

Projeto SisRAS2:
SIStemas Computacionais com Capacidade de Confiabilidade,
Disponibilidade e Utilidade (RAS) 2
(Processo CNPq nº 182980/2013-8)

Relatório de Atividades de Bolsista SisRas2
(Junho de 2013 a Janeiro de 2014)

Nome do Bolsista:
Chrystian de Sousa Guth
Universidade Federal de Santa Catarina

Prof. Dr. Ricardo Augusto da Luz Reis
Universidade Federal do Rio Grande do Sul
Coordenador do Projeto

Florianópolis

Janeiro de 2014

1 Relatório de Atividades do Bolsista

1.1 Identificação

- **Nome:** Chrystian de Sousa Guth;
- **Local de Trabalho:** Universidade Federal de Santa Catarina;
- **Título do Plano de Trabalho:** Avaliação do Impacto do Atraso das Interconexões na Análise de *Timing* no Contexto de uma Ferramenta de *Gate Sizing*;
- **Tipo de Bolsa:** ITI-A;
- **Número do Processo da Bolsa:** 182980/2013-8;
- **Período:** Julho de 2013 - Janeiro de 2014;
- **Orientador:** José Luís Almada Güntzel (INE/UFSC);
- **Coordenador do Projeto:** Ricardo Augusto da Luz Reis (II/UFRGS).

1.2 Resumo

Este documento relata as atividades realizadas pelo bolsista Chrystian de Sousa Guth no período de Junho de 2013 a Janeiro de 2014, no contexto de bolsa ITI-A associada ao “Projeto SisRAS2 - Sistemas Computacionais com Capacidade de Confiabilidade, Disponibilidade e Utilidade (RAS) 2”.

Conforme previsto no plano de trabalho da bolsa, entre Junho de 2013 e Janeiro de 2014 o bolsista realizou um estudo para compreensão das técnicas de análise de *timing* estática (*STA: Static Timing Analysis*) aplicadas no contexto de uma ferramenta de otimização para fluxo industrial.

Nos primeiros meses (entre Junho e Setembro de 2013), o aluno desenvolveu uma ferramenta de STA utilizando a infraestrutura disponibilizada pela competição de *sizing* discreto do ISPD 2013 ([OZDAL et al., 2013](#)), que levava em consideração o atraso das interconexões e a degradação do *slew* através destas. Posteriormente, entre Outubro e Novembro de 2013, foram realizados experimentos a fim de validar a ferramenta, a qual foi documentada na forma de uma monografia de conclusão de curso, apresentada no presente trabalho na [subseção 1.6.6](#). Finalmente, entre Dezembro de 2013 e Janeiro de 2014, a infraestrutura implementada até então começou a ser incorporada à uma técnica de otimização para fluxo *Standard Cell* conhecida como *gate sizing*.

Palavras-chave: automação de projeto eletrônico, biblioteca standard cell, análise de timing estática, complementary metal-oxide semiconductor, gate sizing.

1.3 Introdução

O crescimento da complexidade dos circuitos digitais contemporâneos¹ e a necessidade de um *time-to-market* (tempo de entrega ao mercado) curto faz com que o projeto de tais circuitos adote o fluxo *standard cell*.

Os projetos de circuitos digitais no fluxo *standard cell* são realizados visando, além das funcionalidades requisitadas, a operação em uma frequência especificada. Por isso, diversas otimizações são efetuadas ao longo do fluxo, para que tais funcionalidades consigam ser realizadas na frequência definida.

Análise de *timing* estática, ou *static timing analysis* (GUNTZEL, 2000) (BHASKER; CHADHA, 2009), é uma das técnicas utilizadas para se estimar o atraso crítico de circuitos digitais. A análise de *timing* é chamada de estática quando ela não realiza simulação e portanto, independe de estímulos de entrada, considerando apenas a topologia do circuito. É um processo completo e exaustivo (BHASKER; CHADHA, 2009) que verifica as mais diversas informações de *timing* em um circuito, como os *delays*, *slews*, *slacks* (folgas), *required times* (tempos requeridos) e diversas violações de restrições de projeto.

Uma ferramenta capaz de realizar a análise de *timing*, considerando os atrasos das interconexões, foi desenvolvida, implementando um modelo de grafo otimizado, obtendo tempos de execução cerca de 17 vezes menor que os tempos gastos em uma ferramenta industrial.

1.4 Materiais e Métodos

Nas atividades realizadas foi disponibilizado para uso do bolsista um computador tipo PC com sistema operacional Linux e outros recursos disponíveis no Laboratório de Computação Embarcada (ECL) da UFSC (ECL, 2014), como impressora *laser*, rede de dados etc. Além dos recursos materiais, também foi utilizada a ferramenta de EDA (*Electronic Design Automation*) Synopsys ®*PrimeTime*™ (SYNOPSYS, 2012), para validação da técnica de STA implementada.

¹ Um processador para *desktop* desenvolvido no ano de 2008 tem cerca de 731 milhões de transistores, excluindo a área de memória (INTEL, 2008).

1.5 Revisão Bibliográfica

Com o objetivo de avaliar uma técnica para cálculo das características temporais das interconexões na análise de *timing*, o bolsista realizou uma pesquisa em algumas técnicas e nos manuais do *PrimeTime*.

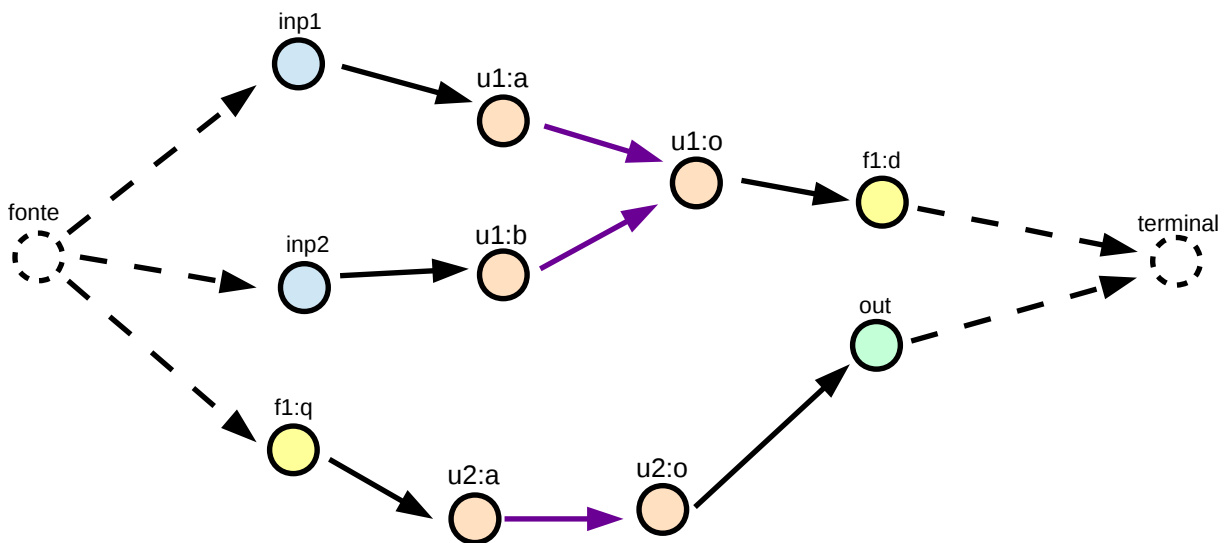
A técnica para cálculo do atraso das características temporais das interconexões implementada é baseada no trabalho (PURI; KUNG; DRUMM, 2002) que por sua vez, ataca as principais desvantagens de se utilizar a abordagem apresentada em (KASHYAP; ALPERT; DEVGAN, 2000).

1.6 Experimentos Desenvolvidos, Resultados e Discussões

1.6.1 Definição do Modelo de Grafo do Circuito

Para que a análise de *timing* seja realizada, é necessário primeiramente, definir um modelo de grafo para representar o circuito. O modelo de grafo escolhido representa os *Timing Points* no conjunto de vértices e os *Timing Arcs* e interconexões no conjunto das arestas (Figura 1). Também são criados vértices especiais, chamados *fonte* e *terminal*. Estes nodos são inseridos no início e no fim do grafo, para que o mesmo possua apenas um ponto de “entrada”, e um ponto de “saída”.

Figura 1 – Grafo de *timing* com representação dos *timing points*, *timing arcs* e interconexões.



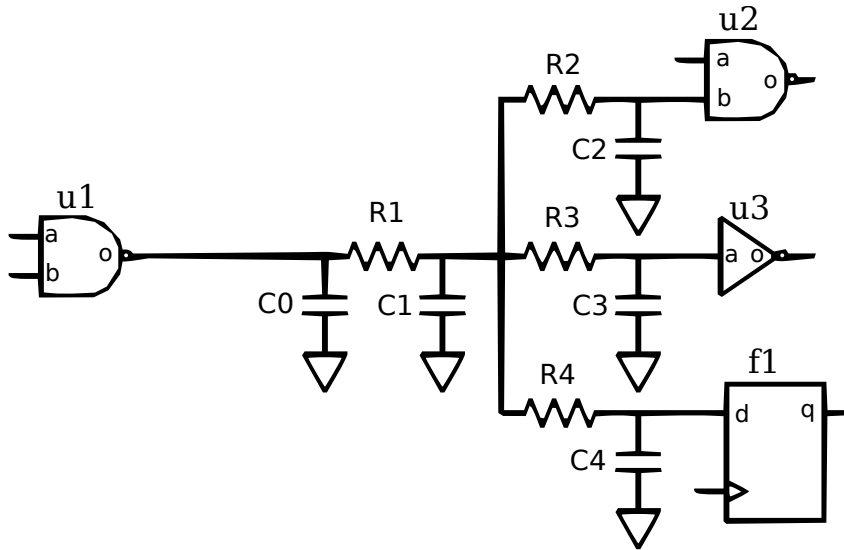
Fonte: o autor.

1.6.2 Representação das Interconexões

As interconexões mais complexas devem ser representadas levando em consideração suas características resistivas e capacitivas. Essas então, são modeladas como árvores RC (Figura 2), que conforme Rabaey, Chandrakasan e Nikolic (2008), têm as seguintes características:

- A rede tem apenas um nodo de entrada, chamado de **fonte** (*source*);
- Todos os capacitores são entre um nodo e o terra;
- A rede não possui *loops* resistivos, por isso é chamada de Árvore.

Figura 2 – Um circuito composto por três portas lógicas ($u1$, $u2$ e $u3$), uma célula sequencial ($f1$) e uma interconexão em forma de árvore RC, que liga a saída de $u1$ às entradas de $u2$, $u3$ e $f1$. A interconexão em questão possui 5 capacitores ($C0$ (*fonte*), $C1$, $C2$, $C3$ e $C4$) e 4 resistores ($R1$, $R2$, $R3$ e $R4$).



Fonte: o autor.

1.6.3 Algoritmo para o Cálculo das Informações Temporais das Interconexões

Kashyap, Alpert e Devgan (2000) propuseram uma técnica para calcular o atraso da interconexão levando em conta o efeito de *resistive shielding*. Com a mesma complexidade da técnica de Elmore, a técnica proposta para cálculo de atraso em uma árvore RC calcula também o valor de capacitância efetiva. Porém, Kashyap, Alpert e Devgan (2000) não consideravam o *driver* da interconexão como sendo uma porta lógica CMOS. Como consequência, sua aproximação para o *slew* na entrada da árvore RC era imprecisa. Como o cálculo da capacitância efetiva de uma árvore depende do *slew* que incide nesta, e o *slew* depende da capacitância vista pelo *driver*, Puri, Kung e Drumm (2002) propuseram uma

técnica que leva em consideração o impacto da capacitância no *slew* do driver, e também, do *slew* no cálculo da capacitância efetiva. Esta técnica foi implementada na ferramenta desenvolvida, e foi validada frente a uma ferramenta industrial.

1.6.4 Cálculo do Atraso das Portas Lógicas

Nas bibliotecas *standard cell* atuais, modelos de atrasos não-lineares² são fornecidos para os *timing arcs* das células disponíveis. Esses modelos, que geralmente são obtidos através de simulações em nível elétrico, são armazenados na forma de *lookup tables*, como a da Figura 3. Uma *lookup table* descreve o *delay* ou o *slew* de uma célula em função de dois fatores: o *slew* na entrada do *timing arc* (colunas), e a capacitância de saída (*load*) (linhas).

Utilizando a *lookup table* da Figura 3 para estimar o *delay* de um dos *timing arcs* de uma célula *CMOS* e supondo que o *slew* na entrada deste *timing arc* seja de 8.0, e a capacitância vista na saída seja 0.1, obtém-se que *delay* = 3.49, pois 3.49 é o valor endereçado pelos índices da função (*slew* e *load*). Caso os valores de *slew* ou *load* não existam na tabela, uma interpolação linear é realizada. Da mesma forma, o cálculo do *slew* do *timing arc* é realizado com base na *lookup table* específica para o *slew*.

Figura 3 – Uma *lookup table* para atraso de subida (*rise delay*) de um arco de *timing*. As linhas são endereçadas por *load* (capacitância de saída da porta lógica) e as colunas por *input slew* (*slew* aplicado na entrada do *timing arc*).

```
rise_delay (delay_table) {
  load (0.0, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2) ;
  input_slew (0.5, 3.0, 5.0, 8.0, 14.0, 20.0, 30.0, 50.0) ;
  values (
    1.17, 1.82, 2.26, 2.76, 3.48, 4.04, 4.82, 6.12,
    1.69, 2.34, 2.86, 3.49, 4.41, 5.11, 6.06, 7.58,
    2.21, 2.86, 3.38, 4.12, 5.22, 6.05, 7.16, 8.90,
    3.25, 3.90, 4.42, 5.20, 6.60, 7.67, 9.08, 11.23,
    5.33, 5.98, 6.50, 7.28, 8.84, 10.30, 12.24, 15.14,
    9.50, 10.15, 10.67, 11.45, 13.01, 14.57, 17.15, 21.33,
    17.83, 18.48, 19.00, 19.78, 21.34, 22.90, 25.50, 30.70
  );
}
```

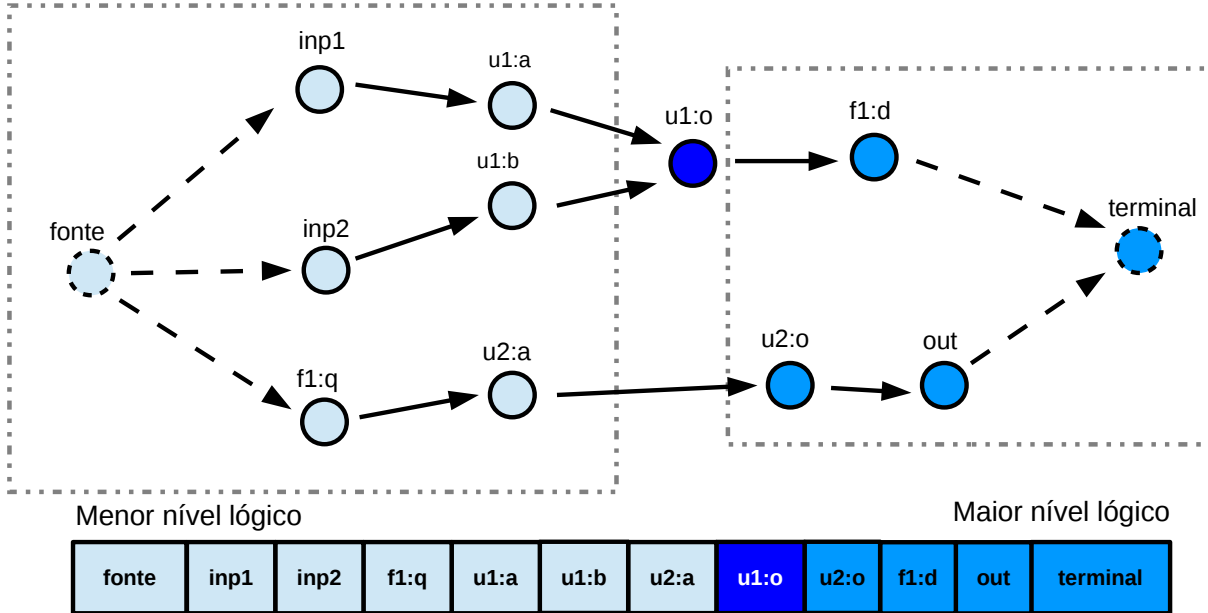
Adaptada de (OZDAL et al., 2013).

1.6.5 Algoritmo para o Cálculo do Atraso do Circuito (Análise de *Timing*)

O algoritmo de STA é baseado nas técnicas conhecidas como PERT/CPM (BHASKER; CHADHA, 2009). Para melhorar o desempenho da técnica, foram adotadas estruturas

² Conhecidos na indústria por *NLDM* (*Non-Linear Delay Model*)

Figura 4 – Na lista ordenada, observando o elemento $u1:o$, os elementos de menor ou de igual nível lógico ($fonte$, $inp1$, $inp2$, $f1:q$, $u1:a$, $u1:b$, $u2:a$) se encontram à esquerda, e os de maior ou igual ($u2:o$, $f1:d$, out , $terminal$) se encontram à direita.



Fonte: O autor.

de dados mais otimizadas para armazenar o grafo: ao guardar os dados do grafo em listas ordenadas topologicamente (Figura 4), o algoritmo passa a ser apenas uma varredura em cada item da lista, atualizando as informações temporais previamente calculadas. Assim, não é mais necessário o uso da fila da técnica PERT/CPM.

1.6.6 O Trabalho de Conclusão de Curso Gerado

O trabalho desenvolvido gerou a monografia de conclusão de curso intitulada “Análise de *Timing* Estática e a Avaliação do Impacto do Atraso das Interconexões em Circuitos Digitais” e apresentada em Novembro de 2013. A mesma pode ser acessada no *site* de projetos do curso de Ciências da Computação da UFSC (GUTH, 2013).

1.6.7 Validação e Resultados

Para realizar a avaliação das técnicas abordadas neste trabalho, uma ferramenta para análise de *timing* na linguagem de programação C++ foi implementada. A ferramenta realiza a análise de *timing* e considera dois possíveis modelos de interconexão: o modelo da capacitância concentrada e o modelo RC concentrado.

Como parte dos experimentos é realizada comparando as informações calculadas pela ferramenta implementada com as informações reportadas pelo *PrimeTime*, o erro

percentual (EP_t) e o erro médio percentual absoluto ($EMPA$) (Equação 1.1 e Equação 1.2) foram adotados como métricas para estimar a qualidade das informações de *timing* reportadas pela ferramenta implementada neste trabalho.

$$EP_t = \frac{(A_t - P_t)}{A_t} \times 100 \quad (1.1)$$

$$EMPA = \frac{\sum_{t=1}^n |EP_t|}{n} \quad (1.2)$$

O erro percentual é calculado para cada uma das informações comparadas com o *PrimeTime* utilizando a Equação 1.1, sendo que A_t é a informação obtida pelo *PrimeTime* e P_t é a informação calculada pela ferramenta implementada. Tais informações usadas para fim de validação da ferramenta foram:

- ***TNS (Total Negative Slack)***: O somatório de *slack* negativo nas saídas primárias.
- ***Violating POs***: Número de saídas primárias violando a restrição de desempenho mínimo.
- ***Runtime (s)***: Tempo de execução, em segundos, para realizar uma análise de *timing* em um circuito, desconsiderando o tempo constante de inicialização da ferramenta.
- ***Critical Path***: Valor do caminho crítico do circuito.

Os resultados dos experimentos serão apresentados posteriormente por meio de gráficos, tabelas e histogramas.

Este trabalho utilizou como base a infraestrutura disponibilizada pela competição de *gate sizing* discreto do ISPD de 2013, a qual fornece:

- Um conjunto de 8 circuitos da competição do ISPD de 2013:
 1. *usb_phy*: com 511 células combinacionais, 98 células sequenciais, 15 entradas e 19 saídas primárias;
 2. *pci_bridge32*: com 27316 células combinacionais, 3359 células sequenciais, 160 entradas e 201 saídas primárias;
 3. *fft*: com 30297 células combinacionais, 1984 células sequenciais, 1026 entradas e 1026 saídas primárias;
 4. *cordic*: com 40371 células combinacionais, 1230 células sequenciais, 34 entradas e 64 saídas primárias;

5. *des_perf*: com 103842 células combinacionais, 8802 células sequenciais, 234 entradas e 201 saídas primárias;
 6. *edit_dist*: com 125000 células combinacionais, 5661 células sequenciais, 2562 entradas e 12 saídas primárias;
 7. *matrix_mult*: com 30297 células combinacionais, 1984 células sequenciais, 3202 entradas e 1600 saídas primárias;
 8. *netcard*: com 884427 células combinacionais, 97831 células sequenciais, 1836 entradas e 10 saídas primárias.
- Uma biblioteca *standard cell* realista, composta por onze células combinacionais de diversas funções lógicas e um célula sequencial;
 - Uma ferramenta de análise de *timing* estática PrimeTimeTM da empresa [Synopsys \(2012\)](#) ® para comparação de resultados;

Os circuitos são compostos por descrições no formato Verilog, capacitâncias parasitas e resistências descritas no formato IEEE SPEF (*Standard Parasitic Exchange Format*) ([IEEE, 1999](#)), e restrições de *timing* descritas no formato SDC (*Synopsys Design Constraints*).

Os resultados dos experimentos realizados para validar a ferramenta de STA desenvolvida mostram alguns aspectos importantes:

- A desconsideração da degradação do *slew* na análise de *timing* pode acarretar erros de cerca de 20%;
- A técnica para cálculo das informações temporais das interconexões implementada apresentou ser **17,02 vezes mais rápida** que o PrimeTime, obtendo resultados para *TNS* (*Total Negative Slack*) e *critical path* que subestimam em cerca de 8,85% e 4,48% respectivamente, os reportados pela ferramenta industrial;
- A relação C_{eff}/C_{total} nos circuitos da competição de *sizing* do ISPD de 2013 mostrou-se na média, próxima de 1. A partir dessa informação, o modelo de capacitância concentrada para calcular os atrasos dos *drivers*, juntamente com a técnica de Elmore com C_{total} e a técnica de [Puri, Kung e Drumm \(2002\)](#) para degradação do *slew* foi avaliada, apresentando estimativas pessimistas em 10,76% para *TNS* nos circuitos testados (exceto o *netcard*) e 6,48% para *critical path*, sendo que o tempo de execução é cerca de **3 vezes menor** que o da técnica considerando a C_{eff} .

Considerações finais

Durante o período de vigência da bolsa, o bolsista avaliou uma técnica de análise de *timing* estática que considera os atrasos das interconexões e os efeito de degradação do *slew* através destas. Para tal, o bolsista implementou uma ferramenta baseada na infraestrutura do ISPD de 2013 ([OZDAL et al., 2013](#)) com os algoritmos para cálculos dos atrasos devidamente implementados. No fim da vigência da bolsa, foi também iniciada a incorporação do modelo de grafo e cálculos de atraso em uma técnica de otimização para fluxo industrial, conhecida como *gate sizing*, porém com término previsto para metade do primeiro semestre de 2014.

Referências

- BHASKER, J.; CHADHA, R. *Static Timing Analysis for Nanometer Designs: A Practical Approach*. 1. ed. [S.l.]: Springer, 2009. Citado 2 vezes nas páginas 2 e 5.
- ECL. *ECL – Embedded Computing Lab*. 2014. Disponível em: <<http://eclab.paginas.ufsc.br/>>. Acesso em: 29/01/2014. Citado na página 2.
- GUNTZEL, J. L. A. *Functional Timing Analysis of VLSI Circuits Containing Complex Gates*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, RS-Brazil, 2000. Citado na página 2.
- GUTH, C. de S. *Análise de Timing Estática e a Avaliação do Impacto do Atraso das Interconexões em Circuitos Digitais*. 2013. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_1459/Relatorio.pdf.2>. Acesso em: 29/01/2014. Citado na página 6.
- IEEE. Ieee standard for integrated circuit (ic) delay and power calculation system. *IEEE Std 1481-1999*, p. i–390, 1999. Citado na página 8.
- INTEL. *Intel(R) Core(TM) i7-920 Processor Specifications*. 2008. Disponível em: <http://ark.intel.com/products/37147/Intel-Core-i7-920-Processor-8M-Cache-2_66-GHz-4_80-GTs-Intel-QPI>. Acesso em: 26/10/2013. Citado na página 2.
- KASHYAP, C. V.; ALPERT, C. J.; DEVGAN, A. An effective capacitance based delay metric for rc interconnect. In: IEEE PRESS. *Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design*. [S.l.], 2000. p. 229–235. Citado 2 vezes nas páginas 3 e 4.
- OZDAL, M. M. et al. An improved benchmark suite for the ispd-2013 discrete cell sizing contest. In: *Proceedings of ACM International Symposium on Physical Design*. [S.l.: s.n.], 2013. p. 168–170. Citado 3 vezes nas páginas 1, 5 e 9.
- PURI, R.; KUNG, D. S.; DRUMM, A. D. Fast and accurate wire delay estimation for physical synthesis of large asics. In: *Proceedings of the 12th ACM Great Lakes symposium on VLSI*. New York, NY, USA: ACM, 2002. (GLSVLSI '02), p. 30–36. ISBN 1-58113-462-2. Disponível em: <<http://doi.acm.org/10.1145/505306.505314>>. Citado 3 vezes nas páginas 3, 4 e 8.
- RABAEY, J. M.; CHANDRAKASAN, A.; NIKOLIC, B. *Digital Integrated Circuits*. 3rd. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008. ISBN 0132219107, 9780132219105. Citado na página 4.
- SYNOPSYS. *Synopsys PrimeTime User's Manual*. 2012. Disponível em: <<http://www.synopsys.com>>. Acesso em: 01/12/2012. Citado 2 vezes nas páginas 2 e 8.