

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

CHARLES CAPELLA LEONHARDT

Roteamento de Circuitos VLSI

Trabalho de Diplomação.

Prof. Dr. Ricardo Augusto da Luz Reis
Orientador

Adriel Mota Ziesemer Junior
Co-orientador

Porto Alegre, junho de 2010.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador da ECP: Prof. Gilson Inácio Wirth

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço aos meus pais Seno Leonhardt e Ana Sueli Capella Leonhardt pelo apoio e compreensão em todos os momentos da minha vida. À minha irmã Siane Capella Leonhardt pelos momentos de diversão e companhia. Aos meus amigos pela confiança e apoio nos momentos bons e ruins.

Agradecimento ao doutorando Adriel Mota Ziesemer Junior pela colaboração neste trabalho e ao professor Ricardo Reis pelo incentivo à pesquisa na área de microeletrônica. Também gostaria de agradecer a todos os colegas do GME pelas dicas e sugestões durante os três anos de trabalho.

O presente trabalho foi realizado com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq – Brasil e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

SUMÁRIO

AGRADECIMENTOS	3
SUMÁRIO	4
LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS	8
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Contextualização	14
1.2 Motivação	15
1.3 Objetivos	16
2 FERRAMENTA ASTRAN	17
3 ROTEAMENTO	20
3.1 Classificação e terminologia	20
3.1.1 Objetivo	20
3.1.2 Tipos de redes	20
3.1.3 Processamento das redes	21
3.2 Algoritmos de pesquisa de caminhos	21
3.2.1 Busca em profundidade ou DFS	21
3.2.2 Busca em largura ou BFS	21
3.2.3 Busca topológica	22
4 ROTEAMENTO GLOBAL	23
5 ROTEAMENTO DETALHADO	25
5.1 Modelo de abstração	25
5.2 Técnica de seleção do nodo origem	27
5.3 Roteamento <i>intracell</i>	27
6 ALGORITMOS UTILIZADOS	28
6.1 A*	28
6.2 Pathfinder	30
6.3 Iterated 1-Steiner Node	31
7 INTEGRAÇÃO ROTEAMENTO GLOBAL E DETALHADO	32
8 DADOS DA IMPLEMENTAÇÃO	35
8.1 Tamanho de código implementado	35
9 RESULTADOS	36
9.1 Roteamento Intracell	36
9.2 Roteamento Global	36
9.3 Roteamento Detalhado	38
9.4 Roteamento Global + Roteamento Detalhado	40

10	CONCLUSÕES.....	43
	REFERÊNCIAS	44

LISTA DE ABREVIATURAS E SIGLAS

ASIC	Application Specific Integrated Circuit (Circuito Integrado de aplicação específica)
ASTRAN	Automatic Synthesis of Transistors (Síntese Automática de Transistores)
BFS	Breadth-First Search
Bi-A*	Bidirectional A*
CAD	Computer-Aided Design (Projeto com Auxílio de Computador)
CI	Circuito Integrado
CIF	Caltech Intermediate Format
CMOS	Complementary Metal-Oxide-Semiconductor
CPA	Crosspoint Assignment
DLM	Discrete Lagrange Multipliers
DFS	Depth-First Search
EDA	Eletronic Design Automation
EDAC	Eletronic Design Automation Consortium
FPGA	Field Programmable Gate Array
GDSII	Graphics Data System II
GHz	Gigahertz
GME	Grupo de Microeletrônica da UFRGS
GLDR	GME Detailed Router
ISPD	International Symposium on Physical Design (Simpósio Internacional de Síntese Física)
LCS*	Lowerbound Coperative Search
LEF	Library Exchange Format
MST	Minimum Spanning Tree (Árvore de Expansão Mínima)
NCR	Negotiation-congestion router (Roteador de negociação de congestionamento)
NMOS	Negative Metal-Oxide Semiconductor
PLACE	Cadence Placement File

PMOS	Positive Metal-Oxide Semiconductor
PSA	Path Search Algorithm (Algoritmo de Pesquisa de Caminho)
SPICE	Simulation Program with Integrated Circuit Emphasis
STL	Standard Template Library
VLSI	Very Large Scale Integration (Integração em Muito Larga Escala)

LISTA DE FIGURAS

<i>Figura 1.1. Ciclo de projeto VLSI</i>	12
<i>Figura 1.2. Etapas da Síntese Física</i>	13
<i>Figura 1.3: Visão geral do problema de roteamento : (a) Circuito posicionado; (b) Roteamento Global; (c) Roteamento Detalhado ([HUA2009]).</i>	14
<i>Figura 2.1: Janela inicial da ferramenta</i>	17
<i>Figura 2.2. Descrição Spice da célula ADD32</i>	18
<i>Figura 2.3. Leiaute da célula ADD32 gerado na ferramenta ASTRAN para a tecnologia 90 nm.</i>	19
<i>Figura 2.4. Roteamento detalhado obtido na ferramenta ASTRAN.</i>	19
<i>Figura 4.1. Circuito dividido em Global Bins e a respectiva grade ([MOF2008]).</i>	23
<i>Figura 4.2. Obtenção do grafo para roteamento global a partir da partição do circuito em Global Bins.</i>	24
<i>Figura 5.1. Switchbox para roteamento detalhado simplificada [SAX2007].</i>	25
<i>Figura 5.2. Dois modelos de roteamento detalhado: (a) baseado em grade; (b) sem grade definida ([HUA2009]).</i>	26
<i>Figura 6.1: Pseudocódigo A*</i>	29
<i>Figura 6.2: Pesquisa utilizando: (a)Lee e (b)A*</i>	29
<i>Figura 6.3: Rip-up e reroute</i>	31
<i>Figura 7.1. Roteamento Global de uma rede de 2 nodos.</i>	33
<i>Figura 7.2. Roteamento Detalhado obtido usando resultados do Roteamento Global.</i> 33	
<i>Figura 7.3. Roteamento Detalhado.</i>	34
<i>Figura 9.1. Gráfico da complexidade em relação a média dos tempos de execução do roteador global para cada tamanho de grade.</i>	38

LISTA DE TABELAS

<i>Tabela 8.1. Número de linhas de código da implementação.</i>	<i>35</i>
<i>Tabela 9.1. Comparação do roteador intracell com e sem otimização</i>	<i>36</i>
<i>Tabela 9.2. Custo total de interconexão para cada capacidade global.....</i>	<i>37</i>
<i>Tabela 9.3. Tempo de execução para cada capacidade global.....</i>	<i>37</i>
<i>Tabela 9.4. Roteamento global de circuitos de teste.....</i>	<i>38</i>
<i>Tabela 9.5. Comparação entre a técnica de seleção da origem e a abordagem original.</i>	<i>39</i>
<i>Tabela 9.6. Comparação da equação 1 do Pathfinder com o RotDL.</i>	<i>40</i>
<i>Tabela 9.7. Comparação da equação 2 do Pathfinder com o RotDL.</i>	<i>40</i>
<i>Tabela 9.8. Comparação do custo de interconexão.</i>	<i>41</i>
<i>Tabela 9.9. Comparação do tempo de execução.....</i>	<i>41</i>
<i>Tabela 9.10. Comparação das tentativas utilizadas.</i>	<i>42</i>

RESUMO

Roteamento é a etapa do fluxo de geração de circuitos integrados onde são realizadas as interconexões entre os diferentes elementos do circuito. Está incluído na etapa de síntese física, que trata da construção do leiaute, uma informação geométrica na qual são representados os transistores e suas interconexões. O roteamento assume grande importância devido à crescente influência das interconexões no atraso total do circuito e o acréscimo de área que um circuito de difícil roteamento pode gerar. Além disto, o tempo de execução costuma ser um fator limitante que impede que algoritmos que possibilitem maiores otimizações sejam usados. Diante disto, este trabalho tem como objetivo o desenvolvimento de uma ferramenta para roteamento global e detalhado de circuitos integrados, que seja capaz de rotear circuitos com grande número de elementos com um bom compromisso entre qualidade de roteamento e tempo de execução. Outro objetivo é desenvolver uma ferramenta para roteamento intracell utilizada na geração de bibliotecas de células. Para atingir os objetivos propostos foi necessário desenvolver soluções adequadas a cada uma das etapas do fluxo de roteamento. O algoritmo para roteamento *intracell* incluindo uma fase de otimização obtém ganho de 0,2% em relação à solução sem otimização desempenhando um papel importante na geração de bibliotecas de células melhores. Quanto ao roteamento detalhado, os resultados indicam a viabilidade do algoritmo quando comparado ao Rotdl, obtendo circuitos com *wirelength* equivalente em 50% menos tempo. Com relação ao roteador global o algoritmo desenvolvido é bastante eficiente, entretanto a conexão com o roteador detalhado ainda é muito incipiente. Por fim, este trabalho foi integrado à ferramenta de síntese física ASTRAN, do GME, colaborando para o aperfeiçoamento do seu fluxo de geração automática.

Palavras-Chave: VLSI, EDA, síntese física, roteamento.

Routing VLSI Circuits

ABSTRACT

Routing is the step in the generation flow of integrated circuits, in that are held the interconnections between different circuit elements. It's included in Physical Synthesis step that builds the layout, a geometric information in which are represented the transistors and its interconnections. Routing is very important due to the growing influence of the interconnections in total delay of the circuit and the addition in area that a circuit with difficult routing can generate. Moreover, the execution time is often a limiting factor that prevents algorithms that allow more optimizations to be used. Therefore, this paper aims to develop a tool for global and detailed routing of integrated circuits, which must be able to route circuits with large number of elements with a good compromise between quality of routing and execution time. Another goal is to develop a tool for intracell routing used in the generation of cell libraries. To achieve the proposed objectives was necessary to develop appropriate solutions to each stage of the routing flow. The algorithm for intracell routing including an optimization phase obtains gain of 0.2% compared to the solution without optimization playing an important role in the generation of better library cells. For detailed routing, the results indicate the feasibility of the algorithm when compared to Rotdl, obtaining equivalent wirelength in 50% less time. The global router algorithm is very efficient, however the connection with the detailed router is still incipient. Finally, this work was integrated into the physical synthesis tool ASTRAN of GME, contributing to the improvement of the automatic generation flow.

Keywords: VLSI, EDA, physical synthesis, routing.

1 INTRODUÇÃO

O projeto de circuitos integrados é um processo extremamente complexo dividido em vários estágios conforme a Figura 1.1. Uma dessas etapas é a síntese física onde a representação do circuito através de uma descrição funcional é convertida para uma representação geométrica (leiaute) [SHE1999]. Esta é uma fase extremamente complexa e assim é normalmente dividida em várias etapas conforme a Figura 1.2. Uma dessas etapas é o roteamento, que é o foco deste trabalho.

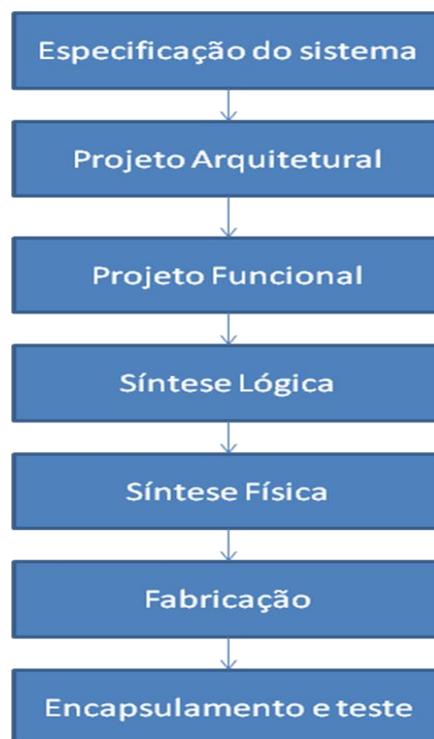


Figura 1.1. Ciclo de projeto VLSI

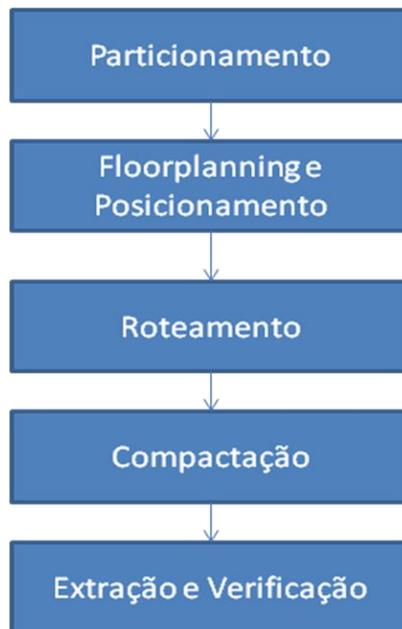


Figura 1.2. Etapas da Síntese Física

O roteamento é o processo posterior ao posicionamento, que determina a localização de cada elemento de um circuito integrado. É a etapa que conecta os componentes posicionados obedecendo a regras de projeto. A especificação de um problema de roteamento consiste da posição dos terminais, do *netlist* que indica quais terminais devem ser conectados e da área disponível para roteamento em cada camada [GER1998].

O roteamento é extremamente dependente da qualidade do resultado do posicionamento, pois este pode inviabilizar a existência de um roteamento que atenda as regras de projeto e as especificações de conexões dentro de um período de tempo aceitável.

Além disso, o roteamento é tipicamente um problema combinatorial extremamente complexo. Para fazê-lo tratável, este é usualmente resolvido por uma abordagem de dois estágios: roteamento global seguido de roteamento detalhado.

Isso é feito porque o aumento do tamanho e complexidade dos circuitos faz com que não seja possível para uma abordagem de somente roteamento detalhado determinar as posições das interconexões do circuito inteiro em tempo factível. Por isso, é utilizada a abordagem de simplificar o problema como um todo e depois tratar com maior detalhamento partes menores do circuito. Assim, o objetivo dessa abordagem é permitir que a ferramenta de roteamento possa lidar com circuitos maiores com qualidade aceitável em um tempo factível.

O roteador global primeiro particiona o circuito em regiões e decide os caminhos entre as regiões para todas as redes. Depois, de acordo com os caminhos obtidos no roteamento global, o roteador detalhado atribui trilhas e vias para as redes. Uma idéia da tarefa de cada etapa é mostrada na Figura 1.3.

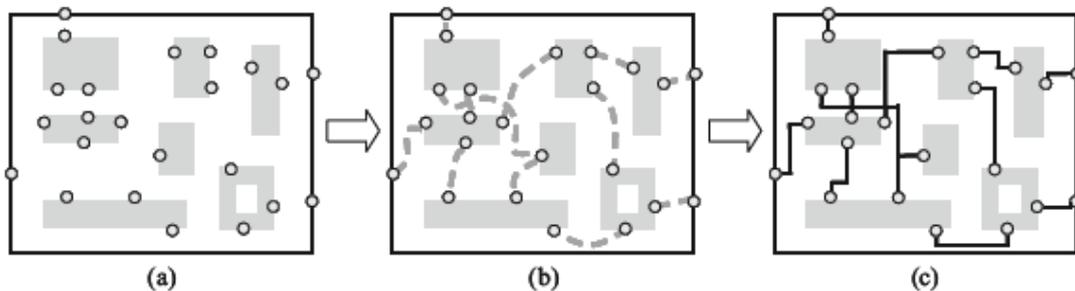


Figura 1.3: Visão geral do problema de roteamento : a) Circuito posicionado; (b) Roteamento Global; (c) Roteamento Detalhado ([HUA2009]).

O roteador global fornece uma boa idéia da dificuldade apresentada ao roteador detalhado através de mapas de congestionamento. Se há uma alta porcentagem de áreas congestionadas, é muito provável que o roteador detalhado não obtenha sucesso em sua tarefa. Limitado por essas restrições deve colaborar com os objetivos gerais de otimização do circuito.

O roteador detalhado realiza a interconexão entre os nodos dentro de cada região. Esse problema é bastante paralelizável, pois o circuito está dividido em várias regiões independentes do ponto de vista do roteamento detalhado. Dessa forma, podem ser utilizadas técnicas que sejam mais demoradas, mas que tragam resultados melhores em termos de custo e atraso do circuito. Ao final dessa etapa é obtida a configuração final de roteamento do circuito.

Outro tipo de roteamento é o *intracell* que é um pouco distinto dos outros, pois está relacionado à interconexão no interior da célula, que é considerada um elemento básico no roteamento global e no detalhado. Esta etapa é muito importante não só para melhorar as características elétricas da célula e reduzir a sua largura (uma vez que a altura costuma ser fixa para todas as células da biblioteca), mas também para prover ao roteador global e posteriormente ao detalhado, um posicionamento dos pinos de entrada e saída da célula.

Como cada célula é utilizada várias vezes no circuito é importante que esse roteamento seja o mais otimizado possível. Assim, técnicas de maior complexidade podem ser utilizadas nessa etapa.

1.1 Contextualização

Ferramentas de CAD são utilizadas em um número cada vez maior de áreas da computação com o objetivo de aumentar a produtividade. No projeto de circuitos VLSI, seu uso tem crescido em importância recentemente devido ao aumento da complexidade dos dispositivos e à necessidade de obter produtos que atendam ao *time-to-market*.

O objetivo do mercado é a produção de dispositivos com um desempenho cada vez melhor e capazes de efetuar um maior número de operações. Por isso, as fábricas de semicondutores se esforçam na tentativa de produzir circuitos integrados com maior densidade de transistores, melhor manufaturabilidade, menor atraso e consumo de potência.

Devido às melhorias feitas no processo de fabricação de semicondutores, já existem circuitos comerciais que contêm centenas de milhões de transistores dentro de um único *chip*. Por outro lado, cada melhoria no processo de fabricação cria um novo conjunto de regras para projeto de circuitos integrados e o uso de ferramentas automáticas de geração de circuitos pode diminuir o tempo necessário para refazer todo o leiaute.

O crescimento na complexidade desses problemas provocou o surgimento de uma considerável indústria de software para automação de projetos eletrônicos, ou EDA. Essa indústria obteve faturamento de US\$ 5,5 bilhões de dólares em 2009 segundo o EDAC [EDA2010].

Partindo do fato de que o roteamento possui grande influência na otimização dessas ferramentas de EDA, torna-se fundamental o desenvolvimento de algoritmos com grande eficiência nesta etapa do fluxo de síntese física de dispositivos semicondutores.

Dessa forma, algoritmos adequados a cada uma das etapas do roteamento são um fator importante na obtenção de circuitos integrados melhores, permitindo que as ferramentas de geração automática de leiaute possam ser usadas competitivamente no desenvolvimento de projetos reais.

1.2 Motivação

Apesar de ser uma das primeiras áreas de CAD que foi automatizada, o roteamento de circuitos VLSI permanece como uma área com significativa atividade de pesquisa e desenvolvimento [ISP2008]. Além disso, se comparado ao problema do posicionamento, existem significativamente menos algoritmos se propondo a resolver este problema.

Os esforços em roteamento são motivados por desafios impostos pela escala nanométrica incluindo:

- bancos de dados de conexões gigantescos que dependem de estrutura de dados e algoritmos extremamente eficientes.
- vias relativamente não confiáveis, o que exige sua duplicação e motiva um esforço extra na diminuição de sua quantidade.
- restrições com relação à integridade do sinal e o impacto considerável da capacitância lateral no atraso de interconexão, que levam a restrições na densidade dos fios.

Esse trabalho se propõe a desenvolver todas as etapas do fluxo de roteamento, construindo algoritmos específicos de acordo com os requisitos de cada uma das etapas. Para isso devem ser estudadas técnicas já aplicadas e testadas novas soluções.

Uma grande vantagem de se desenvolver todo o fluxo de roteamento é o fato de se poder considerar as características dos passos anteriores para obter resultados melhores nas fases posteriores. Além de evitar operações redundantes que podem ser passadas como parâmetros para as fases seguintes.

Outra motivação é auxiliar no aperfeiçoamento de uma ferramenta de síntese física em desenvolvimento no GME, além de expandir o conhecimento acerca de todo o fluxo de roteamento bem como de todo o fluxo de síntese física.

1.3 Objetivos

O objetivo do desenvolvimento de todo o fluxo de roteamento é a sua integração à ferramenta de geração automática de leiaute em desenvolvimento no GME chamada ASTRAN [ZIE2009].

Essa ferramenta gera o leiaute do circuito a partir da descrição *Spice* e dos parâmetros tecnológicos. Ela realiza todas as etapas da síntese física, tais como: geração de células, posicionamento e roteamento. Além disso, pode armazenar em formatos comerciais os resultados de cada uma das etapas separadamente, ou ainda o resultado final.

A partir dessas características, é possível utilizar essa ferramenta como auxiliar no projeto de circuitos integrados utilizando ferramentas comerciais de CAD, sendo possível o intercâmbio de arquivos de roteamento entre as ferramentas.

Para tal intercâmbio, é necessário que os algoritmos de roteamento tenham uma eficiência aceitável para que seja suficientemente vantajoso o uso dessa ferramenta comparado ao uso de ferramentas comerciais.

2 FERRAMENTA ASTRAN

É uma ferramenta que realiza todo o fluxo de síntese do circuito a partir da descrição *Spice* e das regras de leiaute da tecnologia. Está disponível para os sistemas operacionais Windows, Linux e Mac OS (processadores Intel e PowerPC) e está descrita em [ZIE2007] [NES2010].

Na Figura 2.1 é mostrada a interface da ferramenta desenvolvida utilizando C++ e WxWidgets [WXW2009].

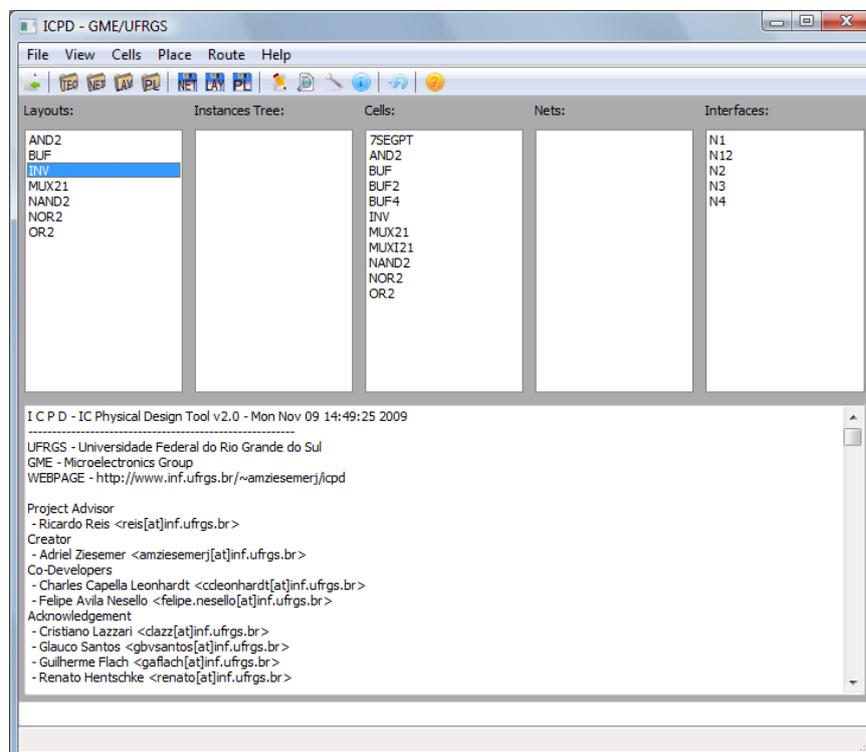


Figura 2.1: Janela inicial da ferramenta

O produto final da ferramenta é o leiaute do circuito que pode ser salvo nos formatos comerciais. A ferramenta possui suporte para várias extensões de arquivos e portabilidade com as ferramentas comerciais da Cadence. Atualmente, suporta os formatos CIF e GDSII para leiautes, PLACE para posicionamento, SPICE para *netlist* e LEF para biblioteca de células - com o qual é possível realizar o posicionamento e roteamento também de *standard cells*. Além desses, possui seus próprios formatos para ler e salvar projetos, regras de desenho, *netlist*, leiautes, posicionamento e roteamento. Isso possibilita a edição em uma ampla variedade de ferramentas. Em relação mais

especificamente ao roteamento, é possível salvar o resultado dessa etapa e posteriormente restaurar este resultado na própria ferramenta ou em outras. Também é possível carregar um arquivo de posicionamento em formatos comerciais e realizar o roteamento utilizando o algoritmo desenvolvido.

Além do leiaute de roteamento também são apresentadas estatísticas com relação a custo de interconexão, *wirelength*, número de pinos e tempo de execução do algoritmo.

Na Figura 2.3 é mostrado o resultado da execução de todo o fluxo da ferramenta ASTRAN para uma célula ADD32 utilizando o algoritmo desenvolvido a partir da descrição spice da Figura 2.2 e do arquivo de tecnologia.

```
.subckt add32 A B CI CO S GND VCC
M22 CO 10 VCC VCC PMOS L=0.35U W=3.2U
M23 VCC A 8 VCC PMOS L=0.35U W=3.2U
M24 8 B 10 VCC PMOS L=0.35U W=3.2U
M25 10 CI 2 VCC PMOS L=0.35U W=3.2U
M26 2 A VCC VCC PMOS L=0.35U W=3.2U
M27 VCC B 2 VCC PMOS L=0.35U W=3.2U
M28 9 10 1 VCC PMOS L=0.35U W=3.2U
M29 1 A VCC VCC PMOS L=0.35U W=3.2U
M30 VCC B 1 VCC PMOS L=0.35U W=3.2U
M31 1 CI VCC VCC PMOS L=0.35U W=3.2U
M32 9 CI 7 VCC PMOS L=0.35U W=3.2U
M33 7 B 6 VCC PMOS L=0.35U W=3.2U
M34 6 A VCC VCC PMOS L=0.35U W=3.2U
M35 VCC 9 S VCC PMOS L=0.35U W=3.2U
M36 CO 10 GND GND NMOS L=0.35U W=1.9U
M37 GND A 5 GND NMOS L=0.35U W=1.9U
M38 5 B 10 GND NMOS L=0.35U W=1.9U
M39 10 CI 20 GND NMOS L=0.35U W=1.9U
M40 20 A GND GND NMOS L=0.35U W=1.9U
M41 GND B 20 GND NMOS L=0.35U W=1.9U
M42 9 10 11 GND NMOS L=0.35U W=1.9U
M43 11 A GND GND NMOS L=0.35U W=1.9U
M44 GND B 11 GND NMOS L=0.35U W=1.9U
M45 11 CI GND GND NMOS L=0.35U W=1.9U
M46 9 CI 4 GND NMOS L=0.35U W=1.9U
M47 4 B 3 GND NMOS L=0.35U W=1.9U
M48 3 A GND GND NMOS L=0.35U W=1.9U
M49 GND 9 S GND NMOS L=0.35U W=1.9U
.ends
```

Figura 2.2. Descrição Spice da célula ADD32

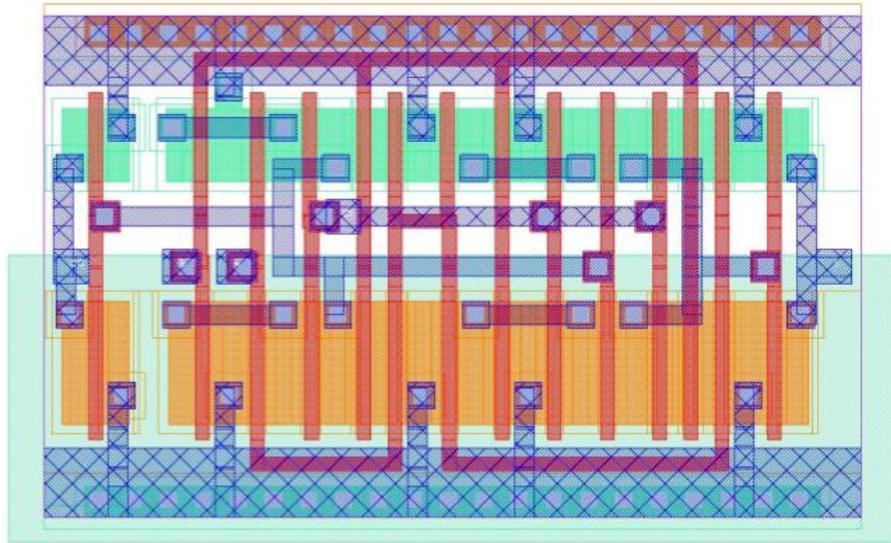


Figura 2.3. Leiaute da célula ADD32 gerado na ferramenta ASTRAN para a tecnologia 90 nm.

Na Figura 2.4 podemos ver o resultado de um roteamento detalhado obtido na ferramenta ASTRAN.

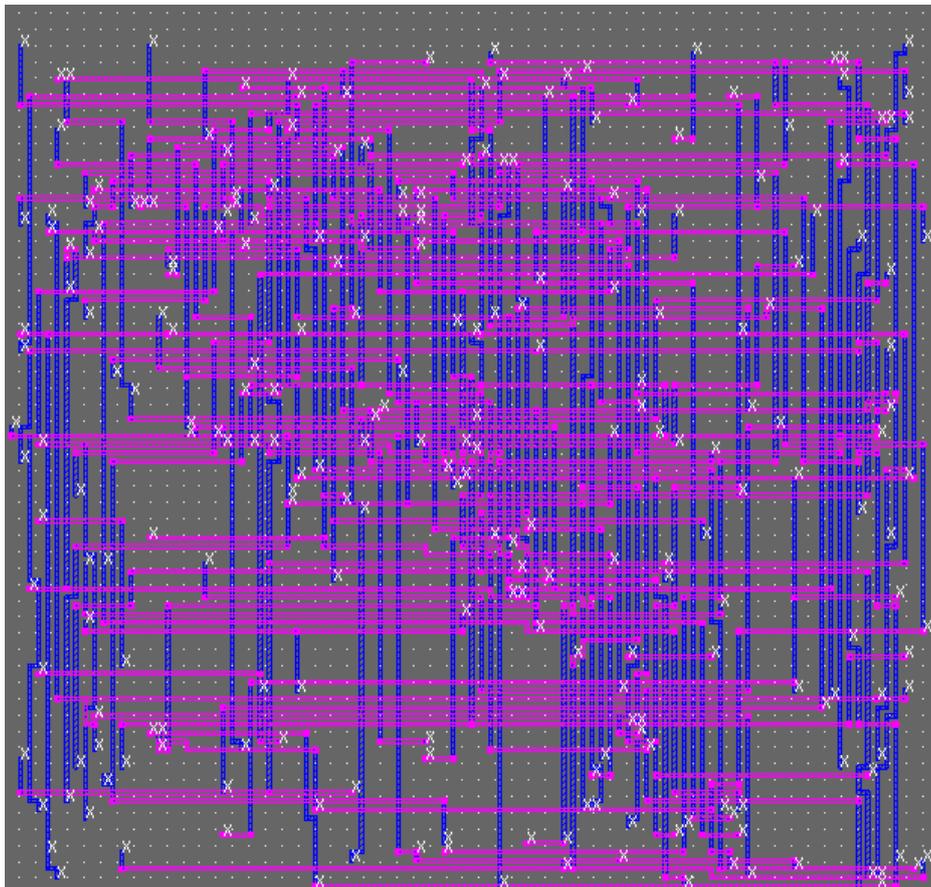


Figura 2.4. Roteamento detalhado obtido na ferramenta ASTRAN.

3 ROTEAMENTO

3.1 Classificação e terminologia

3.1.1 Objetivo

Em relação a aplicação do algoritmo de roteamento temos as seguintes categorias segundo [PRE1988] e [JOH1997]:

- **Roteamento detalhado:** Especifica completa e detalhadamente as rotas de cada conexão. Sendo um problema complexo para o circuito inteiro, é usualmente tratado por partes, restringindo o roteamento detalhado a pequenas porções do circuito.
- **Roteamento global:** Etapa responsável por dividir o problema do roteamento de todo o circuito para um conjunto de roteamentos detalhados suficientemente pequenos para que sejam tratáveis.
- **Roteamento funcionalmente especializado:** Necessário para conexões com características especiais, como alimentação, relógio, barramento ou sinais de entrada e saída.
- **Roteamento tecnologicamente especializado:** Ocorre em situações específicas da tecnologia de fabricação de equipamentos eletrônicos.

Neste trabalho, os algoritmos desenvolvidos são para roteamento detalhado e roteamento global, não sendo adequados às outras classes de problemas apresentados nesta taxonomia.

3.1.2 Tipos de redes

O roteamento é um problema bastante genérico e assim surgem taxonomias diferentes para identificar cada tipo de problema. Em relação ao tipo de rede que vai ser roteada segundo [SAP2003], temos:

- **Redes de sinal:** Compostas por sinais lógicos que devem se propagar pelas células lógicas.
- **Redes de relógio:** Responsáveis pela distribuição do sinal de clock pelo circuito.
- **Redes de alimentação:** Incumbidas de prover VDD e GND para cada célula.

Neste trabalho os algoritmos desenvolvidos são específicos para roteamento de redes de sinal, não sendo adequados para o tratamento dos outros tipos de redes. Isso porque não levam em conta problemas de atraso e degradação do sinal causados pelas interconexões.

3.1.3 Processamento das redes

De acordo com [JOH1997] os algoritmos de roteamento podem ser classificados em relação ao relacionamento entre as diversas redes durante o processamento da seguinte forma:

- **Incrementais ou sequenciais:** Realizam as conexões uma a uma até completar todas ou não ser mais possível efetuá-las [PAN2006];
- **Integrais ou paralelos:** Consideram ao mesmo tempo todas as conexões necessárias e modificam as soluções encontradas até ser obtida a melhor ou alguma satisfatória [RAG1991][ROY2007][CHO2006].
- **Refinadores ou iterativos:** Partem de uma solução inicial simples ou parcial e modificam as soluções encontradas até ser obtida a melhor ou alguma satisfatória.

Os algoritmos utilizados neste trabalho são uma combinação da primeira com a terceira taxonomia. Roteiam as redes em sequência mas depois iterativamente refinam o resultado até obter a solução desejada.

3.2 Algoritmos de pesquisa de caminhos

Esses algoritmos também conhecidos como PSA são utilizados no roteamento de cada rede. São três as principais estratégias de busca utilizando grafos de acordo com [SHE1999]:

3.2.1 Busca em profundidade ou DFS

Nesta abordagem, um vértice v é selecionado para ser expandido entre os nodos mais recentemente visitados. Quando todos os arcos de v foram explorados, o algoritmo retorna para o vértice anterior, que pode conter vértices inexplorados.

3.2.2 Busca em largura ou BFS

A idéia básica dessa estratégia é explorar todos os vértices adjacentes a um vértice antes de explorar qualquer outro vértice. Começando com um vértice v , coloca os vértices alcançáveis em uma fila, e marca o nodo v como visitado. Se um vértice já foi marcado como visitado ele não é adicionado novamente a fila.

Uma das primeiras aplicações de uma busca BFS em roteamento é o algoritmo de LEE [LEE1961], que consiste em marcar os nodos com números sequenciais durante a expansão, permitindo a reconstrução do caminho utilizando retraço *backtracking*, enquanto desvia de obstáculos.

Para diminuir o espaço de busca necessário foi criado o algoritmo A^* que permite uma significativa redução na área de busca, como é mostrado na seção 6.1. Outra estratégia explorada é a busca bidirecional utilizando o A^* , desenvolvida em algoritmos como o LCS* [JOH2000a][JOH2000b]. Essa estratégia consiste em aplicar uma frente de busca na origem e outra no destino que se encontram formando o caminho entre os nodos.

Esses algoritmos são aplicados a versões geométricas do problema sobre uma grade regular, onde os pesos dos arcos correspondem a distância entre os vértices. Entretanto, esse modelo pode ser adaptado para levar em conta também as resistências dos fios, o congestionamento, capacidades de acoplamento entre outros fatores [SAN2006].

A abordagem escolhida para esse trabalho é a do A* por suas vantagens de menor espaço de busca em relação ao algoritmo de LEE e a menor complexidade de implementação quando comparado ao LCS*.

3.2.3 Busca topológica

Em um grafo dirigido acíclico é natural visitar os nodos pais antes de visitar os filhos. Assim, listando os vértices em ordem topológica, se a conexão entre os vértices u e v existe então u aparece antes de v na ordem topológica.

4 ROTEAMENTO GLOBAL

Enquanto o roteador detalhado é responsável pela definição do leiaute final de interconexão, trabalhando sobre uma grade detalhada, o roteador global é aplicado sobre uma região mais abstrata com maior granularidade [SAN2006].

A grade de roteamento global é modelada como um grafo onde os vértices correspondem a um conjunto de regiões retangulares nas quais é dividido o circuito. Essas sub-áreas são conhecidas como **Global Bins** [MOF2008], **Global Routing Cells – GRCs** [JOH2001a] ou **g-cells** [WES2005] [ROY2007] [SWA2008]. Neste trabalho é usada a terminologia dos Global Bins. Os arcos que conectam os vértices representam as interfaces entre cada um dos Global Bins, e são compartilhados por dois Global Bins adjacentes, conforme está indicado na Figura 4.1.

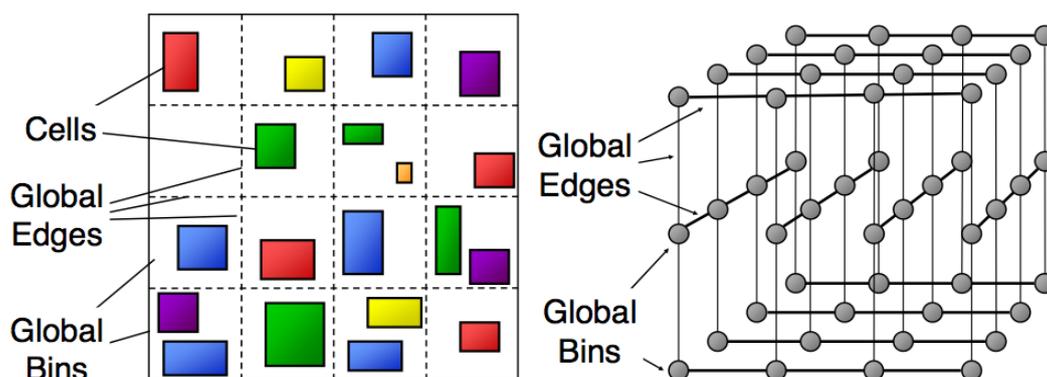


Figura 4.1. Circuito dividido em Global Bins e a respectiva grade ([MOF2008]).

A definição das regiões e das capacidades das interfaces é uma primeira etapa importante na solução do problema do roteamento global. As capacidades são definidas no sentido de modelar a limitação de recursos de conexão entre as células, que podem variar nas camadas do circuito.

Na Figura 4.2, temos um exemplo de um circuito dividido em Global Bins e podemos ver que o número de canais entre os Global Bins é igual a três e portanto, o arco que conecta os dois Global Bins no correspondente grafo de representação tem capacidade três.

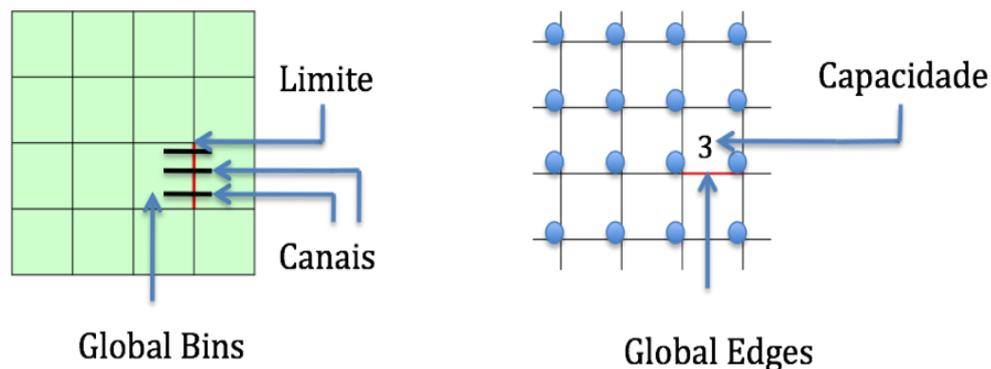


Figura 4.2. Obtenção do grafo para roteamento global a partir da partição do circuito em Global Bins.

Assim a tarefa do roteador global é encontrar uma solução que ao mesmo tempo atende a necessidade de conexões entre os *bins* com o menor *overflow*. O *overflow* é definido como o número de conexões que excederam a capacidade disponível de sua interface.

Limitado por essas restrições, o roteador global deve colaborar com os objetivos gerais de otimização do circuito. Além disso, deve distribuir da forma mais uniforme possível as conexões globais. Para distribuí-las, é essencial evitar regiões altamente congestionadas, que poderiam dificultar ou impossibilitar a existência de um roteamento detalhado.

Como resultado final deve identificar uma sequência de regiões pelas quais determinada rede deve ser roteada, considerando as restrições impostas pelo congestionamento de cada interface entre regiões.

Na solução desenvolvida o circuito posicionado é recebido pelo roteador global e dividido em *bins* que resultam em um grafo bidimensional que é solucionado de acordo com as capacidades. O resultado deste é utilizado posteriormente pelo roteador detalhado que realiza somente as conexões nas regiões que são permitidas pelo roteador global.

Cada uma das redes representadas na grade detalhada é também representada na grade global. Os nodos que estão no mesmo *bin* e pertencem à mesma rede são representados somente uma vez na correspondente rede global, pois correspondem a um mesmo nodo na grade global.

No roteamento global desenvolvido é utilizado o algoritmo Pathfinder [MCM1995] para lidar com o congestionamento entre as redes, e no roteamento de cada uma das redes (roteamento de sinal) é utilizado o algoritmo A*. São definidas regiões que correspondem a divisões do circuito detalhado mapeando o circuito 3D detalhado para um circuito 2D global. Isso foi realizado porque o roteamento global desenvolvido apresentou melhores resultados quando se utiliza roteamento 2D.

5 ROTEAMENTO DETALHADO

O roteador detalhado tem como entrada o resultado do roteamento global com um reduzido número de *overflows*. A partir disso, determina fios às trilhas de roteamento, gerando vias quando necessário, enquanto respeita as restrições quanto a espaçamento e regras de desenho mais sofisticadas.

O algoritmo para realizar o roteamento detalhado desenvolvido utiliza o Pathfinder para gerenciar os conflitos entre as redes além de utilizar o A* como roteador de sinal. A ferramenta desenvolvida foi batizada de GDLR.

A área de atuação do roteador detalhado são as regiões definidas e conectadas no roteamento global também conhecidas como *switchboxes*. Um exemplo de um roteamento de uma *switchbox* está na Figura 5.1, que mostra também os pinos virtuais que são utilizados para definir as conexões com as outras *switchboxes*.

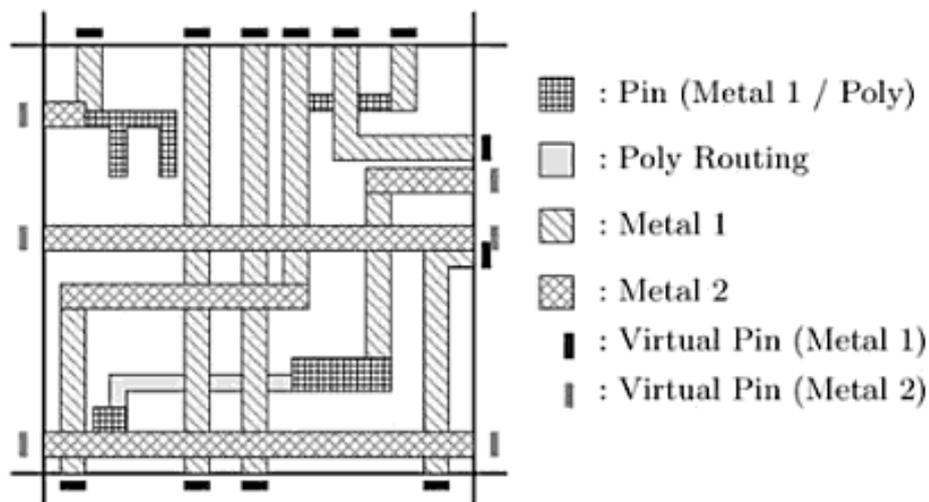


Figura 5.1. Switchbox para roteamento detalhado simplificada [SAX2007].

Neste trabalho não são definidos pinos virtuais pois não foi implementada uma divisão total do roteamento detalhado dos *bins*. Assim, é necessário manter todas as regiões para realizar o roteamento, o que diminui muitas das vantagens da estratégia de dois níveis (roteamento global seguido de roteamento detalhado).

5.1 Modelo de abstração

Existem dois modelos para roteamento detalhado: baseado em grade e sem grade definida.

No modelo baseado em grade, é criada uma grade na região de roteamento, e então o roteador tenta encontrar caminhos de roteamento válidos. O espaço entre linhas de grade adjacentes é chamado de *pitch*, que é definido no arquivo de tecnologia e é maior ou igual à soma da mínima largura e do espaçamento dos fios. Este modelo é muito mais eficiente e de implementação mais simples, pois o problema é simplificado para uma estrutura mais tratável.

Com a grade definida há ainda o compromisso com as regras de projeto, que são verificadas posteriormente nas ferramentas de EDA utilizando DRC. Neste modelo somente são permitidas conexões verticais ou horizontais. Geralmente direções preferenciais são assinaladas às camadas. Isso é feito para fazer o problema mais simples de ser modelado e resolvido, reduzindo o espaço de busca.

O modelo sem grade definida não segue a grade de roteamento e assim pode usar diferentes larguras e espaçamentos de fios. A grande vantagem desse modelo se deve a grande flexibilidade, que possibilita que o circuito seja mais otimizado.

Na Figura 5.2 são mostrados dois exemplos de roteamento, um de cada modelo.

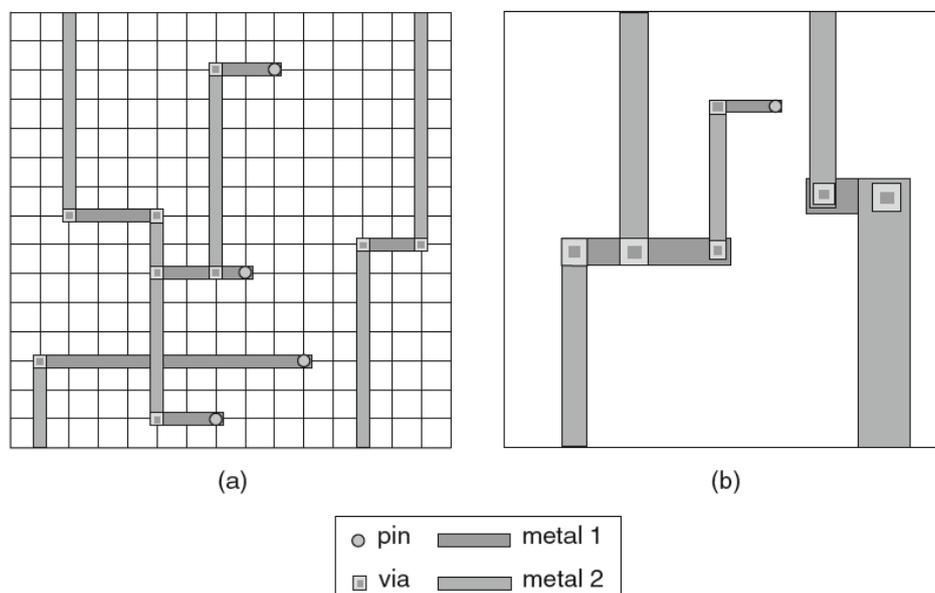


Figura 5.2. Dois modelos de roteamento detalhado: (a) baseado em grade; (b) sem grade definida ([HUA2009]).

Neste trabalho o modelo utilizado é o baseado em grade, que foi escolhido justamente pela eficiência e facilidade de implementação, além de ser o mais utilizado em aplicações práticas [SAN2006].

Neste modelo assim que uma grade de roteamento é definida o problema de roteamento é abstraído em problemas geométricos, no que é conhecido como roteamento simbólico.

O GLDR realiza o roteamento de estruturas de grafo, mais especificamente transpostas para uma estrutura de grade.

5.2 Técnica de seleção do nodo origem

Analisando os resultados obtidos da comparação com o RotDL [FLA2004] (padrão de comparação para roteamento detalhado usado neste trabalho) foi possível notar que era necessário melhorar o tempo de execução do algoritmo. Com esse objetivo várias técnicas foram estudadas.

Uma que obteve bons resultados é a que seleciona como nodo inicial (origem) para o roteador de sinal o nodo mais próximo ao centro geométrico da rede. A razão para essa melhora é que começando de um nodo mais central da rede menos nodos em média são expandidos para completar o roteamento.

Essa técnica também foi aplicada a frações do número total de redes. Os resultados obtidos não foram tão bons quanto aos obtidos quando se aplicava a todas as redes.

5.3 Roteamento *intracell*

O roteamento *intracell* é realizado utilizando o algoritmo Pathfinder para gerenciar os conflitos entre as redes da célula e para roteamento de sinal é utilizado o algoritmo de Lee [LEE1961].

Para otimizar os resultados são utilizados nodos de Steiner. Para adicionar nodos de Steiner à rede é utilizada a abordagem do algoritmo Iterated 1-Steiner [KAH1992].

A otimização é utilizada depois de uma solução ser obtida utilizando o algoritmo Pathfinder. O processo ocorre da seguinte maneira:

- o roteamento de cada rede é desfeito uma rede por vez;
- nodos de steiner são inseridos nesta rede se isso trouxer diminuição de *wirelength*;
- a rede é roteada novamente respeitando o roteamento das outras redes, assim não é necessário lidar com conflitos porque eles não acontecem;
- o processo continua até que não sejam mais adicionados nodos de Steiner a nenhuma das redes.

6 ALGORITMOS UTILIZADOS

No desenvolvimento dos algoritmos foi utilizada a linguagem de programação C++ e as estruturas de dados utilizadas são as da STL. A STL é uma biblioteca altamente otimizada e reusável que está disponível na quase totalidade dos compiladores C++ atuais.

Os algoritmos seguintes são os básicos na implementação dos três tipos de roteadores: *intracell*, detalhado e global.

6.1 A*

O algoritmo A* é usado para encontrar o caminho de menor custo de um nodo inicial para cada um dos nodos destino. Assim é encontrada a MST de cada uma das redes. A chave do algoritmo é a função custo $F(x)$ que leva em conta a distância da origem $G(x)$ e o custo estimado até o destino $H(x)$, como mostrado na fórmula abaixo:

$$F(x) = G(x) + H(x)$$

Usando esta fórmula, os nodos vizinhos mais próximos do objetivo são expandidos primeiro. Isso ocasiona uma diminuição no número de nodos visitados necessários para encontrar o objetivo quando comparado com o algoritmo de Lee. No caso do Lee somente é levada em conta a distância da origem $G(x)$ no cálculo de $F(x)$.

$H(x)$ tem que ser uma heurística admissível e por essa razão não pode ser superestimada. Isso é necessário para garantir que o menor caminho possa ser encontrado.

Quanto mais próximo $H(x)$ for da distância real para o objetivo, menos expansões serão necessárias para encontrar o caminho de menor custo entre os nodos. Este custo é calculado utilizando a distância Manhattan entre o nodo atual e o nodo destino. A distância Manhattan é obtida somando as distâncias nos 3 eixos da grade tridimensional entre os nodos. Na implementação, a distância entre dois nodos adjacentes foi considerada como unitária.

Em relação a busca foi utilizado o modelo de múltiplas origens e múltiplos destinos. Isso significa que começamos com um nodo origem e o nodo destino atingido é adicionado como origem para a busca e esta é reiniciada até que todos os nodos destino tenham sido atingidos.

Na Figura 6.1 é mostrado o pseudocódigo do algoritmo A* utilizado neste trabalho.

Algoritmo A*

- 1) Adiciona s (nodo origem) a lista de abertos.
- 2) Se a lista de abertos está vazia, terminar porque não existe solução.
- 3) Selecionar da lista de abertos o nodo x com o menor $F(x)$ (Se houver empate resolver pelo valor de $H(x)$).
- 4) Retira x da lista de nodos abertos e coloca na lista de nodos fechados.
- 5) Se $x=t$ (destino), terminar com sucesso e realiza o backtrace.
- 6) Senão, expandir x (visitar os vizinhos).
- 7) Para cada nodo v vizinho de x : Se v não estiver nem na lista de abertos nem na lista de fechados calcular $f(x)$;
- 8) Volta para o passo 2.

Figura 6.1: Pseudocódigo A*

Na Figura 6.2 temos um exemplo que mostra duas buscas utilizando Lee e A* em um mesmo espaço de busca. O quadrado identificado por S é o nodo origem e o identificado por T é o nodo destino. Os quadrados preenchidos são os nodos expandidos durante a execução dos algoritmos de busca. Usando Lee, um total de 85 nodos foram visitados até o destino ser atingido. Usando A*, somente 33 nodos foram visitados até que o nodo destino pudesse ser atingido.

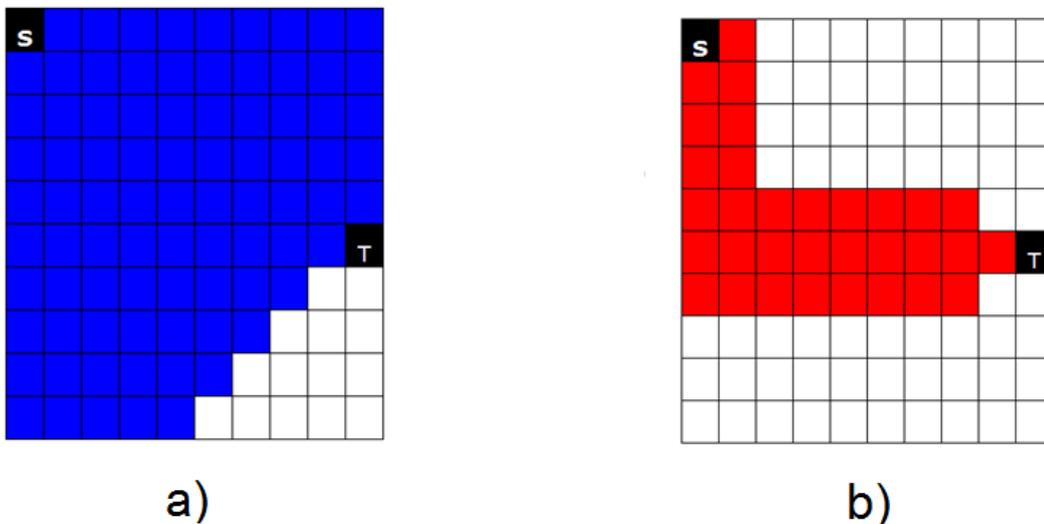


Figura 6.2: Pesquisa utilizando: (a)Lee e (b)A*.

6.2 Pathfinder

É um algoritmo de roteamento usado na negociação de congestionamento de redes ou NCR. Primeiramente desenvolvido para ser utilizado em FPGA's, tem sido utilizado em várias ferramentas de roteamento acadêmicas como em [PAN2006], [CHA2008] e [ROY2007]. É normalmente utilizado somente no roteamento global, onde as arestas têm capacidades não unitárias, mas seu uso para roteamento detalhado não foi muito explorado até o momento.

É usado para gerenciar conflitos surgidos do roteamento independente de cada rede, estabelecendo penalidades. Se um conflito acontece, o roteador ajusta o custo de utilização do nodo congestionado. Isso é feito de acordo com uma das fórmulas abaixo dependendo do objetivo de otimização:

$$C_n = B_n + (H_n * P_n) \quad \text{Equação 1}$$

$$C_n = (B_n + H_n) * P_n \quad \text{Equação 2}$$

Estas fórmulas consideram o custo base B_n , o histórico de congestionamento nas iterações anteriores H_n e a quantidade de redes utilizando esse nodo na iteração atual P_n .

Durante a primeira iteração, P_n é inicializado com 1, assim penalidades não são estabelecidas independentemente de quantos sinais ocupem o nó. Nas iterações posteriores, P_n é incrementado gradualmente em caso de congestionamento. Isso faz com que algumas redes desistam e tentem encontrar uma rota de custo menor.

A chave do algoritmo é o fator H_n . A cada iteração que n é compartilhado, H_n é incrementado lentamente. O efeito desse fator é de aumentar permanentemente o custo da utilização dos nós congestionados de forma que rotas alternativas sejam exploradas.

A métrica P_n é importante para acelerar a execução do algoritmo. Ela insere um fator de ordem na realização das conexões que serve como critério de desempate entre redes que disputam o mesmo nó e também diminui as chances de duas ou mais conexões desistirem do nó ao mesmo tempo. Entretanto, um acréscimo muito abrupto de P_n , pode fazer com que as redes desistam muito rapidamente de nós congestionados, eliminando a competição e tornando o algoritmo demasiadamente sensível à ordem de realização das conexões tal como o esquema de *rip-up e re-route* padrão.

Todos esses fatores contribuem para que o roteador de sinal procure por caminhos alternativos para atingir o destino e assim fazem o algoritmo convergir para uma solução aceitável.

A equação 1 havia sido proposta em [MCM1995], enquanto que a Equação 2 é uma alteração proposta por [ROY2007] em relação ao algoritmo original do Pathfinder.

O processo acontece de forma incremental e termina quando o roteador global não encontra mais conflitos entre as redes utilizando *rip-up e reroute*. O processo de *rip-up e reroute* nada mais é do que desfazer as redes que apresentam conflitos e força-las a alterar a topologia de roteamento para que seja possível o roteamento de ambas redes sem a existência de conflitos. Nas iterações posteriores, somente são roteadas as redes que apresentam conflitos.

Na Figura 6.3 é mostrado um roteamento de duas redes com dois nodos: S (origem) e T (destino). Na parte (a), a rede vermelha é conectada primeiro o que torna o roteamento da rede verde impossível. Na parte (b) é aplicado *rip-up* e a rede verde é roteada primeiro. Na parte (c) a rede vermelha é roteada novamente (reroute) respeitando o roteamento da rede verde sendo possível rotear as duas redes.

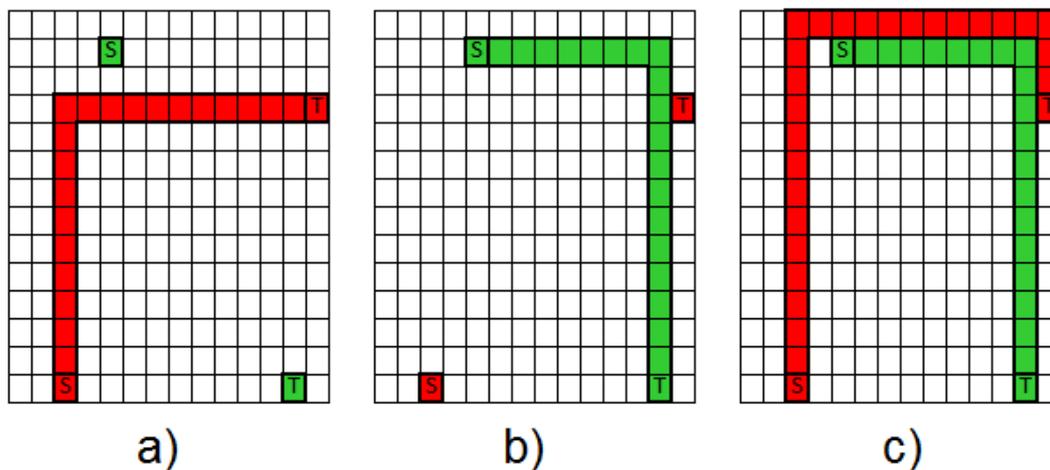


Figura 6.3: Rip-up e reroute

6.3 Iterated 1-Steiner Node

Depois de uma solução factível ser encontrada durante o roteamento *intracell* uma fase de otimização é executada na tentativa de reduzir o *wirelength* de cada rede procurando por nodos especiais conhecidos como nodos de Steiner.

Esses nodos, quando adicionados à rede original, têm a propriedade de reduzir o custo de interconexão, levando o roteador de sinal a usar caminhos que podem ser compartilhados entre os diferentes terminais.

A abordagem utilizada para adicionar nodos de Steiner no roteador *intracell* é a apresentada no algoritmo Iterated 1-Steiner Node [KAH1992], que calcula iterativamente o melhor nodo de Steiner adicionando-o a rede.

7 INTEGRAÇÃO ROTEAMENTO GLOBAL E DETALHADO

Para realizar a integração do roteador global com o detalhado foi utilizada uma função que determina se a região em que está o nodo sendo expandido está entre as regiões válidas para o roteamento da rede atual.

Nesta abordagem não é feito o roteamento somente dentro de cada região, mas também é necessário fazer as conexões entre as regiões respeitando os resultados obtidos na etapa do roteamento global. A única diferença dessa abordagem para a que utiliza somente roteamento detalhado sobre o circuito é que o roteamento é direcionado pelos caminhos obtidos no roteamento global.

Nas figuras seguintes temos um exemplo do fluxo de roteamento realizado para um circuito de tamanho 20x20x2 com tamanho de cada bin de 4x4 e capacidade de conexão entre os bins igual a 3. Nesse cenário com um total de 6 redes sendo roteadas uma foi selecionada para análise.

Na Figura 7.1 temos o roteamento global da rede analisada com os Global Bins identificados. Esses Global Bins correspondem às regiões identificadas para cada nodo na Figura 7.2, que mostra o roteamento detalhado. Podemos notar uma topologia semelhante já que o roteamento detalhado se baseia na configuração do roteamento global. Enquanto que na Figura 7.3 temos o roteamento detalhado sem a utilização de um roteador global.

Na Figura 7.1 os nodos *source* para o roteamento global são as regiões 8 e 11. Assim conforme a Figura 7.1, o roteamento detalhado pode passar pelas regiões 8, 11, 12 e 13. Isso é verificado na Figura 7.2, onde os nodos *source* que estão identificados por um X se conectam utilizando nodos localizados nas regiões especificadas no roteamento global.

Ainda podemos verificar que o roteamento utilizando a abordagem global+detalhado tem resultado equivalente ao obtido utilizando somente o roteamento detalhado para esta rede.



Figura 7.1. Roteamento Global de uma rede de 2 nós.

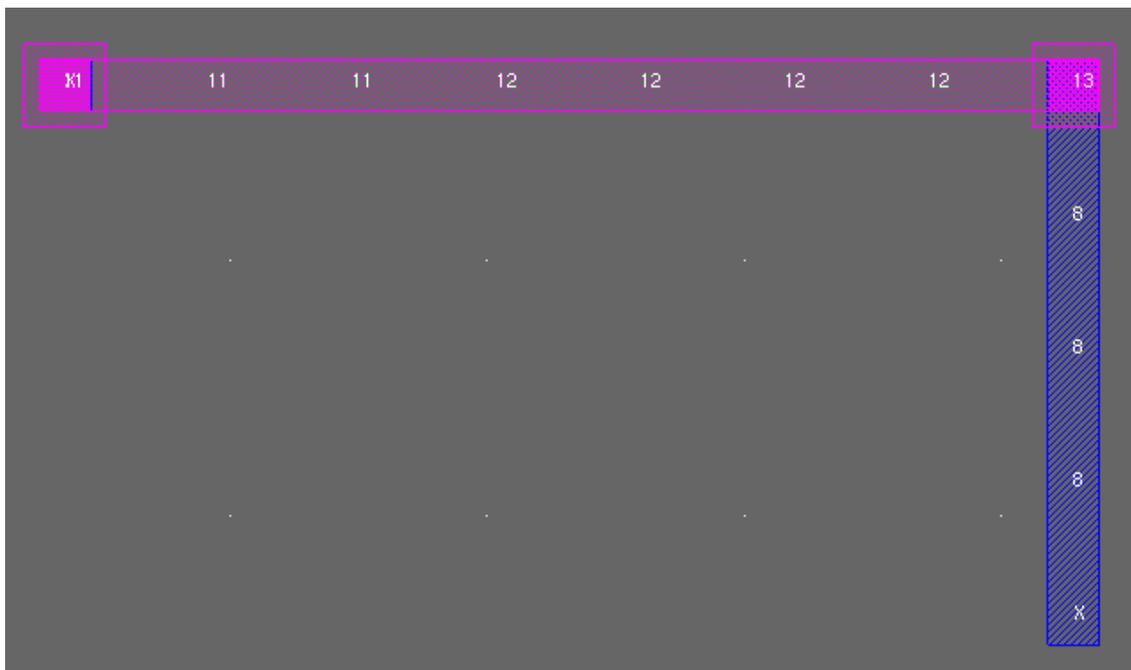


Figura 7.2. Roteamento Detalhado obtido usando resultados do Roteamento Global.

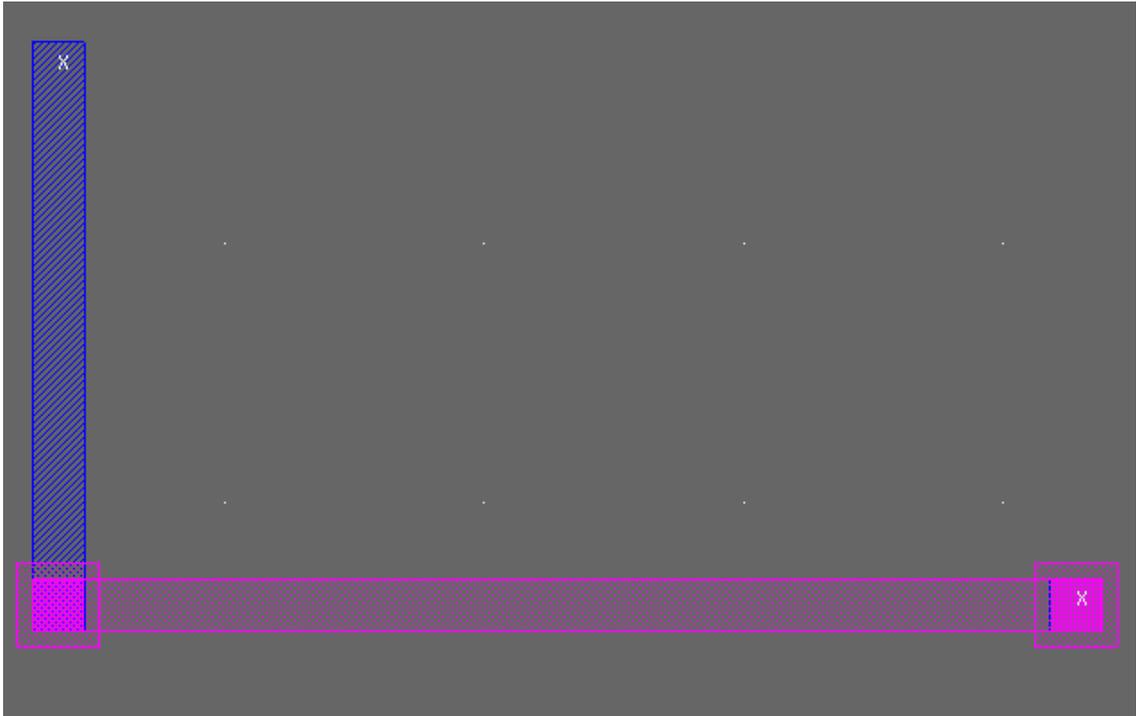


Figura 7.3. Roteamento Detalhado.

8 DADOS DA IMPLEMENTAÇÃO

8.1 Tamanho de código implementado

Os dados da Tabela 8.1 com relação ao número de linhas de código implementadas se referem somente as tarefas diretamente ligadas ao roteamento excluindo dados com relação as outras etapas prévias e posteriores e interfaces da ferramenta.

Tabela 8.1. Número de linhas de código da implementação.

Roteador Intracell	665
Roteador Detalhado	587
Roteador Global	852
Total	2104

9 RESULTADOS

Para gerar os resultados foi utilizada a ferramenta ASTRAN em um PowerPC G5 2GHz com 4GB DDR SDRAM.

9.1 Roteamento Intracell

A comparação foi feita entre a otimização utilizando nodos de Steiner desenvolvida e o algoritmo original sem otimizações. As métricas de comparação são o *wirelength* e o tempo de execução.

As células utilizadas para teste são células reais que são descritas utilizando a linguagem SPICE na ferramenta ASTRAN e sobre as quais são realizadas todas as operações necessárias para obtenção do leiaute. Na Tabela 9.1 temos os resultados em relação à custo de interconexão e tempo de execução para realizar o roteamento de cada célula a partir do netlist usando o algoritmo com e sem otimização.

Tabela 9.1. Comparação do roteador intracell com e sem otimização

Circuito	Grid	Custo			Tempo(s)		
		S/Steiner	C/Steiner	Dif(%)	S/Steiner	C/Steiner	Dif(%)
MUX2x1	50x50	2945	2926	-0.645	0.116	0.238	105.172
XNR30	50x50	3769	3763	-0.159	0.151	0.523	246.358
OAI311	90x90	3847	3842	-0.13	0.076	0.443	482.895
ADD22	100x100	3656	3651	-0.137	0.117	0.472	303.419
OAI444	100x100	10309	10304	-0.049	1.27	3.39	166.929
XCFE	100x100	5538	5532	-0.108	0.463	1.007	117.495
NOR33	100x100	4236	4226	-0.236	0.214	2.124	892.523
Média				-0.21			330.7

Podemos ver que o custo de interconexão diminui em 0,21% mas o tempo de execução aumenta consideravelmente (~330%). Entretanto, é possível notar que este se manteve baixo mesmo para células de maior tamanho.

9.2 Roteamento Global

Para avaliar a implementação desenvolvida serão utilizadas duas métricas: *wirelength* e tempo de execução. O *wirelength* mede o comprimento das conexões utilizadas no roteamento enquanto o tempo de execução mede o tempo que o algoritmo demorou a apresentar o resultado.

Primeiramente foi necessário escolher o tamanho do bin, o valor selecionado foi de 10x10. A partir disso era imperativo escolher também a capacidade global entre as interfaces de bins e com esse objetivo um teste foi desenvolvido para encontrar esse valor. Neste teste o tamanho da grade é de 1000x1000, e os resultados para cada valor de capacidade global são comparados com relação ao tempo de execução e custo total de interconexão da solução. O tempo de execução é a soma dos tempos de execução do roteador global e do detalhado. O custo total de interconexão é o obtido no roteamento detalhado guiado pelo roteador global. Os resultados estão nas Tabelas 9.2 e 9.3, as linhas tracejadas indicam que naquele teste não foi possível solucionar com a capacidade global determinada.

Tabela 9.2. Custo total de interconexão para cada capacidade global.

# Nets	Custo de Interconexão					
	4	5	6	7	8	9
400	2051372	2009997	2000988	1999687.0	1998471	1998452
600	-	-	3173650	3123790.0	3103500	3100361
800	-	-	-	-	4288369	4227505
Média	-	-	-	-	3130113	3108773
Normalizada	-	-	-	-	1.007	1

Tabela 9.3. Tempo de execução para cada capacidade global.

# Nets	Tempo de execução(s)					
	4	5	6	7	8	9
400	16.581	7.284	7.253	6.728	9.7	8.269
600	-	-	41.447	20.944	28.91	39.842
800	-	-	-	-	102.077	195.039
Média	-	-	-	-	46.9	81.05
Normalizada	-	-	-	-	1	1.729

Analisando os resultados obtidos nas duas configurações válidas é possível notar que o custo de interconexão para a solução utilizando capacidade global igual a 9 é 0,7% melhor do que a solução utilizando capacidade global 8. Entretanto, em relação ao tempo de execução a solução com capacidade global igual a 8 tem resultado 73% melhor em relação à outra solução válida. Assim, nos testes posteriores a capacidade global foi determinada como 8.

Na Tabela 9.4 são mostrados os resultados para roteamento global utilizando os parâmetros previamente estabelecidos de tamanho de Bin e capacidade global.

Tabela 9.4. Roteamento global de circuitos de teste.

Grid	# Nets	Custo	Tempo	Tentativas
1000x1000	400	1305252	0.257	3
1000x1000	600	2913364	0.616	4
1000x1000	800	5220532	19.127	52
2000x2000	800	5301828	1.012	3
2000x2000	1200	11830884	2.840	4
2000x2000	1600	20605428	331.170	127
3000x3000	1200	12699076	2.872	3
3000x3000	1800	26414484	6.933	4
3000x3000	2400	47193988	2340.630	243
4000x4000	1600	21944292	5.273	4
4000x4000	2400	46816172	13.323	5
4000x4000	3200	82335680	3061.020	204

Na **Erro! Fonte de referência não encontrada.** é mostrado um gráfico da complexidade em relação ao tempo de execução do roteador global.

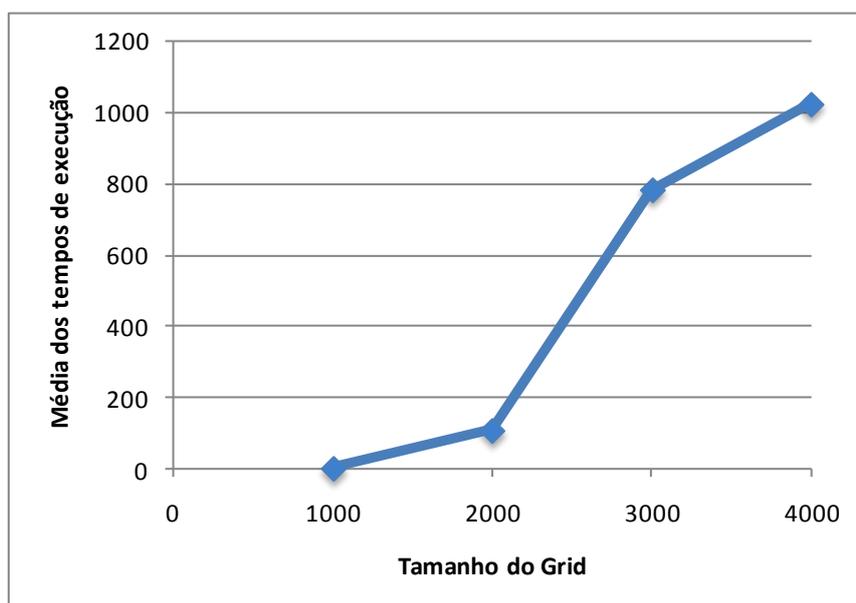


Figura 9.1. Gráfico da complexidade em relação a média dos tempos de execução do roteador global para cada tamanho de grade.

9.3 Roteamento Detalhado

São utilizadas duas métricas para comparar os resultados: *wirelength* e tempo de execução.

O algoritmo usado para comparação é o Rotdl [FLA2004]. É um Algoritmo de roteamento detalhado desenvolvido no GME baseado em *maze routing* com técnicas de *rip-up e reroute* para lidar com bloqueios.

Para ilustrar os resultados, as Tabelas 9.5, 9.6 e 9.7 são apresentadas. A Tabela 9.5 compara os resultados da técnica de seleção do nodo origem com a abordagem original, ambas utilizando a equação 2 do Pathfinder. As Tabelas 9.6 e 9.7 comparam os resultados da abordagem utilizando a técnica de seleção do nodo origem e o RotDL para os mesmos casos de teste.

A coluna #Nets representa o número de redes de cada circuito. As redes são compostas por uma quantidade de nodos variando entre 2 e 5. A comparação entre diferentes abordagens é mostrada por frações da abordagem original. Entre parênteses em cada identificador da coluna está a equação utilizada (E1 ou E2) e se a técnica de seleção da origem é utilizada (+S). Três execuções de cada caso de teste foram executadas e os resultados são a média dessas execuções.

De acordo com a Tabela 9.5, os melhores resultados são obtidos quando selecionamos como nodo origem o mais próximo ao centro geométrico da rede para todas as redes. Nesta configuração o custo de interconexão é aproximadamente o mesmo e o tempo de execução é 3,4% menor que o da abordagem original.

Tabela 9.5. Comparação entre a técnica de seleção da origem e a abordagem original.

Grid	# Nets	Custo de Interconexão		Tempo(s)		Comparação	
		GDLR(E2)	GDLR(E2+S)	GDLR(E2)	GDLR(E2+S)	Custo	Tempo
400x400	120	252034	252352	0.642	0.693	1.001	1.081
400x400	200	415454	413676	3.736	3.356	0.996	0.898
400x400	280	590423	591074	12.965	11.396	1.001	0.879
600x600	180	531329	529873	1.435	1.442	0.997	1.005
600x600	300	917201	916476	9.268	8.876	0.999	0.958
600x600	420	1335346	1336206	53.562	44.615	1.001	0.833
800x800	240	997996	995030	3.929	3.776	0.997	0.961
800x800	400	1648364	1648971	23.230	24.608	1.000	1.059
800x800	560	2324052	2327460	98.877	100.066	1.001	1.012
1000x1000	300	1520575	1517268	5.232	5.701	0.998	1.090
1000x1000	500	2584728	2577709	46.963	43.629	0.997	0.929
1000x1000	700	3692922	3696869	208.526	185.193	1.001	0.888
Média						0.999	0.966

Depois de incorporar a técnica de seleção do nodo origem à ferramenta GDLR, os resultados obtidos com as duas equações do Pathfinder são comparadas com a ferramenta RotDL. Os resultados dessa comparação são mostrados nas tabelas 8.6 e 8.7. Os resultados utilizando a equação 1 estão na Tabela 9.6 e os resultados utilizando a equação 2 estão na Tabela 9.7.

As equações obtêm resultados similares em termos de custo de interconexão enquanto que a segunda equação obtém melhores resultados para tempo de execução. A diminuição de tempo de execução obtida utilizando a equação 2 é de aproximadamente 50% quando comparada ao RotDL.

Tabela 9.6. Comparação da equação 1 do Pathfinder com o RotDL.

Grid	# Nets	Custo de Interconexão		Tempo(s)		Comparação	
		RotDL	GDLR(E1)	RotDL	GDLR(E1)	Custo	Tempo
400x400	120	246731	248387	5	0.778	1.007	0.156
400x400	200	404844	422531	8	15.387	1.044	1.923
400x400	280	576316	592037	11	59.657	1.027	5.423
600x600	180	521273	548652	16	5.954	1.053	0.365
600x600	300	898972	904999	25	18.319	1.007	0.723
600x600	420	1304887	1328778	37	194.155	1.018	5.295
800x800	240	981389	954945	39	4.458	0.973	0.113
800x800	400	1618263	1638575	66	140.199	1.013	2.135
800x800	560	2276922	2332361	95	551.929	1.024	5.789
1000x1000	300	1492669	1566274	107	5.303	1.049	0.050
1000x1000	500	2539368	2679106	197	265.340	1.055	1.347
1000x1000	700	3631745	3516838	241	932.998	0.968	3.866
Média						1.020	2.265

Tabela 9.7. Comparação da equação 2 do Pathfinder com o RotDL.

Grid	# Nets	Custo de Interconexão		Tempo(s)		Comparação	
		RotDL	GDLR(E1)	RotDL	GDLR(E1)	Custo	Tempo
400x400	120	246731	252352	5	0.693	1.023	0.139
400x400	200	404844	413676	8	3.356	1.022	0.420
400x400	280	576316	591074	11	11.396	1.026	1.036
600x600	180	521273	529873	16	1.442	1.016	0.088
600x600	300	898972	916476	25	8.876	1.019	0.350
600x600	420	1304887	1336206	37	44.615	1.024	1.217
800x800	240	981389	995030	39	3.776	1.014	0.096
800x800	400	1618263	1648971	66	24.608	1.019	0.375
800x800	560	2276922	2327460	95	100.066	1.022	1.050
1000x1000	300	1492669	1517268	107	5.701	1.016	0.053
1000x1000	500	2539368	2577709	197	43.629	1.015	0.221
1000x1000	700	3631745	3696869	241	185.193	1.018	0.767
Média						1.020	0.484

9.4 Roteamento Global + Roteamento Detalhado

Nos testes desenvolvidos o tamanho do bin foi fixado em 10x10 e a capacidade global é igual a 8, como foi determinado na seção 9.2.

A coluna #Nets representa o número de redes de cada circuito. As redes são compostas por uma quantidade de nodos variando entre 2 e 5. A comparação entre diferentes abordagens é mostrada por frações da abordagem original. O roteador detalhado utilizado é o GDLR usando a equação 2 do Pathfinder e a técnica de seleção da origem. A comparação é feita entre a abordagem de dois estágios (Roteamento

Global+Detalhado) com a aplicação do roteador detalhado ao circuito. Três execuções de cada caso de teste foram executadas e os resultados são a média dessas execuções. Os resultados estão nas tabelas 9.8, 9.9 e 9.10.

Tabela 9.8. Comparação do custo de interconexão.

Grid	# Nets	Custo de Interconexão		Comparação
		D	G+D	
1000x1000	400	2000539	1998471	0.999
1000x1000	600	3091173	3103500	1.004
1000x1000	800	4185547	4288369	1.025
2000x2000	800	8002737	8015186	1.002
2000x2000	1200	12355111	12405240	1.004
2000x2000	1600	16451583	16835261	1.023
3000x3000	1200	18520069	18574948	1.003
3000x3000	1800	27632374	27707416	1.003
3000x3000	2400	-	38333677	-
4000x4000	1600	-	32662194	-
4000x4000	2400	-	-	-
4000x4000	3200	-	67135690	-
Média				1.008

Tabela 9.9. Comparação do tempo de execução.

Grid	# Nets	Tempo de Execução		Comparação
		D	G+D	
1000x1000	400	4.078	9.591	2.352
1000x1000	600	9.925	29.238	2.946
1000x1000	800	72.162	104.332	1.446
2000x2000	800	17.273	31.110	1.801
2000x2000	1200	37.946	164.811	4.343
2000x2000	1600	2876.670	879.209	0.306
3000x3000	1200	44.249	76.669	1.733
3000x3000	1800	129.169	355.976	2.756
3000x3000	2400	-	4033.460	-
4000x4000	1600	-	234.994	-
4000x4000	2400	-	-	-
4000x4000	3200	-	6388.870	-
Média				2.210

Tabela 9.10. Comparação das tentativas utilizadas.

Grid	# Nets	Tentativas		Comparação
		D	G+D	
1000x1000	400	5	54	10.800
1000x1000	600	48	141	2.938
1000x1000	800	107	70	0.654
2000x2000	800	11	40	3.636
2000x2000	1200	82	250	3.049
2000x2000	1600	357	155	0.434
3000x3000	1200	19	57	3.000
3000x3000	1800	214	178	0.832
3000x3000	2400	-	172	-
4000x4000	1600	-	285	-
4000x4000	2400	-	-	-
4000x4000	3200	-	373	-
Média				3.168

A partir dos resultados podemos notar que não há grande diferença no custo total de interconexão entre as soluções obtendo a abordagem de dois estágios custo 0,7% maior que a abordagem de roteamento detalhado.

Em relação a tempo de execução e principalmente número de tentativas podemos ver que os resultados vão ficando melhores com o aumento do tamanho da grade e principalmente da quantidade de redes para um mesmo tamanho de grade. Além disso, em alguns circuitos somente a abordagem de dois estágios conseguiu realizar o roteamento. Ainda assim devido a limitações de implementação e de ligação do roteador global com o detalhado os resultados em média são piores. É muito provável que caso fosse possível rotear circuitos ainda maiores, os resultados das comparações fossem mais satisfatórios. Isso acontece devido a limitações de memória e de tempo de execução.

10 CONCLUSÕES

Na primeira etapa do trabalho todo o fluxo de roteamento foi estudado para possibilitar o desenvolvimento de algoritmos específicos para cada problema. Também foi importante nesta fase a familiarização com as metodologias de teste adotadas em outros trabalhos correlatos.

O algoritmo para roteamento *intracell* [LEO2008] incluindo uma fase de otimização obtém ganho de 0,2% em relação à solução sem otimização desempenhando um papel importante na geração de bibliotecas de células melhores.

O algoritmo para roteamento detalhado [LEO2009] [LEO2010] apresenta resultados satisfatórios restando serem aplicadas técnicas que possibilitem que seus resultados sejam comparáveis a algoritmos estado da arte. Os resultados indicam a viabilidade do algoritmo quando comparado ao Rotdl, obtendo circuitos com *wirelength* equivalente em 50% menos tempo. Entre as ideias de trabalhos futuros se destaca a que direciona a busca para utilizar nodos de Steiner na procura do caminho do nodo origem para o nodo destino.

O roteador global desenvolvido é bastante eficiente, entretanto a conexão com o roteador detalhado ainda é incipiente. Assim entre os trabalhos futuros se destacam uma melhor integração do roteamento global com o detalhado.

Uma das formas para se atingir esse objetivo é o tratamento de cada região em separado através da definição de crosspoint assignment (CPA) entre as regiões que permitam uma completa independência no roteamento de cada região em relação as demais. A técnica de CPA cria nodos de conexão (pinos virtuais) que são definidos nas interfaces entre as regiões. Para ligar duas regiões ou *switchboxes* basta conectar pinos virtuais entre estas regiões. Essa independência acabaria com limitações de memória e de complexidade que impedem o roteamento de circuitos ainda maiores. Nesses circuitos as vantagens da abordagem roteamento global+detalhado seriam mais evidentes. A única desvantagem dessa abordagem é a diminuição na liberdade do roteamento detalhado levando possivelmente a soluções com maior *wirelength* mas com ganhos consideráveis em relação ao tempo de execução.

Apesar disso já é possível notar que a abordagem de dois estágios é muito promissora apresentando vantagens principalmente em circuitos mais congestionados e de maior complexidade. Mas a principal vantagem se mostra na questão da roteabilidade que é maior utilizando essa abordagem.

REFERÊNCIAS

- [CHA2008] Y.J. Chang, Y.T. Lee, T.C Wang, “NTHU-Route 2.0: A Fast and Stable Global Router”, IEEE/ACM International Conference on Computer-Aided Design, 338-343, San Jose, EUA, 2008.
- [EDA2010] EDA Consortium. Disponível em: < <http://www.edac.org>>. Acessado em: junho de 2010.
- [FLA2004] G. Flach, R. Hentschke, R. Reis. “Algorithms for improvement of RotDL router”. in South Symposium On Microeletronics, Ijuí, 2004.
- [GER1998] S. Gerez, “Algorithms for VLSI Design Automation”, 1ª edição, 1998.
- [HAR1968] P. E. Hart, N. J. Nilsson, “A Formal basis for the heuristic determination of minimum cost paths”, IEEE Transactions of Systems Science and Cybernetics, vol. SSC-4, número 2, Pp. 100-107, 1968.
- [HUA2009] C. Huang-Yu, C. Yao-Wen, “Global and detailed routing”, cap. 12, pp 687-749, Taipei, 2009.
- [ISP2008] ISPD 2008 GLOBAL ROUTING CONTEST. ISPD 2008 Global Routing Contest. Disponível em: <<http://www.sigda.org/ispd2008/contests/ispd08rc.html>>. Acessado em: outubro 2009.
- [JOH1997] M. Johann, “Estrutura de Roteamento em Circuitos VLSI”, Tese de mestrado, 1997.
- [JOH2000a] M. Johann, A. Caldwell, A. Kahng, R. Reis, “A New Bidirectional Heuristic Shortest Path Search Algorithm”, International ICSC Congress on Artificial Intelligence and Applications, 2000.
- [JOH2000b] M. Johann, A. Caldwell, R. Reis, A. Kahng, “Admissibility Proofs for the LCS* Algorithm”, IBERAMIA-SBIA, 2000.
- [JOH2001] M. Johann, “Novos Algoritmos para Roteamento de Circuitos VLSI”. Tese de doutorado, UFRGS, 2001.
- [KAH1992] A. Kahng, “A new class of iterative Steiner tree heuristics with good performance,” IEEE Trans. Computer-Aided Design, Volume 11, Pp. 893-902, Los Angeles, 1992.
- [KAS2000] R. Kastner, E. Bozorgzadeh, M. Sarrafzadeh, “Predictable Routing”, IEEE/ACM international conference on Computer-aided design, pp. 110-114, San Jose, 2000.

- [LEE1961] C. Y. Lee, "An algorithm for path connections and its applications", IRE Transactions on Electronic Computer, vol. EC-10, number2, Pp. 364-365,1961.
- [LEO2008] C. Leonhardt, L. Saldanha, A. Ziesemer, R. Reis, "Improving Pathfinder using Iterated 1-Steiner algorithm", XXIII South Symposium on Microelectronics, Bento Gonçalves, 2008.
- [LEO2009] C. Leonhardt, A. Ziesemer, R. Reis, "Improved Detailed Routing using Pathfinder and A*", XXIV South Symposium on Microelectronics, Pelotas, Brasil, 2009.
- [LEO2010] C. Leonhardt, A. Ziesemer, R. Reis, "GDLR: a Detailed Routing Tool", XXV South Symposium on Microelectronics, Porto Alegre, 2010.
- [MCM1995] L. McMurchie, C. Ebeling, "Pathfinder: a negotiation-based performance-driven router for FPGAs", International Symposium on Field Programmable Gate Arrays, Monterrey, México, 1995.
- [MOF2008] M. D. Moffitt, J. A. Roy, I. L. Markov, "The Coming of Age of (Academic) Global Routing", International Symposium on Physical Design (ISPD), pp. 148-155, Portland, EUA, 2008.
- [NES2010] F. Nesello, C. Leonhardt, A. Ziesemer, R. Reis, "ASTRAN: Ferramenta para Geração Automática de Circuitos VLSI", IBERCHIP, Foz do Iguaçu, 2010.
- [PAN2006] C. M. Pan, C. Chu, "FastRoute: A step to integrate global routing into placement", in International Conference on Computer Aided Design", Pp. 464-471, San Jose, EUA, 2006.
- [PRE1988] M. Baeder, D. Scott, "Physical Design Automation of VLSI Systems", cap. 5, 1988.
- [RAG1991] P. Raghavan, C. Thompson, "Multiterminal Global Routing: a Deterministic Approximation Scheme", Algorithmica, Vol 6, No. 1, 73-82, 1991.
- [ROY2007] J. A. Roy, I. L. Markov, "High-performance routing at the nanometer scale," in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, pp. 496-502, San Jose, EUA, 2007.
- [SAX2007] P. Saxena, R. Shelar, S. Sapatnekar, "Routing Congestion in VLSI Circuits", 1ª edição, 2007.
- [SHE1999] N. Sherwani, "Algorithms for VLSI Physical Design Automation", 3ª edição, 1999.
- [SWA2008] W. Swartz, 'Issues in Global Routing", International Symposium on Physical Design, 2008.
- [WXW2009] WXWIDGETS. WxWidgets. <<http://www.wxwidgets.org/>>. Acessado em: outubro 2009.
- [ZIE2007] A. Ziesemer, C. Lazzari, R. Reis, "Transistor level automatic layout generator for non-complementary CMOS cells", in Very Large Scale Integration – System on Chip(VLSI-SOC), Pp. 116-121, Atlanta, EUA, 2007.
- [ZIE2009] A. M. Ziesemer. ICPD. <<http://www.inf.ufrgs.br/~amziesemerj/icpd/>>. Acessado em: outubro 2009.

[ZHA2008] Y. Zhang, Y. Xu, C. Chu, "FastRoute3.0: A fast and high quality global router based on virtual capacity", IEEE/ACM International Conference on Computer-Aided Design, pp.344-349, San Jose, EUA, 2008.