

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

269533

**Síntese Automática de
Células CMOS**

por

MARCUS KINDEL

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

Prof. Ricardo Augusto da Luz Reis
Orientador

Porto Alegre, outubro de 1997



UFRGS

SABi



05230109

UFRGS
INSTITUTO DE INFORMÁTICA

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Kindel, Marcus

Síntese Automática de Células CMOS / Marcus Kindel. - Porto Alegre: CPGCC da UFRGS, 1997.

79 f.:il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BR - RS, 1997. Orientador: Reis, Ricardo A. L.

1. Geração de Células. 2. Síntese Física. 3. Sistema TRAMO2. 4. Sistema TRAMO3. 5. Projeto TRANCA 6. Geração de Módulos. 7. Projeto de Circuitos Integrados. I. Reis, Ricardo Augusto da Luz. II. Título.

Engenharia elétrica - SBU
Microeletrônica
Gerador: células
Gerador: módulos

CNPq 3 04 03 00 - 6

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA			
N.º CHAMADA: 621.38-181.4(043) K513		N.º REG.: 33802	
ORIGEM: 1		DATA: 15/06/00	CHEQUE: 18.20,00
FUNDO: II	FORN.: II		

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Hélió Trindade

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

Diretor do Instituto de Informática: Prof. Roberto Tom Price

Coordenador do CPGCC: Prof. Flávio Wagner

Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira

Agradecimentos

Muitas pessoas foram fundamentais para a realização desse curso de mestrado. Com certeza, deixarei de citar muitas aqui. Mas a todas, segue o meu reconhecimento e gratidão mais sinceros.

Agradeço a Ricardo Reis pela sua paciência, amizade, orientação e otimismo, que foram fundamentais em todas as etapas de desenvolvimento desse trabalho.

Aos professores Altamiro Suzim, Sergio Bampi, Tiaraju Wagner e Ricardo Jacobi por todas as informações trocadas no decorrer desses anos, bem como pela amizade conquistada ao longo dos anos de convívio.

A César Marcon e André Reis que foram os responsáveis pelas minhas primeiras atividades ainda como auxiliar de pesquisa junto ao grupo, as quais me motivaram a seguir trabalhando nessa área.

Aos colegas José Luís Güntzel e Marcelo Johann pelo convívio diário, pela troca de idéias, pelos trabalhos em equipe, e pelas inúmeras demonstrações de amizade.

A Luigi Carro que, mais do que um professor, tornou-se um grande amigo cujos comentários, sugestões, exigências foram fundamentais na minha formação e cuja capacidade de trabalho é um modelo para mim.

A Fernando Moraes pelos comentários e sugestões apresentados.

Aos meus auxiliares Daniel Branco, Juliana Klas e Artur Severo cuja dedicação e empenho, em muito contribuíram para a concretização desse trabalho.

Aos colegas André Aita, Gustavo Kayser, Jung Choi, Carlos Alba, Luis Gustavo Mählmann, Oscar Hoyos, José Gomez, Luiz Pilger, Érika Cota e Márcio Kreutz, pela amizade e companheirismo.

Aos meus companheiros de graduação e início de jornada na microeletrônica André Hentz e César Crusius.

Aos funcionários da secretaria do Curso de Pós Graduação em Ciência da Computação, do Instituto de Informática, dos laboratórios e da biblioteca pela atenção e presteza com que sempre me trataram.

À Marcela Castro da Silva pelo apoio, compreensão e amizade demonstrados na reta final de desse trabalho.

Aos meus pais que nunca mediram esforços no sentido de me dar a melhor educação possível, que são os responsáveis pelo que sou e que souberam me compreender e me suportar nesse período de ausência quase completa.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico e ao Povo Brasileiro, responsáveis pelo auxílio de custo recebido durante o desenvolvimento desse trabalho.

Portanto, a todos que contribuíram no desenvolvimento desse trabalho, citados aqui ou não: MUITO OBRIGADO.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Sistema de Bibliotecas da UFRGS

K 38802

Sumário

Sumário	5
Lista de Abreviaturas	8
Lista de Figuras	9
Lista de Tabelas	10
Resumo	11
Abstract	12
1 Introdução e motivação	13
1.1 Introdução	13
1.2 Organização da dissertação	13
2 Síntese de Leiaute	15
2.1 Introdução	15
2.2 Estratégias de Projeto	15
2.2.1 Introdução	15
2.3 Estilos de Leiaute	16
2.3.1 CI's personalizáveis por todas as máscaras	17
2.3.2 Geração de módulos em lógica aleatória	17
2.3.2.1 Leiaute <i>full-custom</i> :	18
2.3.2.2 <i>Gate-Matrix</i>	19
2.3.2.3 Geradores baseados em células	19
2.4 Análise da síntese automática com geração de células	19
3 A Metodologia TRANCA	21
3.1 Definição	21
3.2 Características	21
3.2.1 Estrutura de Bandas	21
3.2.2 Transparência efetiva	22
3.2.3 Maleabilidade	22
3.2.4 Gerenciamento de trilhas	22
3.3 O Projeto TRANCA	22

3.3.1	Histórico	23
3.3.2	O Módulo TRAMO	23
3.3.2.1	Topologia das células	24
3.3.2.2	Características dos leiautes produzidos pelo TRAMO	25
3.3.3	O módulo TRAGO	25
3.3.4	O Sistema TROPIC	26
3.3.5	O Sistema MARCELA	27
4	Os Sistemas TRAMO2 e TRAMO3	29
4.1	A Estrutura do sistema TRAMO2	29
4.2	O roteador MARTE	30
4.3	Implementação preliminar	32
4.4	O Sistema TRAMO3	36
4.5	Diferenças entre TRAMO3 e TROPIC	38
5	A Topologia das Células para o TRAMO2	40
5.1	Análise prévia	40
5.2	Transparência Efetiva	40
5.3	Prioridade na alocação de trilhas	42
5.4	Posicionamento básico de transistores	43
5.5	Gerenciamento de altura variável	43
6	A Geração Automática de Células	44
6.1	Considerações iniciais	44
6.2	Definição de Conceitos	45
6.3	Processamento da entrada	46
6.4	Posicionamento dos transistores	46
6.5	Roteamento das conexões	47
6.6	Compactação da descrição simbólica	48
6.7	Estado da ferramenta	49
6.8	Estado do sistema	49
7	Resultados	51
7.1	Implementação de um circuito lógico	51
7.2	Implementação de um bit somador	52

8	Conclusões e trabalhos futuros.....	54
	Anexo A-1 Biblioteca de Células para o TRAMO2.....	55
	Anexo A-2 Descrição do parser de entrada	63
	A-2.1 Descrição do Analisador Sintático para SPICE em YACC	63
	Anexo A-3 Novas topologias para as partes básicas	68
	A-3.1 Topologia A.....	68
	A-3.2 Topologia B.....	69
	A-3.3 Topologia C.....	70
	A-3.4 Topologia D.....	71
	A-3.5 Topologia E.....	72
	A-3.6 Topologia F	73
	Bibliografia	74

Lista de Abreviaturas

AOI	And-or inverter
ASIC	Application Specific Integrated Circuit.
C	Linguagem de programação C.
CAD	Computer Aided Design.
CI	Circuito Integrado
CMOS	Complementary Metal-Oxide Semiconductor technology.
CPGCC	Curso de Pós-Graduação em Ciência da Computação.
EDA	Electronic Design Autoamtion
FPGA	Field-Programmable Gate Arrays.
GNU CC	GNU C Compiler. Produto da Free Software Foundation.
LSI	Large Scale Integration
MSI	Medium Scale Integration
PLD	Programable Logic Device
SSI	Small Scale Integration
UFRGS	Universidade Federal do Rio Grande do Sul.
VHDL	Very high speed integrated circuits Hardware Description Language.
VLSI	Very Large Scale Integration.

Lista de Figuras

FIGURA 2.1 - Diagrama Y	15
FIGURA 2.2 - Fluxo de Projeto Simplificado	16
FIGURA 2.3 - Classificação para os geradores de módulos	18
FIGURA 3.1 - Estrutura de bandas TRANCA.....	22
FIGURA 3.2 - Histórico do projeto TRANCA.....	24
FIGURA 3.3 - Leiaute de uma célula nand de 3 entradas (TRAMO).....	25
FIGURA 3.4 - Exemplo de leiaute TRAGO.....	26
FIGURA 3.5 - Leiaute de um circuito gerado pelo TROPIC.....	27
FIGURA 3.6 - Leiaute de uma Unidade Básica MARCELA.	28
FIGURA 4.1 - Estrutura do Sistema TRAMO2.....	29
FIGURA 4.2 - Estrutura do ambiente de roteamento MARTE.	31
FIGURA 4.3 - Célula nand3 (TRAMO2)	32
FIGURA 4.4 - Leiaute do circuito ALULS181 em <i>Standard Cell</i> (Solo).	33
FIGURA 4.5 - Leiaute do circuito ALULS181 gerado com o TROPIC (Tropic1).....	34
FIGURA 4.6 - Leiaute do circuito ALULS181 gerado com o TROPIC (Tropic2).....	35
FIGURA 4.7 - Leiaute do circuito ALULS181 gerado com o TRAMO2 (Alu80).....	36
FIGURA 4.8 - Estrutura do Sistema TRAMO3.....	37
FIGURA 5.1 - Exemplo de célula TRAMO	41
FIGURA 5.2 - Representação de transparência não efetiva e efetiva	41
FIGURA 5.3 - Esquemas de prioridades para alocação de trilhas.	42
FIGURA 6.1 - Seqüência de passos utilizada na geração simbólica	44
FIGURA 7.1 Circuito lógico.....	51
FIGURA 7.2 - Esquemático e leiaute da célula	51
FIGURA 7.3 - Circuito lógico do somador de 1 bit.....	52
FIGURA 7.4 - Esquemático do circuito somador de 1 bit.....	53
FIGURA 7.5 - Leiautes das células complexas do somador de 1 bit.....	53

Lista de Tabelas

TABELA	2.1 - Número de células possíveis em função da altura da célula.....	20
TABELA	4.1 Dados sobre o circuito e resultados da simulação.....	33

Resumo

Este trabalho apresenta o desenvolvimento de uma nova ferramenta para a síntese automática de células, a partir de uma descrição estrutural no nível lógico. A ferramenta está sendo integrada ao sistema TRAMO3, e visa eliminar a necessidade do uso de biblioteca de células na geração de circuitos.

Uma revisão sobre síntese de leiaute e metodologias de projeto é apresentada. A metodologia TRANCA é descrita de forma sucinta e os sistemas TRAMO2 e TRAMO3, assim como o roteador MARTE são analisados em detalhe para indicar o contexto onde se insere o trabalho.

As principais alternativas para a geração de células são analisadas e o algoritmo descrito em [REI 93b] é utilizado com algumas modificações, levando em conta situações práticas. Os seguintes passos são executados durante a processo: posicionamento dos transistores, roteamento das conexões internas e compactação do leiaute.

Finalmente, alguns melhoramentos no gerador são propostos, de forma a eliminar algumas restrições impostas na primeira versão.

Palavras chave: Geração de Células, Síntese Física, Sistema TRAMO2, Sistema TRAMO3, Projeto TRANCA, Geração de Módulos, Projeto de Circuitos Integrados, Circuitos CMOS.

Title: “Automatic Synthesis of CMOS Cells”

Abstract

This work presents the development of a new tool for automatic cell synthesis, starting from a structural description at the logic level. The tool is currently being integrated to TRAMO3 system, and aims at eliminating the need of cell libraries utilization during the circuit generation.

A brief review about layout synthesis and design methodologies is presented. TRANCA design approach is briefly described and the TRAMO2 and TRAMO3 systems, as well as the MARTE router are analyzed with some detail in order to show the environment where the work is inserted.

The main alternatives for cell generation are analyzed and the algorithm described in [REI 93b] is used with some changes, taking into account practical situations. The following steps are executed during the process: transistor placement, routing of internal connections and layout compaction.

Finally, some improvements to the generator are proposed, in order to remove some restrictions imposed in the first version.

Keywords: Cell Generation, Physical Synthesis, TRAMO2 System, TRAMO3 System, TRANCA Project, Module Generation, Integrated Circuits Design, CMOS Circuits.

1 Introdução e motivação

1.1 Introdução

A tendência da microeletrônica nos dias atuais é o uso intensivo de ferramentas de EDA em todas as fases de projeto de um circuito VLSI. O rápido desenvolvimento da tecnologia de integração implica a automação de vários passos envolvidos no projeto e fabricação dos circuitos VLSI. Além da complexidade dos sistemas vir apresentando um crescimento acentuado, a redução nas dimensões dos componentes no silício introduz compromissos cada vez mais difíceis de serem avaliados pelo projetista, o que torna o projeto automatizado, não mais uma opção, mas uma exigência.

Este trabalho se insere no contexto do projeto TRANCA [REI 87] [REI 88] e, mais especificamente, no desenvolvimento de um sistema de síntese automática de leiaute, denominado TRAMO3. Conforme a descrição anterior, o objeto de pesquisa é a síntese física (etapa onde são definidas as máscaras do circuito a ser fabricado). Embora essa opção contrarie outra tendência atual, que é a pesquisa em níveis de projeto mais abstratos, que por ser um campo relativamente recente, apresenta maiores atrativos, ela se baseia na convicção de que as alternativas disponíveis hoje são apenas satisfatórias para as tecnologias atuais e não serão capazes de atender às exigências crescentes das novas tecnologias submicrônicas e com diversos níveis de interconexão. Seguramente, a tendência de migração para níveis mais altos de abstração é uma necessidade, permitindo tratar a complexidade crescente dos projetos. Da mesma forma, a exigência de chips CMOS de alta qualidade também está aumentando a exigência por abordagens de teste adequadas. Entretanto, a migração para níveis mais abstratos pressupõe que os níveis de projeto mais baixos (i.e. mais próximos da implementação) já estejam resolvidos de forma satisfatória.

Em [REI 93b] é proposto o sistema TRAMO2, bem como sugerida uma abordagem para a geração automática de células, que tenta explorar a metodologia TRANCA ao máximo. A presente dissertação apresenta o desenvolvimento de uma biblioteca de células para o sistema TRAMO2 e a definição e implementação do protótipo de ferramenta para a síntese automática dessas células, dando origem ao sistema TRAMO3.

1.2 Organização da dissertação

As principais estratégias para o projeto de circuitos integrados CMOS são apresentadas no capítulo 2, com especial atenção às metodologias existentes para a síntese de módulos. Os métodos são analisados considerando suas características, aplicações e limitações. Especial atenção é dada às opções de síntese automática de leiaute disponíveis na literatura.

O capítulo 3 apresenta um resumo da descrição da metodologia TRANCA (limitando-se ao indispensável para a compreensão desse texto), bem como cita e comenta os principais sistemas de síntese desenvolvidos segundo esta abordagem.

O capítulo 4 descreve os sistemas TRAMO2 e TRAMO3. Ambos baseiam-se no uso de células. Entretanto o primeiro utiliza células previamente projetadas e caracterizadas, enquanto no segundo essas células são geradas durante a síntese, de acordo com as especificações do projetista.

No capítulo 5 é apresentado o estudo que determinou o modelo das células. Este estudo foi resultado da migração dos leiautes manuais das bibliotecas de células disponíveis no projeto TRANCA, aplicando algumas restrições de forma a tornar o procedimento passível de automatização. São apresentadas algumas células da biblioteca TRAMO2, bem como a suas respectivas caracterizações.

Os algoritmos para a geração de células são apresentados no capítulo 6, onde são descritas as etapas de posicionamento de transistores, roteamento e compactação dos leiautes simbólicos das células-folha.

Finalmente, no capítulo 7, são apresentadas as conclusões do trabalho e as perspectivas de trabalhos futuros dentro desta linha de estudo e pesquisa.

2 Síntese de Leiaute

2.1 Introdução

Um circuito integrado pode ser descrito em termos de três domínios, a saber: (1) comportamental, (2) estrutural, e (3) físico. Estes domínios podem ser hierarquicamente divididos em vários níveis de abstração tais como: nível funcional ou arquitetural, nível de transferência de registradores, nível lógico e nível de circuito.. A relação entre os domínios de descrição e seus níveis de abstração pode ser melhor visualizada com o auxílio do diagrama Y, conforme a fig.2.1 [GAJ 92]. Neste diagrama, as três linhas radiais representam os três domínios de descrição. Ao longo de cada linha são indicados os tipos de objetos naquele domínio. Os círculos marcam os níveis de abstração similares.

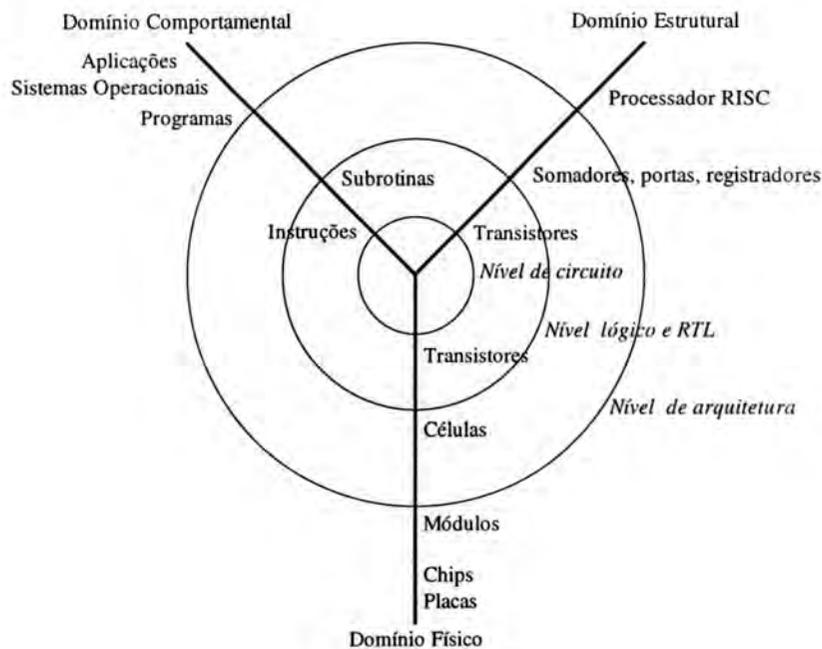


FIGURA 2.1 - Diagrama Y

2.2 Estratégias de Projeto

2.2.1 Introdução

A viabilidade econômica de um circuito integrado é afetada, em grande parte pela eficiência com que o projeto pode ser convertido de um conceito para a sua respectiva implementação. Um bom sistema de projeto VLSI deve fornecer descrições consistentes em todos os três domínios de descrição e em todos os níveis de abstração relevantes

Dada a complexidade de projetar um sistema em silício, o papel das ferramentas de EDA é reduzir a complexidade, aumentar a produtividade e assegurar ao projetista um produto confiável. Um bom método de simplificar a abordagem para um projeto é

através do uso de restrições e abstrações. Usando restrições o projetista da ferramenta torna possível a automatização de procedimentos. Usando abstrações o projetista pode desprezar detalhes e obtém um conceito mais simples, capaz de ser tratado. A visão deste processo simplificado está na fig.2.2.

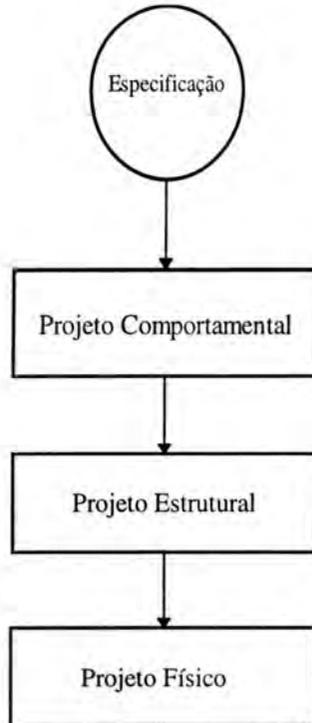


FIGURA 2.2 - Fluxo de Projeto Simplificado

2.3 Estilos de Leiaute

A etapa de síntese física pode ser realizada de diversas maneiras, de forma a melhor atender às particularidades apresentadas ou exigências relativas ao sistema ou módulo a ser implementado. Entre as exigências, pode-se citar: o desempenho elétrico, o consumo de potência e a área ocupada. Outras restrições que não podem ser ignoradas são o tempo para o mercado, a aplicação do circuito e a sua demanda. Como consequência, surgiu no decorrer do desenvolvimento da microeletrônica uma série de estilos de leiaute que diferem entre si principalmente pelas restrições que cada um impõe aos elementos do leiaute.

Existem vários critérios possíveis para se classificar os diferentes estilos de projeto de circuitos integrados. Aqui, segue-se a definição proposta em [REI 93a], que procura classificar as formas de implementação segundo o momento no qual o circuito é personalizado (diferenciado). Sendo assim, tem-se:

- CIs personalizáveis por todas as máscaras;
- CIs personalizáveis pela máscara de metalização;
- CIs personalizáveis após o encapsulamento.

Os circuitos personalizáveis por todas as máscaras têm o seu processo de fabricação diferenciado desde as primeiras etapas. Esta situação faz com que o cliente seja responsável pelo custo total do desenvolvimento, o qual deverá ser amortizado durante a produção do produto que utiliza o CI projetado. Como todas as etapas devem ser realizadas, o tempo de fabricação é mais longo do que nas demais formas de implementação. Por outro lado, há total liberdade na elaboração do leiaute, possibilitando o máximo aproveitamento possível da área de silício.

Para os ASICS personalizáveis pela metalização, ou pré-difundidos, o circuito já se encontra pré-fabricado, e a definição de suas funções dar-se-á a partir das camadas de metalização. Com isso, reduz-se significativamente o tempo de fabricação e os custos associados, pois uma mesma família de pré-difundidos pode ser utilizada para personalizar CIs de diversos clientes. Porém, nesta forma de implementação, o projetista perde flexibilidade, devido ao compromisso com as características elétricas e geométricas inerentes à etapa já realizada. Isto significa que não é possível atingir o máximo desempenho elétrico e máximo aproveitamento da área de silício.

Já para os ASICS personalizáveis após o encapsulamento, ou PLDs, a implementação é realizada estando o componente já fabricado, em condições de comercialização, bastando apenas a sua programação. Conseqüentemente, o tempo de prototipação é reduzido ao projeto lógico, uma vez que todo o leiaute já se encontra definido. Desta forma, há uma dependência muito grande entre as especificações de desempenho e a arquitetura da família de PLDs escolhida.

2.3.1 CIs personalizáveis por todas as máscaras

Dentre as metodologias de projeto de ASICS utilizando personalização através de todas as máscaras, se destacam as abordagens vistas na fig.2.3. A seguir, serão descritas as principais características de cada uma destas abordagens, bem como os critérios adotados na classificação.

Em primeiro lugar, a divisão entre lógica regular e lógica aleatória é feita para explicitar as limitações impostas pelo tipo de gerador de módulo adotado: os geradores de lógica aleatória dão ao usuário um grau maior de liberdade no que tange ao tipo (quanto à função exercida) do circuito que pode ser implementado pelo gerador em questão.

Estes diferentes graus de flexibilidade para a geração de circuitos com diferentes funções se refletem diretamente sobre as linguagens de entrada para estas ferramentas. Estas linguagens devem ser capazes de descrever lógica aleatória (ampla) ou lógica regular (restrita), conforme o caso.

2.3.2 Geração de módulos em lógica aleatória

A geração de módulos em lógica aleatória é um tipo de geração na qual a função comportamental do circuito não é definida pela escolha do gerador de módulos, ao contrário do que ocorre quando se trata dos geradores de lógica regular. Esta característica introduz o seguinte aspecto: como não existe uma estrutura pré-definida, a linguagem de entrada dos geradores em lógica aleatória é geralmente estrutural. Logo, é

necessário descrever as interconexões a serem realizadas entre as células-folha do circuito a ser gerado. Para descrever estas interconexões, a entrada do gerador é normalmente um *netlist* contendo as redes que devem ser conectadas. Este *netlist* deve chegar ao nível das células-folha e portanto pode chegar a descrever interconexões entre transistores, nos casos em que haja geração automática de células-folha.

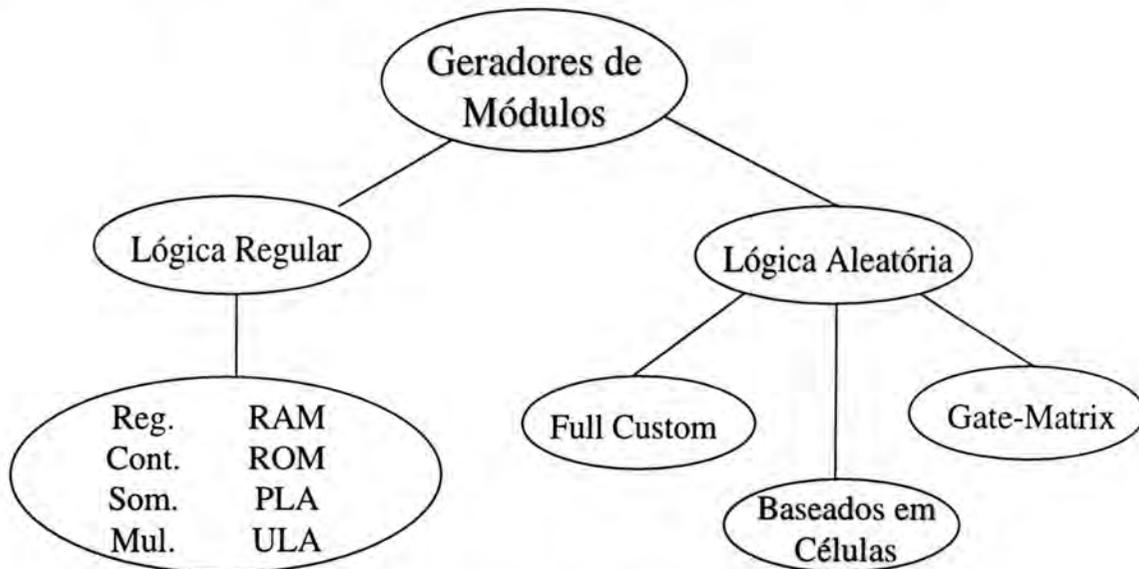


FIGURA 2.3 - Classificação para os geradores de módulos

Na fig.2.3 é também mostrada a divisão feita para a geração de módulos em lógica aleatória: *full-custom*, *gate-matrix* e baseados em células. Estes enfoques serão descritos a seguir.

2.3.2.1 Leiaute *full-custom*:

A metodologia *full-custom* tradicional refere-se ao projeto no qual todas as camadas do circuito são otimizadas. Tradicionalmente o projetista produzia a descrição física do circuito utilizando um editor de leiaute. Não existe qualquer restrição ao posicionamento dos componentes do circuito, bem como de suas interconexões. Como resultado, um especialista pode obter um alto grau de otimização em área, desempenho, ou consumo de potência. A principal desvantagem é o elevado tempo necessário à implementação, quando comparado às outras alternativas.

Um editor de leiaute que apresente facilidades de verificação de regras de projeto (DRC), e de realização de interconexões pode aumentar consideravelmente a produtividade do projetista. O uso de editores simbólicos, que não consideram as dimensões dos componentes, também pode ser útil, embora exija uma etapa posterior de compactação que não é adequada a este estilo de projeto.

Ainda nessa categoria, deve-se incluir a metodologia *full-custom* automática, que compreende metodologias que permitem a obtenção de um leiaute dedicado ao projeto, de forma automática. Em função da crescente automação do processo de síntese de leiaute, esta forma de geração tende a crescer em importância.

2.3.2.2 Gate-Matrix

Os geradores de leiaute do tipo *gate-matrix* têm como característica principal, a disposição dos transistores em uma grade, sempre na posição horizontal. Existem duas tendências bastante difundidas em leiaute *gate-matrix* [UEH 81] e [LOP 80]. A estratégia *gate-matrix* é empregada principalmente na geração de células-folha e geração de células, que é a finalidade mais usual dos geradores *gate-matrix*.

2.3.2.3 Geradores baseados em células

Os geradores baseados em células utilizam uma biblioteca de células (circuitos já desenhados, caracterizados e validados) para produzir circuitos mais complexos. A geração do leiaute fica então restrita ao particionamento, posicionamento e roteamento de células provenientes da biblioteca de células.

Os critérios adotados neste trabalho para classificação dos geradores baseados em células são o uso de canal de roteamento, o uso de *feedthrus* e a granularidade das células-folha. Quanto ao canal de roteamento na direção vertical, pode-se fazer uma divisão grosseira entre geradores de células padrão e geradores para roteamento sobre células. A classificação quanto ao uso de *feedthrus* diz respeito à passagem de um sinal sobre uma célula na direção vertical e é freqüentemente assinalado como uma dupla de pinos (que não contata o interior da célula) localizada nos extremos superior e inferior da célula, respectivamente.

Para o caso em que a célula admite a passagem de um nível de interconexão em qualquer ponto da célula, pode-se dizer que é uma célula transparente [REI 88] ou porosa [LEE 92]. A classificação quanto a granularidade das células-folha avalia o tamanho destas células. Caso a complexidade das células seja LSI ou MSI, tem-se uma ferramenta que trabalha com macroblocos. No caso de células menos complexas (SSI), a altura das mesmas é padronizada para facilitar a geração do leiaute final.

2.4 Análise da síntese automática com geração de células

A síntese automática, em si, apresenta como principal característica a redução no tempo de projeto, que por sua vez simplifica a exploração de soluções alternativas. Outra característica, é a facilidade de migração de um circuito para diferentes tecnologias de processo. A síntese com geração automática de células apresenta outras vantagens tais como:

- Eliminação das bibliotecas de células e de sua manutenção;
- Possibilidade do dimensionamento de transistores, evitando o desperdício apresentado pela abordagem *Standard Cell*.
- Implementação de portas complexas (AOIs).

Quanto a questão das células complexas, dois aspectos sobressaem. Conforme a tab.2.1 [DET 87], o número de células possíveis cresce muito com altura da célula, fazendo com que uma biblioteca de células possua apenas um subconjunto muito pequeno dessas possibilidades. O termo altura nesse contexto se refere ao número

máximo de transistores em série, entre a alimentação e a saída da célula. O cálculo do número de células, conhecido como enumeração de células, é apresentado no anexo 3. Além disso, conforme mostrado em [REI 95] o uso de um mapeador tecnológico que considere o uso de AOIs pode conseguir redução da ordem de 30% no número de transistores utilizados para implementar uma mesma função lógica. Essa redução na quantidade de transistores reduz a área do circuito, seu consumo de potência e pode aumentar o seu desempenho. Se constituindo numa alternativa interessante de projeto.

TABELA 2.1 - Número de células possíveis em função da altura da célula

Altura da célula	Número de células
1	1
2	7
3	87
4	3503
6	425803
7	154793519

Existe, entretanto, uma desvantagem inerente a geração de células que é a não caracterização (prévia) destas. É necessário primeiro sintetizá-las, extrair o seu modelo, para então proceder a simulação. A tecnologia submicron reduz esta desvantagem na medida em que as conexões, desconhecidas em ambos os casos, passam a contribuir de forma mais decisiva no desempenho do circuito.

3 A Metodologia TRANCA

3.1 Definição

Os procedimentos tradicionais de automatização da síntese física de circuitos integrados, freqüentemente adotam topologias simplificadas, de modo a reduzir a complexidade dos algoritmos e heurísticas capazes de tratar os problemas ligados ao posicionamento e roteamento dos elementos que compõem o circuito. Porém, esse processo de simplificação conduz a soluções que deixam a desejar, em termos de área e desempenho, quando comparadas às obtidas nos projetos manuais (*full custom*).

A metodologia TRANCA (TRANSPARENT Cell Approach) [REI 87] foi elaborada a partir das observações feitas em [REI 83], e propõe-se a melhorar a geração automática de leiautes VLSI em lógica aleatória. Isto é feito através do conceito de roteamento sobre células, porém aproveitando, de fato, esta sobreposição ao máximo, em um esforço conjunto entre o projeto das células e o roteamento.

No modelo da metodologia TRANCA, o roteamento é feito, tanto quanto possível, sobre a área ativa dos transistores, utilizando o conceito de transparência ou porosidade. As células utilizadas são especialmente desenhadas para oferecer a característica de transparência às camadas de roteamento, que, em tecnologias atualmente disponíveis, são metal1 e metal2. Neste caso, todos os elementos estão adequadamente alinhados e direcionados, baseando-se em uma grade virtual de trilhas horizontais e verticais que serão alocadas para as conexões.

3.2 Características

As principais características utilizadas na abordagem TRANCA são as seguintes:

- Estrutura de bandas;
- Transparência efetiva;
- Maleabilidade;
- Gerenciamento de trilhas.

3.2.1 Estrutura de Bandas

A estrutura de bandas consiste na divisão do leiaute em bandas delimitadas por linhas de alimentação. As células de uma mesma banda compartilham as mesmas trilhas de alimentação, de modo que o roteamento dessas linhas é efetuado pela simples justaposição das células, conforme pode ser observado na fig.3.1.

3.2.2 Transparência efetiva

O conceito de transparência de um bloco é a capacidade do mesmo ser atravessado por um determinado nível de interconexão, sem nenhum conflito com as conexões internas ao bloco.

Entre as características importantes relacionados com esse conceito está a necessidade de se respeitar tanto uma direção preferencial para cada camada, bem como o alinhamento entre as trilhas de uma mesma camada.

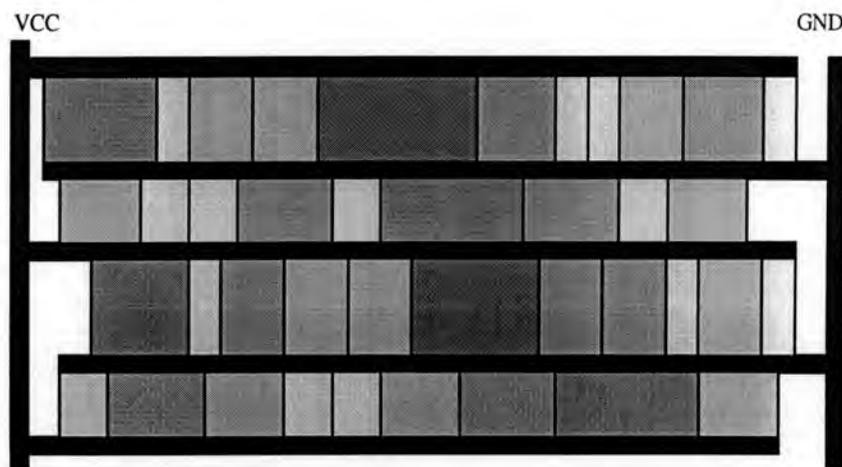


FIGURA 3.1 - Estrutura de bandas TRANCA

3.2.3 Maleabilidade

A maleabilidade consiste na capacidade de um bloco ou célula, adaptar-se a diferentes restrições de altura e largura respeitando a transparência efetiva. Num projeto que envolva vários módulos, cuja distribuição tenha sido determinada por um planejamento topológico prévio, essa característica torna-se importante, pois permite que o módulo se adapte aos espaços disponíveis.

3.2.4 Gerenciamento de trilhas

O gerenciamento de trilhas disciplina o uso de elementos sobre a grade virtual, de forma a evitar a alocação desordenada de trilhas para conexões internas às células, e produzir um alinhamento que favoreça a sua disponibilidade para conexões externas às células [REI 87].

3.3 O Projeto TRANCA

O projeto TRANCA busca desenvolver ferramentas de CAD para a síntese automática de circuitos integrados VLSI em lógica aleatória, aplicando os conceitos da metodologia TRANCA. Com a automatização do processo, procura-se desenvolver a síntese dos circuitos desde o nível de célula-folha até a sua montagem completa, provendo sistemas eficientes e independentes de tecnologia.

3.3.1 Histórico

Durante o desenvolvimento do projeto, foram desenvolvidos vários módulos que aplicam algumas características da metodologia. Inicialmente, foram desenvolvidos dois módulos de síntese básicos seguindo algumas características da metodologia proposta. O sistema TRAMO [LUB 90] procurava desenvolver a síntese utilizando células de biblioteca, mas aproveitando o conceito de transparência, enquanto que o sistema TRAGO [MOR 90] procurava efetuar a síntese de circuitos independente de tecnologia, a partir da lista de transistores.

Para solucionar as deficiências presentes nestes protótipos iniciais e aprofundar as pesquisas em diversos assuntos específicos foram propostos e desenvolvidos novos sistemas.

Encontram-se atualmente em desenvolvimento o sistema TRAMO2 [REI 93b], que utiliza uma biblioteca de células, o sistema TRAMO3, que inclui a síntese automática de células e o sistema MARCELA [GÜN 93], que utiliza a metodologia TRANCA para a geração de circuitos pré-difundidos. O sistema TROPIC [MOR 93][MOR 94], derivado do TRAGO, foi desenvolvido como tese de doutorado no LIRMM, como parte do Projeto CAPES/COFECUB. Um resumo destes sistemas e de suas principais características é apresentado no fluxograma da fig.3.2.

Os sistemas TRAMO2, TRAMO3 e MARCELA utilizam a metodologia TRANCA baseando-se em uma grade ortogonal de dois níveis de metal, sobre a qual o circuito é montado e alinhado, de forma que o roteamento deve ser executado segundo esta grade. O ambiente MARTE [JOH 93][JOH 94], foi desenvolvido para realizar as conexões de acordo com esta grade sobre as células transparentes usadas na metodologia TRANCA.

3.3.2 O Módulo TRAMO

O módulo TRAMO (TRANca Module generator) [LUB 90] realiza a geração automática de blocos em lógica aleatória a partir de células previamente desenhadas segundo a metodologia TRANCA. As células podem ser selecionadas entre as primitivas disponíveis nas bibliotecas ou projetadas conforme a necessidade.

O TRAMO é composto basicamente por dois subsistemas:

- POTRANCA, que realiza o particionamento do circuito e o assinalamento das células nas bandas e
- RETRANCA, responsável pelo posicionamento final (absoluto) e roteamento.

Há ainda um módulo exibidor (EXTRAMO), o qual permite ao usuário visualizar os resultados parciais do processo de síntese, antes da geração das máscaras propriamente dita. Essa facilidade permite a realização de sucessivos particionamentos e posicionamentos para a escolha do conjunto de parâmetros mais conveniente para a síntese definitiva das máscaras.

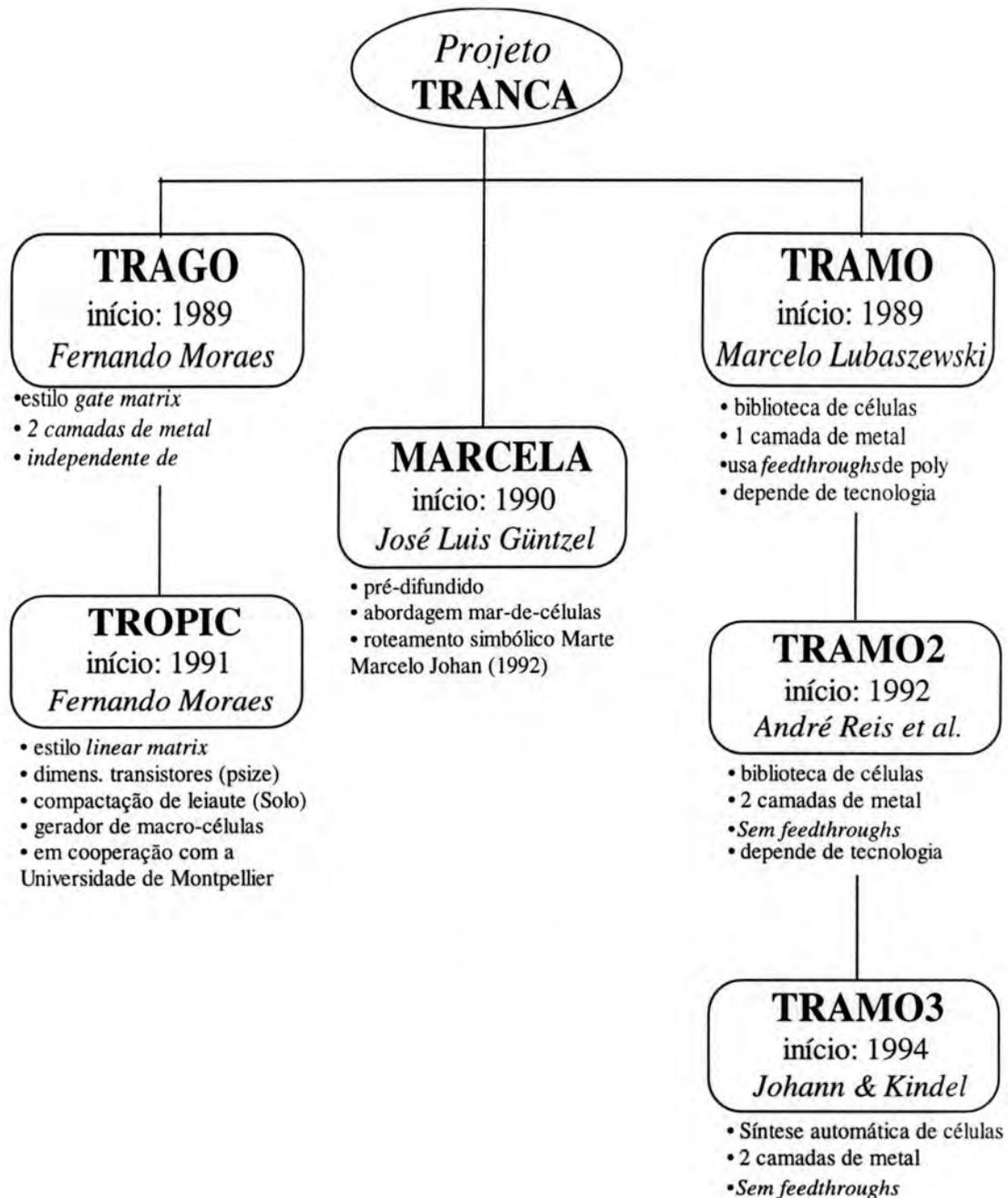


FIGURA 3.2 - Histórico do projeto TRANCA

3.3.2.1 Topologia das células

O desenho dos leiautes das células segue a estrutura de bandas e faz uso do esquema de prioridades na alocação das trilhas. A fig.3.3 mostra o leiaute de uma célula nand de 3 entradas disponível na biblioteca de células do sistema.

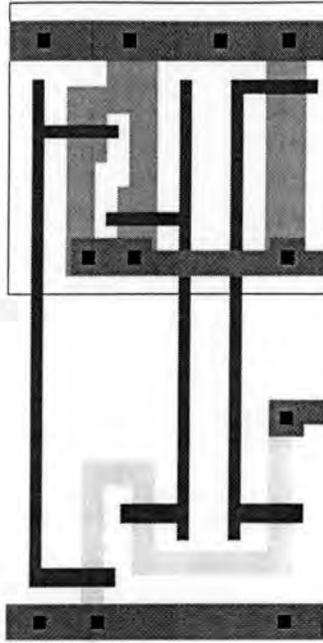


FIGURA 3.3 - Leiaute de uma célula nand de 3 entradas (TRAMO).

As principais características topológicas seguidas são:

- Entradas realizadas com linhas de polissilício na vertical, as quais são acessíveis em todas as trilhas não utilizadas para conexões locais;
- Saídas disponíveis em metal na borda direita da célula.

3.3.2.2 Características dos leiautes produzidos pelo TRAMO

A principal restrição aos leiautes gerado pelo TRAMO está relacionada ao fato do sistema original considerar apenas uma camada de metal para a realização das interconexões. Dessa forma, ligações entre células localizadas em bandas não adjacentes necessitam a inserção de *feedthrus* realizados em polissilício, o que limita o desempenho do circuito gerado.

3.3.3 O módulo TRAGO

O TRAGO (Tranca Automatic GeneratOr) [MOR 90] é um sistema que realiza a síntese baseado na geração automática de células.

As células geradas pelo sistema têm sua topologia baseada na estratégia *gate-matrix*, porém, apresentam somente um par de transistores por coluna. Cada célula é representada por duas tiras horizontais de difusão, as quais podem ser posicionadas sob as trilhas de roteamento de diferentes modos, conforme for mais conveniente. A fig.3.4 mostra um leiaute de um circuito composto por algumas células gerado pelo TRAGO.

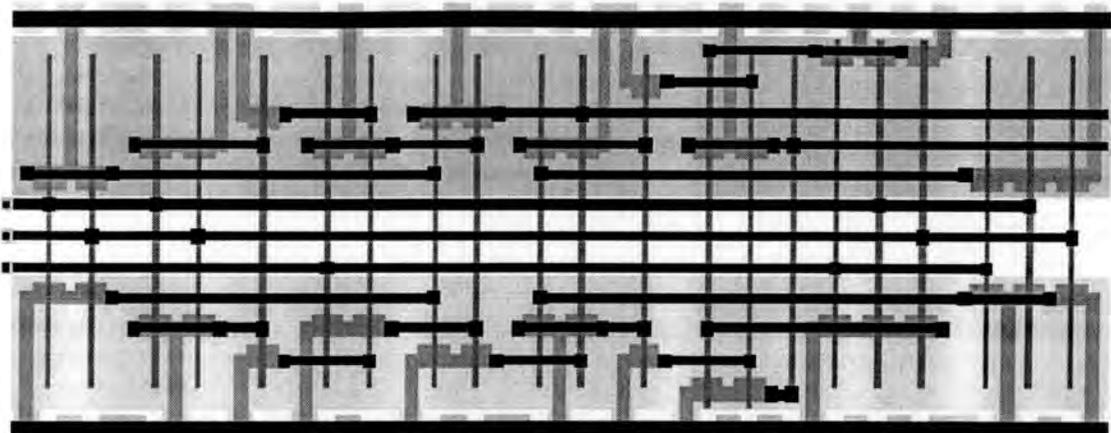


FIGURA 3.4 - Exemplo de leiaute TRAGO

A entrada do TRAGO é um *netlist* Spice hierárquico com a descrição das células e das redes associadas.

Há uma etapa de pré-processamento onde a descrição Spice é planificada e, a seguir, analisada, de forma a agrupar transistores em células básicas, que serão sintetizadas. As células básicas são formadas por conjuntos de transistores conectados em série e/ou em paralelo entre a alimentação e a saída, tais como portas *nand*, *nor* e inversor.

Após a definição das células, são ordenados os pares de transistores complementares com a mesma porta, de modo que estes compartilhem a mesma coluna de polisilício. Este procedimento é denominado pareamento [MOR 90].

Antes de iniciar o particionamento, é realizada uma pré-geração das células básicas, para informar ao particionador o número total de células do circuito, e para cada célula, o número e posição de entradas e saídas e sua largura. O particionamento segue o mesmo algoritmo do TRAMO.

Após o particionamento, é feita a geração de bandas, seguindo três etapas. Esta geração é feita uma vez para cada banda e começa pelas bandas que apresentarem um maior número de conexões com outras bandas. Na primeira etapa, as células são posicionadas internamente à banda de modo a maximizar as conexões por justaposição. Este posicionamento é realizado diversas vezes para cada banda, de acordo com um parâmetro de repetição fornecido pelo usuário. A segunda etapa consiste de um roteamento do tipo *left-edge* [JOH 94] para cada posicionamento, sendo escolhido aquele que resultar em uma banda com o menor número de trilhas. A terceira etapa consiste na alocação das trilhas de metal sob as quais serão desenhados os transistores.

3.3.4 O Sistema TROPIC

O sistema TROPIC, [MOR 93] e [MOR 94], é derivado do sistema TRAGO e foi desenvolvido dentro de um projeto internacional em cooperação com a Universidade de Montpellier, França. A topologia de leiaute utilizada é *linear-matrix* [UEH 81], tendo sido demonstrado que este estilo minimiza capacitâncias parasitas. O TROPIC é pensado como um gerador de macro-células e utiliza dimensionamento de transistores

através de uma ferramenta desenvolvida no laboratório LIRMM de Montpellier, o **Psize** [AZE 92]. Além desta ferramenta, o TROPIC pode utilizar o compactador do SOLO, ferramenta comercial da CADENCE, como o compactador SYLC [SOT 94], desenvolvido na UFRGS, para a tradução do leiaute simbólico para a descrição detalhada das máscaras.

A fig.3.5 mostra o leiaute de um pequeno circuito gerado pelo sistema TROPIC. Embora seja uma ferramenta de bom desempenho, pode-se observar que somente utiliza transparência na segunda camada de metal, para roteamento vertical. A posição horizontal dos transistores impede que se utilize a possibilidade de transparência na camada de metal 1, mesmo quando utilizados transistores grandes ou ligados em série. Este fato também contribui para a formação de canais de roteamento internos às bandas, como no sistema TRAGO.



FIGURA 3.5 - Leiaute de um circuito gerado pelo TROPIC.

3.3.5 O Sistema MARCELA

O sistema MARCELA [GÜN 93] é uma opção do projeto TRANCA que visa a prototipação rápida de Circuitos Integrados para Aplicações Específicas (ASICs) através do uso de matrizes pré-difundidas. Na abordagem MARCELA a matriz é composta por células pré-difundidas ao invés de simples fileiras de transistores como em *sea-of-gates* ou outros *gate-arrays* [REA 85].

As células utilizadas seguem a mesma filosofia TRANCA, e foram escolhidas em uma determinada proporção, conforme estudos práticos, para se adaptarem à implementação de diferentes tipos de circuitos. Um pequeno conjunto de células (1 nand, 1 nor, 2 inversores e 2 *transmission gates*) formam uma unidade básica (UB), como mostra a fig.3.6, a qual é repetida para formar uma matriz (ex: mar1000). A implementação de um circuito requer basicamente 3 etapas de projeto, a partir da escolha de uma matriz: decomposição lógica do circuito nas primitivas disponíveis na matriz; assinalamento, dizendo quais células vão implementar quais funções; e roteamento, através do qual as conexões são realizadas com as camadas de metal sobre a área ativa do circuito.

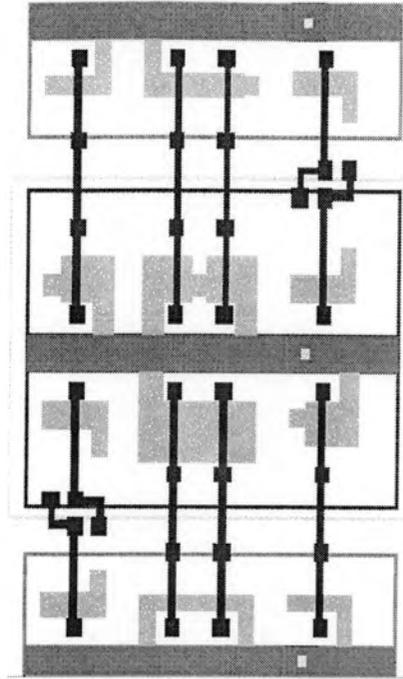


FIGURA 3.6 - Leiaute de uma Unidade Básica MARCELA.

4 Os Sistemas TRAMO2 e TRAMO3

O TRAMO2 e o TRAMO3 são sistemas de síntese de leiaute que visam a geração de blocos em lógica aleatória, constituindo-se em uma alternativa à metodologia de células padrão. Nestes sistemas, os conceitos da metodologia TRANCA são explorados a fundo, de forma a alcançar uma relação entre desempenho elétrico e área ocupada superior às alternativas atualmente disponíveis.

4.1 A Estrutura do sistema TRAMO2

O TRAMO2 é baseado no uso de biblioteca de células. Assim, o mesmo segue passos tradicionais da síntese física: particionamento, posicionamento e roteamento. Entretanto, conforme pode-se observar na fig.3.1, ele apresenta algumas características originais que o diferenciam de outros sistemas conhecidos, conforme será mostrado a seguir.



FIGURA 4.1 - Estrutura do Sistema TRAMO2

Inicialmente o circuito é decomposto em termos das células disponíveis na biblioteca, resultando numa descrição chamada de expandida. A estrutura do circuito (interconexões entre as células) é utilizada na etapa de posicionamento, juntamente com as informações dos tamanhos das células.

A partir dos leiautes das células, é obtida uma abstração simbólica exigida pela etapa de roteamento. Esta abstração inclui a matriz de restrição e a descrição dos terminais para cada célula. Ao final do roteamento detalhado, o leiaute das interconexões é sobreposto ao leiaute das células previamente posicionadas, de forma a completar a descrição das máscaras do circuito.

4.2 O roteador MARTE

O roteamento dos circuitos TRAMO2 e TRAMO3 é realizado com o uso do ambiente MARTE (Maze RouTing Environment) [JOH 94], o qual foi desenvolvido para ser utilizado no sistema TRANCA, sendo utilizado também pelo módulo MARCELA. Entretanto, o MARTE é um roteador de uso genérico, que pode ser utilizado em qualquer leiaute, desde que este seja representado por uma matriz de restrições. Suas principais características são a realimentação e a controlabilidade, bem como a previsão para diversas opções de algoritmos. Isto é possível graças à compatibilidade sintática e semântica dos arquivos de entrada e saída do roteamento simbólico. A estrutura do ambiente, com os principais módulos e os arquivos internos é mostrada na fig.4.2.

A realização do roteamento sobre células, como prevê a metodologia TRANCA, determina que um número muito grande de restrições tenham que ser levadas em consideração. Diferentemente dos leiautes TRAMO e TRAGO, onde os roteamentos vertical e horizontal são tratados em separado, os leiautes MARCELA, TRAMO2 e TRAMO3 oferecem maior liberdade, tratando de forma idêntica todas as direções.

Por esses motivos, o MARTE usa o algoritmo *maze* básico com algumas modificações [PRE 88]. Os roteadores *maze* caracterizam-se por trabalharem sobre uma grade, e sempre encontrarem o caminho mais curto para a interconexão de dois pontos. A área a ser roteada é dividida em quadrados, denominados células, e a cada um é atribuído um código que permite ou não a passagem ou acesso de conexões. O algoritmo básico, ou de Lee [PRE 88], consiste em marcar as células não bloqueadas a partir do ponto de origem até encontrar o destino, ou cobrir toda a área. A marcação é feita com um número de uma seqüência, permitindo que, uma vez encontrado o destino, o algoritmo possa identificar o caminho mais curto até a origem. Feito isto, as células não alocadas para o caminho da conexão são desmarcadas, enquanto que as utilizadas acrescentam restrições sobre a área por elas ocupada.

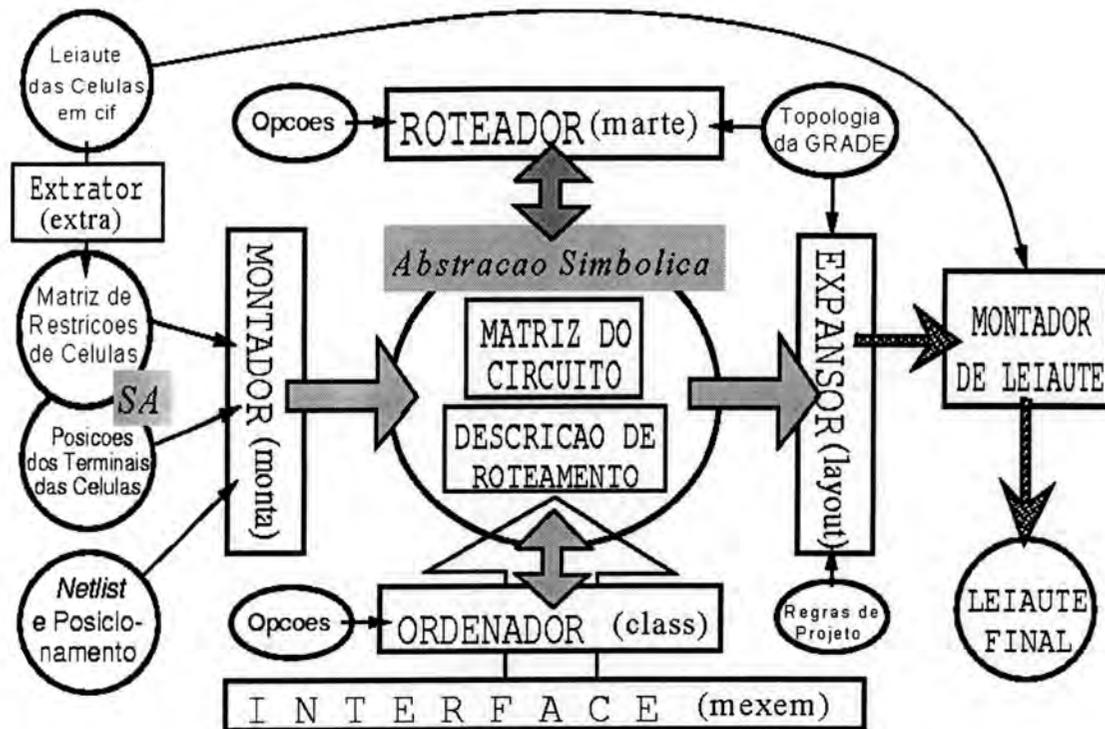


FIGURA 4.2 - Estrutura do ambiente de roteamento MARTE.

No caso das abordagens MARCELA, TRAMO2 e TRAMO3, o roteamento deve ser feito segundo uma grade de metal1 na horizontal e metal 2 na vertical, sendo que entradas e saídas podem ser acessadas em mais de um ponto. Trocas de trilhas só podem ser feitas nas interseções da grade, caso não haja restrição para via. Como o roteamento é feito sobre o leiaute já fixo das células, foi definida uma matriz de restrições, cuja função é armazenar as restrições para cada ponto da grade. Basicamente, há três tipos de restrições: para metal1, para metal2 e para via.

A matriz de restrições de um circuito é montada a partir das matrizes de restrições das células que o compõem. Durante o roteamento, as restrições resultantes das novas conexões vão sendo acrescentadas àquelas já existentes. A matriz, portanto, deve estar presente em memória, sendo gravada posteriormente no arquivo final. Isto implica numa grande área de memória ou *swapping* que o computador deve ter para processar circuitos grandes. A matriz resultante de um roteamento deve permanecer porque o circuito pode ser pós-processado.

Um problema grave dos roteadores *maze* é o bloqueio de entradas por outras conexões. Para evitar isso, o MARTE prevê um mecanismo de reserva de terminais de células.

Além do roteador propriamente dito, fazem parte do ambiente uma interface de edição e visualização, o montador da matriz de restrições e um expensor do roteamento simbólico para leiaute de máscaras.

A utilização do roteador exige que todas as células a serem utilizadas tenham suas matrizes de restrições definidas. Além das restrições, os terminais de entrada e saída também devem ser assinalados.

4.3 Implementação preliminar

Com o objetivo de avaliar o resultado produzido pelo sistema TRAMO2 de forma a comparar com outras implementações, foi gerado de forma semi-automática um circuito de teste (unidade lógica e aritmética de 4 bits). Foi utilizado o posicionador do TRAMO original [LUB 90], apenas adaptado ao problema, e um conjunto de células desenhadas manualmente, segundo as mesmas especificações da geração automática. A fig.4.3 mostra o leiaute de uma destas células.



FIGURA 4.3 - Célula nand3 (TRAMO2)

Esse estudo preliminar foi realizado com o objetivo de determinar a validade das considerações teóricas feitas em [REI 93b], quanto às vantagens do TRAMO2, sobre outras formas de implementação de ASICs, utilizando um caso prático.

O circuito resultante foi comparado com outras implementações. Uma delas gerada utilizando o Cadence Edge 2.1 -Solo- (Standard Cell) e duas outras utilizando o TROPIC, uma com todos os transistores de tamanho 8 micra, chamada de Tropic1 e outra com transistores idênticos a versão gerada com o TRAMO2, chamada de Tropic2.

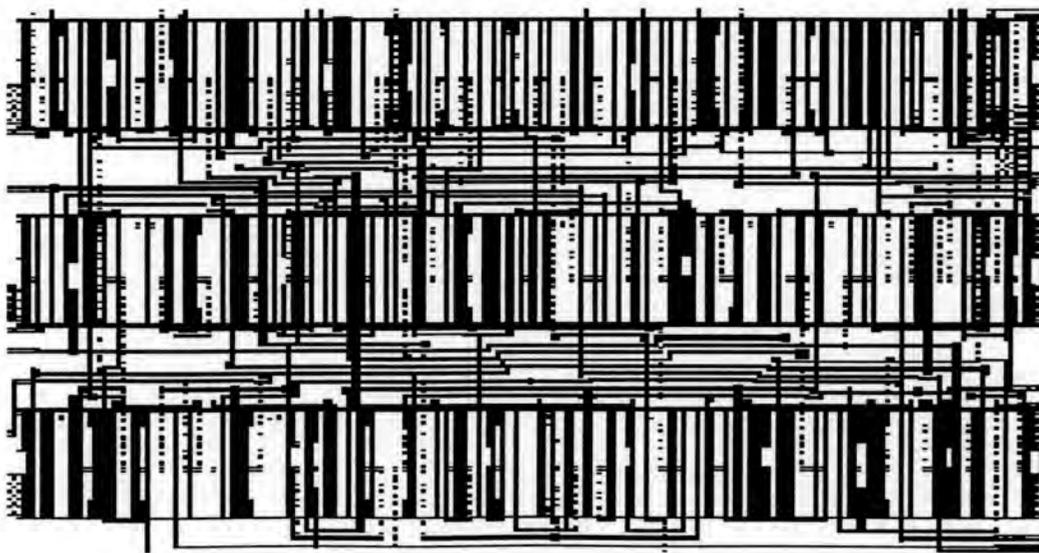
TABELA 4.1 Dados sobre o circuito e resultados da simulação.

Circuito	Área (mm ²)	W _{médio} (μm)	Xstores/mm ²	tdhl (ns)	Potência (mW)	cap. total
Solo	.1499	13.5	3028	11.5	---	---
Tropic1	.1350	8.0	3140	15.6	3.9	10.01
Tropic2	.1298	3.5	3250	21.1	3.1	9.38
TRAMO2	.1225	3.5	3444	18.8	2.8	7.72

Todos os circuitos foram simulados para para comparar o atraso no caminho crítico. As entradas foram excitadas por *buffers* de entrada e capacitância de carga foi de 350fF (estimativa empírica). A implementação em *Standard Cell* (Solo) foi simulada com o simulador funcional Silos2, considerando as capacitâncias parasitas extraídas depois do roteamento detalhado. As implementações geradas pelo TROPIC e pelo TRAMO2 foram simuladas com o simulador elétrico Hspice, considerando as capacitâncias extraídas das células e do roteamento. As figuras 4.4, 4.5, 4.6 e 4.7 apresentam os leiautes de cada uma das implementações.

De acordo com a tab.4.1 verifica-se que o circuito implementado em *Standard Cell* é mais rápido que as versões TRAMO2 e TROPIC, mas usa transistores muito maiores, resultando numa densidade de integração um pouco inferior às alcançadas pelo TRAMO2 e TROPIC. Mesmo considerando-se apenas esse exemplo, pode-se dizer que os sistemas TROPIC e TRAMO2, e generalizando, a geração de módulos, pode produzir leiautes eficientes em termos de área e desempenho, quando comparados à abordagem *Standard Cell* usando células projetadas manualmente.

Neste caso específico, verifica-se que o circuito TROPIC é mais lento que o circuito TRAMO2 com os mesmos tamanhos de transistores.

FIGURA 4.4 - Leiaute do circuito ALULS181 em *Standard Cell* (Solo).

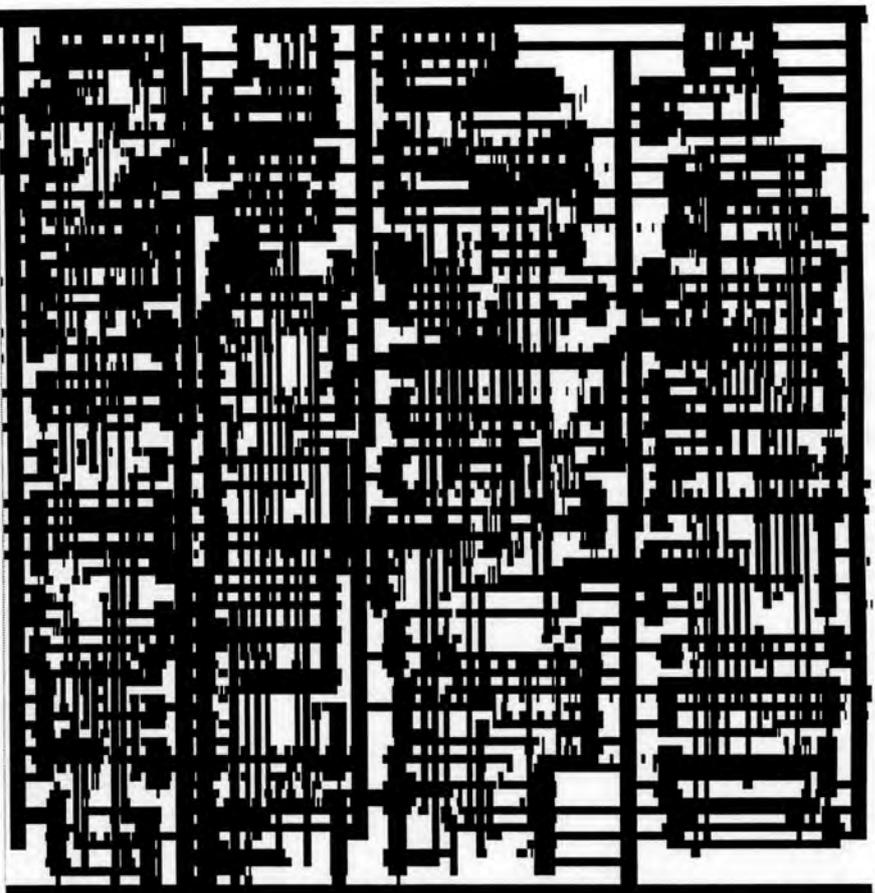


FIGURA 4.5 - Leiante do circuito ALULS181 gerado com o TROPIC (Tropic1).

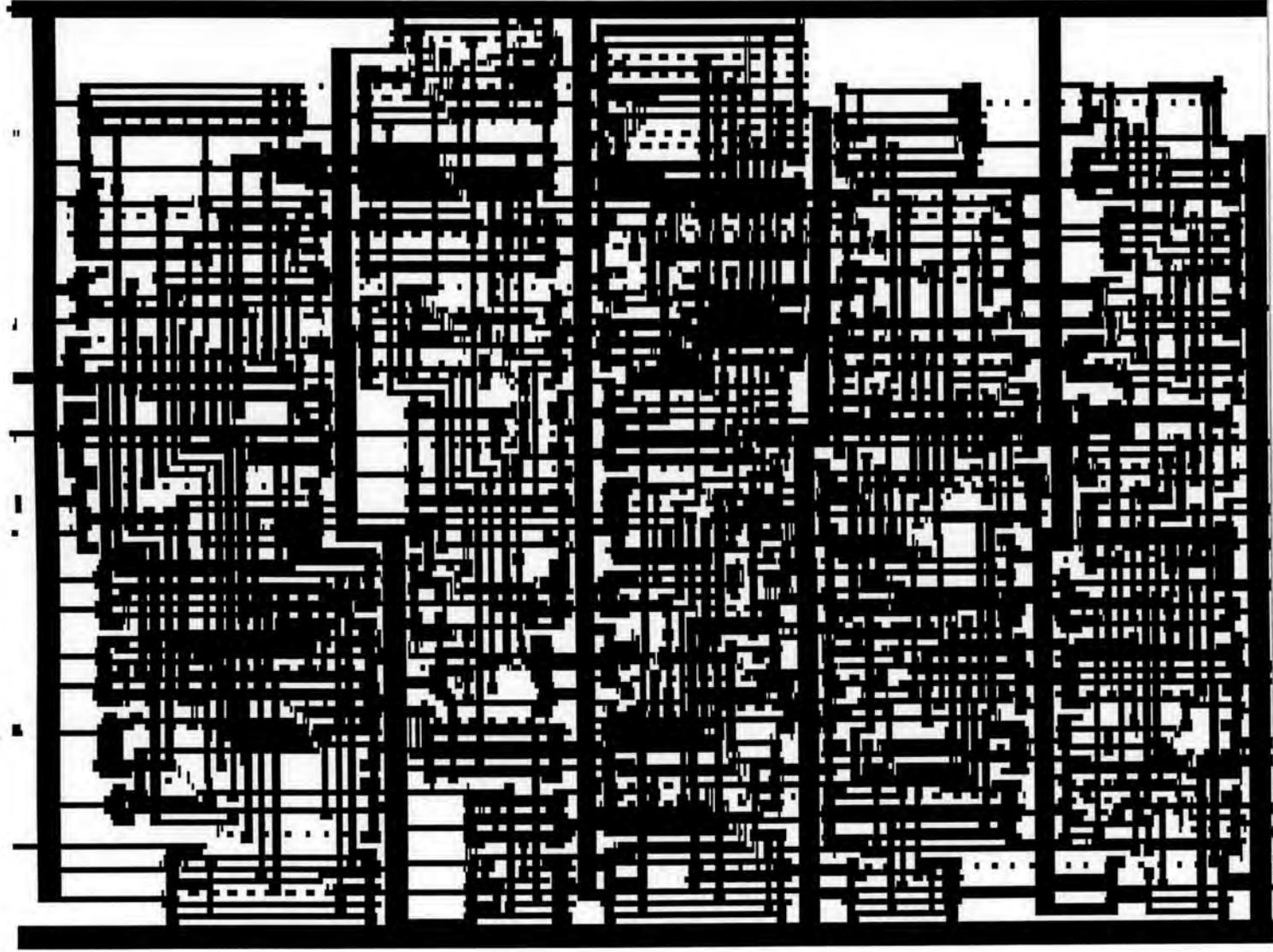


FIGURA 4.6 - Leiaute do circuito ALULS181 gerado com o TROPIC (Tropic2).

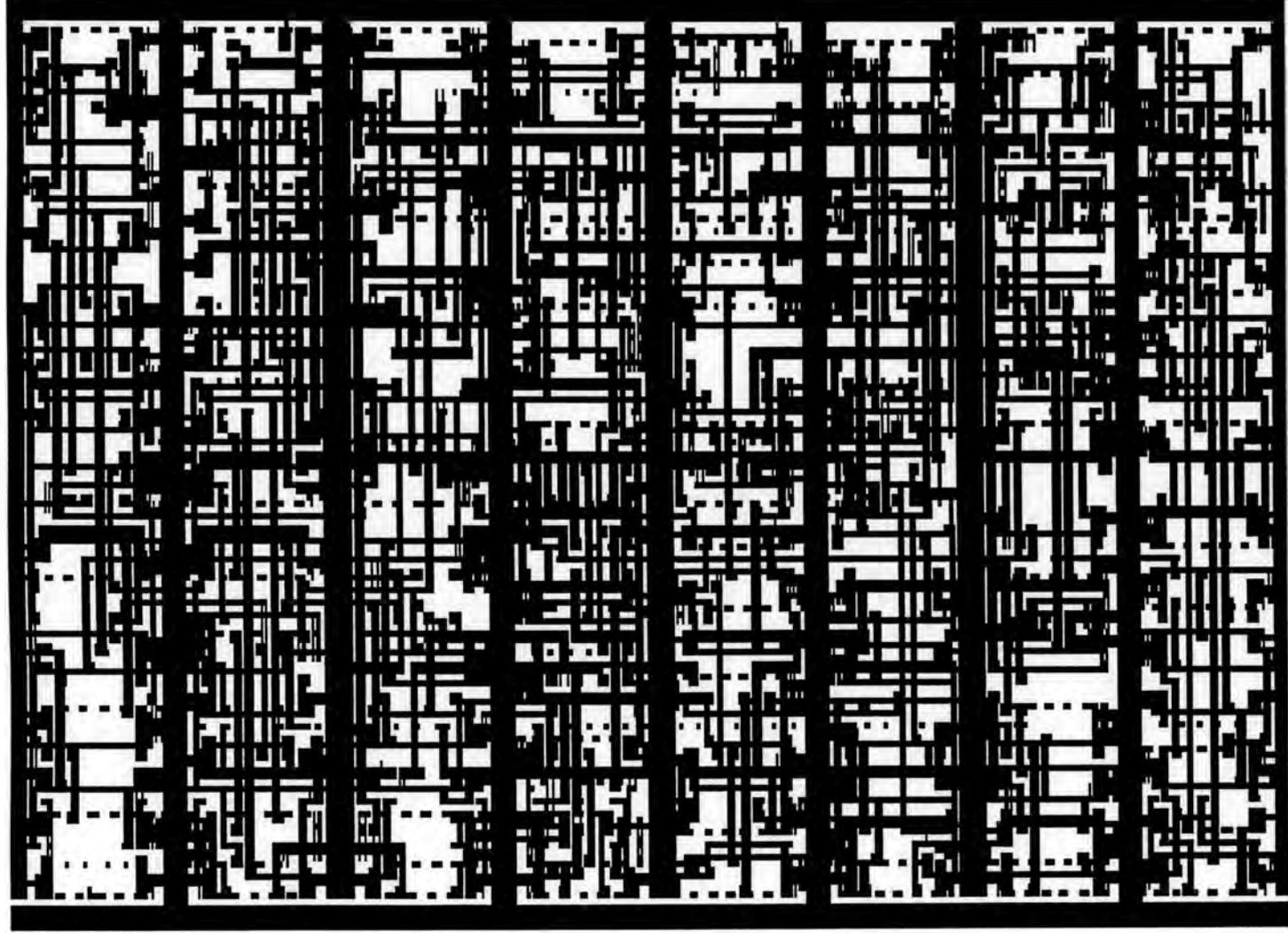


FIGURA 4.7 - Leiaute do circuito ALULS181 gerado com o TRAMO2 (Alu80).

4.4 O Sistema TRAMO3

O sistema TRAMO3 apresenta uma estrutura similar ao TRAMO2, com algumas adaptações de forma a substituir a biblioteca de células deste pela geração automática das células do circuito, conforme indicado pela região sombreada na fig.4.8.

Analogamente ao TRAMO2, o circuito é decomposto em termos de blocos básicos, dependentes das características do gerador de células disponível, resultando

numa descrição expandida, composta por células-folha. No contexto desse trabalho, o termo célula-folha refere-se a células compostas apenas por transistores, e que atendem as restrições do gerador de células utilizado. A informação estrutural de cada célula é passada para o gerador de células, ao passo que a estrutura do circuito (interconexões entre as células-folha) é utilizada na etapa de posicionamento, juntamente com alguma informação simbólica, resultante da primeira etapa da geração de células.

Após a etapa de posicionamento, as células simbólicas são expandidas, levando em conta a altura de cada banda, determinada durante o posicionamento. A expansão da representação simbólica é realizada através da compactação dos leiautes das células, considerando as restrições impostas pela metodologia TRANCA, representadas aqui pelas características do sistema de roteamento MARTE [JOH 94]. Estas restrições estão presentes nas camadas de metal e na disposição dos terminais das células.

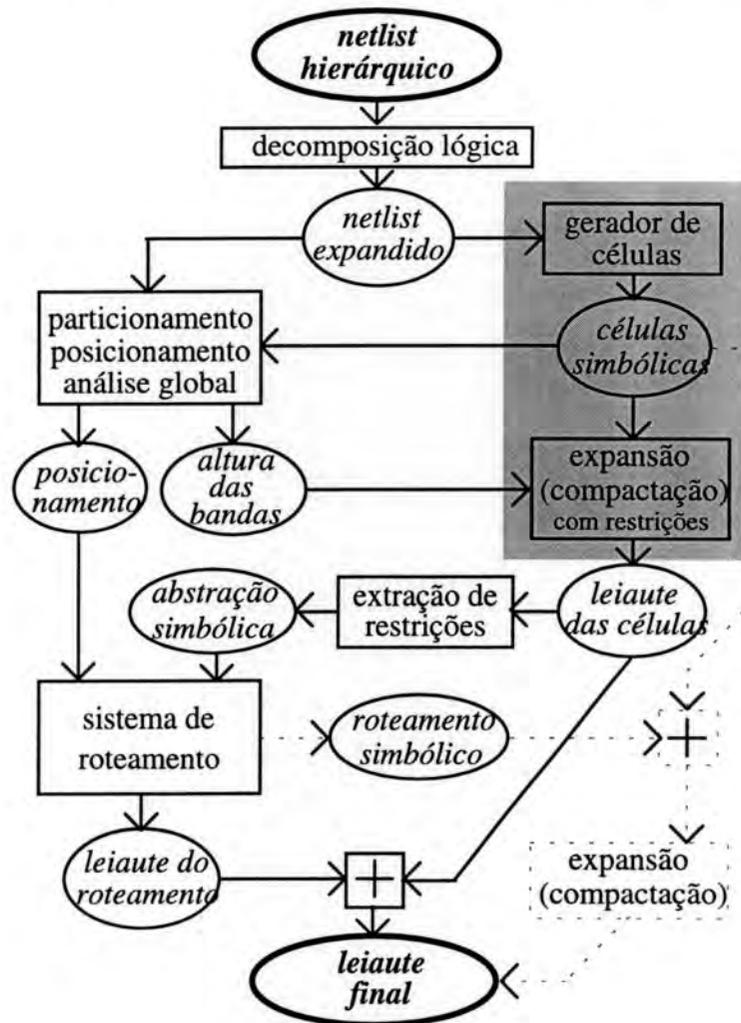


FIGURA 4.8 - Estrutura do Sistema TRAMO3

A partir dos leiautes das células, é obtida uma abstração simbólica exigida pela etapa de roteamento. Esta abstração inclui a matriz de restrição e a descrição dos terminais para cada célula-folha. Ao final do roteamento detalhado, o leiaute das

interconexões é sobreposto ao leiaute das células previamente posicionadas, de forma a completar a descrição das máscaras do circuito.

Desde que células e roteamento são simbolicamente gerados, seria também possível desconsiderar a expansão individual das células, e fazer-se uma compactação global de todo o circuito simbólico. Esta opção é indicada na figura através de linhas tracejadas. Entretanto, essa ausência de compactação global é a principal característica que distingue a síntese utilizada no TRAMO3 de outros sistemas disponíveis na literatura [MOR 90] [LIN 94] [DOM 90], onde esta característica é o principal limite à síntese de circuitos complexos. Entretanto, a opção utilizada no TRAMO3 traz como desvantagem a necessidade de uma previsão de conexões adequada, pois o roteamento é realizado sobre uma abstração do leiaute real, ou seja, não existe a possibilidade de aumentar a capacidade de interconexões no caso de congestionamento.

Além das considerações quanto ao leiaute, a geração automática de células permite o dimensionamento de transistores, levando em conta o *fanout* de cada célula bem como uma análise de caminhos críticos, permitindo um melhor desempenho elétrico.

4.5 Diferenças entre TRAMO3 e TROPIC

Os sistemas TRAMO3 e TROPIC são geradores de módulos que produzem leiaute das máscaras a partir da descrição estrutural no nível lógico de um circuito. Entretanto, apresentam entre si uma série de diferenças, tanto no que tange à topologia dos circuitos produzidos, como na seqüência do processo de síntese.

Quanto a topologia, o TROPIC aplica a abordagem conhecida como *linear matrix*, onde cada banda é composta basicamente por duas linhas de difusão (uma para os transistores p-MOS e outra para os n-MOS). Assim, os transistores são dispostos horizontalmente, compartilhando, sempre que possível, a mesma difusão. Essa é a alternativa para geração de células mais difundida atualmente. Entretanto, conforme é apresentado em [REI 93b], essa topologia não é muito adequada à metodologia TRANCA, no sentido em que não permite explorar a transparência efetiva, além de terminar por gerar uma espécie de canal de roteamento entre os ramos P e N, que pode comprometer o desempenho. O sistema TRAMO3, por outro lado, utiliza uma abordagem original para a geração das células, na qual os transistores são dispostos na vertical, numa tentativa de se aproximar ao resultado que se verifica nos projetos manuais.

Quanto ao fluxo de projeto, a principal diferença entre os sistemas é que, no TROPIC, a etapa de posicionamento e roteamento é realizada sobre uma abstração simbólica do leiaute, sendo que a compactação, que fará a tradução da descrição simbólica para o nível de máscaras, deve ser realizada sobre todo o circuito. Entretanto, a etapa da compactação costuma ser o fator determinante do tamanho máximo dos circuitos que podem ser gerados, pois quanto maior o número de transistores a serem compactados, maior o número de restrições que o compactador tem que levar em consideração. Para evitar essa restrição, o sistema TRAMO3 realiza a compactação no nível de células-folha. Dessa forma, o roteamento é realizado sobre o leiaute já determinado das células. Entretanto, esta característica traz o inconveniente de que a

síntese do leiaute das células deve considerar as exigências de roteamento. No caso do TRAMO3, o posicionador deve fornecer uma estimativa do número de trilhas que deverão ser utilizadas numa banda para conexões intercélulas. Assim, enquanto no TROPIC existe a garantia de todas as conexões serão realizadas (ainda que a custo de perda de desempenho) no TRAMO3 a realização das conexões é dependente desta estimativa inicial.

Dessa forma, pode-se considerar que os sistemas são complementares, havendo situações em que um ou outro deva produzir o melhor resultado. Do ponto de vista do projetista, essa possibilidade de escolha é especialmente interessante.

5 A Topologia das Células para o TRAMO2

Este capítulo apresenta a definição do modelo das células transparentes adequadas ao roteamento *Full Over-the-Cell* presente no sistema TRAMO2 através do sistema de roteamento MARTE [JOH 94]. Devido ao grande número de variáveis e possibilidades, algumas restrições foram impostas, de forma a permitir a automatização da síntese dessas células.

5.1 Análise prévia

Conforme [JOH 94], dois principais objetivos devem ser levados em consideração para a definição do modelo das células a ser utilizado no TRAMO2:

1. Exploração da transparência efetiva;
2. Casamento entre o projeto das células a realização do roteamento, otimizando ambos simultaneamente.

Esta otimização é essencial para o uso eficiente da metodologia TRANCA, conduzindo ao bom aproveitamento da área da célula e buscando também bom desempenho elétrico.

Entre as variáveis que podem ser manipuladas nas células estão: prioridades de alocação das trilhas internas; posicionamento dos terminais de entrada e saída, em especial quanto ao número de pontos de acesso; e posicionamento relativo dos transistores. Essas possibilidades foram avaliadas de forma a otimizar área, considerando a transparência efetiva, e otimizar o desempenho, minimizando tanto quanto possível, as áreas de dreno e fonte dos transistores.

5.2 Transparência Efetiva

O conceito de transparência efetiva, compreendido aqui como o ganho de área resultante da sobreposição de roteamento e transistores, bem como a sua importância na metodologia TRANCA é bem explicada em [REI 93b]. Nesse trabalho mostra-se que a transparência efetiva é obtida principalmente com o uso de transistores posicionados verticalmente (sentido da corrente no transistor) como resultados das trilhas e orientações utilizadas para o roteamento em nossa abordagem. Algumas variações em relação a proposta original [REI 93b] são apresentadas aqui.

Embora o conceito de transparência seja um dos pressupostos originais da metodologia TRANCA, nos primeiros sistemas desenvolvidos, a mesma era obtida às custas de desempenho elétrico, uma vez que eram utilizadas grandes áreas dreno e fonte dos transistores, como pode ser observado na fig.5.1.

Considerando as tecnologias atuais, um canal de transistor pode ser desenhado no espaço entre duas linhas justapostas de metal1. Assim, rigorosamente, a transparência efetiva existe apenas em três situações, como pode-se observar na fig.5.2:

1. Entre as regiões P e N em consequência das regras de separação entre poços, ou entre poço e implante;
2. Sobre um conjunto de transistores em série;
3. Quando são utilizados transistores grandes em forma de "S".

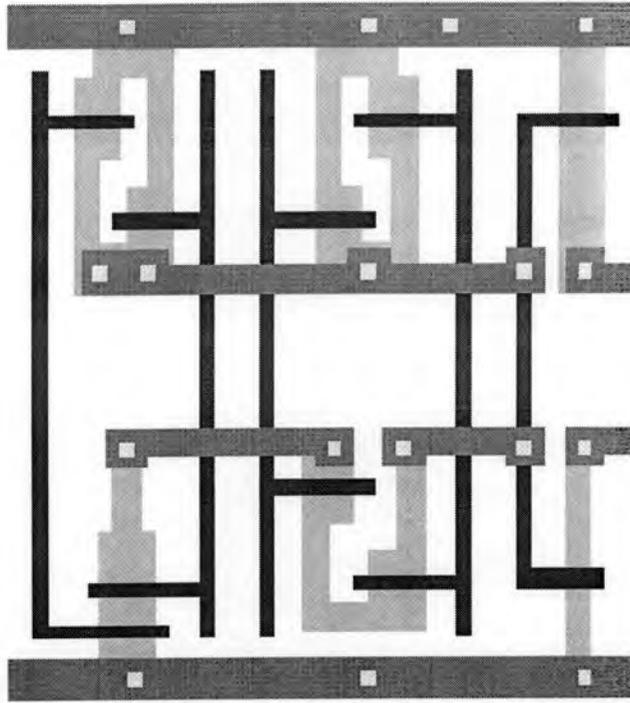


FIGURA 5.1 - Exemplo de célula TRAMO

Sendo assim, a única forma de obter transparência efetiva é através do correto gerenciamento das situações 2 e 3, uma vez que a situação 1 é determinada pela tecnologia.

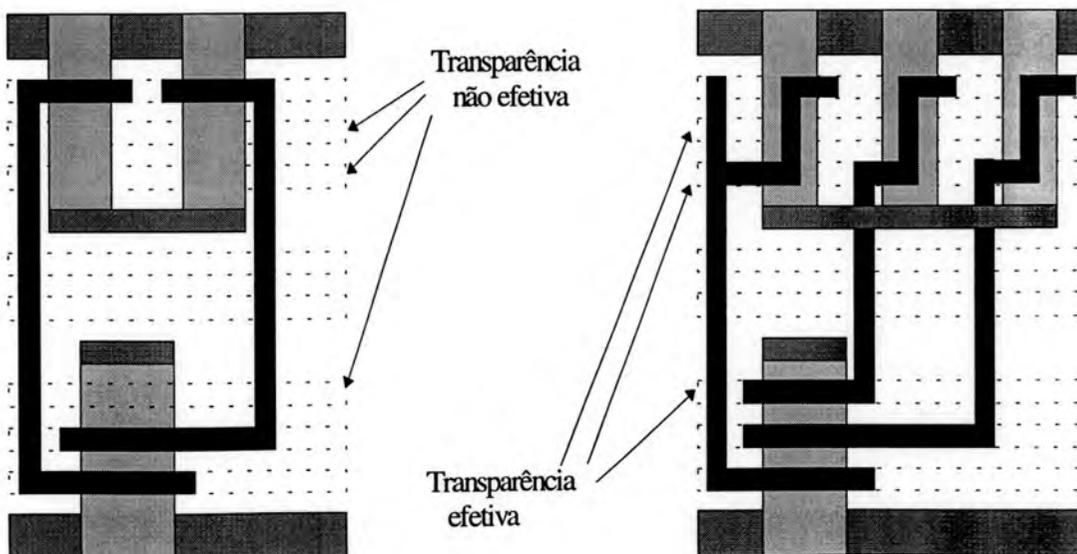


FIGURA 5.2 - Representação de transparência não efetiva e efetiva

5.3 Prioridade na alocação de trilhas

Em todas as implementações TRANCA anteriores, o gerenciamento de trilhas exigia o máximo alinhamento porque o controle da transparência em metal sobre a célula era feito somente com números que informavam quais trilhas estavam livres para conexões externas (intercélulas) e quais não.

A fig.5.3 mostra alguns esquemas de alocação de trilhas possíveis na metodologia TRANCA. A fig.5.3a apresenta um esquema com trilhas justapostas, enquanto as demais apresentam um esquema com trilhas alternadas, diferindo entre si quando a prioridade de uso das trilhas. O esquema 5.3a permite a colocação de contatos na mesma coluna, em trilhas adjacentes. Este esquema apresenta a desvantagem de consumir mais área para um mesmo número de trilhas, quando comparado com os demais esquemas.

Com o sistema MARTE, que implementa uma modelagem mais detalhada, a alocação de trilhas deve continuar sendo gerenciada, mas todo espaço disponível para conexões disponível na célula é considerado durante a etapa de roteamento.

Nos sistemas anteriores as principais trilhas utilizadas para roteamento intracélula eram as mais próximas do centro (não as duas centrais entre as regiões P e N), como mostra a fig.5.3b. Esta escolha tinha como principal consequência o aumento desnecessário e exagerado de transistores que não estavam em série (principalmente), os quais tinham um extremo na alimentação e o outro nestas trilhas centrais. Explorando um esquema de prioridades um pouco mais distribuído ou flexível, podemos reduzir estas distorções, como podemos comprovar nas células que mostraremos a seguir.

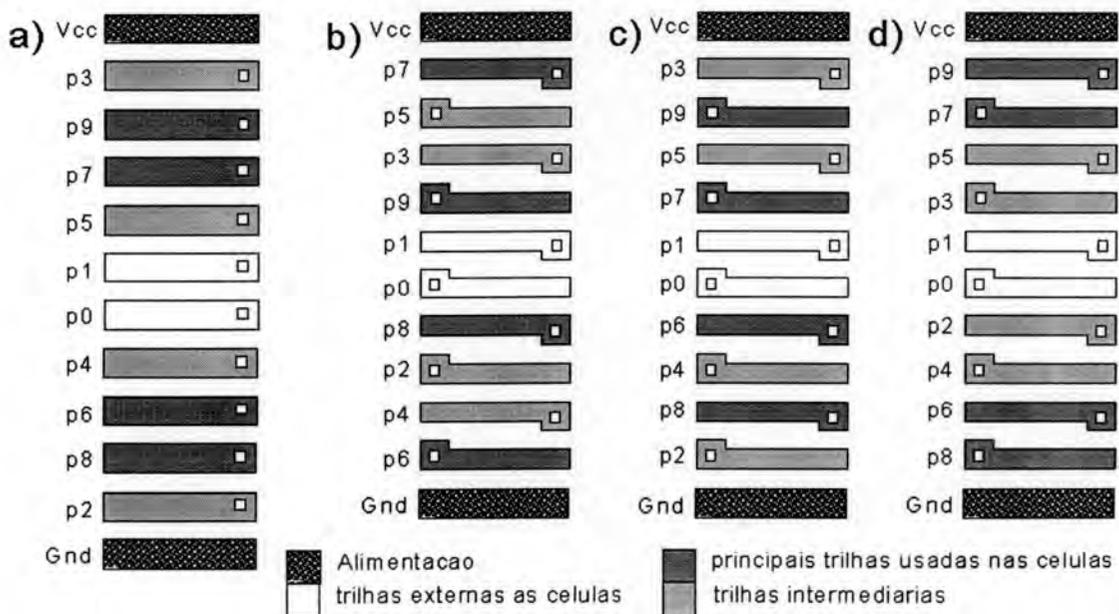


FIGURA 5.3 - Esquemas de prioridades para alocação de trilhas.

Para células pequenas e partes básicas [REI 93b], a principal mudança que isto acarreta é a utilização de trilhas diferentes para os terminais de saída da célula (sempre dois, horizontais, em metal 1). Uma dificuldade que surge é o alinhamento dos terminais

de saída e entrada, que, quando otimizadas nas células, podem não coincidir no acesso a uma célula vizinha.

O esquema de prioridades da fig.5.3c foi feito para permitir um rigoroso alinhamento, permitindo que somente as trilhas mais escuras sejam utilizadas para o desenho da célula. Já o esquema apresentado na fig.5.3d tenta deixar sempre os transistores e a saída o mais próximo possível das alimentações, minimizando as áreas de dreno e fonte, mas rompendo com este alinhamento. Na fig.5.3a pode-se comparar o esquema de prioridades utilizado na abordagem MARCELA, onde, embora as células não utilizem metais, os terminais estão alinhados e provocam a alocação de trilhas nesta ordem.

5.4 Posicionamento básico de transistores

A posição relativa dos transistores também pode ser explorada de forma alternativa, revisando as partes básicas apresentadas em [REI 93b].

Para mais de dois transistores em paralelo, pode-se explorar outra alternativa se estes não puderem aproveitar a transparência efetiva em forma de "S". Com o esquema de prioridades das figuras 5.3c e 5.3d, alguns deles podem ser colocados abaixo dos demais, bastando utilizar um segmento de metal 2 para trazer a alimentação ao centro da célula, reduzindo bastante a sua largura. O uso de uma trilha em metal 2 não traz problemas de roteamento, porque este nível é geralmente livre nos sistemas TRANCA, e está associado à direção vertical, com menor necessidade de roteamento.

A alteração de algumas das topologias de partes básicas ou a divisão de transistores (uso de dois transistores controlados pela mesma porta) também podem ser utilizadas para aprimorar o posicionamento destes, a conectividade interna de células, minimizar o uso de trilhas internas, e maximizar a transparência efetiva.

5.5 Gerenciamento de altura variável

Uma importante questão a ser gerenciada e otimizada na geração células para o sistema TRAMO3 é quanto à altura de banda variável. Com esta alternativa, o uso das prioridades, posicionamentos e observações descritas, varia de forma significativa, principalmente considerando o uso de trilhas alternadas. Para inclusão das outras trilhas, alternadas ou justapostas, é preciso considerar o aproveitamento da transparência efetiva, o gerenciamento das trilhas e o impacto no desempenho dos transistores.

6 A Geração Automática de Células

6.1 Considerações iniciais

O algoritmo para a geração de células, adequadas ao roteamento *Full Over-the Cell* (FOTC), foi baseado na proposta apresentada em [REI 93b]. Modificações com relação ao algoritmo inicial foram propostas, assim como algumas restrições impostas no sentido de tornar o problema tratável. Além disso, a geração de algumas células de interesse prático, que não se enquadram no modelo original, foi considerada.

Conforme mostra a fig.6.1, a gerador de células é basicamente dividido em quatro etapas: processamento do *netlist* de entrada, posicionamento dos transistores, roteamento simbólico e compactação da descrição simbólica.

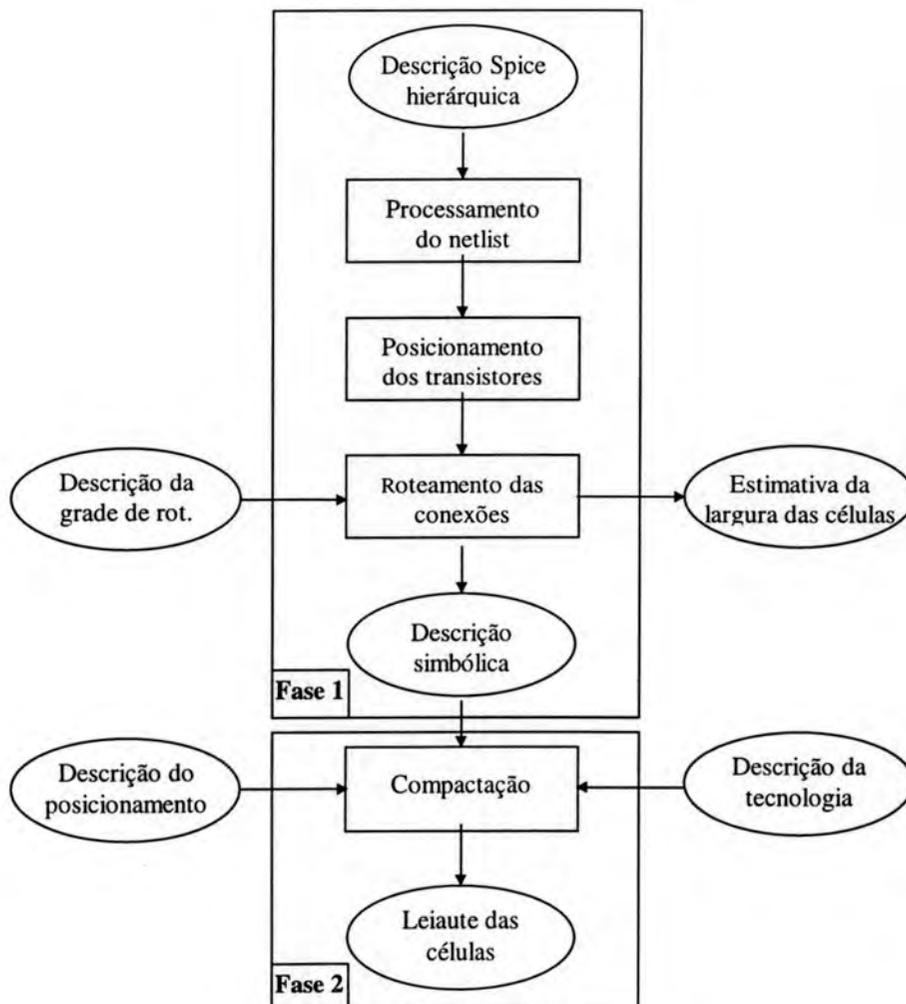


FIGURA 6.1 - Seqüência de passos utilizada na geração simbólica

Estas etapas estão por sua vez divididas em duas fases, que dizem respeito a interação com o posicionador do sistema TRAMO3. A primeira fase é realizada antes do posicionamento.

6.2 Definição de Conceitos

Alguns conceitos devem ser compreendidos antes da descrição do procedimento do gerador de células. Segue a definição desses conceitos em conformidade com [REI 93b].

Dois transistores P e N foram um **par dual** se e somente se:

1. tiverem o mesmo sinal de porta;
2. as conexões em série (paralelo) na região P da célula correspondam a conexões paralelo (série) na região N da célula, para cada par PN de transistores com o mesmo sinal de porta.

Conexão dual série/paralelo (paralelo/série) é um tipo de conexão em que os elementos (transistores ou grupos de transistores) são conectados em série (paralelo) na região P da célula enquanto seus duais são conectados em paralelo (série) na região N.

Lógica CMOS complementar pura é a lógica formada por pares duais PN de transistores conectados dualmente em série/paralelo nas regiões P e N da célula.

Dado um *netlist* em lógica CMOS complementar pura, para a geração de uma célula-folha, existe pelo menos um ponto onde este *netlist* pode ser desmembrado em dois outros *netlists* conectados em série/paralelo (paralelo/série) nas regiões P e N da célula, e assim sucessivamente, até que as parcelas formadoras não possuam mais que um par dual PN cada uma. Assim, qualquer *netlist* que apresente apenas lógica CMOS complementar pura pode ser decomposto em dois outros conectados em série/paralelo, a menos que só possua um par dual de transistores. As conexões existentes nas regiões P e N da célula podem ser representadas por duas árvores binárias de topologia idêntica, mas diferindo quanto a associação (série ou paralelo) atribuída a cada nó. Nessa árvore, os nodos folha representam os transistores, enquanto os demais nodos representam as associações entre transistores ou grupo de transistores.

Árvore dual é o nome dado às árvores binárias descritas no parágrafo acima.

Linhas de corte conjugadas são duas linhas que separam árvores duais em pontos correspondentes de sua topologia ao longo do caminho formado apenas pelos ramos esquerdos a partir do nodo raiz. As árvores binárias consideradas aqui apresentam ainda outra característica importante. Para qualquer nodo da árvore, a sub-árvore da direita nunca tem complexidade superior à da esquerda.

Parte principal: é a parcela do *netlist* que fica abaixo de determinada linha de corte conjugada.

Parte acrescida: é a parcela do *netlist* que fica entre determinada linha de corte conjugada e a próxima linha de corte conjugada superior à primeira.

Parte básica: é uma parcela de *netlist* que possua não mais que quatro pares duais PN.

Parte composta: é uma parcela de *netlist* formada de mais de quatro pares duais PN conectados de maneira não homogênea.

6.3 Processamento da entrada

A descrição de entrada, possivelmente hierárquica, é planificada ao nível de transistores e processada, de modo semelhante ao que é realizado pelos módulos TRAGO e TROPIC, com o objetivo de encontrar as células-folha que definem o circuito. Essas células-folha devem ter as seguintes características:

- apresentar topologia CMOS complementar (ramo p-MOS dual ao ramo n-MOS)
- apresentar apenas conexões série ou paralelo entre transistores.

Essas duas restrições teóricas são normalmente adotadas no projeto lógico, o que não causa grandes limitações na prática. Exceto para o caso de *transmission gates*, que, por isso mesmo, são também considerados aqui.

A leitura do *netlist* de entrada é feita conforme descrito em [REI 93b]. A descrição planificada é percorrida buscando associações série/paralelo entre transistores ou entre associações de transistores já formadas, até que não haja mais nenhuma associação possível. Nesta fase também são encontrados os *transmission gates*, que são pares de transistores com dreno e fonte comuns, mas gates distintos, bem como é realizada a verificação da conectividade do circuito. Assim, qualquer transistor que não pertença a nenhuma célula-folha é reportado e a síntese não continua.

Uma vez gerado o arquivo com a descrição das células-folha, inicia-se o procedimento de geração de cada uma das células, incluindo o posicionamento e roteamento. A compactação é realizada individualmente também, mas numa fase posterior, como será visto em seguida.

Diferentes versões de uma mesmo tipo de célula, por exemplo duas células *nand* com tamanhos de transistores diferentes, são sintetizadas de forma independente. Todavia, há o cuidado de não se fazer o mesmo para células com transistores idênticos

6.4 Posicionamento dos transistores

A fase de posicionamento inicia com a leitura da descrição Spice da célula a ser gerada, procedendo-se a montagem da árvore binária que descreve a topologia da célula, da mesma forma que foi descrito no conceito de árvore dual, ou seja, garantindo que, para cada nodo, seu filho da direita não seja mais complexo que o nodo da esquerda.

Em [REI 93b] foram definidas seis topologias, chamadas de partes básicas, que são formadas por não mais de quatro pares de transistores. A principal característica desse grupo de partes básicas é que elas descrevem qualquer possibilidade de associação de, até 4 transistores por ramo. As partes básicas descrevem as seguintes associações (considerando o ramo P):

- Topologia A: A//B//C e seu dual AsBsC;

- Topologia B: $(AsB)//C$ e seu dual $(A//B)sC$;
- Topologia C: $(AsB)//(CsD)$ e seu dual $(A//B)s(C//D)$;
- Topologia D: $(A//B//C)sD$ e seu dual $(AsBsC)//D$;
- Topologia E: $(AsB)//C//D$ e seu dual $(A//B)s(CsD)$;
- Topologia F: $((AsB)//C)sD$ e seu dual $((A//B)sC)//D$.

O anexo 2 apresenta uma descrição simbólica dessas topologias que difere das apresentadas em [REI 93b] para os casos B, C e E, com o objetivo de melhorar a relação entre a largura das células e o número de trilhas de metal 1 utilizadas. Essas modificações foram determinadas baseado em observações práticas. Entretanto, a definição das topologias é definida por meio de um arquivo, sendo possível alterá-las sem modificação no código do programa.

Uma vez montada a árvore binária, percorre-se a mesma em caminhamento central até que seja feito um casamento com alguma parte básica. Uma vez encontrada a parte básica, se não houver parte principal, a parte encontrada torna-se a principal. Caso contrário, será uma parte acrescida que será composta com a parte principal. Essa composição torna-se a nova parte principal e processo se repete até que tenha se chegado ao topo da árvore.

Dessa forma, o posicionamento dos transistores é feito de forma procedural, com topologias fixas, numa tentativa de incorporar o conhecimento do projetista. A seguir estão listados os principais passos seguidos durante o processo de posicionamento dos transistores:

- Montagem da árvore binária ordenada que representa as conexões série e paralelo entre os transistores;
- A árvore é percorrida (caminhamento central) até que seja identificada como uma parte básica;
- template da parte básica é posicionado e passa a ser considerado uma parte principal;
- Se o nodo raiz ainda não foi alcançado, continua percorrendo a árvore até que seja encontrada uma nova parte básica, que será chamada de parte acrescida;
- Posiciona-se a parte acrescida justaposta à parte principal;
- Repete-se o processo para a parte acrescida até que o nodo raiz seja alcançado

6.5 Roteamento das conexões

O roteamento das conexões para cada parte básica já é pré-definido, restando a interconexão das diferentes partes que compõem a célula. Esse roteamento é realizado seguindo a prioridade de alocação de trilhas indicada na fig.5.3.

Conforme mostrado na fig.6.1, a etapa de roteamento necessita de informações sobre a organização de grade que será utilizada posteriormente pelo roteamento global do circuito. Nesse sentido, as opções para as trilhas horizontais e verticais são:

- Trilhas justapostas, onde pode ser inserido um contato em qualquer ponto da grade;
- Trilhas alternadas, onde os contatos só podem ser colocados de forma alternada.

Na versão atual, está sendo utilizada uma configuração de trilhas alternadas na horizontal e trilhas justapostas na vertical.

Uma vez efetuado o roteamento, o gerador de células gera um arquivo com a lista das células e suas larguras estimadas. esta informação será utilizada pelo posicionador. Não é possível fornecer o valor exato das larguras, uma vez que, esta pode variar em função da altura das células, que será determinada pela necessidade de roteamento intrabanda a ser estimada pelo posicionador.

A seguir estão listados os principais passos seguidos durante o processo de roteamento interno entre as partes que compõem uma célula (principal e acrescida):

- Se o ramo-N estiver em paralelo:
- expande alimentações e saída-N da parte acrescida;
- conecta saída-P da parte principal à saída-P da parte acrescida;
- considera a alimentação da saída-P da parte acrescida como a nova saída-P da parte principal.

6.6 Compactação da descrição simbólica

A etapa de compactação é baseada no compactador SYLC, apresentado em [SOT 94]. Entretanto, devido a peculiaridades do nosso sistema, não foi possível utilizar o compactador diretamente. Sendo assim, decidiu-se pela implementação de um compactador dedicado. O compactador trabalha sobre um grafo de restrições montado a partir da posição dos elementos simbólicos da célula. As restrições são geradas de acordo com um arquivo de tecnologia, que contém as regras de separação e conectividade entre as diversas camadas.

Resumidamente, o compactador é de uma dimensão e trabalha em duas etapas principais:

1. Geração do grafo de restrições: um grafo representando todas as restrições de afastamento e conectividade do leiaute do circuito (para X ou Y) é montado de acordo com a posição dos elementos simbólicos e conforme as regras de projeto. O método de sombreamento [SHE 88] é utilizado.

2. Compactação do grafo: o caminho crítico é determinado através de um algoritmo de *breath-first search*, desenvolvido por Moore em 1959 [PRE 88]. Para encontrar o posicionamento de cada nodo, diversas iterações são realizadas sobre o

grafo. Cada iteração ajusta a posição dos nodos de acordo com as restrições presentes no grafo. O processo termina quando na última iteração a posição de nenhum nodo tenha sido alterada.

Essas etapas são executadas sobre o circuito alternadamente em X e Y até que não seja mais possível compactá-lo.

A razão pela qual foi necessário o desenvolvimento de um compactador dedicado está no fato de que no TRAMO3 as entradas e saídas das células devem se posicionadas sobre a grade de roteamento. Assim, uma parte da descrição simbólica é livre durante a compactação enquanto outra, pode ser movimentada apenas em posições discretas.

6.7 Estado da ferramenta

A etapa de geração simbólica do gerador de células encontra-se operacional tanto em ambiente Unix como DOS. Uma série de restrições foram impostas para tornar essa versão disponível. As principais são:

- Ausência de transistores em 'S'. Como foi visto anteriormente a transparência efetiva só pode ser alcançada completamente quando se usam transistores com W elevado ($W > 5 W_{min}$, por exemplo) através de transistores em 'S';
- O número de transistores em série foi limitado a um máximo de quatro, embora, considerando questões de desempenho, embora essa limitação não seja muito severa.

A etapa de compactação do gerador foi implementada, mas não está funcional, o que restringe o uso da ferramenta, uma vez que outro compactador, que não atende as restrições do sistema deve ser utilizado

Por outro lado, existem alguns procedimentos simples que ainda não foram implementados:

- Introdução de MUX: o uso de *transmission gates* produz layouts com menor densidade quando comparados a circuitos puramente lógicos. Foi feito um estudo de uma configuração específica que implementa um MUX2:1, maneira usual de configuração dos *transmission gates*, que reduz esse problema.
- Uso de transistores *branch*: a colocação de transistores duplicados (com o mesmo *gate*) aumenta o número de trilhas disponíveis para o roteamento. Mesmo podendo reduzir o desempenho, seria uma opção desejável no gerador de células.

6.8 Estado do sistema

O sistema TRAMO3 não está completamente funcional. Além da compactação do gerador de células não estar trabalhando, existem alguns problemas de conectividade entre os módulos. O problema mais sério, diz respeito a geração da matriz de restrições utilizada pelo roteador. Essa matriz contém informações extraídas do layout das células que especificam, para cada ponto da grade de roteamento, que camadas podem atravessar e/ou conectar esse ponto. Da mesma forma, essa matriz contém as informações sobre as entradas e saídas das células, que podem estar disponíveis em

diversas camadas e em mais de um ponto. Assim, para que o sistema esteja completo, a extração da matriz de restrições deve ser realizada de forma automática.

7 Resultados

Este capítulo apresenta alguns resultados do protótipo da ferramenta para a síntese automática de células desenvolvida neste trabalho. Os resultados são parciais pois a etapa de compactação dedicada do gerador não está funcional. Sendo assim, o compactador SYLC foi utilizado para a obtenção dos leiautes apresentados. Entretanto, esses leiautes não pode ser utilizados no sistema TRAMO3, o que impossibilita a síntese circuitos completos.

7.1 Implementação de um circuito lógico

A função booleana $F = \overline{\overline{A + B.C}}$ pode ser implementada pelo circuito lógico apresentado na fig.7.1, composto por duas portas lógicas.

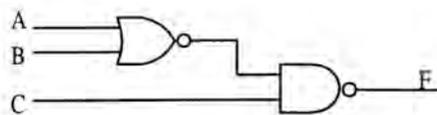


FIGURA 7.1 Circuito lógico

O sistema TRAMO3 pode gerar o leiaute correspondente a este circuito de duas formas distintas:

- implementação direta, onde o gerador de células seria usado para gerar as duas células (nor de duas entradas e nand de duas entradas);
- usando uma célula complexa ($F = \overline{\overline{A.B.C}}$)

A utilização de células complexas normalmente produz circuitos com um número menor de transistores, sendo portanto preferida. Entretanto, nesse caso, devido à baixa complexidade do circuito lógico, a segunda implementação apresenta um custo mais elevado, quanto ao número de transistores, uma vez que, além da célula complexa, são necessários mais dois inversores para a complementação dos literais da função lógica. A fig.7.2 apresenta o esquemático e o leiaute da célula complexa correspondente à função lógica $F = \overline{\overline{A.B.C}}$.

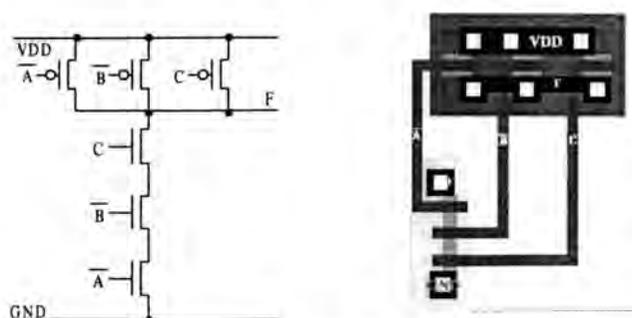


FIGURA 7.2 - Esquemático e leiaute da célula $F = \overline{\overline{A.B.C}}$

7.2 Implementação de um bit somador

A partir das equações booleanas simplificadas para um bit de soma, indicadas abaixo, pode-se obter de forma direta o circuito lógico correspondente:

$$Co = A.B + A.Ci + B.Ci = \overline{\overline{A.B.A.Ci.B.Ci}};$$

$$S = \overline{A.B.Ci} + \overline{A.B.Ci} + A.B.Ci + \overline{A.B.Ci} = A \oplus B \oplus Ci$$

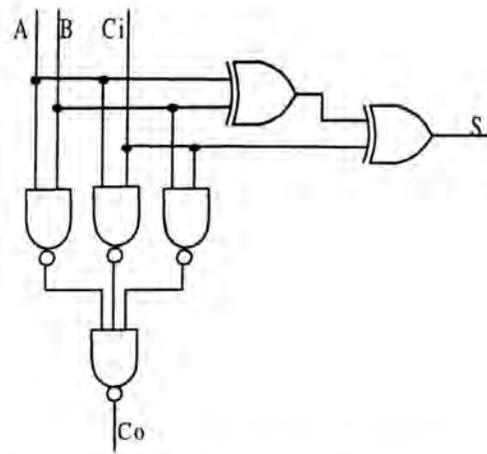


FIGURA 7.3 - Circuito lógico do somador de 1 bit

Este circuito lógico é adequado para ser utilizado em um gerador baseado em biblioteca de células. Entretanto, em um sistema capaz de gerar células complexas, cada equação pode ser implementada por uma única célula específica e, eventualmente, inversores para complementar entradas e/ou saídas. Assim, as equações do somador de um bit podem ser manipuladas, obtendo-se:

$$Co = \overline{(\overline{A+B}).(\overline{A+Ci}).(\overline{B+Ci})};$$

$$S = \overline{(\overline{A+B+Ci}).(\overline{A+B+Ci}).(\overline{A+B+Ci}).(\overline{A+B+Ci})}$$

O circuito que implementa a saída Co exige 3 inversores para cada uma das entradas e mais uma célula com 12 transistores, enquanto o circuito que implementa S exige os mesmos inversores e mais uma célula com 24 transistores. Com o objetivo de reduzir o número de transistores da célula específica para a saída S , pode-se representar S como $f(A, B, Ci, Co)$:

$$S = \overline{Co}.(A+B+Ci) + (A.B.Ci)$$

Finalmente, vamos representar as saídas (S e Co) negadas, a fim de reduzir o número de inversores necessários para completar o circuito:

$$\overline{Co} = \overline{A.B + A.Ci + B.Ci};$$

$$\overline{S} = \overline{\overline{Co}.(A+B+Ci) + (A.B.Ci)}$$

Conforme a fig.7.4, o circuito resultante apresenta dois inversores para cada uma das saídas, uma célula complexa para a saída $\overline{C_o}$, com 12 transistores e uma célula complexa para a saída \overline{S} , com 14 transistores.

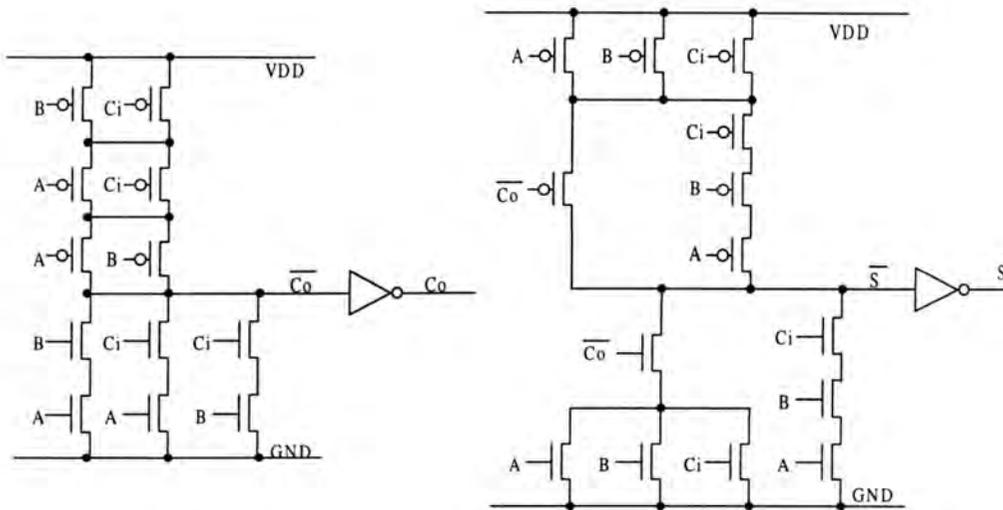


FIGURA 7.4 - Esquemático do circuito somador de 1 bit

A fig.7.5 apresenta os leiautes das duas células complexas. Conforme citado anteriormente, as descrições simbólicas das células foram geradas automaticamente. Entretanto, o leiaute foi obtido com o compactador SYLC, uma vez que o compactador dedicado ainda não está funcional. Sendo assim, esses leiautes não são adequados ao uso no sistema TRAMO3.

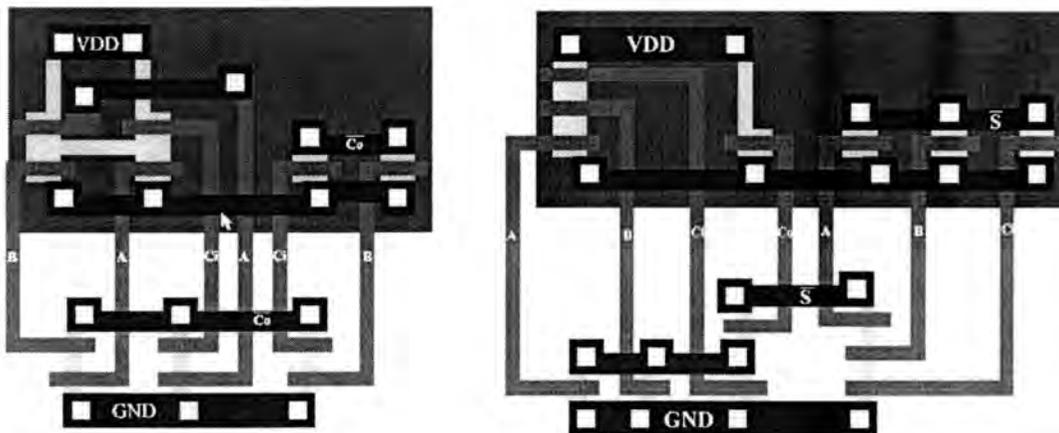


FIGURA 7.5 - Leiautes das células complexas do somador de 1 bit.

8 Conclusões e trabalhos futuros

Este trabalho buscou estudar as alternativas para a síntese de células, tendo em vista, principalmente a sua adequação à metodologia TRANCA e com vistas a sua inserção no sistema TRAMO3. Para tanto, foi apresentada uma revisão sobre síntese de leiaute, um histórico da metodologia TRANCA e os sistemas TRAMO2 e TRAMO3 em profundidade.

Foi apresentado um estudo sobre as características desejáveis para a topologia das células no TRAMO2 (e TRAMO3), tendo em vista as condições de contorno oferecidas (impostas) pelo sistema de roteamento.

Uma descrição da primeira versão da ferramenta foi apresentada, contendo uma série de restrições que devem ser revistas em seguida. A principal atividade a ser realizada de imediato é a conexão do gerador de células ao sistema TRAMO3, a realizar uma série de circuitos de *benchmark*, a fim de efetivamente validar o funcionamento da ferramenta e comparar o desempenho global do sistema com outras alternativas de projeto.

Entre as atividades de médio prazo, incluem-se os melhoramentos citados no capítulo 6, com a implementação de transistores do tipo *branch* e da célula MUX. Além disso, tendo em vista o resultado observado nos circuitos, algumas das restrições incorporadas nessa versão deverão ser removidas.

Alguns trabalhos relacionados com o gerador de células também estão em desenvolvimento no contexto do projeto TRANCA, como o desenvolvimento de um mapeador tecnológico que aproveite a possibilidade da síntese de AOIs. Da mesma forma, ferramentas de análise de *timing* e de dimensionamento de transistores são essenciais para que se possa tirar proveito de todas as possibilidades oferecidas pela geração de células.

Por fim, também encontra-se em desenvolvimento a definição de novas topologias voltadas para tecnologias com mais de dois níveis de metal. A idéia nesse caso é utilizar o primeiro nível de metal livremente no interior da célula, deixando os outros níveis reservados às conexões entre células, com orientações bem definidas.

Anexo A-1 Biblioteca de Células para o TRAMO2

A seguir são apresentadas as células que foram utilizadas para a implementação dos circuitos com o TRAMO2. Para cada células são apresentados o leiaute, números de transistores, dimensões, capacitâncias para cada entrada, e os atrasos de subida e descida entre entradas e saídas.

NOME : INVERSOR

PARÂMETRO	VALOR	UNIDADE
Tamanho	1288,17	μm^2
Cin_A	2,73	fF
Transistores	2	

Descrição:

PARÂMETRO	DE	PARA	CURVA	INTERVALO D
tdlh	A	OUT	$215,61\text{p}+2082,24\text{Cout}$	$20\text{fF}<\text{Cout}<1800\text{fF}$
tdhl	A	OUT	$116\text{p}+1210,28\text{Cout}$	$50\text{fF}<\text{Cout}<770\text{fF}$
tr	A	OUT	$342,54\text{p}+4340,86\text{Cout}$	$20\text{fF}<\text{Cout}<1800\text{fF}$
tf	A	OUT	$252,97\text{p}+2239,99\text{Cout}$	$50\text{fF}<\text{Cout}<770\text{fF}$



NOME : NAND2

PARÂMETRO	VALOR	UNIDADE
Tamanho	1288,17	μm^2
Cin_A	3,42	fF
Cin_B	2,55	fF
Transistores	4	

Descrição:

PARÂMETRO	DE	PARA	CURVA	INTERVALO
tdlh	A	OUT	$302,04\text{p}+3052,51\text{Cout}$	$20\text{fF}<\text{Cout}<1000\text{fF}$
tdhl	A	OUT	$232,96\text{p}+1792,38\text{Cout}$	$70\text{fF}<\text{Cout}<640\text{fF}$
tdlh	B	OUT	$380,19\text{p}+3146,60\text{Cout}$	$20\text{fF}<\text{Cout}<860\text{fF}$
tdhl	B	OUT	$235,14\text{p}+1858,24\text{Cout}$	$50\text{fF}<\text{Cout}<550\text{fF}$
tr	A	OUT	$466,32\text{p}+6585,98\text{Cout}$	$20\text{fF}<\text{Cout}<1000\text{fF}$
tf	A	OUT	$352,05\text{p}+4013,68\text{Cout}$	$70\text{fF}<\text{Cout}<640\text{fF}$
tr	B	OUT	$609,06\text{p}+6682,95\text{Cout}$	$20\text{fF}<\text{Cout}<860\text{fF}$
tf	B	OUT	$336,50\text{p}+4041,96\text{Cout}$	$50\text{fF}<\text{Cout}<550\text{fF}$



NOME : NAND3

PARÂMETRO	VALOR	UNIDADE
Tamanho	1791,9	μm^2
Cin_A	3,64	fF
Cin_B	2,84	fF
Cin_C	1,71	fF
Transistores	6	

Descrição:

PARÂMETRO	DE	PARA	CURVA	INTERVALO D
tdlh	A	OUT	$423,91\text{p}+2859,04\text{Cout}$	$50\text{fF}<\text{Cout}<670\text{fF}$
tdhl	A	OUT	$436,44\text{p}+2344,68\text{Cout}$	$50\text{fF}<\text{Cout}<660\text{fF}$
tdlh	B	OUT	$518,76\text{p}+2887,30\text{Cout}$	$50\text{fF}<\text{Cout}<640\text{fF}$
tdhl	B	OUT	$485,74\text{p}+2374,17\text{Cout}$	$50\text{fF}<\text{Cout}<630\text{fF}$
tdlh	C	OUT	$621,32\text{p}+2863,01\text{Cout}$	$20\text{fF}<\text{Cout}<650\text{fF}$
tdhl	C	OUT	$507,10\text{p}+2398,48\text{Cout}$	$50\text{fF}<\text{Cout}<580\text{fF}$
tr	A	OUT	$743,93\text{p}+2859,04\text{Cout}$	$50\text{fF}<\text{Cout}<670\text{fF}$
tf	A	OUT	$693,44\text{p}+2344,68\text{Cout}$	$50\text{fF}<\text{Cout}<660\text{fF}$
tr	B	OUT	$904,63\text{p}+2887,30\text{Cout}$	$50\text{fF}<\text{Cout}<640\text{fF}$
tf	B	OUT	$685,85\text{p}+2374,17\text{Cout}$	$50\text{fF}<\text{Cout}<630\text{fF}$
tr	C	OUT	$1091,9\text{p}+6571,87\text{Cout}$	$20\text{fF}<\text{Cout}<650\text{fF}$
tf	C	OUT	$694,96\text{p}+5958,90\text{Cout}$	$50\text{fF}<\text{Cout}<580\text{fF}$

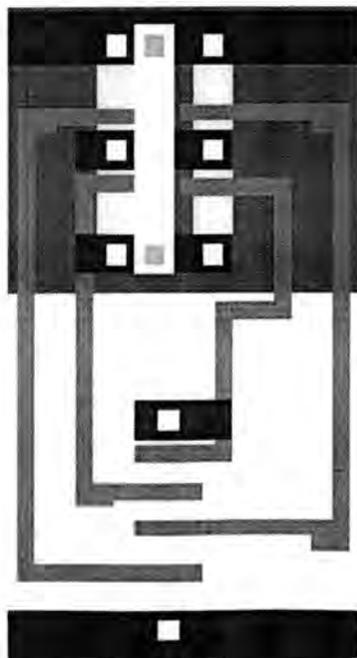


NOME : NAND4

PARÂMETRO	VALOR	UNIDADE
Tamanho	2338,74	μm^2
Cin_A	4,45	fF
Cin_B	3,01	fF
Cin_C	3,82	fF
Cin_D	2,63	fF
Transistores	8	

Descrição:

PARÂMETRO	DE	PARA	CURVA	INTERVALO
tdlh	A	OUT	777,85p+2332,33Cout	50fF<Cout<610fF
tdhl	A	OUT	800,49p+3065,80Cout	50fF<Cout<590fF
tdlh	B	OUT	538,83p+2506,26Cout	50fF<Cout<560fF-
tdhl	B	OUT	633,14p+3148,48Cout	20fF<Cout<600fF
tdlh	C	OUT	878,87p+2270,70Cout	50fF<Cout<600fF
tdhl	C	OUT	871,17p+2953,81Cout	50fF<Cout<580fF
tdlh	D	OUT	700,83p+2257,45Cout	50fF<Cout<600fF
tdhl	D	OUT	792,61p+3154,00Cout	50fF<Cout<550fF
tr	A	OUT	1341,1p+6766,64Cout	50fF<Cout<610fF
tf	A	OUT	1266,46p+7778,7Cout	50fF<Cout<590fF
tr	B	OUT	986,92p+6689,25Cout	50fF<Cout<560fF-
tf	B	OUT	1224,82p+7925,08Cout	20fF<Cout<600fF
tr	C	OUT	1520,24p+6734,89Cout	50fF<Cout<600fF
tf	C	OUT	1256,44p+7877,3Cout	50fF<Cout<580fF
tr	D	OUT	1147,3p+6764,46Cout	50fF<Cout<600fF
tf	D	OUT	1252,72p+7828,14Cout	50fF<Cout<550fF



NOME : NOR2

PARÂMETRO	VALOR	UNIDADE
Tamanho	1548,18	μm^2
Cin_A	2,94	fF
Cin_B	2,21	fF
Transistores	4	

Descrição:

PARÂMETRO	DE	PARA	CURVA	INTERVALO
tdlh	A	OUT	$315,21\text{p}+3743,13\text{Cout}$	$20\text{fF}<\text{Cout}<1500\text{fF}$
tdhl	A	OUT	$327,80\text{p}+1421,48\text{Cout}$	$20\text{fF}<\text{Cout}<560\text{fF}$
tdlh	B	OUT	$572,32\text{p}+3848,51\text{Cout}$	$20\text{fF}<\text{Cout}<1200\text{fF}$
tdhl	B	OUT	$465,93\text{p}+1341,21\text{Cout}$	$20\text{fF}<\text{Cout}<560\text{fF}$
tr	A	OUT	$528,13\text{p}+7446,90\text{Cout}$	$20\text{fF}<\text{Cout}<1500\text{fF}$
tf	A	OUT	$264,67\text{p}+3300,54\text{Cout}$	$20\text{fF}<\text{Cout}<560\text{fF}$
tr	B	OUT	$567,35\text{p}+7371,67\text{Cout}$	$20\text{fF}<\text{Cout}<1200\text{fF}$
tf	B	OUT	$441,4\text{p}+3227,16\text{Cout}$	$20\text{fF}<\text{Cout}<560\text{fF}$

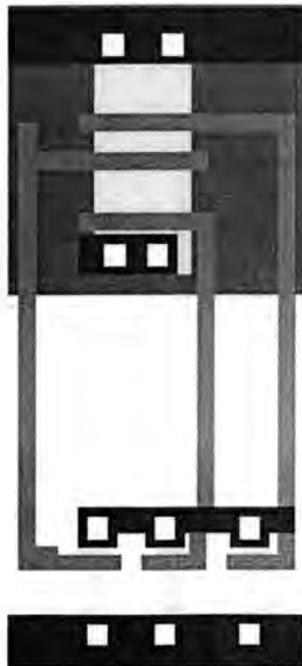


NOME : NOR3

PARÂMETRO	VALOR	UNIDADE
Tamanho	2075,22	μm^2
Cin_A	2,94	fF
Cin_B	2,21	fF
Cin_C	3,44	fF
Transistores	6	

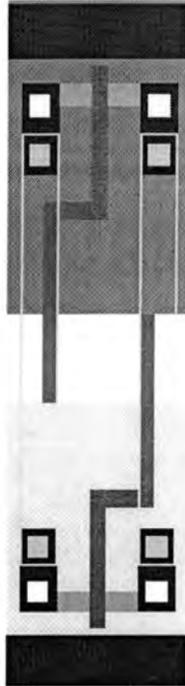
Descrição:

PARÂMETRO	DE	PARA	CURVA	INTERVALO
tdlh	A	OUT	315,21p+3743,13Cout	20fF<Cout<1500fF
tdhl	A	OUT	327,80p+1421,48Cout	20fF<Cout<560fF
tdlh	B	OUT	572,32p+3848,51Cout	20fF<Cout<1200fF
tdhl	B	OUT	465,93p+1341,21Cout	20fF<Cout<560fF
tdlh	C	OUT	696,39p+3714,68Cout	20fF<Cout<1600fF
tdhl	C	OUT	520,07p+1393,64Cout	20fF<Cout<540fF
tr	A	OUT	925,21p+8225,90Cout	20fF<Cout<1500fF
tf	A	OUT	339,40p+3291,21Cout	20fF<Cout<560fF
tr	B	OUT	1037,15p+8072,3Cout	20fF<Cout<1200fF
tf	B	OUT	544,54p+3343,17Cout	20fF<Cout<560fF
tr	C	OUT	1022,9p+8072,84Cout	20fF<Cout<1600fF
tf	C	OUT	737,22p+3619,97Cout	20fF<Cout<540fF



NOME : TG

PARÂMETRO	VALOR	UNIDADE
Tamanho	1791,9	μm^2
Transistores	2	



Anexo A-2 Descrição do parser de entrada

As regras sintáticas que definem a linguagem de entrada (subconjunto de SPICE) do gerador de células são apresentados aqui no formato YACC. Da mesma forma, os *tokens* para a análise léxica da linguagem são apresentados no formato LEX.

A-2.1 Descrição do Analisador Sintático para SPICE em YACC

```

/* File: cgspice.y */
/* Category: Cell Generator component */
/* Description: yacc/bison description for Spice Parser */
/* Author: Marcus Kindel */
/* Last update: 14/08/96 */

%{
#include "cgspice.h"
#include "hash.h"
#include "cellgen.h"

/* altera o tipo default de yylval */

/* comunicacao com o modulo lex.yy.c */
extern int yylineno;

/* comunicacao com o modulo subc.c */
extern char *iname; /* nome do arquivo edif */

extern hashtable *globalht;

pp_node *pino = 0;
pp_list *lpinos = 0;

xt *xstor = 0;
xt_node *xstornode = 0;
xt_list *xstorlist = 0;

cs *subcall = 0;
cs_node *subcallnode = 0;
cs_list *subcalllist = 0;

ds *subdef = 0;
ds_node *subdefnode = 0;
ds_list *subdeflist = 0;

cc *circuito = 0;

%}

%union {

```

```

int num;
char *str;
}

%token <str> XSTOR
%token <str> SUBCALL
%token <str> INT
%token <str> STR
%token <str> STREOL
%token EQUAL
%token FLOATP1
%token FLOATP2
%token FLOATP3
%token COMMENT
%token SUBCKT
%token ENDS
%token END

%type <num> node
%type <num> subthead
%type <num> data
%type <num> subdata

%%

goal:          spice
              ;

spice:         datalist {initspice();}
              | datalist END {initspice();}
              ;

datalist:      data {initdata($1);}
              | datalist data {initdata($2);}
              ;

data:          subcalldef {return 0;}
              | xstordef {return 1;}
              | COMMENT {return 2;}
              | subcdef {return 3;}
              ;

subdatalist:   subdata {initsubdata($1);}
              | subdatalist subdata {initsubdata($2);}
              ;

subdata:       subcalldef {return 0;}
              | xstordef {return 1;}
              | COMMENT {return 2;}
              ;

subthead:      SUBCKT STR nodelist {$$ =
initsubthead($2);}
              ;

```

```

subcdef:          subhead subdatalist ENDS {initsubcdef($1,
0);}
      |
      subhead subdatalist ENDS STR
{initsubcdef($1, $4);}
      ;

subcalldef:      SUBCALL nodelist STREOL {initsubcalldef($1,
$3);}
      ;

node:           INT {$$ = initnode($1);}
      |
      STR {$$ = initnode($1);}
      ;

nodelist:       node {initnodelist($1);}
      |
      nodelist node {initnodelist($2);}
      ;

xstordef:       XSTOR node node node node STR paramlist
{initxstordef($1, $2, $3, $4, $6);}
      ;

paramlist:      paramlist param
      ;

param:          STR EQUAL val {free($1);} /* not used */
      |
      STR {free($1);} /* not used */
      ;

val:           INT {free($1);} /* not used */
      |
      float
      ;

float:          FLOATP1
      |
      FLOATP2
      |
      FLOATP3
      ;

%%

#include "funcl.c"

```

A-2.2 Descrição do Analisador Léxico para SPICE em LEX

```

%{
#include "cgspice.h"
#ifdef MSDOS
#include "y_tab.h"
#else
#include "y.tab.h"
#endif
void mstring(void);

%}

%option yylineno

%s AA

integer      [+]?[0-9]+
pfix1        [0-9]+ "." [0-9]*
pfix2        "." [0-9]+
floatp1      {integer} (T|G|MEG|K|MIL|M|U|u|N|P|F)?
floatp2      [+]?({pfix1}|{pfix2}) ([eE][+-]?[0-9]+)?
floatp3      [+]?({pfix1}|{pfix2}) (T|G|MEG|K|MIL|M|U|u|N|P|F)? [+
]?( {pfix1}|{pfix2}) (T|G|MEG|K|MIL|M|U|u|N|P|F)?
spec        [!@#$$%^&*()\-+|\\\/?<>:~]
%%
^(M|m) [_a-zA-Z0-9]+
{mstring();return XSTOR;}
^(X|x) [_a-zA-Z0-9]+
AA;mstring();return SUBCALL;}
{integer}
{mstring();return INT;}
[_a-zA-Z]+({spec}|[_a-zA-Z0-9])*
{mstring();return STR;}
<AA>[_a-zA-Z]+({spec}|[_a-zA-Z0-9])*/( [ \t]*\n)
0;mstring();return STREOL;}
=
EQUAL;}
{floatp1}
FLOATP1;}
{floatp2}
FLOATP2;}
{floatp3}
FLOATP3;}
^(SUBCKT|subckt)
SUBCKT;}
^(ENDS|ends)
ENDS;}
^(END|end)
END;}
^"*" [^\n]*
COMMENT;}
\n
[ \t]
;
```

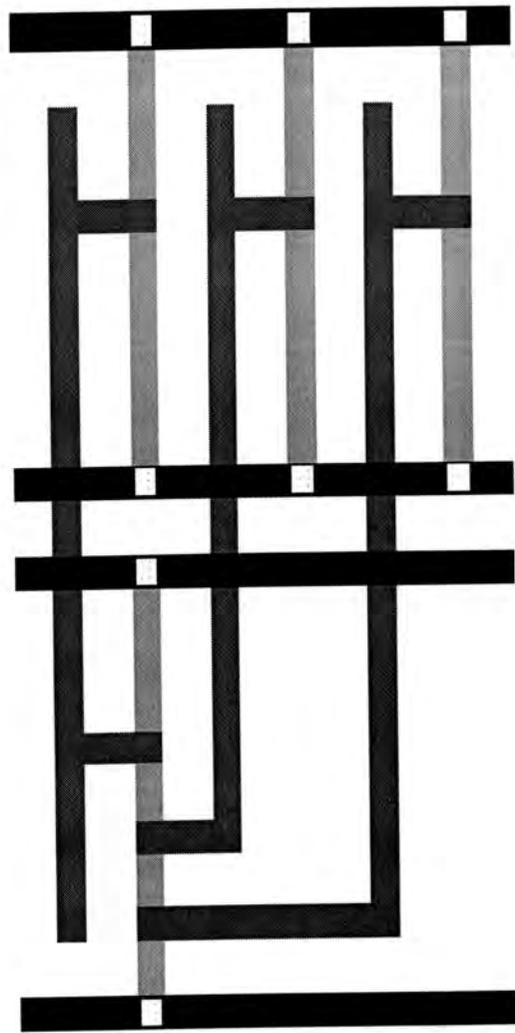
```
.                                     {ECHO;}
%%

int yywrap(void)
{
    return(1);
}

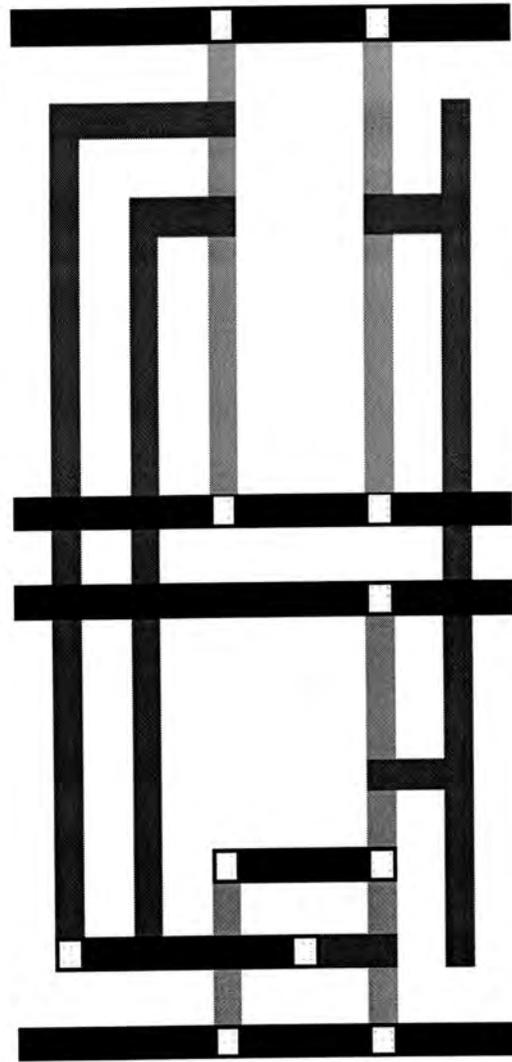
void mstring(void)
{
    aloca(yylval.str, char, yyleng+1);
    strcpy(yylval.str, yytext);
}
```

Anexo A-3 Novas topologias para as partes básicas

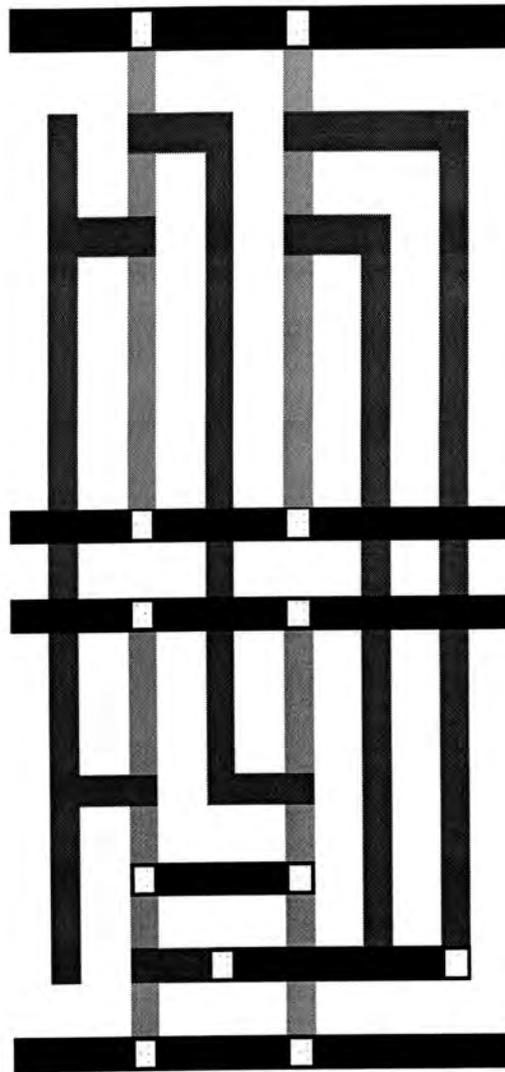
A-3.1 Topologia A



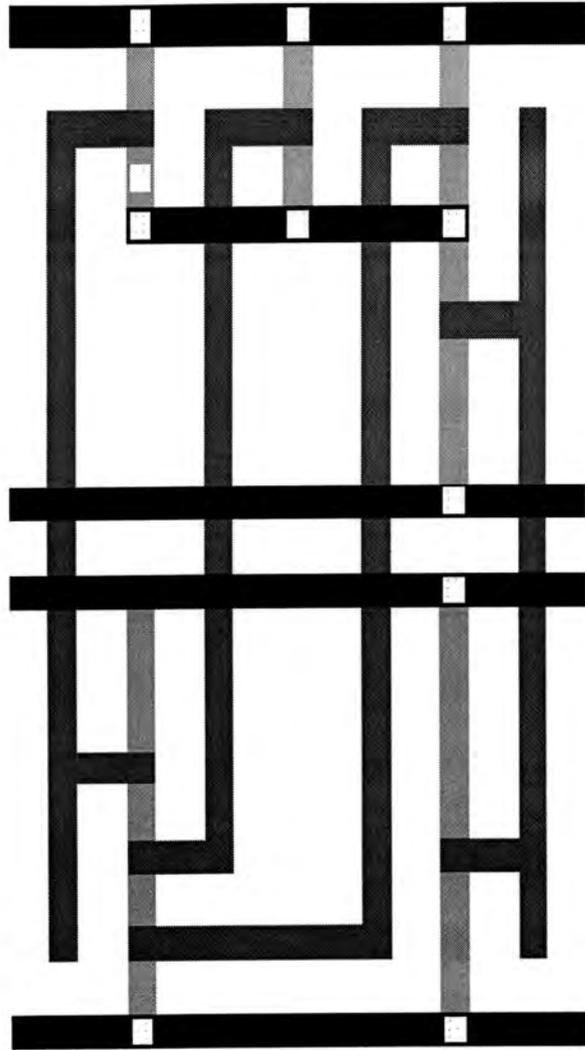
A-3.2 Topologia B

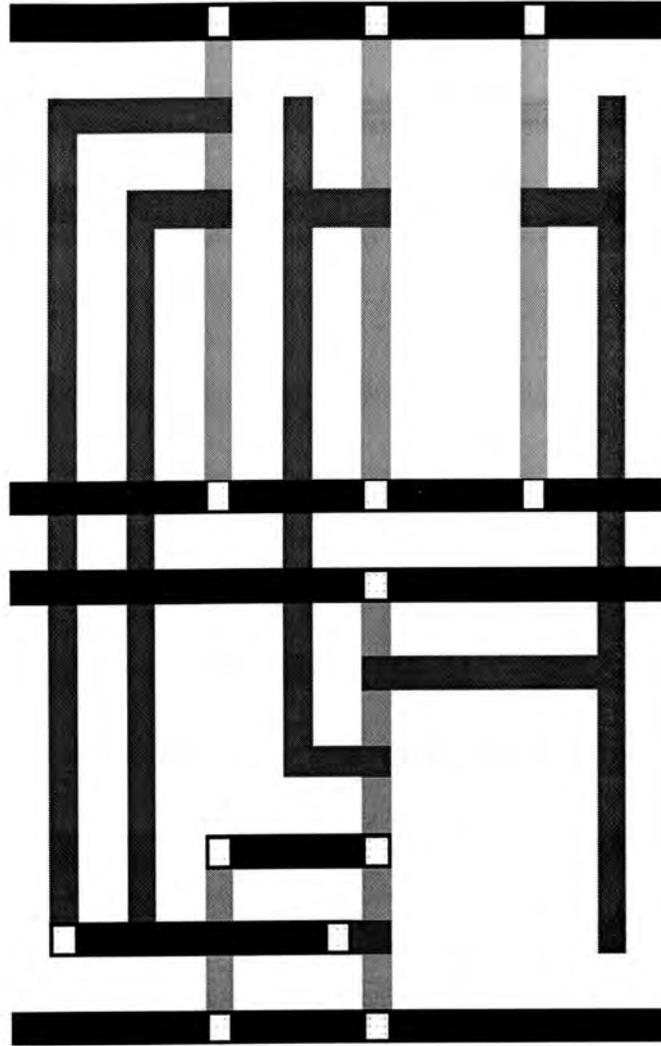


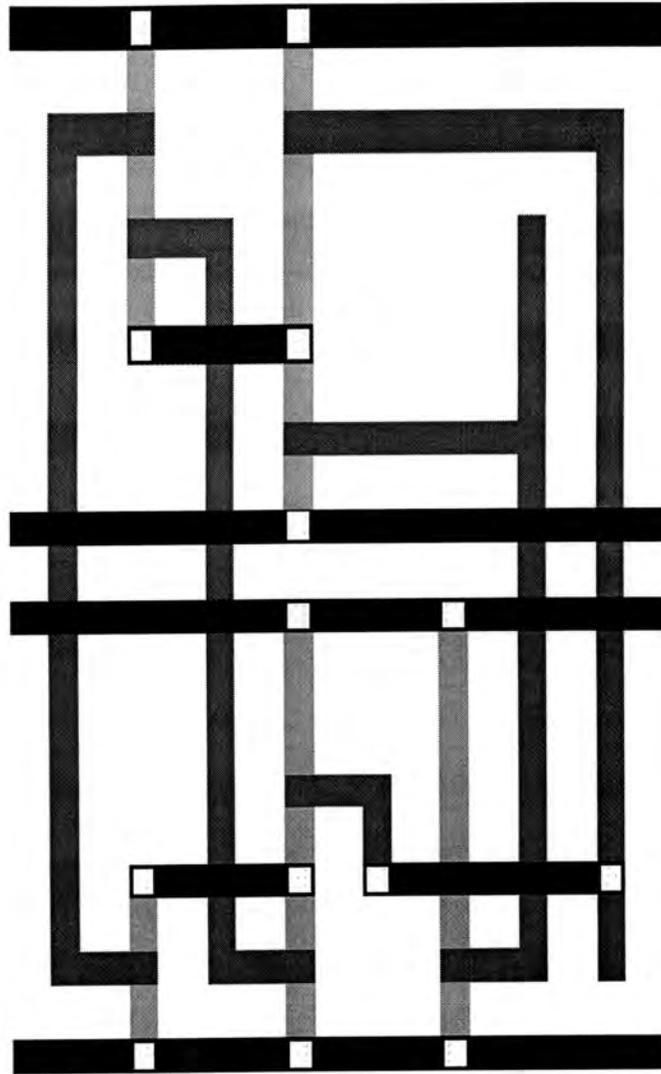
A-3.3 Topologia C



A-3.4 Topologia D



A-3.5 Topologia E

A-3.6 Topologia F

Bibliografia

- [ANN 94] ANNE, Pramod et al. Comparative analysis in new CMOS leaf cells for OTC routing. In: ISCAS, 1994. **Proceedings...** New York: IEEE, 1994. p. 191-194.
- [AUG 70] AUGUSTSON J. G.; MINKER, J. An analysis of some graph theoretical cluster techniques. **Journal of ACM**, New York, v. 17, n. 4, p. 571-588, Oct. 1970.
- [AZE 92] AZEMARD, N.; BONZOM, V.; AUVERGNE, D. P.SIZE: a sizing aid for optimized designs. In: IEEE EUROPEAN DESIGN AUTOMATION CONFERENCE, 1992, Hamburg, Germany. **Proceedings...** Washington: IEEE, 1992. p. 160-165.
- [BAL 88] BALTUS, Donald G.; ALLEN, Jonathan. SOLO: A generator of efficient layouts from optimized MOS circuit schematics. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 25., 1988, Anaheim, Us. **Proceedings...** New York: IEEE, 1988. p. 445-452.
- [BIL 81] BILGORY, Avinoam; GAJSKI, Daniel D. Automatic generation of cells for recurrence structures. In: DESIGN AUTOMATION CONFERENCE, 18., 1981, Nashville, US. **Proceedings...** New York: ACM/IEEE, 1988. p. 306-313.
- [BOE 88] BOEHNER, Michael. LOGEX - An automatic logic extractor from transistor to gate level for CMOS technology. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 25., 1988, Anaheim, Us. **Proceedings...** New York: IEEE, 1988. p. 517-522.
- [BOW 90] BOWER, Wayne; SEAQUIST, Carl. A framework for industrial layout generators. In: DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. 743p. p. 419-424.
- [BOY 88] BOYER, David G. Symbolic compaction review. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 25., 1988, Anaheim, Us. **Proceedings...** New York: IEEE, 1988. p. 383-389.
- [CHU 95] CHUANG Weitong et al. Timing and area optimization for standard-cell VLSI circuit design. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, New York, v. 14, n. 3, p. 308-320, Mar. 1995.
- [COU 96] COUDERT Olivier. Gate sizing: a general purpose optimization approach. In: THE EUROPEAN DESIGN AND TEST CONFERENCE, 1996, Paris. **Proceedings...** Los Alamitos: IEEE Computer Society, 1996.

- [DET 87] DETJENS, E. et al. Technology mapping in MIS. In: ICCAD, 1987, Santa Clara, Us. **Proceedings...** New York: IEEE, 1987. p. 116-119.
- [DOM 90] DOMIC, A. Layout synthesis of MOS digital cells. In: DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. p. 241-245.
- [FUK 95] FUKUI, M.; SHINOMIYA, N.; AKINO, T. A new layout synthesis for leaf cell design. In: International Conference on Very Large Scale Integration. **Proceedings...** Amsterdam: North-Holland, 1995. p. 259-264.
- [GAJ 92] GAJSKI, Daniel D. et al. **High-level synthesis: introduction to chip and system design.** Massachusetts: Kluwer, 1992. 359p.
- [GIG 90] GIDWANI, Roshan; SHERWANI, Naveed A. MISER: an integrated three layer gridless channel router and compacter. In: DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. p. 698-703.
- [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.
- [GUP 96] GUPTA, A. et al. **XPRESS: A cell layout generator with transistor folding.** In: THE EUROPEAN DESIGN AND TEST CONFERENCE, 1996, Paris. **Proceedings...** Los Alamitos: IEEE, 1996.
- [HIL 90] HILL, Dwight D.; PREAS, Bryan. Benchmarks for cell synthesis. In: DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. p. 317-320.
- [HOK 91] HO, King C.; SASTRY, Sarma. Flexible transistor matrix (FTM). In: IEEE DESIGN AUTOMATION CONFERENCE, 28., San Francisco. **Proceedings...** New York: IEEE, 1991. p. 475-480.
- [HSI 90] HSIEH, Yung-Ching. LiB: a cell layout generator. In: DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. p.474-479.
- [HWA 90] WHANG, C. et al. A fast transistor chaining algorithm for CMOS cell layout. In: TCAD, 1990. **Proceedings...** [S.l. : s.n.]. p. 781-786.
- [JOH 93] JOHANN, Marcelo de Oliveira; REIS, Ricardo. MARTE: MAZe RouTer Environment. In: CONGRESSO DA SBMICRO, 8. Campinas, 1993. **Anais...** São Paulo: SBMICRO, 1993. p.37-39.
- [JOH 94] JOHANN, Marcelo de Oliveira. **Desempenho do Sistema de Roteamento MARTE.** Porto Alegre, CPGCC/UFRGS, 1994. 52p. (TI-407).

- [JOH 95] JOHANN, M. et al. A comparison between random logic layout design methodologies. In: CONGRESS OF THE BRAZILIAN MICROELECTRONICS SOCIETY, 10., 1995, Canela, RS. **Proceedings...** Porto Alegre: I.I./UFRGS, 1995. v. 1, p. 1-10.
- [JUS 90] JUST, Knut M. et al. A layout Generator for SCVS logic blocks. In: DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. p. 468-473.
- [KIM 92] KIM, S; OWENS, R. M.; IRWIN, M. J. Experiments with a performance driven module generator. In: DESIGN AUTOMATION CONFERENCE, 29., 1992, Anaheim, Us. **Proceedings...** Los Alamitos: IEEE, 1992. p. 687-690.
- [KIM 94] KIM, J.; KANG, S. M.; SAPATNEKAR, S. S. High performance CMOS macromodule layout synthesis. In: ISCAS, 1994. **Proceedings...** New York: IEEE, 1994. p. 179-182.
- [LEE 92] LEE, J.; WONG, C. K. A performance-aimed cell compactor with automatic jogs. **IEEE transactions on Computer-Aided design**, New York, v.11, n.12, p.1495-1507, Dec. 1992.
- [LEF 90] LEFEBVRE, Martin; CHAN, Chonh; MARTIN, Grant. Transistor placement and interconnect algorithms for leaf cell synthesis. In: EUROPEAN DESIGN AUTOMATION CONFERENCE, 1990. Glasgow, Gb. **Proceedings...** Washington: IEEE, 1990. p. 119-123.
- [LIN 94] LIN, How-Rern et al. Cell height driven transistor sizing in a cell based module design. In: THE EUROPEAN DESIGN AND TEST CONFERENCE, 1994, Paris. **Proceedings...** Los Alamitos: IEEE, 1994.
- [LOP 80] LOPEZ, A. D.; LAW, H. S. A dense gate matrix layout method for MOS VLSI. **IEEE Transactions on Electronic Devices**, New York, v. ED-27, n. 8, p. 1671-1675, 1980.
- [LUB 90] LUBASZEWSKI, Marcelo S. **Geração automática de lógica aleatória utilizando a metodologia TRANCA**. Porto Alegre: CPGCC da UFRGS, 1990. 234p. Dissertação de Mestrado.
- [MAI 88] MAILHOT, Frederic; DEMICHEL, Giovanni. Automatic layout and optimization of static CMOS cells. In: INTERNATIONAL CONFERENCE ON COMPUTER DESIGN, 1988, Rye .Brook, New York. **Proceedings...** Washington: IEEE, 1988. p. 180-185.
- [MAL 96] MALY, Wojciech. Future of testing: reintegration of design, testing and manufacturing. In: THE EUROPEAN DESIGN AND TEST CONFERENCE, 1996, Paris. **Proceedings...** Los Alamitos: IEEE, 1996.

- [MAR 88] MARPLE, D.; SMULDERS, M.; HEGEN, H. An efficient compactor for 45° layout. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 25., 1988, Anaheim, Us. **Proceedings...** Los Alamitos: IEEE, 1988. p. 396-402.
- [MAZ 91] MAZIASZ, Robert L. Exact width and height minimization of CMOS cells. In: IEEE DESIGN AUTOMATION CONFERENCE, 28., 1991. San Francisco. **Proceedings...** New York: IEEE, 1991. 783p. p. 487-495.
- [MEY 88] MEYER, Steve. A data structure for circuit net lists. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 25., 1988, Anaheim, Us. **Proceedings...** Los Alamitos: IEEE, 1988. p. 613-616.
- [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 93] MORAES, F. et al. Flexible Macrocell Layout Generator. In: ACM/SIGDA PHYSICAL DESIGN WORKSHOP, 4., 1993 **Proceedings...** [S.l. : s.n.], 1993 p. 105-116.
- [MOR 94] MORAES, F.; ROBERT, M.; AUVERGNE, D.; REIS, R. An Efficient Layout Synthesis Approach for CMOS Random Logic Circuits. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE MICROELETRÔNICA, 9., 1994, Rio de Janeiro. **Proceedings...** Rio de Janeiro: SBMICRO, 1994. p. 48-57.
- [MOR 95] MORAES, F. et al. Towards optimal use of CMOS complex gates in automatic layout synthesis. In: CONGRESS OF THE BRAZILIAN MICRO-ELECTRONICS SOCIETY, 10, 1995, Canela, RS. **Proceedings...** Porto Alegre: I.L/UFRGS, 1995. v. 1, p. 1-10.
- [MOR 96] MORAES, F. et al. Pre-Layout performance prediction for automatic macro-cell synthesis. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 1996, Atlanta. **Proceedings...** Piscataway: IEEE, 1996. v.4., p. 814-817.
- [MUR 95] MURGAI, R.; BRAYTON, R. K.; VINCENTELLI, A. S. Decomposition of logic functions for minimum transition activity. In: THE EUROPEAN DESIGN AND TEST CONFERENCE, 1995, Paris. **Proceedings...** Los Alamitos: IEEE, 1995.
- [MUS 96] MUSOLL, E.; CORTADELLA, J. Optimizing CMOS circuits for low power using transistor reordering. In: THE EUROPEAN DESIGN AND TEST CONFERENCE, 1996, Paris. **Proceedings...** Los Alamitos: IEEE, 1996.
- [OBE 88] OBERMEIER, F. W.; KATZ, R. H. An electrical optimizer that considers physical layout. In: ACM/IEEE DESIGN AUTOMATION

- CONFERENCE, 25., 1988, Anaheim, Us. **Proceedings...** Los Alamitos: IEEE, 1988. p. 453-459.
- [PIG 84] FIGUET, Christian et al. A metal-oriented layout structure for CMOS logic. **IEEE Journal of Solid-State Circuits**, New York, v. SC-19, n. 3, p. 425-436, June 1984.
- [PRE 88] LORENZETTI, Michael J.; BAEDER, D. Scott. Routing. In: PREAS, B. T. (Ed.). **Physical Design Automation of VLSI Systems**. Menlo Park: Benjamin/Cummings, 1988. p.157-210.
- [REA 85] READ, John W. **Gate arrays: design techniques and applications**. New York: McGraw-Hill, 1985. 349 p.
- [REI 83] REIS, Ricardo A. L. **TESS: evaluateur topologique predictif pour la génération automatique des plans de masse de circuits VLSI**. Grenoble: Institut Polytechnique de Grenoble, 1983. 353 p.
- [REI 87] REIS, R. A New Standard Cell CAD Methodology. In: IEEE CUSTOM INTEGRATED CIRCUITS CONFERENCE, Portland, Us. **Proceedings...** New York: IEEE, 1987
- [REI 88] REIS, Ricardo A. L.; GOMES, Rogério; LUBASZEWSKI, Marcelo S. An efficient design methodology for standard cell circuits. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 1988, Helsinki. **Proceedings...** Piscataway: IEEE, 1988. v. 2, p. 1213-1216.
- [REI 93a] REIS, André et al. **Formas de implementação de ASICS digitais..** Campinas: SBMICRO, 1993. 106p.
- [REI 93b] REIS, André I. **Geração de células transparentes**. Porto Alegre: CPGCC da UFRGS, 1993. 128p. Dissertação de Mestrado.
- [REI 95] REIS, A. et al. Extensive Use of CMOS Complex Gates with Terminal Suppressed BDDs. **Journal of the Brazilian Computer Society**, Rio de Janeiro, v. 2, n.2, p.-63-75, Nov. 1995.
- [SAI 95a] SAIT, Sadiq M.; YOUSSEF, Habib. Layout generation. **VLSI physical design automation - theory and practice**. Piscataway: IEEE Press, 1995. 426p. p. 333-380.
- [SAI 95b] SAIT, Sadiq M.; YOUSSEF, Habib. Layout editors and compaction. **VLSI physical design automation - theory and practice**. Piscataway: IEEE Press, 1995. 426p. p. 381-407.
- [SCH 88] SCHIELE, W. L. Compaction with incremental over-constraint resolution. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 25., Anaheim, Us. **Proceedings...** Los Alamitos: IEEE, 1988. p. 390-395.

- [SHE 93] SHERWANI, Naveed A. Compaction. **Algorithms for VLSI physical design automation**. Norwell: Kluwer Academic Publishers, 1993. 488p. p. 383-407.
- [SHE 94] SHEW, Paul-Waie et al. Efficient algorithms for two and three-layer over-the-cell channel routing. In: ISCAS, 1994. **Proceedings...** New York: IEEE, 1994. p. 183-186.
- [SIN 90] SINGH, Uminder; CHEN, C. Y. Roger. A transistor reordering technique for gate matrix layout. In: DESIGN AUTOMATION CONFERENCE, 27., 1990, Orlando. **Proceedings...** New York: IEEE, 1990. p. 462-467.
- [SOT 94] SOTO, Fernando M. A. **SYLC e MRS: compactador e montador hierárquico simbólicos para o sistema CHARRUA**. Porto Alegre: CPGCC da UFRGS, 1994. 114p. Dissertação de Mestrado.
- [UEH 81] UEHARA, T.; vanCLEEMPUT, W. M. Optimal layout of CMOS functional arrays. **IEEE Transactions on Computers**, New York, v. C-30, n. 5, p. 305-312, May 1981.
- [VEL 96] VELASCO, A. J. et al. An efficient algorithm for optimal pairing and chaining in layout generation. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 1996, Atlanta. **Proceedings...** Piscataway: IEEE, 1996. v. 4., p. 775-778.
- [WIM 87] WIMER, Shumel; PINTER, Ron Y.; FELDMAN, Jack A. Optimal chaining of CMOS transistors in a functional cell. **IEEE Transactions on Computer Aided-Design**, New York, v. CAD-6, n. 5, p. 795-801, Sept. 1987.

Informática



UFRGS

CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Síntese Automática de Células CMOS

por

Marcus Kindel

Dissertação apresentada aos Senhores:

Prof. Dr. Fernando Gehm Moraes (PUC-RS)

Prof. Dr. Marcelo Soares Lubaszewski (IEE/UFRGS)

Prof. Tiaraju Vasconcellos Wagner (IEE/UFRGS)

Prof. Dr. Ricardo Pezzuol Jacobi

Vista e permitida a impressão.

Porto Alegre, 30/05/2000

Prof. Dr. Ricardo Augusto da Luz Reis

Orientador