

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TRAPP
Uma Ferramenta
para
Particionamento/Posicionamento
de
Células para a
Metodologia TRANCA

por

Paulo Armando Schermer

Dissertação submetida como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Ricardo Augusto da Luz Reis
Orientador

Porto Alegre, Dezembro de 1995.



UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Schermer, Paulo Armando
 TRAPP uma Ferramenta para
 Particionamento/Posicionamento de Células para a
 Metodologia TRANCA / Paulo Armando Schermer. — Porto
 Alegre: CPGCC da UFRGS, 1995.

Dissertação (mestrado) — Universidade Federal do
 Rio Grande do Sul. Curso de Pós-Graduação em Ciência da
 Computação, Porto Alegre, 1995. Reis, Ricardo Augusto da
 Luz, Orient.

1. Microeletrônica. 2. Projeto VLSI. 3. Ferramentas
 PAC. I Reis, Ricardo Augusto da Luz. II. TRAPP uma
 Ferramenta para Particionamento/Posicionamento de Células
 para a Metodologia TRANCA.

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA	
N.º CHAMADA 621.38-181.4(043) S326T	N.º REG: 32657
ORIGEM: D	DATA: 09.09.96
CAIA: 27/8 96	PREÇO: R\$ 30,00
FUNDO: II	FORN.: II

Microeletrônica -
SBU/II

Síntese automa-
tica

CNPq 3.04.03.00-6

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Hégio Trindade

Pró-Reitor de Pesquisa a Pós-Graduação: Prof. Cláudio Scherer

Diretor do Instituto de Informática: Prof. Roberto Tom Price

Coordenador do CPGCC: Prof. José Palazzo Moreira de Oliveira

Bibliotecária Chefe do Instituto de Informática: Zita Prates de Oliveira

AGRADECIMENTOS

Agradeço ao orientador, professor e amigo Ricardo Reis, pelo exemplo de dedicação à pesquisa. Agradeço a confiança em mim depositada .

Agradeço aos companheiros TRANCA, José Güntzel, Marcus Kindel, Reginaldo Tavares, Luís Malhmann e Marcelo Johann, pelas contribuições de idéias para o sucesso deste trabalho.

Ao auxiliar Gustavo César Frederico pela implementação do CIPPATO, pela ajuda constante nas tarefas do projeto e por sua grande amizade.

A Gilberto Marchioro, pela amizade e interesse permanente em colaborar para o sucesso no desenvolvimento de ferramentas do GME.

Aos colegas Evandro Bender, Gustavo Kayser, Yung Choi, André Aita, Gilson Wirth, Luis Felipe Uebel, Ricardo Krug, Sandro Rigo, Sandro Soares pela inúmeras horas de descontração.

A todos os professores do CPGCC pelo constante interesse na realização de pesquisa e pela receptividade ao fornecimento de informações.

Aos funcionários da secretaria do Curso de Pós-Graduação em Ciência da Computação, aos funcionários do Instituto de Informática e aos funcionários do laboratório e da biblioteca, pela atenção e boa vontade com que me trataram.

Por fim, agradeço à minha família.

SUMÁRIO

CONVENÇÕES DE NOTAÇÃO	9
LISTA DE FIGURAS.....	10
LISTA DE TABELAS	12
LISTA DE ABREVIATURAS	13
RESUMO	15
ABSTRACT	17
1 INTRODUÇÃO	19
1.1 Motivação e Objetivos	19
1.2 Organização do Trabalho	20
2 SÍNTESE AUTOMÁTICA.....	22
2.1 Introdução.....	22
2.2 Projeto Físico	23
2.2.1 Planejamento Topológico.....	25
2.2.2 Geração de Módulos	26
2.2.2.1 Módulos em Lógica Regular.....	27
2.2.2.2 Módulos em Lógica Aleatória	27
2.3 Estilos de Projeto	28
2.3.1 Estilo " <i>Full-Custom</i> "	28
2.3.2 Estilo " <i>Standard Cell</i> "	29
2.3.3 Estilo " <i>Gate-Array</i> "	31

2.3.4 Estilo " <i>Cell Based</i> "	31
2.4 Particionamento/Posicionamento: uma Análise do Problema.....	31
2.4.1 O Particionamento	31
2.4.2 O Posicionamento	33
3 O SISTEMA TRANCA	34
3.1 Introdução.....	34
3.1.1 A Integração das Ferramentas TRANCA.....	34
3.2 As Características TRANCA	35
3.2.1 Estrutura de Bandas	35
3.2.2 Maleabilidade de Células e Blocos.....	36
3.2.3 Transparência de Células e Blocos.....	37
3.2.4 Gerenciamento de Trilhas	37
3.3 TRAMO	38
3.3.1 O Posicionador do TRAMO.....	39
3.3.1.1 O Particionamento.....	39
3.3.1.2 Posicionamento	40
3.4 TRAGO	41
3.4.1 Pré-Geração de Células	42
3.4.2 Particionamento	42
3.4.3 A Geração de Bandas.....	43
3.5 MARTE.....	44
3.6 MARCELA	44
3.6.1 A Adaptação Lógica	46
3.6.2 Assinalamento de Células de Base	46

3.7 O TRAMOII	47
3.7.1 Aspectos do Ambiente	48
4 ESTRATÉGIAS DE POSICIONAMENTO	50
4.1 Classificação dos Algoritmos de Posicionamento.....	50
4.2 Posicionamento por Crescimento de Aglomerados.....	50
4.2.1 Posicionamento por Geração Aleatória.....	52
4.3 Posicionamento por Particionamento	52
4.3.1 Método de Troca de Pares	53
4.3.2 Método de Migração de Grupos.....	54
4.3.3 Um Algoritmo para Posicionamento via Particionamento de Circuito em n Blocos	57
4.3.4 <i>SHARP-LOOKING</i> , Uma Heurística de Particionamento.....	60
4.4 Posicionamento por Recozimento Simulado	61
4.5 Método de Evolução Simulada	62
5 O ALGORITMO TRAPP.....	64
5.1. Introdução.....	64
5.2 Definição do Trapp	65
5.3 Conceitos Básicos para o Algoritmo.....	66
5.3.1 Particionamento	66
5.3.1.1 A Área Particionada	66
5.3.1.2 As Partições Horizontais	67
5.3.1.3 As Partições Verticais	67
5.3.2 Posicionamento.....	68
5.3.3 Função Objetivo.....	68
5.3.4 O Peso e o Envelope das Redes.....	68

5.3.4.1 O Peso Mínimo	69
5.3.4.2 O Peso Atual.....	70
5.3.4.3 O Fator de Equilíbrio.....	70
5.4 O Conceito de Equilíbrio de Conexões.....	71
5.5 A Representação Simbólica	71
5.5.1 Considerações sobre os Pinos de Interface de uma Célula.....	72
5.5.2 Considerações Sobre Redes.....	73
5.6 O Mapa de Congestionamento	76
5.7 O Grafo de Adjacência de Redes.....	77
6 O AMBIENTE TRAPP	79
6.1 Visão Geral.....	79
6.1.2 O Cippato	80
6.2 As Fases do Algoritmo	80
6.2.1 O Particionamento/Posicionamento Inicial.....	81
6.2.3 O Balanceamento de Redes	84
6.2.4 A Geração dos Caminho das Redes	87
6.3 O Mapa de Congestionamento	90
7 AVALIAÇÃO E CONCLUSÕES DO ALGORITMO TRAPP	93
7.1 O Estágio Atual do Protótipo Avaliado	93
7.2 Metodologia Utilizada para Avaliação.....	93
7.2.1 Os Circuitos Utilizados para Avaliação.....	94
7.3 Análise dos Resultados Encontrados.....	95
7.4 Implementações Futuras.....	98
7.5 Conclusões Finais.....	99

ANEXO A-1 DESCRIÇÃO SPICE DE UM CIRCUITO.....	100
ANEXO A-2 ARQUIVO DE BIBLIOTECA DE CÉLULAS.....	101
ANEXO A-3 ARQUIVO DE POSICIONAMENTO GERADO	102
BIBLIOGRAFIA	103

CONVENÇÕES DE NOTAÇÃO

Texto escrito em TIMES ROMAN

Termos Técnicos escritos em TIMES ITALIC

LISTA DE FIGURAS

Figura 2.1 - Representação dos níveis de projeto.....	22
Figura 2.2 - Representação de um chip.....	24
Figura 2.3 - Fluxo no desenvolvimento do leiaute de um chip.....	25
Figura 2.4 - Otimizações de topologia.....	26
Figura 2.5 - Geradores de Módulos.....	27
Figura 2.6 - Uma célula <i>standard cell</i> padrão.....	30
Figura 3.1 - O ambiente <i>Tentos for Windows</i>	35
Figura 3.2 - Estrutura de bandas TRANCA.....	36
Figura 3.3 - A transparência de células.....	37
Figura 3.4 - O gerenciamento de trilhas.....	38
Figura 3.5 - O particionamento do circuito no TRAMO.....	39
Figura 3.6 - Equilíbrio de regiões.....	40
Figura 3.7 - O posicionamento intrabanda.....	41
Figura 3.8 - O fluxo do TRAGO.....	42
Figura 3.9 - A geração de bandas.....	43
Figura 3.10 - O ambiente MARTE.....	44
Figura 3.11 - Uma matriz MARCELA.....	45
Figura 3.12 - Etapas do ambiente MARCELA.....	45
Figura 3.13 - Particionamento por quadratura.....	47
Figura 3.14 - O fluxo da quinta versão do módulo TRAMOII.....	49
Figura 4.1 - Estratégia de Crescimento de Aglomerados.....	51
Figura 4.2 - Algoritmo de crescimento de aglomerados.....	52
Figura 4.3 - Métodos de particionamento.....	53
Figura 4.4 - Gráfico de configurações de posicionamento.....	54
Figura 4.5 - Minimização de cruzamentos de redes.....	54
Figura 4.6 - O fluxo estabelecido pelo algoritmo de migração de grupos.....	55
Figura 4.7 - Uma análise de <i>look-ahead</i>	56
Figura 4.8 - Redes de distribuição.....	57
Figura 4.9 - Fluxo básico do algoritmo de redes de distribuição.....	58
Figura 4.10 - Movimentos baseados no envelope das redes.....	59
Figura 4.11 - O particionamento de um leiaute pela filosofia SHARP-LOOKING...	60

Figura 4.12 - O fluxo do algoritmo SHARP-LOOKING.....	61
Figura 4.13 - A idéia do algoritmo de recozimento simulado.	62
Figura 5.1 - A representação de uma banda	67
Figura 5.2 - As partições.	68
Figura 5.3 - O envelope de uma rede.	69
Figura 5.4 - Equação do peso mínimo.	69
Figura 5.5 - Os pesos de uma rede.	70
Figura 5.6 - A localização de pinos para células TRAMOII.	72
Figura 5.7 - A rede e a sua representação.	73
Figura 5.8 - O fluxo do algoritmo de geração de nodos.	75
Figura 5.9 - A geração de nodos.	76
Figura 5.10 - O mapa de congestionamento.....	77
Figura 5.11 - O grafo de adjacência.....	78
Figura 6.1 - O Ambiente TRAPP.....	79
Figura 6.2 - Estruturas da lista de células e da lista de redes	81
Figura 6.3 - O Fluxo do Particionamento/Posicionamento Inicial	82
Figura 6.4 - A Seleção de uma Rede	83
Figura 6.5 - O fluxo do balanceamento de redes	85
Figura 6.6 - O algoritmo do balanceamento de redes	85
Figura 6.7 - A minimização de uma rede	87
Figura 6.8 - Espelhamento de rede para o nodo tipo L.....	88
Figura 6.9 - Espelhamento de rede para o nodo tipo T	89
Figura 6.10 - Espelhamento de rede para os nodo tipo R.....	89
Figura 6.11 - Linhas de controle de um bloco.....	91
Figura 6.12 - A estrutura dos caminhos de uma rede	91
Figura 6.13 - O preenchimento do mapa de congestionamento.	92
Figura 7.1 - Posicionamento do circuito alu2 via TRAMO.	97
Figura 7.2 - Posicionamento do circuito alu2 via TRAPP.	98

LISTA DE TABELAS

Tabela 7.1 - A biblioteca de células utilizada.	93
Tabela 7.2 - Células implementadas por subcircuito.....	94
Tabela 7.3 - Os circuitos utilizados e suas descrições.....	95
Tabela 7.4 - Os circuitos posicionados.	95

LISTA DE ABREVIATURAS

Apontador	variável de memória que contém um endereço para outra região ou variável.
ASIC	Circuito integrado de aplicação específica (<i>Application Specific Integrated Circuit</i>).
Benchmark	conjunto de testes padronizados realizados para avaliar características de sistemas.
CAD	Projeto auxiliado pelo computador (<i>Computer Aided Design</i>).
CI	Circuito Integrado.
CPGCC	Curso de Pós-Graduação em Ciência da Computação.
DOS	Sistema Operacional em disco (<i>Disk Operator System</i>).
Ferramenta	um simples programa executável capaz de realizar uma função de projeto específica.
Framework	é uma coletânea de programas e interfaces que junto com o Sistema Operacional definem o cenário no qual as ferramentas são desenvolvidas, integradas e operadas.
MARCELA	Mar de Células.
MARTE	MAze RouTing Environment.
Metodologia	é a especificação de uma sequência de tarefas.

PAC	Projeto automatizado pelo computador (<i>Project Automated by the Computer</i>).
TRAGO	TRAnca Automatic Generator.
TRAMO	TRAnca MOdule generator.
TRANCA	TRANSPARENT Cell Approach.
TRAPP	TRANSPARENT Placement by Partitioning.
UFRGS	Universidade Federal do Rio Grande do Sul.

RESUMO

Este trabalho propõe e avalia um novo algoritmo para o posicionamento de células de circuitos que utilizam a metodologia de projeto TRANCA. O algoritmo proposto realiza o posicionamento por particionamento, em *n-blocos*, baseado no conceito de *balanceamento de redes*, realizando um pré-roteamento global. A maioria dos algoritmos de posicionamento por particionamento são baseados na heurística de Kernighan-Lin[KER 70] e Fiducia-Mattheyses[FID 82] com migração de grupos. Estes algoritmos utilizam uma função de corte mínimo para diminuir o cruzamento de redes entre as duas partições, produzindo regiões saturadas. Sendo assim, o conceito de balanceamento de redes significa a busca de um equilíbrio no comprimento das conexões para evitar a criação de regiões saturadas, diminuindo o tempo computacional e facilitando a etapa de roteamento.

Apresenta-se uma visão geral de síntese automática. Descreve-se os estilos de projeto mais utilizados, define-se e analisa-se o problema de particionamento e posicionamento de células.

As principais características da metodologia TRANCA são apresentadas. Resume-se as principais características das ferramentas de síntese TRANCA, destacando-se as etapas de particionamento e posicionamento de cada uma, visando o aproveitamento destas características positivas.

Com o propósito de fundamentar os conceitos usados para o desenvolvimento do algoritmo, apresenta-se os métodos de posicionamento mais relevantes, dando destaque àqueles baseados em particionamento. Descreve-se algumas das heurísticas existentes.

Os conceitos utilizados para o desenvolvimento do algoritmo são então descritos. O algoritmo consiste basicamente da distribuição das conexões, utilizando um **mapa de congestionamento** do circuito, o que caracteriza um pré-roteamento global. O mapa de congestionamento é montado sobre as partições geradas no circuito. Além do mapa de congestionamento, a descrição dos caminhos das redes é realizada sobre um modelo definido para controlar o cruzamento de redes. Após a definição dos conceitos, o ambiente criado para o algoritmo é apresentado.

Com o objetivo de validar os conceitos estudados e aqueles propostos, implementou-se um protótipo, chamado TRAPP(TRANSPARENT Placement by Partitioning), e um visualizador de posicionamento chamado CIPPATO.

Finalmente, alguns resultados do protótipo desenvolvido e uma avaliação sobre o comportamento deste protótipo são apresentados. Propõem-se também implementações alternativas e direções para trabalhos futuros.

PALAVRAS-CHAVES: Microeletrônica, Projeto VLSI, Ferramentas PAC.

**TITLE: "A TRAPP TOOL FOR PARTITIONING/PLACEMENT OF METHODOLOGY
TRANCA'S CELLS"**

ABSTRACT

This work proposes and evaluates a new algorithm for cells' placement, for use on TRANCA[REI 87] layouts. The algorithm proposed makes a placement by partitioning using multiple steps, based on the concept of net balancing, in order to make a global pre-routing. Most partitioning algorithms are based on the Kernighan-Lin[KER 70] and Fiducia-Mattheyses[FID 82] heuristics with migration groups. These algorithms use a min-cut heuristic to decrease the crossing nets between the two blocks, producing saturated regions. Therefore, the nets balancing concept means to search for a balance in the connections size to avoid saturated regions, decreasing a computation time and to increase the routing performance.

The global vision of automatic synthesis is shown. The main design styles are described and the placement and partitioning problems are analysed.

The main features of TRANCA methodology are shown. A summary about the TRANCA synthesis tools is presented, emphasizing the partitioning and placement step in each one. This main features are evaluated.

The basic ideas that supported the development of the algorithm are described. The algorithm provides a connection distribution, using a congestion map of the circuit that describes a global pre-routing. The congestion map is generated based on the circuit partitioning. In addition (to the congestion map), the net paths are defined to control the crossing nets. After the definition of the concepts, the environment created for the algorithm is showed.

The most important placement methods are studied and presented in order to provide a general picture of the problem. Among them, specific attention is given to those based on partitioning. Some particular heuristics are detailed.

A prototype system called TRAPP(TRANSPARENT Placement by Partitioning) was developed to evaluate this approach. It is completed by a placement viewer, CIPPATO.

Finally, some results and conclusions are presented. New implementations and directions for further works are proposed too.

KEYWORDS: Microeletronic, VLSI Design, PAC Tools

1 INTRODUÇÃO

O desenvolvimento de hardware se salientou nas últimas décadas pelo avanço da microeletrônica com a geração de circuitos VLSI, possibilitando a integração de sistemas digitais completos em uma mesma pastilha, com alguns milhões de transistores por chip. A microeletrônica assume um papel relevante na sociedade atual. Nos países desenvolvidos e nos que querem desenvolver-se, a utilização e desenvolvimento de circuitos integrados são hoje considerados estratégicos, sendo vitais para manter a competitividade de uma vasta gama de produtos industriais.

Caracterizando-se como uma área que evolui muito rapidamente, verifica-se na microeletrônica uma competitividade acirrada na definição de novos caminhos e solução. Neste ambiente de transformação constante e complexidade, a rapidez do processo de desenvolvimento de projeto de CIs é extremamente necessária para o seu sucesso tecnológico e comercial.

No momento atual em que se encontra a indústria de microeletrônica, não se pode pensar em projetos VLSI sem o uso de ferramentas para a automação do processo de concepção, considerando o custo/benefício de reduzir o ciclo de projeto. Em razão deste benefício que permite viabilizar o investimento em projetos de circuitos VLSI grandes empresas desenvolvem sistemas de CAD que automatizam importantes fases do projeto, tais como os sistemas MENTOR e CADENCE, por exemplo. O desenvolvimento de ambientes e ferramentas de CAD também apresenta-se como uma atividade importante em muitos grupos de pesquisa.

1.1 Motivação e Objetivos

A motivação deste trabalho de dissertação está relacionada ao desafio de desenvolvimento de uma ferramenta de CAD para particionamento/posicionamento de células utilizando a metodologia de projeto TRANCA. Busca-se um particionador/posicionador que seja adaptável a cada uma das abordagens TRANCA, em especial do módulo TRAMOII[REI 91], utilizando novas soluções em particionamento e posicionamento de células.

Os objetivos básicos desta dissertação podem ser descritos como:

- desenvolver um algoritmo apropriado a metodologia TRANCA, buscando flexibilidade para as futuras evoluções tecnológicas como o aumento do número de níveis de metal utilizados.

- realizar o processo de particionamento e posicionamento como etapas distintas a fim de possibilitar as mais diferentes condições de resultados, conforme a abordagem da metodologia TRANCA. Este processo deve ser suficientemente flexível para possibilitar seu aproveitamento em outras metodologias de projeto de circuitos..

- apresentar um modelo de particionamento/posicionamento caracterizado pela geração de leiautes ditos equilibrados, onde não se condiciona o conceito de corte-mínimo para as partições geradas, mas sim uma distribuição das conexões pelo circuito, facilitando a fase de roteamento.

1.2 Organização do Trabalho

Este trabalho relata o estudo e a experiência adquirida no planejamento de um posicionador baseado em particionamento e no desenvolvimento de um protótipo do sistema, a ferramenta TRAPP, para módulos TRANCA. Conta com sete capítulos que relatam as características do processo de síntese automática, a metodologia de projeto TRANCA, algoritmos de posicionamento, a descrição e o desenvolvimento do algoritmo e uma avaliação dos resultados obtidos com o protótipo.

Na introdução foi relatada a perspectiva atual para o desenvolvimento de ferramentas de CAD e os motivos que conduziram ao desenvolvimento do TRAPP. Apresenta-se no segundo capítulo, de forma resumida, o processo de síntese automática e analisa-se o problema de particionamento/posicionamento. No terceiro capítulo apresenta-se o sistema TRANCA, seus módulos e as ferramentas de CAD mais significativas para direcionar a tarefa de implementação, e discute-se suas características, vantagens e

restrições que influenciaram no desenvolvimento do TRAPP. O quarto capítulo consta de um resumo sobre posicionamento em geral, apresentando uma classificação encontrada nas principais fontes bibliográficas e procurando abordar cada classe de algoritmos. No quinto capítulo são detalhados os conceitos utilizados para o desenvolvimento do algoritmo. No capítulo seis apresenta-se o ambiente, especificando o comportamento do algoritmo e dando uma visão global do sistema. O capítulo sete é a conclusão, onde se apresenta uma avaliação de performance do algoritmo através do protótipo realizado e propõe-se futuras direções de pesquisa para o aperfeiçoamento do algoritmo.

2 SÍNTESE AUTOMÁTICA

2.1 Introdução

A complexidade envolvida no projeto de sistemas digitais requer a um grande número de informações necessárias para o seu tratamento. Como a quantidade de informações envolvidas é crescente, uma classificação de níveis de projeto é necessária. Um modelo de representação de projetos bastante utilizado é o apresentado em [GAJ 92]. Tal modelo se baseia num diagrama em forma de Y, figura 2.1, onde cada eixo está associado a um domínio de representação. Os domínios considerados são: comportamental; estrutural e físico.

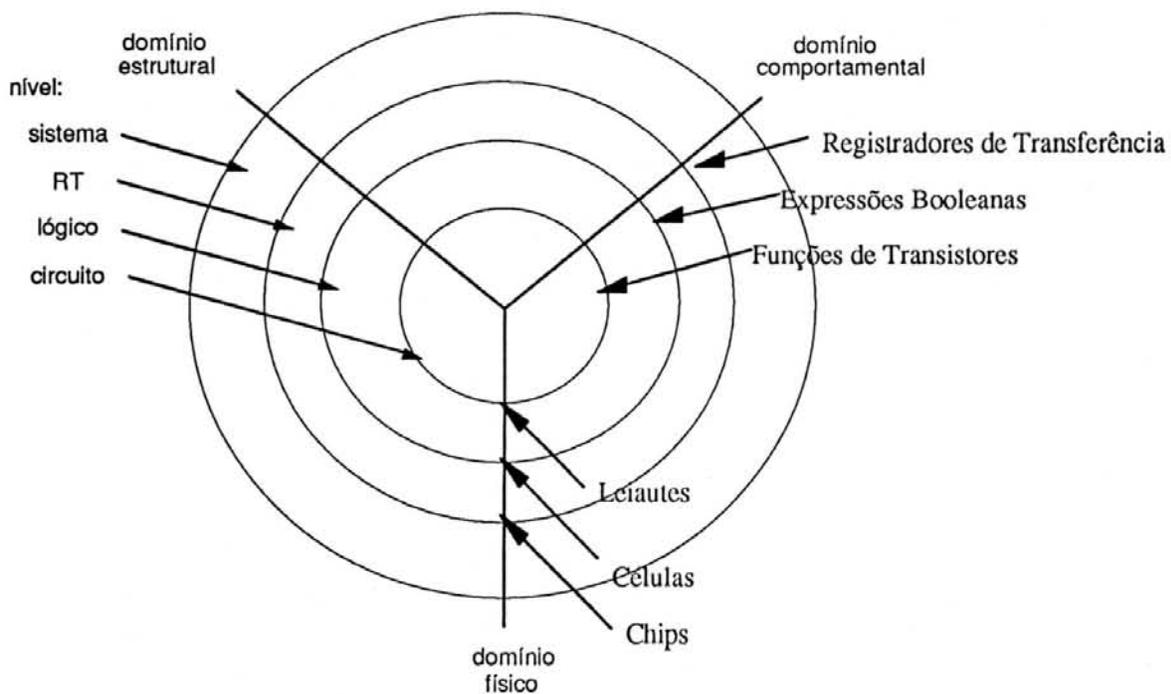


Figura 2.1 - Representação dos níveis de projeto

- Comportamental : representa o que o projeto se propõem a fazer, sem explicitar os recursos utilizados.
- Estrutural : constitui um conjunto de representações intermediárias entre o comportamento e a realização física.

- Físico: abrange descrições somente voltadas para a implementação (fabricação/prototipação) do projeto.

Usualmente, uma transformação que parte de uma descrição de comportamento e gera uma descrição estrutural ou física é denominada síntese. A automação do processo permite maior precisão e eficiência em termos de tempo para a manipulação de informações. Dentro do processo de síntese automática de leiaute de circuitos integrados, o sistema ideal de CAD é do tipo compilador de silício, ou seja, a partir de uma descrição comportamental de alto nível de abstração do circuito é possível obter-se automaticamente seu leiaute.

2.2 Projeto Físico

Neste trabalho, pelo fato de tratar de particionamento/posicionamento destaca-se o projeto físico que representa a transformação de uma descrição estrutural para a descrição a nível físico, a síntese de leiaute. Em linhas gerais podemos considerar as etapas de projeto físico como:

- Particionamento, realizar uma análise prévia sobre o circuito, dividindo-o em grupos de elementos ou células;
- Posicionamento, determinar a posição das células que compõem o leiaute, considerando fatores de otimização;
- Roteamento, o objetivo da fase de roteamento é realizar as interconexões entre as células de acordo com a lista de conexões especificada;
- Compactação, o objetivo desta etapa é compactar o leiaute em todas direções, reduzindo a área total do circuito e o tamanho das conexões;

Para uma visão geral do projeto físico de um chip pode-se estabelecer a representação de um leiaute final de um chip, figura 2.2, dividindo-o em módulos(blocos funcionais), construídos a partir de células compostas que, por sua vez, são constituídas de células primitivas ou folhas. A partir da hierarquia, o desenvolvimento de um chip pode ser dividido em uma primeira etapa relativa ao planejamento topológico dos módulos e uma

segunda etapa relativa a geração dos módulos conforme as restrições indicadas pelo planejamento topológico, figura 2.3.

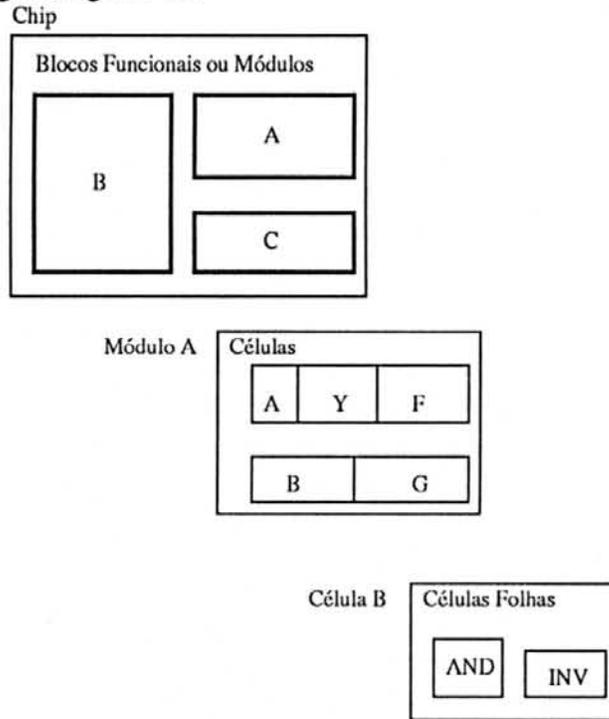


Figura 2.2 - Representação de um chip

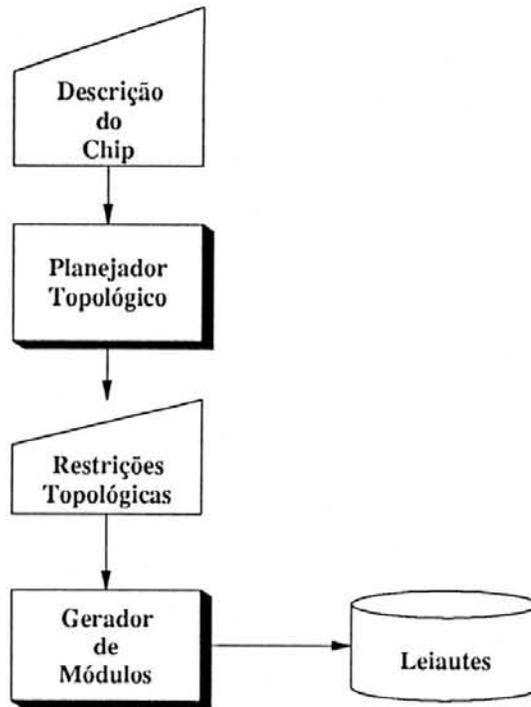


Figura 2.3 - Fluxo no desenvolvimento do leiaute de um chip

2.2.1 Planejamento Topológico

O planejamento topológico tem como objetivo determinar a planta baixa do CI visando minimizar a área em silício e a redução do comprimento médio das conexões. Segundo [REI 89], esta otimização do leiaute caracteriza-se :

- Reduzir a probabilidade de ocorrência de defeitos de materiais no espaço onde está sendo realizado o chip, em função da diminuição de área do circuito;
- Melhor desempenho elétrico;
- Aumentar o número de circuitos em um mesmo *wafer*.

O planejador topológico deve visar a otimização global em vez de otimizações locais, [REI 83], como pode ser visto na figura 2.4. A figura 2.4 mostra o bloco A da esquerda com uma área menor do que a do bloco A da direita, mas produz uma solução global indesejável. Para a obtenção de uma otimização global deve-se utilizar a característica de maleabilidade dos blocos funcionais.

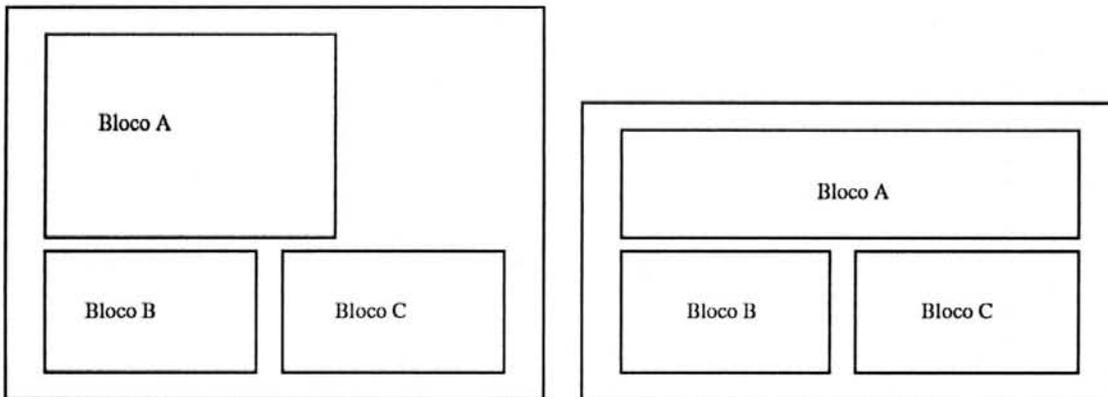


Figura 2.4 - Otimizações de topologia

2.2.2 Geração de Módulos

Geradores de módulos são ferramentas de software especializadas em gerar o leiaute, conforme uma determinada arquitetura. Estes geradores se dividem em geradores de lógica aleatória e geradores de lógica regular. Ao gerador de módulos, figura 2.5, cabe a tarefa de promover as **conexões** internas de maneira a facilitar a interconexão de blocos não adjacentes. A figura 2.5 mostra uma classificação de geradores de módulo.

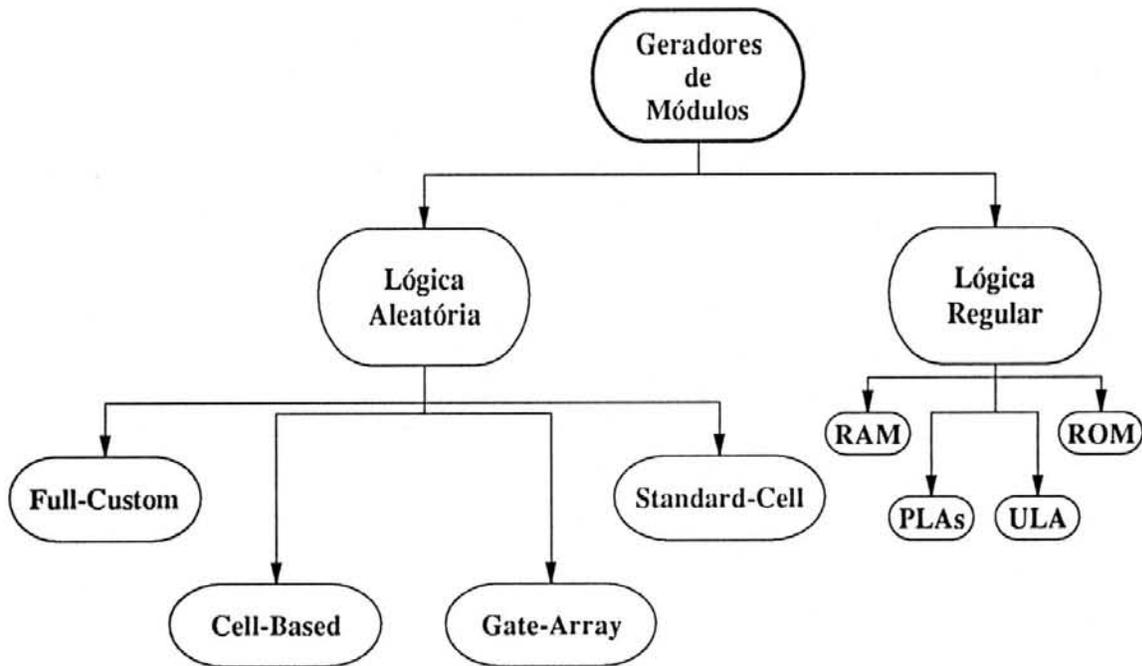


Figura 2.5 - Geradores de Módulos

2.2.2.1 Módulos em Lógica Regular

Os módulos em lógica regular são definidos por um gabarito a ser preenchido por um conjunto de células, apresentando restrições na topologia. Este gabarito pode ser visto como um leiaute de matrizes de células de tipos específicos, onde o roteamento é realizado pela justaposição das células. Exemplos de módulos em lógica regular pode são:

- Memórias(RAM e ROM);
- PLAs(Arranjos lógicos programáveis).
- ULAs(Unidades lógicas e aritméticas).

2.2.2.2 Módulos em Lógica Aleatória

Os módulos de lógica aleatória são definidos como módulos irregulares, sem um gabarito fixo para a disposição das células primitivas. Neste tipo de gerador de módulo a representação utilizada como entrada é geralmente estrutural, uma vez que o elevado grau

de liberdade para a personalização do módulo exige a descrição das interconexões das células do circuito gerado. Exemplos deste tipo de módulos são:

- Módulos projetados pela abordagem "*Standard Cell*";
- Módulos projetados pela abordagem "*Cell Based*";
- Módulos projetados pela abordagem "*Full-Custom*".

2.3 Estilos de Projeto

A adoção de um estilo a outro para o projeto de um circuito integrado é muitas vezes função da aplicação a que ela se destina. Nem sempre a implementação física de mais alto desempenho e área mais compacta será a mais vantajosa economicamente, se levarmos em conta os elevados custos de projeto relativos a esta implementação. Em vista disso, existem várias abordagens para o desenvolvimento de uma aplicação particular. Comentaremos quatro deles:

- "*full-custom*";
- "*standard cell*";
- "*gate array*";
- "*cell based*".

2.3.1 Estilo "*Full-Custom*"

O estilo "*full-custom*" adota uma abordagem em que o projetista do circuito é responsável pelo posicionamento de cada célula, transistor e interconexões do leiaute final. Neste estilo, a habilidade do projetista determina a qualidade do leiaute, onde a otimização fica sob sua inteira responsabilidade.

A automação das fases de projeto "*full-custom*" é muito complexa, devido à liberdade do projetista em gerar células complexas (variadas formas de altura e largura) sem a utilização de um modelo regular de implementação. A tarefa de posicionamento torna-se um problemática, devido às diversas combinações possíveis de posições e formas das

células, o que eleva o grau de complexidade. A utilização de ferramentas de CAD para este estilo pode se restringir ao uso de:

- Um editor de leiautes, utilizado para desenhar as células, posicionar e realizar as conexões (no GME o Edlex é a atual ferramenta);
- Verificadores de regras de projeto são indispensáveis pela liberdade deste estilo e os inevitáveis erros que o projetista possa cometer;
- Roteadores iterativos;
- Compactadores de leiaute, considerando um maior controle sobre as regras de tecnologia.

No caso da utilização de uma representação simbólica para o desenho do circuito, esta abstração com um bom expensor/compactador pode otimizar bastante o trabalho do projetista pela independência de tecnologia. Atualmente no CPGCC da UFRGS existe o editor simbólico Charrua[MAR 89].

A utilização deste terá os maiores custos de projeto. A complexidade e o tempo dispendido são muito grandes. Torna-se necessário o emprego de grandes equipes de projetistas trabalhando em tarefas de depuração e otimização do leiaute dos módulos. A abordagem "*full-custom*" é empregada geralmente em aplicações onde o desempenho do circuito é um fator crítico (por exemplo, aplicações em tempo real) ou em situações em que o volume de produção esperado compensa os investimentos em projeto de um circuito extraindo-se o máximo que o processo de fabricação pode oferecer. Essa abordagem é utilizada também em projetos onde a velocidade é um atrativo considerável de "*marketing*", como por exemplo no caso de microprocessadores.

2.3.2 Estilo "*Standard Cell*"

Na abordagem *standard cell* os geradores de módulo utilizam uma biblioteca de células para a geração do leiaute. Neste estilo a geração do leiaute é restrita ao particionamento, posicionamento e roteamento de células existentes.

O *standard cell* tradicional se caracteriza por células de altura constante, pinos de interface nas extremidades inferior e superior, figura 2.6, e os leiautes apresentam a utilização de canais de roteamento para interconexão.

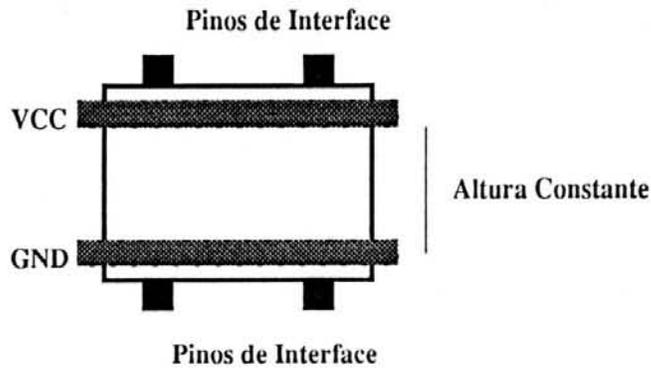


Figura 2.6 - Uma célula *standard cell* padrão

Os canais de roteamento podem ter alturas diferentes. As conexões entre células que pertencem a fileiras não adjacentes são realizadas por intermédio de espaços vazios entre as células ou células de interconexão chamadas *feedthrough*, figura 2.7. O projeto baseado neste estilo consome um tempo menor do que baseado no estilo *full-custom*, dependendo de um investimento considerável no desenvolvimento de uma biblioteca que possua vários tipos e opções (dimensionamento de transistores) de células.

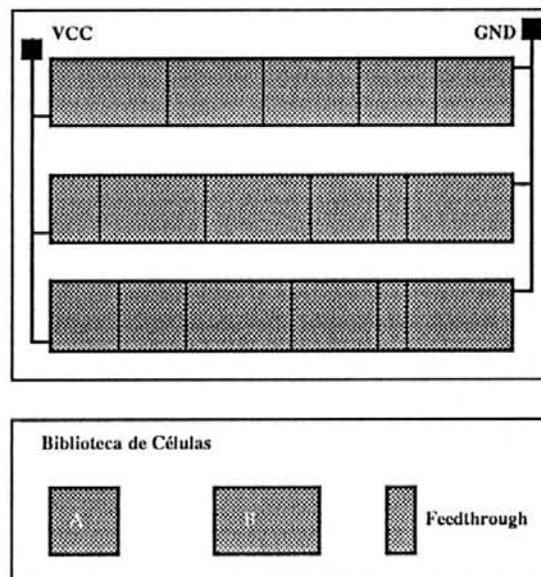


Figura 2.7 - Um leiaute *standard cell*

2.3.3 Estilo "Gate-Array"

O estilo *gate-array* se caracteriza pelo uso de uma matriz bidimensional de transistores pré-difundida e, portanto, de estrutura restrita. A personalização da matriz consiste na própria implementação do projeto. A personalização se dá pela interconexão de transistores utilizando-se uma ou mais camadas de metal, o que define o instanciamento de células sobre a matriz.

As principais vantagens deste estilo são :

- baixo custo de fabricação para pequenas quantidades;
- ferramentas de PAC com implementação simplificada.

E as principais desvantagens são:

- grande consumo de área;
- impossibilidade de ajuste elétrico.

2.3.4 Estilo "Cell Based"

O estilo "*cell-based*" é baseado em células que não possuem altura e largura padronizadas, diferenciando-se desta maneira do estilo "*standard cell*". Conforme veremos adiante, este estilo se insere a abordagem TRAMOII da metodologia TRANCA.

2.4 Particionamento/Posicionamento: uma Análise do Problema

Nesta seção resumiremos as principais características e papel do particionamento e posicionamento no processo de síntese, visando a análise posterior das limitações de cada um deles.

2.4.1 O Particionamento

Segundo [GAJ 92], particionamento é o processo de dividir um grupo de objetos em subgrupos, de maneira que uma dada função custo seja otimizada, respeitando-se restrições

de projeto. A decomposição do problema de síntese reflete-se na complexidade que envolve a geração de um circuito. Sendo assim um algoritmo de particionamento, em se tratando de leiaute, pode garantir bons resultados para as fases de posicionamento e roteamento. O particionamento pertence à classe de problemas NP-completos, sendo assim implementado via de regra através de heurísticas.

O particionamento, conforme [PRE 88], é fundamental para muitos problemas de CAD, como por exemplo:

- Encapsulamento: a lógica é particionada em blocos, considerando limites de interface e restrições de área;
- Análise de Aglomerados: em certos leiautes o particionamento é usado para organizar redes densas, como base para o módulo de posicionamento;
- Análise de Partições para Síntese de Alto Nível: a análise precisa da área de leiaute e conexões é fundamental para a síntese de alto nível;
- Simulação de Hardware e Teste: um bom particionamento minimiza o número de sinais inter-bloco que devem ser multiplexados para o simulador de hardware.

A definição de um algoritmo de particionamento de leiaute depende da análise das características da metodologia de projeto e estilo de leiaute adotado para a sua geração. A modelagem de restrições de projeto e definição da função objetivo são fundamentais para harmonizar o tratamento dos parâmetros que direcionam o algoritmo. Os principais parâmetros que caracterizam os algoritmos em geral são:

- interconexões entre partições;
- número de terminais de cada partição;
- área de cada partição;
- número de partições.

2.4.2 O Posicionamento

A etapa de posicionamento consiste em estabelecer uma posição para os objetos(células) do leiaute dentro de uma determinada área. A partir de uma etapa de particionamento se estabelece uma fronteira entre o posicionamento, geralmente, pelo posicionamento relativo realizado pelo particionamento. O posicionamento relativo considera que as células de uma partição somente podem ser reposicionadas com relação às outras células pertencentes a sua mesma partição.

Semelhante ao particionamento, o posicionamento estabelece objetivos para a formulação de um algoritmo, dos quais, dois são fundamentais a qualquer ferramenta:

- Área de leiaute e,
- Complexidade do roteamento.

A definição de outros aspectos para a realização do posicionamento é particular a implementações mais específicas. A colocação de vários objetivos sob a forma de procedimentos computacionais para uma avaliação imediata é extremamente difícil, impondo a definição de algumas funções restritas àqueles objetivos que se julgue mais significativos. Dentro destes objetivos básicos procuram-se as melhores soluções para problemas secundários, seguindo-se uma ordem de prioridades pré-estabelecidas.

3 O SISTEMA TRANCA

Este capítulo apresenta um resumo da metodologia de projeto TRANCA. Inicialmente, discute-se as principais características desta metodologia. A seguir, apresentam-se os geradores de módulos TRAMO[LUB 90], TRAGO[MOR 90a], MARCELA[GÜN 93] e TRAMOII[REI 93], que deram origem ao desenvolvimento do TRAPP, e a ferramenta de roteamento MARTE[JOH 92a][JOH 92b].

3.1 Introdução

O projeto TRANCA(TRANSPARENT-Cell Approach) [REI 87], em andamento no GME/CPGCC da UFRGS, tem como principal objetivo o desenvolvimento de ferramentas de CAD para síntese de leiaute de blocos em lógica aleatória. Em 1990, como resultado deste esforço, dois geradores de módulo TRAMO e TRAGO, foram produzidos. A implementação destes dois geradores envolveu o desenvolvimento de ferramentas de CAD como posicionador e roteador, e ferramentas de apoio como planificador, extrator etc. Após estes dois módulos, foram desenvolvidos o sistema TRAMOII e MARCELA. Este último propõe uma abordagem pré-difundida para o sistema TRANCA.

3.1.1 A Integração das Ferramentas TRANCA

Para os ambientes DOS[MOR 91], Windows[MAL 94] e UNIX[SOU 91], existem gerenciadores orientados a menus que interligam as ferramentas de CAD e de apoio. O gerenciador TENTOS é a interface para ambiente DOS e Windows, figura 3.1, e a interface gráfica para o projeto TRANCA em ambiente UNIX[SOU 91]. Ambos encapsulam a comunicação entre as diferentes ferramentas, via arquivos, para a realização de procedimentos através de opções de iteração com o usuário. A figura 3.1 apresenta a tela principal da ferramenta TENTOS *for Windows*. Para o ambiente UNIX existe no GME/CPGCC o framework SILEX[MAR 90], que integra de todas as ferramentas de microeletrônica do GME, segundo um protocolo próprio de integração de ferramentas.

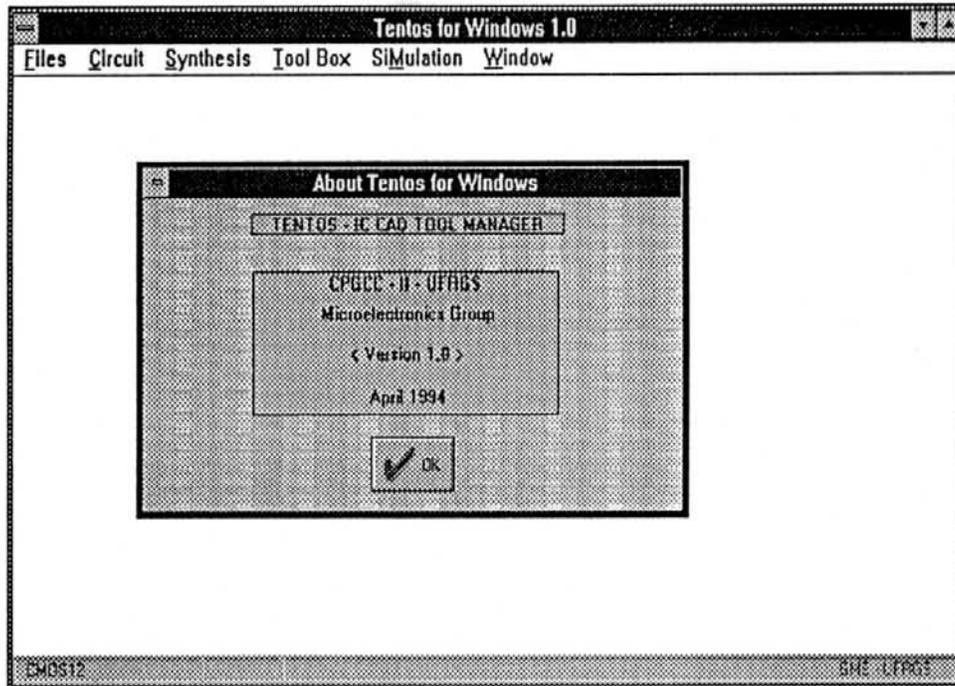


Figura 3.1 - O ambiente Tentos *for Windows*

3.2 As Características TRANCA

A metodologia de projeto TRANCA originou-se a partir de estudos estatísticos realizados sobre blocos funcionais de vários microprocessadores comerciais desenhados manualmente [REI 83]. A partir da observação destes blocos foi proposta uma metodologia de projeto que visa obter um leiaute bastante semelhante ao desenhado manualmente [REI 87]. Para tanto, esta metodologia caracteriza-se por:

- Estrutura de Bandas;
- Maleabilidade;
- Transparência de células e blocos;
- Gerenciamento de trilha;

3.2.1 Estrutura de Bandas

A estrutura de bandas TRANCA consiste em tiras de células delimitadas pelas linhas de alimentação (VCC e GND) implementadas como dois pentes com dentes intercalados,

figura 3.2. As células de uma mesma banda possuem a mesma altura e duas bandas consecutivas devem ser espelhadas no eixo horizontal para que não haja coincidência as linhas de alimentação.

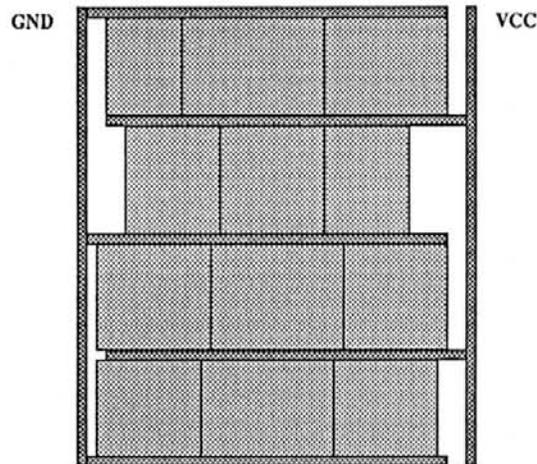


Figura 3.2 - Estrutura de bandas TRANCA

3.2.2 Maleabilidade de Células e Blocos

O conceito de maleabilidade de um bloco ou uma célula refere-se à capacidade de alterar-se a forma, via alteração topológica de seus elementos internos. A maleabilidade classifica os blocos funcionais em duros ou moles. No caso de lógica aleatória os blocos serem classificados como moles.

A maleabilidade de um bloco deve ser considerada sempre que a transparência de suas células for utilizada para realizar o roteamento. Evidentemente o particionador/posicionador deve, neste caso considerar os limites de saturação de trilhas para a disposição de células no leiaute.

Em se tratando de restrições de área a maleabilidade de uma célula é uma característica que pode facilitar circunstancialmente a etapa de posicionamento inicial.

3.2.3 Transparência de Células e Blocos

A transparência de uma célula ou bloco é definida pelo número de trilhas que pode cruzar o bloco ou célula em um determinada direção. Na metodologia **TRANCA**, a transparência horizontal é definida pelo número de trilhas de metal1 disponíveis para conexões. A transparência vertical, por sua vez, está relacionada ao número de trilhas verticais à estrutura de banda, que utilizam um o segundo nível de metal para o roteamento, **TRAMOII**[REI 93].

A figura 3.3 mostra claramente a representação destes conceitos de transparência.

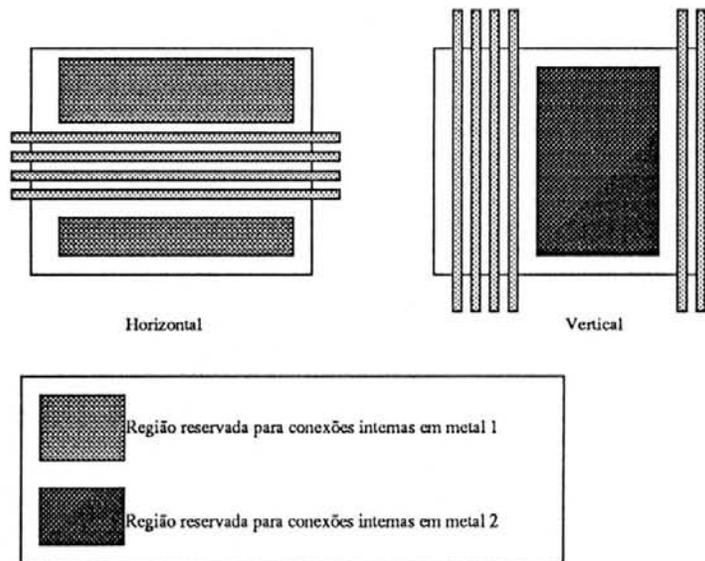


Figura 3.3 - A transparência de células

3.2.4 Gerenciamento de Trilhas

As trilhas de roteamento na metodologia **TRANCA** constituem um recurso limitado, necessitando um gerenciamento para garantir o sucesso da fase de roteamento. Na definição das prioridades para alocação de trilhas, visando a obtenção de um roteamento eficiente, é importante considerar uma estratégia de roteamento global na composição de bandas do leiaute.

A figura 3.4 mostra duas bandas em que as células são posicionadas de acordo com uma alocação de trilhas para permitir o roteamento sobre células.

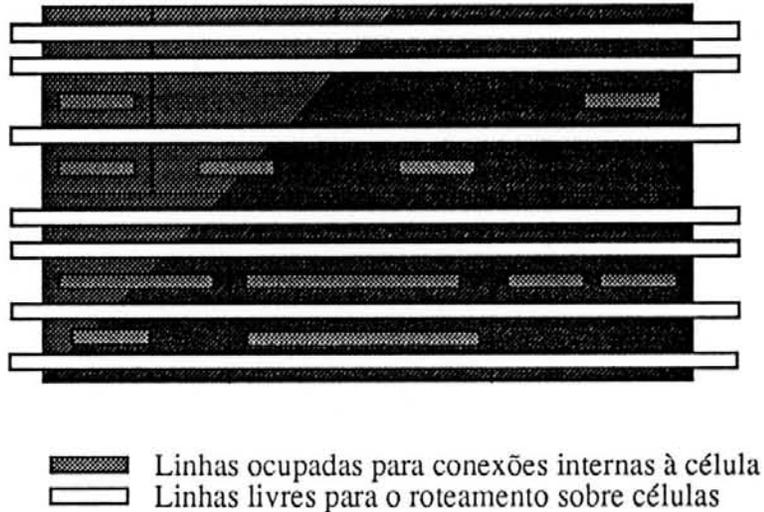


Figura 3.4 - O gerenciamento de trilhas

3.3 TRAMO

Como o primeiro gerador de módulos utilizando a metodologia TRANCA, o TRAMO[LUB 90][SOT 91] é baseado na utilização de uma biblioteca de células desenhadas segundo a metodologia descrita anteriormente. Para a composição de um leiaute este gerador de módulos divide-se em três etapas:

- Particionamento;
- Posicionamento;
- Roteamento;

Os formatos utilizados para descrição do circuito são os seguintes:

- a entrada é descrita na linguagem NILOTRANCA, derivada da linguagem de descrição de nível lógico(NILO) de [WAG 87].
- o formato de saída utilizado é uma descrição RS[TOD 86] semelhante ao formato CIF.

3.3.1 O Posicionador do TRAMO

O subsistema POTRANCA é responsável pelas fases de particionamento e posicionamento das células do leiaute. O algoritmo implementado por esta ferramenta baseia-se nas heurísticas apresentadas em [KER 70] [FID 82]. As seguintes etapas são implementadas no POTRANCA:

- O particionamento em bandas do circuito;
- O posicionamento interno a cada banda.

3.3.1.1 O Particionamento

O particionamento é realizado em duas etapas: um particionamento horizontal e outro vertical, figura 3.5. O particionamento realiza um posicionamento relativo das células, dividindo o módulo em regiões e cada região em bandas. No processo de divisão de regiões o objetivo principal é realizar um posicionamento que controle a saturação de trilhas, favorecendo o uso de células de interconexão. O particionamento das regiões em bandas procura a adjacência de redes, estimulando as conexões internas à banda.

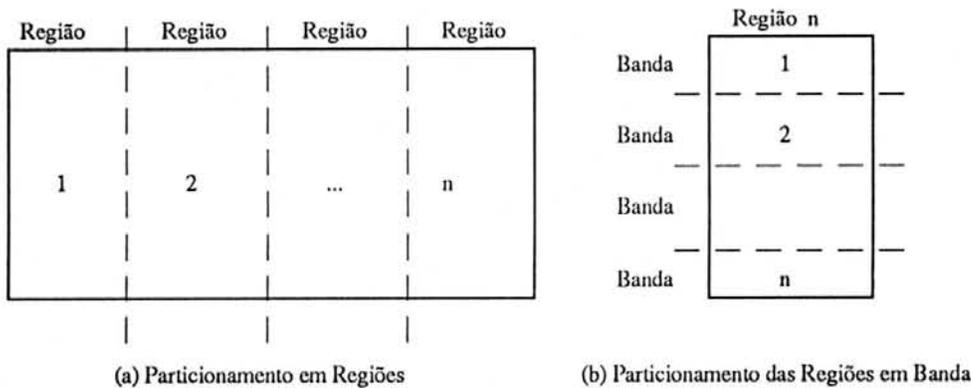


Figura 3.5 - O particionamento do circuito no TRAMO

A orientação dos sinais é realizada durante as duas etapas do particionamento. Um aspecto importante desta fase é o fator de compensação para equilibrar o espaço físico de cada banda e conseqüentemente das regiões, conforme mostra a figura 3.6.

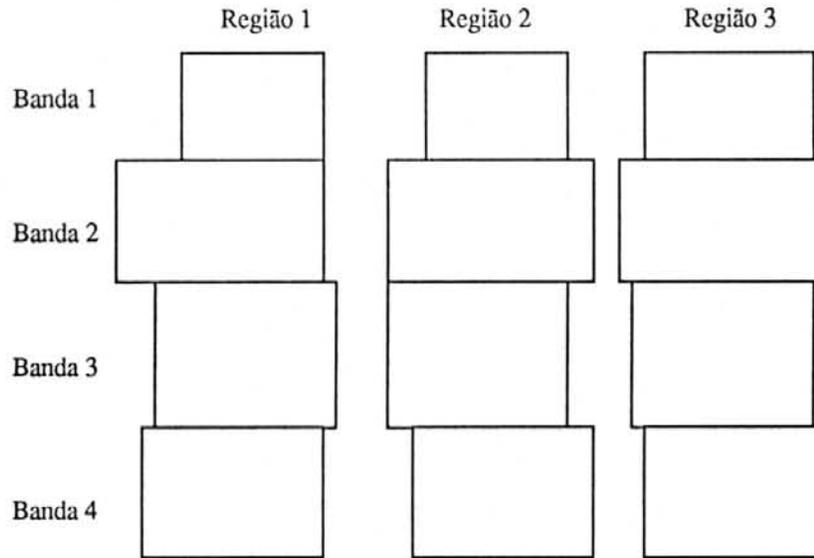


Figura 3.6 - Equilíbrio de regiões

3.3.1.2 Posicionamento

O posicionamento absoluto das células é realizado pelo posicionador intrabanda, baseado num misto de crescimento de aglomerados e corte mínimo. A figura 3.7 mostra a dinâmica utilizada para realizar este posicionamento. A partir de uma linha central estabelecida pela maior banda do circuito e o estabelecimento de linhas para condicionar a ordenação de interfaces realiza-se uma varredura do circuito estabelecendo apontadores para cada banda.

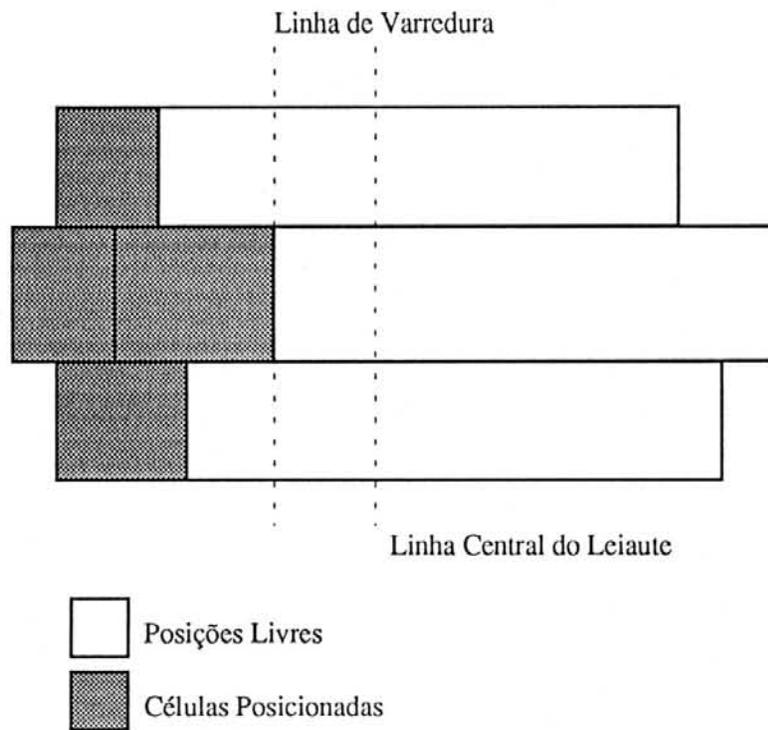


Figura 3.7 - O posicionamento intrabanda

São utilizados para a escolha de célula a ser posicionada. Os seguintes escores:

- Escore de orientação oeste/leste;
- Escore de orientação de interface livre de restrição;
- Escore de ordenação;
- Escore de vizinhança;
- Escore global.

3.4 TRAGO

O módulo TRAGO[MOR 90a] realiza a geração automática de blocos em lógica aleatória baseada na geração automática de células, segundo o estilo gate-matrix. Neste sistema um segundo nível de metal é utilizado. O fluxo do TRAGO é apresentado na figura 3.8.

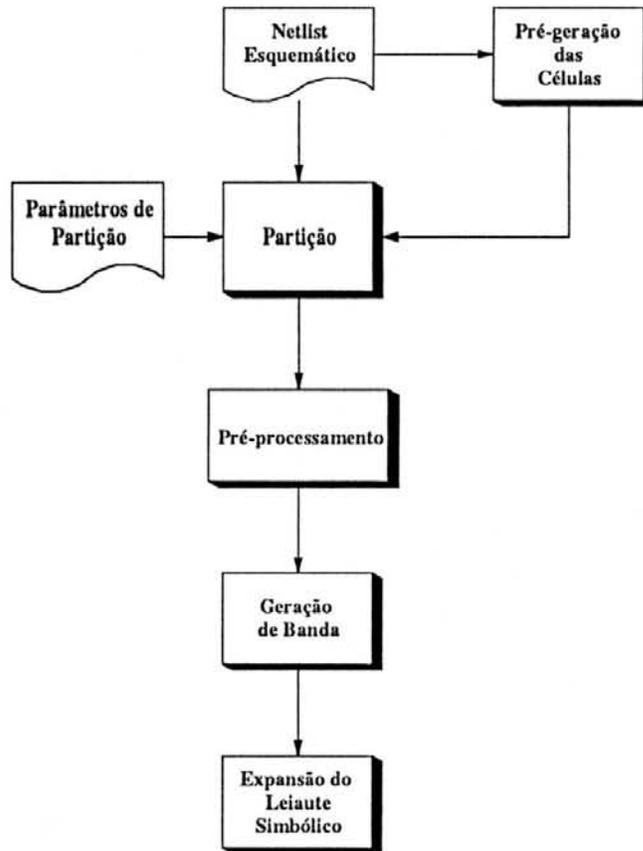


Figura 3.8 - O fluxo do TRAGO

3.4.1 Pré-Geração de Células

O processo de síntese no TRAGO é feito a partir de células básicas. As células básicas são formadas por conjuntos de transistores conectados em série e/ou paralelo entre a alimentação e a saída. Esta etapa se divide em duas fases:

- planificação da netlist de entrada do circuito;
- hierarquização e mapeamento de todas células básicas da netlist.

3.4.2 Particionamento

O módulo de particionamento é o mesmo utilizado pelo TRAMO, o subsistema POTRANCA descrito na seção 3.4.1. A diferença reside no fato que o TRAGO utiliza apenas a informação de bandas geradas, ignorando a informação de partição.

3.4.3 A Geração de Bandas

Com o objetivo de gerar cada banda do circuito, esta etapa inicia pelas bandas que apresentarem um número maior de redes compartilhadas. O posicionamento intrabanda é baseado em técnicas de crescimento por aglomerados [PRE 88], procurando favorecer as conexões por justaposição. Nesta etapa o roteamento intrabanda é realizado com um número mínimo de trilhas, utilizando o algoritmo *left-edge*. O fluxo desta etapa pode ser visto na figura 3.9.

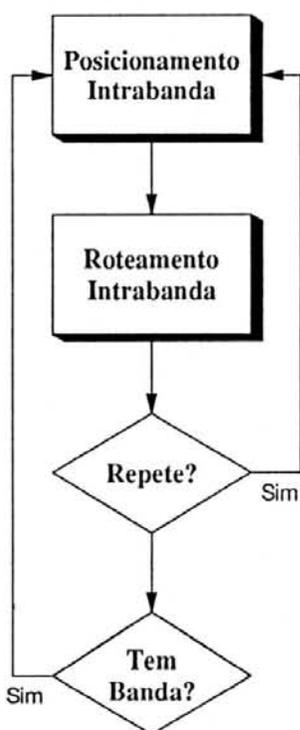


Figura 3.9 - A geração de bandas

O roteamento vertical é realizado em metal 2, excluindo a necessidade de células de interconexão.

3.5 MARTE

MARTE[JOH 92b] é uma ferramenta de roteamento do tipo *maze-router*, desenvolvida para os módulos MARCELA e TRAMOII. O algoritmo deste roteador é baseado na heurística proposta por Lee[LEE 61], que se resume basicamente ao roteamento sobre uma grade fixa e a procura do menor caminho. O ambiente e estrutura do MARTE são mostrados na figura 3.10. Otimizações deste já foram realizadas, conforme pode ser visto em [JOH 94].

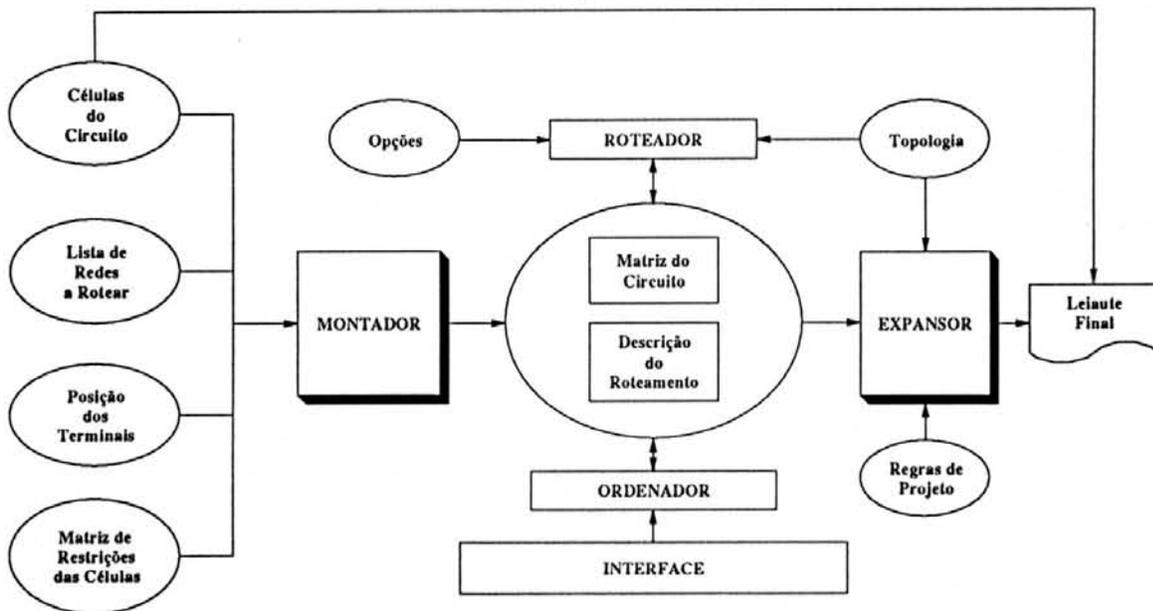


Figura 3.10 - O ambiente MARTE

3.6 MARCELA

O módulo MARCELA[GÜN 93] propõe uma abordagem pré-difundida ao estilo de leiaute TRANCA. A grande diferença do MARCELA com relação a maioria dos estilos pré-difundidos é a utilização de matrizes de células pré-difundidas ao invés de transistores. A composição de cada posição da matriz é definida como unidade básica, figura 3.11, e estas por sua vez podem implementar uma série de células de base. Uma célula de base, célula folha, implementa uma função da menor complexidade possível.

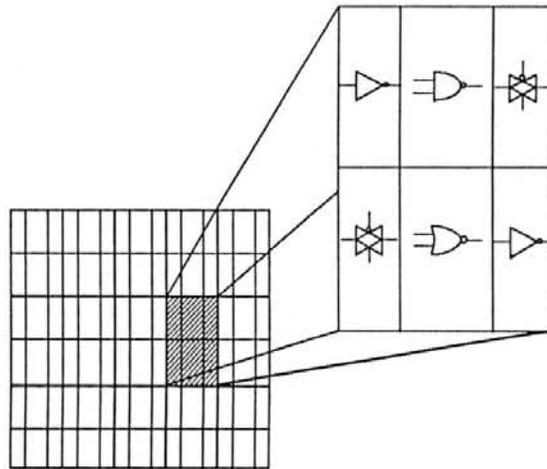


Figura 3.11 - Uma matriz MARCELA

As fases de projeto no ambiente MARCELA são mostradas na figura 3.12.

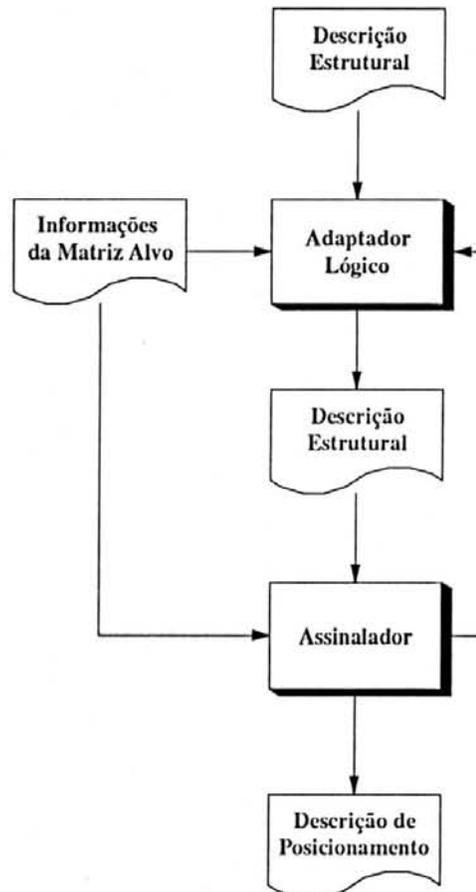


Figura 3.12 - Etapas do ambiente MARCELA

3.6.1 A Adaptação Lógica

A adaptação lógica realiza a geração do equivalente MARCELA para a matriz alvo, substituindo as funções lógicas da descrição original do circuito por aquelas em uma biblioteca. As otimizações necessárias para melhorar esta etapa são avaliadas no primeiro passo do assinalamento, o pré-assinalamento. A avaliação pode exigir a realimentação no laço.

3.6.2 Assinalamento de Células de Base

O assinalamento consiste em alocar uma região da matriz, mapeando sobre esta as primitivas existentes na descrição equivalente MARCELA. O assinalamento de células pode ser comparado ao posicionamento de células em geradores do tipo *standard cell*, considerando como unidade básica uma célula. O objetivo deste posicionamento consiste em arranjar as unidades básicas(células de mesmo tamanho) sobre a superfície com o propósito de ocupar a menor área.

O procedimento de assinalamento se divide em duas etapas:

- alocação da matriz, por comparação a um posicionamento inicial (um algoritmo construtivo);
- otimização da ocupação da matriz (em termos de posicionamento seria por meio de um algoritmo iterativo).

O posicionamento gerado no MARCELA utiliza um algoritmo de trocas baseado em técnicas de partição oriundo da heurística de Fiduccia-Matheyses[FID 82]. O particionamento é gerado por uma técnica de quadratura[BRE 77], figura 3.13, e a partir dos blocos particionados é aplicado o método de trocas para garantir a aglomeração das unidades básicas mais fortemente conectadas [HAN 76]. Este processo é realizado recursivamente enquanto os blocos gerados forem maiores que uma unidade básica.

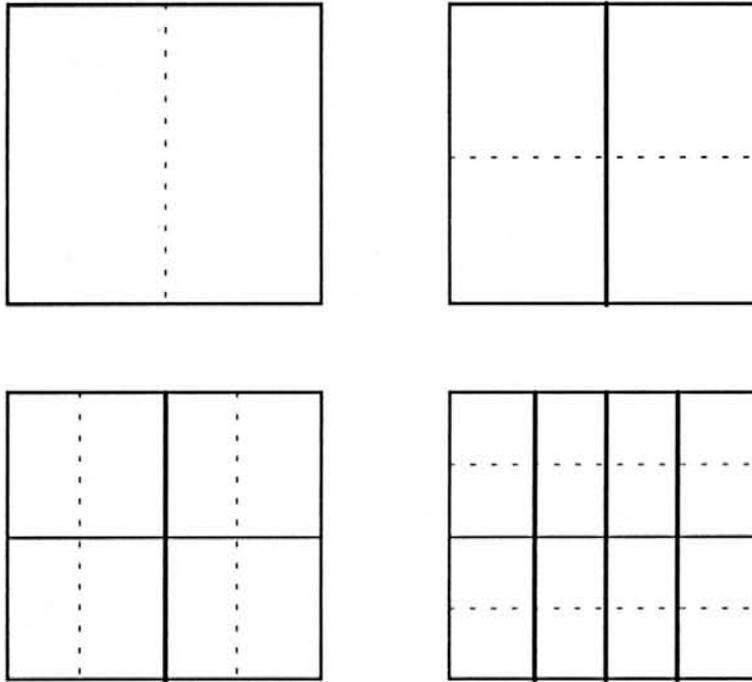


Figura 3.13 - Particionamento por quadratura.

3.7 O TRAMOII

O TRAMOII é um gerador de módulo segundo a metodologia TRANCA que utilize níveis de metal para o roteamento. O módulo TRAMOII[REI 91] surgiu pela necessidade de introduzir o uso de metal 2 no TRAMO. Este sofreu algumas modificações de maneira a incorporar características positivas do TRAGO para a otimização.

As principais características do TRAMOII são:

- Geração automática de células;
- Leiaute baseado em células transparentes;
- Roteamento simbólico sobre restrições reais.

3.7.1 Aspectos do Ambiente

Conforme proposto em [REI 93], a implementação progressiva do TRAMOII seria realizada em cinco etapas:

- primeira versão: particionamento/posicionamento e roteamento da biblioteca;
- segunda versão: geração prévia da biblioteca de células;
- terceira versão: geração do leiaute real após o particionamento;
- quarta versão: inserção eventual de canais de roteamento;
- quinta versão: uso de tabelas de relação de aspecto.

Para o desenvolvimento da ferramenta TRAPP adotamos diretamente a última versão, figura 3.14.

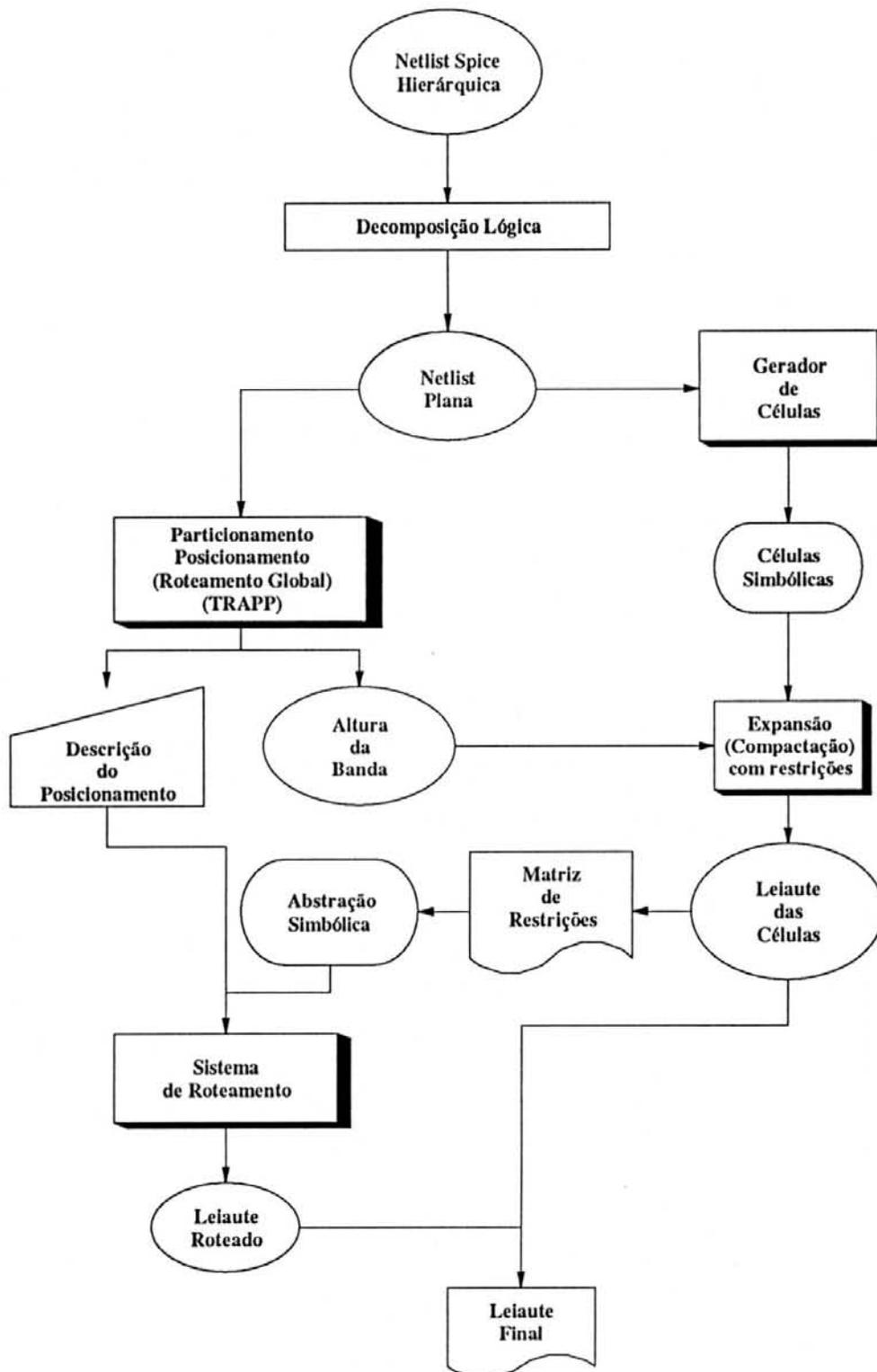


Figura 3.14 - O fluxo da quinta versão do módulo TRAMOII

4 ESTRATÉGIAS DE POSICIONAMENTO

Este capítulo discute os principais métodos utilizados no posicionamento relevantes para o desenvolvimento do sistema TRAPP. Apresenta-se uma classificação de algoritmos de posicionamento e a caracterização básica de cada método, dando destaque ao posicionamento baseado em estratégias de particionamento. A descrição de alguns algoritmos com heurísticas particulares é apresentada.

4.1 Classificação dos Algoritmos de Posicionamento

Os algoritmos de posicionamento podem ser classificados quanto à descrição de entrada, em construtivos e iterativos e, quanto ao processo de posicionamento, em :

- Algoritmos por crescimento de aglomerados;
- Algoritmos por particionamento;
- Algoritmos por processo de simulação.

Os algoritmos construtivos se caracterizam por realizarem o posicionamento inicial, ou seja, nenhum posicionamento prévio de célula do leiaute é fornecido. Os algoritmos iterativos realizam o refinamento de um posicionamento previamente realizado por algoritmos construtivos ou a partir de uma solução aleatória. Maiores detalhes sobre classificações de algoritmos de posicionamento podem ser vistos em [PRE 88] e [SHE 93].

A utilização e composição de mais de uma técnica torna difícil a classificação dos algoritmos, fazendo com que cada autor qualifique o seu algoritmo segundo critérios próprios. Nas seções seguintes apresentamos os principais algoritmos de posicionamento, descrevendo os passos gerais de cada um dentro de um escopo global de diferentes autores.

4.2 Posicionamento por Crescimento de Aglomerados

A técnica de crescimento de aglomerados é tipicamente uma abordagem ascendente de posicionamento. Esta técnica consiste basicamente de três tarefas: seleção de uma semente; seleção de um novo elemento a ser posicionado; posicionamento de cada

elemento. Na primeira etapa seleciona-se um ou mais elementos como semente, formando aglomerados iniciais. No passo seguinte, escolhe-se um ou mais elementos candidatos ao posicionamento. A última etapa encarrega-se de posicionar aquela célula escolhida no passo anterior que possua o maior peso. O processo é repetido até todos elementos terem sido posicionados.

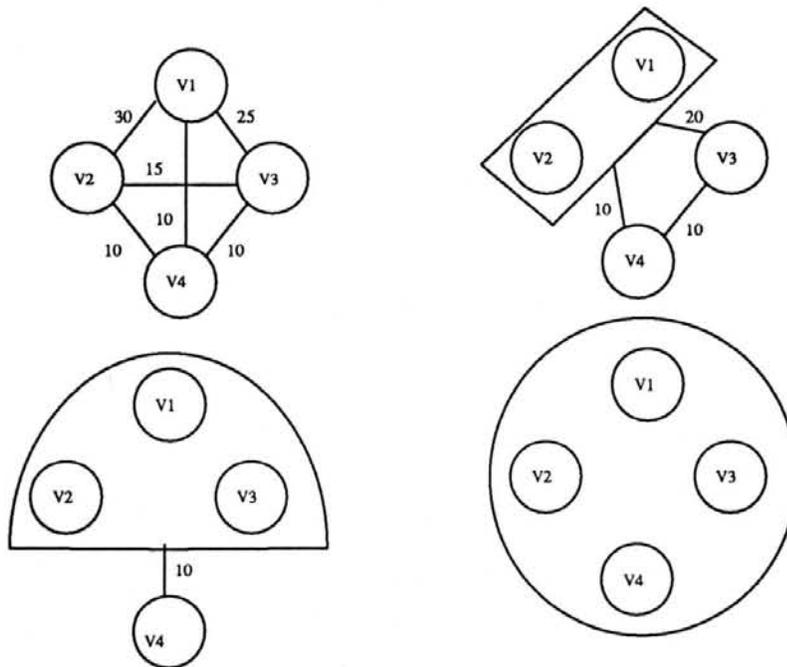


Figura 4.1 - Estratégia de Crescimento de Aglomerados.

Na figura 4.1 mostra-se passo a passo e através da modelagem de grafos, os procedimentos de seleção e aglomeração de nodos, segundo o algoritmo descrito em pseudo-código na figura 4.2.

O posicionamento por crescimento de aglomerados é um método construtivo que pode ser utilizado para o posicionamento inicial de algoritmos iterativos. Os resultados do posicionamento deste método são insatisfatórios por não analisarem dados suficientes, tais como os escores para a seleção da próxima célula, por exemplo, calculados sem considerar seleções futuras.

Para cada vértice i
Inicialize vértice i para cada elemento a aglomerar
Enquanto existe vértice para aglomerar
 Procure a aresta i, j com o maior peso
 Junte V_i, V_j em um novo nodo
 Remova todas arestas envolvendo V_i e V_j
 Adicione uma nova aresta entre V_{ij} para cada nodo
 V_k adjacente
 Recalcule o peso das arestas geradas
 Fim do Laço
Retorne o aglomerado

Figura 4.2 - Algoritmo de crescimento de aglomerados.

4.2.1 Posicionamento por Geração Aleatória

Uma variação ou degeneração do crescimento de aglomerados é a geração aleatória do posicionamento. Este método é bastante simples e frequentemente utilizado para gerar o posicionamento inicial. O fluxo deste algoritmo é basicamente um simples laço de repetição, onde cada elemento não posicionado é selecionado aleatoriamente e associado à primeira posição livre. O procedimento é repetido até todos elementos estarem posicionados.

4.3 Posicionamento por Particionamento

Algoritmos baseados em particionamento se caracterizam por dividir o circuito em dois ou mais blocos, sendo esta divisão realizada por diversas estratégias. Exemplos destas estratégias são: divisão por biseção, quadratura ou fatiamento, figura 4.3. Para qualificar o particionamento estabelece-se uma função objetivo que permite determinar o controle da qualidade do posicionamento. A função objetivo é definida a cada implementação, mas geralmente com o objetivo de minimizar o número de redes que cruzam as linhas divisórias (linhas de corte) entre os blocos gerados pelo método de particionamento.

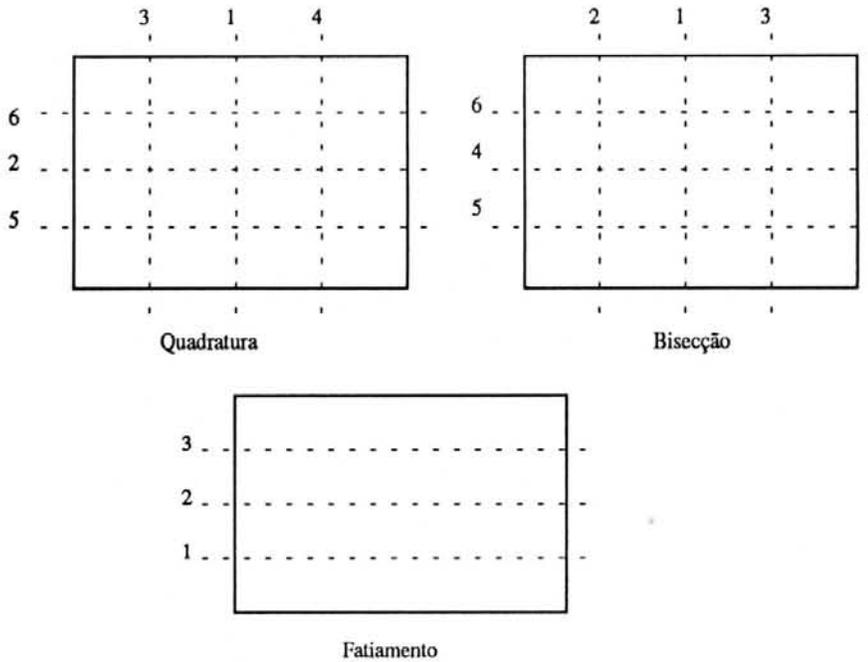


Figura 4.3 - Métodos de particionamento.

A utilização de técnicas de particionamento para o posicionamento deve-se principalmente à característica descendente do particionamento, que considera desde o início do processo o circuito como um todo. Os resultados dos algoritmos que utilizam particionamento são melhores do que aqueles obtidos com métodos ascendentes, mas o tempo computacional requerido aumenta substancialmente conforme a definição da função objetivo.

4.3.1 Método de Troca de Pares

O método de troca de pares é a mais simples utilização dos algoritmos baseados em posicionamento por particionamento. Este método se baseia na troca de posição de dois elementos(o par) desde que haja um ganho de custo para a função objetivo. O procedimento se repete enquanto houver ganho para a função objetivo. O maior problema deste algoritmo se dá devido aos mínimos locais, ou seja, o fato de não permitir movimentos com custo deficitário pode evitar que configurações posteriores que representariam uma solução final mais otimizada sejam atingidas. A figura 4.4 mostra graficamente a solução gerada por este tipo de algoritmo através de uma curva representando as soluções, onde a posição de parada não representa a melhor seleção.

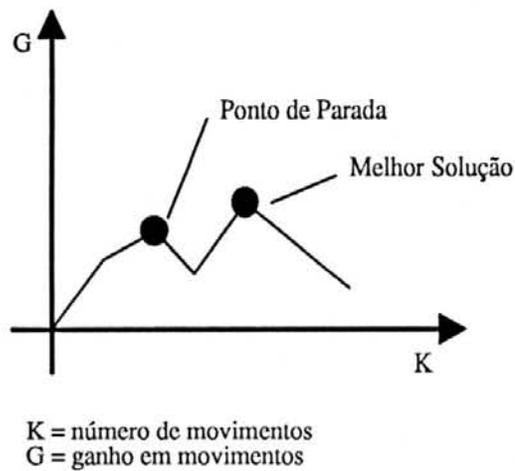


Figura 4.4 - Gráfico de configurações de posicionamento.

4.3.2 Método de Migração de Grupos

Este algoritmo também é conhecido como algoritmo de Kernighan e Lin [KER 70] em sua implementação original. A partir da geração de dois blocos balanceados pelo número de elementos em cada um, a função objetivo a ser realizada é a de corte mínimo estabelecida por uma linha de separação entre as partições. O corte mínimo corresponde ao menor número de redes cruzando tal linha. Para sua obtenção, o algoritmo promove movimentos entre as duas partições minimizando, a cada passo, o cruzamentos de redes entre partições, figura 4.5.

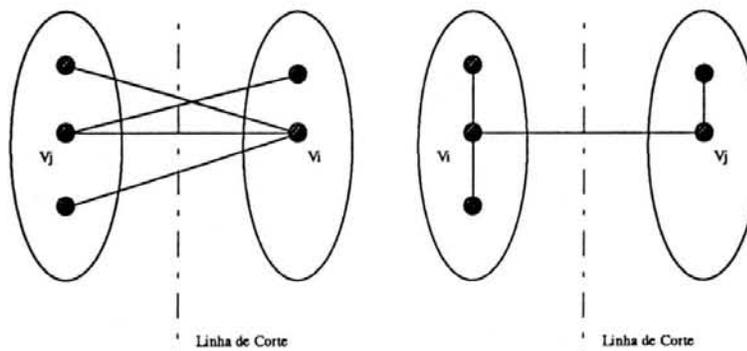


Figura 4.5 - Minimização de cruzamentos de redes.

O fluxo deste algoritmo é apresentado na figura 4.6. As primeiras implementações deste método apresentavam restrições quanto às redes compartilhadas por mais de dois

elementos. A cada iteração do algoritmo o custo computacional era elevado. Em versões mais recentes, se o número médio de conexões por nodo é relativamente pequeno, o tempo computacional é proporcional a $n \log(n)$, mas se a densidade aumenta o custo cresce proporcionalmente a n^2 .

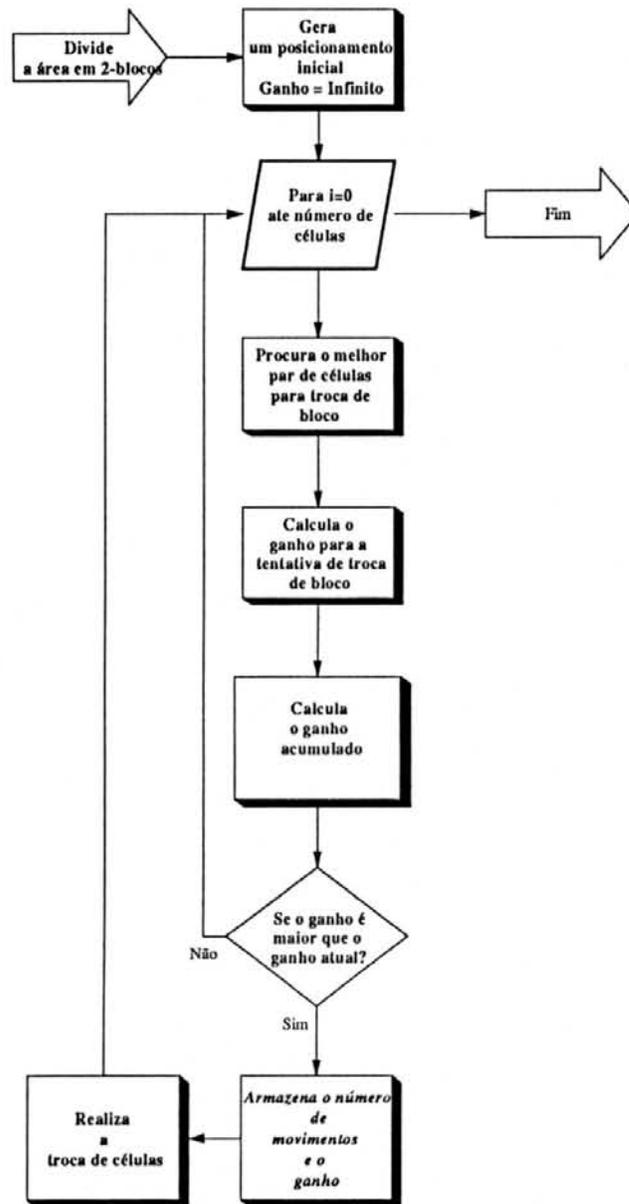


Figura 4.6 - O fluxo estabelecido pelo algoritmo de migração de grupos.

Kernighan e Schweikert propuseram em [SCH 72] uma nova estratégia para o cálculo do ganho em redes compartilhadas por mais de um elemento.

A versão gerada por Fiduccia-Mattheyses[FID 82] tem como objetivo maior a linearização do processo. Este algoritmo propõe as seguintes melhorias sobre a versão de Kernighan-Lin[KER 70]:

- definição de estruturas de dados mais otimizadas, possibilitando o tratamento correto nos seguintes casos:
 - Pesquisas desnecessárias para a realização de trocas de células;
 - Cálculos desnecessários do ganho com trocas;
 - Atualizações desnecessárias dos valores de ganho nas trocas.
- balanceamento de partições baseado no tamanho dos blocos, ao invés do número de células por bloco.

Em uma versão produzida por Krisnamurthy[KRI 84] a principal característica introduzida no algoritmo foi o uso de *look-ahead*. Por exemplo, a figura 4.7 mostra que movendo v1 ou v2 o congestionamento da linha de corte será o mesmo. Entretanto, ao mover-se v2 habilita-se v5 para posterior movimento, diminuindo o número de linhas cruzando a linha de corte estabelecida.

A partir da versão de Fiduccia-Mattheyses surgiram inúmeros algoritmos, adicionando novas heurísticas e adaptando para n blocos com vistas à otimização do método de particionamento. Na seção abaixo destacamos alguns novos algoritmos oriundos deste.

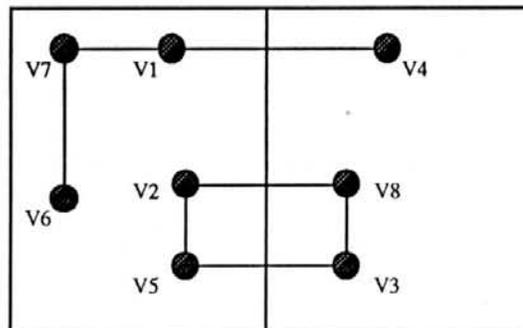


Figura 4.7 - Uma análise de *look-ahead*.

4.3.3 Um Algoritmo para Posicionamento via Particionamento de Circuito em n Blocos

Este algoritmo trata-se de uma extensão do algoritmo linear de Fiduccia-Mattheyses adaptado para n blocos. O modelo proposto por [STR 91] é baseado na definição de um modelo próprio para as redes e de uma função eficiente de custo de rede dos n blocos gerados. O número de blocos a serem gerados, n , é no máximo igual ao número de células a posicionar. Os blocos gerados podem ter tamanhos e formatos diferentes.

A modelagem utilizada para formular a representação das redes é fundamental para o desempenho da função de custo associada. Neste algoritmo propõe-se a definição de uma rede de distribuição para representar cada rede do circuito. Uma rede de distribuição consiste de um conjunto $\{ N_{h,1} \dots N_{h,i} \dots N_{h,n} \}$ onde $N_{h,i}$ é o número de células conectadas à rede h na sub-área i . Um exemplo pode ser visto na figura 4.8.

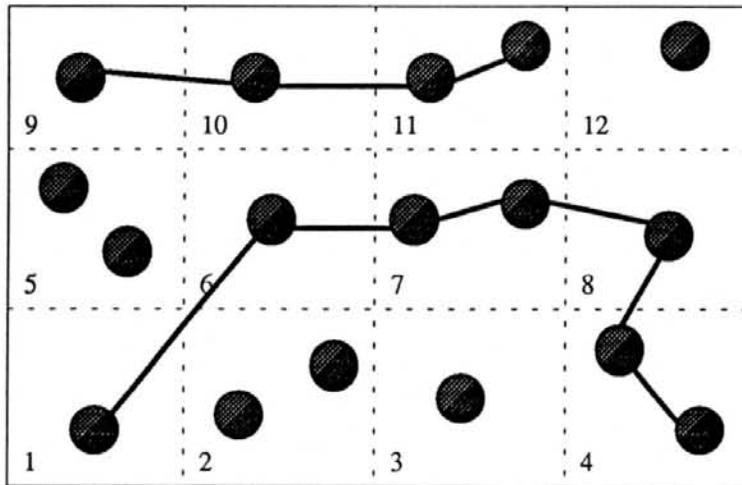


Figura 4.8 - Redes de distribuição.

No exemplo da figura 4.8 as duas redes de distribuição podem ser representadas como:

- (a) - $\{ 1,0,0,2,0,1,2,1,0,0,0,0 \}$;
- (b) - $\{ 0,0,0,0,0,0,0,0,1,1,2,0 \}$;

A função de custo de rede $F(N_h)$ associa um custo com a rede de distribuição N_h . A função de custo calcula o peso de uma rede, utilizando o tamanho do envelope das sub-áreas que compõem a rede conforme figura 4.8. O fluxo básico do algoritmo pode ser visto na figura 4.9.

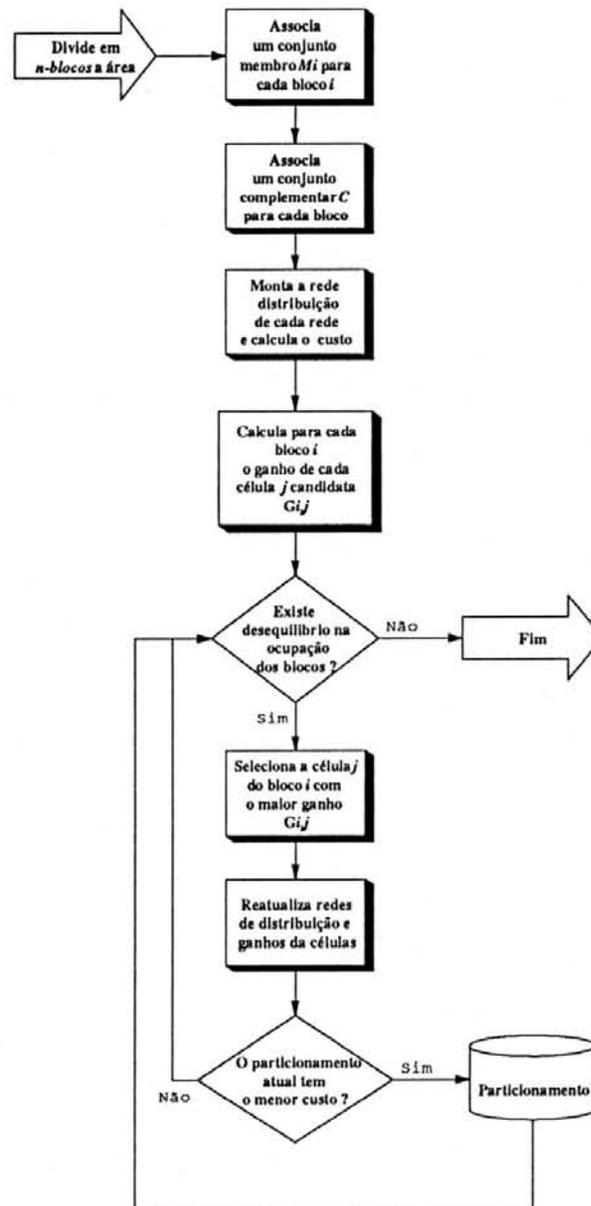


Figura 4.9 - Fluxo básico do algoritmo de redes de distribuição.

O primeira etapa do algoritmo divide a área em n sub-áreas. A cada sub-área associa-se um conjunto de células e um outro conjunto complementar para a realização de movimentos. Como o tamanho de cada partição pode ser diferente armazena-se o tamanho das partições. A última tarefa desta etapa é a montagem das redes de distribuição e o cálculo do custo de cada uma delas.

A partir do particionamento inicial estabelece-se uma função de qualidade segundo:

- A soma dos custos de todas as redes;
- O equilíbrio entre o tamanho das partições e o tamanho ocupado pelas células.

Na fase iterativa o algoritmo seleciona os elementos para realizar os movimentos, segundo os escores calculados na primeira etapa. Nos movimentos considera-se somente uma célula por vez. O ponto de finalização do método é quando não existe mais células para realizar movimentos sobre partições que obtenham ganho segundo os requisitos estabelecidos pela função de qualidade do particionamento. A figura 4.10 mostra o movimento de células de uma rede para melhorar o posicionamento com o uso do conceito de envelope.

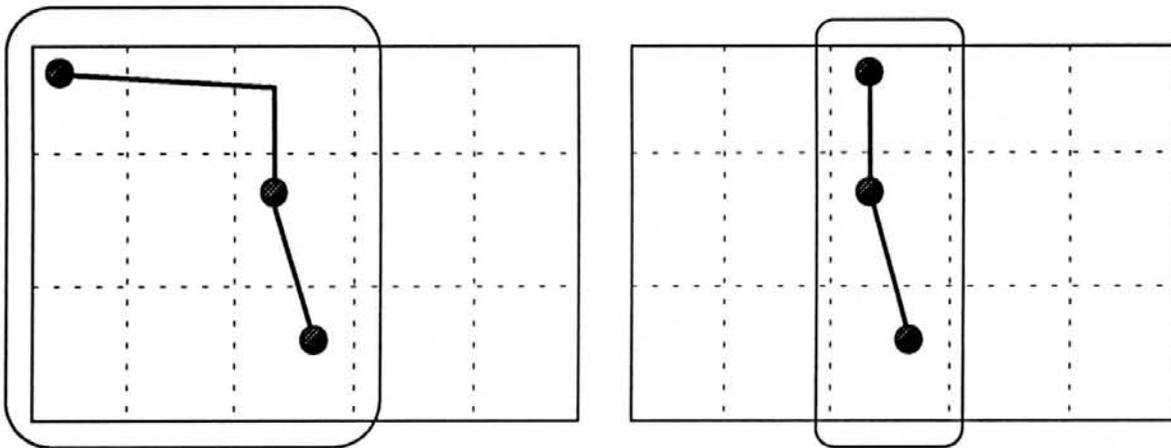


Figura 4.10 - Movimentos baseados no envelope das redes.

4.3.4 SHARP-LOOKING, Uma Heurística de Particionamento

Um nova técnica para posicionamento baseado em particionamento foi desenvolvida por [BAP 91]. SHARP é uma heurística baseada no particionamento do leiaute em nove regiões, figura 4.11a. No leiaute particionado representa-se as partições pelo conjunto $S = \{ S_1, \dots, S_9 \}$ e as fronteiras das partições figura 4.11b, pelo conjunto $C = \{ C_1, \dots, C_9 \}$.

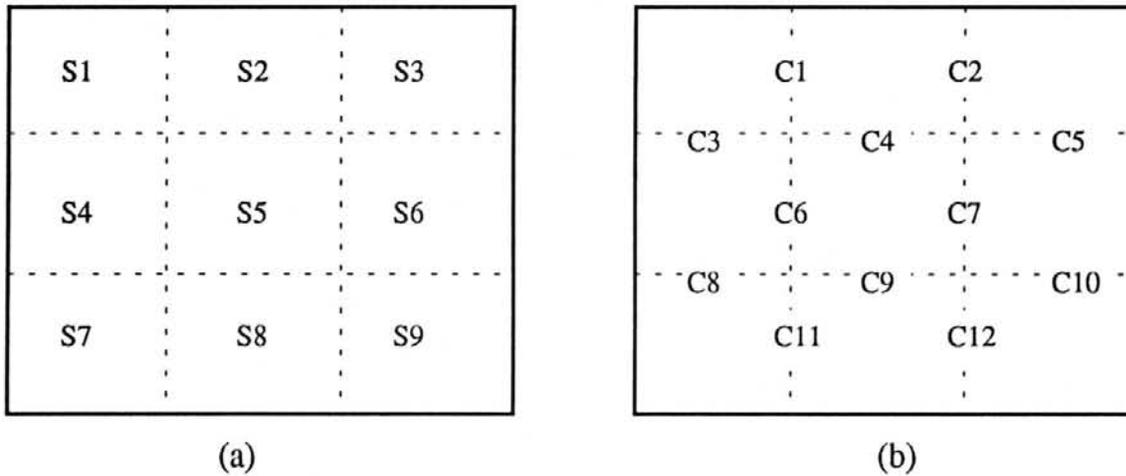


Figura 4.11 - O particionamento de um leiaute pela filosofia SHARP-LOOKING.

A filosofia SHARP se baseia no uso intensivo da representação das redes por árvores de Steiner e num controle de congestionamento pelo conjunto de fronteiras das partições. O algoritmo realiza o posicionamento das células nas partições, selecionando a melhor distribuição da rede. O fluxo seguido pelo algoritmo pode ser visto na figura 4.12.

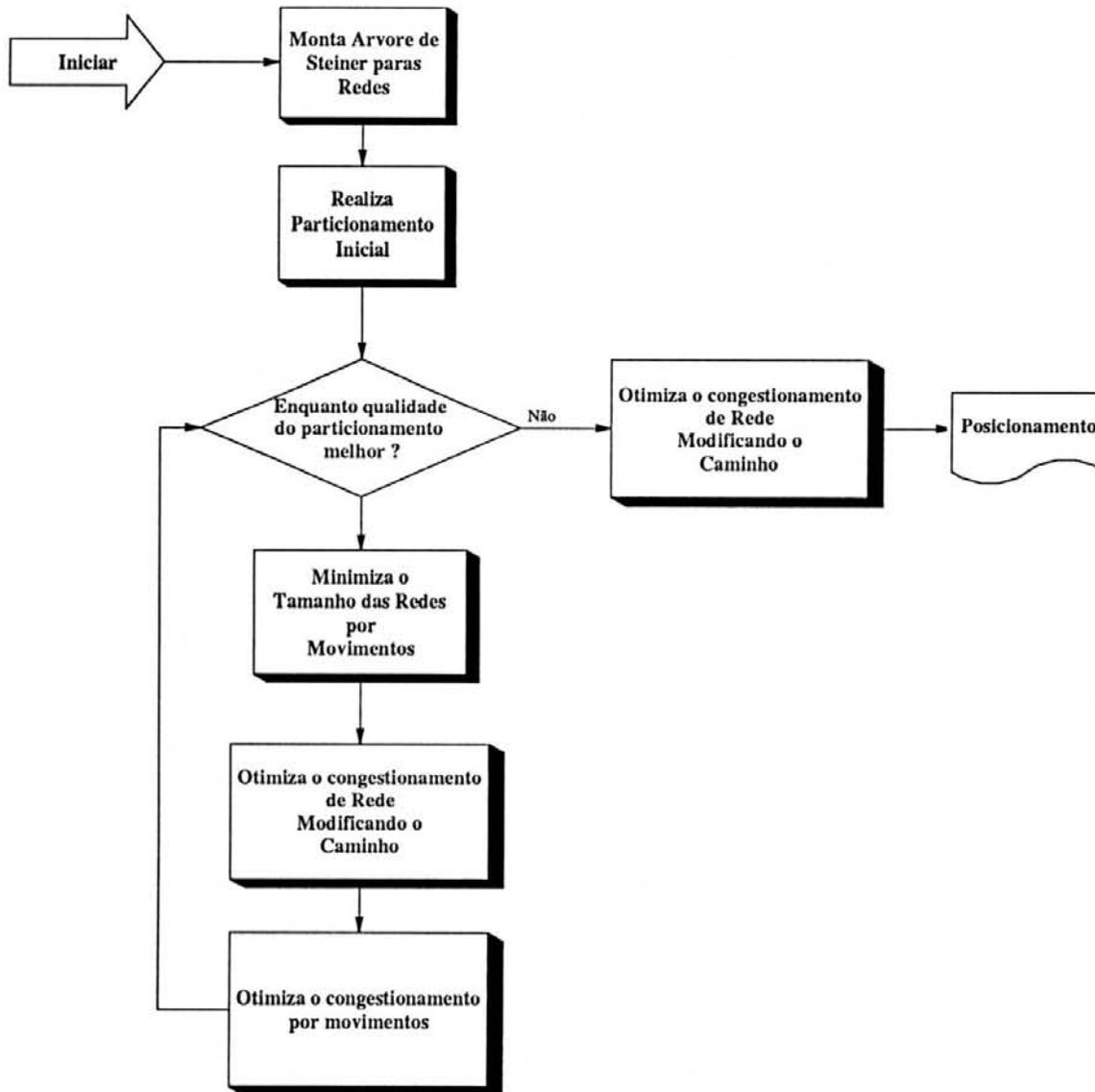


Figura 4.12 - O fluxo do algoritmo SHARP-LOOKING.

4.4 Posicionamento por Recozimento Simulado

O recozimento simulado é um método de grande poder para resolver problemas de otimização. O recozimento simulado é baseado em mecanismos estatísticos e motivado por uma analogia com comportamento físico dos materiais na presença de calor. Os métodos

baseados em recozimento simulado começam com um posicionamento inicial gerado aleatoriamente. O sucesso desta metodologia consiste na habilidade de realizar movimentos para superar os mínimos locais e atingir o ótimo global. Na figura 4.13 mostra-se o algoritmo básico deste método.

```

Obtenha uma solução S
Obtenha uma temperatura inicial  $T > 0$ 
Enquanto não congelado ( $T > 0$ ) faça o seguinte
  Realize o seguinte laço L vezes
    Escolha uma  $S'$  de S
     $\Delta = \text{custo}(S') - \text{custo}(S)$ 
    Se  $\Delta \leq 0$ 
       $S = S'$ 
    Se  $\Delta > 0$ 
       $S = S'$  com probabilidade  $e^{-\Delta/T}$ 
   $T = rT$ 
Retorne S.

```

Figura 4.13 - A idéia do algoritmo de recozimento simulado.

A cada posicionamento gerado calcula-se um escore para comparar ao escore do posicionamento anterior. Se o escore for menor a nova configuração é aceita, senão o posicionamento pode ser aceito com uma probabilidade definida por uma função peculiar a cada algoritmo. No algoritmo descrito na figura 4.13 o escore é função da diferença de custos (DELTA), da temperatura T e de um número aleatório r.

Em geral o método de recozimento simulado produz resultados satisfatórios, mas com um alto custo computacional.

4.5 Método de Evolução Simulada

O método de evolução simulada ou algoritmo genético é o oposto do recozimento simulado. Os algoritmos de evolução simulada se baseiam em um processo de realização de

um novo posicionamento a partir da combinação de dois possíveis posicionamentos para o mesmo leiaute.

Como exemplo deste método o algoritmo genético GASP[SHA 90] se baseia na analogia do processo de evolução biológica. Operadores genéticos são desenvolvidos para combinar posicionamentos existentes, reproduzindo novos posicionamentos.

O algoritmo começa com um conjunto inicial de configurações(posicionamentos) aleatórios, baseados na mesma população(células). A partir disto, o algoritmo realiza as iterações(geração) para a realização de novos posicionamentos. Para cada indivíduo da população(célula) mede-se o seu *fitness* (semelhante a uma estimativa do tamanho das redes compartilhadas). A cada iteração são selecionados os indivíduos com o melhor *fitness* de dois posicionamentos para comporem o novo posicionamento através da aplicação dos operadores genéticos. Os operadores genéticos são : *crossover*, a combinação de posicionamentos; *mutation*, semelhante à troca de pares e o *inversion*, significando um espelhamento de células.

A regra básica é gerar novos posicionamentos a partir de posicionamentos de melhor qualidade, excluindo do processo de geração os posicionamentos de baixa qualidade.

5 O ALGORITMO TRAPP

O desenvolvimento da abordagem TRAMOII[REI 93], apresentada na seção 3.8, originou a necessidade de uma ferramenta de posicionamento automático para juntamente com o sistema de roteamento MARTE[JOH 92a], permitir a realização de um módulo de geração de leiautes totalmente automático e flexível.

A fim de prover subsídios para o desenvolvimento deste algoritmo apresentou-se nos capítulos anteriores 2, 3 e 4 as bases para o planejamento do algoritmo, aperfeiçoando algumas idéias e inserindo otimizações para o seu enriquecimento. Conforme visto no capítulo 2, o problema do particionamento/posicionamento envolve um grande número de parâmetros a serem definidos e controlados para a execução do algoritmo, gerando dificuldades para se desenvolver uma implementação que atenda todas restrições. Como o atendimento de todos objetivos ao mesmo tempo torna-se inviável, a definição de prioridades é fundamental para ordenar o processo. O algoritmo do **TRAPP** foi desenvolvido segundo as características TRANCA, a partir de prioridades definidas por uma função objetivo.

5.1. Introdução

O posicionamento de células é uma fase decisiva para o sucesso da geração de um leiaute, sendo responsável diretamente pelo desempenho do circuito. Em outras palavras, um bom posicionamento produz um tamanho menor nas interconexões e uma menor área, obviamente se aliado a uma boa estratégia de roteamento. Sendo assim, conforme visto no capítulo 4, muitos projetistas de leiaute têm se dedicado ao desenvolvimento de algoritmos para a geração de posicionamentos mais otimizados, destacando-se os algoritmos baseados na utilização de particionamento oriundos do algoritmo de Kernighan-Lin[KER 70].

Analisando-se as técnicas de posicionamento existentes, brevemente apresentadas no capítulo 4, e as características TRANCA, capítulo 3, constata-se que as técnicas de particionamento, utilizando migração de grupos é fortemente indicado. Com o objetivo de aprimorar-se, alia-se um conceito de equilíbrio de conexões, conceito este desenvolvido pelo

autor, e uma análise de aglomerados que pode ser produzido para o posicionamento inicial já justifica o desenvolvimento de um sistema de posicionamento baseado em migração de grupos.

O primeiro objetivo no desenvolvimento do sistema TRAPP é o de completar a automação do módulo de geração TRAMOII. Posteriormente este sistema será utilizado em outras abordagens TRANCA com necessidades diferentes. Em função disso o algoritmo realiza um posicionamento completo, gerando um posicionamento inicial e iterativo. O algoritmo é do tipo construtivo. A idéia do posicionamento não baseado em geração aleatória, tem o objetivo de produzir uma ferramenta que exija um menor tempo computacional, se possível linear, visando o tratamento de circuitos grandes.

A utilização da técnica de particionamento é normalmente baseada na divisão em duas partições, cuja função objetivo é o corte mínimo. O conceito de corte mínimo estabelece o menor número de conexões finais entre as duas partições, onde a análise local à linha de corte é suficiente. A utilização do conceito de equilíbrio de conexões, relevante em se tratando de particionamento em *n-blocos*, exige uma análise global de todas regiões do circuito.

No que segue, são apresentados os conceitos envolvidos no desenvolvimento do algoritmo e a descrição de toda a modelagem necessária para sua implementação .

5.2 Definição do Trapp

O sistema TRAPP(TRANSPARENT Placement by Partitioning) é um posicionador baseado em uma técnica de partição em *n-blocos* oriundo da classe de migração de grupos e utilizando o conceito de "*full over-the-cell routing*" da metodologia TRANCA.

O Trapp possui as seguintes características:

- Representação Simbólica;
- O formato de entrada é um netlist spice;
- Descrição do posicionamento das células e dos caminhos das redes;

- A qualidade do particionamento é qualificada pela utilização de um mapa de congestionamento;
- As bandas podem ter alturas diferentes.

5.3 Conceitos Básicos para o Algoritmo

5.3.1 Particionamento

O particionamento dentro do contexto do algoritmo Trapp considera o processo de dividir inicialmente a área de posicionamento em fatias verticais, inserindo linhas para o mapeamento das redes. Após este processo, a divisão em bandas do circuito é efetuada conforme o posicionamento das células, sendo então criadas partições com duas linhas para mapeamento das conexões.

5.3.1.1 A Área Particionada

A área a ser particionada é definida por valores de relação de aspecto fornecidos pelo usuário e por informações sobre as dimensões simbólicas e o número de trilhas de roteamento das células que vão compor um determinado leiaute.

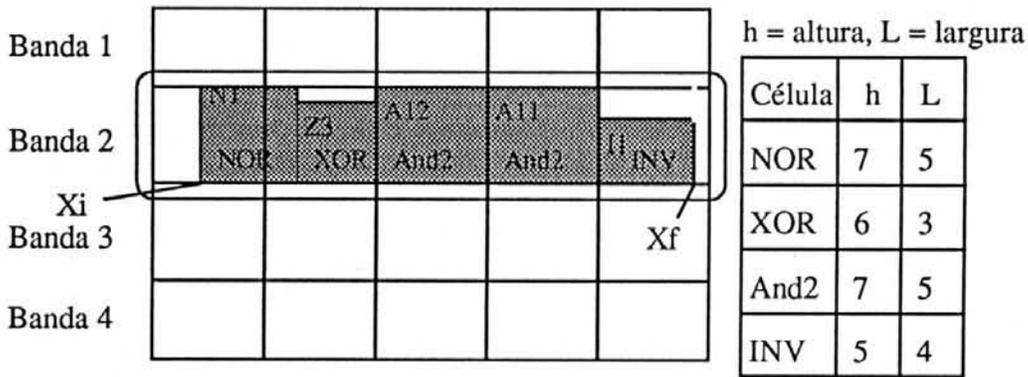
A partir da definição da área, o formato do circuito é definido pela relação de aspecto desejada. As dimensões da área a ser particionada são definidas pelo somatório da área de todas as células do circuito mais um percentual de ajuste para compensar os fatores de forma das bandas geradas.

Os valores de relação de aspecto fornecidos devem considerar a importância da capacidade de maleabilidade das dimensões geradas, ou seja, certos formatos podem produzir posicionamentos que não permitem garantir o roteamento. Considerações sobre maleabilidade são apresentadas posteriormente.

5.3.1.2 As Partições Horizontais

As partições horizontais, bandas, podem ter alturas diferentes. A altura de uma banda é definida pela altura da célula de maior altura na banda. A altura das células também pode ser utilizada como um fator de decisão no posicionamento.

Para cada banda horizontal considera-se a altura, suas células pertencentes e seu fator de forma, figura 5.1, para o controle do posicionador.



Banda 2 { Fator_de_Forma (Xi,Xf), h = 7, Células = (N1, Z3, A12, A11, I1) };

Figura 5.1 - A representação de uma banda

5.3.1.3 As Partições Verticais

A largura das partições verticais é definida pela menor largura de célula pertencente ao circuito. As partições verticais possuem todas a mesma largura, ocasionando a ocupação de mais de uma partição por células maiores. A divisão pela célula de menor largura possibilita a inserção de um maior número de linhas de controle para gerenciar a ocupação de regiões no leiaute.

Para fins de entendimento, a figura 5.2 apresenta a definição de bloco como o menor subconjunto formado pelo particionamento vertical e horizontal.

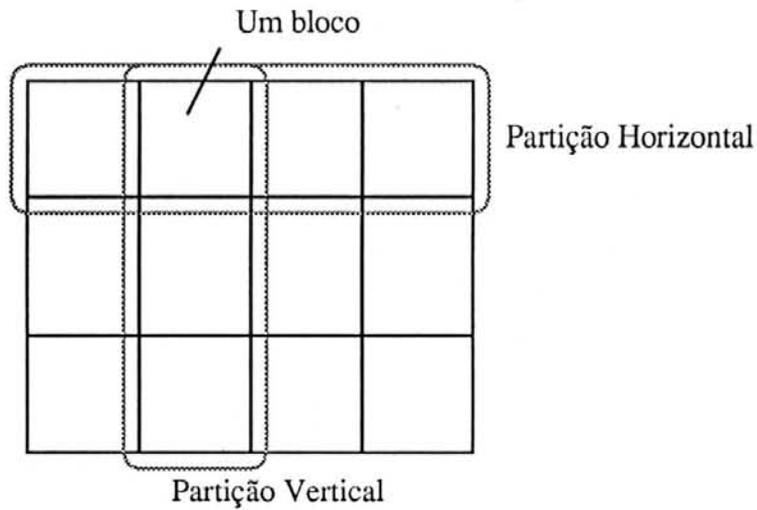


Figura 5.2 - As partições.

5.3.2 Posicionamento

O posicionamento é realizado em duas etapas no algoritmo que se propõe. Em uma primeira etapa o posicionamento inicial das células é realizado em conjunto com o particionamento do circuito em bandas. A partir deste posicionamento inicial, existe a parte iterativa do algoritmo que realiza otimizações.

5.3.3 Função Objetivo

A função objetivo do algoritmo Trapp é definida como uma função que deve ser otimizada nas fases de particionamento e posicionamento para a busca do equilíbrio do peso das redes. Além do fator de equilíbrio utilizado nestas fases, o uso do mapa de congestionamento, se habilitado, coordena as ações do particionamento inicial.

5.3.4 O Peso e o Envelope das Redes

O peso das redes é representado por três tipos de pesos atribuídos:

- O peso mínimo;
- O peso atual;
- O fator de equilíbrio.

O envelope das redes é necessariamente a área que cada rede ocupa dentro do circuito, sendo este conceito utilizado por facilitar a manipulação de dados para o controle de cada rede de um circuito. A figura 5.3 destaca o conceito de envelope.

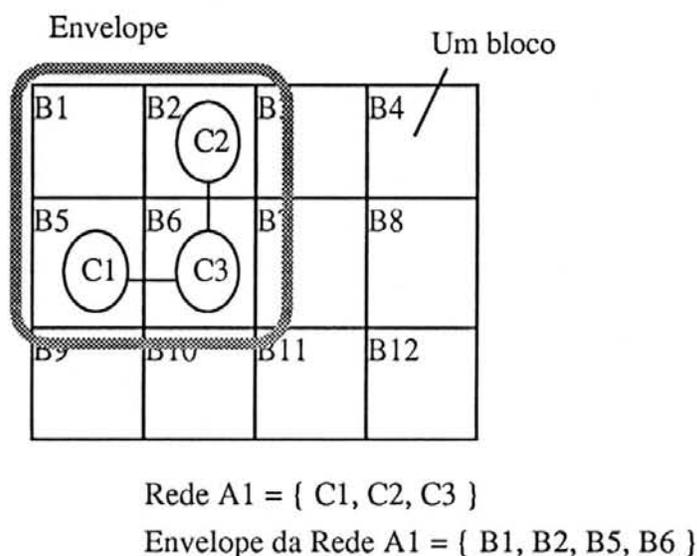


Figura 5.3 - O envelope de uma rede.

5.3.4.1 O Peso Mínimo

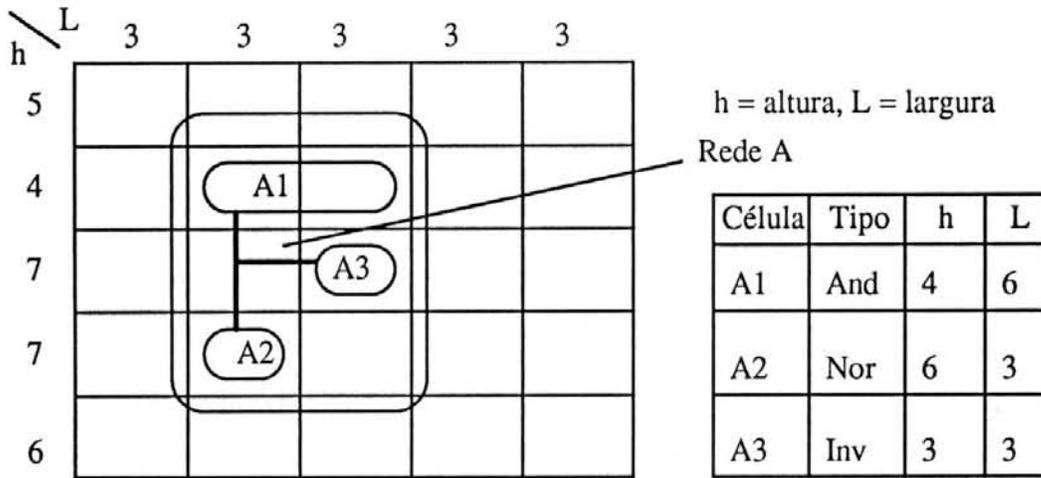
O peso mínimo, equação na figura 5.4, é determinado pelo somatório das larguras das células da rede mais a maior altura de célula pertencente a rede, o que corresponde a um envelope mínimo. A figura 5.5 apresenta um exemplo de peso mínimo.

$$pm(N_j) = \sum_{i=0}^n Lc_{ij} + \text{Maior}(H_i, \dots, H_n)$$

Lc_{ij} = largura da célula i da rede j ;

H_i = altura da célula i da rede.

Figura 5.4 - Equação do peso mínimo.



Peso Mínimo = 18;

Peso Atual = 24;

Fator de Equilíbrio = 6.

Figura 5.5 - Os pesos de uma rede.

5.3.4.2 O Peso Atual

O peso atual é determinado pelo semiperímetro (a metade do envelope) que uma rede ocupa no circuito. Neste caso a utilização do envelope pode causar empates no peso de redes que efetivamente não ocupem a mesma área, conforme o número de células da rede. Porém, o fator de equilíbrio compensa esta distorção de atribuição de pesos. A figura 5.5 apresenta um exemplo de peso atual de uma rede.

5.3.4.3 O Fator de Equilíbrio

O fator de equilíbrio é definido como a diferença entre o peso atual e o peso mínimo, figura 5.5. O fator de equilíbrio é o peso que efetivamente decide em primeira instância a escolha de posicionamento de uma rede, outros fatores serão apresentados mais adiante. Juntamente com o mapa de congestionamento, o fator de equilíbrio pode caracterizar a situação de posicionamento de uma rede em um leiaute.

5.4 O Conceito de Equilíbrio de Conexões

A literatura apresenta, em geral, o conceito de particionamento intimamente ligado ao conceito de minimização de linhas de corte. Esta definição geralmente demanda um maior tempo computacional e produz leiautes com regiões saturadas no circuito, dificultando ou até impossibilitando a fase de roteamento.

Na definição do algoritmo do Trapp partiu-se do objetivo de geração de um posicionamento que possibilite uma fase de roteamento otimizada e realizável, considerando-se que a utilização de uma função objetivo, seção 5.3.3, gere leiautes com o menor tamanho possível para cada rede, ocupando de maneira mais uniforme o espaço para roteamento.

Para o algoritmo Trapp os leiautes são considerados equilibrados após a fase de posicionamento, quando os fatores de equilíbrio de redes adjacentes não apresentem distorções muito significativas, beneficiando alguma rede com o prejuízo de outra.

A diferença de não necessariamente procurar-se o corte mínimo não exclui a possibilidade de alcançar o menor comprimento de redes do circuito. As redes podem ter o fator de equilíbrio igual ao seu peso mínimo, situação ideal para uma rede. Por exemplo, em situações de conexões de justaposição, desde que no somatório global do comprimento das redes do circuito seja o menor com esta configuração e a condição de congestionamento se apresente viável.

5.5 A Representação Simbólica

A tarefa de posicionamento deste algoritmo não impõe a utilização de todas as informações sobre uma célula, sendo suficiente o uso da altura, largura das células, e se houver, a transparência para realização dos procedimentos envolvidos. Com o objetivo de ter-se independência de tecnologia, a utilização de uma representação simbólica facilita este propósito e ao mesmo tempo cria um método para o encapsulamento do **TRAPP** em outras ferramentas de síntese, deixando a tarefa de descrição de células a um gerador de células folha que seja adaptável a diferentes tecnologias e metodologias.

Dentro da concepção do fluxo necessário para a geração de leiaute no TRAMOII, o particionamento/posicionamento conta com o uso de uma tabela com diferentes alturas de uma mesma célula, ficando a critério do posicionamento a escolha da mais apropriada e cabendo ao gerador de células folhas a geração conforme os parâmetros utilizados no particionador/posicionador. O particionamento/posicionamento utiliza como unidade trilhas de roteamento, considerando o mapeamento da área do circuito como uma grande matriz.

5.5.1 Considerações sobre os Pinos de Interface de uma Célula

A descrição de uma célula utilizada para o posicionamento não considera a informação exata sobre pinos de interface. No caso do gerador de módulos TRAMOII a representação simbólica de uma célula considera que os pinos não estão necessariamente nas bordas, em trilhas fixas, mas podem estar em posições internas, figura 5.6. A escolha do posicionamento dos pinos fica transparente ao posicionador. Para metodologias sem transparência, a localização exata dos pinos não modificaria o resultado final pela utilização do conceito de envelope.

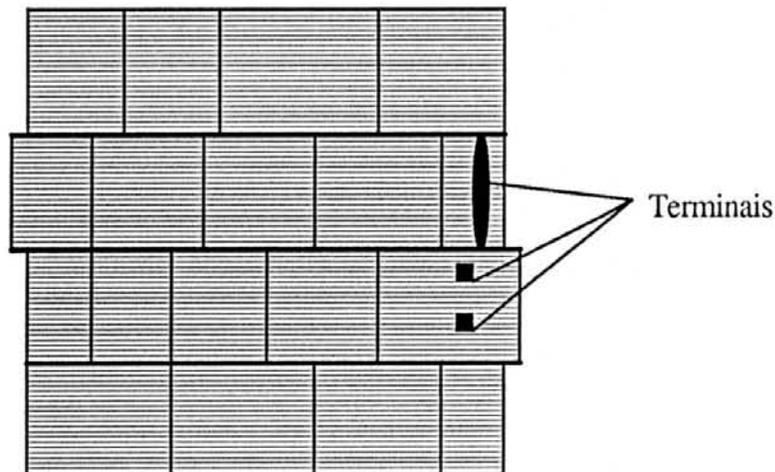


Figura 5.6 - A localização de pinos para células TRAMOII.

5.5.2 Considerações Sobre Redes

A definição de rede neste algoritmo envolve todas as células que possuam um nodo de ligação em comum. A figura 5.7 exemplifica este conceito. Esta representação de rede é utilizada devido à necessidade de um gerenciamento das conexões para a fase de roteamento, bem como devido à necessidade de um controle de alocação de espaço para as células em uma área de leiaute.

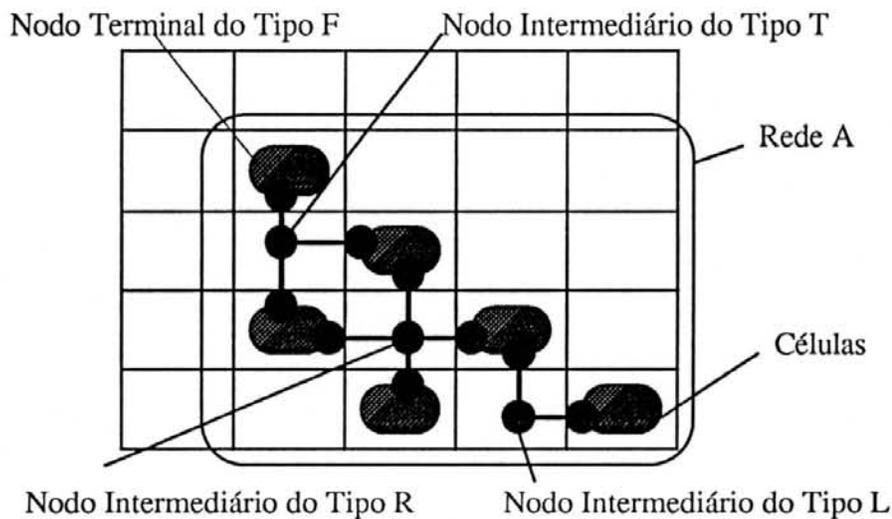


Figura 5.7 - A rede e a sua representação.

O controle das ligações de cada rede impõe a necessidade de uma estrutura mais adequada à descrição de um possível caminho ocupado pelas redes. Para tanto, desenvolveu-se, a partir dos princípios de roteamento global a geração de pontos de controle, chamados de nodos, a fim de auxiliar no gerenciamento de saturação de trilhas ocupadas.

Estes nodos são gerados a partir de um algoritmo, desenvolvido como parte de nosso trabalho, baseado na utilização de árvores de Steiner. Uma classificação quanto à ocorrência e à direção do caminho que o nodo simboliza é utilizada.

A classificação quanto à ocorrência, figura 5.7, é a seguinte:

- Terminal, atributo que indica a existência de uma célula junto ao nodo;

- Intermediário, atributo que indica que o nodo não está sobre uma célula;

Em relação à direção do caminho da ligação, figura 5.7, a classificação é a seguinte:

- Tipo F, conexão final ou inicial, apenas um caminho;
- Tipo L, dois caminhos possíveis;
- Tipo T, nodo representando três possíveis caminhos;
- Tipo R, nodo indicando quatro possíveis caminhos.

O algoritmo para a geração de nodos, figura 5.8, primeiramente ordena as células pela menor posição nas coordenadas X e Y, respectivamente, e a partir desta ordem, começa a geração dos nodos. A figura 5.9 exemplifica esta geração. Neste algoritmo é importante salientar que o objetivo é a busca do menor caminho, mas não é necessariamente o encontrado. O passo de geração de nodos intermediários é fundamental para este propósito.

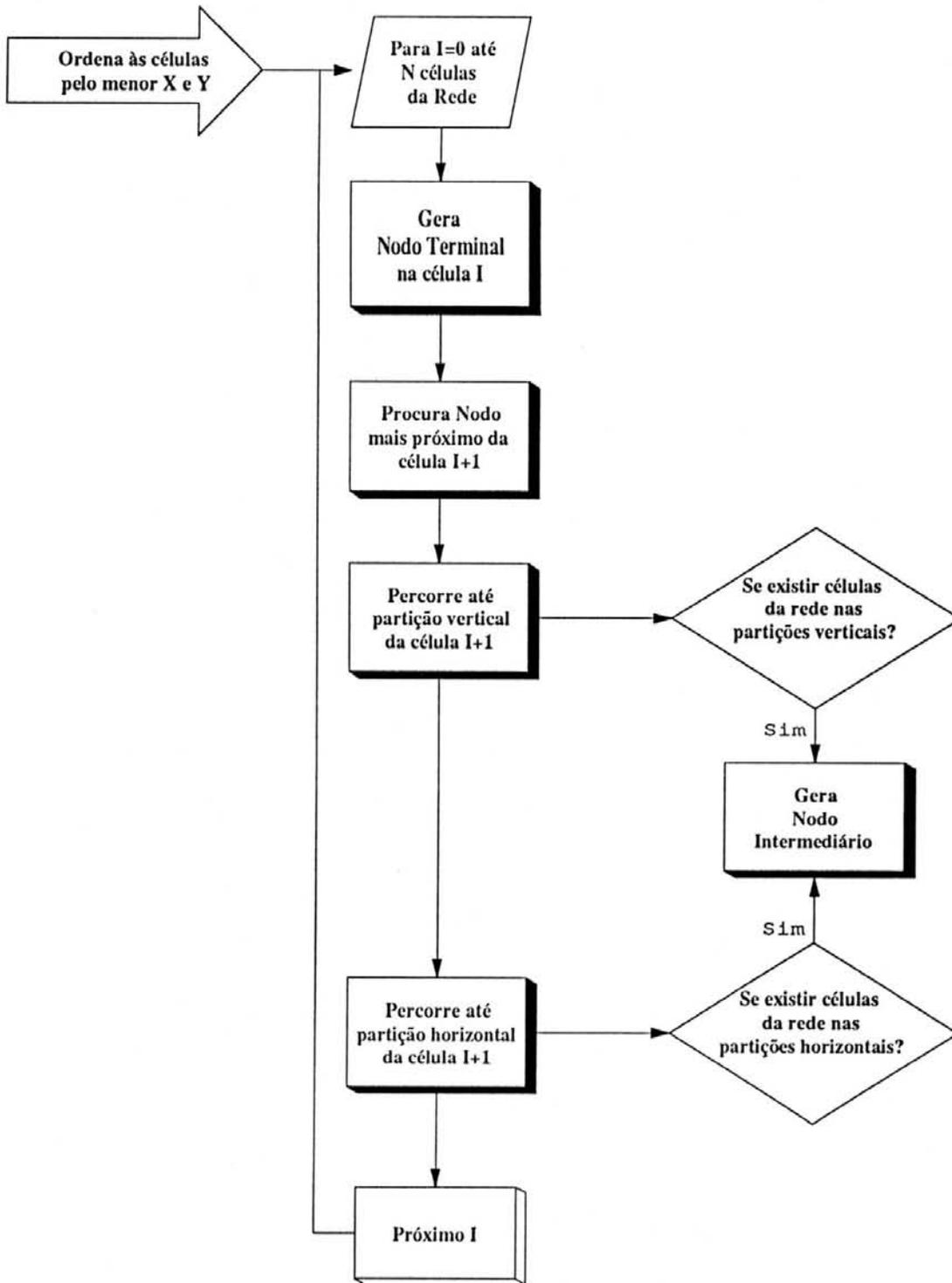


Figura 5.8 - O fluxo do algoritmo de geração de nodos.

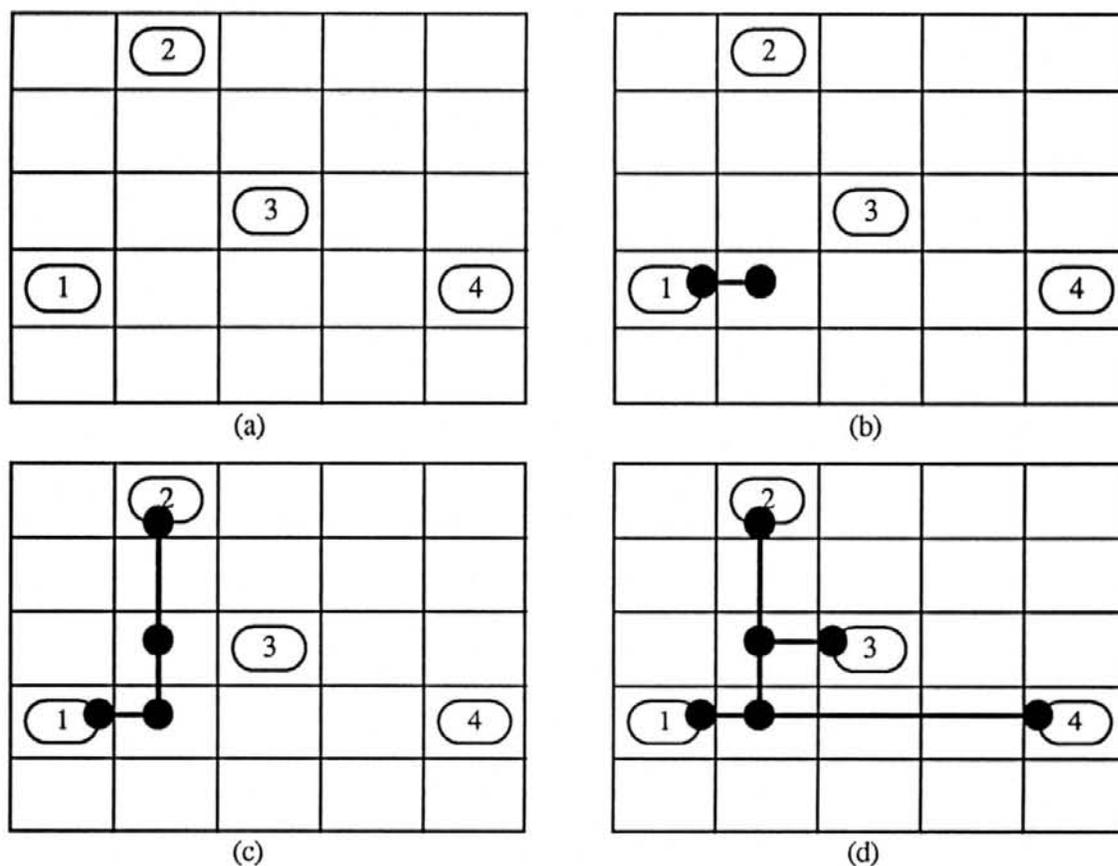


Figura 5.9 - A geração de nodos.

A utilização desta representação é importante para o funcionamento do mapa de congestionamento descrito na próxima seção.

5.6 O Mapa de Congestionamento

A partir do conceito de equilíbrio de conexões, seção 5.4, a montagem de uma representação que caracterize simultaneamente de maneira local e global as redes do circuito torna-se bastante útil para a realização do posicionamento, visando a distribuição de conexões no leiaute.

O mapa de congestionamento proposto é composto pelas linhas verticais e horizontais de corte de cada partição, seção 5.3, utilizada de acordo com a área do

envelope que se esteja analisando. Por exemplo, na figura 5.10b o mapa de congestionamento de um bloco é apresentado em separado do mapa com uma visão global. Este último aparece na figura 5.10a.

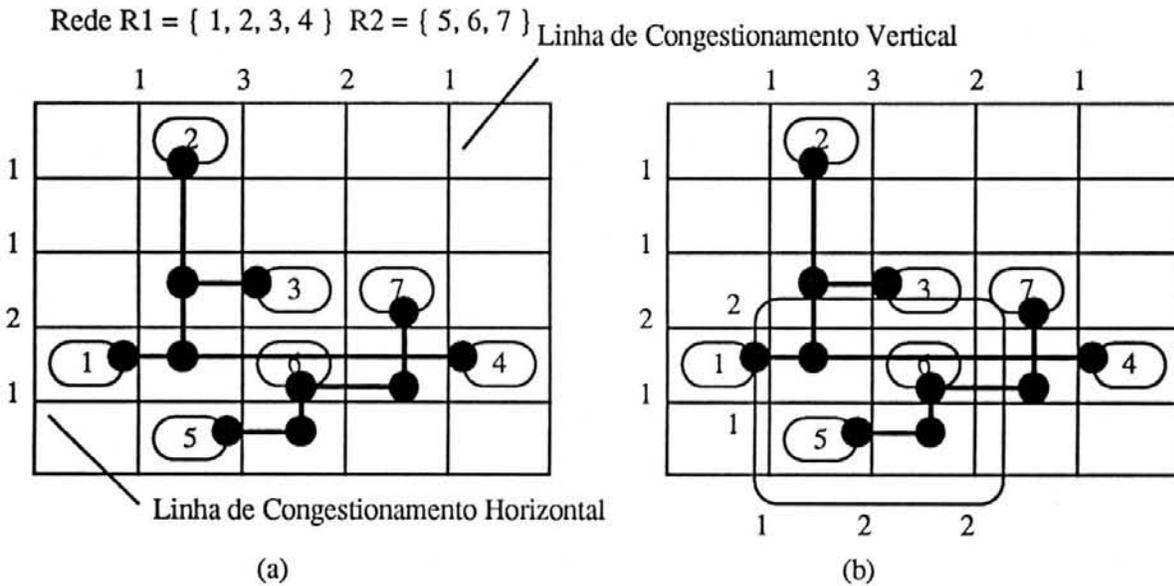


Figura 5.10 - O mapa de congestionamento.

Diretamente ligado à descrição do caminho das redes do circuito, o mapa é utilizado para garantir o sucesso do roteamento. Em condições de saturação, o mapa indica se há a necessidade de inserção de canais de roteamento para a realização do leiaute.

5.7 O Grafo de Adjacência de Redes

Com o propósito de construir uma estrutura de dados que represente o grau de agrupamento das células de um circuito, o grafo de adjacência ou intersecção, também utilizado por [HAG 92], denota uma boa modelagem para este tipo de representação.

O grafo de adjacência, figura 5.11, é formado a partir de um grafo de células, onde os vértices são as células e as arestas as ligações das células. No grafo de adjacência de

redes, os vértices são as redes e as arestas representam as adjacências com peso denotado conforme o propósito de ligação.

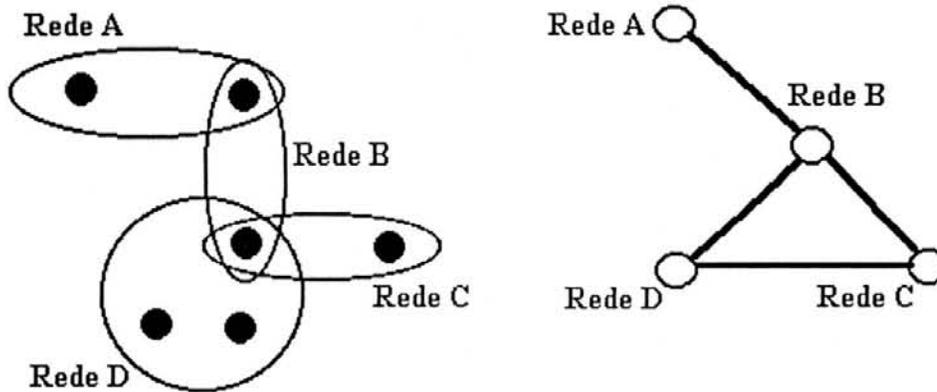


Figura 5.11 - O grafo de adjacência.

O uso do grafo de adjacência no TRAPP acrescenta um peso de adjacência às arestas do grafo, calculado a partir do grau de aglomeração das redes e modificado dinamicamente durante o posicionamento inicial a fim de definir uma melhor escolha. O grafo é utilizado como ponto de partida para modelar o posicionamento de cada rede.

Neste capítulo apresentou-se as principais características do algoritmo. No próximo capítulo veremos a arquitetura e o fluxo realizado por todas as fases do algoritmo.

6 O AMBIENTE TRAPP

Neste capítulo apresenta-se o funcionamento do particionador/posicionador TRAPP. O fluxo para a execução de cada etapa do algoritmo é apresentado em detalhes.

6.1 Visão Geral

O ambiente do algoritmo envolve a utilização de uma ferramenta complementar para visualização simbólica, o CIPPATO. Este ambiente estabelece a divisão do algoritmo em etapas, a fim de organizar as opções do sistema fornecidas pelo usuário e de estabelecer prioridades para as necessidades de particionamento/posicionamento da metodologia TRANCA. A figura 6.1 apresenta uma visão geral do ambiente TRAPP.

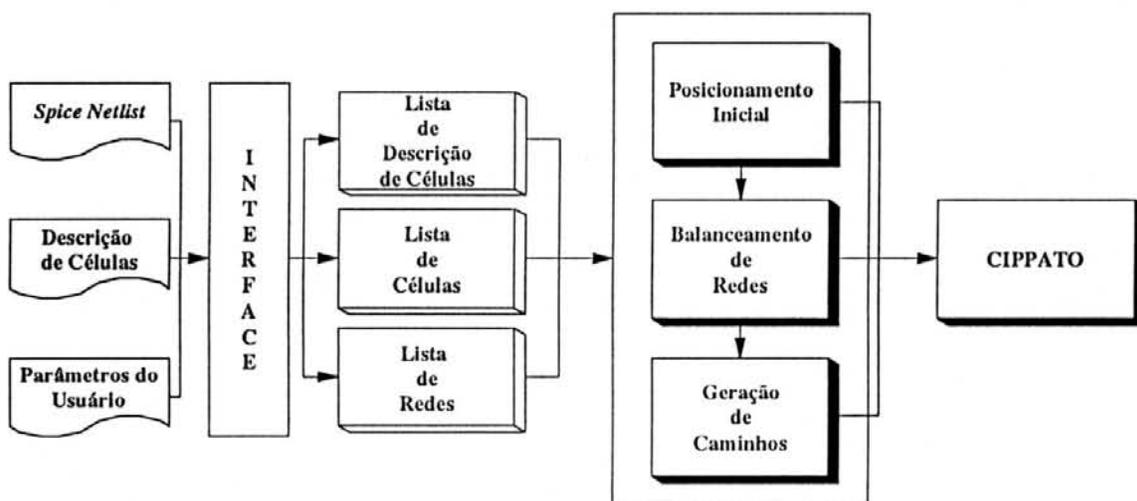


Figura 6.1 - O Ambiente TRAPP

Conforme visto na figura 6.1 o ambiente TRAPP se divide em três blocos principais: a interface com o usuário, o particionador/posicionador propriamente dito e a visualização externa através do CIPPATO.

A interface com o usuário se dá através de: uma *netlist spice*, descrevendo as conexões das células e as interfaces (pode ser hierárquica ou plana, desde que a descrição

das células seja correspondente); a descrição das células, contendo a altura, largura e a transparência da célula vertical e horizontal; e a definição de relação de aspecto. Um exemplo destes arquivos pode ser visto nos anexos A1 e A2.

O particionador/posicionador se divide em três etapas: particionamento/posicionamento inicial, o balanceamento de redes e geração dos caminhos das redes.

6.1.2 O Cippato

O CIPPATO é uma ferramenta para visualização do posicionamento que foi desenvolvida para complementar o ambiente TRAPP. Esta ferramenta possui as seguintes características:

- Visualização dos caminhos gerados. A partir de uma descrição destes caminhos o CIPPATO apresenta o posicionamento;
- Visualização para indicar o nível de ocupação das regiões do circuito apresentado. Neste modo as conexões são realizadas não efetuando a geração de linhas verticais e horizontais para representar as redes, mas gerando conexões ponto a ponto;
- Visualização por redes e níveis de metal. Neste caso, apresentase a rede ou o nível de metal separadamente.

6.2 As Fases do Algoritmo

Com o objetivo de se produzir resultados rapidamente, mas com um bom nível de posicionamento, e gerar posicionamentos mais refinados, para garantir a fase de roteamento, o algoritmo foi dividido em três fases. A primeira etapa já produz um posicionamento de bom nível que conforme o leiaute pode ser bastante próximo do resultado final. As duas etapas finais realizam o refinamento para complementar o posicionamento de leiautes mais complexos .

6.2.1 O Particionamento/Posicionamento Inicial

A partir do projeto de uma ferramenta que estabeleça o posicionamento em um menor tempo computacional e adaptável a diferentes abordagens, o melhor aproveitamento de procedimentos básicos necessários a qualquer algoritmo de particionamento/posicionamento é o principal objetivo desta etapa.

Esta etapa se inicia pela leitura da *netlist spice*, o arquivo de descrição simbólica de células e a relação de aspecto desejada. A partir da leitura destes arquivos são gerados a lista de células, a lista de descrição das células, a lista de redes e a área de posicionamento. A figura 6.2 apresenta os dados pertencentes à estrutura de lista de células e estrutura de lista de redes.

<pre> Estrutura Lista_de_Células { IDentificador; Lista_de_Reddes; Banda; IDdescritor; PosicaoX; Apontador Next_Célula; Instância_da_célula; }; </pre>	<pre> Estrutura Lista_de_Reddes { IDrede; Lista_de_Células; Envelope_da_Redde; Apontador Next_Redde; Caminho_da_rede; }; </pre>
--	---

Figura 6.2 - Estruturas da lista de células e da lista de redes

A heurística utilizada nesta etapa é o posicionamento de redes, ou seja, a seleção de uma rede a ser posicionada antecede a seleção de uma célula a ser posicionada. Como a principal característica deste algoritmo é dada pelo equilíbrio de conexões, seção 5.4, o posicionamento direcionado a nível de rede facilita a busca inicial de um equilíbrio das conexões. O posicionamento de redes é feito através do uso do grafo de intersecção, seção 5.7, montado a partir do relacionamento da lista de células e da lista de conexões. A figura 6.3 mostra o fluxo que estabelece esta etapa do algoritmo, na qual a montagem do grafo de intersecção é o primeiro passo.

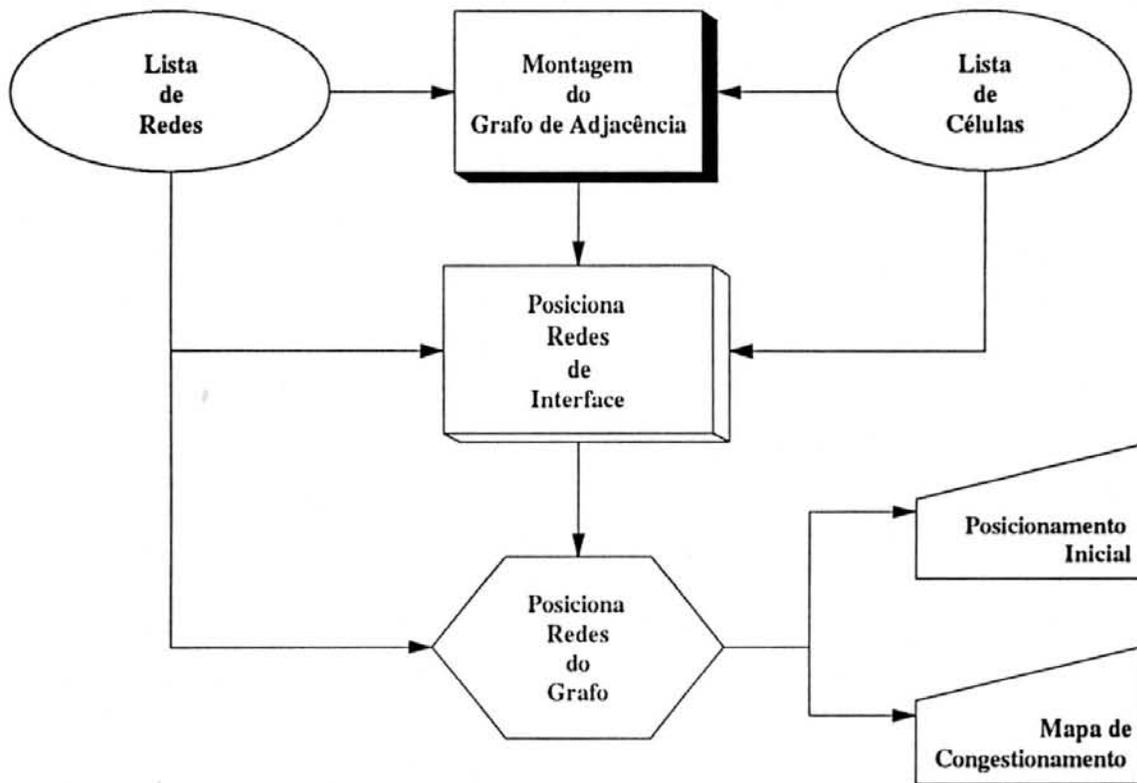


Figura 6.3 - O Fluxo do Particionamento/Posicionamento Inicial

A partir da montagem do grafo de intersecção o passo seguinte é o posicionamento das redes de interface (redes que contém os sinais de entrada e saída), utilizadas como redes sementes para iniciar o processo. O posicionamento das redes de interface é realizado conforme especificado na *netlist spice*. Após o posicionamento das redes de interface começa a seleção da próxima rede. A escolha da próxima rede é feita a partir de prioridades que se estabelecem para atribuir o peso das arestas do grafo de intersecção. As principais prioridades estabelecidas são:

- Número de células em comum, denota o grau de adjacência de uma rede;
- Número de células da rede já posicionadas, as redes com maior número de células já posicionadas tem preferência;
- Número de células da rede, rede com menos células são selecionadas primeiro.

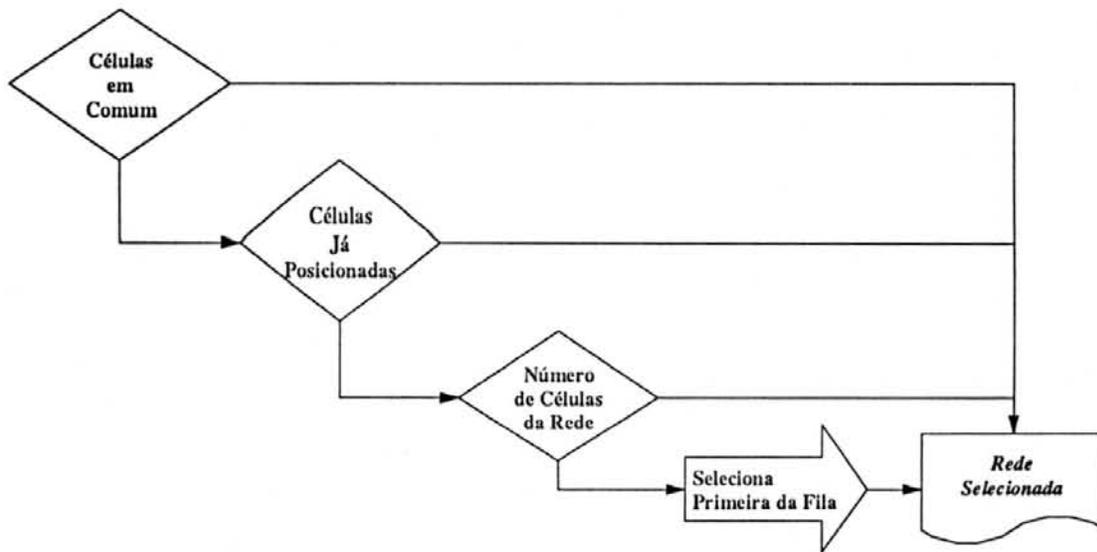


Figura 6.4 - A Seleção de uma Rede

A figura 6.4 apresenta o processo de seleção da próxima rede, onde, se houver empate nos três requisitos, seleciona-se a primeira da fila. Determinada a rede a ser posicionada, a próxima tarefa é a escolha de cada célula da rede. A escolha da célula é feita segundo dois procedimentos de classificação:

- Menor Envelope, segundo o gerenciamento do espaço físico, o envelope considera todas as redes a que a célula pertence;
- Equilíbrio das conexões, é realizado segundo o mapa de congestionamento, seção 6.3, escolhendo a melhor configuração entre as células em seleção.

Um aspecto importante nesta etapa é o gerenciamento físico que controla o posicionamento inicial para produzir um crescimento ordenado, considerando as células que ainda não foram posicionadas. O controle é feito sobre a ocupação de cada banda, nem sempre permitindo a geração do menor envelope para não comprometer o espaço destinado às próximas células.

Considerando uma divisão entre o particionador e o posicionador, ao particionador cabe a divisão do circuito em setores verticais e a divisão em bandas, conforme o andamento do posicionamento inicial das células, caracterizando o particionamento físico do circuito. O particionamento é realizado logicamente a partir da montagem do grafo de adjacência de

redes (células + conexões em comum). A atribuição de pesos às arestas do grafo, para a seleção das redes, é controlada pelo particionador, considerando o equilíbrio das conexões.

Ao posicionador fica a tarefa de decidir a melhor posição da célula dentro da banda, garantindo à célula o espelhamento que melhor otimizar as conexões por justaposição. A posição determinada é descrita simbolicamente em unidades de trilhas.

Após o posicionamento de todas as redes calcula-se o peso atual, o peso mínimo e o fator de balanceamento. Juntamente com o mapa de congestionamento e a lista de bandas, os pesos descrevem o posicionamento inicial necessário para começar a nova etapa do algoritmo.

6.2.3 O Balanceamento de Redes

A etapa de balanceamento de redes se caracteriza pela realização de movimentos das células a partir da seleção de redes com elevado fator de balanceamento. O balanceamento de uma rede busca o equilíbrio e, caso seja possível, o menor tamanho para as conexões.

O fluxo desta etapa, figura 6.5, inicia com o posicionamento inicial do circuito e o seu mapa de congestionamento, conforme o algoritmo de balanceamento de redes, figura 6.6. O primeiro passo é a geração da lista de redes em estágio de otimização e a lista de redes a otimizar. A lista de redes a otimizar é inicializada com todas as redes do circuito e a lista de redes em estágio de otimização é inicializada pela rede com o maior fator de equilíbrio, retirada da lista de redes a otimizar.

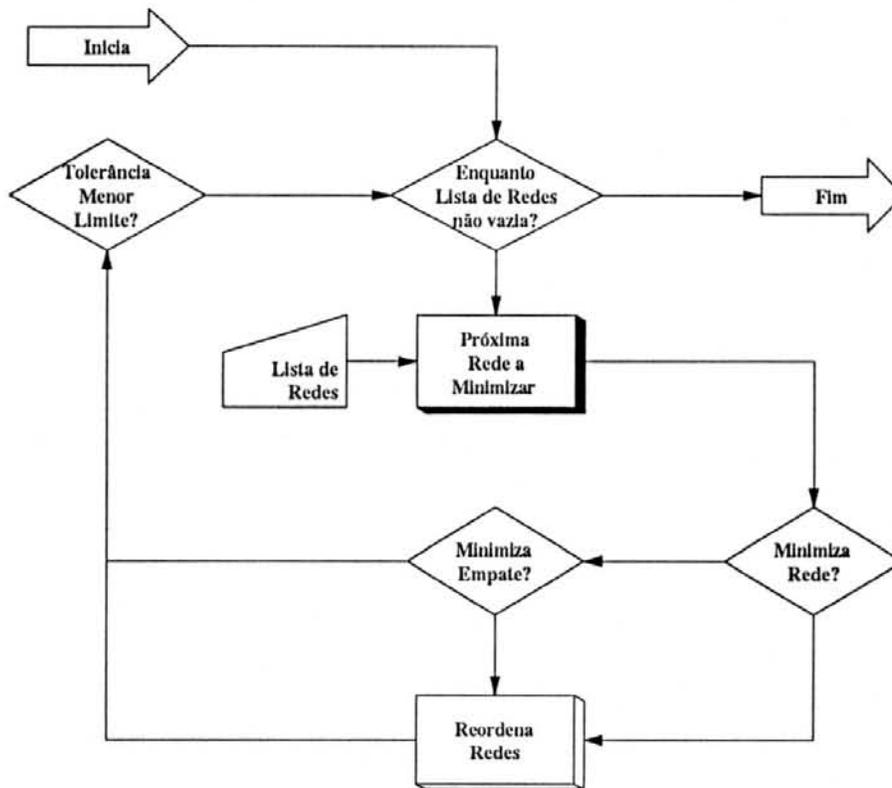


Figura 6.5 - O fluxo do balanceamento de redes

Ordena redes pelo maior fator de equilíbrio

Cria Lista_de_Espera_a_Minimizar

Cria Lista_para_Minimizar

Enquanto Lista_de_Espera não vazia

&& tolerância < limite faça

retorno = Minimiza_Rede (lista_para_minimizar)

Se retorno == MINIMIZADO

Se existe rede na lista_de_espera com fator > rede na lista_para_minimizar

Adiciona rede na lista_para_minimizar

Tolerância = 0

Senão se retorno == NÃO_FEZ_NADA

Se lista_de_espera não vazia && tolerância == limite

Adiciona rede na lista_para_minimizar

Se tolerância < limite

tolerância++;

Minimiza_Empate (lista_para_minimizar)

Fim do laço

Figura 6.6 - O algoritmo do balanceamento de redes

A partir da inicialização da lista de redes em estágio de otimização, a principal tarefa é minimizar a rede com maior fator de equilíbrio pertencente a esta lista. O estágio de minimização de uma rede compreende duas etapas diferentes:

- A minimização de uma rede com sucesso: envolve o movimento de células com diminuição do fator de equilíbrio e ganho na qualidade do posicionamento;
- A minimização de empate: envolve o movimento de células sem ganho de qualidade do posicionamento, mas possibilita o congelamento de balanceamento, com base em uma tolerância estabelecida pelo usuário.

A minimização de uma rede consiste na busca da compactação do envelope da rede, figura 6.7, envolvendo o movimento das células da rede. O movimento das células da rede em minimização pode ser o de troca de pares ou simplesmente o posicionamento de uma ou n células da rede em posições livres. Para a realização dos movimentos das células são avaliados dois fatores relevantes na qualificação do posicionamento:

- Ganho global em termos de redes adjacentes envolvidas: a minimização da rede selecionada deve ser tal que as redes adjacentes possam ter seu fator de equilíbrio elevado (isto significa um aumento de envelope) para diminuir o fator da rede em processamento, mas não possam ter o seu fator de equilíbrio maior do que o novo fator de equilíbrio da rede em minimização;
- A viabilidade do mapa de congestionamento: a partir de cada novo movimento de células, um novo mapa de congestionamento é produzido, indicando a viabilidade ou não do posicionamento.

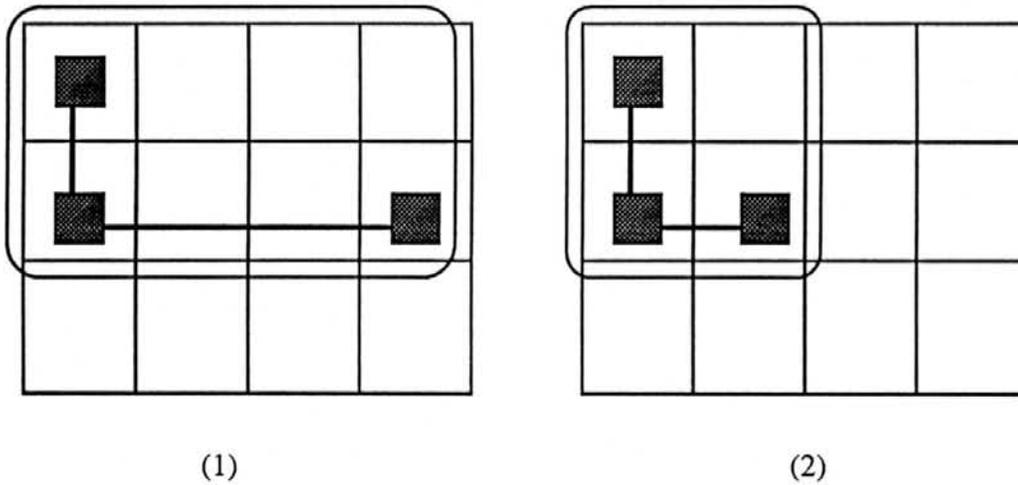


Figura 6.7 - A minimização de uma rede

Após a minimização de uma rede, o próximo passo é a reordenação das redes na lista de redes a otimizar e na lista de redes em otimização. As redes com fator de equilíbrio maior ou igual ao menor fator de equilíbrio de uma rede pertencente à lista de redes em otimização, são retiradas da lista a otimizar e inseridas na lista em otimização.

Conforme a figura 6.6, o algoritmo termina quando a lista de redes a otimizar estiver vazia e não existirem mais possibilidades de diminuir o fator de equilíbrio das redes, dentro dos limites de tolerância estabelecidos pelo usuário. A tolerância estabelecida visa flexibilizar a determinação do número de tentativas de otimização de uma rede.

Uma observação importante é que para cada tentativa de movimento de células o peso atual da rede, o fator de equilíbrio e mapa de congestionamento são determinados para qualificar o novo posicionamento. O maior gargalo de atualização e análise é quanto ao mapa de congestionamento, que envolve uma estrutura mais complexa, discutida na seção 6.3

6.2.4 A Geração dos Caminho das Redes

O propósito de geração dos caminho das redes é o de integrar a característica de roteamento global do TRAPP ao roteador a ser utilizado. No caso do sistema TRANCA a

integração com o MARTE, por ser um roteador do tipo maze, a ordem das conexões é importante para o sucesso do roteamento.

Os caminhos gerados pelo algoritmo dos caminhos das redes para configurar o mapa de congestionamento produz sempre o mesmo resultado. A maneira como o mapa de congestionamento é preenchido, seção 6.3, viabiliza a avaliação de saturação considerando-se a ocupação do envelope da rede. O objetivo desta etapa é, de a partir do mapa de congestionamento, detectar as regiões mais saturadas e modificar o caminho das redes sem alterar o tamanho do envelope.

A partir do mapa de congestionamento, a análise do envelope de cada rede do leiaute permite a localização das redes em regiões saturadas. Então, para cada rede, o algoritmo promove uma única vez uma série de otimizações, procurando melhorar a rede selecionada e aquelas inseridas no mesmo envelope. As operações de ajuste dos caminhos da rede são baseadas na análise dos nodos de controle da rede. Para cada nodo, conforme o número de caminhos, promove-se o movimento e geração de novos nodos, baseados em linhas de espelhamento, figuras 6.8, 6.9 e 6.10.

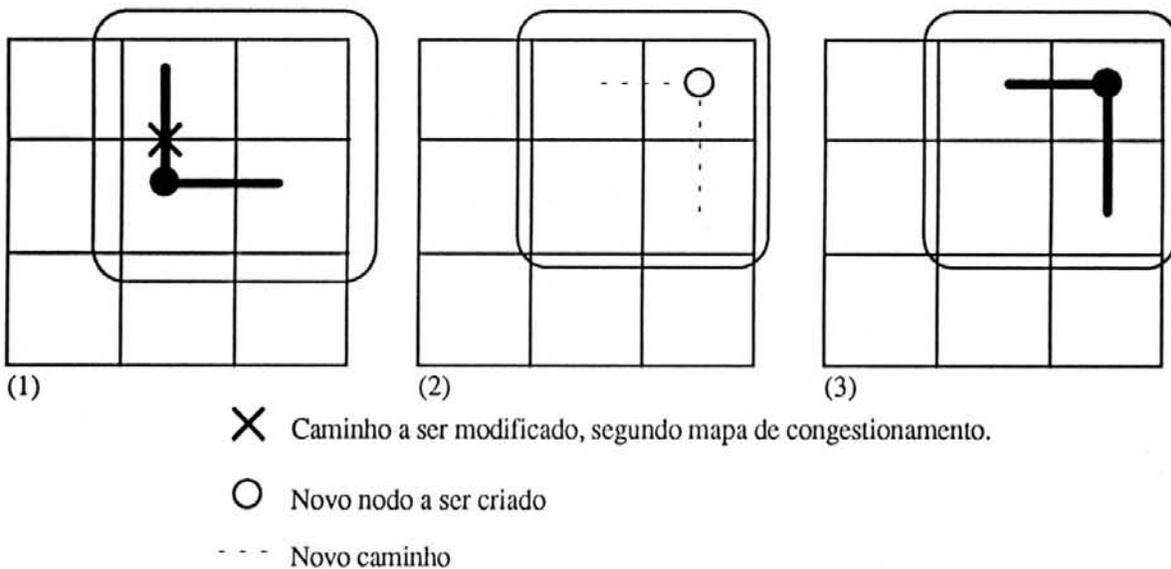


Figura 6.8 - Espelhamento de rede para o nodo tipo L.

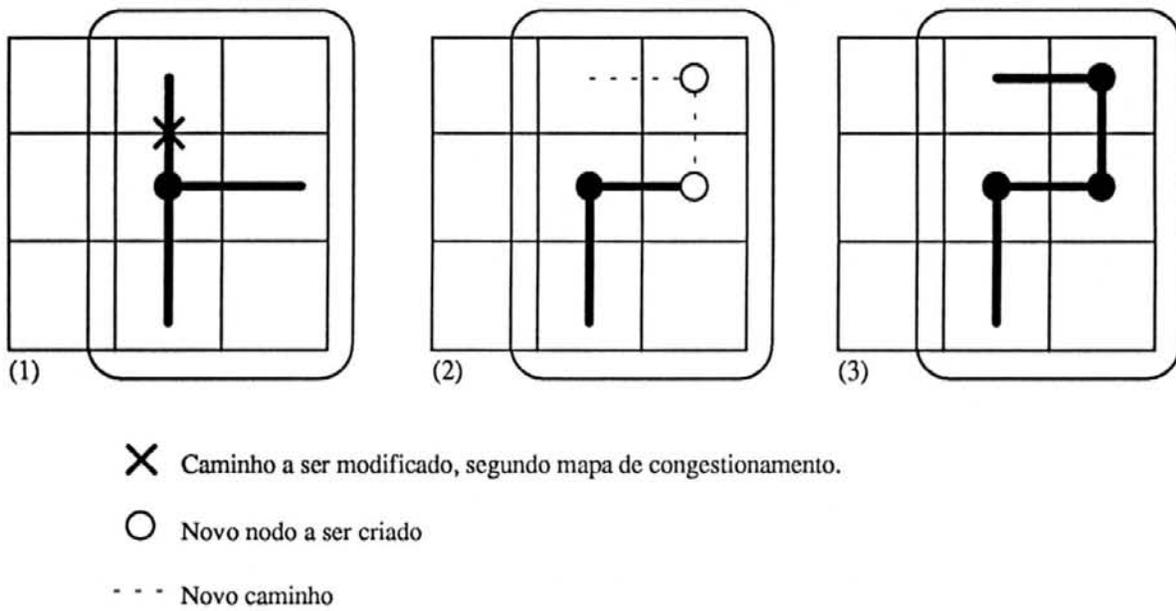


Figura 6.9 - Espelhamento de rede para o nodo tipo T

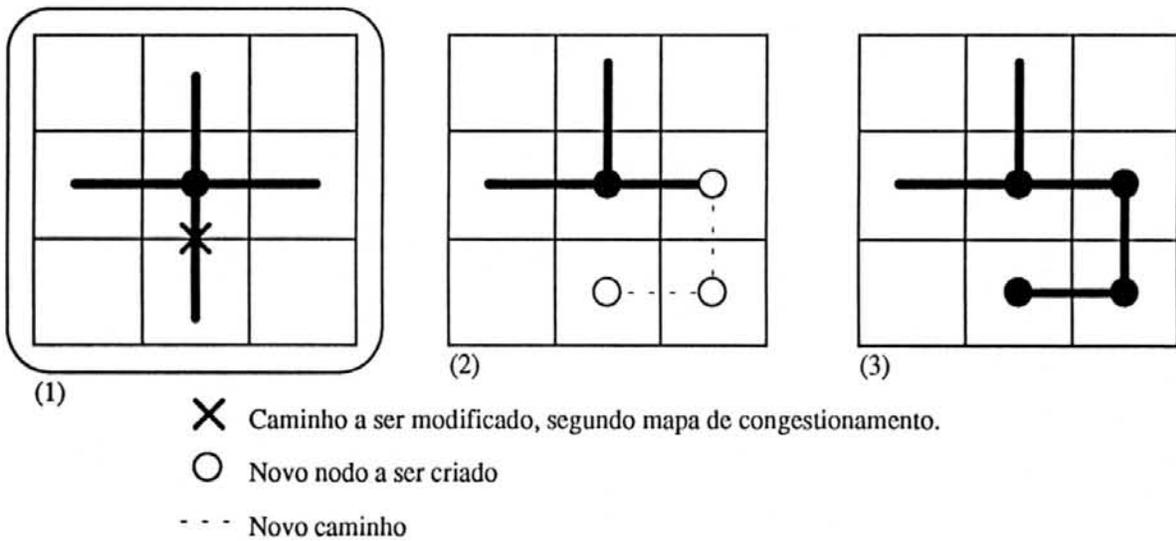


Figura 6.10 - Espelhamento de rede para os nodo tipo R

Nas figuras 6.8, 6.9 e 6.10 apresentamos o espelhamento para os três tipos de situações mais simples, sendo possível, com poucas atualizações e testes, a geração de novos caminhos. Na realidade, todas as três situações podem ser decompostas como o

tratamento de um nodo de duas conexões, apenas permanecendo o nodo se não forem retirado todos os cruzamentos de redes. Em cada figura observa-se três estágios: no primeiro o caminho a ser modificado é apresentado; no segundo estágio é feito o espelhamento do novo caminho; e no último estágio o novo caminho é produzido. Nas figuras 6.9 e 6.10 o espelhamento poderia ser feito por outro bloco, conforme indicação do status do mapa de congestionamento.

6.3 O Mapa de Congestionamento

O mapa de congestionamento está presente nas três etapas do algoritmo. Na primeira etapa o mapa é montado a medida que as células da rede vão sendo posicionadas. Na segunda etapa o posicionador modifica os caminhos da rede, tendo que atualizar as linhas de controle de cada partição envolvida nos movimentos de células. Na geração de caminhos o mapa de congestionamento é definitivamente atualizado com a ocupação correta de cada rede.

O princípio de montagem e utilização do mapa de congestionamento é baseado na estrutura das linhas de controle de um bloco, figura 6.11. Na estrutura de controle de uma linha de cada bloco existem duas variáveis importantes:

- O número de trilhas livres: altura da banda menos o número de trilhas ocupadas pelas redes, menos o número de trilhas ocupadas pela respectiva transparência (horizontal ou vertical). A ocupação de trilhas por transparência do bloco é dada pela célula com maior número de trilhas ocupadas internamente, no caso de haver mais de uma célula por bloco;

- O número de trilhas ocupadas pelas redes: no caso de dois níveis de metal , indica conexões em metal 1 ocupadas pelas linhas de controle vertical e indica conexões em metal 2 ocupadas pelas linhas de controle horizontal .

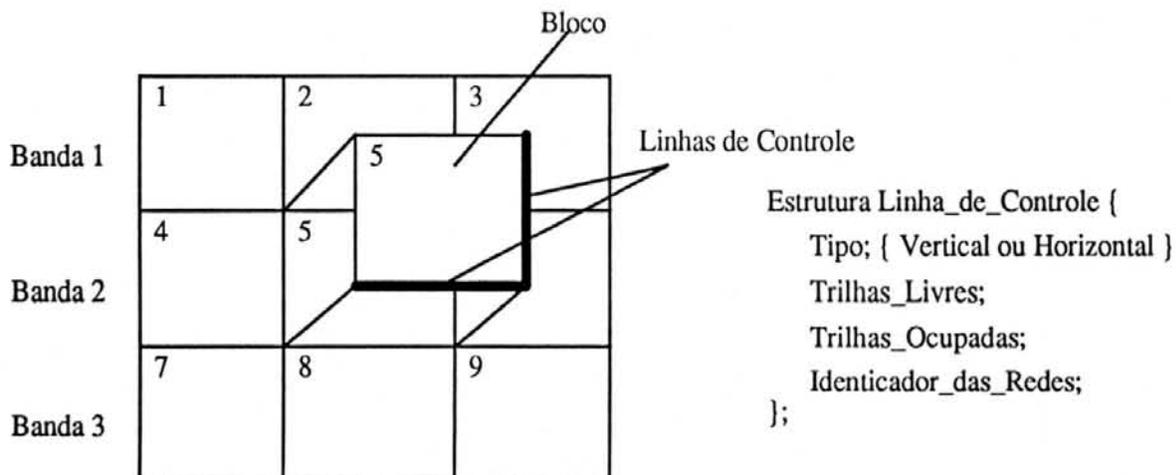


Figura 6.11 - Linhas de controle de um bloco

Para configurar o mapa de congestionamento é importante ressaltar a maneira como uma rede preenche as linhas de controle de cada bloco, e como cada rede é reordenada. Durante os movimentos de células e durante a geração do caminho definitivo de uma rede. Conforme os diferentes tipos de nodos apresentados, seção 5.5.2, uma estrutura para a representação do caminho de uma rede pode ser montada, figura 6.12, onde a ligação de dois nodos, simbolizando um caminho, permite o tratamento de conexões múltiplas como conexões dois a dois.

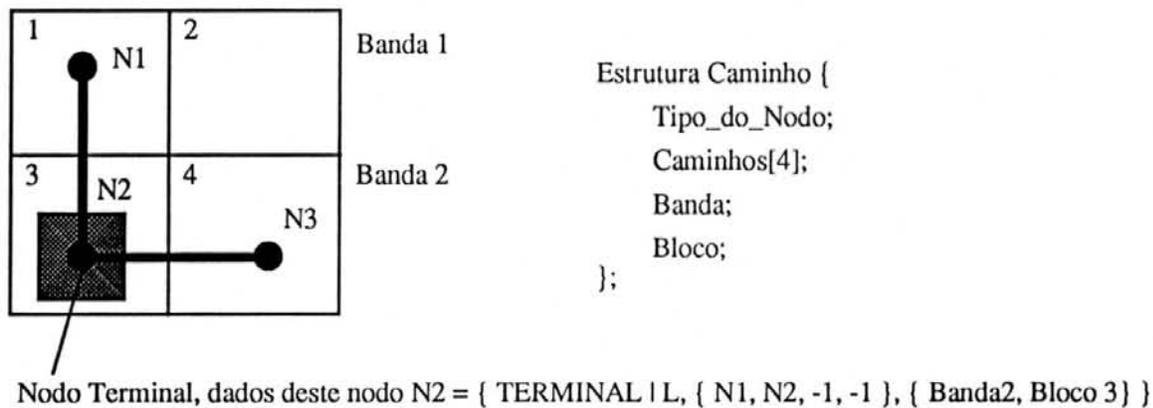


Figura 6.12 - A estrutura dos caminhos de uma rede

A manipulação do caminho das redes fica simplificada, permitindo em poucos passos a troca de posição das células ou dos nodos da redes, para a modificação do caminho.

O preenchimento do mapa de congestionamento não funciona necessariamente conforme o algoritmo de geração de caminhos da rede. Como o algoritmo sempre gera o mesmo caminho, procurando o menor, o principal objetivo do algoritmo de caminhos é quantificar o número de cruzamentos verticais e horizontais sobre as partições, a fim de qualificar o mapa de congestionamento. Portanto, o preenchimento do mapa de congestionamento utiliza trilhas livres dentro do envelope, e respeitar o tipo de conexão (metal 1 ou metal 2), ficando destinado à última fase do algoritmo a verificação de possíveis erros. Na figura 6.13(a) apresenta-se o caminho projetado, na figura 6.13(b) o caminho ocupado e na 6.13(c) as linhas de controle ocupadas para configurar o mapa de congestionamento. Como pode ser visto, os pontos de controle gerados pelo caminho previsto não são iguais aos preenchidos devido à ocupação no mapa de congestionamento. O caminho então produz cruzamentos para o preenchimento de uma linha de controle mais próxima.

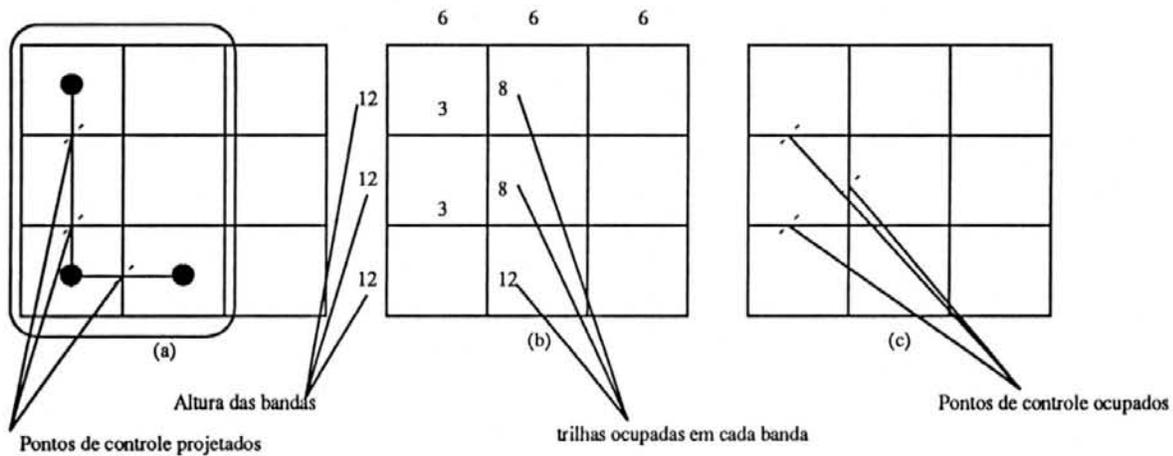


Figura 6.13 - O preenchimento do mapa de congestionamento.

7 AVALIAÇÃO E CONCLUSÕES DO ALGORITMO TRAPP

Neste capítulo apresenta-se os principais aspectos dos leiautes gerados e as conclusões. Nesta avaliação são considerados cinco *netlist spice* gerados a partir dos *benchmarks ISCAS 89*.

7.1 O Estágio Atual do Protótipo Avaliado

O protótipo do algoritmo TRAPP utilizado apresenta uma implementação das duas primeiras etapas de posicionamento. Cada etapa conta com uma série de rotinas auxiliares para verificar o andamento do algoritmo, fato este que não permite uma avaliação precisa do tempo computacional despendido. O protótipo foi desenvolvido em linguagem C++, funcionando em ambientes *UNIX* e *DOS*.

7.2 Metodologia Utilizada para Avaliação

Para a realização dos testes foi necessário utilizar algumas ferramentas para preparar a entrada do posicionador. A partir da biblioteca de células, que pode ser vista na tabela 7.1, os circuitos foram planejados.

Tabela 7.1 - A biblioteca de células utilizada.

Nome	Altura	Largura	Metal 1	Metal 2
inv	12	3	4	0
nand2	12	3	4	0
nand3	12	4	4	0
nand4	12	6	5	1
nand5	12	8	5	0
nor2	12	3	4	0
nor3	12	5	4	0

Obs: unidade simbólica em trilhas de roteamento.

A primeira etapa é a transformação do *netlist ISCAS* para um *netlist spice* através da ferramenta *SBNET*. A ferramenta *SBNET* além de transformar uma descrição *ISCAS* em *spice*, inclui os subcircuitos *spice* até o nível de transistores para a planificação do circuito, já que os circuitos só podem ter as células da biblioteca para a realização do roteamento.

Após a geração da *netlist spice* é feita a planificação do circuito através da ferramenta *MARFLAT*. A ferramenta *MARFLAT* planifica o *netlist* até o nível de transistores ou ao nível de células folha, conforme a descrição dos subcircuitos. Na tabela 7.2, apresentamos as células implementadas através de subcircuitos, mostrando as células da biblioteca utilizada para a planificação.

Tabela 7.2 - Células implementadas por subcircuito.

Biblioteca Nova Célula	inv	nand2	nand3	nand4	nor2	nor3
and2	1	1	X	X	X	X
and3	1	X	1	X	X	X
and4	1	X	X	1	X	X
or2	1	X	X	X	1	X
or3	1	X	X	X	X	1
ffd	1	4	X	X	X	X

Finalmente, após a *netlist* planificada o posicionamento é realizado, e o resultado é fornecido conforme mostra o apêndice A-3. A fim de caracterizar a qualidade do posicionamento, os circuitos foram roteados. Uma ferramenta de previsão de conexões necessárias ao roteamento, a ferramenta *PREV*, foi utilizada para dimensionar as necessidades de roteamento do circuito. Para comparar os resultados, o mesmo circuito foi também posicionado e roteado pelo *TRAMO*.

7.2.1 Os Circuitos Utilizados para Avaliação

Para a realização do posicionamento foram selecionados os *benchmarks ISCAS 89*. Os circuitos utilizados para a avaliação são descritos na tabela 7.3, onde são apresentados as células utilizadas, o número de redes, o número de células e o número de entradas e saídas.

Tabela 7.3 - Os circuitos utilizados e suas descrições.

	entradas	saídas	nands	invs	nors	Total
s27	4	1	17	6	6	29
s208	10	1	36	81	30	147
alu2	6	1	44	36	25	105
s386	7	7	117	165	35	317
s1196	14	14	309	360	151	1220

7.3 Análise dos Resultados Encontrados

Conforme apresentado na tabela 7.4, os circuitos posicionados apresentam resultados razoavelmente uniformes, produzindo índices de roteamento aproximados. A tabela 7.4 apresenta a área utilizada (em unidades de trilhas de roteamento), o número de conexões verticais e horizontais, o número de conexões não realizadas e o percentual de roteamento.

Tabela 7.4 - Os circuitos posicionados.

	Área	Previsão Vertical	Previsão Horizontal	Redes Roteadas	Redes Não roteadas	Percentual de Sucesso
s27	54x24	417	661	28	0	100%
s208	135x60	4815	6478	168	12	93.3%
s386	168x96	9957	10848	296	26	92%
s1196	252x156	47155	46691	1296	142	90.1%
alu2 #1	117x48	2758	4255	280	18	94%
alu2 #2	51x120	5526	2358	276	23	92.6%
alu2 #3	62x96	4392	2628	299	0	100%

Como se pode concluir da tabela, a exceção dos circuitos s27 e alu2#3, não se atingiu cem por cento de roteamento. O insucesso no roteamento dos circuitos é fruto de alguns problemas detectados durante os testes. Os principais problemas que foram detectados e prejudicaram o índice de roteamento são:

- Os caminhos gerados: a implementação inicial do mapa de congestionamento não se mostrou muito eficiente, além do roteador nem sempre considerar o mesmo roteamento projetado;
- O posicionamento intrabanda: a utilização de uma área muito compacta e próxima entre as células foi prejudicial por não considerar espaços para as conexões interbanda.

Para efeito de comparação realizaram-se três posicionamentos do circuito ALU2 (tabela 7.4). O primeiro posicionamento foi feito pelo TRAMO e os outros dois pelo TRAPP. Nos dois posicionamentos utilizando o TRAPP, verifica-se um índice de roteamento bastante semelhante. A área obtida nos dois posicionamentos é praticamente a inversão da altura pela largura, mostrando a boa receptividade do algoritmo a diferentes áreas. No posicionamento realizado pelo TRAMO, como pode ser visto na figura 7.1 existe uma série de espaços entre as células, o que resulta em uma área 15% maior do que a área utilizada pelo TRAPP. O posicionamento realizado pelo TRAPP pode ser visto na figura 7.2.

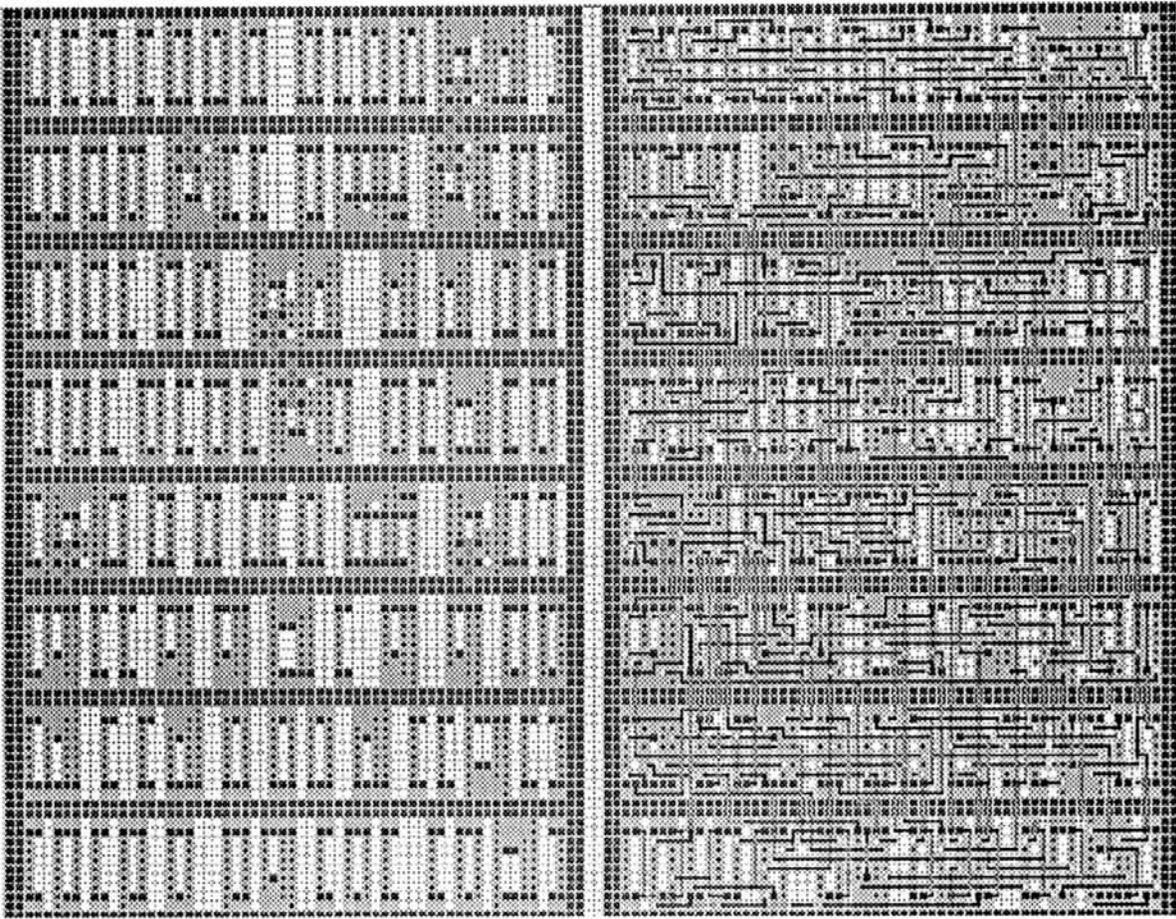


Figura 7.1 - Posicionamento do circuito alu2 via TRAMO.

As figuras 7.1 e 7.2 apresentam o posicionamento visualizado pela interface MEXEM do ambiente MARTE [JOH 92b]. Na figura 7.1 o lado esquerdo apresenta o circuito posicionado sem roteamento e ao lado o roteamento gerado pelo MARTE [JOH 92b].

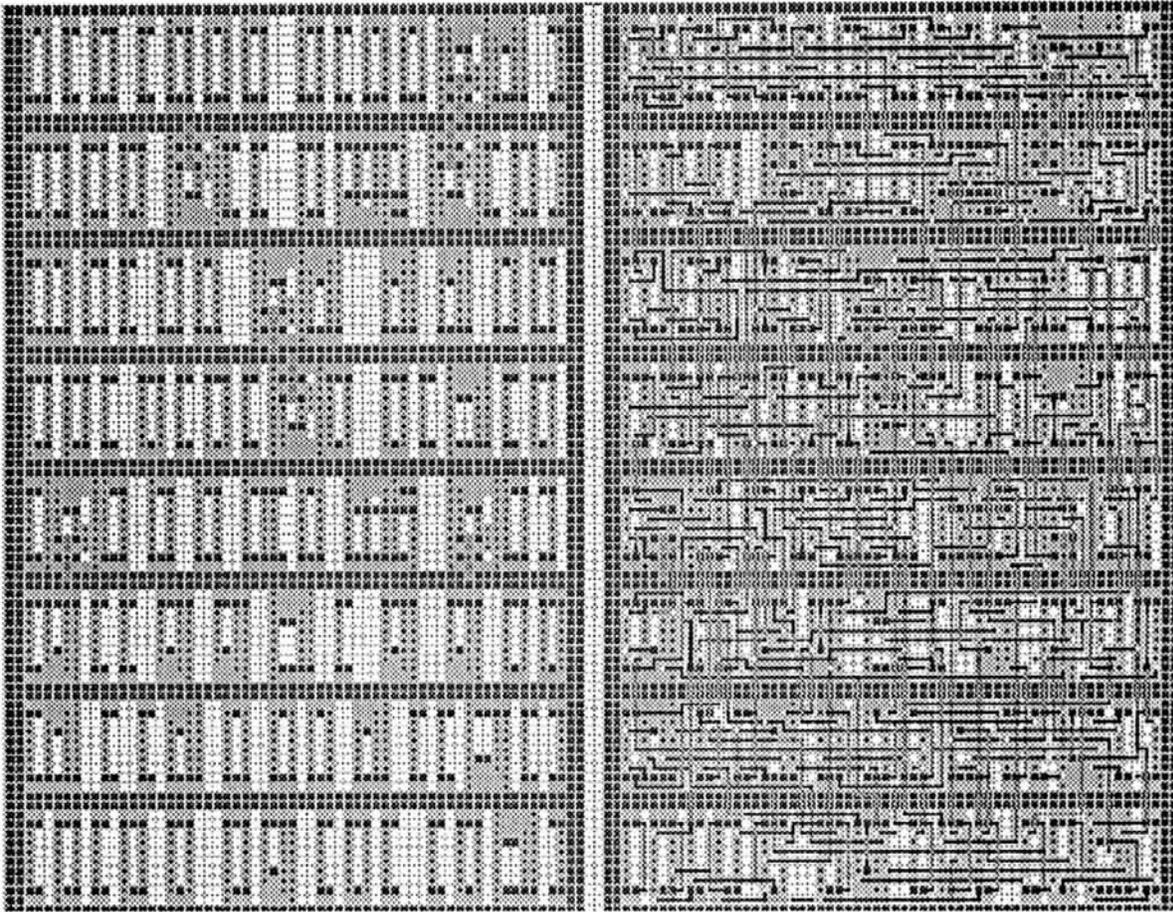


Figura 7.2 - Posicionamento do circuito alu2 via TRAPP.

7.4 Implementações Futuras

A partir da avaliação dos resultados obtidos verificou-se que uma série de pequenos ajustes que poderiam levar um melhor aproveitamento de diversas etapas, aumentando a performance de tempo do algoritmo, e flexibilizando algumas características para melhorar o índice de roteamento. As próximas etapas de implementação do TRAPP serão então:

- Geração dos caminhos das redes, para indicar ao roteador os caminhos e aumentar as possibilidades de roteamento;

- Opção de indicar um peso diferenciado às conexões verticais e horizontais, permitindo algumas facilidades conforme a transparência das células a serem utilizadas;
- Ajuste nas estruturas de dados, para aumentar a performance de tempo e para linearizar o algoritmo, principalmente na primeira etapa;
- A exploração de novos algoritmos baseados em árvores de *Steiner*, algoritmos que já considerem o congestionamento, trabalhando em conjunto com a distribuição de conexões e possibilitando aumentar a performance em termos de tempo.

7.5 Conclusões Finais

Os objetivos traçados neste trabalho para o planejamento e desenvolvimento de uma ferramenta de posicionamento para a metodologia TRANCA, em especial ao módulo TRAMOII, foram atingidos.

O protótipo da ferramenta TRAPP demonstrou que o algoritmo adotado, mesmo não estando completamente implementado, é bastante promissor por apresentar resultados positivos conforme evidenciado nos diversos experimentos realizados. A estratégia de distribuição das conexões conduziu a um posicionamento uniforme. O posicionamento foi testado para diferentes áreas de um mesmo circuito, e apesar de produzir resultados melhores em determinadas áreas, em geral os índices de melhora foram pequenos. Os resultados dos testes realizados, são de grande importância para a reformulação e o aperfeiçoamentos do algoritmo proposto.

ANEXO A-1 DESCRIÇÃO SPICE DE UM CIRCUITO

*****PROJETO TRAMOII / Circuito s27

X0 G10 313_X0 INV
X1 G10 G5 314_X0 NAND2
X2 G5 313_X0 315_X0 NAND2
X3 314_X0 316_X0 CK NAND2
X4 CK 315_X0 316_X0 NAND2
X5 G11 313_X1 INV
X6 G11 G6 314_X1 NAND2
X7 G6 313_X1 315_X1 NAND2
X8 314_X1 316_X1 CK NAND2
X9 CK 315_X1 316_X1 NAND2
X10 G13 313_X2 INV
X11 G13 G7 314_X2 NAND2
X12 G7 313_X2 315_X2 NAND2
X13 314_X2 316_X2 CK NAND2
X14 CK 315_X2 316_X2 NAND2
X15 G0 G14 INV
X16 G11 G17 INV
X17 G14 G6 63_X5 NAND2
X18 63_X5 G8 INV
X19 G12 G8 53_X6 NOR2
X20 53_X6 G15 INV
X21 G3 G8 53_X7 NOR2
X22 53_X7 G16 INV
X23 G16 G15 G9 NAND2
X24 G14 G11 G10 NOR2
X25 G5 G9 G11 NOR2
X26 G1 G7 G12 NOR2
X27 G2 G12 G13 NOR2

ANEXO A-2 ARQUIVO DE BIBLIOTECA DE CÉLULAS

.grid tramo0

.name tram95b.lib

INV	2	2	12	4	0	3	10	N	Y;
NAND2	4	3	12	4	0	3	6	N	Y;
NAND3	6	4	12	4	0	4	20	N	Y;
NAND4	8	6	12	5	1	6	12	Y	Y;
NAND5	10	8	12	5	0	7	20	N	Y;
NOR2	4	3	12	4	0	2	8	N	Y;
NOR3	6	5	12	4	0	4	10	N	Y;
begvcc	0	2	12	12	0	0	0	Y	Y;
endgnd	0	2	12	12	0	0	0	Y	Y;

ANEXO A-3 ARQUIVO DE POSICIONAMENTO GERADO

```
.circuito s27
.xpoints 54
.ypoints 24
.library tram95b
X0 0 0 0 0 inv G10 313_X0 ;
X1 3 0 0 0 nand2 G10 G5 314_X0 ;
X24 7 0 0 0 nor2 G14 G11 G10 ;
X3 11 0 0 0 nand2 314_X0 316_X0 CK ;
X4 15 0 0 0 nand2 CK 315_X0 316_X0 ;
X8 19 0 0 0 nand2 314_X1 316_X1 CK ;
X9 23 0 0 0 nand2 CK 315_X1 316_X1 ;
X13 27 0 0 0 nand2 314_X2 316_X2 CK ;
X14 31 0 0 0 nand2 CK 315_X2 316_X2 ;
X6 35 0 0 0 nand2 G11 G6 314_X1 ;
X7 39 0 0 0 nand2 G6 313_X1 315_X1 ;
X17 43 0 0 0 nand2 G14 G6 63_X5 ;
X5 47 0 0 0 inv G11 313_X1 ;
X16 50 0 0 0 inv G11 G17 ;
X2 1 12 0 1 nand2 G5 313_X0 315_X0 ;
X15 5 12 0 1 inv G0 G14 ;
X25 8 12 0 1 nor2 G5 G9 G11 ;
X23 12 12 0 1 nand2 G16 G15 G9 ;
X20 16 12 0 1 inv 53_X6 G15 ;
X19 19 12 0 1 nor2 G12 G8 53_X6 ;
X26 23 12 0 1 nor2 G1 G7 G12 ;
X27 27 12 0 1 nor2 G2 G12 G13 ;
X10 31 12 0 1 inv G13 313_X2 ;
X11 34 12 0 1 nand2 G13 G7 314_X2 ;
X12 38 12 0 1 nand2 G7 313_X2 315_X2 ;
X18 42 12 0 1 inv 63_X5 G8 ;
X21 45 12 0 1 nor2 G3 G8 53_X7 ;
X22 49 12 0 1 inv 53_X7 G16 ;
```

BIBLIOGRAFIA

- [BAP 91] BAPAT, S.; COHOON, J. P. Sharp-Looking Geometric Partitioning. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. p.172- 175.
- [BRE 77] BREUER, M. A. A Class of Min-cut Placement Algorithms. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 14., 1977, New Orleans. **Proceedings...** New York: IEEE, 1977. p.284-290.
- [FID 82] FIDUCIA, C. M.; MATHEYSES, R. M. A Linear-Time Heuristic for Improving Network Partitions. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 19., 1982, Las Vegas. **Proceedings...** New York: IEEE, 1982. p.175-181.
- [GAJ 92] GAJSKI, Daniel D.; DUTT, N. D.; WU, A. C-H., et al. Partitioning. In: **High-level Synthesis - Introduction to Chip and System Design.** Massachusetts: Kluwer Academic, 1992. 359p. p.179-212.
- [GÜN 93] GÜNTZEL, José L. A. **Geração de Circuitos Utilizando Matrizes de Células Pré-difundidas.** Porto Alegre: CPGCC da UFRGS, 1993. 174p.Dissertação de Mestrado.
- [HAG 92] HAGEN, LARS. **New Spectral Methods for Ratio Cut Partitioning and Clustering.** In: IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN, [S.l.], p. 1074-1085, Sept. 1992.
- [HAN 76] HANNAN, M. P. K. et al. Some Experimental Results on Placement Techniques. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 13., 1976, San Francisco. **Proceedings...** New York: IEEE, 1976. p 214-224.

- [JOH 92] JOHANN, Marcelo de Oliveira. **MARTE - Ambiente de Roteamento Simbólico**. Porto Alegre: Instituto de Informática da UFRGS, 1992. 106p. Projeto de Diplomação.
- [JOH 92a] JOHANN, Marcelo de Oliveira. **Desenvolvimento do Sistema de Roteamento Simbólico MARTE**. Porto Alegre: Instituto de Informática da UFRGS, 1992. 74p. Projeto de Diplomação.
- [JOH 94] JOHANN, M.; REIS, R. A. da L. Performance Improvements and Results of the MARTE Routing System. In: SEMINÁRIO INTERNO DA MICROELETRÔNICA, 9., 1994, Porto Alegre. **Anais...** Porto Alegre: CPGCC da UFRGS, 1994. p.29-32.
- [KER 70] KERNIGHAN, B.; LIN, S. An Efficient Heuristic Procedure for Partitioning Graphs. **The Bell System Technical Journal**, New York, v. 49, n. 2, p.291-307, Feb 1970.
- [KRI 84] KRISHNAMURTHY, B. An Improved Min-Cut Algorithm for Partitioning VLSI Networks. **IEEE Transactions on Computers**, New York, v.33, n.5, p.438-446, May 1984.
- [LEE 61] LEE, C. An Algorithm for Path Connections and Its Applications. **Transactions on Electronic Computers**, [S.l.], v. 10, p.346-365, Sept. 1961.
- [LUB 90] LUBASZEWSKI, Marcelo S. **Geração Automática de Lógica Aleatória Utilizando a Metodologia TRANCA**. Porto Alegre: CPGCC da UFRGS, 1990. 232p. Dissertação de Mestrado.
- [MAL 94] MÄHLMANN, L. G. G.; FRAGOSO, J. L.; REIS, R. A. L. TENTOS FOR WINDOWS: A Microeletronics CAD Tool Box. In: SEMINÁRIO INTERNO DA MICROELETRÔNICA, 9., 1994, Porto Alegre. **Anais...** Porto Alegre: SIM da UFRGS, 1994. p.57-60.

- [MAR 89] MARCHIORO, G. F.; CARRO, L. Editor Simbólico para Cicuitos Integrados. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE MICROELETRÔNICA, 4., 1989, Porto Alegre. **Anais...** Porto Alegre: SBMICRO da UFRGS, 1989. p.837-848.
- [MOR 90] MORAES, Fernando G. **TRAGO - Síntese Automática de Leiaute para Circuitos em Lógica Aleatória.** Porto Alegre: CPGCC da UFRGS, 1990. 199p. Dissertação de Mestrado.
- [MOR 90a] MORAES, F. G.; PEREIRA, C. E.; MARCHIORO, G. F. **ESQUELETO - Editor de Esquemas Elétricos.** In: CONGRESSO DA SOCIEDADE BRASILEIRA DE MICROELETRÔNICA, 5., 1990, Campinas. **Anais...** Campinas: SBMICRO, 1990.
- [MOR 91] MORAES, F. G.; REIS, R. A. L. **TENTOS - Gerenciador de Software para Microeletrônica.** Porto Alegre: CPGCC da UFRGS, 1991. 36p. (RP- 155).
- [PRE 88] PREAS, B. T.; KARGER, P. G. Placement, Assignment and Floorplanning. In: PREAS, Bryan T. **Physical Design Automation of VLSI Systems.** Menlo Park: Benjamin/Cummings, 1988. p. 87-155.
- [REI 83] REIS, Ricardo A. L. **TESS: Evalueur Topologique Predictif pour la Génération Automatique des Plans de Masse de Circuits VLSI.** Grenoble: Institut Polytechnique de Grenoble, 1983.
- [REI 85] REIS, R. A. L. Estratégias Básicas para a Concepção Automática de Layout de Circuitos em Lógica Aleatória. In: **SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 2., 1985, Porto Alegre. Anais...** Porto Alegre: SBC, 1985. p.223-36.

- [REI 87] REIS, R. A. L. A New Standard Cell CAD Methodology. In: ACM/IEEE CUSTOM INTEGRATED CIRCUITS CONFERENCE, 1987, Portland. **Proceedings...** New York: IEEE, 1987. p. 385-388.
- [REI 89] REIS, A. I. et al. MODEM - Desenvolvimento de um ASIC para Modens de Banda Base. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE MICROELETRÔNICA, 4., 1989, Porto Alegre. **Anais...** Porto Alegre: SBMICRO da UFRGS, 1989. p.617-626.
- [REI 91] REIS, A. I.; REIS, R. A. da L. TRAMOII - Proposta para um Gerador de Leiaute Baseado em Células para Circuitos CMOS Digitais com dois Níveis de Metal. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 6., 1991, Campinas. **Anais...** Campinas: SBC/SBMICRO/CTI, 1991. p.90-99.
- [REI 93] REIS, A. I. **Geração de Células Transparentes.** Porto Alegre: CPGCC da UFRGS, 1993. 128p. Dissertação de Mestrado.
- [SAN 89] SANCHIS, L. A. Multiple-Way Network Partitioning. **IEEE TRANSACTIONS ON COMPUTERS**, New York, v.38, n.1,p.62-80, Jan. 1989.
- [SCH 72] SCHWEIKERT, D. G.; KERNIGHAN, B. W. A Proper Model for the Partitioning of Electrical Circuits. In: ACM/IEEE DESIGN AUTOMATION WORKSHOP, 9., 1972. **Proceedings...** New York: IEEE, 1972. p. 57-62.
- [SHA 90] SHAHOOKAR, K.; MAZUMMBER, P. Gasp - A Generic Algorithm for Standard Cell Placement. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. p. 660 - 665.

- [SHE 93] SHERWANI, Naveed A. **Algorithms for VLSI Physical Design Automation**. Massachusetts: Kluwer Academic, 1993. 488p.
- [SOT 91] SOTILLE, Mauro A. et al. **Manual do Sistema TRAMO - Versão 1.0**. Porto Alegre: CPGCC da UFRGS, 1991, 142p. (RP-144).
- [SOU 91] SOUZA, Carlos E. S. **Uma Interface Gráfica para o Projeto TRANCA em Ambiente UNIX**. Porto Alegre: Instituto de Informática da UFRGS, 1991. Projeto de Diplomação.
- [STR 91] STRAVERS, P. Partitioning a Network into n Pieces with a Time Efficient Net Cost Function. In: ACM/IEEE EUROPEAN DESIGN AUTOMATION CONFERENCE, 1991, Amsterdam. **Proceedings...** [S.l.]: IEEE, 1991. p. 177-182.
- [TOD 86] TODESCO, A. **Manual do Sistema RS**. Porto Alegre: CPGCC da UFRGS, 1986.
- [WAG 87] WAGNER, F. R.; FREITAS, C. M. D. S. NILO - **Uma Linguagem para a Descrição de Hardware no nível de Portas Lógicas**. Porto Alegre: Pós-Graduação em Ciência da Computação da UFRGS, Março 1987. (RP-66).

Informática



UFRGS

*TRAPP Uma Ferramenta para Particionamento/
Posicionamento de Células para a Metodologia TRANCA*

por

Paulo Armando Schermer

Dissertação apresentada aos Senhores:

Wilhelmus A. M. Van Noije

Prof. Dr. Wilhelmus A. M. Van Noije (USP)

Ricardo Jacobi

Prof. Dr. Ricardo Pezzuol Jacobi

Marcelo Lubaszewski

Prof. Dr. Marcelo Lubaszewski (DELET/UFRGS)

Vista e permitida a impressão.
Porto Alegre, 03/06/96.

Prof. Dr. Ricardo Augusto da Luz Reis

Prof. Dr. Ricardo Augusto da Luz Reis,
Orientador.

Flávio Rech Wagner

Prof. Flávio Rech Wagner
Coordenador do Curso de Pós Graduação
em Ciência da Computação - CPG
Instituto de Informática - UFRGS