FERRAMENTA PARA A CARACTERIZAÇÃO TEMPORAL DE PORTAS LÓGICAS CMOS

Ingrid Cardoso Machado
Acadêmica do curso de Engenharia da Computação – FURG
ingridmachado@furg.br
Paulo Butzen, Cristina Meinhardt
Professores do Centro de Ciências Computacionais – FURG
paulobutzen@furg.br, cristinameinhardt@furg.br

Resumo. *Características elétricas de circuitos integrados são obtidas através da caracterização elétrica dos mesmos. Estas características são essenciais para os projetistas de hardware conhecerem o comportamento dos circuitos desde as etapas iniciais do fluxo de projeto. Dentro da caracterização elétrica existe a caracterização temporal, responsável por retorna dados relativos aos atrasos do circuito. Este trabalho apresenta o desenvolvimento de uma ferramenta para a caracterização temporal automática de portas lógicas CMOS, de forma a acelerar a obtenção das características elétricas de circuitos integrados.*

Palavras-chave: *Caracterização temporal. Circuitos integrados. Portas lógicas.*

1. INTRODUÇÃO

Os sistemas computacionais estão cada vez mais presentes no cotidiano das pessoas. Um sistema computacional contém circuitos integrados e o circuito integrado é usualmente projetado a partir de portas lógicas na tecnologia CMOS. O projeto de circuitos integrados (CI) visa otimizar três principais fatores: área, potência e desempenho. Por isso, no fluxo de projeto de CIs é necessário conhecer as características de atraso e potência dos componentes, ou seja, caracterizar previamente as portas lógicas CMOS. A caracterização elétrica permite obter os dados de atraso e potência do sistema computacional nas etapas iniciais de projeto. (RABAEY et al., 2003)

O fluxo de projeto mais utilizado para a construção de CIs é o *Standard Cell*. Em determinado momento deste fluxo, o circuito é descrito como um conjunto de portas lógicas que implementam uma função. Fazendo uma análise da função, é possível extrair dados que são usados para simulação elétrica. Dependendo da função e do número de entradas, determinar os dados necessários para poder simular eletricamente a porta lógica pode se tornar uma tarefa complexa, o que torna o processo suscetível a erros. (WESTE et al., 2010)

Este trabalho propõe uma ferramenta para a caracterização elétrica automática de portas lógicas CMOS, através da determinação dos dados necessários para realizar a simulação elétrica. A simulação elétrica é realizada com o auxílio do simulador elétrico NGSPICE (2013).

Na próxima Seção são apresentados conceitos importantes a respeito da caracterização temporal. Na Seção 3 é descrito o processo de desenvolvimento da ferramenta. A seção 4 apresenta as conclusões a respeito do desenvolvimento da ferramenta e trabalhos futuros.

2. CARACTERIZAÇÃO TEMPORAL

A caracterização temporal fornece os dados referentes aos valores de desempenho das portas lógicas. O desempenho de CIs pode ser medido através dos seus atrasos. Os

atrasos podem ser de dois tipos: atrasos de propagação e atrasos de transição, como ilustrados na Fig. 1 (PEDRONI, 2010).

Os atrasos de propagação são medidos a partir de 50% de uma transição de uma onda de entrada e até 50% da transição correspondente da onda de saída. Atrasos de propagação indicam o tempo que a saída demora para mudar de estado após uma transição na entrada. Quando a saída muda do sinal lógico "1" para o sinal lógico "0" o atraso é denominado tempo de propagação *high-to-low (tphl)* e quando a saída muda do sinal lógico "0" para o sinal lógico "1" o atraso é denominado tempo de propagação *low-to-high (tplh)*.

Os atrasos de transição são medidos somente em relação ao sinal de saída e indicam quanto tempo a saída leva para mudar de nível lógico. São medidos em 10% e 90% da onda de saída. Quando é uma mudança de "1" para "0" o atraso medido é denominado tempo de descida (*tfall)* e quando a mudança é de "0" para "1" o atraso medido é denominado tempo de subida (*trise)*. (RABAEY et al., 2003)

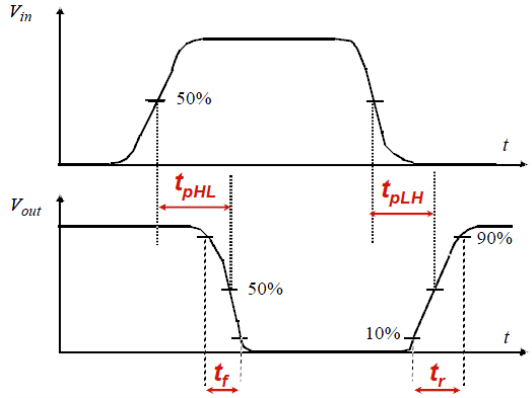

Figura 1. Atrasos de propagação e de transição.

A ferramenta tem como entrada as condições de contorno. As condições de contorno definem parâmetros que serão usados pelo simulador elétrico para realizar a simulação e fornecer dados mais realistas. Dentre as condições de contorno necessárias encontram-se:

- *Netlist*: arquivo com a descrição dos transistores que implementam a porta lógica;
- *Supply*: valor da fonte de tensão que alimentará o circuito;
- Tecnologia: arquivo referente ao modelo dos transistores;
- Capacitor de saída: capacitor que é ligado à saída da porta e ao *ground* para emular uma carga na saída da porta;
- *Inslope*: tempo para uma onda trocar de estado lógico;
- *Transition step*: tempo entre transição de estados das ondas de entrada;
- *Simulation step*: passo que será utilizado pelo simulador para rodar a simulação.

Com estas informações é possível gerar todos os dados necessários para realizar a simulação completa. Esses dados incluem os comandos para gerar as formas de onda de entrada e comandos para medida de atrasos.

2. DESENVOLVIMENTO DA FERRAMENTA

A ferramenta foi desenvolvida utilizando a linguagem de programação C++ e o *framework* QT DIGIA (2013) para o desenvolvimento da interface gráfica. Quando usada no modo terminal é permitido o uso de *scripts* para a geração de diversas simulações em uma única chamada. Ao executar no modo de interface gráfica é necessário inserir as condições de contorno em cada campo da *interface* gráfica da ferramenta, ilustrada na Fig.2.

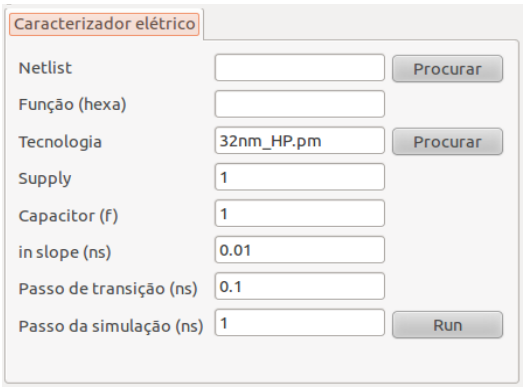

Figura 2. Interface gráfica da ferramenta

A Fig.3 ilustra o algoritmo principal da ferramenta, ou seja, os passos necessários para a sua execução, com a ordem de execução de cada rotina utilizada.

Algoritmo 1: Caraterização Temporal

1: M = Gera_Arcos(k, O)
2: P_A = Gera_Formas_de_Onda(M)
3: I_A = Gera_Sinais_das_Fontes(P_A)
4: $Measure_A$ = Gera_Medidas (I_A, VR_A, VF_A)
5: Resultados = Simulação (Gera_Arquivos(I_A, $Medidas_A$, netlist))

Figura 3. Algoritmo para a caracterização temporal

O primeiro passo obtém os arcos, que são encontrados através da análise da tabela verdade da função *(k)* lógica simulada. Para esta ferramenta, a função lógica é descrita pela saída *(O)* em formato hexadecimal. O conjunto de todos os arcos da função é denominado S_A e é usado para preencher uma matriz de adjacência *M* de tamanho (2^K-1) x (2^K-1), ou seja, com uma linha e uma coluna para cada combinação de entrada. Os arcos são inseridos na matriz como valores booleanos. Células preenchidas com o valor 1 indicam a existência de arcos e células preenchidas com o valor 0 indicam que transições entre a combinação das entradas não causam mudanças no estado da saída.

A Figura 4 (a) mostra a tabela verdade para a porta NAND2. A função é representada como $E_{(h)}$ em hexadecimal na Fig.4 (b). O conjunto dos arcos da função é mostrado na Fig.4 (c) e a matriz de adjacência montada com esse conjunto é mostrada na Fig.4 (d).

A função **Gerar_Arcos***(k, O)*, mostrada na Fig.5, descreve a obtenção dos arcos e a construção da matriz de adjacência. Após a construção da matriz de adjacência, a função **Gerar_Formas_de_Onda***(M)*, mostrada na Fig. 6, é chamada. Ela é um algoritmo guloso que, através da matriz *(M)*, encontra uma combinação dos arcos encontrados. Essa combinação é usada para definir as formas de onda das entradas da função, denominada caminho *(P_A)*. Esse caminho define o comportamento da saída durante a simulação. As formas de onda são definidas pela função. Algoritmos de busca e largura e busca em profundidade foram avaliados para gerar o melhor caminho, para otimizar o número de passos na simulação (tempo de execução). Devido à falha em encontrar soluções razoáveis, fez-se o uso de um algoritmo guloso. O algoritmo guloso atualmente utilizado apresenta bons resultados e pequenas modificações estão sendo exploradas a fim de minimizar o tempo de execução (CORMEN et al., 2001).

Com o caminho gerado pela função anterior são gerados os comandos para a medida dos atrasos de propagação e de transição. Esses comandos são passados para o simulador elétrico, indicando em quais pontos da simulação as medidas deverão ser realizadas.

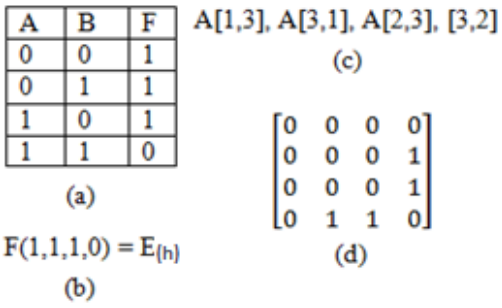

Figura 4. Tabela verdade (a), conversão da saída para hexadecimal (b), arcos da função (c) e matriz de adjacência (d)

Algoritmo 2: Gera_Arcos(k, O)

1: Initialization
2: S_A = Ø, i = 0, j = 0
3: **for each** i ≤ (2^k-1)
4: **for each** j ≤ (2^k-1)
5: **if** O[i] ≠O[j] **then**
6: S_A = A[i,j]
7: M[i,j] = 1
8: **else**
9: M[i,j] = 0
10: **return** M

Figura 5. Algoritmo para a geração dos arcos e da matriz de adjacências

Algoritmo 3: Gera_Formas_de_Onda(*M*)

```
1:  Initialization
2:        P_A = Ø, i = 0, j = 0
3:  for each i  ≤ ( 2^k-1)
4:     for  each j ≤ ( 2^k-1)
5:        if M_ij = 1 then
6:           P_A = (i, j)
7:           M[i,j] = 0
8:           i = j
9:  return P_A
```

Figura 6. Algoritmo para a geração das formas de onda

Reunindo as condições de contorno, as formas de onda de entrada e os comandos para medida dos atrasos, a ferramenta monta o arquivo de simulação que será utilizado pelo simulador elétrico. Isto é feito pela função **Gera_Arquivos** *(I, E, Netlist)*, ilustrada na Fig. 7. Após a simulação, o arquivo de simulação e o arquivo com os atrasos gerados são armazenados, devidamente identificados.

Algoritmo 4: Gera_Arquivos(*I, E, netlist*)

```
1:  Initialization
2:        k = 0, i = 0
3:  for each i  ≤ ( 2^k-1)
4:     create F[i];
5:     for  each E in (E[0] ... E[n])
6:        F[i] = E[j] + I[i,j]
7:     i++
8:  return F
```

Figura 7. Algoritmo para a geração do arquivo de simulação completo

Como exemplo de utilização da ferramenta são mostrados resultados para uma porta NAND de duas entradas (NAND2). A Tab. 1 exibe os resultados dos valores de atraso de uma simulação para a porta lógica NAND2, com o valor de *supply* = 1V, capacitor = 1f, *inslope* = 10ps, na tecnologia preditiva de 32nm (CAO et al, 2000).

Tabela 1. Atrasos de propagação e transição para a porta NAND2

Propagação (ps)		Transição (ps)	
tphl_b_a1	11,63	tfall_b_a1	20,37
tplh_b_a1	9,69	trise_b_a1	19,82
tphl_a_b1	10,36	tfall_a_b1	20,73
tplh_a_b1	8,83	trise_a_b1	17,64

3. CONCLUSÕES

A caracterização de portas lógicas é uma tarefa importante no fluxo de projeto. Este trabalho descreveu o desenvolvimento de uma ferramenta para caracterização temporal automática de portas lógicas. A automação deste processo permite reduzir o tempo para a obtenção de dados da caracterização elétrica. Trabalhos futuros incluem a otimização da ferramenta para reduzir o tempo de simulação.

4. REFERÊNCIAS

RABAEY, Jan M. et al. Digital Integrated Circuits. Prentice-Hall, 2003. Ed. 2.

QT DIGIA. **Qt Creator IDE**. Disponível em: <http://qt.digia.com>. Acesso em jul. 2013.

NGSPICE - Disponível em: <http://ngspice.sourceforge.net>. Acesso em jul. 2013

CORMEN, Thomas H. et al. Introduction to Algorithms. MIT Press, 2001.

CAO, Y., et al. Paradigm of Preditive MOSFET and Interconnect Modeling for Early Circuit Design. **Proc. IEEE Custom Integrated Circuits Conference**. P. 201-204, 2000.

WESTE, N. H. E., HARRIS, D. M., CMOS VLSI Design: a Circuit and Systems Perspective. Ed. 4, 2010.

PEDRONI, V. A., Eletrônica Digital Moderna e VHDL. Ed. 1, 2010.