

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO – CTC**

Chrystian de Sousa Guth

**OTIMIZAÇÃO DE ATRASO PÓS-POSICIONAMENTO
EXPLORANDO RAMOS NÃO-CRÍTICOS DE ÁRVORES
DE STEINER**

Florianópolis

2016

Chrystian de Sousa Guth

**OTIMIZAÇÃO DE ATRASO PÓS-POSICIONAMENTO
EXPLORANDO RAMOS NÃO-CRÍTICOS DE ÁRVORES
DE STEINER**

Dissertação submetida ao Programa de Pós-Graduação em Ciências da Computação – PPGCC para a obtenção do Grau de Mestre em Ciência da Computação.

Orientador: Prof. José Luís Almada Güntzel, Dr.

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

de Sousa Guth, Chrystian

Otimização de Atraso Pós-Posicionamento Explorando Ramos
Não Críticos de Árvores de Steiner / Chrystian de Sousa
Guth ; orientador, José Luís Almada Güntzel - Florianópolis,
SC, 2016.

102 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico. Programa de Pós-Graduação em
Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. síntese física. 3.
posicionamento. 4. timing. 5. árvores de Steiner. I. ,
José Luís Almada Güntzel. II. Universidade Federal de Santa
Catarina. Programa de Pós-Graduação em Ciência da Computação.
III. Título.

Chrystian de Sousa Guth

**OTIMIZAÇÃO DE ATRASO PÓS-POSICIONAMENTO
EXPLORANDO RAMOS STEINER NÃO-CRÍTICOS**

Esta dissertação foi julgada adequada para obtenção do título de mestre e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 18 de dezembro de 2015.

Prof.^a. Carina Friedrich Dorneles, Dr.^a.
Coordenadora do Programa

Banca Examinadora:

Prof. José Luís Almada Güntzel, Dr.
Universidade Federal de Santa Catarina
Orientador

Prof. Marcelo de Oliveira Johann, Dr.
Universidade Federal do Rio Grande do Sul

Gustavo Reis Wilke, Dr.
Synopsys Inc. (R&D Engineer)

Prof. Laércio Lima Pilla, Dr.
Universidade Federal de Santa Catarina

RESUMO

O crescente impacto das interconexões no desempenho dos circuitos aumentou a importância do projeto físico na última década. No contexto das tecnologias contemporâneas, é imprescindível se considerar informações de interconexões nas estimativas de atraso, para que otimizações no projeto físico não invalidem otimizações de desempenho realizadas durante a síntese lógica. Uma das técnicas de otimização utilizadas durante o projeto físico é o posicionamento guiado por atraso (*TDP: timing-driven placement*). Dado um posicionamento inicial do circuito, *TDP* move um número limitado de células com o objetivo de reduzir (ou mesmo corrigir, se possível) as violações de atraso crítico do circuito. O *TDP* pode ser realizado de maneira global ou incremental. Este trabalho propõe e avalia uma técnica de *TDP* incremental que reposiciona um subconjunto de células a fim de otimizar o atraso referente às interconexões mais críticas do circuito tentando, ao mesmo tempo, preservar a qualidade do posicionamento inicial. A técnica modela explicitamente as interconexões com árvores de Steiner, as quais são capazes de capturar informações sobre a topologia do roteamento final. Aplicada em circuitos industriais previamente otimizados, a técnica proposta proporcionou reduções médias de violações de atraso de 34% a 62%, considerando as restrições de deslocamento *short* e *long*, respectivamente.

Palavras-chave: síntese física. posicionamento. *timing*. Árvores de Steiner.

ABSTRACT

The growing impact of interconnections on circuit performance has increased the importance of physical design in the last decade. In the context of the contemporary technologies, it is essential that circuit delay estimates consider interconnect information to avoid that physical synthesis optimizations invalidate upstream optimizations. Timing-driven placement (TDP) is one of the optimization techniques used during physical synthesis. Given an initial circuit placement, TDP moves a limited number of cells targeting at reducing (or even correcting, if possible) the circuit timing violations. TDP can be performed in a global fashion or incrementally. This work proposes and evaluates an incremental TDP technique that moves a subset of cells to optimize the delay of the most critical interconnections in the circuit, while trying to preserve the initial placement quality. The technique explicitly models the interconnections as Steiner trees, which are able to capture information on the interconnection topologies in the final routing. The proposed technique was applied on previously optimized industrial circuits having produced average reductions of 34% and 62% in timing violations, concerning short and long maximum displacement restrictions, respectively.

Keywords: physical synthesis. placement. timing. Steiner trees.

LISTA DE FIGURAS

Figura 1	Passos principais de um fluxo de projeto <i>VLSI</i> . Síntese física destacada à direita. Adaptado de (KAHNG, 2011).	20
Figura 2	Circuito: superblue18 (KIM et al., 2015a) (KIM et al., 2015b). Adaptado pelo autor.	22
Figura 3	Visualização do circuito b19 (KIM; HU; VISWANATHAN, 2014) na ferramenta gráfica desenvolvida. As cores dos retângulos representam os <i>slacks</i> das células do circuito. As linhas pretas são a árvore de relógio estimada pelo algoritmo <i>FLUTE</i> (CHU; WONG, 2008). As linhas vermelhas indicam a topologia do caminho crítico.	25
Figura 4	Leiaute de três inversores com capacidades de corrente de saída diferentes. Fonte: Adaptado de (KAHNG, 2011).	28
Figura 5	Circuito superblue18 (KIM et al., 2015a). Adaptado pelo autor.	29
Figura 6	Passo de legalização para alinhamento e remoção das sobreposições.	29
Figura 7	O losango rosa representa a restrição de deslocamento máximo (<i>maximum displacement</i>) da célula em azul. Os retângulos verdes indicam posições legais, ou seja, posições alinhadas e dentro da área de deslocamento máximo. Os retângulos vermelhos indicam posições ilegais, desalinhadas e/ou além da restrição de deslocamento máximo. Note que a coordenada de referência é a inferior esquerda. Portanto, o retângulo verde que está posicionado sobre o limite da “área de deslocamento máximo” representa uma posição legal.	30
Figura 8	Comprimento <i>Clique</i> : $14.5\mu\text{m}$. Fonte: (KAHNG, 2011)..	31
Figura 9	Comprimento <i>Star</i> : $15\mu\text{m}$. Fonte: (KAHNG, 2011).	32
Figura 10	<i>HPWL</i> : $9\mu\text{m}$. Fonte: (KAHNG, 2011).	32
Figura 11	<i>StWL</i> : $10\mu\text{m}$. Fonte: (KAHNG, 2011).	33
Figura 12	(a) Uma interconexão representada pelo modelo <i>StWL</i> . (b) A mesma interconexão, após redução de <i>HPWL</i> . O <i>StWL</i> é a soma do comprimento dos segmentos da árvore de Steiner, representados pelas linhas pretas. Já o <i>HPWL</i> , corresponde ao comprimento do semiperímetro do envelope, representado pela caixa pontilhada.	33
Figura 13	Grafo de <i>timing</i> para o circuito simple (KIM et al., 2015a).	35
Figura 14	(a) Uma interconexão representada como uma árvore de	

Steiner. (b) Cada segmento da árvore de Steiner é representado utilizando o modelo RC-distribuído. 37

Figura 15 Um circuito sequencial composto de dois *flip-flops*. 39

Figura 16 Configuração que **respeita** a restrição *setup* para o circuito da Figura 15. 39

Figura 17 Configuração que **viola** a restrição *setup* para o circuito da Figura 15. 40

Figura 18 Configuração que **respeita** a restrição *hold* para o circuito da Figura 15. 40

Figura 19 Configuração que **viola** a restrição *hold* para o circuito da Figura 15. 41

Figura 20 Estratégia para adicionar peso às interconexões. O peso é mantido em 0 para as que possuem *HPWL* menor que a restrição c . Para as interconexões com $HPWL > c$, o peso é aumentado até que o seu valor atinja w_{max} . Fonte: (RAJAGOPAL et al., 2003). 50

Figura 21 (a) uma interconexão pertencente ao caminho crítico do circuito; (b) os *slacks* de cada ponto de Steiner da interconexão; (c) os *slacks* após a redução dos segmentos $s_1 \rightarrow j$ e $s_2 \rightarrow l$ 51

Figura 22 Passos do algoritmo de Choi e Bazargan (2003). Fonte: (CHOI; BAZARGAN, 2003, p. 464). Adaptada pelo autor. 53

Figura 23 Parcelas que compõem o impacto da interconexão e_j no atraso do circuito. Fonte: (LUO; NEWMARK; PAN, 2006). 57

Figura 24 Visão geral da técnica. O *slack* na saída primária crítica é melhorado movimentando-se apenas células não críticas. 60

Figura 25 Proporção de capacitância referente a ramos de Steiner não críticos nas interconexões dos caminhos críticos dos circuitos do *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014). .. 61

Figura 26 (a) Uma interconexão que conecta 4 células modelada como uma árvore de Steiner. (b) A redução do ramo associado à célula j é capaz de reduzir a carga capacitiva da interconexão. O potencial de redução deste ramo do comprimento l_j para o comprimento l'_j pode ser estimado utilizando a Equação 4.1. 62

Figura 27 *ZFT* para a célula i . O *ZFT* é a posição que minimiza a soma das forças entre i e cada uma das células que estão conectadas a esta (a, b, c e d). $c(u, v)$ corresponde ao peso (prioridade) da interconexão entre células u e v . Fonte: (KAHNG, 2011). 67

Figura 28 Movimento da célula não crítica l em direção oposta ao *ZFT*. Em rosa está representada a restrição de deslocamento máximo. i, j e k são células críticas no modo *early*. 68

Figura 29 (a) Uma interconexão do caminho crítico do circuito <code>mgc_matrix_mult</code> (KIM; HU; VISWANATHAN, 2014) antes da otimização. (b) A mesma interconexão após a otimização.	75
Figura 30 Reduções percentuais de <i>WNS early</i> na restrição <i>long</i> partindo das soluções obtidas nos experimentos apresentados no Capítulo 4. No circuito <code>leon3mp</code> a técnica não conseguiu sequer selecionar candidatos.	76
Figura 31 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>mgc_edit_dist</code> , no decorrer de 10 iterações.	93
Figura 32 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>mgc_matrix_mult</code> , no decorrer de 10 iterações.	94
Figura 33 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>vga_lcd</code> , no decorrer de 10 iterações.	94
Figura 34 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>b19</code> , no decorrer de 10 iterações.	95
Figura 35 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>leon2</code> , no decorrer de 10 iterações.	95
Figura 36 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>mgc_edit_dist</code> , no decorrer de 10 iterações.	96
Figura 37 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>mgc_matrix_mult</code> , no decorrer de 10 iterações.	97
Figura 38 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>vga_lcd</code> , no decorrer de 10 iterações.	97
Figura 39 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>b19</code> , no decorrer de 10 iterações.	98
Figura 40 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>leon3mp</code> , no decorrer de 10 iterações.	98
Figura 41 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>leon2</code> , no decorrer de 10 iterações.	99
Figura 42 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>netcard</code> , no decorrer de 10 iterações.	99
Figura 43 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>mgc_edit_dist</code> , no decorrer de 10 iterações.	100
Figura 44 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>mgc_matrix_mult</code> , no decorrer de 10 iterações.	101
Figura 45 Valores obtidos para <i>WNS</i> e <i>TNS late</i> no circuito <code>vga_lcd</code> , no decorrer de 10 iterações.	101

Figura 46 Valores obtidos para *WNS* e *TNS late* no circuito **b19**,
no decorrer de 10 iterações. 102

LISTA DE ABREVIATURAS E SIGLAS

TDP	timing-driven placement	21
ITDP	incremental timing-driven placement	21
StWL	Steiner tree wirelength	22
HPWL	half perimeter wirelength	22
ABU	average bin utilization	23
LR	lagrangian relaxation	24
STA	static timing analysis	34
DAG	directed-acyclic graph	34
QP	quadratic programming	50
PO	primary output	59
ZFT	zero-force target	67

LISTA DE SÍMBOLOS

(X_l, Y_b)	coordenada inferior-esquerda dos limites físicos do <i>chip</i> ...	28
(X_r, Y_t)	coordenada superior-direita dos limites físicos do <i>chip</i>	28
d_M	distância Manhattan	30
$G(\mathcal{V}, \mathcal{A})$	grafo de <i>timing</i>	34
\mathcal{V}	conjunto de vértices (pinos) do grafo de <i>timing</i>	34
\mathcal{A}	conjunto de arestas (arcos de <i>timing</i> ou par fonte-destino de uma interconexão) do grafo de <i>timing</i>	34
\mathcal{PO}	união do conjunto dos pinos de dados de elementos sequenciais e dos <i>pads</i> e saída do circuito	34
\mathcal{PI}	conjunto dos <i>pads</i> de entrada	34
\mathcal{A}_c	conjunto dos arcos de <i>timing</i> combinacionais	34
\mathcal{A}_s	conjunto dos arcos de <i>timing</i> sequenciais	34
σ_i	<i>slew</i> de um pino	36
$\delta_{(i,j)}$	atraso de uma aresta do grafo de <i>timing</i>	36
l_i	comprimento de um segmento de interconexão	36
R_i	resistência do <i>driver</i> i de uma interconexão	37
C_n	valor de capacitância de um pino n	37
r	resistência por unidade de comprimento de fio	37
c	capacitância por unidade de comprimento de fio	37
$\hat{\sigma}_j$	<i>slew</i> da resposta ao impulso em um destino j de uma interconexão	37
β_j	segundo momento da resposta ao impulso no destino j de uma interconexão	37
C_j^{out}	capacitância concentrada no pino de saída de um arco de <i>timing</i> (i, j)	38
L	sobrescrito que indica o modo de propagação pessimista de atraso (<i>late</i>)	40
E	sobrescrito que indica o modo de propagação otimista de atraso (<i>early</i>)	40
at_i	tempo de chegada de um pino	40
rt_i	tempo requerido de um pino	40
s_i	<i>slack</i> de um pino	40
\mathcal{C}	conjunto de células de um circuito	46

W_j	largura da célula j	46
H_j	altura da célula j	46
W_{site}	largura do <i>site</i>	46
H_{row}	altura da banda <i>standard cell</i>	46
D_{max}	restrição de deslocamento máximo	46
x_j	componente horizontal da posição da célula j	46
y_j	componente vertical da posição da célula j	46
x_j^0	componente horizontal da posição inicial da célula j	46
y_j^0	componente vertical da posição inicial da célula j	46

SUMÁRIO

1 INTRODUÇÃO	19
1.1 CONTEXTUALIZAÇÃO DO PROBLEMA	19
1.2 JUSTIFICATIVA	21
1.3 OBJETIVOS	22
1.4 CONTRIBUIÇÕES CIENTÍFICAS	23
1.5 CONTRIBUIÇÕES TÉCNICAS	24
1.6 ORGANIZAÇÃO DESTE DOCUMENTO	24
2 CONCEITOS FUNDAMENTAIS	27
2.1 LEIAUTE E LEGALIZAÇÃO	27
2.2 MODELOS DE INTERCONEXÃO	30
2.3 ANÁLISE DE <i>TIMING</i> ESTÁTICA	34
2.3.1 Modelo de Grafo Adotado na <i>STA</i>	34
2.3.2 Informações Temporais de um Circuito	35
2.3.2.1 Atraso e <i>Slew</i> de uma Interconexão	36
2.3.2.2 Atraso e <i>Slew</i> de um Arco de <i>Timing</i>	38
2.3.3 As restrições <i>setup</i> e <i>hold</i>	38
2.3.4 Propagação das Informações Temporais na <i>STA</i>	40
2.3.4.1 Propagação de <i>slew</i>	43
2.3.5 Algoritmo de <i>STA</i> e Caminho Crítico	43
2.3.6 Métricas para Quantificação das Violações de Atraso 45	
2.4 POSICIONAMENTO INCREMENTAL GUIADO POR ATRASO	46
3 REVISÃO BIBLIOGRÁFICA	47
3.1 LEGALIZAÇÃO NAS TÉCNICAS DE <i>TDP</i>	47
3.2 TRABALHOS QUE UTILIZAM OS MODELOS CLIQUE E <i>STAR</i>	48
3.3 TRABALHOS QUE UTILIZAM <i>HPWL</i>	53
3.4 TRABALHOS QUE UTILIZAM ÁRVORES DE STEINER .	58
3.5 RESUMO	58
4 OTIMIZAÇÃO DE ATRASO PÓS-POSICIONAMENTO EXPLORANDO RAMOS DE STEINER NÃO-CRÍTICOS	59
4.1 VISÃO GERAL DA TÉCNICA PROPOSTA	59
4.2 AVALIAÇÃO DO POTENCIAL DE OTIMIZAÇÃO DOS RAMOS DE STEINER NÃO-CRÍTICOS	60
4.3 ALGORITMO PARA EXPLORAÇÃO DOS RAMOS DE STEINER NÃO-CRÍTICOS	62
4.3.1 Seleção dos Candidatos	63
4.3.2 Ordenamento dos Candidatos	64

4.3.3	Movimentação dos Candidatos	64
4.4	ANÁLISE DA COMPLEXIDADE E DOS PARÂMETROS DO ALGORITMO	66
4.5	MODIFICAÇÃO DA TÉCNICA PARA REDUÇÃO DE VIOLAÇÕES <i>EARLY</i>	66
5	AVALIAÇÃO EXPERIMENTAL DA TÉCNICA PROPOSTA	69
5.1	INFRAESTRUTURA E CONFIGURAÇÃO EXPERIMENTAL	69
5.2	REDUÇÃO DAS VIOLAÇÕES <i>LATE</i>	70
5.3	REDUÇÃO DAS VIOLAÇÕES <i>EARLY</i>	73
6	CONCLUSÕES E TRABALHOS FUTUROS	77
	REFERÊNCIAS	79
	APÊNDICE A – Lista de Produções	87
	APÊNDICE B – Redução de violações <i>Late</i> no decorrer de 10 iterações	93

1 INTRODUÇÃO

Este capítulo apresenta, inicialmente, uma breve contextualização do problema abordado neste trabalho. Em seguida, através de um exemplo, destaca-se a abordagem utilizada juntamente com a justificativa. Finalmente, são apresentados os objetivos e as contribuições alcançadas.

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA

O crescimento da complexidade dos circuitos digitais contemporâneos e a necessidade de um tempo de entrega ao mercado (*time-to-market*) curto faz com que o projeto de tais circuitos adote o fluxo *standard cell*. O projeto de um circuito digital inicia com a especificação do *hardware*, passa pelas etapas de síntese lógica, síntese física e verificação, concluindo pela prototipagem, teste, fabricação e entrega ao mercado.

O crescente impacto das interconexões no desempenho dos circuitos integrados vem aumentando a importância da síntese física no contexto da otimização do atraso (ALPERT; CHU; VILLARRUBIA, 2007) (ALPERT et al., 2012). O projeto de circuitos nas tecnologias de semicondutor de metal-óxido complementar (*CMOS: Complementary metal-oxide-semiconductor*) com comprimento mínimo de canal (dos transistores) superior a 90nm podia fazer uso de modelos de atraso de interconexão simplificados, uma vez que os atrasos das células¹ eram dominantes. Assim, a otimização do atraso dos circuitos podia basear-se apenas nas estimativas fornecidas pela síntese lógica. Contudo, nas tecnologias *CMOS* posteriores, onde o comprimento mínimo do canal dos transistores é igual ou inferior a 90nm, as capacitâncias e resistências dos fios são maiores do que as capacitâncias e resistências intrínsecas dos transistores e portanto, os atrasos dos fios são dominantes em relação aos atrasos das células (HO; MAI; HOROWITZ, 2001) (SAXENA et al., 2004). Desta forma, no contexto de tais tecnologias, é imprescindível se considerar informações de interconexões nas estimativas de atraso, para que otimizações no projeto físico não invalidem otimizações realizadas durante a síntese lógica (PAPA et al., 2011).

O lado direito da Figura 1 apresenta os principais passos da síntese física no fluxo de projetos. Segundo Kahng (2011), estes passos

¹Célula é uma instância de uma porta lógica. Ver Seção 2.1.

são:

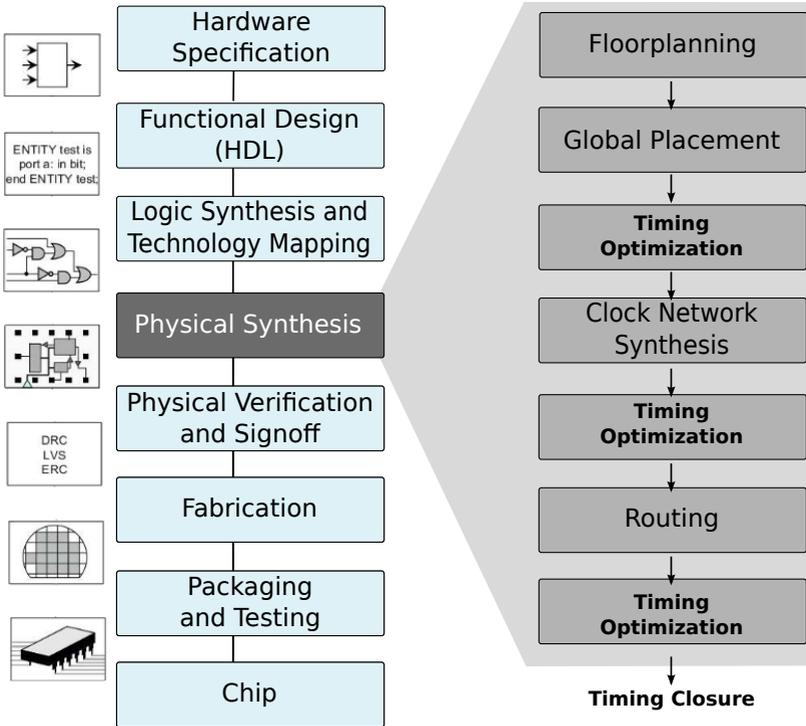


Figura 1 – Passos principais de um fluxo de projeto *VLSI*. Síntese física destacada à direita. Adaptado de (KAHNG, 2011).

- **Planejamento Topológico (*Floorplanning*):** determina a área de posicionamento, os formatos e o posicionamento dos sub-circuitos (ou módulos), bem como o posicionamento dos *pads* de entrada e saída e macroblocos;
- **Posicionamento Global (*Global Placement*):** distribui as células pela área de posicionamento disponível visando minimizar uma estimativa do comprimento das interconexões do circuito;
- **Síntese da Árvore de Relógio (*Clock Network Synthesis*):** distribui o sinal de relógio da fonte até os elementos sequenciais com o objetivo de minimizar a defasagem do relógio (*skew*), que é a diferença entre os tempos de chegada do sinal de relógio nas entradas dos registradores (BHASKER; CHADHA, 2009);

- **Roteamento (*Routing*)**: aloca os recursos necessários para que seja feito o roteamento do circuito (roteamento global). Em seguida assinala cada segmento de interconexão a trilhas ou colunas, escolhendo também o nível de metal a ser usado por cada segmento (roteamento detalhado).

Entre as etapas da síntese física, diversas técnicas são empregadas na otimização de atraso (*timing optimization*). Entre elas citam-se *gate sizing*, inserção de *buffers* e posicionamento guiado por atraso (*TDP: timing-driven placement*) (KAHNG, 2011). Apesar da grande importância do *gate sizing* e da inserção de *buffers* na síntese física, somente o *TDP* é capaz de alterar os comprimentos das interconexões.

O objetivo de *TDP* é otimizar o atraso crítico do circuito encurtando o comprimento das interconexões, porém, sem degradar a qualidade do posicionamento original (inicial) (KIM; HU; VISWANATHAN, 2014). Para isso, o *TDP* troca as posições das células do circuito com o objetivo de reduzir (ou corrigir, se possível) as violações de atraso.

TDP pode ser realizado na etapa de posicionamento global ou de maneira incremental, como uma etapa de otimização pós-posicionamento. O **TDP global** atribui pesos às interconexões com base em suas criticalidades e posiciona o circuito inteiro (negligenciando qualquer posicionamento anterior), visando minimizar o comprimento ponderado das interconexões. Já o **posicionamento incremental guiado por atraso** (*ITDP: incremental timing-driven placement*) é aplicado em um subconjunto de células a fim de otimizar o atraso referente às interconexões mais críticas, tentando preservar a qualidade do posicionamento anterior (solução inicial). Este trabalho de mestrado aborda o problema de *ITDP*.

1.2 JUSTIFICATIVA

O critério de seleção das portas a serem movidas constitui-se em característica determinante para a eficácia das técnicas de *ITDP*. Usualmente, as técnicas de *ITDP* se concentram em mover as células do caminho crítico a fim de encurtar as interconexões que determinam o atraso do circuito. Infelizmente, diversas técnicas negligenciam o impacto dos ramos não críticos dessas interconexões no atraso do caminho crítico. Isto ocorre por diversas razões, sendo a escolha do modelo de interconexões (Seção 2.2), uma das mais importantes. A Figura 2 mostra parte de um circuito industrial, onde os pontos azuis representam as células e os retângulos cinzas representam os módulos ou macroblocos

do circuito (ver Seção 2.1 para a definição de macrobloco). O caminho crítico é destacado em vermelho e é encurtado por uma técnica de *TDP* que se concentra neste caminho (*Critical Path Shortening*). Observe que os ramos não críticos deste caminho crítico – em verde – são negligenciados, resultando em perda de oportunidade de otimização uma vez que a carga capacitiva do caminho crítico poderia ser reduzida ainda mais caso tais ramos fossem encurtados. Neste caso, a aplicação de um passo de otimização com estratégia apropriada (*Non-Critical Branches Shortening*) é necessária para encurtar os ramos não críticos deixados para trás.

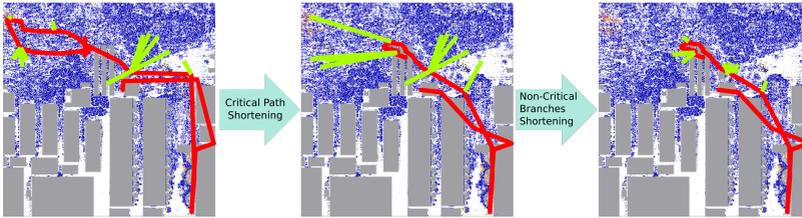


Figura 2 – Circuito: `superblue18` (KIM et al., 2015a) (KIM et al., 2015b). Adaptado pelo autor.

As diversas técnicas da literatura acabam relevando o impacto dos ramos não críticos, pelo uso de modelos de atraso não precisos. O presente trabalho busca contornar esta limitação otimizando o atraso de circuitos através do reposicionamento apenas das células não críticas. Portanto, este trabalho de mestrado busca verificar a seguinte hipótese: “o atraso crítico de um circuito pode ser significativamente reduzido mediante o reposicionamento das células não críticas.”.

1.3 OBJETIVOS

Este trabalho de mestrado tem como objetivo geral otimizar o atraso dos circuitos mediante o reposicionamento de células não críticas. Os objetivos específicos são listados a seguir:

1. Modelar as interconexões utilizando o modelo de comprimento de fio de árvore de Steiner (*StWL: Steiner tree wirelength*), em vez do comprimento de fio de semiperímetro (*HPWL: half perimeter wirelength*) (Ver Seção 2.2), a fim de obter uma maior precisão na estimativa dos movimentos;

2. Utilizar a variação do atraso das interconexões como métrica para estimar o potencial de otimização de movimentos de células não críticas;
3. Propor uma técnica de *ITDP* que realize movimentos legalizados nas células não críticas com base na métrica e no modelo de interconexões adotados;
4. Estender a técnica de *ITDP* proposta para tratar das violações *early*, isto é, violações de atraso relacionadas à restrição *hold* dos *flip-flops*;
5. Implementar um protótipo em **C++** e avaliar experimentalmente a técnica proposta.

1.4 CONTRIBUIÇÕES CIENTÍFICAS

Esta seção apresenta as contribuições alcançadas por este trabalho de mestrado.

- Proposta e avaliação de uma técnica de otimização de atraso pós-posicionamento para o projeto de circuitos digitais:
 - Modelagem explícita das interconexões utilizando *StWL*;
 - Aplicação da técnica proposta, no contexto de posicionamento incremental, respeitando restrições de deslocamento máximo (*maximum displacement*) com legalização instantânea;
 - Experimentos realizados sobre a solução obtida pelas três primeiras equipes colocadas na competição *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014). Tais experimentos mostraram que a técnica proposta reduz as violações *late* eficientemente sem causar prejuízos nas outras métricas de qualidade, *e.g.*, *average bin utilization (ABU)*, *TNS* e *WNS early*.
- Extensão de tal técnica para tratar de violações *early* após a otimização sobre as violações *late*;
- A técnica proposta foi aplicada como passo final de um fluxo de *ITDP* composto por três passos. O fluxo foi inscrito na categoria *Problem C: Incremental Timing-Driven Placement* do *CAD*

Contest @ ICCAD 2015 (KIM et al., 2015a) e obteve os melhores resultados considerando a métrica de qualidade da competição. O fluxo obteve também o menor tempo de execução entre os participantes. Conseqüentemente, tal fluxo recebeu o prêmio de primeiro colocado;

Como consequência das relevantes contribuições deste trabalho de mestrado, publicou-se o artigo (LIVRAMENTO et al., 2015) na conferência *International Conference on Computer-Aided Design (ICCAD)* de 2015, cuja relevância reflete na qualificação A1 do Qualis do ano de 2012².

1.5 CONTRIBUIÇÕES TÉCNICAS

- Adaptação da infraestrutura da competição *CAD Contest @ International Conference on Computer-Aided Design (ICCAD) 2014* (KIM; HU; VISWANATHAN, 2014) a fim de fornecer uma base de desenvolvimento de diversas técnicas de posicionamento e portanto, contribuir com esta e futuras pesquisas;
- Reaproveitamento de código e infraestrutura oriundos da implementação de um protótipo de ferramenta de *ITDP* baseada em relaxação Lagrangeana (*LR: lagrangian relaxation*). A ferramenta ficou classificada em 5º lugar na competição *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014) e técnica de otimização na qual a ferramenta se baseia foi relatada no artigo (GUTH et al., 2015), publicado no evento *International Symposium on Physical Design (ISPD) 2015 (Qualis 2012: A2)*;
- Uma ferramenta gráfica para visualização de posicionamento. A ferramenta é capaz de ler os formatos de arquivo mais utilizados na indústria e possibilita ao usuário uma interação manual, além da visualização do efeito de algoritmos de posicionamento no atraso e na densidade dos circuitos (Figura 3).

1.6 ORGANIZAÇÃO DESTE DOCUMENTO

Este documento está organizado da seguinte maneira: Os conceitos fundamentais para plena compreensão da técnica proposta neste

²A lista completa de produções pode ser conferida no Apêndice A.

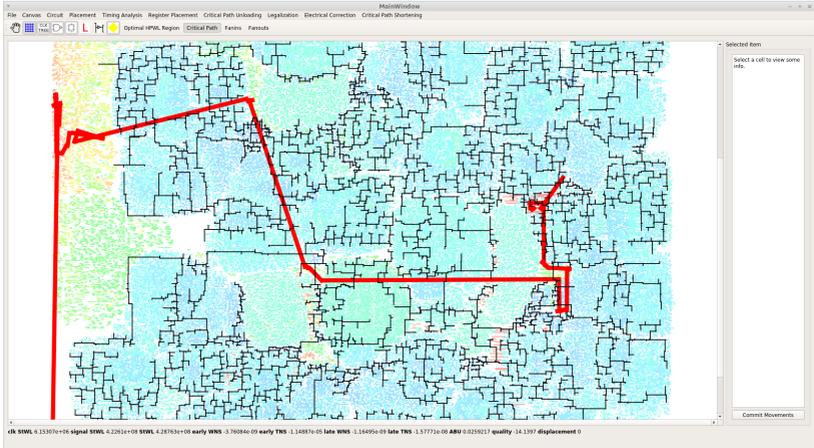


Figura 3 – Visualização do circuito b19 (KIM; HU; VISWANATHAN, 2014) na ferramenta gráfica desenvolvida. As cores dos retângulos representam os *slacks* das células do circuito. As linhas pretas são a árvore de relógio estimada pelo algoritmo *FLUTE* (CHU; WONG, 2008). As linhas vermelhas indicam a topologia do caminho crítico.

trabalho são apresentados no Capítulo 2. O Capítulo 3 apresenta a revisão dos trabalhos relacionados, destacando suas características e limitações. O Capítulo 4 apresenta a técnica de *ITDP* para otimizar o atraso de circuitos, explorando os ramos de Steiner não críticos. O Capítulo 5 apresenta a avaliação experimental da técnica proposta. Finalmente, o Capítulo 6 apresenta as conclusões.

2 CONCEITOS FUNDAMENTAIS

Este Capítulo apresenta alguns conceitos que são necessários para plena compreensão dos trabalhos correlatos (Capítulo 3) e da técnica proposta neste trabalho (Capítulo 4). Primeiramente, a Seção 2.1 apresenta os conceitos associados ao leiaute dos circuitos integrados, tais como célula, macrobloco, legalização e restrição de deslocamento máximo. Em seguida, a Seção 2.2 apresenta os principais modelos de interconexões adotados por técnicas de posicionamento incremental guiado por atraso (*ITDP: incremental timing-driven placement*). A Seção 2.3 apresenta informações relevantes sobre o cálculo do atraso dos circuitos. Finalmente, a Seção 2.4 apresenta a formulação do problema de *ITDP*.

2.1 LEIAUTE E LEGALIZAÇÃO

A descrição da geometria das máscaras utilizadas na fabricação do circuito integrado é referida genericamente por leiaute. Em particular, utiliza-se o termo **célula** para fazer referência ao leiaute de uma porta lógica ou ao leiaute de um elemento de armazenamento (*latch* ou *flip-flop*). Para viabilizar o projeto automático de circuitos integrados com milhões de portas lógicas, o fluxo atual de projeto baseia-se no conceito de *standard cells*, segundo o qual um conjunto limitado de células é projetado e caracterizado para uma dada tecnologia de fabricação (KAHNG, 2011). Os resultados da caracterização, em termos de atraso e potência, são tabulados e armazenados, juntamente com as descrições geométricas (área, posições dos pinos de entrada e saída *etc.*), em um repositório referenciado por **biblioteca de células**. Para cada função lógica, geralmente há mais de uma versão de leiaute, cada uma com capacidade de corrente de saída diferente (referenciadas por *X1, X2, X4 etc.*), o que dá origem a células distintas. Além disso, as células seguem uma topologia bastante restrita, todas com um formato retangular e com altura fixa. A Figura 4 mostra o leiaute de três células de uma biblioteca.

Conforme apresentado no Capítulo 1, a síntese física é responsável pela definição da topologia dos elementos do circuito (transistores e conexões) na superfície da lâmina de silício (*silicon wafer*). Durante o posicionamento global ou incremental, os circuitos são modelados como um hipergrafo de células (vértices), onde as hiperarestas representam

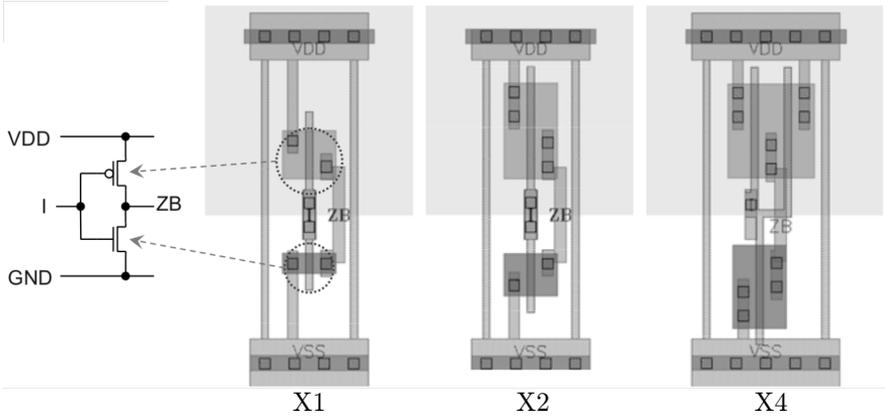


Figura 4 – Leiaute de três inversores com capacidades de corrente de saída diferentes. Fonte: Adaptado de (KAHNG, 2011).

as interconexões do circuito. Uma interconexão conecta a saída de uma célula, chamada de *driver*, às entradas de outras células, chamadas receptoras.

A etapa de planejamento topológico define os limites físicos do *chip* (X_l , X_r , Y_b , Y_t), a posição dos macroblocos, bem como a posição dos *pads* (pinos de entrada/saída) (Figura 5). Um **macrobloco** corresponde ao leiaute de um subcircuito implementado por terceiros, e por isso restringe a área de posicionamento. A área de posicionamento é dividida em bandas *standard cell* (linhas) e *sites* (colunas). Esta grade é necessária para que o roteamento possa ser realizado posteriormente.

O posicionamento global distribui as células pela área disponível. Nesta etapa, as células são consideradas como pontos, os quais podem assumir posições desalinhadas em relação às bandas e *sites*, e portanto, sobreposições entre células e macroblocos são admitidas.

A última etapa do posicionamento global determina as posições exatas das células, alinhando-as a uma grade e removendo as sobreposições (processo chamado de **legalização**) (Figura 6), a qual serve como referência para as posições válidas. Tal etapa recebe o nome de posicionamento detalhado.

Executado após o posicionamento global, o *ITDP* visa reduzir (ou remover) as violações de atraso mediante o reposicionamento de um subconjunto de células, com base em estimativas de atraso do circuito. Como o *ITDP* parte de uma solução inicial de posicionamento, é desejável (ou mesmo obrigatório) que a qualidade do posicionamento inicial

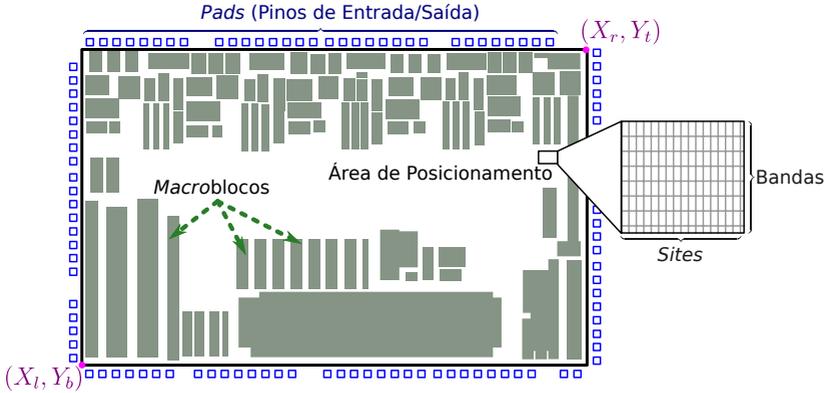


Figura 5 – Circuito superblue18 (KIM et al., 2015a). Adaptado pelo autor.



Figura 6 – Passo de legalização para alinhamento e remoção das sobreposições.

não seja deteriorada. Neste contexto, entende-se como qualidade o perfil de densidade do circuito e o comprimento total das interconexões. Portanto, para que mudanças no *ITDP* não sejam demasiado drásticas, uma restrição de “deslocamento máximo” (*maximum displacement*)¹ é incorporada ao problema (KIM; HU; VISWANATHAN, 2014) (Figura 7). A Seção 2.4 apresenta a formulação do problema de *ITDP*.

O *ITDP* busca encurtar o comprimento das interconexões que mais impactam no atraso do circuito. Tal impacto pode ser dividido em duas partes: 1) parcela correspondente ao atraso das interconexões dos caminhos críticos (proporcional às suas resistências e capacitâncias); 2) carga capacitiva do caminho crítico, que tem influência sobre os atrasos das células presentes nesses caminhos. Assim, o foco das técnicas de *ITDP* é diminuir o comprimento das interconexões do(s) caminho(s)

¹A restrição de deslocamento máximo define a máxima distância Manhattan que uma célula pode assumir em relação a sua posição inicial.

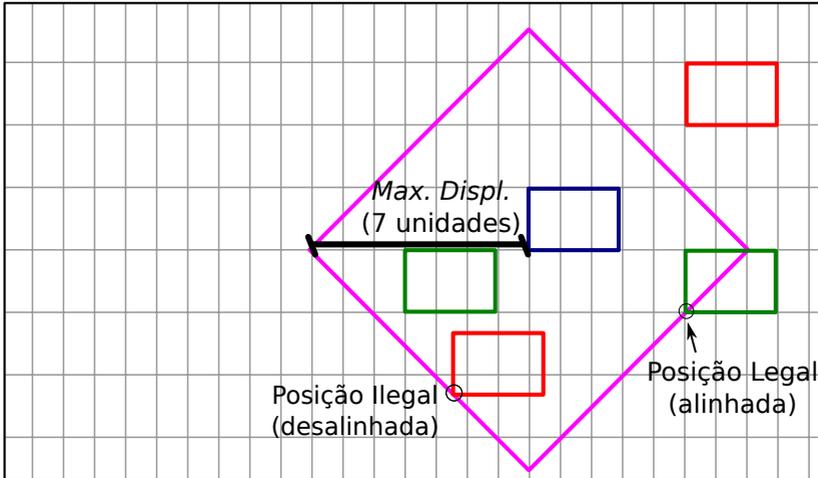


Figura 7 – O losango rosa representa a restrição de deslocamento máximo (*maximum displacement*) da célula em azul. Os retângulos verdes indicam posições legais, ou seja, posições alinhadas e dentro da área de deslocamento máximo. Os retângulos vermelhos indicam posições ilegais, desalinhadas e/ou além da restrição de deslocamento máximo. Note que a coordenada de referência é a inferior esquerda. Portanto, o retângulo verde que está posicionado sobre o limite da “área de deslocamento máximo” representa uma posição legal.

crítico(s), para que o atraso do circuito seja otimizado.

Como informações de roteamento ainda não estão disponíveis durante o *ITDP*, os comprimentos das interconexões precisam ser estimados com base em um modelo apropriado. A próxima seção apresenta os principais modelos.

2.2 MODELOS DE INTERCONEXÃO

A distância Manhattan provê uma estimativa aceitável para o comprimento de interconexões de dois pinos, já que o roteamento é feito em segmentos horizontais e verticais (KAHNG, 2011). A distância Manhattan entre dois pinos p_1 e p_2 é obtida somando-se os módulos das diferenças entre os valores de suas coordenadas:

$$d_M(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2| \quad (2.1)$$

Para interconexões com mais de dois pinos, os modelos mais utilizados são *Clique*, *Star*, *HPWL* e *StWL* (KAHNG, 2011), os quais são detalhados a seguir.

O modelo ***Clique*** (Figura 8) corresponde a um grafo completo dos pinos da interconexão. O comprimento utilizando o modelo *Clique* é obtido pela equação

$$Clique = \frac{2}{p} \sum_{e \in clique} d_M(e) \quad (2.2)$$

onde e é uma aresta do grafo *Clique*, p é o número de vértices do grafo, e $d_M(e)$ é a distância Manhattan entre os pinos da aresta e .

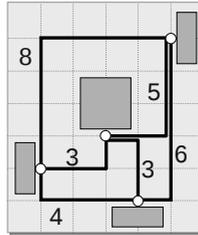


Figura 8 – Comprimento *Clique*: $14.5\mu\text{m}$. Fonte: (KAHNG, 2011).

O modelo ***Star*** (Figura 9) considera um pino como sendo o *fonte* (*source*) e os receptores são chamados sumidouros (*sinks*). Alternativamente, adiciona-se um ponto no centro da interconexão, o qual se conecta ao fonte e a cada destino. Assim, o comprimento da interconexão é estimado somando-se a distância Manhattan entre cada par de pinos (arestas). Como o modelo corresponde a uma árvore, apenas $p-1$ arestas são necessárias (onde p é o número de vértices, ou pinos), sendo vantajoso utilizar este modelo para interconexões com muitos pinos.

Ambos modelos *Clique* e *Star* são muito empregados em técnicas analíticas de posicionamento, que utilizam formulações de programação linear ou programação quadrática, como no posicionamento global. Porém, eles superestimam o comprimento real das interconexões.

O **comprimento de fio de semiperímetro (HPWL: half perimeter wirelength)** (Figura 10) estima o comprimento da inter-

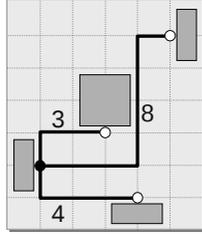


Figura 9 – Comprimento *Star*: $15\mu m$. Fonte: (KAHNG, 2011).

conexão usando a metade do perímetro do menor envelope que envolve seus pinos (*bounding box*). Este modelo é amplamente adotado por técnicas de *ITDP* por possuir custo computacional muito baixo e por estimar com exatidão o comprimento de interconexões com até três pinos (CHOI; BAZARGAN, 2003) (CHOWDHARY et al., 2005) (LUO; NEWMARK; PAN, 2006) (REN et al., 2007) (REN; PAN; KUNG, 2005) (PAPA et al., 2008). Porém, para interconexões com mais de três pinos, este modelo subestima o roteamento final.

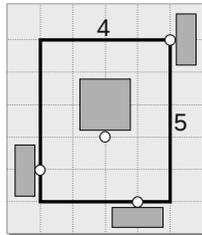


Figura 10 – *HPWL*: $9\mu m$. Fonte: (KAHNG, 2011).

O comprimento de fio de árvore de Steiner (*StWL: Steiner tree wirelength*) (Figura 11) é obtido modelando a interconexão como uma árvore de Steiner, e somando o comprimento de seus segmentos. Segundo Obermeier e Johannes (2004), este modelo é o que apresenta melhor correlação com o roteamento final, porém pode apresentar um custo computacional elevado para interconexões com muitos pinos.

Basear-se no *HPWL* para realizar o encurtamento das interconexões pode conduzir a um processo de otimização pouco efetivo, já que uma redução de *HPWL* não implica necessariamente em uma redução de *StWL*. A Figura 12 apresenta exemplos de como o *HPWL* pode levar a posicionamentos indesejados. O pontilhado azul apresenta a área

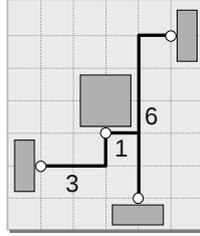


Figura 11 – $StWL$: $10\mu m$. Fonte: (KAHNG, 2011).

delimitada pelo $HPWL$ de uma interconexão que pertence ao caminho crítico. Tal interconexão conecta duas células críticas (em vermelho) e quatro não críticas (em verde). Suponhamos que uma técnica de posicionamento, cujo objetivo é reduzir o $HPWL$ das interconexões do caminho crítico, realiza movimentos nas células de maneira que estas assumam a configuração apresentada na parte central da Figura 12. Neste caso, mesmo com o aumento no $StWL$, a técnica **aceitaria** tais movimentos, uma vez que a estimativa do comprimento das interconexões utilizando o $HPWL$ reduziu. Outro exemplo de limitação do modelo $HPWL$ pode ser visto na parte mais à direita da figura. Neste caso, apesar dos movimentos reduzirem o $StWL$, a técnica **rejeitaria** tais movimentos, uma vez que estes não reduzem o $HPWL$.

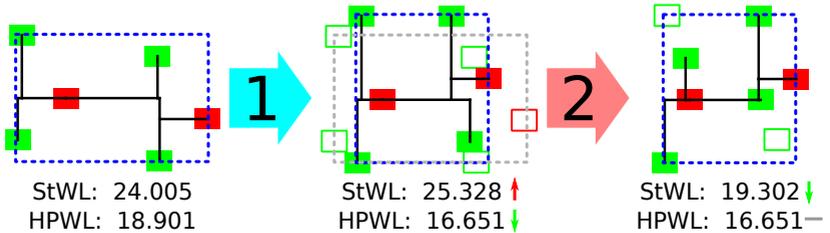


Figura 12 – (a) Uma interconexão representada pelo modelo $StWL$. (b) A mesma interconexão, após redução de $HPWL$. O $StWL$ é a soma do comprimento dos segmentos da árvore de Steiner, representados pelas linhas pretas. Já o $HPWL$, corresponde ao comprimento do semiperímetro do envelope, representado pela caixa pontilhada.

O $StWL$ possui alto custo computacional para ser adotado na etapa de posicionamento global. Porém, este alto custo pode ser tolerado nas técnicas de $ITDP$, uma vez que estas tendem a selecionar uma quantidade bastante limitada de células para serem reposiciona-

das. Por constituir-se em um modelo mais preciso de roteamento, o uso de *StWL* levaria a estimativas muito mais precisas de atraso, e conseqüentemente, a um melhor controle sobre a convergência da solução.

2.3 ANÁLISE DE *TIMING* ESTÁTICA

Para se estimar com precisão o atraso de um circuito, faz-se necessário o uso de um motor de *timing* (*timing engine*). Os motores de *timing* mais adequados às técnicas de otimização iterativas são aqueles que realizam a **análise de *timing* estática (*STA: static timing analysis*)** (GUNTZEL, 2000) (SAPATNEKAR, 2004) (BHASKER; CHADHA, 2009), um procedimento que percorre o grafo de *timing* e propaga as informações temporais com base nos modelos de atrasos.

2.3.1 Modelo de Grafo Adotado na *STA*

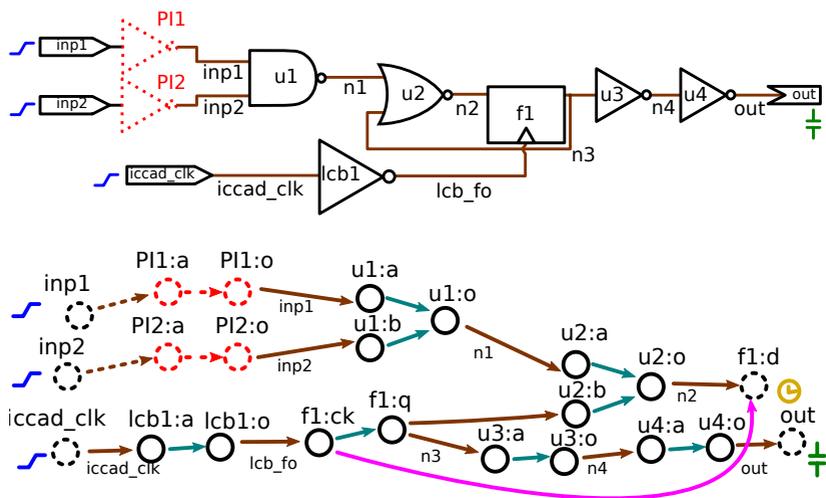
A *STA* geralmente modela o circuito como um grafo direcionado acíclico (*DAG: directed acyclic graph*) $G(\mathcal{V}, \mathcal{A})$: O conjunto \mathcal{V} é composto pelos vértices do grafo, os quais representam os pinos do circuito; O conjunto \mathcal{A} compreende as arestas, que podem representar arcos de *timing* combinacionais e sequenciais ou pares fonte-destino de uma interconexão.

Define-se **pino** (*timing point*) como sendo um pino de entrada ou de saída de uma célula ou um *pad* do circuito. O conjunto dos *pads* de entrada do circuito é denotado por \mathcal{PI} . Adicionalmente, define-se \mathcal{PO} , o conjunto de saídas primárias do circuito, o qual é composto pela união do conjunto dos *pads* de saída com o conjunto dos pinos de dados dos elementos sequenciais. Note que $(\mathcal{PO} \cup \mathcal{PI}) \subset \mathcal{V}$ e $\mathcal{PO} \cap \mathcal{PI} = \emptyset$.

Um **arco de *timing* combinacional** corresponde a funções de cálculo de atraso e *slew* entre um pino de entrada e um pino de saída de uma célula. O conjunto de arcos de *timing* combinacionais é denotado por \mathcal{A}_c . Um **arco de *timing* sequencial** define as restrições de *setup* e *hold* de um elemento sequencial (*e.g.*, *f1* da Figura 13). O conjunto de arcos de *timing* sequenciais é chamado \mathcal{A}_s . A Seção 2.3.3 apresenta as restrições *setup* e *hold* e a Seção 2.3.2 apresenta com mais detalhes os termos *slew*, e atraso.

A Figura 13 apresenta um circuito (acima) juntamente com seu grafo correspondente (centro). As restrições de projeto determinam que a *STA* considere que existam células conectadas às entradas primárias

do circuito para simular com mais precisão o *slew*. O exemplo da Figura 13 apresenta tais células, chamadas de **drivers de entrada primária** (*input drivers*), como inversores na cor vermelha, cujos nomes são *PI1* e *PI2*.



— Legenda —

- Pino de célula combinacional ou sequencial
- ⊙ Pino de *driver de entrada primária*
- ⊙ Entrada ou saída primária
- ⌋ Slews de entrada
- Arcos combinacionais
- Arcos combinacionais de *drivers de entradas primárias*
- Interconexões
- Arco sequencial
- ⊙ Restrição de desempenho
- ⊥ Capacitância da saída

Figura 13 – Grafo de *timing* para o circuito *simple* (KIM et al., 2015a).

2.3.2 Informações Temporais de um Circuito

- ***Slew* (tempo de transição):** é o tempo que leva para o sinal realizar uma transição completa. Se o sinal transiciona de nível lógico alto (V_{dd}) para nível lógico baixo ($0V$), a transição é de

descida. Dualmente, se o sinal transiciona de nível lógico baixo para nível lógico alto, a transição é chamada de subida. O *slew* em um arco de *timing* combinacional (representado por $\sigma_{(i,j)}$) é calculado em função do *slew* no pino de entrada e da carga capacitiva no pino de saída. O *slew* em um pino i é denotado por σ_i .

- **Atraso (*delay*):** tempo para que o sinal se propague da entrada até a saída de uma aresta (denotado por $\delta_{(i,j)}$), a qual representa um arco de *timing* ou um par fonte-destino de uma interconexão. Para um arco de *timing*, o atraso é obtido em função de sua capacitância de saída ($C_{out} = \text{output load}$) e de seu *slew* de entrada. Nas interconexões, o atraso é função de suas capacitâncias e resistências. **Notação:** $\delta_{(i,j)}$, sendo (i,j) uma aresta do grafo de *timing*;

Conforme apresentado nesta seção, o *slew* e o atraso são calculados para as arestas do grafo de *timing*, as quais podem representar tanto pares de pinos fonte-destino de interconexões quanto arcos de *timing*. As funções de atraso são detalhadas nas próximas duas subseções.

2.3.2.1 Atraso e *Slew* de uma Interconexão

O atraso e o *slew* de uma interconexão podem ser estimados quando esta é representada na forma de uma árvore RC. A Figura 14 (a) apresenta uma interconexão representada como uma árvore de Steiner, ligando a célula i às células j, k e l . s_1 e s_2 são pontos de Steiner da árvore. A Figura 14 (b) apresenta a árvore RC equivalente. Cada segmento de Steiner recebe um valor de capacitância e resistência concentradas, proporcionais aos comprimentos desses segmentos $(l_1, l_2, l_j, l_k, l_l)$. Os destinos da interconexão são representados por capacitores, referentes às capacitâncias dos pinos em que a interconexão se liga.

Para uma interconexão com N segmentos, o atraso do *driver* i até um destino k pode ser aproximado pelo atraso de Elmore (1948), apresentado na equação a seguir:

$$\delta_{(i,k)} = \sum_{n \in N} (R_i + r \cdot \sum_{\substack{m \in \{i \dots n\} \\ \cap \{i \dots k\}}} l_m) \cdot (c \cdot l_n + C_n) \quad (2.3)$$

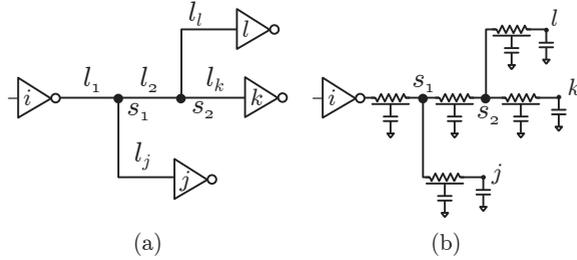


Figura 14 – (a) Uma interconexão representada como uma árvore de Steiner. (b) Cada segmento da árvore de Steiner é representado utilizando o modelo RC-distribuído.

onde R_i é a resistência do *driver* i da interconexão. C_n é o valor de capacitância do pino de entrada n de uma célula destino da interconexão. r e c são constantes que definem valores de resistência e capacitância por unidade de comprimento de fio.

O *slew* da resposta ao impulso ($\hat{\sigma}_j$) em um destino j de uma interconexão, que tem o pino i como fonte, é calculado conforme observado em (ELMORE, 1948) (GUPTA; TUTUIANU; PILEGGI, 1997), sendo aproximado com boa precisão por:

$$\hat{\sigma}_j = \sqrt{2\beta_j - \delta(i, j)} \quad (2.4)$$

onde β_j corresponde ao segundo momento da resposta ao impulso no destino j e $\delta(i, j)$ é o atraso de Elmore (1948) do par fonte-destino (i, j) .

Apesar do atraso de Elmore (1948) apresentar uma boa precisão na estimativa do atraso das interconexões, ele não considera o *slew* e o efeito de *resistive shielding*. Para este fim, outras técnicas mais apropriadas podem ser encontradas na literatura (KASHYAP; ALPERT; DEVGAN, 2000) (ALPERT; DEVGAN; KASHYAP, 2000) (PURI; KUNG; DRUMM, 2002). Neste trabalho, devido à infraestrutura adotada, utilizou-se o atraso de Elmore (1948), porém, a técnica apresentada no Capítulo 4 suporta qualquer modelo de interconexões que seja baseado em árvores RC.

2.3.2.2 Atraso e *Slew* de um Arco de *Timing*

No contexto deste trabalho, o atraso e o *slew* de um arco de *timing* combinacional (i, j) , representado por $\delta_{(i,j)}$, são aproximados pelas equações lineares 2.5 e 2.6, respectivamente. Os símbolos a, b, c, x, y e z são constantes obtidas previamente durante a caracterização das portas lógicas, e são disponíveis na biblioteca de atraso, para cada arco de *timing*. σ_i é o valor de *slew* na entrada do arco de *timing*. A carga capacitiva é aproximada pelo valor de capacitância concentrada na saída do arco de *timing* (i, j) e é denotada por C_j^{out} .

$$\delta_{(i,j)} = a + b\sigma_i + cC_j^{out} \quad (2.5)$$

$$\sigma_{(i,j)} = x + y\sigma_i + zC_j^{out} \quad (2.6)$$

Este modelo foi adotado por estar em conformidade com a infraestrutura da competição *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014), a qual foi adotada para a geração dos resultados experimentais apresentados no Capítulo 5. Porém, no caso de um fluxo de projeto industrial, o modelo utilizado seria o *NLDM* (*Non-Linear Delay Model*), o qual anota os valores dos atrasos e *slews* das células em tabelas disponibilizadas nas bibliotecas *Standard Cell*.

2.3.3 As restrições *setup* e *hold*

A *STA* pode ser realizada em circuitos sequenciais, considerando além da restrição de desempenho estipulada previamente, as restrições *setup* e *hold* dos elementos de armazenamento. Para tal, os circuitos sequenciais devem ser modelados como um conjunto de subcircuitos combinacionais delimitados por elementos sequenciais. Para avaliar o cumprimento de tais restrições, a *STA* propaga as informações temporais do grafo de *timing* em ordem topológica de maneira pessimista e otimista, denominadas de modo *late* e modo *early*, respectivamente. Serão empregados os sobrescritos L e E para as notações relacionadas ao modo *late* e *early*, respectivamente.

A Figura 15 apresenta a representação genérica de um circuito sequencial composto de dois *flip-flops* $f1$ e $f2$ (a rede de relógio é abstraída), os quais delimitam um circuito combinacional qualquer, denominado *comb*. O maior (menor) atraso do circuito combinacional é denotado por δ_{comb}^L (δ_{comb}^E), ao passo que $\delta_{(clk,f1;q)}^L$ ($\delta_{(clk,f1;q)}^E$) corres-

ponde ao maior (menor) atraso da fonte de relógio (clk) até a saída do *flip-flop* $f1$.

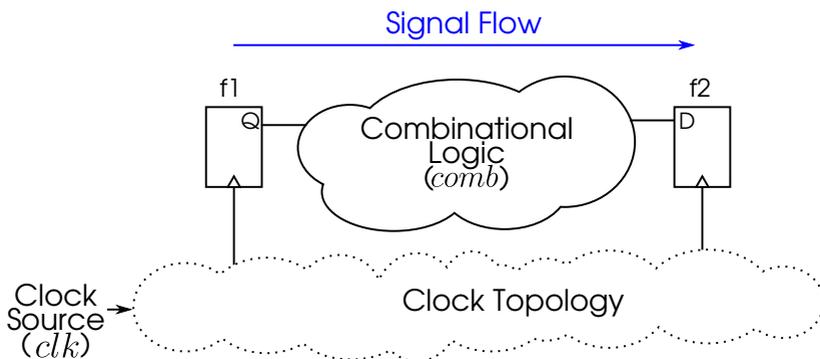


Figura 15 – Um circuito sequencial composto de dois *flip-flops*.

O modo *late* está relacionado à restrição de *setup* dos elementos sequenciais. O tempo de *setup* de um *flip-flop* especifica quanto tempo antes da borda de relógio que ativa sua transição o sinal no pino de dados deve estar estável, para que ele possa ser capturado e armazenado corretamente. Uma violação na restrição de *setup*, chamada de violação *late*, ocorre quando o sinal se estabiliza no pino de dados após momento permitido. A Figura 16 apresenta um caso onde uma transição no sinal do elemento sequencial $f1$ da Figura 15 se estabiliza no pino de dados do *flip-flop* $f2$ em um momento anterior ao estabelecido pela restrição de *setup*, resultando em uma folga na restrição de desempenho (*slack*) no modo *late* ($\delta_{(clk,f1;q)}^L + \delta_{comb}^L \leq t_{f2}^{su}$). Por outro lado, a Figura 17 apresenta um caso onde o sinal no pino de dados do elemento $f2$ se estabiliza após o momento permitido, gerando assim, uma violação na restrição *setup* (*violation*) no modo *late* ($\delta_{(clk,f1;q)}^L + \delta_{comb}^L > t_{f2}^{su}$).

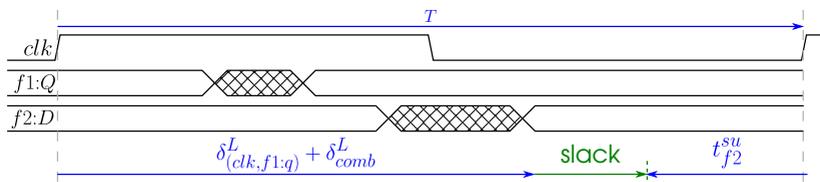


Figura 16 – Configuração que **respeita** a restrição *setup* para o circuito da Figura 15.

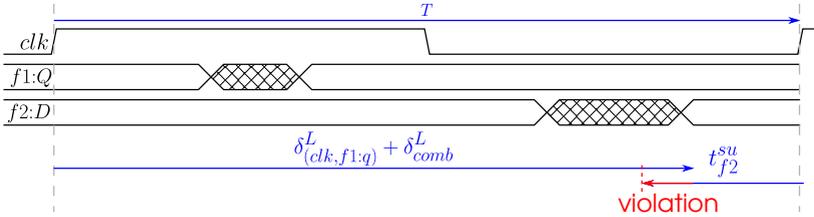


Figura 17 – Configuração que **viola** a restrição *setup* para o circuito da Figura 15.

O modo *early* está relacionado à restrição de *hold* dos elementos sequenciais. O tempo de *hold* de um *flip-flop* especifica o momento mais cedo em que o sinal no pino de dados deve estar estável, para que ele possa ser capturado e armazenado corretamente. A Figura 18 apresenta um caso onde uma transição no sinal do elemento sequencial *f1* da Figura 15 se estabiliza no pino de dados do *flip-flop* *f2* em um momento posterior ao estabelecido pela restrição de *hold*, resultando em uma folga na restrição de desempenho (*slack*) no modo *early* ($\delta_{(clk, f1:q)}^E + \delta_{comb}^E \geq t_{f2}^h$). Caso o sinal se estabilize antes do momento definido pela restrição, o sinal de um estado anterior deste elemento sequencial pode ser sobrescrito antes de ter sido propagado corretamente, causando uma violação *early*. No exemplo da Figura 15, tal situação ocorre se $\delta_{(clk, f1:q)}^E + \delta_{comb}^E < t_{f2}^h$ (Figura 19).

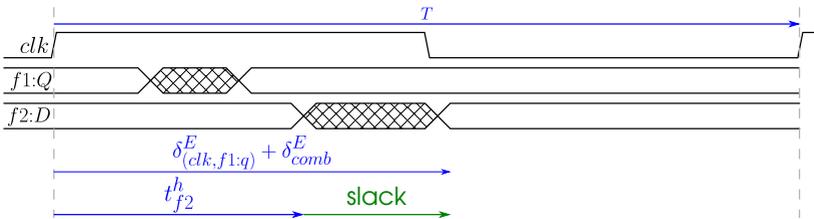


Figura 18 – Configuração que **respeita** a restrição *hold* para o circuito da Figura 15.

2.3.4 Propagação das Informações Temporais na *STA*

As principais informações propagadas são os tempos de chegada, os *slows*, os tempos requeridos e os *slacks* em cada pino:

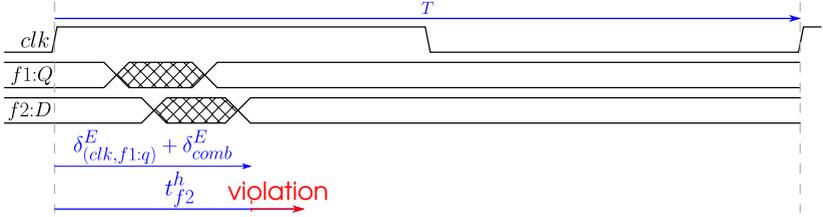


Figura 19 – Configuração que **viola** a restrição *hold* para o circuito da Figura 15.

Tempo de chegada (*arrival time*) de um pino: é o tempo máximo que transcorre desde um sinal em uma entrada primária passar por uma transição até o instante em que esta transição se manifesta no referido pino. O tempo de chegada *late* em um pino i corresponde ao maior entre os tempos de chegada *late* das arestas que incidem para i (Equação 2.7), onde o tempo de chegada *late* em uma aresta (j, i) é calculado como a soma entre o tempo de chegada *late* no pino j (at_j^L) e o atraso de (j, i) ($\delta_{(j,i)}$). Analogamente, calcula-se o tempo de chegada *early* (Equação 2.8) propagando a menor entre as somas de tempo de chegada *early* na entrada (at_j^E) e atraso ($\delta_{(j,i)}^E$).

$$at_i^L = \max_{(j,i) \in \mathcal{A}} (at_j^L + \delta_{(j,i)}^L) \quad (2.7)$$

$$at_i^E = \min_{(j,i) \in \mathcal{A}} (at_j^E + \delta_{(j,i)}^E) \quad (2.8)$$

Tempo requerido (*required time*) de um pino: é o tempo máximo permitido para que uma transição em um sinal de entrada se manifeste no referido pino. Para os *pads* de saída, o tempo requerido *late* (rt_i^L) é igual ao período do relógio (T) e o tempo requerido *early* (rt_i^E) é de 0s (Equações 2.9 e 2.10).

$$rt_i^L = T \quad (2.9)$$

$$rt_i^E = 0 \quad (2.10)$$

Para os pinos de dados das células sequenciais, o tempo requerido é obtido em função das restrições de *setup* e *hold*. Considere um *flip-flop* $f1$, o qual possui pinos de entrada de relógio e dados chamados de $f1ck$ e $f1d$, respectivamente. Como $f1$ é um elemento sequencial, existe uma aresta $(f1ck, f1d) \in \mathcal{A}_s$. As equações 2.11 e 2.12 são utilizadas

para o cálculo dos tempos requeridos rt_{f1d}^L e rt_{f1d}^E no pino de dados do *flip-flop* $f1$, sendo t_{f1}^{su} e t_{f1}^h os tempos de *setup* e *hold* deste *flip-flop*.

$$rt_{f1d}^L = T + at_{ck}^E - t_{f1}^{su} \quad (2.11)$$

$$rt_{f1d}^E = at_{ck}^L + t_{f1}^h \quad (2.12)$$

Para os demais pinos, o tempo requerido é propagado em ordem topológica reversa. O tempo requerido *late* em um pino i corresponde ao menor entre os tempos requeridos *late* das arestas que incidem de i (Equação 2.13), onde o tempo requerido *late* em uma aresta (i, j) é calculado como a diferença entre o tempo requerido *late* no pino j (rt_j^L) e o atraso de (i, j) ($\delta_{(j,i)}$). Analogamente, calcula-se o tempo requerido *early* (Equação 2.14) propagando a menor diferença de tempo requerido *early* na saída (rt_j^E) e atraso ($\delta_{(i,j)}^E$).

$$rt_i^L = \min_{(i,j) \in A} (rt_j^L - \delta_{(i,j)}^L) \quad (2.13)$$

$$rt_i^E = \max_{(i,j) \in A} (rt_j^E - \delta_{(i,j)}^E) \quad (2.14)$$

Slack (folga) de um pino: indica quanto um determinado pino viola ou respeita a restrição de desempenho. No modo *late*, o *slack* de um pino i (Equação 2.15) é calculado como a diferença entre tempo requerido (rt_i^L) e tempo de chegada (at_i^L). Já no modo *early*, o *slack* de um pino i (Equação 2.16) é a diferença entre o tempo de chegada (at_i^E) e o tempo requerido (rt_i^E).

$$s_i^L = rt_i^L - at_i^L \quad (2.15)$$

$$s_i^E = at_i^E - rt_i^E \quad (2.16)$$

Um valor de *slack* igual a zero ($s_i = 0$) indica o cumprimento da restrição de desempenho. Um valor positivo para o *slack* ($s_i > 0$) indica o cumprimento, com folga, da restrição de desempenho. Um valor negativo ($s_i < 0$) indica uma **violação de atraso**. Suponha que em uma saída primária do circuito o tempo requerido seja de $50ps$. Se o tempo de chegada for de $50.2ps$, o *slack* seria de $-0.2ps$, indicando uma violação de atraso de $0.2ps$. Se o tempo de chegada fosse de $48ps$, não existiria violação de atraso neste caso.

2.3.4.1 Propagação de *slew*

O *slew* nas entradas primárias dos circuitos são definidos previamente e são constantes. Dada uma célula qualquer, o *slew late* (*early*) em seu pino de saída i será o maior (menor) entre os *slews* de seus arcos de *timing* no modo *late* (*early*), conforme definido na Equação 2.17 (2.18).

$$\sigma_i^L = \max_{(j,i) \in \mathcal{A}} (\sigma_{(j,i)}^L) \quad (2.17)$$

$$\sigma_i^E = \min_{(j,i) \in \mathcal{A}} (\sigma_{(j,i)}^E) \quad (2.18)$$

Para uma interconexão cujo pino *driver* é i , a propagação do *slew* para qualquer destino j é realizada estendendo o *slew* de resposta ao impulso $\hat{\sigma}_j$ neste destino para uma rampa, conforme apresentado em (KASHYAP et al., 2004) (Equações 2.19 e 2.20).

$$\sigma_i^L = \sqrt{(\sigma_j^L)^2 + \hat{\sigma}_o^2} \quad (2.19)$$

$$\sigma_i^E = \sqrt{(\sigma_j^E)^2 + \hat{\sigma}_o^2} \quad (2.20)$$

2.3.5 Algoritmo de *STA* e Caminho Crítico

O pseudocódigo do algoritmo de *STA* é apresentado no Algoritmo 1. O algoritmo recebe como entradas o grafo de *timing*, valores para os *slews* e tempos de chegadas nos pinos de entrada primária, a restrição de desempenho T e os valores de capacitância para os pinos de saída primária. O algoritmo calcula as informações temporais (tempo de chegada, *slew*, *slack* e tempo requerido) em todos os pinos do circuito. As linhas 1 até 12 iteram pelos vértices do grafo de *timing* em ordem topológica, propagando os tempos de chegada e os *slews* conforme apresentado nas Equações 2.7 e 2.8, e nas Equações 2.17 até 2.20, respectivamente. As linhas 13 até 21 computam os tempos requeridos nas saídas primárias através das Equações 2.9 até 2.12. Finalmente, as linhas 22 até 27 propagam os tempos requeridos em ordem topológica reversa para enfim obter os valores de *slack* em todos os pinos do circuito, utilizando as equações 2.13 até 2.16.

Algoritmo 1: STATIC_TIMING_ANALYSIS

input : $G(\mathcal{V}, \mathcal{A})$; $s_i^L, s_i^E, at_i^L, at_i^E \forall i \in \mathcal{PL}; T; C_i \forall i \in \mathcal{PAD}$
output: $at_i^L, at_i^E, \sigma_i^L, \sigma_i^E, rt_i^L, rt_i^E, s_i^L, s_i^E \forall i \in \mathcal{V}$

- 1 **foreach** $i \in \mathcal{V}$ *in topological order* **do**
- 2 $at_i^L \leftarrow \max_{(j,i) \in \mathcal{A}} (at_j^L + \delta_{(j,i)}^L)$
- 3 $at_i^E \leftarrow \min_{(j,i) \in \mathcal{A}} (at_j^E + \delta_{(j,i)}^E)$
- 4 $\sigma_i^L \leftarrow \max_{(j,i) \in \mathcal{A}} (\sigma_{(j,i)}^L)$
- 5 $\sigma_i^E \leftarrow \min_{(j,i) \in \mathcal{A}} (\sigma_{(j,i)}^E)$
- 6 **foreach** $(i, j) \in \mathcal{A}$ **do**
- 7 $\delta_{(i,j)}^L \leftarrow$ late interconnection or timing-arc delay
- 8 $\delta_{(i,j)}^E \leftarrow$ early interconnection or timing-arc delay
- 9 $\sigma_{(i,j)}^L \leftarrow$ late interconnection or timing-arc *slew*
- 10 $\sigma_{(i,j)}^E \leftarrow$ early interconnection or timing-arc *slew*
- 11 **end**
- 12 **end**
- 13 **foreach** $(ck, d) \in \mathcal{A}_s$ **do**
- 14 $f =$ flip-flop that owns (ck, d)
- 15 $rt_d^L \leftarrow T + at_{ck}^E - t_f^{su}$
- 16 $rt_d^E \leftarrow at_{ck}^L + t_f^h$
- 17 **end**
- 18 **foreach** $i \in \mathcal{PAD}$ **do**
- 19 $rt_i^L \leftarrow T$
- 20 $rt_i^E \leftarrow 0$
- 21 **end**
- 22 **foreach** $i \in \mathcal{V}$ *in reverse topological order* **do**
- 23 $rt_i^L \leftarrow \min_{(i,j) \in \mathcal{A}} (rt_j^L - \delta_{(i,j)}^L)$
- 24 $rt_i^E \leftarrow \max_{(i,j) \in \mathcal{A}} (rt_j^E - \delta_{(i,j)}^E)$
- 25 $s_i^L \leftarrow rt_i^L - at_i^L$
- 26 $s_i^E \leftarrow at_i^E - rt_i^E$
- 27 **end**

A execução da *STA* permite identificar os caminhos críticos do circuito. Define-se **caminho crítico**, o caminho com maior atraso do circuito. Este, por sua vez, é estimado somando os atrasos de todas as suas arestas, partindo de uma entrada primária e terminando em uma saída primária. Alternativamente, pode-se definir como caminho crítico aquele que possui o menor *slack* do circuito. Utiliza-se o termo **célula crítica** para uma célula que possui *slack* negativo em algum

de seus pinos. Complementarmente, **célula não crítica** é aquela que possui apenas *slacks* não-negativos em seus pinos, ou seja, maiores ou iguais a zero.

2.3.6 Métricas para Quantificação das Violações de Atraso

Com os valores de *slack* estimados nas saídas primárias, as violações de atraso podem ser mensuradas por duas métricas: **pior slack negativo (WNS: worst negative slack)**, que corresponde à quantidade de violações de atraso na saída primária do caminho crítico (equações 2.21 e 2.22) e **total de slack negativo (TNS: total negative slack)**, que corresponde à soma das violações de atraso em todas as saídas primárias do circuito (equações 2.23 e 2.24).

$$\text{WNS}^L = \min(\min_{i \in \mathcal{PO}}(s_i^L), 0) \quad (2.21)$$

$$\text{WNS}^E = \min(\min_{i \in \mathcal{PO}}(s_i^E), 0) \quad (2.22)$$

$$\text{TNS}^L = \sum_{i \in \mathcal{PO}} \min(0, s_i^L) \quad (2.23)$$

$$\text{TNS}^E = \sum_{i \in \mathcal{PO}} \min(0, s_i^E) \quad (2.24)$$

No fluxo de projeto de circuitos digitais, faz-se o uso de diversas técnicas de otimização com o objetivo de reduzir tais métricas que quantificam as violações de atraso. As violações *late* são tratadas mediante a redução do atraso dos circuitos. O *TDP* aborda tal problema encurtando o comprimento das interconexões mais críticas do circuito, reduzindo a carga capacitiva dos caminhos críticos e consequentemente, seus atrasos. Se a solução das violações *late* durante a síntese física for impossível, poder-se-ia, após a fabricação do circuito, reduzir a frequência de operação, obtendo-se porém, um circuito mais lento.

As violações no modo *early* são mais difíceis de serem resolvidas. Uma vez que tais violações estão relacionadas aos caminhos mais rápidos do circuito, não é possível resolvê-las modificando a frequência de relógio, por isso tais violações precisam ser resolvidas no máximo, até o fim da síntese física. Diversas técnicas podem ser aplicadas para resolver as violações *early*, como a inserção de *buffers* (PAVISIC et al., 2003), remoção de *buffers*, *gate sizing* o *TDP* (KOTECHA et al., 2011). No caso

do *TDP*, alongam-se as interconexões críticas (para o modo *early*) e/ou reposicionam-se os elementos sequenciais. Porém, alongando-se as interconexões, aumenta-se a dificuldade para realização do roteamento, além de dificultar o controle do *slew*, causando impactos negativos no atraso e no consumo de potência dinâmica do circuito.

2.4 POSICIONAMENTO INCREMENTAL GUIADO POR ATRASO

O objetivo do posicionamento incremental guiado por atraso (*ITDP: incremental timing-driven placement*) é minimizar as violações de atraso mediante o reposicionamento de um subconjunto de células do circuito, respeitando as restrições de legalidade e deslocamento máximo. Minimizar as violações de atraso implica em maximizar o somatório de *WNS* e *TNS* nos modos *late* e *early*, já que se tratam de valores de *slack* **negativos**.

Considere o conjunto \mathcal{C} como sendo o conjunto das células do circuito. As equações 2.25 até 2.31 apresentam a formulação para o problema do *ITDP*. A função objetivo é apresentada na Equação 2.25. As restrições das equações 2.26 e 2.27 garantem que nenhuma célula será posicionada além da área de posicionamento. W_j e H_j correspondem à largura e altura de uma célula j . x_j e y_j são as coordenadas de j . As equações 2.28 e 2.29 correspondem à restrição de alinhamento. W_{site} e H_{row} correspondem à largura dos *sites* e altura das bandas, respectivamente. A Inequação 2.30 impede que haja sobreposições entre as células de uma mesma linha. Finalmente, a restrição da Inequação 2.31 garante que nenhuma célula será posicionada além da restrição de deslocamento máximo (D_{max}).

$$\mathbf{Max} : WNS^L + WNS^E + TNS^L + TNS^E \quad (2.25)$$

$$\mathbf{S.t.} : X_l \leq x_j \leq X_r - W_j \quad (2.26)$$

$$: Y_b \leq y_j \leq Y_t - H_j \quad (2.27)$$

$$: x_j = step_x \times W_{site}, step_x \in \mathbb{Z} \quad (2.28)$$

$$: y_j = step_y \times H_{row}, step_y \in \mathbb{Z} \quad (2.29)$$

$$: x_k + W_k \leq x_{k+1}, y_k = y_{k+1} \quad (2.30)$$

$$: \max_{j \in \mathcal{C}} (|x_j - x_j^0| + |y_j - y_j^0|) \leq D_{max} \quad (2.31)$$

3 REVISÃO BIBLIOGRÁFICA

Esta seção apresenta uma revisão dos trabalhos correlatos, isto é, aqueles que apresentam técnicas de posicionamento guiado por atraso (*TDP: timing-driven placement*) globais ou incrementais. Entretanto, a análise crítica de tais técnicas leva em conta o escopo do presente trabalho, ou seja, redução do atraso mediante o reposicionamento de células não críticas, de maneira incremental. Os trabalhos são classificados de acordo com três características:

1. Modelo de interconexão adotado (Seção 2.2);
2. Se a técnica move células não críticas na otimização de atraso;
3. Abordagem de legalização utilizada.

A Tabela 1 apresenta um resumo dos trabalhos, os quais são detalhados nas Seções 3.2 até 3.4, agrupados por modelo de interconexão. Na Tabela-resumo, três tipos de legalização classificam os trabalhos correlatos:

- **Iterativa:** quando as técnicas reposicionam as células ignorando as restrições de legalidade e no fim de cada iteração, legalizam o circuito todo;
- **Passo Final:** quando a legalização é realizada após o fim do fluxo de otimização de atraso;
- **On-the-fly:** quando a legalização é realizada a cada movimento.

3.1 LEGALIZAÇÃO NAS TÉCNICAS DE *TDP*

Diversas técnicas de *TDP* posicionam as células assumindo posições desalinhadas e/ou desconsiderando as sobreposições durante a otimização (WANG; LILLIS; SANYAL, 2005) (RAJAGOPAL et al., 2003) (VISWANATHAN et al., 2010) (CHOI; BAZARGAN, 2003) (CHOWDHARY et al., 2005) (LUO; NEWMARK; PAN, 2006) (REN; PAN; KUNG, 2005). Porém, aplicar a legalização como um passo posterior à otimização de *timing* pode acarretar em degradação das estimativas de atraso, impactos significativos no perfil de densidade, e até mesmo, impossibilitar a legalização (PAPA et al., 2008). Por este motivo, algumas técnicas de posicionamento incremental guiado por atraso (*ITDP: incremental*

timing-driven placement) consideram as restrições de legalidade e resolvem as violações durante a otimização de atraso (REN et al., 2007)(PAPA et al., 2008). Isto acarreta em melhores estimativas de atraso e melhor controle sobre a convergência da técnica de otimização. Este tipo de legalização é chamada *on-the-fly*.

3.2 TRABALHOS QUE UTILIZAM OS MODELOS CLIQUE E *STAR*

Rajagopal et al. (2003) propuseram uma técnica de posicionamento baseado em forças (*force directed placement*). Os autores apontam que como a importância dos receptores de uma interconexão (no atraso do circuito) não é uniforme, é desejável que as restrições não sejam aplicadas às interconexões como um todo, mas sim considerando cada receptor individualmente. Para tal os autores utilizam o modelo *Star* para representar as interconexões.

A abordagem proposta adiciona restrições apenas às interconexões críticas, para que o impacto no *wirelength* do circuito seja mínimo. A restrição de uma interconexão é um valor limite superior (*upper-bound*) para o seu valor de *HPWL*. Os autores utilizam heurísticas para identificar as interconexões críticas. Tais heurísticas consideram a capacidade de corrente e degradação do *slew* através dos *drivers*, a capacitância das interconexões e de seus pinos receptores, e a variação de *slew* através dos *drivers*. Após identificar as interconexões críticas, aumenta-se o peso daquelas que não respeitam suas respectivas restrições. Além disso, o algoritmo também aumenta o peso dos segmentos que ligam o *driver* aos receptores críticos. O peso de uma interconexão é aumentado até um certo valor limite w_{max} , pois os autores apontam que a partir de um momento, o aumento do peso para de colaborar com a redução do perímetro das interconexões (Figura 20).

Limitação: a técnica não adiciona peso às arestas *Star* que conectem células que estejam dentro da área delimitada pelo *HPWL*. Os autores argumentam que movimentos nessas células não são capazes de reduzir o atraso das interconexões, uma vez que não reduzem o comprimento da interconexão quando este é estimado pelo *HPWL*. Esta estratégia faz com que a técnica **negligencie o impacto dos ramos não críticos** das interconexões críticas no atraso do circuito, se estes estiverem dentro da área delimitada pelo *HPWL*. Os resultados experimentais mostram uma degradação no atraso do circuito de aproximadamente 11% causada pela aplicação da **legalização como um passo final**.

Tabela 1 – Resumo dos trabalhos correlatos.

Trabalho	Modelo de Interconexão	Move células não críticas	Legalização
Rajagopal et al. (2003)	<i>Star</i>	X	Iterativa
Wang, Lillis e Sanyal (2005)	<i>Star</i>	X	Iterativa
Viswanathan et al. (2010)	<i>Star</i>	X	Iterativa
Obermeier e Johannes (2004)	Clique	X	Passo final
Choi e Bazargan (2003)	<i>HPWL</i>	X	Iterativa
Chowdhary et al. (2005)	<i>HPWL</i>	X	Iterativa
Ren, Pan e Kung (2005)	<i>HPWL</i>	X	Passo final
Luo, Newmark e Pan (2006)	<i>HPWL</i>	X	Passo final
Ren et al. (2007)	<i>HPWL</i>	X	<i>On-the-fly</i>
Papa et al. (2008)	<i>HPWL</i>	X	<i>On-the-fly</i>
Ajami e Pedram (2001)	<i>StWL</i>		–
Este Trabalho	StWL	X	<i>On-the-fly</i>

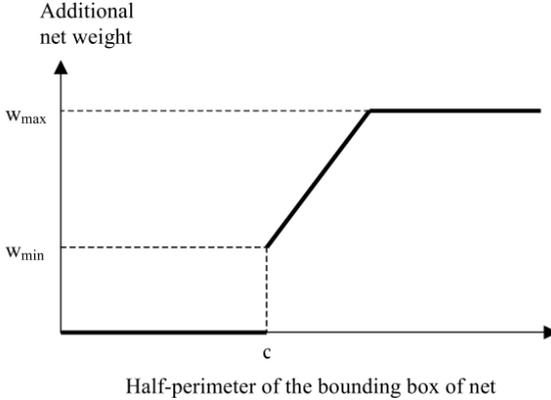


Figura 20 – Estratégia para adicionar peso às interconexões. O peso é mantido em 0 para as que possuem *HPWL* menor que a restrição c . Para as interconexões com $HPWL > c$, o peso é aumentado até que o seu valor atinja w_{max} . Fonte: (RAJAGOPAL et al., 2003).

Wang, Lillis e Sanyal (2005) propuseram uma técnica baseada em programação linear (*LP: linear programming*) que modela o atraso dos caminhos nas restrições do problema. A técnica proposta compreende três passos iterativos: 1) extração de um subcircuito a ser otimizado; 2) solução do problema linear que relaxado, isto é, que posiciona otimamente as células, ignorando as restrições de legalidade; 3) aplicação de uma técnica para remover as sobreposições. O atraso das interconexões são incorporados às restrições do problema na forma de funções lineares de seus valores de *HPWL*.

Limitação: os autores utilizam o *HPWL* na estimativa do comprimento das interconexões para incorporar seus atrasos na formulação linear. Porém, como o *HPWL* subestima o comprimento das interconexões, o atraso não é modelado precisamente. O problema linear é resolvido sem considerar as restrições de legalidade e é legalizado iterativamente. Tal estratégia de legalização pode causar degradações na estimativa de atraso do circuito. A infraestrutura utilizada na avaliação experimental é composta por circuitos pequenos (até 8383 células) e os tempos de execução não foram reportados em valores absoluto. Portanto não é possível concluir se a técnica proposta é escalável.

Obermeier e Johannes (2004) apresentaram um método global de *TDP* baseado em programação quadrática (*QP: quadratic programming*), onde as interconexões do circuito recebem pesos de acordo

com suas sensibilidades e *slacks*. A técnica utiliza o modelo *Clique* para incorporar à formulação quadrática cada segmento individualmente, baseando-se em árvores de Steiner para o assinalamento dos pesos. Após a otimização, a técnica baseada em fluxo em redes (DOLL; JOHANNES; ANTREICH, 1994) é aplicada para se obter um posicionamento legalizado.

Os autores argumentam que nem sempre os segmentos com menor *slack* são os que devem ser reduzidos. Por exemplo, considerando a interconexão apresentada na Figura 21, se os pinos i e k têm *slack* negativo e os pinos l e j têm *slack* positivo, em um caso extremo onde i e k assumem a mesma posição, o *slack* do caminho crítico só melhoraria se os segmentos $s_1 \rightarrow j$ e $s_2 \rightarrow l$ fossem encurtados, uma vez que esses seriam os únicos movimentos que ainda poderiam ser realizados para diminuir a carga carregada pelo pino i . A técnica de Obermeier e Johannes (2004), portanto, utiliza os *slacks* juntamente da sensibilidade de cada segmento para assinalar os pesos às interconexões. A sensibilidade leva em conta o impacto dos ramos (inclusive os não críticos) nos atrasos do caminho crítico. Ou seja, além dos segmentos críticos que devem ser reduzidos, a análise de sensibilidade permite também reduzir os ramos não críticos que apresentem impacto negativo nos atrasos dos caminhos críticos.

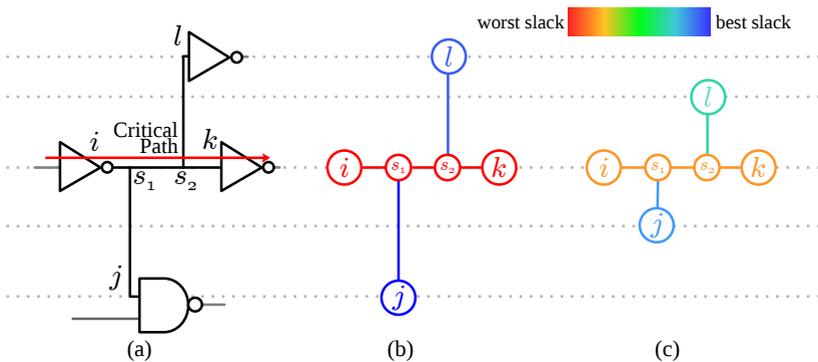


Figura 21 – (a) uma interconexão pertencente ao caminho crítico do circuito; (b) os *slacks* de cada ponto de Steiner da interconexão; (c) os *slacks* após a redução dos segmentos $s_1 \rightarrow j$ e $s_2 \rightarrow l$.

Limitações: Os autores legalizam o circuito como um **passo final**, após a otimização global de atraso. Este passo de legalização pode degradar muito o atraso do circuito ou até invalidar otimizações realizadas em passos anteriores no fluxo de projeto. Apesar da técnica

se basear no modelo de comprimento de fio de árvore de Steiner (*StWL: Steiner tree wirelength*) para assinalar os pesos, o **modelo Clique** – que superestima o comprimento das interconexões – é utilizado na formulação quadrática do problema, por questões de diferenciabilidade.

Viswanathan et al. (2010) apresentaram o algoritmo *ITOP*, uma abordagem para o problema de *ITDP* que realiza movimentos globais e otimizações incrementais a fim de otimizar o atraso do circuito. A técnica legaliza o circuito a cada iteração do *ITOP* que compreende três passos principais: **1) otimização do caminho crítico:** aplica pesos nas interconexões com base nos *slacks* e reposiciona as células através de buscas locais. As interconexões são modeladas com o modelo *Star*. As arestas *Star* referentes a células críticas possuem peso 10. As arestas *Star* que conectam células críticas a células não críticas possuem peso 5. As demais arestas *Star* possuem peso 1. A busca local de uma célula é realizada avaliando o potencial de oito movimentos em volta de sua posição atual. Enfim, o movimento com melhor potencial é realizado. O potencial de um movimento é a redução do comprimento (quadrático) ponderado dos ramos *Star*. Uma estratégia de tunelamento que relaxa a restrição de deslocamento máximo considera os macroblocos durante a otimização. **2) mitigação do congestionamento:** o congestionamento do circuito é reduzido movendo células não críticas de áreas congestionadas para áreas não congestionadas; **3) compressão do histograma de *slack*:** Um passo de otimização global é realizado com o objetivo de melhorar o *slack* do circuito. Este passo envolve inserir *buffers* e aplicar *gate sizing* em um grande conjunto de caminhos no circuito. Os autores justificam a realização desse passo afirmando que o movimento das células críticas do circuito, ao longo das iterações, pode impactar negativamente o atraso de caminhos não críticos. Os autores concluem também que a realização da compressão do histograma de *slack* também melhora a convergência do algoritmo.

Limitações: O uso do modelo *Star* superestima o comprimento das interconexões. Além disso, os únicos movimentos que são realizados visando otimização de atraso do circuito são aqueles feitos nas células críticas, ao passo que o movimento das não críticas só tem por objetivo impactar no perfil de densidade do circuito. Apesar disto, os autores modelam o impacto das células não críticas sobre o atraso do circuito aumentando o peso de seus ramos *Star*, porém esta estratégia pode não ser suficiente, uma vez que **os receptores não críticos não são aproximados do *driver* crítico**. A estratégia de tunelamento adotada relaxa a restrição de deslocamento máximo, sendo impraticável em um estágio avançado do fluxo de projeto, cuja restrição se torna

mais apertada.

3.3 TRABALHOS QUE UTILIZAM *HPWL*

Choi e Bazargan (2003) propuseram uma técnica de *ITDP* baseada na aplicação de restrições para o *HPWL* das interconexões de maneira analítica, a fim de reduzir o atraso do circuito. Para que o impacto causado no posicionamento seja mínimo, a técnica é focada apenas nas interconexões dos caminhos críticos do circuito.

A Figura 22 apresenta a visão geral das etapas do algoritmo de Choi e Bazargan (2003), que opera sobre um circuito já posicionado e realiza, iterativamente, três passos: Figura 22 (a): **Identificação** das interconexões do caminho crítico; Figura 22 (b): **Restrição** de *HPWL* e contração das interconexões do caminho crítico; Figura 22 (c): **Legalização** do circuito. A Figura 22 (d) mostra que ao fim da iteração, um novo posicionamento com atrasos reduzidos é obtido.

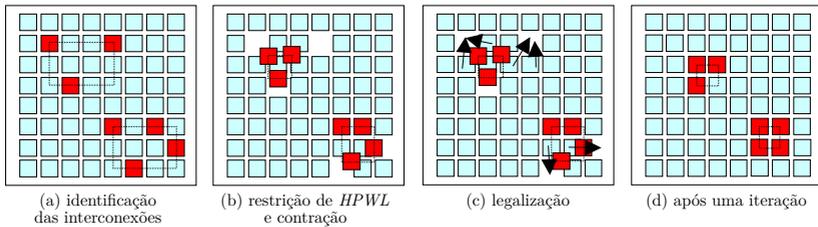


Figura 22 – Passos do algoritmo de Choi e Bazargan (2003). Fonte: (CHOI; BAZARGAN, 2003, p. 464). Adaptada pelo autor.

Os autores propõem uma formulação baseada em programação linear (*LP: linear programming*) para realizar a contração das interconexões. Primeiramente, os atrasos dos caminhos críticos do circuito são modelados em função da variação das componentes x e y do *HPWL* de suas interconexões. O objetivo é minimizar o somatório das variações em x e y dos *HPWL* das interconexões dos caminhos críticos (Equação 3.1). As restrições do problema (Equação 3.2) certificam que dado um atraso alvo (d_{it}) e o atraso atual de cada caminho $path_j$ (d_j), a redução de atraso nas interconexões (estimados pelo *HPWL*) destes caminhos deve ser positiva. Para incorporar o *HPWL* na função objetivo linear, os autores consideram as componentes x e y separadamente. ΔB_{ijx} é a variação no *HPWL* em respeito à componente x da interconexão j , pertencente ao caminho i . Semelhantemente, ΔB_{ijy} se refere à compo-

nente y do *HPWL* da interconexão.

$$\text{Minimize } \sum_{i,j} (\Delta B_{ijx} + \Delta B_{ijy}) * \quad (3.1)$$

$$\text{Sujeito a } path_i : \sum_{j \in i} \left(\frac{\partial D}{\partial B_{ijx}} \Delta B_{ijx} + \frac{\partial D}{\partial B_{ijy}} \Delta B_{ijy} \right) \geq (d_i - d_{it}) \quad (3.2)$$

As interconexões que participam de mais caminhos aparecerão em mais restrições, porém, cada interconexão só aparecerá uma vez na função objetivo (modificação indicada por *). Assim, as interconexões que participam de mais caminhos receberão maior redução que as outras.

Assumindo os novos limites para os *HPWL* das interconexões dos caminhos críticos, as células que se conectam a essas interconexões são reposicionadas, para que satisfaçam a essa nova restrição. Um programa quadrático formula o segundo problema que é de reposicionar as células, causando o menor deslocamento possível, para que elas respeitem a restrição de *HPWL* imposta pelo programa linear.

A contração das interconexões pode gerar algumas sobreposições que precisam ser removidas para que o posicionamento seja considerado legal. Os autores utilizam uma técnica de particionamento, que intercala bisseções verticais e horizontais para distribuir as células pela área do chip, removendo as sobreposições remanescentes do passo anterior.

Limitações da técnica: A técnica se baseia em **reduzir o *HPWL*** das interconexões críticas, mas conforme apresentado no capítulo anterior, isto não é suficiente para encurtar o comprimento real dessas interconexões. Além disso, **os ramos não críticos das interconexões críticas não são explorados** individualmente na otimização do atraso do circuito. Finalmente, a **legalização é realizada em um passo final**, posterior à otimização de atraso, o que pode causar severa degradação de atraso.

Chowdhary et al. (2005) propuseram um modelo mais preciso para o atraso dos circuitos. Tal modelagem incorpora a um programa linear não só os atrasos das interconexões, mas também os atrasos das células, os *slews* e a propagação dos tempos de chegada. Os atrasos das células são funções lineares dos *slews* de entrada e das capacitâncias de saída. Estas são aproximadas como um valor proporcional ao *HPWL* das interconexões. As restrições do programa linear modelam a variação dos atrasos e *slews* e não seus valores diretamente. Os autores

afirmam que esta abordagem de análise de *timing* diferencial é mais precisa, pois é baseada em valores iniciais precisos reportados por uma ferramenta de *STA*. A técnica move apenas as células críticas e seus *fanouts* para reduzir o número de sobreposições (e consequentemente manter a qualidade do posicionamento inicial) e legaliza o circuito após cada iteração do algoritmo que resolve a formulação linear.

Limitações: os próprios autores apontam o **uso do HPWL** como limitação já que este modelo pode apresentar correlação ruim com o roteamento final. Outra limitação é a aplicação da **legalização como passo final** de otimização, que pode degradar as estimativas de atraso.

Ren, Pan e Kung (2005) propuseram métricas de sensibilidade para o ponderamento das interconexões. Inicialmente, os autores apresentam uma métrica para estimar a sensibilidade do comprimento de uma interconexão a uma pequena variação em seu peso $S_W^L(i)$ (Equação 3.3):

$$S_W^L(i) = -L(i) \frac{W_{src}(i) + W_{sink}(i) - 2W(i)}{W_{src}(i)W_{sink}(i)} \quad (3.3)$$

Sendo $L(i)$ a distância entre o fonte e o destino da interconexão. W_{src} é o somatório dos pesos referentes a célula fonte e W_{sink} é o somatório dos pesos da célula destino. $W(i)$ é o peso da própria interconexão.

Em seguida, os autores formulam a sensibilidade do atraso ao comprimento de interconexão $S_L^T(i)$, utilizando o atraso de Elmore (1948) (Equação 3.4):

$$S_L^T(i) = rcL(i) + cR_d + rC_l \quad (3.4)$$

Onde r e c são constantes que determinam os valores de resistência e capacitância por unidade, em uma dada tecnologia. R_d e C_l são a resistência e a carga capacitiva do *driver*, respectivamente.

Enfim, os autores derivam a sensibilidade do *slack* ao peso $S_W^{Slk}(i)$ (Equação 3.5):

$$S_W^{Slk}(i) = -S_L^T(i)S_W^L(i) \quad (3.5)$$

A partir destas três funções de sensibilidade, os autores propõem uma função de sensibilidade do total *slack* negativo (*TNS: total*

negative slack) para o peso S_W^{TNS} (Equação 3.6). Para tal, eles definem outra sensibilidade, a do *TNS* para o atraso de uma interconexão $S_T^{TNS}(i)$ que é igual a menos o número de *POs* críticas influenciadas pela interconexão i .

$$S_W^{TNS} = S_T^{TNS}(i)S_L^T(i)S_W^L(i) \quad (3.6)$$

Os pesos das interconexões são calculados por meio de relaxação lagrangiana (*LR: lagrangian relaxation*). Os pesos são então incorporados a uma ferramenta industrial de *TDP* (VILLARRUBIA et al., 1989).

Limitação: Uso do modelo *HPWL*. Além disso, os autores realizam **legalização no fim da otimização**, o que pode causar severas degradações no desempenho.

Luo, Newmark e Pan (2006) propõem uma técnica de *TDP* baseada em *LP*. Os autores ressaltam a importância de se considerar a propagação dos atrasos e *slews* pelos caminhos dos circuitos, e que esse fenômeno não é levado em consideração em muitas técnicas de posicionamento. Para representar essa dependência topológica que o atraso de uma célula possui, sua estimativa considera a propagação do *slew* por dois estágios, mantendo um bom compromisso entre precisão e tempo de computação.

A técnica de Luo, Newmark e Pan (2006) assinala pesos às interconexões com base nas suas sensibilidades em relação ao atraso dos caminhos. A sensibilidade de uma interconexão e_j (S_j) é o impacto que ela causa no atraso do caminho (D_j) devido a uma mudança em seu comprimento (L_j):

$$S_j = \frac{\partial D_j}{\partial L_j} \quad (3.7)$$

A Figura 23 apresenta as parcelas que compõem o impacto da interconexão e_j no atraso do circuito. A célula n_i é *driver* da interconexão e_j . A interconexão tem como destino a célula n_{i+1} . O impacto causado pela interconexão e_j no atraso do caminho é composto por três termos:

$$D_j = Dg_i + De_j + Dg_{i+1} \quad (3.8)$$

- 1) ΔDg_i é a variação no atraso do *driver* (porta n_i) causada

pela variação de capacitância na interconexão e_j ; 2) ΔDe_j é a variação no próprio atraso da interconexão, calculado pelo atraso de Elmore (ELMORE, 1948); 3) ΔDg_{i+1} é o impacto causado pela propagação do *slew* no atraso do destino (porta n_{i+1}) da interconexão.

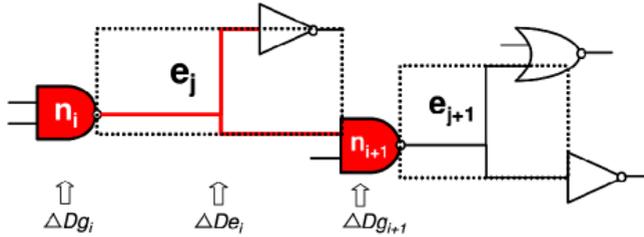


Figura 23 – Parcelas que compõem o impacto da interconexão e_j no atraso do circuito. Fonte: (LUO; NEWMARK; PAN, 2006).

Os autores formulam o problema utilizando *LP*. A técnica é aplicada nas células pertencentes à **rede de criticidade** (*criticality adjacency network*) que compreende as células e interconexões do caminho crítico e as células e interconexões conectadas com células críticas. Considerando um número fixo de caminhos críticos, as interconexões da rede de criticidade recebem restrições em seus *HPWL* com base em suas sensibilidades. As interconexões com maior peso terão seus *HPWL* reduzidos e as com maior peso serão expandidas.

Limitações da técnica: Semelhantemente a outras técnicas aqui apresentadas, os autores utilizam apenas os *HPWL* e os *slacks* para a estimativa de pesos das interconexões devido à simplicidade de incorporar tais técnicas à formulação de *LP*. Conforme comentado anteriormente, o *slack* e *HPWL* podem não ser suficientes para determinar quais interconexões devem ser reduzidas a fim de melhorar o atraso do circuito, já que **negligenciam o impacto das células não críticas** sobre o atraso das críticas.

Ren et al. (2007) movem as células críticas legalizando cada movimento. Os autores modelam o atraso do circuito em função do *HPWL* das interconexões críticas. O objetivo da otimização é de reduzir o comprimento das interconexões (*HPWL*), ponderando cada segmento de interconexão. Os pesos são assinalados em função do *slack* nas interconexões. **Limitação:** Assinalar os pesos com base apenas nos *slacks* não é suficiente para a otimização do atraso crítico. Tal estratégia **negligencia o impacto dos ramos não críticos** das interconexões críticas no atraso do circuito.

Papa et al. (2008) propuseram um modelo linear de atraso para interconexões *bufferizadas*. Os autores modelam o problema com *LP* incorporando uma restrição para impedir que caminhos não críticos tornem-se críticos durante a otimização. Eles estimam o valor de capacitância das interconexões do circuito com base em seus *HPWL* e legalizam cada movimento. **Limitação:** A técnica é aplicada para mover apenas o *latches* críticos. Além disso, os autores se **baseiam no *HPWL* das interconexões para estimar seus atrasos**. Isto leva a técnica a **negligenciar o posicionamento das células não críticas** conectadas às interconexões críticas.

3.4 TRABALHOS QUE UTILIZAM ÁRVORES DE STEINER

Ajami e Pedram (2001) movem as células críticas e os pontos de Steiner no contexto pós-roteamento. Os autores modelam o problema como um problema de otimização que maximiza o pior slack negativo (*WNS: worst negative slack*) incorporando o *timing* do circuito nas restrições. Os autores utilizam o modelo *StWL* para obter a capacitância das interconexões, porém, baseiam-se no *HPWL* para estimar as resistências. Além disso, os autores não mencionam qual abordagem de legalização é adotada. **Limitação:** Os resultados experimentais mostram que a técnica não possui boa escalabilidade, pois leva aproximadamente 1h27m para otimizar um circuito com apenas 6417 células.

3.5 RESUMO

Os pontos levantados nesta revisão são os mais relevantes no escopo do presente trabalho de mestrado. As limitações dos trabalhos apresentados mostram que existe espaço para novas técnicas de *ITDP* que explorem o potencial do modelo *StWL*. Este trabalho portanto tenta contornar as limitações das técnicas de *ITDP* modelando as interconexões de tal maneira. Isto não torna a otimização inviável, uma vez que a técnica se concentra em mover apenas as células não críticas conectadas ao caminho crítico, mitigando o custo de se utilizar um modelo preciso para as interconexões. A técnica proposta é apresentada no próximo capítulo.

4 OTIMIZAÇÃO DE ATRASO PÓS-POSICIONAMENTO EXPLORANDO RAMOS DE STEINER NÃO-CRÍTICOS

Este trabalho visa otimizar o atraso dos circuitos reduzindo a carga capacitiva das interconexões pertencentes ao(s) caminho(s) crítico(s), mediante o reposicionamento das células não críticas. A solução proposta tem como uma de suas características fundamentais o uso de árvores de Steiner (*StWL: Steiner tree wirelength*) como modelo de interconexões, com o intuito de aumentar a eficácia da otimização. Assim, a técnica apresentada neste capítulo baseia-se na exploração dos ramos de Steiner não críticos das interconexões dos caminhos críticos.

Este capítulo é organizado como segue. Primeiramente, a Seção 4.1 apresenta a visão geral da técnica proposta para encurtar os ramos de Steiner não críticos. Em seguida, a Seção 4.2 apresenta o cálculo para avaliação do potencial de otimização de cada ramo de Steiner. O algoritmo proposto é detalhado na Seção 4.3 e é seguido pela análise de sua complexidade na Seção 4.4. Finalmente, a Seção 4.5 apresenta uma modificação da técnica original, para que esta seja aplicada na redução de violações *early*.

4.1 VISÃO GERAL DA TÉCNICA PROPOSTA

Considere o circuito da Figura 24 (a). A saída primária (*PO: primary output*) X é a que possui o pior *slack* ($-10ps$), e portanto, $WNS = -10ps$. O caminho crítico associado a esta *PO* (apresentado pelas linhas vermelhas) parte da entrada primária D , segue pelas células i , j , k e l e termina na *PO* X . As ramificações do caminho crítico nas saídas das células i e j , por levarem a saídas primárias com *slack* positivo, recebem o nome de **ramos de Steiner não críticos**, sendo representados na Figura 24 (a) pelas linhas verdes.

Os ramos de Steiner não críticos impactam a carga capacitiva das células do caminho crítico, influenciando seus atrasos. A Figura 24 (b) mostra que se as células não críticas m , n e o forem aproximadas de seus pontos de ramificação, o *slack* do caminho crítico melhora de $-10ps$ para $-5ps$ sem comprometer o *slack* positivo das demais *POs*. Esta é a ideia central da técnica proposta neste trabalho.

A fim de se avaliar o espaço de otimização a ser explorado pela técnica, realizou-se um experimento para determinar a proporção de

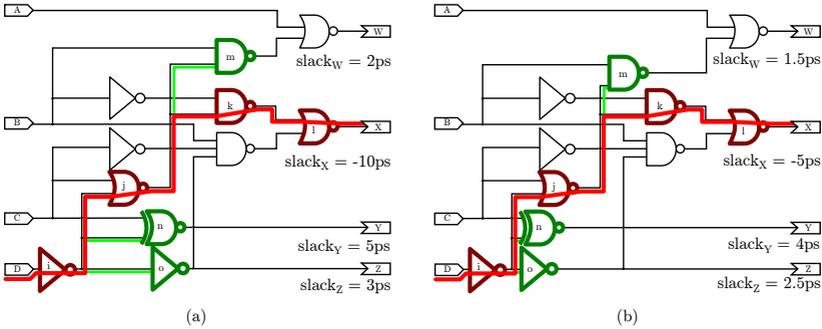


Figura 24 – Visão geral da técnica. O *slack* na saída primária crítica é melhorado movimentando-se apenas células não críticas.

capacitância não crítica nas interconexões dos caminhos críticos dos circuitos do *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014). A Figura 25 apresenta os resultados deste experimento. Para cada circuito da infraestrutura, os gráficos de barra mostram no eixo x as interconexões do caminho crítico e no eixo y , para cada interconexão, a proporção de capacitância referente a ramos de Steiner não críticos em relação à capacitância total da interconexão. É notável que para muitas interconexões, a proporção é muito maior que 30%, atingindo quase 100% em alguns casos (*e.g.*, circuitos *mgc_edit_dist* e *mgc_matrix_mult*). Isto indica que é possível otimizar o atraso destes circuitos reduzindo a carga capacitiva do caminho crítico, através do reposicionamento das células não críticas conectadas a este caminho.

4.2 AVALIAÇÃO DO POTENCIAL DE OTIMIZAÇÃO DOS RAMOS DE STEINER NÃO-CRÍTICOS

Certos ramos de Steiner podem ter maior potencial para reduzir o atraso do caminho crítico do que outros. Por causa disso, a técnica proposta analisa o potencial de otimização de cada um dos ramos de Steiner não críticos separadamente. Suponha que se deseje avaliar o potencial (decremento de $\delta_{i,k}$) de encurtar o ramo Steiner não crítico associado ao pino j da Figura 14, de l_j para l'_j . Para isso, calcula-se tal potencial como a diferença entre o atraso de Elmore considerando o comprimento l_j e o atraso de Elmore quando o comprimento é l'_j , como apresentado na Equação 4.1. A Equação 4.2 generaliza o cálculo do potencial quando qualquer ramo Steiner não crítico é encurtado, como

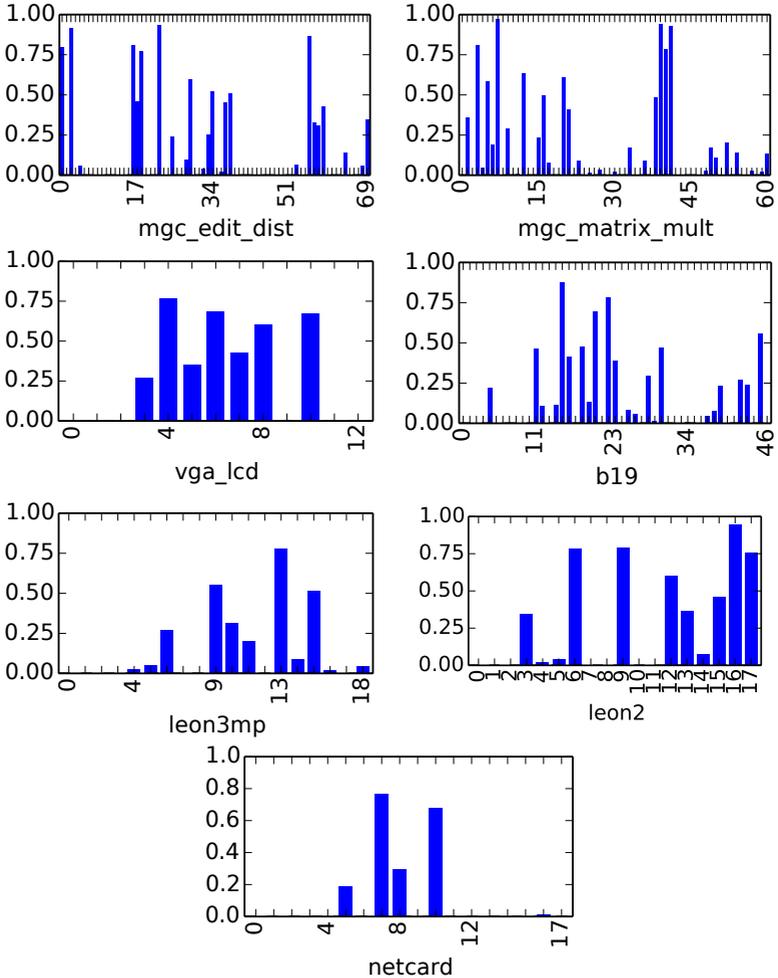


Figura 25 – Proporção de capacidade referente a ramos de Steiner não críticos nas interconexões dos caminhos críticos dos circuitos do *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014).

apresentado em (OBERMEIER; JOHANNES, 2004).

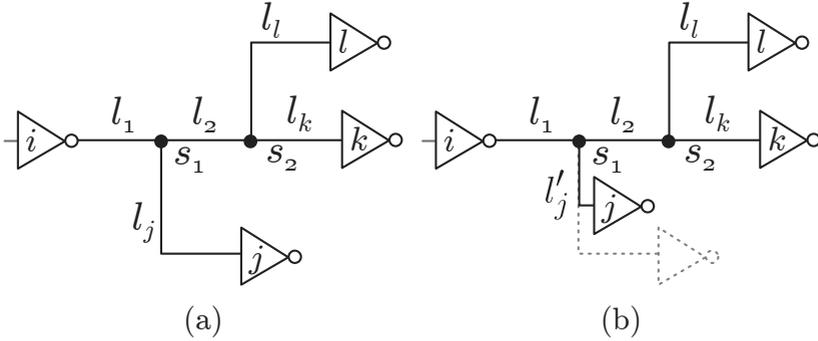


Figura 26 – (a) Uma interconexão que conecta 4 células modelada como uma árvore de Steiner. (b) A redução do ramo associado à célula j é capaz de reduzir a carga capacitiva da interconexão. O potencial de redução deste ramo do comprimento l_j para o comprimento l'_j pode ser estimado utilizando a Equação 4.1.

$$\delta_{(i,k)} - \delta_{(i,k)|_{(l_j=l'_j)}} = (R_i + r \cdot l_1) \cdot (c \cdot (l_j - l'_j)) \quad (4.1)$$

$$\delta_{(i,k)} - \delta_{(i,k)|_{(l_j=l'_j)}} = (R_i + r \cdot \sum_{\substack{m \in \{i \dots j\} \\ \cap \{i \dots k\}}} l_m) \cdot (c \cdot (l_j - l'_j)) \quad (4.2)$$

Para o exemplo da Figura 14, a situação ideal seria quando os segmentos relacionados aos pinos não críticos (j e l) fossem encurtados totalmente. Em outras palavras, quando as células j e l fossem posicionadas sobre os pontos de Steiner s_1 e s_2 , respectivamente. Porém, estes pontos de Steiner podem estar fora da área legal (restrição de deslocamento máximo) das células j e l , ou ainda, podem estar bloqueados por macroblocos fixos. A técnica proposta, detalhada na próxima seção, considera os dois problemas mencionados enquanto encurta os ramos de Steiner não críticos.

4.3 ALGORITMO PARA EXPLORAÇÃO DOS RAMOS DE STEINER NÃO-CRÍTICOS

O Algoritmo 2 apresenta a técnica proposta para encurtar ramos de Steiner não críticos. O *loop* da Linha 1 até a Linha 6 chama as

funções de alta hierarquia. Primeiramente, a Linha 2 atualiza os *slacks* do circuito através da análise de *timing* estática (*STA: static timing analysis*). A Linha 3 seleciona as células candidatas para encurtar os ramos de Steiner não críticos. Esta função está descrita na Seção 4.3.1. A Linha 4 ordena as células candidatas com base em seus potenciais para reduzir o atraso do caminho crítico. A função de ordenamento está descrita na Seção 4.3.2. Finalmente, a Linha 5 move as células para suas posições finais. Tal função está descrita na Seção 4.3.3.

Algoritmo 2: Shorten Non-Critical Steiner Tree Branches

```

1 for  $i \leftarrow 1$  to MAX_ITERATIONS do
2   static_timing_analysis();
3   candidates  $\leftarrow$  SELECT_CANDIDATE_CELLS();
4   movements  $\leftarrow$  RANK_CANDIDATE_CELLS(candidates);
5   MOVE_CANDIDATE_CELLS(movements);
6 end
7 return best solution over all iterations;
```

4.3.1 Seleção dos Candidatos

A função `SELECT_CANDIDATE_CELLS` (Algoritmo 3) inicialmente ordena as *POs* de acordo com os seus *slacks* negativos e obtém as k^1 piores *POs*. Em seguida, a Linha 3 até a Linha 12 iteram por cada aresta no caminho de pior *slack* pertencente a cada uma das *POs* escolhidas, selecionando os candidatos através do *loop* da Linha 7 até a Linha 9. As células selecionadas são os destinos das interconexões iteradas no *loop* externo ². Apenas as células com *slack* positivo são selecionadas para que a degradação no atraso destas, decorrente do movimento, não piore o *WNS* ou *TNS* do circuito. Note que um candidato é uma relação entre uma célula e uma interconexão, isto é, uma célula pode ser candidata a otimizar o atraso de mais de uma interconexão, já que mais de um caminho crítico pode passar por esta porta. Por fim, a Linha 13 retorna o conjunto de candidatos selecionados.

¹O valor k é uma constante determinada pelo usuário. Quanto maior o valor de k , mais saídas primárias são consideradas e portanto, a otimização pode levar a maiores reduções de total *slack* negativo (*TNS: total negative slack*).

²Devido ao fato de que o movimento de elementos sequenciais pode alterar drasticamente a estimativa da árvore de relógio e conseqüentemente, alterar a restrição de desempenho, somente elementos combinacionais são selecionados no algoritmo.

Algoritmo 3: SELECT_CANDIDATE_CELLS()

```

1  $PO \leftarrow k$  primary outputs with worst negative slacks;
2  $candidates \leftarrow \emptyset$ ;
3 foreach  $po \in PO$  do
4    $pin \leftarrow po$ ;
5   while  $pin \neq primary\_input$  do
6      $net \leftarrow$  interconnection connected to the input of  $pin$ ;
7     foreach  $cell \in \{i \mid i \in receiver(net) \wedge slack(i) > 0\}$ 
8       do
9          $\mid$  insert  $(cell, net)$  in  $candidates$ ;
10      end
11      $pin \leftarrow$  pin with worst slack of  $driver(net)$ ;
12   end
13 return  $candidates$ ;

```

4.3.2 Ordenamento dos Candidatos

A função `RANK_CANDIDATE_CELLS`, apresentada no Algoritmo 4, tem como objetivo identificar o potencial de cada movimento candidato e classificá-lo de acordo com uma função de sensibilidade. Para cada candidato $(cell, net)$, as Linhas 4 até 13 visitam cada ponto de Steiner da respectiva interconexão ($steiner_point(net)$) e calculam seu potencial de otimização e sensibilidade. Os pontos de Steiner fora da área legal são mapeados para os limites da respectiva área nas Linhas 5 até 8. A Linha 9 calcula o potencial de otimização quando a célula candidata é movida para o ponto alvo. Para obter melhores estimativas, a técnica posiciona a célula e gera uma árvore de Steiner para posição alvo. A Linha 11 calcula a sensibilidade (relação entre potencial e deslocamento) para identificar quais são as células que possuem maior redução de atraso com o menor movimento. Finalmente, na Linha 16, a função retorna um conjunto de movimentos ordenados pelas suas sensibilidades.

4.3.3 Movimentação dos Candidatos

A função `MOVE_CANDIDATE_CELLS` é apresentada no Algoritmo 5. Primeiramente, o movimento com melhor sensibilidade é removido do

Algoritmo 4: RANK_CANDIDATE_CELLS(*candidates*)

```

1  movements  $\leftarrow \emptyset$ ;
2  foreach (cell, net)  $\in$  candidates do
3    curr  $\leftarrow$  current location of cell;
4    foreach target  $\in$  steiner_point(net) do
5      if target is outside of legal area then
6        m  $\leftarrow$  line segment from curr to target;
7        target  $\leftarrow$  intersection of m with the boundaries
          of legal area;
8      end
9      pot  $\leftarrow$  potential of moving cell to target according to
        Equation (4.2);
10     dist  $\leftarrow$  Manhattan distance between curr and target;
11     sensitivity  $\leftarrow \frac{pot}{dist}$ ;
12     insert (sensitivity, target, cell) in movements;
13   end
14 end
15 quick_sort(movements) according to the highest
    sensitivity;
16 return movements;

```

conjunto na Linha 2. Em seguida, a Linha 3 posiciona a célula correspondente ao movimento na posição alvo. A Linha 4 remove quaisquer sobreposições causadas por este movimento, utilizando a técnica proposta por Chow et al. (2014).

Legalizar após cada movimento é vantajoso, pois é possível avaliar as consequências imediatamente e desfazer a operação, caso necessário. Além disso, esta estratégia ajuda a manter controladas as perturbações no atraso do circuito quando comparada a um passo único de legalização no fim da otimização.

Como outras células também podem ser candidatas a encurtar ramos de Steiner não críticos na mesma interconexão, a Linha 5 recomputa o potencial dessas e reinsere-as no conjunto de movimentos.

Algoritmo 5: MOVE_CANDIDATE_CELLS(*movements*)

```

1 while movements is not empty do
2   (sensitivity, target, cell) ← extract best movement from
   movements;
3   place cell in target;
4   legalize cell;
5   Re-rank every cell in the same interconnection;
6 end

```

4.4 ANÁLISE DA COMPLEXIDADE E DOS PARÂMETROS DO ALGORITMO

A complexidade do Algoritmo 2 é dominada por três fatores: número de iterações, a constante k (número de *POs* selecionadas para traçar os caminhos críticos) e número de pontos de Steiner alvo considerados. Determinou-se experimentalmente que o número de iterações pode ser considerado constante ($\text{MAX_ITERATIONS} = 5$), conforme apresentado no Apêndice B. O número de *POs* impacta no número de células candidatas a serem consideradas. Obviamente, considerar muitas *POs* amplia o espaço para a otimização. Por outro lado, o aumento no número de *POs* pode torná-la impraticável pelo aumento no tempo de execução. Nos experimentos realizados, observou-se que considerar somente 3 *POs* é o suficiente para obter significantes reduções, considerando *WNS* e *TNS*. O número de pontos de Steiner alvo em cada interconexão também tem impacto considerável no desempenho, devido ao fato de que a avaliação de cada alvo gera uma nova árvore de Steiner. Portanto, restringiu-se o cálculo de potencial aos 20 pontos mais longe da célula avaliada. Apesar desta estratégia não influenciar a maioria das interconexões (já que elas possuem menos de 20 pontos de Steiner), ela pode reduzir o tempo de execução drasticamente em circuitos que possuam interconexões muito longas.

4.5 MODIFICAÇÃO DA TÉCNICA PARA REDUÇÃO DE VIOLAÇÕES *EARLY*

A técnica original proposta tem como foco a redução das violações de atraso *late* por meio da redução da carga capacitiva dos caminhos críticos. Porém, é possível modificar a técnica proposta a fim

de reduzir violações de atraso *early*, que são relacionadas à restrição *hold* dos elementos sequenciais.

Nas técnicas de posicionamento baseado em forças (*force-directed placement*), as células e interconexões são modeladas de maneira análoga a um sistema de molas. Segundo tal modelo, as células são atraídas em direção aos seus *fanins* e *fanouts* com força de atração diretamente proporcional às distâncias. Eventualmente, as células são posicionadas em uma configuração que atinge o equilíbrio de forças. No posicionamento, o comprimento das interconexões é minimizado se todas as células forem posicionadas em seus pontos de equilíbrio. A posição de equilíbrio de uma célula i é chamada de *zero-force target (ZFT)* e corresponde ao centro de massa das interconexões de i (KAHNG, 2011). A Figura 27 apresenta a posição de equilíbrio para a célula i , conectada a quatro outras células, a, b, c e d .

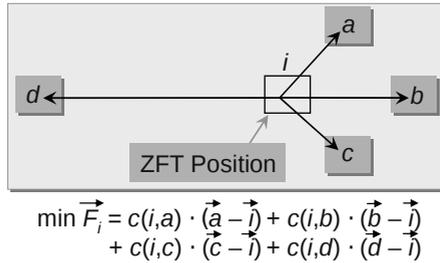


Figura 27 – *ZFT* para a célula i . O *ZFT* é a posição que minimiza a soma das forças entre i e cada uma das células que estão conectadas a esta (a, b, c e d). $c(u, v)$ corresponde ao peso (prioridade) da interconexão entre células u e v . Fonte: (KAHNG, 2011).

Para reduzir violações *early*, o objetivo de uma técnica de *TDP* passa a ser aumentar o comprimento de interconexões (ao invés de reduzir), uma vez que uma violação *early* é presente quando um caminho possui atraso menor que o estipulado pela restrição *hold* de um determinado *flip-flop*. Portanto, a abordagem proposta é de mover as células não críticas (no modo *early*) na direção oposta à posição do *ZFT*, a fim de aumentar o comprimento dos ramos não críticos dos caminhos críticos. Assim, espera-se que a carga capacitiva destes caminhos aumente, aumentando, conseqüentemente, seus atrasos.

A abordagem proposta para reduzir violações *early* modifica os passos de **seleção** e **ordenamento** dos candidatos. O Algoritmo 3 modificado percorre os caminhos críticos com base nos *slacks early* para

selecionar os candidatos. É importante que o movimento candidato não acrescente violações de atraso tanto no modo *early* quanto no modo *late*. Portanto, são selecionadas como candidatas as células que possuam *slack* não-negativo em ambos os modos.

O passo de ordenamento é modificado para avaliar movimentos em direção oposta ao *ZFT*, ao invés de avaliar movimentos em direção aos pontos de Steiner. O critério de ordenamento do conjunto de movimentos candidatos passa a ser o potencial de **aumento do atraso de Elmore (1948)**. Os demais passos do algoritmo continuam iguais aos da técnica original. A Figura 28 ilustra a abordagem utilizada para movimentar uma célula não crítica conectada a uma interconexão do caminho crítico *early*. O movimento da célula não crítica l é avaliado na direção oposta ao centro de massa, representado pelo retângulo tracejado rotulado de *target*.

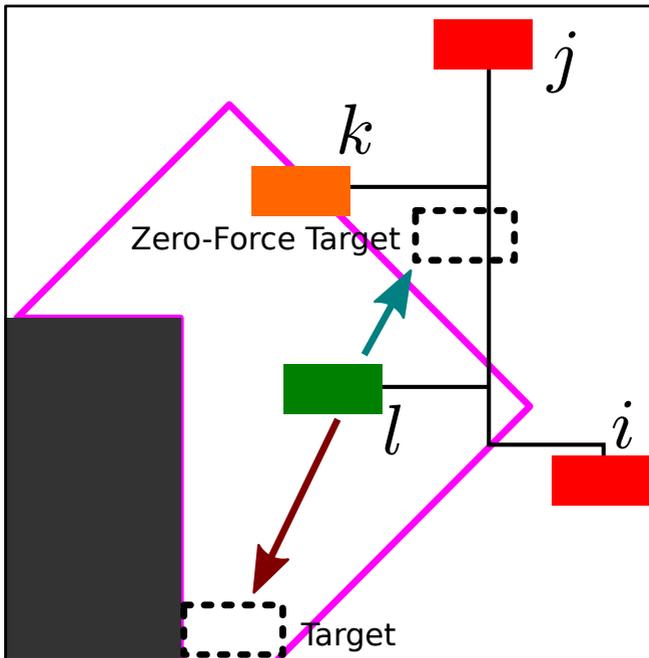


Figura 28 – Movimento da célula não crítica l em direção oposta ao *ZFT*. Em rosa está representada a restrição de deslocamento máximo. i , j e k são células críticas no modo *early*.

5 AVALIAÇÃO EXPERIMENTAL DA TÉCNICA PROPOSTA

A técnica proposta foi prototipada utilizando a linguagem C++ em conjunto com as bibliotecas QtCore e QtGUI (QT, 2015) para as funções relacionadas a geometria. O protótipo implementado faz uso de uma ferramenta própria de análise de *timing* estática (*STA*) e ambos (programa e ferramenta de *STA*) utilizam o algoritmo *FLUTE* (CHU; WONG, 2008) na computação das árvores de Steiner.

Os experimentos foram realizados aplicando o protótipo nos resultados de posicionamento obtidos pelas três primeiras equipes colocadas na competição *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014) em ambiente Linux com 12GB de memória RAM e duas CPUs Intel®Xeon®E5-5620 operando a 2.4GHz. A avaliação dos resultados foi realizada com a mesma métrica utilizada na competição.

Este capítulo está organizado em três Seções. A Seção 5.1 apresenta a infraestrutura utilizada nos experimentos. A Seção 5.2 apresenta os resultados obtidos, em termos de redução de *TNS* e *WNS late*. Finalmente, a Seção 5.3 apresenta os resultados obtidos pela técnica modificada (para violações *early*) nas soluções de posicionamento obtidas no primeiro experimento.

5.1 INFRAESTRUTURA E CONFIGURAÇÃO EXPERIMENTAL

A infraestrutura adotada consiste de 7 circuitos que possuem entre 131 mil e 959 mil células, um conjunto de *scripts* para validação e uma biblioteca *standard cell* (KIM; HU; VISWANATHAN, 2014). Para cada circuito a infraestrutura fornece um posicionamento inicial (gerado por um posicionador global) e restrições *short* e *long*, que limitam o máximo deslocamento das células em relação a suas posições iniciais. A competição teve por objetivo avaliar o potencial de técnicas de *ITDP* de reduzir o número das violações de atraso *late* e *early*, mantendo a legalidade dos circuitos. A avaliação dos resultados é feita a partir das métricas *WNS* e *TNS* e da métrica de densidade *average bin utilization (ABU)* (KIM; HU; VISWANATHAN, 2014).

Todos os experimentos selecionaram 3 *POs* para traçar os caminhos críticos e 5 iterações como critério de parada, uma vez que estes parâmetros foram julgados suficientes para obtenção dos melhores resultados em termos de redução das violações de atraso, conforme

apresentado no Apêndice B.

5.2 REDUÇÃO DAS VIOLAÇÕES *LATE*

A técnica aqui proposta é focada em reduzir as violações de atraso no modo *late*. Porém, foi observado que além de a técnica efetivamente reduzir as violações *late*, ela não aumenta as violações *early*. Por este motivo, os experimentos avaliados nesta seção só apresentam os valores para *WNS late* e *TNS late*.

A Tabela 2 apresenta os resultados para a restrição de deslocamento máximo *short*, a qual impõe uma menor área para deslocamento. Para cada circuito, essas tabelas mostram *WNS*, *TNS*, *ABU* e a métrica de qualidade obtida por cada um dos três primeiros colocados na competição (KIM; HU; VISWANATHAN, 2014). A técnica não foi aplicada nas soluções sem violações de atraso *late*. Tais casos tampouco foram considerados nos cálculos das reduções médias.

Na restrição *short* (mais apertada), a técnica obteve reduções médias de 23% e 34% para *WNS* e *TNS*, respectivamente. Ao observarmos a redução média em relação às soluções do vencedor da competição (cada110), nota-se que a técnica obteve 18.4% e 30.8% de redução em *WNS* e em *TNS*. Além disso, a técnica reduz as violações de atraso em 16 casos de um total de 18. O impacto no perfil de densidade, indicado pela coluna *ABU*, é bem pequeno. Como consequência, a qualidade, medida pela métrica da competição, é melhorada em 17%.

A Tabela 3 apresenta os resultados para a restrição de deslocamento *long*¹. Na restrição *long*, a técnica obteve reduções médias ainda maiores nas violações de atraso: 62% e 67% para *WNS* e *TNS*, respectivamente. Também é importante observar que a técnica proposta removeu todas as violações de atraso em 7 casos, e as reduziu nos demais. Similarmente aos experimentos na restrição *short*, a técnica causou pouco impacto no *ABU*. A métrica de qualidade foi melhorada em 23%.

A Tabela 4 reporta o tempo de execução, em segundos, para se otimizar as soluções de cada uma das três equipes primeiras colocadas na competição. Observe que o tempo de execução para otimizar os circuitos menores (até 219 mil células) é relativamente pequeno (menos de 30 segundos). Devido às características de roteamento do *vga_1cd*, a técnica apresentou um tempo de execução maior (aproximadamente

¹Na restrição *long*, as áreas legais das células são aproximadamente 1000 vezes maiores quando comparadas às áreas legais na restrição *short*.

Tabela 2 – Resultados obtidos pela técnica proposta sobre as soluções dos três primeiros colocados da competição *CAD Contest @ ICCAD 2014* assumindo a restrição *short* (KIM; HU; VISWANATHAN, 2014).

circuito	solução	cada110 (primeiro colocado)				cada019 (segundo colocado)				cada075 (terceiro colocado)			
		WNS (s)	TNS (s)	ABU (e-2)	qualidade	WNS (s)	TNS (s)	ABU (e-2)	qualidade	WNS (s)	TNS (s)	ABU (e-2)	qualidade
edit_dist cells: 131k macros: 13	time	-7,0E-10	-3,4E-8	0,0	864,6	-5,3E-10	-1,5E-8	0,8	1.137,6	-5,6E-10	-2,7E-8	0,1	991,5
	prop.	-6,6E-10	-2,9E-8	0,0	943,7	-4,9E-10	-1,2E-8	0,8	1.192,0	-5,1E-10	-2,1E-8	0,2	1.087,9
	melhora	6,5%	14,2%	0,0%	9,2%	7,5%	19,1%	0,0%	4,8%	9,1%	23,1%	-0,1%	9,7%
matrix_mult cells: 155k	time	-4,0E-10	-1,7E-9	0,0	534,5	-2,2E-10	-1,0E-9	1,0	1.032,9	-3,0E-10	-1,4E-9	0,1	790,9
	prop.	-2,7E-10	-7,7E-10	0,0	1.039,3	-6,1E-11	-1,5E-10	1,0	1.533,1	-1,5E-10	-4,6E-10	0,2	1.331,4
	melhora	31,5%	55,2%	0,0%	94,5%	72,1%	85,1%	0,0%	48,4%	49,8%	68,1%	-0,1%	68,3%
vga_lcd cells: 165k macros: 0	time	-3,3E-10	-2,2E-9	2,2	1.454,5	-4,1E-10	-3,1E-9	2,2	1.394,6	-2,9E-10	-7,9E-10	0,9	1.492,0
	prop.	-2,7E-10	-1,7E-9	2,3	1.475,9	-3,3E-10	-2,4E-9	2,4	1.422,1	-2,2E-10	-3,7E-10	1,0	1.519,6
	melhora	18,4%	22,8%	-0,1%	1,5%	19,1%	21,7%	-0,2%	2,0%	25,6%	53,0%	-0,1%	1,9%
b19 cells: 219k macros: 0	time	-2,1E-10	-5,3E-10	2,6	1.546,3	-1,5E-10	-3,1E-10	3,5	1.580,8	-4,4E-10	-1,8E-9	0,2	1.373,4
	prop.	-1,7E-10	-3,7E-10	2,6	1.572,2	-6,9E-11	-1,3E-10	3,5	1.625,5	-3,8E-10	-1,4E-9	0,2	1.427,5
	melhora	17,2%	31,0%	0,0%	1,7%	53,3%	56,9%	0,0%	2,8%	13,5%	24,2%	0,0%	3,9%
leon3mp cells: 649k macros: 0	time	0	0	0,8	1.550,5	-3,6E-10	-7,7E-10	1,6	1.514,0	0	0	1,3	1.564,7
	prop.	0	0	0,8	1.550,5	-1,6E-10	-2,1E-10	1,6	1.526,5	0	0	1,3	1.564,7
	melhora	—	—	—	55,5%	73,2%	0,0%	0,8%	—	—	—	—	—
leon2 cells: 749k macros: 0	time	0	0	2,5	1.585,6	-8,7E-9	-4,7E-6	2,9	819,5	-4,4E-9	-5,3E-7	2,5	1.253,9
	prop.	0	0	2,5	1.585,6	-8,7E-9	-4,7E-6	2,9	819,5	-4,4E-9	-5,3E-7	2,5	1.253,9
	melhora	—	—	—	—	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
netcard cells: 959k macros: 12	time	0	0	1,1	1.568,4	-3,6E-9	-2,5E-8	2,4	1.279,0	0	0	1,3	1.561,1
	prop.	0	0	1,1	1.568,4	-3,4E-9	-2,3E-8	2,4	1.287,9	0	0	1,3	1.561,1
	melhora	—	—	—	—	3,9%	6,4%	0,0%	0,7%	—	—	—	—
melhora média		18,4%	30,8%	<0,1%	26,7%	30,2%	37,5%	<0,1%	8,5%	19,6%	33,7%	<0,1%	16,8%

Tabela 3 – Resultados obtidos pela técnica proposta sobre as soluções dos três primeiros colocados da competição *CAD Contest @ ICCAD 2014* assumindo a restrição *long* (KIN, HU, VISWANATHAN, 2014).

circuito	solução	cada110 (primeiro colocado)			cada019 (segundo colocado)			cada075 (terceiro colocado)					
		WNS (s)	TNS (s)	ABU (e-2)	qualidade	WNS (s)	TNS (s)	ABU (e-2)	qualidade	WNS (s)	TNS (s)	ABU (e-2)	qualidade
edit_dist cells: 131k macros: 13	time	-7,00E-10	-3,4E-8	0,0	864,6	-5,4E-10	-1,8E-8	1,5	1.113,1	-5,8E-10	-3,5E-8	0,1	907,0
	prop.	-5,3E-10	-1,7E-8	0,0	1.115,2	-4,3E-10	-1,0E-8	1,5	1.266,9	-5,4E-10	-3,1E-8	0,1	975,2
matrix_mult cells: 155k macros: 16	melhora	17,3%	49,1%	0,0%	29,0%	20,4%	45,2%	0,0%	13,8%	7,9%	10,9%	0,0%	7,5%
	time	0	0	0,0	1.641,8	-8,0E-11	-1,7E-10	0,9	1.520,6	-3,9E-10	-1,9E-9	0,2	415,2
vga_led cells: 165k macros: 0	prop.	0	0	0,0	1.641,8	0	0,9	1.674,5	-1,6E-10	-1,8E-10	0,3	1.339,2	
	melhora	—	—	—	—	100,0%	100,0%	0,0%	10,1%	59,4%	90,6%	-0,1%	222,5%
b19 cells: 219k macros: 0	time	-2,5E-10	-1,8E-9	3,0	1.473,8	-3,6E-10	-2,1E-9	3,4	1.403,3	-4,2E-10	-4,1E-9	0,2	1.413,6
	prop.	0	0	3,4	1.561,0	0	3,6	1.534,8	0	0	0	1,1	1.564,6
leon3mp cells: 649k macros: 0	melhora	100,0%	100,0%	-0,5%	5,9%	100,0%	100,0%	-0,3%	9,4%	100,0%	100,0%	-0,9%	10,7%
	time	-1,1E-10	-2,1E-10	2,7	1.608,6	0	0	3,5	1.661,3	-1,0E-10	-2,0E-10	2,8	1.615,0
leon2 cells: 749k macros: 0	prop.	0	0	2,7	1.668,3	0	0	3,5	1.661,3	0	0	2,8	1.672,2
	melhora	100,0%	100,0%	0,0%	3,7%	—	—	—	—	100,0%	100,0%	0,0%	3,5%
netcard cells: 959k macros: 12	time	0	0	0,8	1.550,5	-3,0E-10	-5,5E-10	1,9	1.512,6	0	0	1,3	1.564,7
	prop.	0	0	0,8	1.550,5	0	0	1,9	1.532,1	0	0	1,3	1.564,7
melhora média	time	0	0	2,5	1.585,6	-8,7E-9	-4,7E-6	3,2	823,0	-1,0E-8	-1,4E-5	3,1	266,9
	prop.	0	0	2,5	1.585,6	-7,3E-9	-4,4E-6	3,3	896,9	-7,7E-9	-1,4E-5	3,2	401,2
melhora	time	—	—	—	—	15,4%	5,9%	-0,2%	9,0%	22,5%	4,2%	-0,1%	50,3%
	prop.	0	0	1,1	1.568,4	-3,5E-9	-2,4E-8	2,5	1.279,3	-8,5E-10	-6,9E-9	1,0	1.464,9
melhora	time	0	0	1,1	1.568,4	-3,0E-9	-2,0E-8	2,6	1.311,4	-4,6E-10	-4,7E-9	1,2	1.489,5
	prop.	—	—	—	—	14,3%	18,8%	-0,1%	2,5%	45,8%	31,9%	-0,1%	1,7%
melhora média	time	72,4%	83,0%	-0,2%	12,9%	58,4%	61,7%	-0,1%	7,7%	55,9%	56,3%	-0,2%	49,4%
	prop.	—	—	—	—	—	—	—	—	—	—	—	—

7 minutos). Nos circuitos grandes, o maior tempo de execução foi para otimizar o circuito `leon2` da equipe `cada019`, levando aproximadamente 1,7 horas. Os tempos de execução observados nos experimentos realizados mostraram que é vantajoso aplicar a técnica proposta sobre qualquer ferramenta convencional de *TDP*.

As Figuras 29 (a) e 29 (b) mostram, respectivamente, *snapshots* antes e após a otimização de uma interconexão do caminho crítico de um circuito que integra o *benchmark* da competição *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014). As formas geométricas representam as células conectadas à interconexão. O caminho crítico é indicado pela seta vermelha. Os retângulos coloridos representam as demais células do circuito. A cor vermelha (azul) indica que uma célula possui *slack* negativo (positivo). A cor verde indica ausência de *slack* (*i.e.*, *slack* = 0). O retângulo cinza é um macrobloco.

Percebe-se que a otimização realmente encurta os ramos de Steiner não críticos, conforme observado nas posições finais das células conectadas a esta interconexão (Figura 29 (b)). Tal redução impacta no atraso da célula que alimenta esta interconexão. A otimização desta e das outras interconexões pertencentes ao caminho crítico contribuem com a redução do *WNS* de $-4,412353 \times 10^{-10}$ para $-3,32094 \times 10^{-10}$.

5.3 REDUÇÃO DAS VIOLAÇÕES *EARLY*

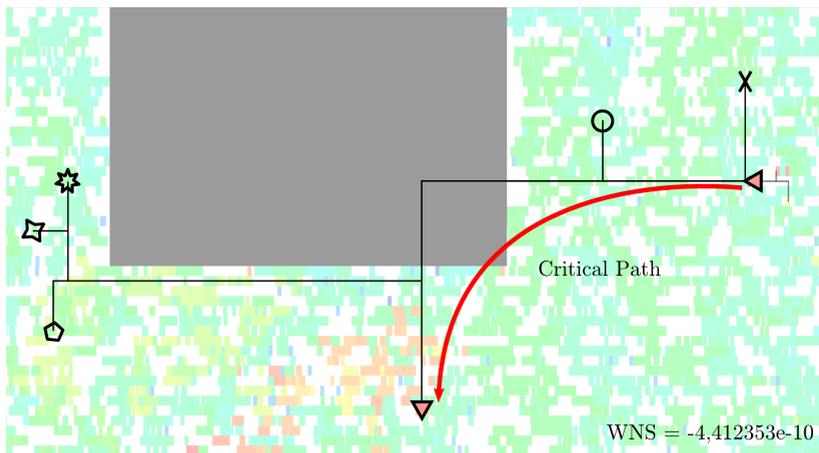
A técnica modificada para redução de violações *early* foi aplicada nas soluções obtidas pelos experimentos apresentados na seção anterior, com os mesmos parâmetros (3 saídas primárias e 5 iterações).

O gráfico de barras da Figura 30 apresenta os resultados em termos de redução percentual de *WNS early*, remanescentes do experimento anterior, na restrição *long*. Neste experimento, observou-se que os caminhos críticos *early* são bem menores (tanto em número de células, quanto em comprimento de interconexões) quando comparados com os caminhos críticos *late*. Isto explica o fato de que na maioria dos casos de teste, as reduções de *WNS* foram menores que 10%. No caso especial do circuito `leon3mp`, a técnica modificada não selecionou sequer candidatos, ou seja, as soluções iniciais (pós-experimentos da seção anterior) não possuem células não críticas no modo *early* conectadas à saída de células pertencentes a caminhos críticos.

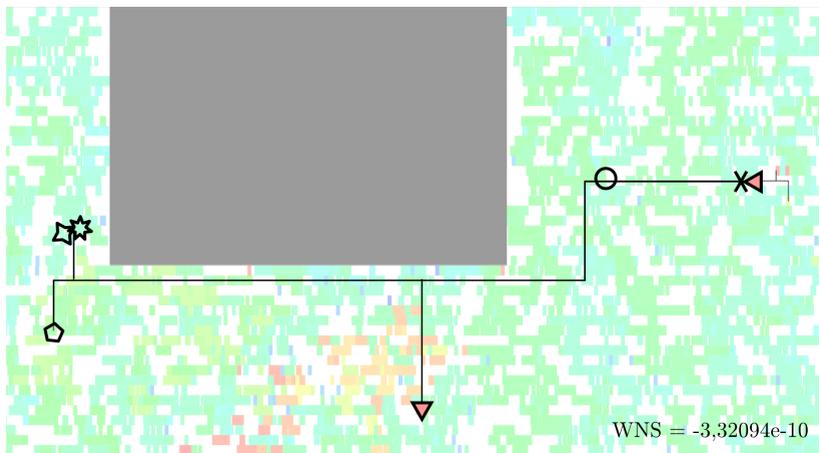
Apesar das limitações, a técnica modificada obteve reduções de *WNS* significativas nos circuitos `b19` e `mgc_matrix_mult` de 10% até aproximadamente 35%, respectivamente.

Tabela 4 – Tempo de execução em segundos para otimizar as soluções obtidas por cada um dos três primeiros colocados na competição *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014). Para cada circuito, o tempo de execução na restrição *short (long)* é apresentada à esquerda (direita).

solução	edit_dist		matrix_mult		vga_lcd		b19		leon3mp		leon2		netcard	
	short	long	short	long	short	long	short	long	short	long	short	long	short	long
cada110 (primeiro colocado)	8	10	9	9	—	91	90	15	9	—	—	—	—	—
cada019 (segundo colocado)	10	9	11	7	7	104	379	17	7	762	944	2.501	6.065	1.259
cada075 (terceiro colocado)	12	13	13	17	17	120	403	22	11	—	—	3.261	4.780	—
														5.724



(a)



(b)

Figura 29 – (a) Uma interconexão do caminho crítico do circuito `mgc_matrix_mult` (KIM; HU; VISWANATHAN, 2014) antes da otimização. (b) A mesma interconexão após a otimização.

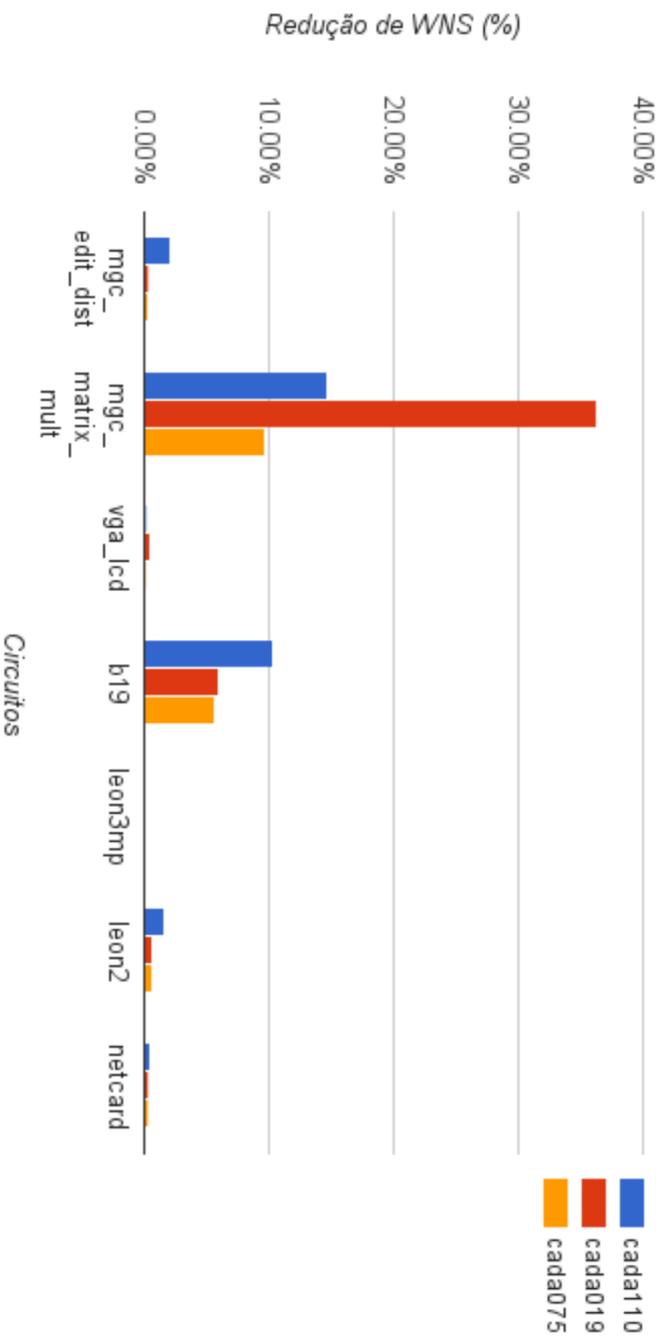


Figura 30 – Reduções percentuais de *WNS early* na restrição *long* partindo das soluções obtidas nos experimentos apresentados no Capítulo 4. No circuito *leon3mp* a técnica não conseguiu sequer selecionar candidatos.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho propôs uma técnica de posicionamento incremental guiado por atraso (*ITDP: incremental timing-driven placement*) baseada no modelo de comprimento de fio de árvore de Steiner (*StWL: Steiner tree wirelength*) para realizar o **encurtamento de ramos não críticos** das interconexões do caminho crítico *late*, reduzindo as violações de atraso.

A técnica proposta adota uma abordagem de legalização *on-the-fly*, isto é, removendo as sobreposições e realizando o alinhamento às bandas *standard cell* **durante a otimização**, mantendo o controle sobre as degradações de atraso, o que é esperado de uma técnica de *ITDP*.

A técnica proposta foi validada empiricamente mediante sua aplicação nos circuitos do *CAD Contest @ ICCAD 2014* (KIM; HU; VISWANATHAN, 2014). Para tanto foi desenvolvido em C++ um protótipo de ferramenta o qual tirou proveito das características da infraestrutura da referida competição, tais como formatos de arquivos (que são padrões industriais), *scripts* para validação e ferramenta de geração de árvores de Steiner (CHU; WONG, 2008).

Os resultados experimentais mostraram que, sob a restrição de deslocamento *short*, a técnica obteve reduções médias de 23% e 34% para as métricas pior *slack* negativo (*WNS: worst negative slack*) e total *slack* negativo (*TNS: total negative slack*), respectivamente. Comparando-se com as soluções do vencedor da competição, a técnica obteve 18,4% e 30,8% de redução para *WNS* e *TNS*, respectivamente. Além disso, a técnica foi capaz de reduzir as violações de atraso em 16 casos de um total de 18. Sob a restrição de deslocamento *long*, a técnica proporcionou reduções médias de 62% e 67% para *WNS* e *TNS*, respectivamente, tendo conseguido remover todas as violações de atraso em 7 casos, e reduzido as violações nos demais casos. Para ambas restrições de deslocamento, *short* e *long*, a aplicação da técnica não deteriorou a qualidade do posicionamento. Pelo contrário, a técnica resultou em melhora da métrica *average bin utilization (ABU)* em ambos casos.

A técnica é capaz de reduzir significativamente as violações de atraso com um tempo de execução relativamente baixo. Portanto a técnica é escalável, já que para otimizar os maiores circuitos (*leon3mp* com 649191 células, *leon2* com 794286 e *netcard* com 958792 células), a técnica levou no máximo 1,7 horas.

Esta dissertação também propôs uma modificação à técnica de

ITDP original, para ser utilizada na redução de violações *early*. Os experimentos realizados com a técnica modificada não apresentaram reduções significativas nas violações *early* devido ao fato de que os circuitos testados apresentam caminhos críticos *early* muito curtos e com poucas (ou nenhuma) ramificações não críticas.

Como perspectiva de trabalhos futuros, antevê-se a necessidade de aplicar a técnica proposta em um fluxo completo de projeto, com avaliação das métricas de violações de atraso com base em um roteamento realista (e não apenas por aproximações por meio de árvores de Steiner), considerando inclusive, circuitos menos artificiais como os da competição *CAD Contest @ ICCAD 2015* (KIM et al., 2015a), que contam com até aproximadamente 2 milhões de células, diversos macroblocos e células fixas. Na edição 2015 desta competição, a árvore de relógio foi dividida em diversas árvores de relógio locais, contornando a limitação do ano anterior de que a topologia da árvore de relógio era muito sensível a pequenos movimentos nas células sequenciais.

Outro possível trabalho futuro seria o desenvolvimento de métricas e heurísticas para tornar a seleção e ordenamento dos candidatos mais eficientes. Uma característica que pode ser utilizada para aprimorar a seleção dos candidatos é a capacidade de corrente dos seus *drivers* críticos. Os atrasos de células com baixa capacidade de corrente são mais sensíveis a variações em suas capacitâncias de saída. Portanto, esta característica pode ser incorporada a uma heurística de poda do espaço de busca de candidatos a encurtar ramos não críticos de árvores de Steiner. Na extensão para redução de violações *early*, é necessária a incorporação de restrições que controlem o aumento do *slew* no alongamento das interconexões.

Finalmente, através dos experimentos e da publicação realizados, pode-se concluir que este trabalho confirma a hipótese de que “o atraso crítico de um circuito pode ser significativamente reduzido mediante o reposicionamento das células não críticas”.

REFERÊNCIAS

AJAMI, A.; PEDRAM, M. Post-layout timing-driven cell placement using an accurate net length model with movable steiner points. In: *Design Automation Conference, 2001. Proceedings of the ASP-DAC 2001. Asia and South Pacific*. [S.l.: s.n.], 2001. p. 595–600.

ALPERT, C.; CHU, C.; VILLARRUBIA, P. The coming of age of physical synthesis. In: *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*. [S.l.: s.n.], 2007. p. 246–249. ISSN 1092-3152.

ALPERT, C. et al. Placement: Hot or not? In: *Computer-Aided Design (ICCAD), 2012 IEEE/ACM International Conference on*. [S.l.: s.n.], 2012. p. 283–290. ISSN 1092-3152.

ALPERT, C. J.; DEVGAN, A.; KASHYAP, C. A two moment rc delay metric for performance optimization. In: *Proceedings of the 2000 International Symposium on Physical Design*. New York, NY, USA: ACM, 2000. (ISPD '00), p. 69–74. ISBN 1-58113-191-7. <<http://doi.acm.org/10.1145/332357.332377>>.

BHASKER, J.; CHADHA, R. *Static timing analysis for nanometer designs: a practical approach*. [S.l.]: Springer Science & Business Media, 2009.

CHOI, W.; BAZARGAN, K. Incremental placement for timing optimization. In: *Computer Aided Design, 2003. ICCAD-2003. International Conference on*. [S.l.: s.n.], 2003. p. 463–466.

CHOW, W.-K. et al. Cell density-driven detailed placement with displacement constraint. In: *Proceedings of the 2014 International Symposium on Physical Design*. New York, NY, USA: ACM, 2014. (ISPD '14), p. 3–10. ISBN 978-1-4503-2592-9. <<http://doi.acm.org/10.1145/2560519.2560523>>.

CHOWDHARY, A. et al. How accurately can we model timing in a placement engine? In: *Proceedings of the 42Nd Annual Design Automation Conference*. New York, NY, USA: ACM, 2005. (DAC '05), p. 801–806. ISBN 1-59593-058-2. <<http://doi.acm.org/10.1145/1065579.1065792>>.

CHU, C.; WONG, Y.-C. Flute: Fast lookup table based rectilinear steiner minimal tree algorithm for vlsi design. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 27, n. 1, p. 70–83, Jan 2008. ISSN 0278-0070.

DOLL, K.; JOHANNES, F.; ANTREICH, K. Iterative placement improvement by network flow methods. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 13, n. 10, p. 1189–1200, Oct 1994. ISSN 0278-0070.

ELMORE, W. The transient response of damped linear networks with particular regard to wideband amplifiers. *Journal of applied physics*, AIP Publishing, v. 19, n. 1, p. 55–63, 1948.

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.

GUPTA, R.; TUTUIANU, B.; PILEGGI, L. The elmore delay as a bound for rc trees with generalized input signals. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 16, n. 1, p. 95–104, Jan 1997. ISSN 0278-0070.

GUTH, C. et al. Timing-driven placement based on dynamic net-weighting for efficient slack histogram compression. In: *Proceedings of the 2015 Symposium on International Symposium on Physical Design*. New York, NY, USA: ACM, 2015. (ISPD '15), p. 141–148. ISBN 978-1-4503-3399-3. <<http://doi.acm.org/10.1145/2717764.2717766>>.

HO, R.; MAI, K.; HOROWITZ, M. The future of wires. *Proceedings of the IEEE*, v. 89, n. 4, p. 490–504, Apr 2001. ISSN 0018-9219.

KAHNG, A. B. *VLSI physical design: from graph partitioning to timing closure*. [S.l.]: Springer Science & Business Media, 2011.

KASHYAP, C. et al. Closed-form expressions for extending step delay and slew metrics to ramp inputs for rc trees. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 23, n. 4, p. 509–516, April 2004. ISSN 0278-0070.

KASHYAP, C. V.; ALPERT, C. J.; DEVGAN, A. An "effective" capacitance based delay metric for rc interconnect. In: *Proceedings of the 2000 IEEE/ACM International Conference on Computer-aided Design*. Piscataway, NJ, USA: IEEE

Press, 2000. (ICCAD '00), p. 229–235. ISBN 0-7803-6448-1.
 <<http://dl.acm.org/citation.cfm?id=602902.602954>>.

KIM, M.-C. et al. Iccad-2015 cad contest in incremental timing-driven placement and benchmark suite. In: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. Piscataway, NJ, USA: IEEE Press, 2015. (ICCAD '15), p. 921–926. ISBN 978-1-4673-8389-9. <<http://dl.acm.org/citation.cfm?id=2840819.2840949>>.

KIM, M.-C. et al. *ICCAD-2015 CAD Contest in Incremental Timing-driven Placement and Benchmark Suite*. 2015. [Http://cad-contest.el.cycu.edu.tw/problem_C/default.html](http://cad-contest.el.cycu.edu.tw/problem_C/default.html).

KIM, M.-C.; HU, J.; VISWANATHAN, N. Iccad-2014 cad contest in incremental timing-driven placement and benchmark suite. In: IEEE PRESS. *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*. [S.l.], 2014. p. 361–366.

KOTECHA, P. et al. *Method of minimizing early-mode violations causing minimum impact to a chip design*. Google Patents, ago. 9 2011. US Patent 7,996,812. <<https://www.google.com/patents/US7996812>>.

LIVRAMENTO, V. et al. Exploiting non-critical steiner tree branches for post-placement timing optimization. In: *Computer-Aided Design (ICCAD), 2015 IEEE/ACM International Conference on*. [S.l.: s.n.], 2015. p. 528–535.

LUO, T.; NEWMARK, D.; PAN, D. Z. A new lp based incremental timing driven placement for high performance designs. In: *Proceedings of the 43rd Annual Design Automation Conference*. New York, NY, USA: ACM, 2006. (DAC '06), p. 1115–1120. ISBN 1-59593-381-6.
 <<http://doi.acm.org/10.1145/1146909.1147190>>.

OBERMEIER, B.; JOHANNES, F. M. Quadratic placement using an improved timing model. In: *Proceedings of the 41st Annual Design Automation Conference*. New York, NY, USA: ACM, 2004. (DAC '04), p. 705–710. ISBN 1-58113-828-8.
 <<http://doi.acm.org/10.1145/996566.996760>>.

PAPA, D. et al. Physical synthesis with clock-network optimization for large systems on chips. *Micro, IEEE*, v. 31, n. 4, p. 51–62, July 2011. ISSN 0272-1732.

PAPA, D. et al. Rumble: An incremental timing-driven physical-synthesis optimization algorithm. *Computer-Aided Design of*

Integrated Circuits and Systems, IEEE Transactions on, v. 27, n. 12, p. 2156–2168, Dec 2008. ISSN 0278-0070.

PAVISIC, I. et al. *Method in integrating clock tree synthesis and timing optimization for an integrated circuit design*. Google Patents, abr. 15 2003. US Patent 6,550,044. <<https://www.google.com/patents/US6550044>>.

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. <<http://doi.acm.org/10.1145/505306.505314>>.

QT. *Cross-platform application & UI development framework*. 2015. [Http://www.qt.io/](http://www.qt.io/).

RAJAGOPAL, K. et al. Timing driven force directed placement with physical net constraints. In: ACM. *Proceedings of the 2003 international symposium on Physical design*. [S.l.], 2003. p. 60–66.

REN, H.; PAN, D.; KUNG, D. Sensitivity guided net weighting for placement-driven synthesis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 24, n. 5, p. 711–721, May 2005. ISSN 0278-0070.

REN, H. et al. Hippocrates: First-do-no-harm detailed placement. In: *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*. Washington, DC, USA: IEEE Computer Society, 2007. (ASP-DAC '07), p. 141–146. ISBN 1-4244-0629-3. <<http://dx.doi.org/10.1109/ASPDAC.2007.357976>>.

SAPATNEKAR, S. *Timing*. [S.l.]: Springer Science & Business Media, 2004.

SAXENA, P. et al. Repeater scaling and its impact on cad. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 23, n. 4, p. 451–463, April 2004. ISSN 0278-0070.

VILLARRUBIA, P. et al. Ibm risc chip design methodology. In: *Computer Design: VLSI in Computers and Processors, 1989. ICCD '89. Proceedings., 1989 IEEE International Conference on*. [S.l.: s.n.], 1989. p. 143–147.

VISWANATHAN, N. et al. Itop: Integrating timing optimization within placement. In: *Proceedings of the 19th International Symposium on Physical Design*. New York, NY, USA: ACM, 2010. (ISPD '10), p. 83–90. ISBN 978-1-60558-920-6. <<http://doi.acm.org/10.1145/1735023.1735048>>.

WANG, Q. B.; LILLIS, J.; SANYAL, S. An lp-based methodology for improved timing-driven placement. In: *Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific*. [S.l.: s.n.], 2005. v. 2, p. 1139–1143 Vol. 2.

APÊNDICE A – Lista de Produções

A.1 ARTIGOS PUBLICADOS

A.1.1 *Proceedings of the International Conference on Computer-Aided Design, 2015*

•**Qualis:** A1

•**Título:** *Exploiting Non-Critical Steiner Tree Branches for Post-Placement Timing Optimization*

•**Autores:** LIVRAMENTO, V. ; GUTH, CHRYSTIAN ; NETTO, Renan O. ; GUNTZEL, JOSE LUIS ; SANTOS, L. C. V.

•**Abstract:** *The increasing impact of interconnections on the overall circuit performance renders physical design a crucial step to timing closure. Several techniques are used to optimize timing within the flow, such as gate sizing, buffer insertion, and timing-driven placement (TDP). Unfortunately, gate sizing and buffer insertion are not capable of modifying the length of interconnections. Although TDP is able to shorten critical interconnection by finding new legal locations for a subset of cells, it generally overlooks the impact of non-critical branches on the delay of critical cells. This work proposes a post-placement timing optimization technique to reduce the capacitive load of critical cells by shortening non-critical Steiner tree branches. To shorten such branches, our technique uses computational geometry for finding effective cell movements that consider maximum displacement constraints and macro blocks. Our experiments evaluate the capability of our technique to further reduce the timing violations from a TDP solution. We applied our technique on the solutions obtained by the top 3 teams in the ICCAD 2014 TDP Contest, where short and long displacement constraints are defined. For the short constraints, the average reductions assuming worst and total late negative slack metrics are 23% and 34%. Considering the long constraints, the average reductions are 62% and 67%. We also present extensions of our technique to tackle related physical design problems such as early violations reduction and electrical correction.*

A.1.2 *Proceedings of the 2015 Symposium on International Symposium on Physical Design - ISPD '15*

•**Qualis:** A2

•**Título:** *Timing-Driven Placement Based on Dynamic Net-Weighting for Efficient Slack Histogram Compression*

•**Autores:** GUTH, Chrystian ; LIVRAMENTO, VINICIUS ; NETTO, RENAN ; FONSECA, RENAN ; GÜNTZEL, JOSÉ LUÍS ; SANTOS, LUIZ

•**Abstract:** *Timing-driven placement (TDP) finds new legal locations for standard cells so as to minimize timing violations while preserving placement quality. Although violations may arise from unmet setup or hold constraints, most TDP approaches ignore the latter. Besides, most techniques focus on reducing the worst negative slack and let the improvements on total negative slack as a secondary goal. However, to successfully achieve timing closure, techniques must also reduce the total negative slack, which is known as slack histogram compression. This paper proposes a new Lagrangian Relaxation formulation for TDP to compress both late and early slack histograms. To solve the problem, we employ a discrete local search technique that uses the Lagrange multipliers as net-weights, which are dynamically updated using an accurate timing analyzer. To preserve placement quality, our technique uses a small fixed-size window that is anchored in the initial location of a cell. For the experimental evaluation of the proposed technique, we relied on the ICCAD 2014 TDP contest infrastructure. The results show that our technique significantly reduces the timing violations from an initial global placement. On average, late and early total negative slacks are improved by 85.03% and 42.72%, respectively, while the worst slacks are reduced by 71.55% and 34.40%. The overhead in wirelength is less than 0.1%.*

A.2 ARTIGOS SUBMETIDOS

A.2.1 *Proceedings of Design Automation Conference (DAC), 2016 (Sob revisão)*

- Qualis:** A1
- Título:** *Exploiting bipartite graph matching for fast local clock network optimization during incremental timing-driven placement*
- Autores:** Renan Netto, Vinícius Livramento, **Chrystian Guth**, Luiz C. V. dos Santos, José Luís Güntzel

A.2.2 *ACM Transactions on Design Automation of Electronic Systems 2015 (Sob revisão)*

- Qualis:** B2
- Título:** *Clock-Tree-Aware Incremental Timing-Driven Placement*
- Autores:** Vinícius Livramento, Renan Netto, **Chrystian Guth**, José Luís Güntzel, Luiz C. V. dos Santos

A.3 PRÊMIOS

A.3.1 **2015 CAD Contest (Problem C: Incremental Timing-Driven Placement) @ ICCAD 2015**

- Equipe:** First Place (cada014)
- Membros:** Vinícius Livramento, Chrystian Guth, Renan Netto, Prof. José Güntzel e Prof. Luiz Santos
- Colocação:** 1º Lugar

A.3.2 **2014 CAD Contest (Problem B: Incremental Timing-Driven Placement) @ ICCAD 2014**

- Equipe:** UFSC-BRAZIL (cada035)

●**Membros:** Chrystian Guth, Vinícius Livramento, Renan Netto,
Prof. José Güntzel e Prof. Luiz Santos

●**Colocação:** 5º Lugar

**APÊNDICE B - Redução de violações *Late* no decorrer de
10 iterações**

Este apêndice apresenta os valores obtidos de WNS e TNS late pela técnica proposta neste trabalho, partindo das soluções dos três primeiros colocados no *CAD Contest @ ICCAD 2014*. Os resultados foram amostrados ao longo de dez iterações para cada caso de teste.

As seções B.1, B.2 e B.3 apresentam os experimentos partindo das soluções do terceiro, segundo e primeiro colocado, respectivamente, considerando 3 POs no passo de seleção dos candidatos. Cada uma das seções a seguir apresenta os resultados na forma de gráficos de curvas, onde as linhas sólidas azuis (tracejadas vermelhas), relacionadas ao eixo vertical esquerdo (direito), traçam os valores de WNS (TNS) late ao longo das iterações.

B.1 TIME CADA075 (TERCEIRO COLOCADO)

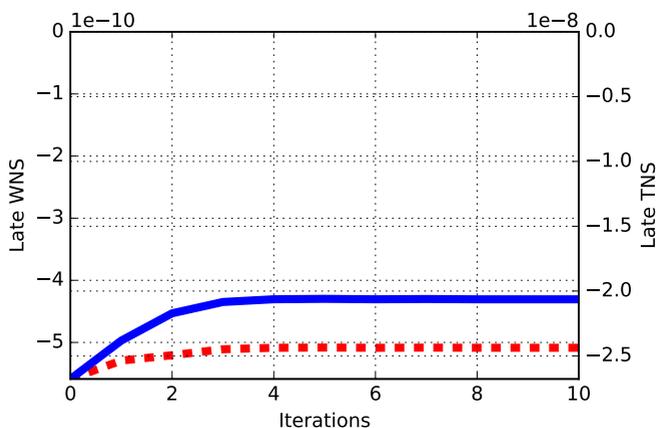


Figura 31 – Valores obtidos para WNS e TNS late no circuito `mgc_edit_dist`, no decorrer de 10 iterações.

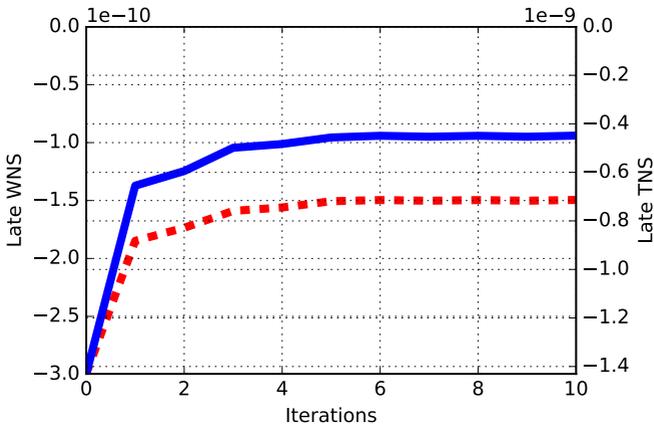


Figura 32 – Valores obtidos para *WNS* e *TNS late* no circuito *mgc_matrix_mult*, no decorrer de 10 iterações.

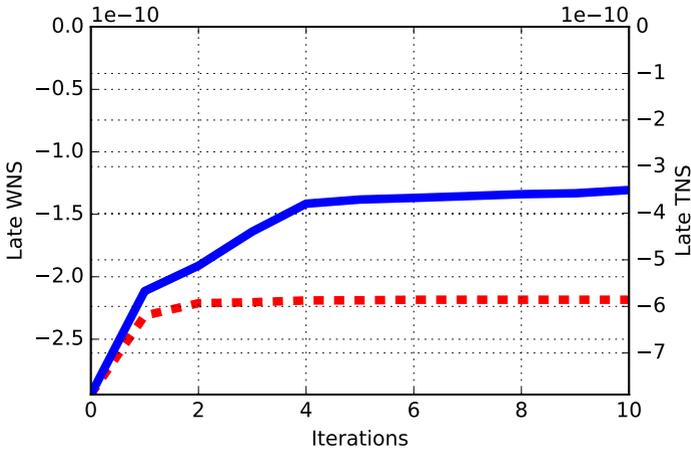


Figura 33 – Valores obtidos para *WNS* e *TNS late* no circuito *vga_1cd*, no decorrer de 10 iterações.

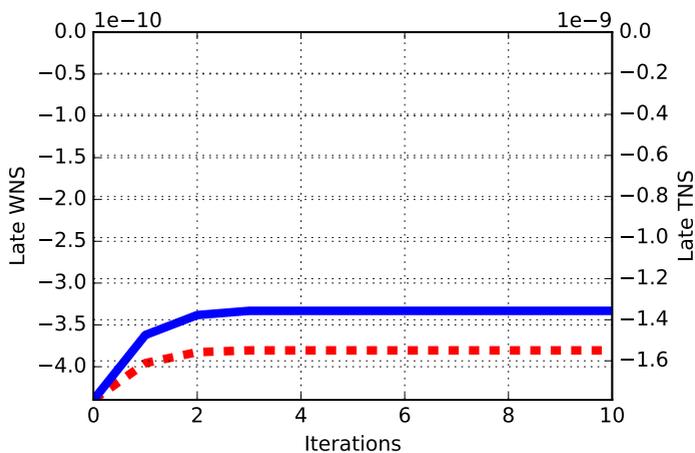


Figura 34 – Valores obtidos para *WNS* e *TNS late* no circuito b19, no decorrer de 10 iterações.

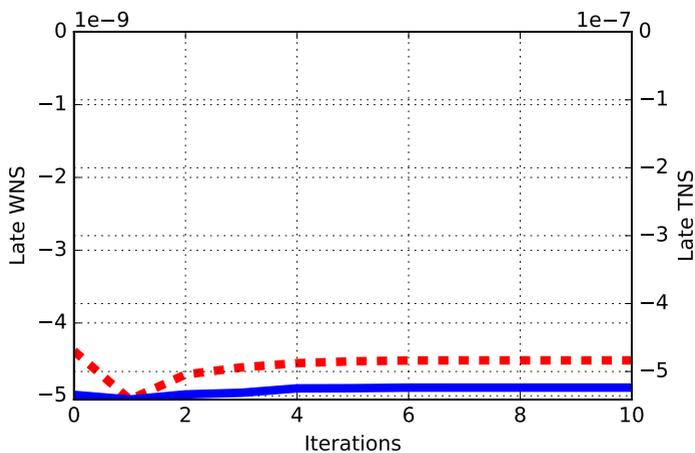


Figura 35 – Valores obtidos para *WNS* e *TNS late* no circuito leon2, no decorrer de 10 iterações.

B.2 TIME CADA019 (SEGUNDO COLOCADO)

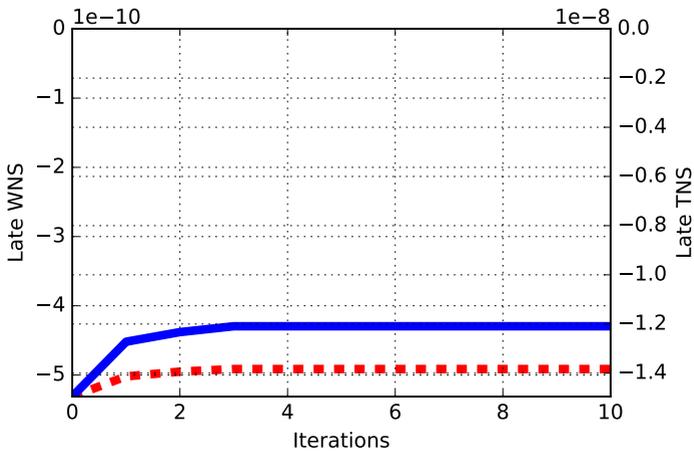


Figura 36 – Valores obtidos para *WNS* e *TNS late* no circuito *mgc_edit_dist*, no decorrer de 10 iterações.

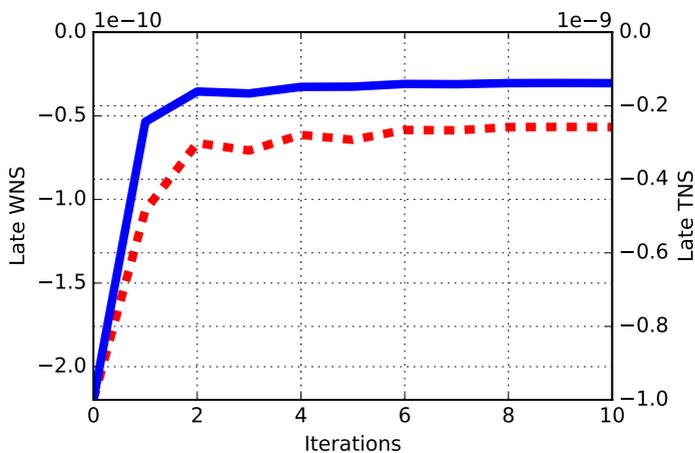


Figura 37 – Valores obtidos para WNS e TNS late no circuito `mgc_matrix_mult`, no decorrer de 10 iterações.

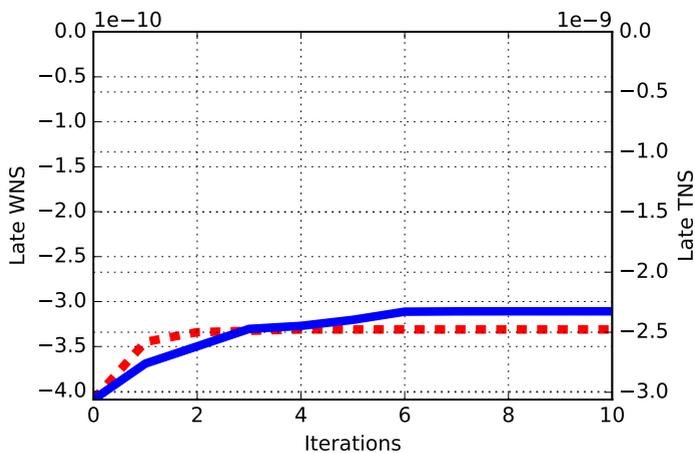


Figura 38 – Valores obtidos para WNS e TNS late no circuito `vga_1cd`, no decorrer de 10 iterações.

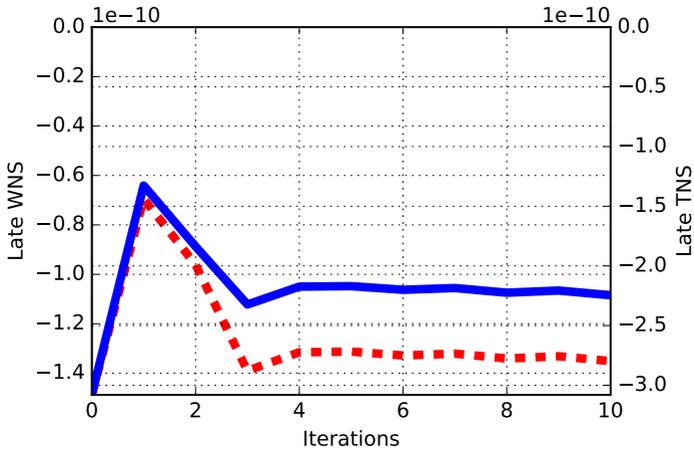


Figura 39 – Valores obtidos para *WNS* e *TNS late* no circuito b19, no decorrer de 10 iterações.

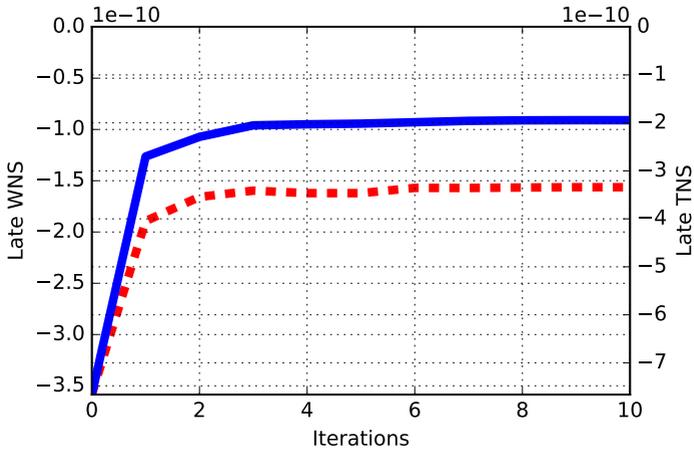


Figura 40 – Valores obtidos para *WNS* e *TNS late* no circuito leon3mp, no decorrer de 10 iterações.

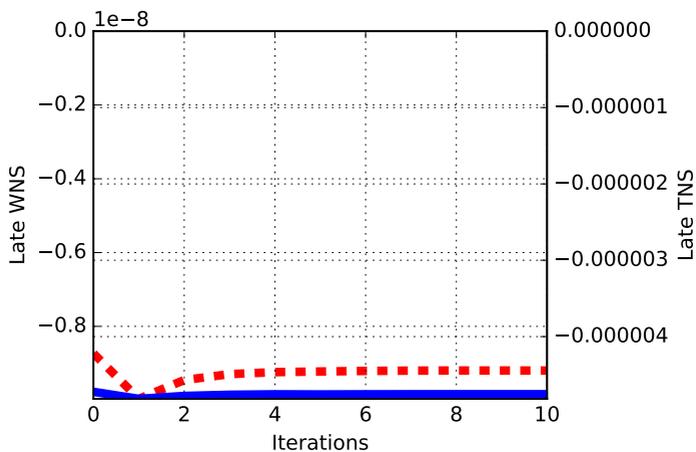


Figura 41 – Valores obtidos para WNS e TNS late no circuito leon2, no decorrer de 10 iterações.

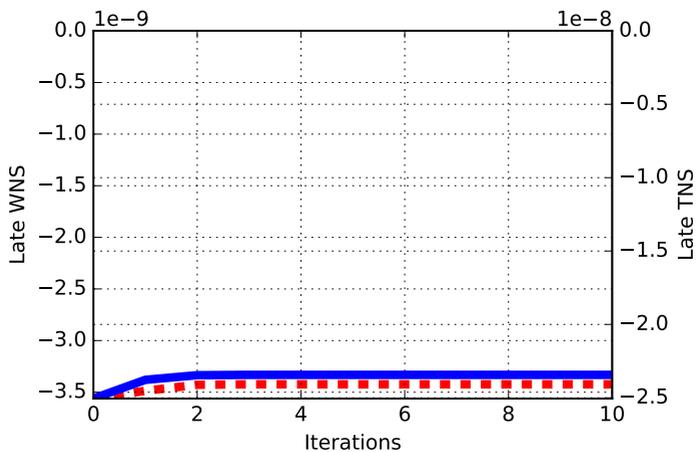


Figura 42 – Valores obtidos para WNS e TNS late no circuito netcard, no decorrer de 10 iterações.

B.3 TIME CADA110 (PRIMEIRO COLOCADO)

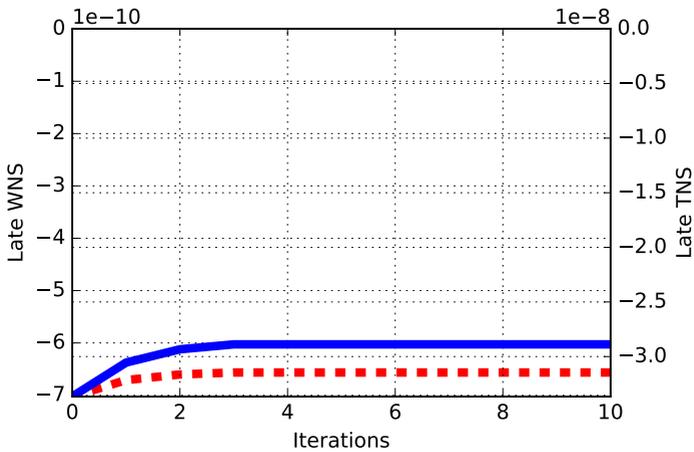


Figura 43 – Valores obtidos para *WNS* e *TNS late* no circuito *mgc_edit_dist*, no decorrer de 10 iterações.

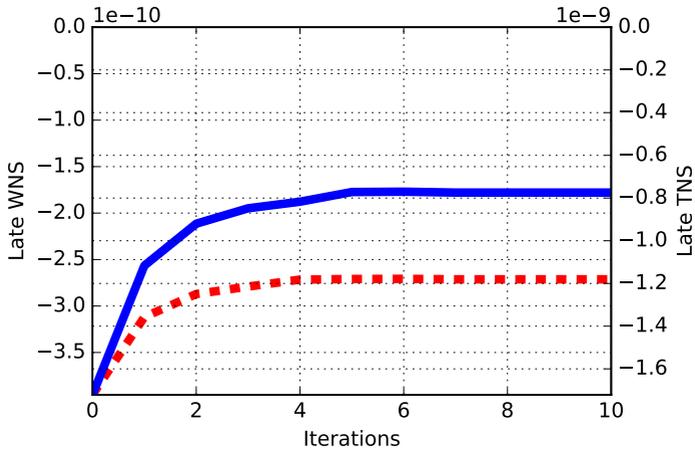


Figura 44 – Valores obtidos para WNS e TNS late no circuito `mgc_matrix_mult`, no decorrer de 10 iterações.

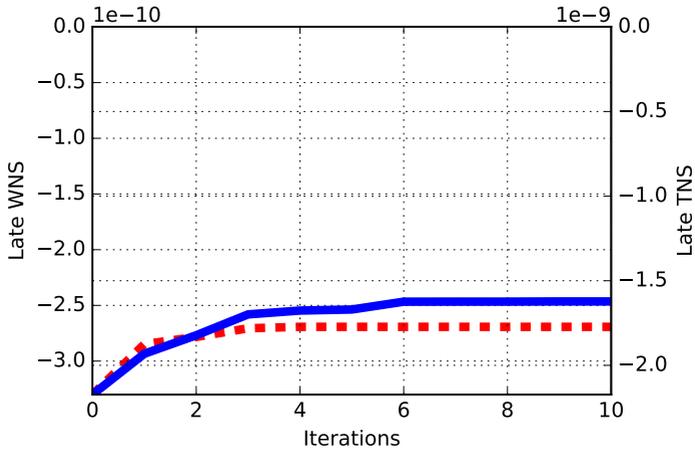


Figura 45 – Valores obtidos para WNS e TNS late no circuito `vga_1cd`, no decorrer de 10 iterações.

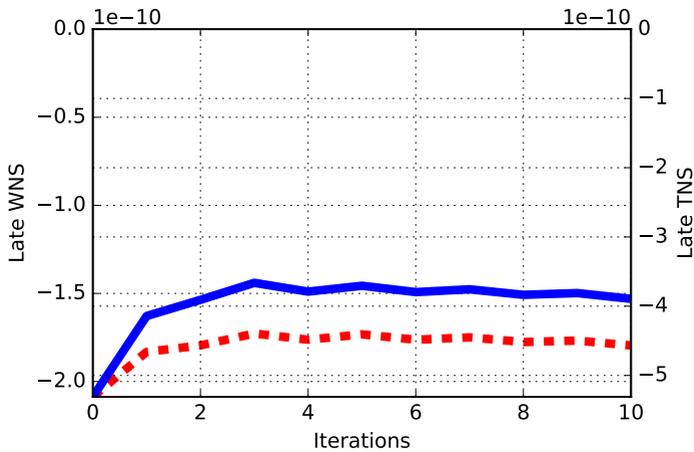


Figura 46 – Valores obtidos para WNS e TNS late no circuito b19, no decorrer de 10 iterações.

B.4 CONCLUSÃO

Observa-se que, em geral, 5 iterações são suficientes para a obtenção dos melhores resultados em termos WNS e TNS ¹. A partir de 5 iterações, observa-se que os valores de WNS e TNS estabilizam. Em função de tal comportamento do algoritmo, decidiu-se utilizar 5 iterações nos experimentos apresentados no Capítulo 5.

¹Como se tratam de números negativos, os melhores valores para WNS e TNS são os maiores que estes podem assumir, isto é, os números negativos mais próximos de 0, ou se possível, 0.