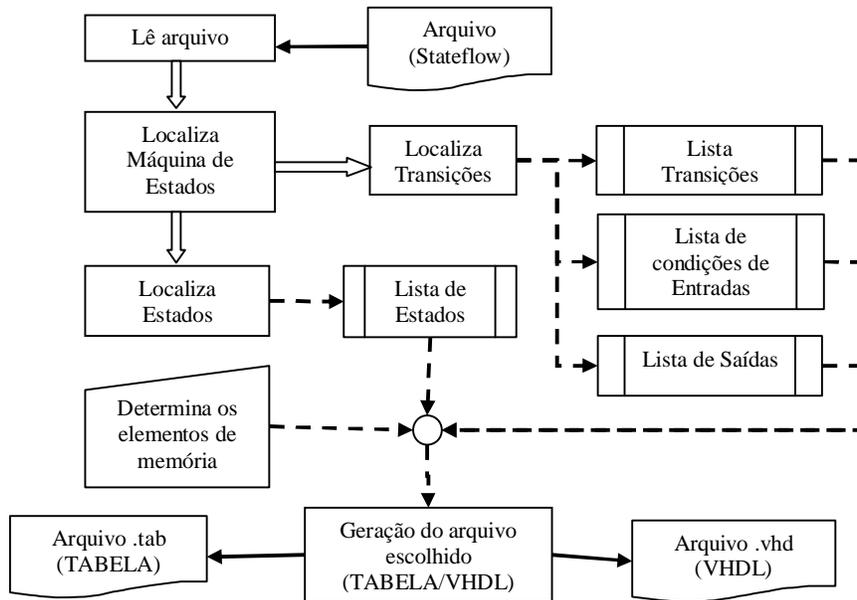


Figura 4. Diagrama funcional do programa SF<sup>2</sup>Tab<sup>+</sup>VHDL.

#### 4 AVALIAÇÃO DA METODOLOGIA

Para validar essa metodologia de ensino foi utilizado um circuito simples capaz de detectar uma sequência de três (0) consecutivos sem sobreposição, cujo diagrama de transição de estados está apresentado na “Figura 1”. O estado A é considerado o estado inicial e os estados A, B e C formam o ciclo que detecta a sequência de três (0) consecutivos. É possível notar que quando o detector está no estado B e recebe (1) na entrada retorna para o estado A, o mesmo acontece para o estado C. A “Figura 5” ilustra o diagrama de transição de estado desse circuito modelado em ambiente Stateflow<sup>®</sup>.

Como pode ser observado na “Figura 5”, os retângulos com cantos arredondados representam os estados, as setas representam a transição de um estado para outro conforme a condição descrita em seu rótulo, assim, a letra *u* representa a variável de entrada no instante de tempo *t*, e a variável *z* representa a saída nesta determinada transição.

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)

A alocação de estado escolhida está apresentada na “Tabela 1”. É importante ressaltar que diferentes alocações podem dar origem a diferentes circuitos, ou seja, um circuito contendo um número maior ou menor de operadores lógicos.

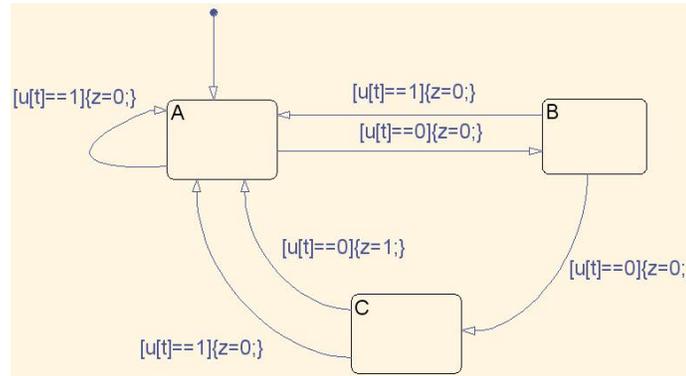


Figura 5. Diagrama de transição de estado do circuito detector de três (0) consecutivos.

A partir da máquina de estados modelada e simulada em ambiente Stateflow<sup>®</sup> é utilizado o programa SF<sup>2</sup>Tab<sup>+</sup>VHDL para gerar a tabela de transição de estados que servirá como entrada para o TABELA, de onde será feita a minimização das funções booleanas do circuito e, também, a descrição em VHDL comportamental.

O TABELA gera uma descrição detalhada da minimização obtida apresentando o custo de cada função (forma de soma de produtos), os estados irrelevantes, os implicantes primos. Na “Figura 6” ilustra-se a descrição obtida pelo TABELA.

```

FUNCAO D1
=====
MI NTERMOS : 1;
DON' T CARE STATES : 7; 3;
IMPLI CANTES PRIMOS ESSENCIAI S :
ESSENCIAL: 1 REDUNDANCIA: 2 -> 0X1
CUSTO FINAL DE D1 = 2

FUNCAO D0
=====
MI NTERMOS : 0;
DON' T CARE STATES : 7; 3;
IMPLI CANTES PRIMOS ESSENCIAI S :
ESSENCIAL: 0 REDUNDANCIA: 0 -> 000
CUSTO FINAL DE D0 = 3

FUNCAO Z0
=====
MI NTERMOS : 2;
DON' T CARE STATES : 7; 3;
IMPLI CANTES PRIMOS ESSENCIAI S :
ESSENCIAL: 2 REDUNDANCIA: 1 -> 01X
CUSTO FINAL DE Z0 = 2

CUSTO TOTAL DAS 3 FUNCOES = 7

```

Figura 6. Diagrama de transição de estados do circuito detector de três (0) consecutivos.

A partir dessa descrição é utilizada uma terceira ferramenta chamada TAB2VHDL. Mais detalhes sobre o TAB2VHDL podem ser encontrados em (TANCREDO, 2002).

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)

Essa ferramenta é capaz de ler a descrição da minimização gerada pelo TABELA e traduzir para o código VHDL estrutural correspondente. O código VHDL comportamental e estrutural referente ao circuito detector de três (0) consecutivos estão apresentados na “Figura 7 - (a) e (b)”, respectivamente.

<pre> ENTITY Mealy_De IS   PORT(     ent : IN INTEGER RANGE 0 TO 1;     sai : OUT INTEGER RANGE 0 TO 1;     clk : IN BIT); END Mealy_De;  ARCHITECTURE Ipsdd OF Mealy_De IS   TYPE tipo_est IS (S0,S1,S2);   SIGNAL est_atual , prox_est : tipo_est; BEGIN   PROCESS (ent, est_atual)   BEGIN     CASE est_atual IS       WHEN S0 =&gt;         IF (ent = 0) THEN           sai &lt;= 0;           prox_est &lt;= S1;         ELSE           sai &lt;= 0;           prox_est &lt;= S0;         END IF;       ...       ...     END CASE;   END PROCESS;    PROCESS   BEGIN     WAIT UNTIL clk'EVENT AND clk = '1';     est_atual &lt;= prox_est;   END PROCESS;  END Ipsdd; </pre>	<pre> ENTITY CASO IS   PORT(     CLK, CLR : IN BIT;     XO : IN BIT;     Q0, Q1 : OUT BIT;     ZO : OUT BIT ); END CASO;  ARCHITECTURE RTL OF CASO IS   SIGNAL VEO, VE1: BIT;   SIGNAL D1, D0 : BIT; BEGIN   PROCESS(CLK, CLR)   BEGIN     IF CLR = '0' THEN       VEO &lt;= '0';     ELSIF CLK'EVENT and CLK = '1' THEN       VEO &lt;= D0;     END IF;     Q0 &lt;= VEO;   END PROCESS;    PROCESS(CLK, CLR)   BEGIN     IF CLR = '0' THEN       VE1 &lt;= '0';     ELSIF CLK'EVENT and CLK = '1' THEN       VE1 &lt;= D1;     END IF;     Q1 &lt;= VE1;   END PROCESS;    D1 &lt;= ( NOT(XO) AND (VE0));   D0 &lt;= ( NOT(XO) AND NOT(VE1) AND NOT(VE0));   ZO &lt;= ( NOT(XO) AND (VE1));  END RTL; </pre>
--	--

(a)

(b)

Figura 7. Código VHDL gerado pelos programas SF<sup>2</sup>Tab<sup>+</sup>VHDL e TAB2VHDL, respectivamente.

Tabela 1. Alocação de estados para o circuito detector de três (0) consecutivos.

Estado	Q1	Q0
<b>A</b>	0	0
<b>B</b>	0	1
<b>C</b>	1	0

Tabela 2. Resultado da síntese realizada pelo Quartus II para o VHDL comportamental e estrutural.

	Comportamental	Estrutural	Total
<b>Elementos lógicos</b>	4 (0,021 %)	3 (0,015 %)	18.752
<b>Funções combinacionais</b>	4 (00,21 %)	3 (0,015 %)	18.752
<b>Registadores</b>	3 (0,015 %)	2 (0,010 %)	18.752
<b>Pinos</b>	3 (0,95 %)	6 (1,9 %)	315

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)



Assim, têm-se o código VHDL em dois âmbitos: comportamental e estrutural, descrevendo o mesmo circuito. Com isso é possível fazer um comparativo entre as duas descrições através de uma análise da síntese gerada pelo Quartus II. A “Tabela 2” mostra o resultado da síntese realizada para o VHDL comportamental e estrutural.

É possível observar a diferença de um circuito previamente otimizado (a descrição estrutural), que utiliza uma quantidade menor de elementos lógicos do FPGA. É importante ressaltar, também, que as ferramentas de síntese disponíveis comercialmente têm dificuldades em sintetizar descrições VHDL em níveis altos de abstração, já que a linguagem VHDL não foi inicialmente proposta para síntese de hardware e sim para descrevê-lo. Portanto, quanto mais baixo for o nível de abstração, como em uma descrição estrutural, melhor será o resultado da síntese gerada.

## 5 CONCLUSÃO

Nesse trabalho foi proposta uma nova metodologia de ensino de circuitos lógicos, onde se desenvolveu uma nova ferramenta denominada SF<sup>2</sup>Tab<sup>+</sup>VHDL. Essa ferramenta é capaz de traduzir máquinas de estados finitos descritas pelo modelo de Mealy, modeladas e simuladas em ambiente Stateflow<sup>®</sup> para uma tabela de transição de estados e para a descrição em código VHDL comportamental. Assim, é possível criar uma interação com outras ferramentas já desenvolvidas possibilitando a realização de projetos de máquinas sequenciais, de maneira prática e rápida, e um comparativo entre descrições diferentes de um mesmo circuito. Isso possibilita ao aluno uma visão do desenvolvimento de projetos de circuitos lógicos e de como funciona a sua implementação em FPGA, apontando as vantagens e desvantagens de diferentes métodos de análise de um mesmo projeto.

A ferramenta desenvolvida mostrou-se bastante eficiente na tradução dos modelos em ambiente Stateflow<sup>®</sup>, tanto na geração do código VHDL como na tabela de transição de estados, ambos a um custo computacional baixo. Pretende-se ainda, estender esta metodologia de ensino acrescentando mais funcionalidades ao SF<sup>2</sup>Tab<sup>+</sup>VHDL. Incluindo, também, a tradução de circuitos descritos pelo modelo de Moore e para a linguagem Verilog, que possibilitará a comparação entre linguagens diferentes e a síntese realiza em cada uma delas.

## AGRADECIMENTOS

Os autores agradecem ao programa de Pós Graduação em Engenharia Elétrica da Faculdade de Engenharia de Ilha Solteira - UNESP, à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES e à Fundação para o Desenvolvimento da UNESP - FUNDUNESP.

## REFERÊNCIAS BIBLIOGRÁFICAS

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)



COMER, David J.. **Digital Logic and State Machine Design**. 3. ed. New York, USA: Oxford University Press, 1995. 573 p.

DIAS, Giorjety Licorini. **Ferramentas para Integração de Redes de Petri e VHDL na Síntese de Sistemas Digitais**. 2007. 179 f. Dissertação (Defesa de Dissertação de Mestrado) - Curso de Engenharia Elétrica, Universidade Estadual Paulista, Ilha Solteira, 2007.

D'AMORE, Roberto. **VHDL. Descrição e Síntese de Circuitos Digitais**. Rio de Janeiro, RJ: LTC, 2005. 259 p.

KARRIS, Steven T.. **Introduction to Stateflow with applications**. USA: Orchard Publications, 2007. 366 p.

SILVA, Alexandre César Rodrigues Da. **Contribuição à minimização e simulação de circuitos lógicos**. 1989. 138 f. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, 1989.

TANCREDO, Leandro de Oliveira. **TAB2VHDL: Um ambiente de síntese lógica para máquinas de estados finitos**. 2002. 130 f. Dissertação (Defesa de Dissertação de Mestrado) - Curso de Engenharia Elétrica, Universidade Estadual Paulista, Ilha Solteira, 2002.

## ENVIRONMENT DIDACTIC FOR TEACHING OF FINITE STATE MACHINE

**Abstract:** *In this paper is proposed a methodology for teaching of finite state machine. For it, was developed a tool called SF<sup>2</sup>Tab<sup>+</sup>VHDL, capable of translating finite state machines described by Mealy model, modeled in environment Stateflow<sup>®</sup>, to describe into VHDL language and to the correspondent table of transition states. With this translation is possible the interaction with other tools already developed, enabling a comparative analysis between different forms of analysis of a same circuit and different levels of abstraction of the language VHDL. Therefore, it is obtained a methodology for teaching of design of logic circuits, in particular sequential machines, creating an environment for logic synthesis highly didactic that can be used for didactics.*

**Key-words:** Synthesis, VHDL, finite state machine

**Secretaria Executiva: Factos Eventos.**

Rua Ernesto de Paula Santos 1368, salas 603/604. Boa Viagem Recife - PE CEP: 51021-330

PABX:(81) 3463 0871

E-mail: [cobenge2009@factos.com.br](mailto:cobenge2009@factos.com.br)