

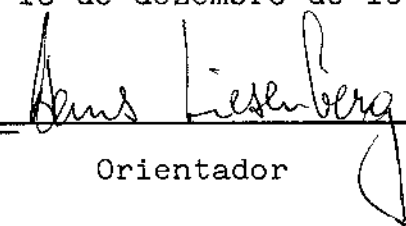
TÍTULO DA TESE

Ferramenta Automática de Posicionamento de Células para Projeto de Circuitos Integrados.

Este exemplar corresponde a redação final da tese devidamente corrigida pelo Sr. **EDUARDO MANOEL ARAÚJO**, e aprovada pela Comissão Julgadora.

Campinas, 18 de dezembro de 1987

Prof. Dr. \_\_\_\_\_

  
Orientador

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para obtenção do Título de Mestre em **Ciência da Computação**.

EDUARDO MANOEL ARAUJO

FERRAMENTA AUTOMÁTICA DE POSICIONAMENTO DE CÉLULAS  
PARA PROJETO DE CIRCUITOS INTEGRADOS

Tese apresentada para obtenção do título de Mestre em Ciência da Computação pelo Instituto de Matemática Estatística e Ciência da Computação da Universidade Estadual de Campinas.

Campinas  
1987

## DEDICATÓRIA

Aos melhores pais deste mundo Elci e Napoleão,  
a minha querida esposa Elizabeth e aos  
pequeninos Otávio Cesar e Henrique Daniel.

## AGRADECIMENTOS

Agradeço a Deus pela graça da vida, aos meus pais pelo seu amor e orientação na minha educação e formação; a minha esposa pelo seu amor, paciência e incentivo a esta realização; e a meu avô Maneco pelo seu apoio espiritual sempre presente nas horas difíceis.

Agradeço, em especial, ao Professor, Orientador e Amigo Hans Kurt Edmund Liesenberg pelo seu apoio, dedicação e determinação. Agradeço também ao suporte financeiro que me foi dado pela Companhia Paranaense de Energia (COPEL); pelo Conselho Nacional de Pesquisas (CNPq) e pela Universidade Estadual de Campinas (UNICAMP), sem o qual não seria possível a realização deste trabalho.

## SUMÁRIO

1.0	PROJETO DE CIRCUITOS INTEGRADOS . . . . .	1
1.1	Introdução . . . . .	1
1.2	Abordagens de projeto . . . . .	5
1.3	Compilador de silício . . . . .	9
1.4	O problema do traçado . . . . .	13
1.5	Escopo da tese . . . . .	15
2.0	POSICIONAMENTO . . . . .	16
2.1	Classificação e técnicas de posicionamento . . . . .	16
2.2	O posicionador no projeto de circuitos integrados . . . . .	21
2.3	Estrutura da ferramenta . . . . .	23
2.4	Representação de posicionamento relativo . . . . .	26
3.0	O GERADOR DE POSICIONAMENTO AUTOMÁTICO . . . . .	35
3.1	Descrição estrutural de células . . . . .	35
3.2	Reconhecimento de regularidades . . . . .	37
3.3	Geração do posicionamento inicial . . . . .	40
3.4	Melhoramento do posicionamento inicial . . . . .	46
4.0	CONCLUSÃO . . . . .	50
4.1	Resumo do trabalho . . . . .	50
4.2	Contribuições principais . . . . .	55

4.3	Futuras extensões . . . . .	56
5.0	BIBLIOGRAFIA . . . . .	59
<b>Apêndice A. ESPECIFICAÇÃO SINTÁTICA DA LINGUAGEM DE ENTRADA .</b>		<b>66</b>
A.1	Introdução . . . . .	66
A.2	Descritor de célula . . . . .	66
A.3	Cabeçalho do descritor de célula . . . . .	67
A.4	Especificação dos terminais . . . . .	68
A.5	Corpo do descritor de célula . . . . .	70
A.6	Corpo composto . . . . .	71
A.7	Posicionamento de exemplares de subcélulas . . . . .	72
A.8	Especificação de interligações . . . . .	76
<b>Apêndice B. EXEMPLOS DE ENTRADA E SAÍDA . . . . .</b>		<b>78</b>
B.1	Exemplo 1 . . . . .	78
B.1.1	Entrada . . . . .	78
B.1.2	Saída . . . . .	80
B.2	Exemplo 2 . . . . .	84
B.2.1	Entrada . . . . .	84
B.2.2	Saída . . . . .	89
	<b>Índice Remissivo . . . . .</b>	<b>93</b>

## LISTA DE ILUSTRAÇÕES

Figura 1. Níveis de abstração no projeto de circuitos integrados . . . . .	2
Figura 2. Sistemas baseados em células . . . . .	7
Figura 3. Estrutura de um dispositivo lógico programável . . . . .	9
Figura 4. Processo de compilação de silício . . . . .	11
Figura 5. Grafos de posição de canal . . . . .	27
Figura 6. Grafos de posicionamento relativo . . . . .	28
Figura 7. Operações de redução de adjacência . . . . .	30
Figura 8. Redução do grafo de posicionamento relativo . . . . .	32
Figura 9. Grafo de posicionamento relativo não redutível . . . . .	33
Figura 10. Desdobramento de grafo não redutível . . . . .	34
Figura 11. Grafo de interseção de canais . . . . .	40
Figura 12. Tipos de desdobramento de canal . . . . .	42
Figura 13. Desdobramento de canal . . . . .	43
Figura 14. Algoritmo de determinação da largura de um posicionamento . . . . .	45
Figura 15. Algoritmo de melhoramento do posicionamento inicial . . . . .	49
Figura 16. Quadro de tempos de execução . . . . .	53
Figura 17. Comparativo de razão de aparência e taxa de ocupação . . . . .	54
Figura 18. Posicionamento com razão de aparência 1.0 . . . . .	81
Figura 19. Posicionamento com razão de aparência 0.8 . . . . .	82
Figura 20. Posicionamento com razão de aparência 0.6 . . . . .	83

Figura 21. Posicionamento com razão de aparência 1.0	. . . .	90
Figura 22. Posicionamento com razão de aparência 0.8	. . . .	91
Figura 23. Posicionamento com razão de aparência 0.6	. . . .	92



## SINOPSE

Devido a crescente complexidade no processo de projeto de circuitos integrados existe uma tendência natural de automatização deste processo. Uma das fases do processo de projeto de circuitos integrados é o seu traçado, que consiste no posicionamento das subcélulas que compõem o circuito e posterior roteamento das interligações entre estas subcélulas. Na etapa de posicionamento é alocado um espaço no plano de posicionamento para cada uma destas subcélulas. Na fase de roteamento é estabelecido um caminho e alocado um espaço para as interligações através dos canais (áreas não ocupadas pelas subcélulas).

O posicionamento das subcélulas pode ser realizado de forma absoluta ou relativa. No posicionamento absoluto os canais tem dimensões fixas. Se na fase de roteamento a largura de algum dos canais for insuficiente para permitir a alocação do espaço para as interligações, então o posicionamento deve ser refeito prevendo o alargamento dos canais. No posicionamento relativo existe a flexibilidade de ajuste da largura dos canais durante o roteamento, uma vez que só são definidas relações de adjacência entre as subcélulas.

A ferramenta de posicionamento automático aqui descrita utiliza uma linguagem de entrada que permite especificar as subcélulas que compõem um circuito e a configuração de suas interligações. A ferramenta está dividida em três fases: reconhecimento de regularidades, posicionamento inicial e melhoramento do posicionamento inicial.

Durante a fase de reconhecimento de regularidades são identificados e agrupados os conjuntos de subcélulas que possuem uma estrutura regular de interconexão, para a qual se conhece um posicionamento eficiente.

Para obtenção do posicionamento inicial é utilizada a técnica de crescimento epitaxial ou construtivo. Nesta técnica as subcélulas são incorporadas uma a uma no plano de posicionamento obedecendo a um critério de máxima conexidade com o conjunto das subcélulas já posicionadas.

Na fase de melhoramento do posicionamento inicial são realizadas trocas de pares de subcélulas. Para restringir o número de trocas é delimitado inicialmente para cada subcélula, a vizinhança do ponto ideal para o seu posicionamento. Os candidatos para troca com a subcélula em questão são as subcélulas nesta vizinhança. Dentre as trocas realizadas são aceitas aquelas que efetivamente melhoram o posicionamento atual.

O resultado final do posicionamento automático é uma expressão de posicionamento relativo envolvendo as subcélulas que compõem o circuito especificado.

## ABSTRACT

There is a natural tendency to automate the process of integrated circuit design due to its growing complexity. One of the integrated circuit design phases is the layout generation, which is carried out by first positioning the circuit's subcells and later on routing the interconnections between them. During the positioning stage a space for each of the circuit's subcells is allocated on the floorplan. While during the routing stage, a path through channels is established for each interconnection and space for the tracks is allocated.

The positioning of subcells can be accomplished in two ways: absolute or relative. In absolute positionings channels have fixed dimensions. If, at the routing stage, the width of some channel is found to be too narrow to allow for the allocation of space to the interconnections, then the positioning must be redone. In the relative placement, only adjacency relations between cells are defined, so there is flexibility to accept varying channel dimension demands.

The automatic placement tool described here has an input language which allows one to specify the subcells of a circuit and its interconnection configuration. The tool executes in three phases:

regularity recognition, initial placement and improvement of the initial placement.

During the regularity recognition phase, subcells groups that have a regular interconnection structure, for which an efficient placement is known, are identified and positioned.

The constructive or epitaxial growth technique is applied to obtain the initial placement. With this technique the subcells are incorporated to the floorplan, one at a time, following the maximum connectivity criteria with the set of already positioned subcells.

During the improvement of the initial placement, interchanges of pairs of subcells are tried. For each subcell, its ideal position and a neighborhood of this point are determined. To restrict the number of trial interchanges, the candidates to be paired with each subcell are the subcells in that neighborhood. Among the trial interchanges, those which effectively improve the current placement are chosen.

The final result of the automatic placement is a relative placement expression of all circuit's subcells.

## 1.0 PROJETO DE CIRCUITOS INTEGRADOS

### 1.1 Introdução

Com o atual estado da tecnologia de fabricação de circuitos integrados já é possível colocar em uma única pastilha centenas de milhares de dispositivos elementares. A expectativa é que este número de portas lógicas por circuito continue dobrando a cada um ou dois anos. Segundo C. H. Sequin [Seq83], o maior obstáculo na construção de sistemas com uma maior escala de integração é a barreira da complexidade. Para que seja possível explorar completamente o potencial da integração em altíssima escala (VLSI) é necessário que o projetista tenha em mãos uma metodologia de desenvolvimento de projeto e um conjunto de ferramentas que o auxiliem em tal cometimento.

Durante o projeto de um circuito integrado o mesmo passa por diversas fases de refinamento nas quais são adicionados mais e mais detalhes [Lie86]. Em cada fase o circuito é manipulado em um certo nível de abstração onde determinados aspectos são ressaltados. Em uma escala decrescente de abstração é possível enumerar os seguintes níveis usados mais comumente: comportamental, funcional, lógico, elétrico e geométrico. No nível comportamental é feita uma

especificação dos requisitos do sistema, o que ele deve fazer, o tempo de resposta desejado e restrições diversas tais como tamanho e consumo. Na Figura 1 na página 2 é apresentado um esquema mostrando a hierarquia definida pelos diversos níveis de abstração.

---

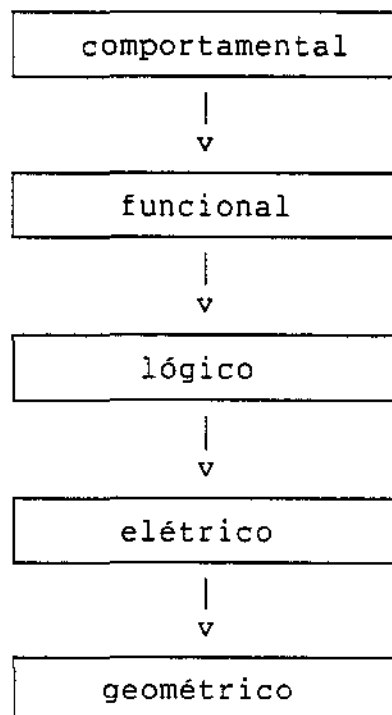


Figura 1. Níveis de abstração no projeto de circuitos integrados

---

No nível de abstração seguinte, o funcional, são definidas estratégias para solução do problema. São definidos os componentes

de "hardware" e "software" do projeto. Neste nível o circuito é usualmente descrito através de uma linguagem descritiva de hardware.

No nível lógico as interações entre os diversos blocos, que compõem o sistema em projeto, são tratadas em termos de sinais que possuem essencialmente dois estados (verdadeiro e falso) com possíveis estados intermediários indicando transições entre estes estados. No nível elétrico estes mesmos sinais são representados através de grandezas físicas tais como tensão ou corrente.

No nível geométrico, de menor abstração, o circuito é representado por figuras geométricas que definem as dimensões e o posicionamento dos dispositivos elementares e suas interligações. Esta descrição geométrica é utilizada para confeccionar as máscaras necessárias durante o processo fotolitográfico utilizado na fabricação do circuito. As dimensões de cada figura são determinadas de acordo com o funcionamento desejado do dispositivo e da tecnologia em que está sendo implementado.

Para que o projeto de um circuito possa ser corretamente implementado é necessário que a sua especificação geométrica satisfaça determinadas regras de projeto e de composição. As regras de projeto existem devido a limitações tecnológicas dos processos de fabricação de circuitos integrados. Elas especificam parâmetros tais como: largura mínima exequível em determinada



camada, espaçamento mínimo entre as figuras geométricas e sobreposição mínima de objetos em camadas diferentes. As regras de composição especificam as restrições a observar, tanto a nível elétrico como a nível geométrico, para que as células que compõem o circuito possam ser posicionadas e interligadas sem exigir o conhecimento de suas organizações internas.

Dentro deste contexto será feita uma revisão das abordagens mais comuns de projeto. Uma das linhas de pesquisa em automação do processo de projeto de circuitos é conhecida como compilação de silício. O processo de compilação de silício consiste de se obter a partir da especificação funcional de um circuito a especificação geométrica das máscaras necessárias a sua fabricação. Uma das fases deste processo é o traçado automático, onde o problema consiste em posicionar e interligar uma série de células primitivas que implementam as funções básicas. Na etapa de posicionamento um conjunto de figuras geométricas retangulares devem ser dispostas sem superposição, procurando-se minimizar a área ocupada pelo circuito e o comprimento total das interligações. Na etapa de roteamento é efetivamente definida a trajetória das interligações e é alocado espaço para as mesmas.

Neste capítulo serão detalhadas as principais técnicas de posicionamento de células enquadrando as mesmas em uma classificação que permitirá a definição mais precisa do escopo deste trabalho.

## 1.2 Abordagens de projeto

Podemos dividir as abordagens de projeto de circuitos integrados em dois grandes grupos: projeto personalizado e projeto semi-personalizado. A abordagem de projeto personalizado é um processo praticamente artesanal em que cada dispositivo elementar é dimensionado, posicionado e interligado manualmente. A abordagem de projeto semi-personalizado se baseia em células pré-projetadas e possivelmente já pré-fundidas sobre o substrato semicondutor.

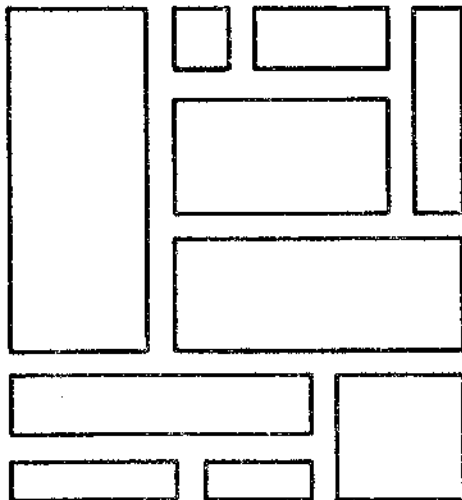
O projeto personalizado de modo geral leva à produção de circuitos mais compactos e com uma velocidade de operação maior que a normalmente conseguida em projetos semi-personalizados. Em contrapartida o esforço de projeto e o tempo de desenvolvimento do primeiro são bem maiores. Nesta abordagem a experiência do projetista é de uma importância decisiva na eficiência e compacidade do circuito projetado.

A abordagem de projeto semi-personalizada pode ser dividida em quatro grupos [Den83]: i) matriz de portas lógicas ; ii) sistemas baseados em células ; iii) dispositivos lógicos programáveis e iv) matriz de componentes analógicos. A matriz de portas lógicas consiste de uma pastilha com portas lógicas pré-fundidas distribuídas em um arranjo matricial. Neste caso o projetista deve se preocupar unicamente com o problema da interligação destas

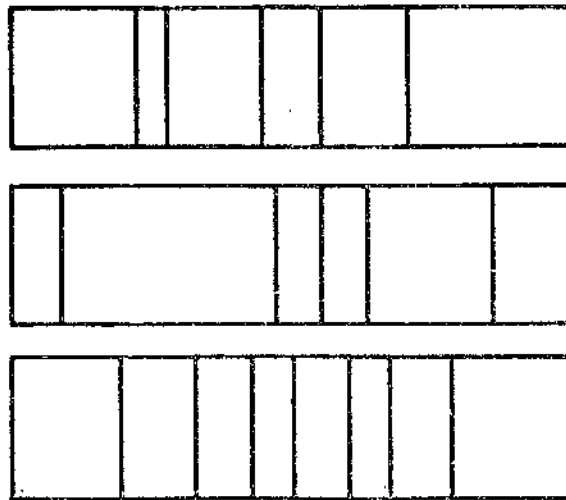
portas de tal forma a obter um circuito com o comportamento desejado. É possível nesta abordagem a utilização de uma biblioteca de módulos onde cada módulo é descrito através do número de portas lógicas adjacentes necessárias para a sua confecção e a configuração de suas interligações internas.

Uma célula define um sistema eletrônico. Sendo geralmente encapsulada por um retângulo delimitador sobre o qual são posicionados os terminais (pontos de conexão dos sinais internos da célula com os sinais externos). O retângulo delimitador e os pontos terminais estabelecem a interface entre o mundo exterior e o sistema encapsulado. As células podem ser de dois tipos: primitivas ou compostas. Células primitivas são definidas em termos de dispositivos elementares e suas interligações e são geralmente projetadas utilizando ferramentas computadorizadas de auxílio a projeto. As especificações de células primitivas são normalmente agregadas em uma biblioteca. Células compostas são aquelas que estão definidas a partir de células primitivas e/ou compostas.

Nos sistemas baseados em células, a partir de células primitivas existentes em biblioteca, o projetista concebe o circuito desejado e especifica as células de que necessita e como elas deverão ser interligadas. Na Figura 2 na página 7 são mostrados dois exemplos de projeto utilizando sistemas baseados em células.



células gerais



células de altura padrão

Figura 2. Sistemas baseados em células

---

Uma técnica bastante difundida dentro desta abordagem é a utilização de células primitivas com altura padrão projetadas para serem justapostas horizontalmente formando um barramento para sinais de alimentação e possivelmente de relógio. As outras interligações são traçadas no espaço entre fileiras de componentes. O custo maior nesta abordagem está no desperdício de área que costuma acontecer, apesar desta abordagem ser um bom compromisso entre o projeto personalizado e a utilização de matrizes de portas lógicas.

Os dispositivos lógicos programáveis representam uma família de componentes cujos membros são caracterizados por um conjunto de funções lógicas e uma topologia de interconexões inerente a sua estrutura matricial. Eles são basicamente capazes de implementar funções lógicas combinacionais. Podem ser obtidos circuitos sequenciais através da realimentação de algumas entradas por certas saídas. A estrutura geral destes dispositivos é mostrada na Figura 3 na página 9.

Nesta classe de dispositivos a limitação tende ser a rigidez de seu modo de operação, o número de entradas e saídas e o número de termos de produto que compõem as funções. Outro ponto bastante ponderado é a facilidade com que estes componentes lógicos programáveis podem ser gerados de maneira automática.

As matrizes de componentes analógicos são destinadas à implementação de funções puramente analógicas. Células analógicas como amplificadores operacionais, osciladores e comparadores podem ser montados rapidamente através da interligação apropriada dos diversos componentes primitivos.

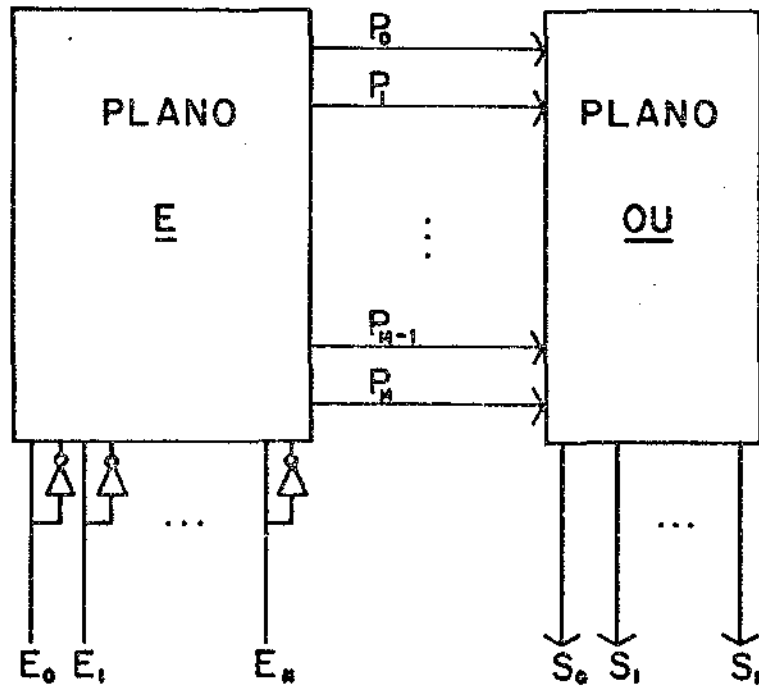


Figura 3. Estrutura de um dispositivo lógico programável

### 1.3 Compilador de silício

Um compilador de silício é uma ferramenta que transforma uma especificação de um circuito num nível mais alto de abstração em uma especificação de nível mais baixo de maneira automática. A entrada para tal ferramenta é usualmente uma descrição funcional do circuito podendo conter algumas diretivas para a orientação do processo de compilação. A partir desta entrada é gerada

automaticamente a especificação geométrica do circuito a partir da qual é derivado o conjunto de máscaras utilizado na fabricação do circuito [Hir82]. A especificação geométrica nada mais é do que o dimensionamento e posicionamento dos dispositivos elementares e de suas interligações nas diversas camadas inerentes ao processo de fabricação utilizado.

O propósito das diretivas para este tipo de ferramenta é o de dar ao projetista um certo grau de controle sobre o processo de traduções sucessivas efetuadas pelo compilador como, por exemplo, a distribuição dos dispositivos gerados, permitindo assim que a sua experiência e o seu conhecimento global do projeto possa ser melhor aproveitado no direcionamento do processo de compilação.

Um processo típico de compilação de silício atual é delineado na Figura 4 na página 11 e é composto de cinco passos [Den83]. Um exemplo clássico deste processo é o compilador FIRST [Ber83].

A linguagem de entrada suporta alguma forma de descrição funcional e também um subconjunto de construções sintáticas usuais de linguagens de programação, e é totalmente independente de tecnologia. Estas características contribuem para reduzir a quantidade de código necessária à definição de um sistema.

Num primeiro passo é produzida, a partir da especificação funcional fornecida pelo projetista, uma lista de células

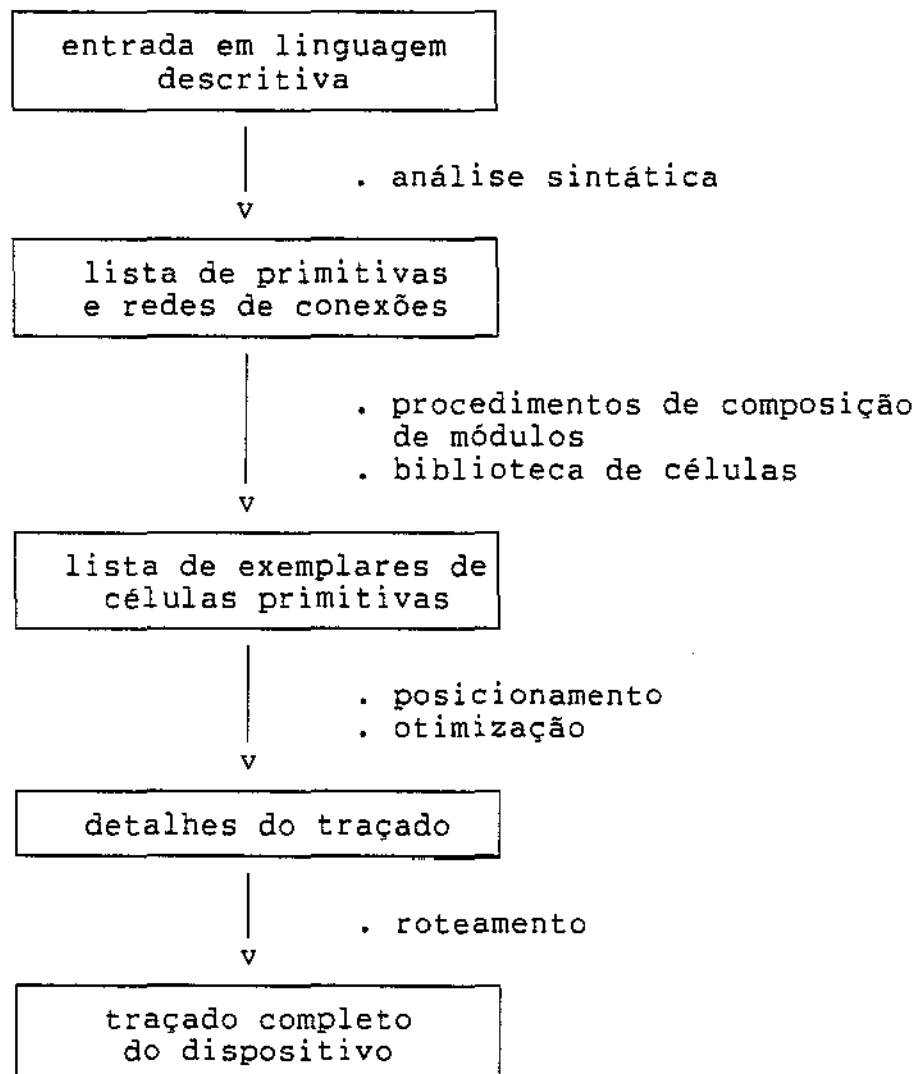


Figura 4. Processo de compilação de silício

---

primitivas e suas interconexões (especificação estrutural). Estas células primitivas representam funções de baixo nível do compilador e têm a sua representação física detalhada em uma



biblioteca. Como exemplo destas células primitivas no compilador FIRST temos: multiplicar, somar, atrasar.

A seguir são criados exemplares das células primitivas definidas na lista criada anteriormente. Para formar o circuito completo é preciso posicionar e interligar estes exemplares. Tanto a biblioteca de células como o roteamento são certamente dependentes da tecnologia.

O compilador de silício é potencialmente uma ferramenta poderosa na abordagem semi-personalizada de projeto. Em primeiro lugar, o usuário não precisa ter conhecimentos aprofundados da tecnologia a ser usada para fabricar o circuito em projeto, uma vez que o compilador exerce a função do especialista. No caso do compilador FIRST por exemplo, o compilador é extremamente veloz, requerendo tipicamente alguns segundos de UCP por circuito compilado. Isto por sua vez estimula a experimentação com alternativas durante o projeto. No entanto a maior das vantagens realmente é a característica de correção por construção dos circuitos compilados em relação às regras de projeto e de composição, visto que não é permitido ao projetista intervir diretamente no processo de compilação.

Uma revisão dos compiladores de silício atualmente implementados bem como comentários sobre a filosofia de cada ferramenta é apresentada em [Lie85].

#### 1.4 O problema do traçado

O traçado consiste em posicionar um conjunto de células primitivas (posicionamento) e interligá-las como definido na especificação estrutural do circuito (roteamento).

O posicionamento consiste basicamente na disposição de todas as células que compõem o circuito no plano, sem superposição, de tal forma a permitir a geração de um traçado bastante denso procurando deixar espaço suficiente para as interligações e minimizar o comprimento total das mesmas. É usual, devido a complexidade já inerente ao problema do posicionamento, o formato das células ser retangular sendo considerado, para o problema do traçado, apenas o contorno de cada célula com a posição dos terminais para estabelecer as conexões externas. Em [Hud84] são apresentados alguns algoritmos para posicionamento de polígonos retilíneos.

Na fase de roteamento é estabelecido um conjunto de segmentos de conexão que estabelecem as interligações para cada sinal (conjunto de pontos terminais eletricamente equivalentes). Deve-se levar em conta, nesta fase, a largura das conexões e a camada de roteamento. No caso de roteamento de conexões de um sinal em mais de uma camada é feito um corte de contato preenchido com metal para interligar as camadas. Na tecnologia nMOS, cujo processo é descrito em [Mea80], as conexões longas são realizadas

preferencialmente em metal, sendo o polissilício utilizado normalmente para evitar o cruzamento de duas conexões em metal. Esta estratégia se baseia nas características elétricas mais favoráveis do metal.

Para matrizes de portas lógicas, o posicionamento consiste no mapeamento de cada porta lógica do projeto para uma porta lógica pré-fundida no substrato. O roteamento é então realizado nas áreas livres (canais) existente entre as portas lógicas.

Todo este processo deve obedecer a um conjunto de regras de projeto que especificam basicamente distâncias mínimas entre objetos que definem os dispositivos elementares e suas interligações, bem como as larguras mínimas exequíveis. Estas regras refletem as limitações do processo de fabricação e estão intimamente relacionadas com a tecnologia sendo utilizada.

No final do processo de traçado é produzida a descrição de um conjunto de foto-máscaras que contém uma descrição geométrica de cada uma das camadas que será produzida durante a fabricação do circuito. Uma visão geral dos processos de fabricação de circuitos integrados é apresentada em [Mur82].

Uma técnica usualmente empregada é a especificação da geometria em termos de unidades  $\lambda$ , onde  $\lambda$  representa a resolução mínima executável do processo de fabricação. Isto permite que para

avanços da tecnologia, dentro de certos limites, uma mesma especificação geométrica possa ser reaproveitada simplesmente substituindo-se o valor da unidade  $\lambda$  pelo valor da nova resolução mínima do novo processo de fabricação. Os limites do reaproveitamento são determinados pela variação não linear entre a redução de dimensões e as características elétricas dos dispositivos elementares. Este escalamento de projeto pode ser ineficiente em área, no caso de haverem variações diferentes de dimensões mínimas de objetos em camadas distintas. Neste caso a definição das regras de projeto é feita de maneira conservadora uma vez que o escalonamento deve ser uniforme.

### 1.5 Escopo da tese

Este trabalho visa descrever uma ferramenta automática de posicionamento de células adequada para projetos de circuitos de altíssima escala de integração (VLSI). A ferramenta recebe como entrada a descrição estrutural de um circuito (célula) e a partir dela produz um posicionamento relativo para as subcélulas que compõem tal descrição. Esta ferramenta pode fazer parte de um módulo de traçado automático de um compilador de silício ou ser utilizada de forma isolada independente de qualquer integração.

## 2.0 POSICIONAMENTO

### 2.1 Classificação e técnicas de posicionamento

O posicionamento pode ser realizado de duas formas: absoluto ou relativo. No posicionamento absoluto cada célula, e conseqüentemente os seus terminais, tem uma posição fixa na área reservada ao circuito em termos de um sistema de coordenadas. Neste caso, se na fase de roteamento, na qual é alocado o espaço para as interligações, não se conseguir a realização de todas as ligações entre células, simplesmente abandona-se o posicionamento atual e o roteamento parcial já feito e produz-se um novo posicionamento. Este processo é iterativo até que se obtenha um posicionamento que possibilite traçar todas as interligações.

O posicionamento relativo define a topologia do traçado simplesmente através de relações de adjacência entre as células. A partir de um posicionamento relativo um roteador pode aumentar ou diminuir a área de roteamento entre as células, ditas canais, desde que preserve as relações de adjacência especificadas. Isto evita as iterações necessárias no caso de um posicionamento absoluto. Ao final do processo de roteamento é então atribuída uma posição absoluta a cada célula.

As ferramentas de posicionamento podem ser divididas em dois grupos [Lie85]: geradores de posicionamento inicial e melhoradores de posicionamentos. Os geradores de posicionamento derivam um posicionamento a partir da especificação do circuito, procurando já otimizar a área total do posicionamento e o comprimento total das interligações. Normalmente é conseguido apenas um ótimo local. Os melhoradores de posicionamento partem de um posicionamento inicial e tentam escapar do ótimo local conseguido na fase anterior em busca dos mesmos objetivos.

Os geradores de posicionamento inicial podem ser divididos em três classes: randômicos, por particionamento e construtivos. Nos geradores de posicionamento randômicos as células são posicionadas de maneira aleatória.

Na segunda classe de geradores de posicionamento, por particionamento, um conjunto de células é sucessivamente particionado em dois subconjuntos de cardinalidade pré-definida ou que ocupem aproximadamente a mesma área. O algoritmo também procura minimizar as interações entre os dois subconjuntos, e por esta razão é chamado de algoritmo de corte mínimo.

A terceira classe, dos geradores de posicionamento inicial construtivos [Han76], ou por crescimento epitaxial como é também conhecida, seleciona as células uma por vez, baseando-se na avaliação de alguma função que expresse, por exemplo, a conectividade

das células não posicionadas em relação às já posicionadas. A escolha do local de posicionamento da célula pode ainda levar em conta a distância da mesma às células já posicionadas com as quais interage. Esta classe de geradores de posicionamento pode também partir inicialmente de um subconjunto de células já posicionadas.

Os melhoradores de posicionamento podem ser divididos em seis classes [Han76], do ponto de vista algorítmico: troca de pares padrão, troca de pares na vizinhança, algoritmo de Steinberg, força dirigida relaxada, troca por força dirigida e pares de força dirigida relaxados.

A troca de pares padrão, partindo do posicionamento inicial, consiste em trocas tentativas sucessivas com cada par de células e se esta troca melhorar o posicionamento, então ela é efetivada. Para se comparar dois posicionamentos é computado um valor de mérito para cada um deles. Um valor de mérito bem simples pode ser visto neste caso como uma equação que computa a relação entre algumas variáveis, tais como: área do posicionamento e comprimento previsto das interconexões. Nesta classe de melhoradores, todos os  $n.(n-1)/2$  pares de células de um posicionamento têm uma tentativa de troca e uma avaliação do valor de mérito.

A troca de pares na vizinhança é conceitualmente idêntica a troca de pares padrão, somente que para cada célula, o seu par para

troca só é considerado entre as células vizinhas. A distância que define a vizinhança é neste caso um parâmetro do algoritmo.

O algoritmo de Steinberg aplica iterativamente um algoritmo linear a conjuntos de células independentes em relação aos sinais. Um conjunto de células independente é composto de células que não tem nenhum sinal que as interligue. Uma iteração é definida como segue: é escolhido um conjunto de células independentes, estas células são removidas do posicionamento e a seguir reposicionadas de tal forma a minimizar o custo global (distância das conexões no caso) de posicionamento destas células. O custo de posicionar uma célula deste conjunto independe da localização das outras células do conjunto de células independente. Após iterar em todos os conjuntos independentes gerados, este ciclo termina e o próximo ciclo se inicia. O tamanho e número de conjuntos independentes, bem como o número de ciclos são parâmetros do algoritmo.

Nos algoritmos de força dirigida relaxada, é computado um vetor de força para cada célula, que leva em conta a conexidade e a distância vetorial entre as células. Usando este vetor de força é possível computar a localização alvo, que corresponde ao ponto onde a soma das forças na célula considerada é nula. Para cada célula é computada uma função "Q" que exprime o número de conjuntos de sinais a que a célula pertence. As células são escolhidas para relaxação em ordem decrescente do valor de "Q".



O algoritmo de força dirigida relaxada usa uma técnica de sequencialmente "relaxar" (reduzir as forças sobre) cada célula. A célula escolhida para relaxação é colocada na sua localização alvo e bloqueada, se esta localização estiver livre, e uma nova célula é escolhida para relaxação. Se a localização alvo estiver ocupada e a célula não bloqueada, aquela célula é retirada e é a próxima a ser relaxada. Isto pode desencadear uma sucessão de reposicionamentos. Toda sequência de células envolvidas numa sucessão de reposicionamento, inclusive aquela que desencadeou o processo, é chamado de cordão. Se a localização alvo da célula escolhida para relaxação estiver ocupada por uma célula bloqueada, a célula escolhida deve ser posicionada na posição desbloqueada que resultar no menor valor do vetor de força sobre a célula. Os cordões terminam somente se uma célula relaxada ocupar uma posição vazia. Quando um cordão termina, o custo deste novo posicionamento é calculado, e se for menor que o custo original então o cordão é aceito e todas as células do cordão são bloqueadas pelo resto do ciclo. O ciclo termina quando todas as células começam um cordão ou estão em um cordão já aceito. As células são escolhidas em ordem decrescente do número de sinais a que cada célula pertence.

Na troca por força dirigida, como no método da força dirigida relaxada é calculado o vetor de força da célula escolhida para relaxação, dita célula primária. É então feita uma troca tentativa com um dos seus três vizinhos na orientação do seu vetor de força (dois adjacentes diretos definidos pelas componentes horizontal e

vertical do vetor e um na diagonal definido pela direção do vetor). Se uma destas trocas tentativas resultar em uma redução do valor de mérito do posicionamento, a troca é aceita e são feitas trocas tentativas com os novos vizinhos. Se nenhuma das trocas resultar em redução de custo, é escolhida uma nova célula para relaxação. Quando todas as células tiverem sido células primárias o ciclo termina.

Os melhoradores de posicionamento através de relaxação de pares de força dirigida também são baseados no conceito de vetores de força e localização alvo. Uma célula "A" é escolhida para relaxação. É então feita a troca tentativa com uma célula "B" na "epsilon"-vizinhança da localização alvo da célula "A" se a localização alvo da célula "B" estiver na "epsilon"-vizinhança da célula "A". Da mesma forma a troca é aceita se resultar em uma redução do valor de mérito do posicionamento. O ciclo termina quando todas as células tiverem sido escolhidas para relaxação.

## **2.2 O posicionador no projeto de circuitos integrados**

Um compilador de silício pode ser visto como um conjunto de ferramentas que, de forma automática traduz uma descrição estrutural ou funcional de um circuito em uma especificação

geométrica equivalente. Werner [Wer83a;Wer83b] divide este processo de compilação em duas partes. A primeira parte é responsável pela tradução da descrição de mais alto nível em uma descrição intermediária mais detalhada, indicando que células serão utilizadas e de que maneira deverão ser interligadas. É conveniente que esta descrição intermediária ainda seja conceitualmente independente do processo de fabricação. A segunda parte é composta das ferramentas que se utilizam de informações tecnológicas, como: geradores de traçado, simuladores elétricos, verificadores de regras de projeto e outras ferramentas desta natureza.

O gerador de traçado é o responsável pela tradução da descrição em código intermediário de um circuito em uma especificação geométrica equivalente. Este processo envolve praticamente duas fases: o posicionamento automático de tais células e posteriormente o roteamento automático de suas interligações. Assim a ferramenta automática de posicionamento de células que aqui é proposta pode ser vista dentro do processo de compilação de silício como parte do gerador de traçado.

Esta ferramenta automática de posicionamento também pode operar de maneira independente de qualquer integração com outras ferramentas, como parte do processo de projeto. A entrada neste caso é a descrição estrutural do circuito a ser implementado utilizando a gramática definida no "Apêndice A. ESPECIFICAÇÃO

SINTÁTICA DA LINGUAGEM DE ENTRADA" na página 66. A saída é um posicionamento relativo das células especificadas na descrição estrutural dada.

### 2.3 Estrutura da ferramenta

A ferramenta está logicamente dividida em três fases: reconhecimento de regularidades, geração do posicionamento inicial e melhoramento do posicionamento. No reconhecimento de regularidades do projeto, a estrutura de interconexões é examinada e aquelas células que apresentarem uma estrutura regular de interligação e estiverem fortemente conexas serão agrupadas de acordo com o padrão eficiente de posicionamento conhecido para esta estrutura regular. O objetivo é evitar que tais estruturas sejam posicionadas de uma forma dispersa no plano de posicionamento causando roteamentos longos e ineficientes. Um agrupamento resultante de uma estrutura regular é tratado nos próximos passos do posicionador como um bloco único, reduzindo desta forma o esforço computacional que seria necessário para processar, em cada fase do posicionamento automático, as células destes blocos de forma individual.

Na segunda fase, para a geração de um posicionamento inicial, a ferramenta utiliza o processo construtivo, podendo, se desejado, ser fornecido um posicionamento parcial inicial. Assim as células são incorporadas uma a uma na configuração em construção no plano de posicionamento, segundo um critério de conexidade.

A terceira fase corresponde ao melhoramento do posicionamento inicial. O processo aqui utilizado é o de troca de pares de células padrão. Entre cada troca é avaliado um valor de mérito que determina se a troca melhora o posicionamento original. Caso a troca melhore o posicionamento original ela é aceita.

No projeto de circuitos de altíssima escala de integração, devido ao grau de complexidade, é desejável, além da utilização de uma metodologia estruturada de projeto, visualizar o mesmo através de uma hierarquia de composição. Uma hierarquia de composição pode ser representada através de um grafo direcionado com um único vértice fonte e um ou mais vértices sorvedouro. Vértice fonte é aquele que tem pelo menos uma aresta dele emergente e nenhuma aresta nele incidente. Vértice sorvedouro é aquele que tem pelo menos uma aresta nele incidente e nenhuma aresta dele emergente. O vértice fonte representa o circuito em projeto. Os vértices sorvedouro representam células primitivas. Os demais vértices representam células compostas. Uma aresta direcionada de um vértice A para um vértice B indica que um ou mais exemplares da

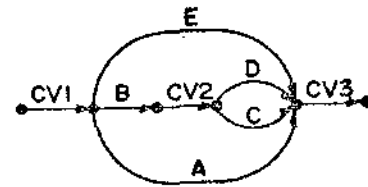
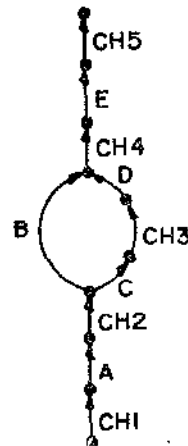
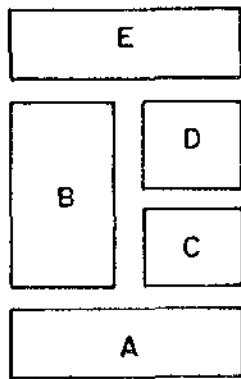
célula representada pelo vértice B fazem parte da composição da célula representada pelo vértice A.

Para projetos de circuitos integrados, cuja descrição seja dada através de uma hierarquia de composição, o gerador de posicionamento automático é invocado em cada vértice do grafo que representa tal hierarquia cuja célula associada ainda não tenha sido roteada. É conveniente que o processo de chamada do gerador de posicionamento automático seja de baixo para cima, isto é, partindo daquelas células na hierarquia cujas subcélulas que a compõem já estejam totalmente traçadas em direção ao vértice fonte do grafo associado à hierarquia de composição visto que as dimensões das subcélulas já estão bem definidas. Desta forma os exemplares de subcélulas que compõem determinada célula na hierarquia de composição seriam posicionados e em seguida seria executado o seu roteamento. Quando se tem o traçado do conjunto de exemplares de subcélulas que compõem uma célula na hierarquia, este traçado é congelado, sendo a sua composição tratada deste ponto em diante como uma célula única com suas dimensões finais obtidas ao final de seu roteamento.

## 2.4 Representação de posicionamento relativo

Preas e vanCleemput [Pre79] utiliza em seu algoritmo de posicionamento dois grafos direcionados e acíclicos. São os grafos de posição de canal horizontal e vertical. Um exemplo de tais grafos é dado na Figura 5 na página 27. Cada vértice nestes grafos representa o limite entre uma célula e um canal. Entenda-se célula como exemplar de célula neste contexto. As arestas representam a dimensão física de um canal ou de uma célula.

Para a representação de um posicionamento relativo os grafos de posição de canal horizontal e vertical podem ser simplificados [Lie84]. Primeiramente pode ser eliminada toda informação de dimensão. A próxima simplificação é fazer com que os vértices destes grafos representem canais e as arestas representem as células. Com tais simplificações aplicadas aos grafos da Figura 5 na página 27, obtem-se os grafos de posição de canal horizontal e vertical modificados, chamados de grafos de posicionamento relativo (ver Figura 6 na página 28). Se for adicionada a cada um destes grafos uma aresta, representando a área na qual o posicionamento está inserido, o grafo de posicionamento relativo horizontal se torna o dual do vertical. Consequentemente o posicionamento relativo pode ser representado através de um único grafo, caso seja suposta uma ordenação das arestas emergentes de cada vértice nestes dois grafos. Para representar um



a. posicionamento

b. horizontal

c. vertical

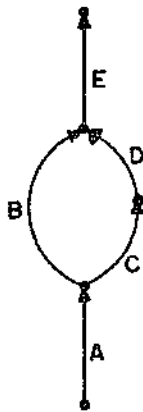
Figura 5. Grafos de posição de canal

posicionamento relativo escolheu-se então o grafo de posição de canal horizontal modificado.

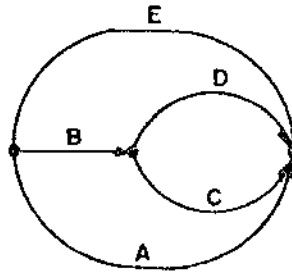
As propriedades do grafo de posicionamento relativo simplificado desta forma são as seguintes:

1. o grafo é planar, direcionado e acíclico;
2. cada vértice representa um canal horizontal;





a.vertical



b.horizontal

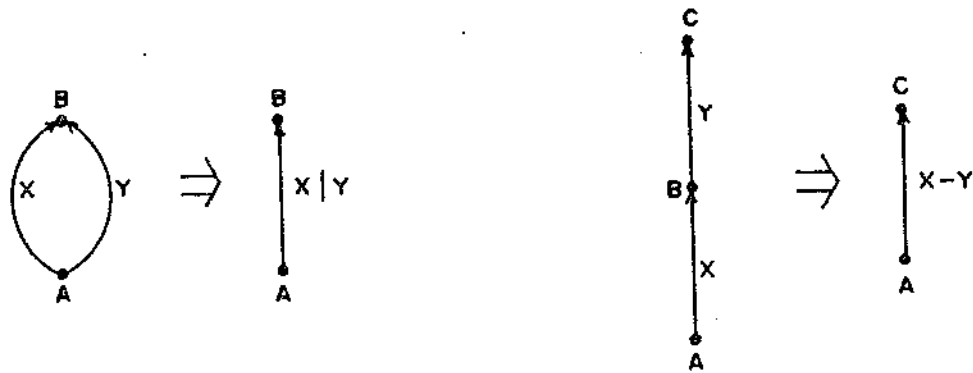
Figura 6. Grafos de posicionamento relativo

3. cada aresta representa uma célula e inicia em um vértice que representa o canal adjacente a sua borda sul e termina em um vértice que representa o canal adjacente a sua borda norte;
4. as arestas que saem de um vértice são ordenadas seguindo a relação "a esquerda de";
5. um canal vertical é definido por uma face do grafo;
6. existe um único vértice fonte, representando o canal mais ao sul, e um único vértice sorvedouro, representando o canal mais ao norte.

A partir do grafo de posicionamento relativo é possível se derivar uma expressão de posicionamento relativo, que é uma notação linear equivalente ao mesmo. Para tanto é suposto que as arestas do grafo são rotuladas com o nome das células que elas representam e são definidas duas operações de redução sobre o grafo (Figura 7 na página 30):

1. redução de adjacência horizontal - se duas arestas sucessoras simples rotuladas "x" e "y" têm origem em um mesmo vértice "A" e apontam para um mesmo vértice "B", e a aresta "x" é predecessora da aresta "y" pela relação "à esquerda de", então ambas podem ser trocadas por uma única aresta rotulada com a expressão "x | y" com origem em "A" e apontando para "B" ;
2. redução de adjacência vertical - se somente uma aresta simples rotulada "x" tem origem em um vértice "A", apontando para um vértice "B", e somente uma aresta simples rotulada "y" tem origem no vértice "B" apontando para um vértice "C" então as duas arestas podem ser substituídas por uma única aresta rotulada "x - y" com origem no vértice "A" apontando para o vértice "C". O vértice "B" é removido do grafo por esta operação.

Ficam assim definidos dois operadores "|" e "-" que indicam a adjacência horizontal e vertical, respectivamente, entre células. Sempre que houver ambiguidade na ordem de aplicação dos operadores



a.horizontal

b.vertical

Figura 7. Operações de redução de adjacência

"|" e "-" devem ser utilizados os parênteses para indicar a ordem correta de sua aplicação.

Para se obter a expressão de posicionamento equivalente a um determinado grafo de posicionamento relativo, deve-se aplicar sucessivamente as operações de redução até que reste apenas uma única aresta com origem em um vértice que representa o canal mais ao sul do posicionamento e apontando para um outro vértice que representa o canal mais ao norte do posicionamento. Este grafo de posicionamento é denominado de totalmente reduzido e o rótulo de sua única aresta é a expressão de posicionamento relativo equivalente ao grafo original. Um exemplo de tal redução, passo a

passo, é mostrado na Figura 8 na página 32. Nas expressões de posicionamento relativo é possível ainda se ter operadores unários precedendo as células para indicar rotação e reflexão das mesmas.

Infelizmente nem todos os grafos de posicionamento relativo são totalmente redutíveis (ver Figura 9 na página 33). Para superar tal problema é definida uma operação de desmembramento de tais grafos. Esta operação é definida sobre grafos não redutíveis e cria dois subgrafos onde o processo de redução é continuado. A operação é definida como segue:

1. construção do primeiro subgrafo: com respeito ao vértice fonte são possíveis duas situações para o grafo de posicionamento relativo não redutível. Duas ou mais arestas emergem do vértice fonte (neste caso rotulado  $V_0$ ) ou apenas uma aresta emerge do vértice fonte e aponta para um vértice  $V_0$  do qual duas ou mais arestas emergem. No último caso o vértice fonte e a aresta que emerge dele são adicionados ao primeiro subgrafo. O subgrafo consiste da aresta  $E_0$  mais a esquerda emergente de  $V_0$  e apontando para  $V_1$ , mais o caminho alternativo de  $V_0$  a  $V_1$  que não contém  $E_0$ . A primeira aresta deste caminho alternativo é a imediatamente a direita de  $E_0$ . Se durante a determinação do caminho for atingido um vértice do qual emergem várias arestas então deve-se tomar como próxima aresta a mais a esquerda;

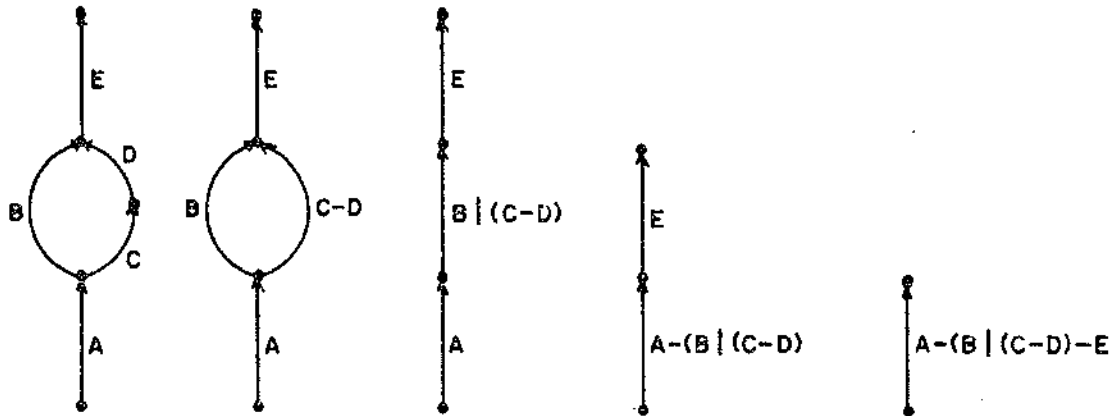


Figura 8. Redução do grafo de posicionamento relativo

2. construção do segundo subgrafo: é obtido a partir do grafo não redutível com a remoção de todos os vértices do primeiro subgrafo, que só tem arestas incidentes e emergentes pertencentes àquele subgrafo. Devem ser também removidas todas as arestas do primeiro subgrafo exceto aquelas estritamente necessárias à manutenção do segundo subgrafo como um grafo de fonte simples. Um grafo de fonte simples é aquele que possui um único vértice fonte.

Os dois subgrafos têm pelo menos uma aresta em comum. Assim, toda informação para regenerar o grafo não redutível está contida nos dois subgrafos. As sub-expressões derivadas da redução de cada

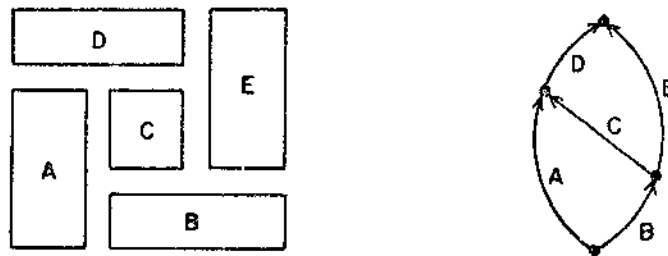
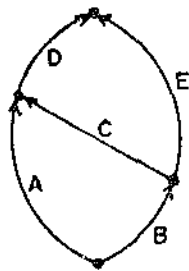
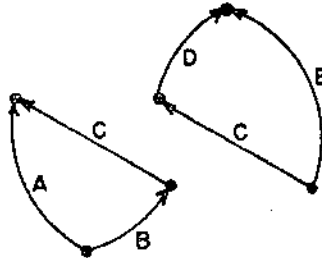


Figura 9. Grafo de posicionamento relativo não redutível

subgrafo são ligadas pelo operador "&", que indica o desdobramento, para produzir a expressão final (Figura 10 na página 34). Para derivar a sub-expressão do segundo grafo pode ser necessária uma nova aplicação da operação de desdobramento, mas as operações de redução sempre têm precedência sobre esta. Expressões derivadas de grafos totalmente redutíveis são ditas expressões puras enquanto que expressões que contém o operador "&" são ditas expressões mistas. Expressões mistas são mais difíceis de serem tratadas, do ponto de vista do usuário, mas elas são em geral capazes de descrever posicionamentos mais compactos.



a.grafo



b.desdobramento

$$A|(B-C) \& (C-D)|E$$

c.expressão

Figure 10. Desdobramento de grafo não redutível

---

### 3.0 O GERADOR DE POSICIONAMENTO AUTOMÁTICO

#### 3.1 Descrição estrutural de células

A descrição estrutural da célula cujas subcélulas precisam ser posicionadas deve ser especificada utilizando a gramática apresentada no "Apêndice A. ESPECIFICAÇÃO SINTÁTICA DA LINGUAGEM DE ENTRADA" na página 66. Esta descrição está dividida em três partes : descrição dos tipos das subcélulas, declaração dos exemplares destas subcélulas e estrutura de interconexões. Na descrição dos tipos das subcélulas utilizadas são especificadas suas interfaces (as suas dimensões e a posição de seus terminais) e, caso seja composta, a sua decomposição em termos de suas subcélulas e suas interligações.

Na declaração de exemplares de subcélulas, além de subcélulas isoladas é possível definir agrupamentos de subcélulas, ou seja, conjuntos de subcélulas com posicionamento relativo entre si já definido. Tais conjuntos de subcélulas serão manipulados em bloco pelo posicionador. A facilidade de definição de agrupamentos de subcélulas permite ao projetista ter um certo grau de controle sobre o posicionamento final das subcélulas.



Na descrição da estrutura de interconexões é especificado para cada sinal o conjunto de terminais de subcélulas que a ele pertence.

A partir da descrição estrutural textual é montada em uma estrutura de dados a descrição de cada subcélula. Em seguida é construído um vetor de apontadores para o exemplares das subcélulas isoladas e os agrupamentos declarados.

Para a verificação da conexidade das subcélulas é construída uma matriz de conexidade, que é simétrica em relação a diagonal principal. A estrutura de interconexões informa para cada sinal todos os terminais dos exemplares de subcélulas que devem ser interligados. A partir desta informação é possível construir para um circuito composto de 'n' subcélulas, uma matriz de conexidade 'C' de 'n' linhas por 'n' colunas. Cada elemento  $C(i,j)$  desta matriz, para  $i$  diferente de  $j$ ,  $i$  e  $j$  maior que zero e menor ou igual a 'n', é um número que expressa a quantidade de sinais que ligam a subcélula  $i$  à subcélula  $j$ . Para  $i$  igual a  $j$  este valor não é significativo para o processo de posicionamento.

### 3.2 Reconhecimento de regularidades

Antes de iniciar o posicionamento propriamente dito, procura-se identificar em cada nó (vértice do grafo representante da hierarquia de composição) se existe alguma estrutura regular de interligação entre subcélulas que compõem a célula definida no nó em questão. A idéia aqui é identificar estruturas para as quais são conhecidos padrões de posicionamento eficientes em termos de espaço ocupado e roteamento de interconexões. Se estas estruturas de interligações forem inicialmente reconhecidas, agrupadas e congeladas será evitado o esforço computacional necessário ao tratamento individual de cada subcélula e as subcélulas em questão não são dispersadas no posicionamento final resultando em circuitos mais densos e velozes.

Uma das estruturas regulares mais comum em projetos são as fileiras, grupos de células que interagem bastante com duas células do grupo em questão de tal forma que um posicionamento eficiente seja uma sequência linear das células deste grupo. A versão atual do reconhecedor só trata de estruturas deste tipo. Outros tipos de estruturas regulares passíveis de reconhecimento seriam por exemplo as estruturas matriciais de dispositivos para implementação de memórias e estrutura de processadores interligados em árvore.

Uma preocupação razoável nesta fase é com o comprimento total de tais seqüências que pode atingir extensões muito longas. Para tanto é realizada uma quebra destas seqüências em seqüências menores em pontos com grau de acoplamento mais baixos de tal forma que seja mantida a razão de aparência especificada (relação entre largura e altura do posicionamento).

O primeiro passo nesta fase é a construção de uma matriz de conexidade por face de exemplar de subcélula. A partir da informação existente na descrição das interconexões é possível se construir para um circuito composto de 'n' subcélulas, uma matriz de conexidade por face 'F'. Cada elemento  $F(i,fa,j,fb)$  desta matriz, para  $i$  diferente de  $j$ ,  $i$  e  $j$  maior que zero e menor ou igual a 'n', expressa o número de sinais que especificam a interligação de terminais na face  $fa$  da subcélula  $i$  com terminais da face  $fb$  da subcélula  $j$ .

As subcélulas que compõem a célula, para a qual foi invocada a ferramenta de posicionamento pode ser composta de: subcélulas primitivas, subcélulas compostas e agrupamentos de subcélulas. Durante a fase de reconhecimento de regularidades são analisadas apenas as subcélulas com traçado já definido.

Para cada subcélula  $S1$  é pesquisada, para cada face  $F1$  de  $S1$ , a subcélula  $S2$  cuja face  $F2$  é mais conexa com  $F1$ . Se existir uma tal subcélula  $S2$  é então verificado o grau de acoplamento entre as

faces F1 e F2. O grau de acoplamento é definido pela relação entre o número de terminais que ligam F1 a F2 e o número total de terminais da face F1 da subcélula S1, expresso em percentagem. Se o grau de acoplamento entre F1 e F2 for maior que um grau de acoplamento limite especificado na ferramenta e o comprimento da subcélula S1 mais o comprimento da subcélula S2 não violarem a razão de aparência especificada, então as subcélulas S1 e S2 são colocadas em uma lista, denominada de lista de acoplamentos. Esta lista já leva em conta a orientação de S1 e de S2.

No decorrer do algoritmo é formada uma lista de sublistas de acoplamentos. Assim, durante a execução do algoritmo, pode ocorrer de a subcélula S1 ou a subcélula S2, ou ambas, já pertencerem a alguma sublista de acoplamentos. Neste caso se as faces F1 da subcélula S1 e a face F2 da subcélula S2 ainda estiverem livres para acoplamento o mesmo será realizado. Desta forma em determinados momentos o algoritmo faz o acoplamento de duas sublistas de acoplamentos, isto se o seu comprimento total não violar a razão de aparência.

Ao final da execução do algoritmo tem-se uma lista de sublistas de acoplamentos a serem realizados. Na etapa seguinte, cada sublista de acoplamentos é consultada, sendo então montada uma expressão de posicionamento relativo para as subcélulas que a compõem. A partir deste ponto, estas subcélulas passam a ser tratadas como uma subcélula composta única.

### 3.3 Geração do posicionamento inicial

A estrutura utilizada na representação interna para os posicionamentos é um grafo de interseção de canal. Canais são as regiões não ocupadas pelas subcélulas já posicionadas. Nestes grafos, cada vértice representa uma interseção de canais, e cada aresta representa um canal. Na Figura 11 na página 40 é apresentado um posicionamento relativo de subcélulas e o respectivo grafo de interseção de canais. Na estrutura de dados que descreve o vértice superior esquerdo de cada face do grafo de interseção de canais é armazenado o endereço da estrutura de dados que descreve a subcélula associada aquela face do grafo.

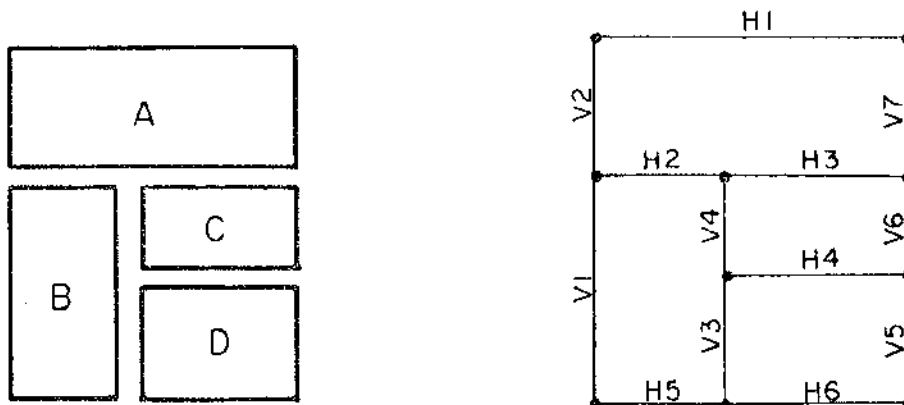


Figura 11. Grafo de interseção de canais

---

O primeiro passo na geração do posicionamento inicial é a pesquisa da subcélula ou agrupamento destas (tratado como simples célula neste contexto) mais conexo, na matriz de conexidade. Assim para cada subcélula é calculada a somatória do número de sinais com os quais interage com as demais subcélulas. É então escolhida como primeira subcélula a posicionar aquela que apresentar a maior somatória do número de sinais de ligação com as demais subcélulas.

A seguir é construído o grafo de interseção de canais para esta subcélula. O grafo de interseção de canais que representa o posicionamento com uma única subcélula consiste de quatro vértices e quatro arestas formando um ciclo. De uma forma mais abstrata, como as subcélulas tem formato retangular, este grafo pode ser visto como um retângulo circunscrito a representação retangular da subcélula em questão. Na estrutura de dados de um dos vértices, por convenção o superior esquerdo, é armazenado o endereço da estrutura de dados que descreve esta primeira subcélula.

Para o posicionamento de cada uma das demais subcélulas é necessário o desdobramento de um canal do grafo de interseção de canais. A forma de desdobramento de um canal depende do tipo de interseção de cada um dos extremos do canal. Na Figura 12 na página 42 é apresentado um quadro que resume os tipos de desdobramento em função dos tipos de interseção dos extremos do canal.

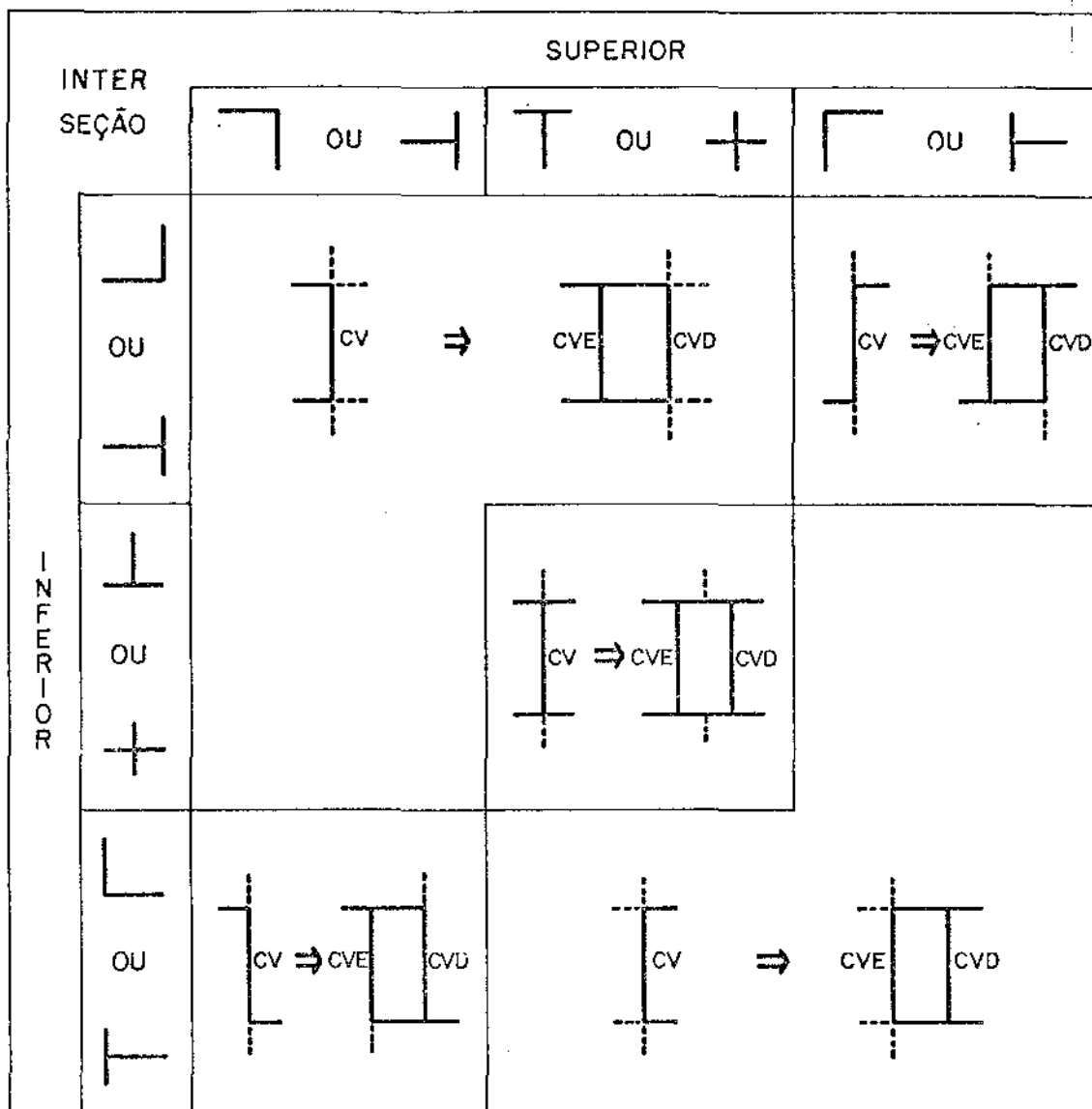


Figura 12. Tipos de desdobramento de canal

Para inclusão de uma subcélula 'E' no posicionamento da Figura 11, por exemplo, desdobrando-se o canal V4 ou o canal V3-V4 respectivamente, de acordo com o quadro da Figura 12 na página 42 teríamos os grafos dados na Figura 13 na página 43.

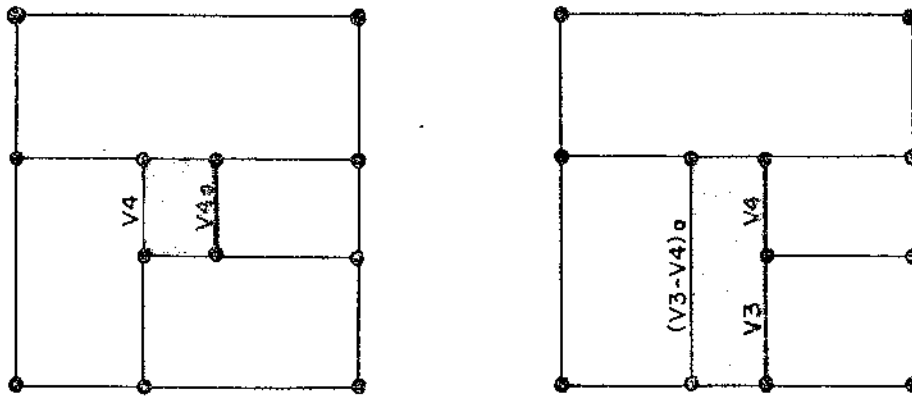


Figura 13. Desdobramento de canal

A próxima subcélula a ser posicionada é escolhida em função da sua conexão com o conjunto de subcélulas já posicionadas. A subcélula não posicionada, mais conexa com o conjunto de subcélulas é a escolhida. O objetivo desta escolha é o de posicionar as subcélulas mais conexas próximas umas das outras, o que tende a reduzir também o comprimento total das interligações.

Para a subcélula escolhida é feita uma tentativa de posicionamento da mesma em cada canal do grafo de interseção de canais, fazendo o desdobramento de canal conforme a configuração das interseções. Após cada tentativa é avaliado um valor de mérito que leva em conta a razão de aparência do posicionamento e a razão entre a área efetivamente ocupada e a área total do posicionamento. Como área total do posicionamento é computada a área do menor retângulo que envolve as subcélulas posicionadas.



Para o cálculo da razão de aparência é preciso calcular a largura e a altura do menor retângulo que envolve as subcélulas já posicionadas considerando a subcélula escolhida como posicionada no local referente a abertura de determinado canal. Uma vez que o número de tentativas pode ser bastante grande, o grafo de interseção de canais é deixado intacto, sem a inclusão efetiva da subcélula escolhida. O tipo de abertura do canal bem como as dimensões da subcélula a ser posicionada são considerados quando do cálculo das dimensões do posicionamento. Em cada canal a subcélula escolhida pode ser colocada em duas orientações diferentes, no que se refere as dimensões do posicionamento. Com a direção do lado maior paralela à direção da largura do posicionamento e com a direção do lado maior perpendicular à direção desta largura.

Para o cálculo das dimensões do posicionamento é executado um procedimento recursivo. Este procedimento utiliza o grafo de interseção de canais. O argumento de chamada é um apontador para a estrutura de dados que descreve o vértice superior esquerdo "v0" do grafo. Na Figura 14 na página 45 é descrito o algoritmo utilizado neste procedimento para o cálculo da largura do posicionamento.

Para o cálculo da altura de um posicionamento é utilizado o mesmo algoritmo, fazendo-se apenas as modificações necessárias de direção e referencial.

```

Procedimento DIMENSAO_X (v:apt_vértice ; var largura:inteiro);
Início
  Se 'v' está na borda leste do posicionamento
  então largura := 0
  senão largura := DIMENSAO_X (v->adjacente[leste]) +
    largura da célula associada a 'v'
  Se v->adjacente[sul] tem uma célula associada
  então largura := MAX (largura,DIMENSAO_X
    (v->adjacente[sul]))
Fim

```

Figura 14. Algoritmo de determinação da largura de um posicionamento

---

Em relação ao algoritmo tradicional de crescimento epitaxial o algoritmo aqui descrito para geração do posicionamento inicial difere basicamente em dois aspectos: a situação inicial de aplicação do algoritmo e a flexibilidade na direção do seu crescimento durante a construção. No algoritmo tradicional de crescimento epitaxial deve ser encontrado inicialmente uma fileira de subcélulas que tenha um comprimento total aproximadamente igual a largura desejada para o posicionamento final. Para tal fileira de subcélulas é construído o grafo de interseção de canais que a representa. Já o algoritmo de geração do posicionamento inicial

aqui descrito, parte de uma subcélula (ou de um agrupamento de subcélulas tratado como uma única subcélula neste contexto) mais conexa com as demais subcélulas, sem maiores restrições. Sendo construído para esta subcélula o grafo de interseção de canais.

Na sequência do algoritmo tradicional de crescimento epitaxial é escolhida uma subcélula por vez, segundo um critério de conexidade, para ser incorporada ao posicionamento em construção, como no algoritmo de geração do posicionamento inicial aqui descrito. No primeiro algoritmo porém, os canais do grafo a serem desdobrados na tentativa de inclusão desta subcélula no posicionamento em construção são somente aqueles cuja abertura não implique em aumento na largura do posicionamento final. No segundo algoritmo não existe tal restrição, ou seja todos os canais são passíveis de abertura, sendo que a razão de aparência é levada em conta quando do cálculo do valor de mérito.

#### **3.4 Melhoria do posicionamento inicial**

A partir do posicionamento inicial para as subcélulas, na fase de melhoria, são realizadas trocas tentativas de pares de subcélulas. O algoritmo de melhoria do posicionamento inicial é executado em um ciclo que se repete para cada subcélula que o

compõe, a qual é tratada nesta descrição como subcélula base. O número de tentativas possíveis cresce exponencialmente em relação ao número de subcélulas. Para limitar este número, a cada subcélula base são calculadas as coordenadas do ponto ideal para o seu posicionamento. Este ponto é determinado pela média das coordenadas do centro de cada subcélula pertencente ao conjunto das subcélulas conexas a subcélula considerada. Esta média é ponderada no número de sinais que ligam cada subcélula posicionada a subcélula base.

As subcélulas candidatas a troca com a subcélula base devem ter o seu centro dentro de um retângulo. Cada lado deste retângulo é igual a duas vezes um parâmetro de tolerância definido na ferramenta multiplicado pela dimensão total do posicionamento na direção do lado considerado. Este parâmetro é expresso em percentagem. O centro do retângulo é exatamente o ponto ideal para o posicionamento da subcélula considerada.

Definido o espaço de subcélulas candidatas a troca com a subcélula base é então avaliado um valor de mérito para o posicionamento atual ( $vm1$ ). Para cada possível posicionamento resultante da troca de posição entre a subcélula base e cada candidata a troca é avaliado um valor de mérito. O maior valor de mérito obtido dentre todas as possíveis trocas ( $vm3$ ), no espaço delimitado, é comparado com o valor de mérito  $vm1$ . Se  $vm3$  for maior que  $vm1$  então a troca

entre a subcélula base e a respectiva candidata, que resultou no valor de mérito  $vm_3$ , é efetivada.

A estruturação do algoritmo utilizado na fase de melhoramento do posicionamento inicial é apresentado na Figura 15 na página 49. O valor de mérito aqui computado leva em conta a razão de aparência e a razão entre a área ocupada e a área total do posicionamento.

Após a execução do algoritmo acima apresentado terão sido realizadas todas as trocas dentro do escopo da pesquisa que resultarem em uma melhora efetiva do posicionamento inicial em termos de eficiência de ocupação de área e de razão de aparência.

```

Para cada subcélula Ai do posicionamento
  Calcular o valor de mérito do posicionamento atual (vm1)
  Armazenar vm1 em vm3
  {Computar o conjunto T de subcélulas candidatas a troca
  Para cada subcélula Aj pertencente ao conjunto T
    {Considerar a troca da subcélula Ai com a subcélula Aj
    Computar o valor de mérito para esta troca (vm2)
    Se vm2 maior que vm3
      Então Armazenar vm2 em vm3
      Armazenar j em c
    }
  }
  Se vm3 maior que vm1
    Então Efetivar a troca de Ai com Ac
  }

```

Figura 15. Algoritmo de melhoramento do posicionamento inicial

---

## 4.0 CONCLUSÃO

### 4.1 Resumo do trabalho

Foi apresentado o conceito de projeto de circuitos integrados em suas diversas abordagens. O projeto totalmente automático está expresso no conceito de compilação de silício, onde a partir de uma descrição funcional, um compilador deve realizar a tradução desta descrição em um conjunto de células interligadas. Estas células pertencem ao conjunto de células básicas do compilador de silício em analogia ao conjunto básico de instruções do código gerado por um compilador de linguagem de programação de alto nível.

Uma das fases do projeto automático de circuitos integrados é a que corresponde ao traçado do circuito (conforme definido na página 4), onde é efetivamente alocado o espaço para cada uma das células básicas bem como para as suas interligações. O posicionamento automático trata de encontrar uma disposição relativa ou absoluta para as células que seja compacta e minimize o problema na fase seguinte que corresponde ao roteamento das ligações. As ferramentas de posicionamento estão divididas em dois grupos : geradores de posicionamento inicial e melhoradores de

posicionamento. Foram abordadas as diferentes técnicas utilizadas em cada um destes grupos.

O posicionador automático aqui apresentado possui uma fase preliminar na qual é feito um reconhecimento de regularidades agrupando aquelas subcélulas que apresentem uma estrutura de interligação regular. A fase seguinte, de geração do posicionamento inicial, é realizada de forma construtiva, as subcélulas são posicionadas uma a uma obedecendo a um critério de máxima conexidade com o grupo de subcélulas já posicionadas e da influência de sua inclusão no agrupamento sobre a razão de aparência e a taxa de ocupação do posicionamento em construção.

Na última etapa do posicionamento automático é feito um melhoramento do posicionamento inicial através da troca de pares de subcélulas. Para cada subcélula é determinado um conjunto vizinhança de subcélulas candidatas a serem seu par na troca. Este conjunto de subcélulas é constituído daquelas subcélulas que estão na vizinhança de um ponto ideal para o posicionamento da subcélula considerada. Este ponto é determinado por uma média ponderada das coordenadas do centro de cada subcélula. O peso desta ponderação é o número de sinais que interligam a subcélula considerada às demais subcélulas. A subcélula considerada é então trocada sucessivamente com cada subcélula do conjunto vizinhança. Finalmente é avaliado se a troca, que resultou no melhor



posicionamento, melhora o posicionamento inicial. Em caso afirmativo esta troca é efetivada.

A ferramenta foi desenvolvida em linguagem PASCAL. O conjunto de programas fonte tem um total de seis mil e quinhentas linhas, das quais a rotina de posicionamento corresponde a 46%.

Na Figura 16 na página 53 é apresentado um quadro que mostra o tempo de execução da ferramenta para cada um dos exemplos cuja entrada e saída foi apresentada no "Apêndice B. EXEMPLOS DE ENTRADA E SAÍDA" na página 78. Foi computado o tempo de execução em cada uma das fases: reconhecimento de regularidades, geração do posicionamento inicial e melhoramento do posicionamento inicial. Os dois exemplos foram executados três vezes cada um variando apenas a razão de aparência desejada ao posicionamento final. Estes exemplos foram executados em um microcomputador de 16 bits, com microprocessador 80286, com um drive de disco rígido e relógio com frequência de 10 mega Hertz.

Para o comparativo de eficiência em termos de taxa de ocupação de área do posicionamento final, bem como de razão de aparência solicitada em relação a razão de aparência final obtida, foi montado um quadro que está apresentado na Figura 17 na página 54.

Exem plo	No. de subcé lulas	Razão  Aparên cia	Tempo em seg. por fase			Tempo  Total
			Rec. Reg.	Posic. Inic.	Melhora mento	
1	5	1.0	2.1	1.9	0.4	4.4
		0.8	2.2	1.9	0.3	4.4
		0.6	2.2	1.8	0.3	4.3
2	17	1.0	10.3	9.1	1.9	21.3
		0.8	10.3	8.8	1.1	20.2
		0.6	10.3	8.7	1.1	20.1

Figura 16. Quadro de tempos de execução

---

Exem plo	No. de subcé lulas	Razão de Aparência Solicitada	Razão de Aparência Obtida	Taxa de Ocupação
1	5	1.00	1.00	0.81
		0.80	0.71	0.83
		0.60	0.50	0.91
2	17	1.00	0.90	0.74
		0.80	0.78	0.65
		0.60	0.58	0.80

Figura 17. Comparativo de razão de aparência e taxa de ocupação

---

## 4.2 Contribuições principais

Neste trabalho estão sendo apresentadas basicamente duas inovações que contribuem para um melhor desempenho e flexibilidade na etapa de posicionamento. A primeira delas consiste no reconhecimento de regularidades que não só reduz o esforço computacional nas fases seguintes como também evita que subcélulas que estejam fortemente interligadas venham, em alguma fase do processo automático de posicionamento, a ser posicionadas com uma disposição relativa diferente daquela comprovadamente mais eficiente.

A segunda inovação aqui apresentada dá ao projetista algum controle, bem como alguma flexibilidade, no posicionamento. Ela consiste na possibilidade de o projetista informar, quando define os exemplares das subcélulas que compõem a célula em projeto, agrupamentos de subcélulas já com uma disposição relativa definida e que passam a ser tratados como um bloco único. Desta forma aquelas partes do projeto que são perfeitamente definidas e cuja disposição relativa ideal é conhecida do projetista podem ser agrupadas e congeladas. Isto reduz o esforço computacional e também evita que as subcélulas que compõem o agrupamento sejam dispersas no plano de posicionamento pelos algoritmos utilizados no posicionamento automático.

O algoritmo aqui descrito na fase de geração do posicionamento inicial difere do algoritmo tradicional de crescimento epitaxial, na medida em que este permite um crescimento do posicionamento em qualquer direção partindo de uma única subcélula inicialmente posicionada. Já o algoritmo de posicionamento inicial por crescimento epitaxial fixa a largura do posicionamento final e parte de uma fileira de subcélulas que em conjunto tenham um comprimento total igual a tal largura. O crescimento do posicionamento se dá em uma única direção e num só sentido.

Uma característica que dá uma grande flexibilidade, principalmente na fase de roteamento das interligações, é o fato de o posicionamento automático obtido ser relativo. Isto permite ao roteador ajustar durante o processo de roteamento a dimensão dos canais, sem que seja necessário um novo posicionamento, cada vez que as interligações a serem roteadas em um canal não couberem no mesmo.

#### 4.3 Futuras extensões

A ferramenta de posicionamento automático aqui apresentada, possui alguns pontos cuja abordagem é suscetível de um refinamento ou maior abrangência.

Na fase do reconhecimento de regularidades, houve uma concentração no sentido de localizar aqueles agrupamentos de células, que devido ao seu alto grau de conexidade, seria conveniente que fossem posicionadas próximas umas das outras. Um possível aperfeiçoamento da ferramenta está na busca do reconhecimento de outros padrões de regularidade para os quais se conhece um posicionamento eficiente. Um exemplo característico é a ligação de processadores numa estrutura em árvore [Lei81].

Durante a fase de construção do posicionamento inicial, para se comparar entre diversas possíveis colocações para uma nova célula a incluir no posicionamento, é calculado um valor de mérito. Este valor avalia a razão de aparência do posicionamento e a razão entre a sua área ocupada e a área total do menor retângulo que circunscreve o posicionamento. É interessante que se adicione no cálculo deste valor de mérito uma variável que exprima a distância da célula em relação ao ponto considerado ideal para o seu posicionamento. O ponto ideal pode ser computado considerando um modelo simples que exprima a força de atração entre as células, baseado na sua conexidade. O ponto ideal é aquele onde a força de atração sobre a célula é nula. Isto teria o efeito de posicionar a célula próximo ao ponto em que as forças de atração sobre a mesma são nulas.

A ferramenta de posicionamento relativo de subcélulas aqui apresentada foi desenvolvida voltada para aplicação no traçado de

circuitos descritos por uma hierarquia de composição. Neste caso a ferramenta é invocada em cada nó da hierarquia para posicionar as subcélulas que compõem a célula representada pelo nó em questão. Tendo em vista este contexto de aplicação, não foi feita uma redução no espaço de busca de posições possíveis para subcélulas selecionadas a serem incorporadas no posicionamento em construção. Para aplicação da ferramenta no posicionamento de um grande número de subcélulas é interessante se aplicar uma redução em tal espaço de busca. Esta redução para ser eficiente deve levar em conta a posição ideal para a subcélula a ser adicionada no posicionamento.

## 5.0 BIBLIOGRAFIA

- [Asa82] T. Asano, "An Optimum Gate Placement Algorithm for MOS One-Dimensional Arrays", Journal of Digital Systems, Vol. VI, No. 1, p. 1-27 (1982).
- [Bar84] A.M. Barone & J.K. Morrell, "Custom Chip/Card Design System", IBM Journal Res. & Develop., Vol. 28, No. 5, p. 590-5 (09/84).
- [Ber83] N. Bergmann, "A Case Study of the F.I.R.S.T. Silicon Compiler", Third Caltech Conference on Very Large Scale Integration, p. 413-30 (03/83).
- [Bla84] T. Blank, "A Survey of Hardware Accelerators Used in Computer-Aided Design", IEEE Des. & Test Comput., Vol. 1, No. 3, p. 21-39 (08/84).
- [Bre83] G. Brebner & D. Buchanan, "On Compiling Structural Descriptions to Floorplans", IEEE International Conference on Computer-Aided Design ICCAD-83, Santa Clara, CA, USA, p. 6-7 (09/83).



- [Car79] H.W. Carter, M.A. Breuer & Z.A. Syed, "Incremental Processing Applied to Steinberg's Placement Procedure", 16th Design Automation Conference, San Diego, CA, USA, p. 26-31 (06/79).
- [Che83] C.K. Cheng & E.S. Kuh, "Partitioning and Placement Based on Network Optimization", IEEE International Conference on Computer-Aided Design ICCAD-83, Santa Clara, CA, USA, p. 86-7 (09/83).
- [Cos87] E.M. da Costa, "Introdução ao Projeto de Circuitos Integrados", II Escola Brasileiro-Argentina de Informática, Universidad Nacional del Centro de La Provincia de Buenos Aires, Tandil, Argentina (09-22/02/87).
- [DeM82] H. De Man, "Computer Aided Design Techniques for VLSI", no livro "Design Methodologies for VLSI Circuits", editado por P.G. Jaspers, C.H. Sequin & F. van de Wiele, Sijthoff & Noordhoff, p. 173-225 (1982).
- [Den83] P.B. Denyer, "Silicon Compilers and VLSI", no livro: Semi-Custom IC Design, editado por P.J. Hicks, Peter Peregrinus Ltd, p. 186-96 (1983).

- [Dun83] A.E. Dunlop & B.W. Kernighan, "A Placement Procedure for Polycell VLSI Circuits", IEEE International Conference on Computer-Aided Design ICCAD-83, Santa Clara, CA, USA, p. 51-2 (09/83).
- [Dun85] A.E. Dunlop & B.W. Kernighan, "A Procedure for Placement of Standard-Cell VLSI Circuits", IEEE Trans. Comput. Aided Des. Integrated Circuits & Systems, Vol. CAD-4, No. 1, p. 92-8 (01/85).
- [Eld84] W.H. Elder, P.P. Zenewicz & R.R. Alvarodiaz, "An Interactive System for VLSI Chip Physical Design", IBM Journal Res. Develop., Vol. 28, No. 5 (09/84).
- [Fie84] R.D. Fiebrich, Y.Z. Liao, G. Koppelman & E. Adams, "PSI: A Symbolic Layout System", IBM Journal Res. Develop., Vol. 28, No. 5, p. 572-80 (09/84).
- [Han76] M. Hanan, P.K. Wolff & B.J. Agule, "Some Experimental Results on Placement Techniques", 13th Design Automation Conference, San Francisco, CA, EUA, p.214-24 (06/76)
- [Hir82] S. Hirschhorn & A.M. Davis, "The Revolution in VLSI Design: Parallels between Software and VLSI Engineering", Sixteenth Asilomar Conference on Circuits,

Systems and Computers, Pacific Grove, CA, EUA, p. 539-48 (11/82).

- [Hor81] C.S. Horng & M. Lie, "An Automatic / Interactive Layout Planning System for Arbitrarily-sized Rectangular Building Blocks", 18th Design Automation Conference, Nashville, TN, EUA, p. 293-300 (07/81).
- [Hud84] J.A. Hudson & J.A. Wisniewski, "Module Positioning Algorithms for Rectilinear Macrocell Assemblies", ACM IEEE 21th Design Automation Conference Proceedings 84, Albuquerque, NM, USA, p. 672-5 (06/84).
- [Joh84] S.C. Johnson & S. Mazor, "Silicon Compilers Lets Systems Makers Design their own VLSI Chips", Electron. Des., Vol. 32, No. 20, USA, p. 167-81 (10/84).
- [Koz84] K. Kozminski & E. Kinnen, "An Algorithm for Finding a Rectangular Dual of a Planar Graph for Use in Area Planning for VLSI Integrate Circuits", ACM IEEE 21th Design Automation Conference Proceedings 84, Albuquerque, NM, USA, p. 655-6 (06/84).
- [Lei84] S.M. Leinwand & Y.-T. Lay, "An Algorithm for Building Floor-plans", ACM IEEE 21th Design Automation Conference Proceedings 84, Albuquerque, NM, USA, p. 663-4 (06/84).

- [Lie84] H.K.E. Liesenberg & D.J. Kinniment, "Relative Placement Representation", SIGDA Newsletters, Vol. 14, No. 2, p. 3-7 (06/84).
- [Lie85] H.K.E. Liesenberg, "Layout Module for a Silicon Compiler", Tese de Doutorado, University Newcastle-Upon-Tyne, Inglaterra, 251 p. (04/85).
- [Lie86] H.K.E. Liesenberg, "Projetos VLSI e seu Suporte Computacional", V Escola de Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais (10-18/07/86).
- [Lin84] R. Linsker, "Wire Routing Subject to Constraints", IBM Technical Disclosure Bulletin, Vol. 27, No. 2, p.978-82 (07/84).
- [McG83] R.W. McGuffin & W.R. Young, "An Overview of VLSI Design Automation Requirements", University / Government / Industry Microelectronics Symposium UGIM-83, College Station, TX, USA, p. 113-9 (05/83).
- [Mee80] C.A. Mead & L.A. Conway, "Introduction to VLSI Systems", Addison-Wesley Publishing Company, Inc. (1980).

- [Mur82] S. Muroga, "VLSI System Design", John Wiley & Sons, Inc. (1982).
- [Ous85] J.K. Outsterhout, G.T. Hamachi, R.N. Mayo, W.S. Scott & G.S. Taylor, "The Magic VLSI Layout System", IEEE Des. & Test. Comput., Vol. 2, No. 1, USA, p. 19-30 (02/85).
- [Pre79] B.T. Preas & W.M. vanCleemput, "Placement Algorithms for Arbitrarily Shaped Blocks", ACM IEEE 16th Design Automation Conference, San Diego, CA, USA, p. 474-80 (06/79).
- [Pre79a] B.T. Preas & W.M. vanCleemput, "Routing Algorithms for Hierarchical IC Layout", Proc. Int. Symposium Circuits and Systems, Tóquio, Japão, p. 482-5 (1979).
- [Seq83] C.H. Sequin, "Managing VLSI Complexity : an Outlook", Proceedings of the IEEE, Vol. 71, No. 1, p. 149-70 (01/83).
- [Ste84] K.R. Stevens, "A Review of Current Placement and Routing Techniques for Integrated Circuits", IEEE Colloquium on IC Design - Gate Arrays (Digest No. 99), Londres, Inglaterra, p. 1-8 (11/84).

- [Ste85] L.I. Steinberg & T.M. Mitchell, "The Redesign System : A Knowledge Based Approach to VLSI CAD", IEEE Des. & Test Comput., Vol. 2, No. 1, USA, p. 45-54 (02/85).
- [Tri81] S. Trimberger, J.A. Rowson, C.R. Lang & J.P. Gray, "A Structured Design Methodology and Associated Software Tools", IEEE Transactions on Circuits and Systems, Vol. CAS-24, No. 7, p. 618-34 (07/81).
- [Ued85] K. Ueda, H. Kitazawa & I. Harada, "CHAMP : Chip Floor Plan for Hierarchical VLSI Layout Design", IEEE Trans. Comput.-Aided Des. Integrated Circuits & Systems, Vol. CAD-4, No. 1, p. 12-22 (01/85).
- [Wer83a] J. Werner, "Progress Toward the 'Ideal' Silicon Compiler. Part 1: The Front End", VLSI Design, Vol. 4, No. 5, p. 38-41 (09/83).
- [Wer83b] J. Werner, "Progress Toward the 'Ideal' Silicon Compiler. Part 2: The Layout Problem", VLSI Design, Vol. 4, No. 6, p. 78-81 (10/83).

## Apêndice A. ESPECIFICAÇÃO SINTÁTICA DA LINGUAGEM DE ENTRADA

### A.1 Introdução

A linguagem de entrada cuja sintaxe é aqui descrita foi especificada por [Lie85] e tem um escopo de maior abrangência. Ela foi especificada como linguagem de entrada para uma ferramenta que faz o traçado completo de um circuito integrado (ver definição na página 4). Existem por este motivo algumas diretivas como as que invocam ferramentas específicas, para esticar e justapor subcélulas, que foram mantidas na especificação aqui apresentada, embora não façam parte do contexto referente ao posicionamento relativo de subcélulas.

### A.2 Descritor de célula

O elemento básico da linguagem é o descritor de célula que representa a definição de uma célula na hierarquia de composição do circuito especificado. O descritor de célula contém toda a

informação estrutural e de interface conhecida sobre a mesma num estado particular do processo de geração do traçado.

```
<descriptor de célula> ::= (* <cabeçalho do descriptor de célula>  
    <corpo do descriptor de célula> *)
```

### A.3 Cabeçalho do descriptor de célula

O cabeçalho do descriptor contém o nome da célula, as suas dimensões reais ou previstas, nome dos seus pontos terminais e possível posição no retângulo delimitador da célula. O identificador da célula pode ser complementado por um número qualquer de índices, sem importar quantos índices estão sendo usados em outros descritores de célula com o mesmo identificador. Não existe entretanto nenhuma semântica associada aos índices. Eles apenas representam um mecanismo de atribuição de nomes para distinção entre definições de célula associadas com o mesmo identificador, para se conseguir unicidade de nomes.

```
<cabeçalho do descriptor de célula> ::= <nome da célula>  
    ( <comprimento> , <altura> )  
    <especificação dos terminais>
```



<nome da célula> ::= <identificador da célula> |  
                  <identificador da célula> [ <lista de índices> ]

<lista de índices> ::= <constante inteira> |  
                  <constante inteira> , <lista de índices>

<comprimento>, <altura> ::= - | <constante inteira>

#### A.4. Especificação dos terminais

A especificação dos terminais define a interface da célula com o mundo exterior. Os terminais são identificados, se possível, associados a uma das bordas do retângulo delimitador da célula e associados a uma camada. Os nomes dos terminais também podem ser indexados. Os índices no nome dos terminais tem o mesmo significado que no caso do nome de célula. Os nomes associados aos terminais de uma mesma célula devem em geral ser únicos, dentro da sua definição. O seu escopo é restrito a definição da célula onde estão declarados. A equivalência elétrica é expressa através da utilização de um mesmo nome para diferentes pontos terminais dentro da mesma definição de célula. A ordem de posicionamento dos pontos terminais em uma mesma borda, se os mesmos não tem posição absoluta definida, é aquela em que eles aparecem na sua

declaração. A posição de um terminal em uma borda é dada por um deslocamento em relação a um ponto de referência. Para as bordas verticais este ponto de referência é o seu extremo inferior, enquanto que para as bordas horizontais é o extremo esquerdo. Um ponto terminal livre é aquele que ainda não foi associado a nenhuma das bordas. O símbolo - representa informação indefinida e o símbolo & representa informação nula.

<especificação dos terminais> ::=

<terminais livres> |  
<terminais associados a uma borda> |  
<terminais livres>  
<terminais associados a uma borda>

<terminais livres> ::= [+ <lista de terminais> +]

<terminais associados a uma borda> ::=

[ <terminais norte> ] [ <terminais leste> ]  
[ <terminais sul> ] [ <terminais oeste> ]

<terminais norte>,<terminais leste>,  
<terminais sul>,<terminais oeste> ::= & |  
<lista de terminais>

<lista de terminais> ::= <terminal> |  
<terminal> , <lista de pontos terminais>





compostas diferentes. Para evitar a repetição de um descritor de célula em um texto de entrada, ele pode ser definido somente uma vez em um nível sintático mais elevado onde ele é global a todos os seus irmãos a sua direita (células definidas no mesmo nível sintático, cuja definição vem após a definição considerada no texto de entrada) e aos seus descendentes. Assim, o nível sintático no qual uma célula está definida não é necessariamente o mesmo onde ela é usada como tipo em uma declaração de exemplares de subcélulas.

```
<corpo composto> ::= <lista de tipos de subcélulas>  
    <posicionamento de exemplares de subcélulas>  
    <especificação de interligações>
```

```
<lista de tipos de subcélulas> ::= <descritor de célula> |  
    <descritor de célula> <lista de tipos de subcélulas>
```

#### **A.7 Posicionamento de exemplares de subcélulas**

Os exemplares de subcélulas são declarados e posicionados na cláusula de posicionamento de exemplares de subcélulas. Estão disponíveis três tipos de cláusulas de posicionamento: absoluto, relativo e indefinido. Um posicionamento absoluto é expresso em

termos de coordenadas da origem dos exemplares das subcélulas relativamente a origem da células que compõem. Um posicionamento relativo é definido por relações de adjacência em uma sequência de expressões de posicionamento. Seis operadores de adjacência são utilizados com este propósito. Os operadores **autoleft** e **autobelow** representam as relações de adjacência horizontal e vertical, respectivamente, entre exemplares de subcélulas ou agrupamentos de exemplares de subcélulas que supostamente serão interligados por uma ferramenta automática de roteamento. Os operadores **abutleft** e **abutbelow** representam as relações de adjacência horizontal e vertical, respectivamente, entre exemplares de subcélulas ou agrupamentos de exemplares de subcélulas a serem interligados por uma ferramenta automática que estique e justaponha os exemplares das células. Se um ou mais operandos dos operadores **abutleft** e **abutbelow** estão descritos em termos de um ou mais dos operadores **autoleft** e **autobelow** então a ferramenta automática que estica e justapõe não será invocada. Neste caso os operadores **abutleft** e **abutbelow** são tratados como operadores **autoleft** e **autobelow**. Os operadores **left** e **below** são usados como operadores de adjacência sem qualificação quanto a ferramenta de interligação a utilizar. Os índices nos nomes dos exemplares tem o mesmo mecanismo e significado que tem no caso dos nomes de células. Os exemplares de subcélulas devem ter nome único dentro de uma definição. Um exemplar de subcélula ou um agrupamento de exemplares de subcélulas pode ser precedido por um operador de transformação que indica como ele tem que ser rotacionado ou refletido antes de ser

posicionado. Os operadores GAELIC ("An Introduction Guide to GAELIC Language", Rutherford and Appleton Laboratories) e o operador u são utilizados com este propósito. O operador u indica uma transformação indefinida.

```
<posicionamento de exemplares de subcélulas> ::=  
    <posicionamento absoluto> | <posicionamento relativo> |  
    <posicionamento indefinido>
```

```
<posicionamento relativo> ::= <folha> | <sub_pos_just> |  
    <sub_pos_auto> | <sub_pos_nqualif>
```

```
<folha> ::= <transformação> ( <nome do exemplar de subcélula> :  
    <tipo do exemplar de subcélula> )
```

```
<transformação> ::= u | xx | rrr | rr | r | y | ry | x | rx
```

```
<sub_pos_abut> ::= <transformação> ( <posicionamento relativo>  
    <operador_abut> <posicionamento relativo> )
```

```
<operador_abut> ::= abutleft | abutbelow
```

```
<sub_pos_auto> ::= <transformação> ( <posicionamento relativo>  
    <operador_auto> <posicionamento relativo> )
```

```
<operador_auto> ::= autoleft | autobelow
```

<nome do exemplar de subcélula> ::=

<identificador do exemplar de subcélulas> |  
<identificador do exemplar de subcélulas>  
[ <lista de índices> ]

<tipo do exemplar de subcélula> ::= <nome da célula>

<sub\_pos\_nqualif> ::= <transformação> (  
    <posicionamento relativo> <operador\_nqualif>  
    <posicionamento relativo> )

<operador\_nqualif> ::= left | below

<posicionamento absoluto> ::= <átomo> ( <x> , <y> ,  
    <transformação> ) | <átomo> ( <x> , <y> ,  
    <transformação> ) <posicionamento absoluto>

<posicionamento indefinido> ::= <lista de átomos parcialmente  
    agrupados ou não>

<lista de átomos parcialmente agrupados ou não> ::=  
    <átomo> | <agrupamento> |  
    <átomo> <lista de átomos parcialmente agrupados ou não>  
    | <agrupamento>  
    <lista de átomos parcialmente agrupados ou não>



<átomo> ::= <nome de exemplar de subcélula> :  
          <tipo de exemplar de subcélula> .

<agrupamento> ::= <posicionamento relativo>

#### A.8 Especificação de interligações

A especificação de interligações representa a descrição dos requisitos de interligações internas e, num estágio mais adiante do projeto, a sua implementação em termos de conexões e contatos. Contatos, por suposição, estão implicitamente definidos pelo cruzamento de duas conexões pertencentes ao mesmo sinal, ou por uma justaposição de um terminal com uma conexão ou com um outro terminal em diferentes camadas. Se uma célula tem mais de um terminal com o mesmo nome (terminais eletricamente equivalentes) então o símbolo "-" precedendo a especificação de sua interligação indica que as conexões internas entre estes terminais foram dimensionadas para permitir que elas sejam utilizadas como parte das conexões externas. O símbolo self em uma declaração de sinal se refere a célula composta sendo definida pelo descritor de célula corrente.

<especificação de interligações> ::= <lista de sinais>

<lista de sinais> ::= <sinal> | <sinal> <lista de sinais>

<sinal> ::= (+ <largura do sinal> :  
[ <especificação de sinal> ] +) |  
(+ <largura do sinal> : [ <especificação de sinal> ]  
<lista de conexões> +)

<especificação de sinal> ::= <ponto> | <ponto> ,  
<especificação de sinal>

<ponto> ::= <nome de terminal> /  
<nome de exemplar de subcélula> | - <nome de terminal> /  
<nome de exemplar de subcélula> |  
<nome de terminal> / self | - <nome de terminal> / self

<lista de conexões> ::= <conexão> | <conexão> <lista de conexões>

<conexão> ::= ( <largura> , <x1> , <y1> , <x2> , <y2> ,  
<camada> )

## Apêndice B. EXEMPLOS DE ENTRADA E SAÍDA

### B.1 Exemplo 1

#### B.1.1 Entrada

```
(* master (-,-)
  []
  [vdd(-,4,metal)]
  []
  [gnd(-,4,metal)]

  (* sub1 (30,10)
    [vdd(5,4,metal)]
    [c3(5,4,metal)]
    []
    [gnd(5,4,metal)]
    =(sub1.def)
  *)
  (* sub2 (30,10)
    [vdd(5,4,metal)]
```

```

    []
    [c1(5,2,poly),c2(15,2,poly)]
    [gnd(3,4,metal),c3(7,2,poly)]
    =(sub2.def)
*)
(* sub3 (20,50)
    [vdd(5,4,metal)]
    []
    [c1(5,2,poly),c2(15,2,poly)]
    [gnd(5,4,metal),c3(15,2,poly)]
    =(sub3.def)
*)
(* sub4 (30,10)
    [vdd(5,4,metal)]
    []
    [c1(5,2,poly),c2(15,2,poly)]
    [gnd(3,4,metal),c3(7,2,poly)]
    =(sub4.def)
*)
(* sub5 (20,50)
    [vdd(5,4,metal)]
    []
    [c1(5,2,poly),c2(15,2,poly)]
    [gnd(5,4,metal),c3(15,2,poly)]
    =(sub5.def)
*)

```

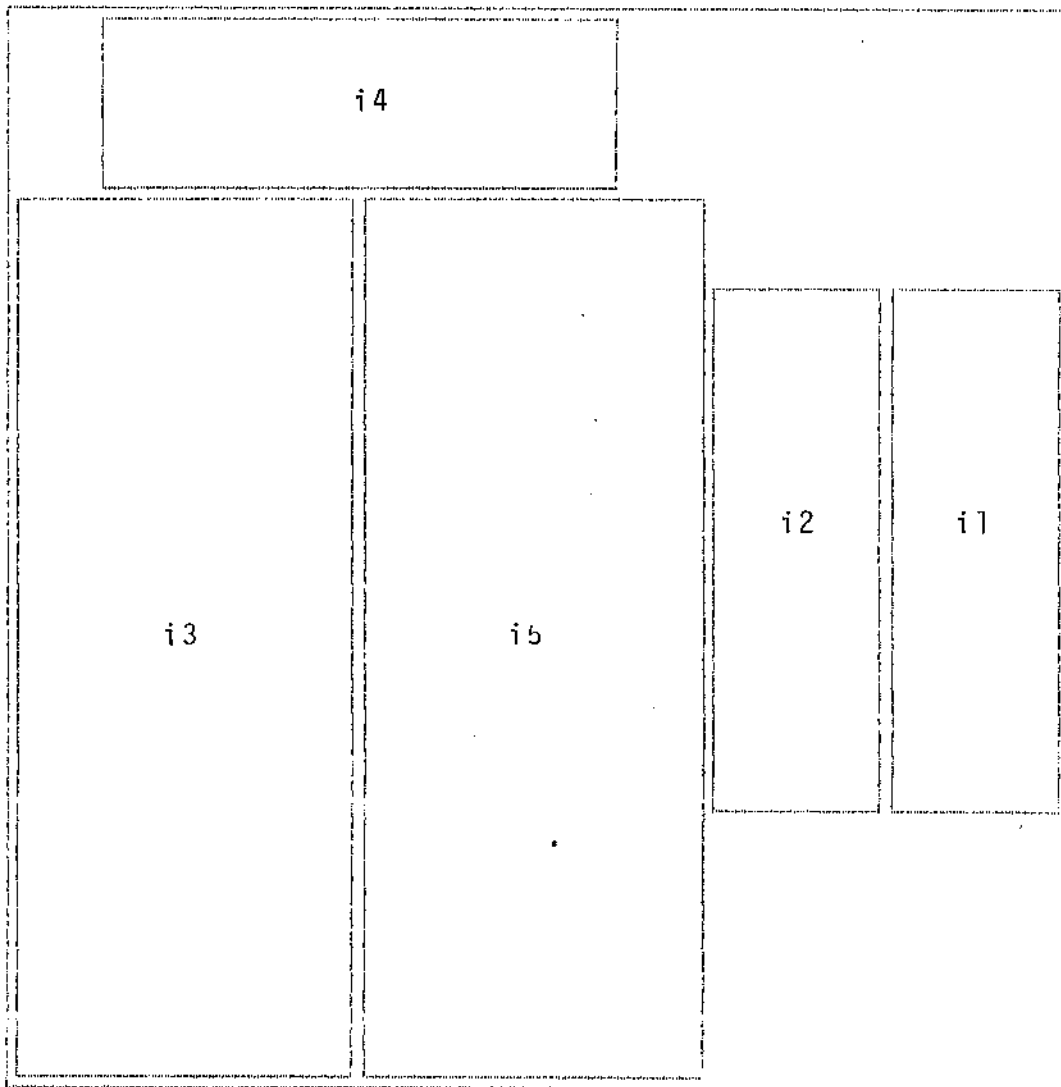
```
i1:sub1 i2:sub2 i3:sub3 i4:sub4 i5:sub5
(+2:[vdd/self,vdd/i1,vdd/i2,vdd/i3,vdd/i4,vdd/i5]+)
(+2:[gnd/self,gnd/i1,gnd/i2,gnd/i3,gnd/i4,gnd/i5]+)
(+2:[c3/i1,c3/i2,c3/i5]+)
(+2:[c1/i2,c1/i3,c1/i4]+)
(+2:[c2/i2,c2/i3,c2/i5]+)
*)
```

### B.1.2 Saída

Para este exemplo de entrada foram realizadas três execuções da ferramenta, alterando em cada uma delas apenas a razão de aparência desejada.

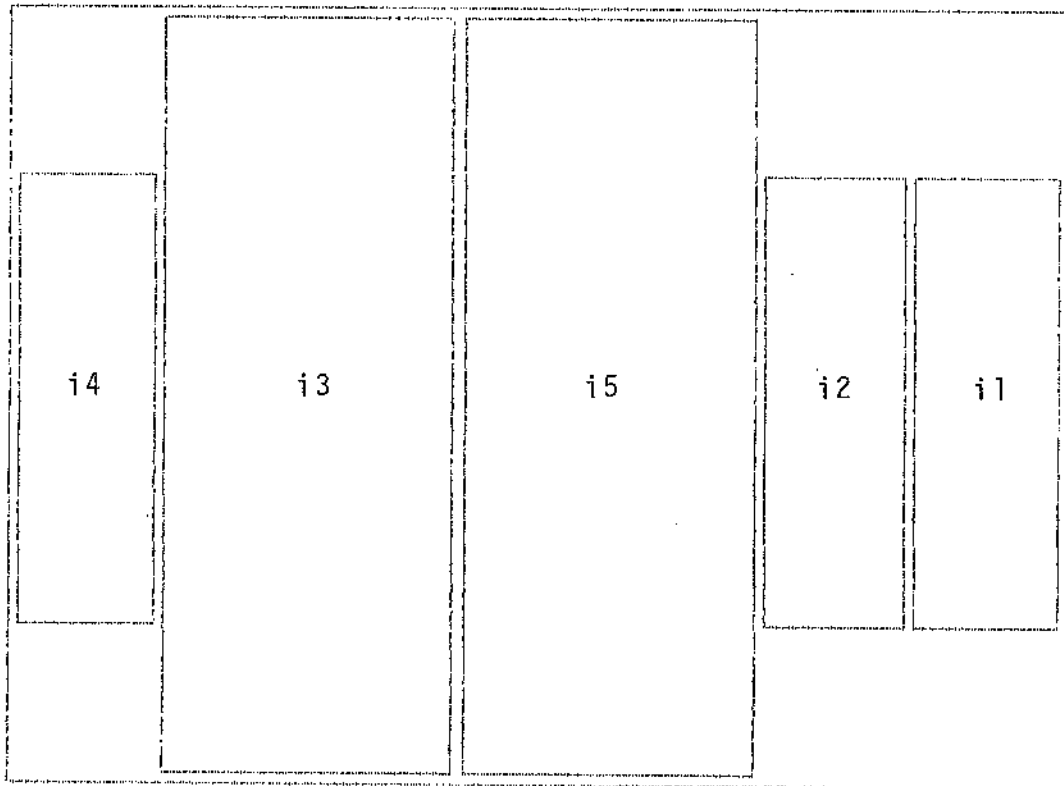
Nas expressões de posicionamento relativo aqui apresentadas o operador unário "R" precedendo o nome de uma subcélula, ou precedendo uma expressão parcial de posicionamento relativo, indica que a mesma está posicionada com uma rotação de noventa graus em relação a sua definição na descrição do circuito.

A seguir são apresentados o gráfico e a expressão de posicionamento relativo correspondente a cada uma das execuções:



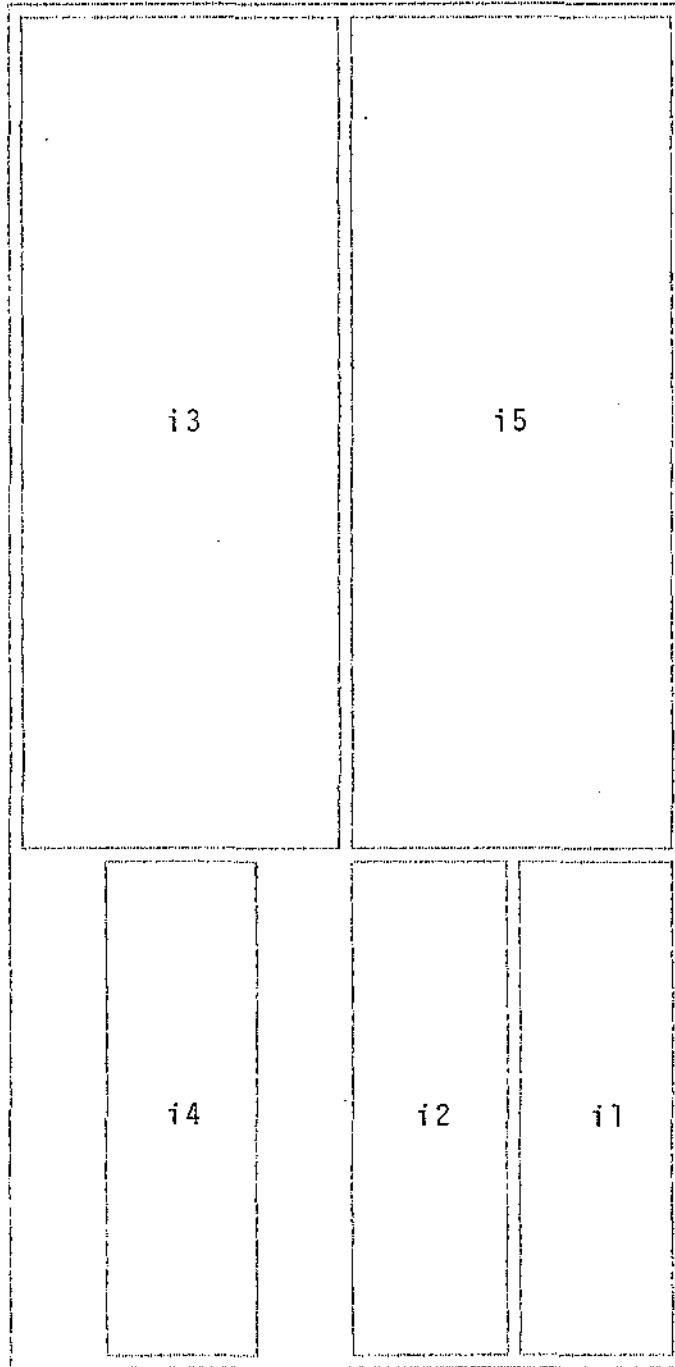
$$(( i3 | i5 ) - i4 ) | ( R i2 | R i1 )$$

Figura 18. Posicionamento com razão de aparência 1.0



R i4 | i3 | i5 | ( R i2 | R i1 )

Figura 19. Posicionamento com razão de aparência 0.8



$$( R i4 | ( R i2 | R i1 ) ) - ( i3 | i5 )$$

Figura 20. Posicionamento com razão de aparência 0.6



## B.2 Exemplo 2

### B.2.1 Entrada

```
(* master (-,-)
  [vdd(-,4,metal)]
  [p1(-,2,poly),p2(-,2,poly),p3(-,2,poly)]
  [gnd(-,4,metal)]
  [p4(-,2,poly),p5(-,2,poly),p6(-,2,poly)]

(* a (10,20)
  [gnd(2,5,metal)]
  [p1(2,15,poly),p2(2,12,poly),p3(2,8,poly),p4(2,5,poly)]
  [vdd(2,5,metal)]
  [p5(2,5,poly),p6(2,8,poly),p7(2,12,poly),p8(2,15,poly)]
  =(a.def)
*)

(* b (10,10)
  [gnd(2,5,metal)]
  [p1(2,7,poly),p2(2,3,poly)]
  [vdd(2,5,metal)]
  [p3(2,3,poly),p4(2,7,poly)]
  =(b.def)
```

```

*)
(* c (25,20)
  [gnd(2,12,metal)]
  [p1(2,15,poly),p2(2,10,poly),p3(2,5,poly)]
  [vdd(2,12,metal)]
  [p4(2,5,poly),p5(2,10,poly),p6(2,15,poly)]
  =(c.def)
*)
(* d (20,20)
  [gnd(2,10,metal)]
  [p1(2,15,poly),p2(2,10,poly),p3(2,5,poly)]
  [vdd(2,10,metal)]
  [p4(2,5,poly),p5(2,10,poly),p6(2,15,poly)]
  =(d.def)
*)
(* e (60,15)
  [gnd(2,30,metal)]
  [p1(2,11,poly),p2(2,4,poly)]
  [vdd(2,30,metal)]
  [p3(2,4,poly),p4(2,11,poly)]
  =(e.def)
*)
(* f (10,50)
  [gnd(2,5,metal)]
  [p1(2,40,poly),p2(2,25,poly),p3(2,10,poly)]
  [vdd(2,5,metal)]

```

```

    [p4(2,10,poly),p5(2,25,poly),p6(2,40,poly)]
    =(f.def)
*)
(* g (10,25)
    [gnd(2,5,metal)]
    [p1(2,20,poly),p2(2,12,poly),p3(2,5,poly)]
    [vdd(2,5,metal)]
    [p4(2,5,poly),p5(2,12,poly),p6(2,20,poly)]
    =(g.def)
*)
(* h (10,30)
    [gnd(2,5,metal)]
    [p1(2,22,poly),p2(2,15,poly),p3(2,8,poly)]
    [vdd(2,5,metal)]
    [p4(2,8,poly),p5(2,15,poly),p6(2,22,poly)]
    =(h.def)
*)
(* i (50,25)
    [gnd(2,25,metal)]
    [p1(2,20,poly),p2(2,12,poly),p3(2,5,poly)]
    [vdd(2,25,metal)]
    [p4(2,5,poly),p5(2,12,poly),p6(2,20,poly)]
    =(i.def)
*)
a1:a a2:a a3:a a4:a b1:b b2:b c1:c c2:c d1:d d2:d d3:d
e1:e f1:f g1:g h1:h h2:h i1:i

```

```

(+2:[vdd/self,vdd/a1,vdd/a2,vdd/a3,vdd/a4,vdd/b1,vdd/b2,
    vdd/c1,vdd/c2,vdd/d1,vdd/d2,vdd/d3,vdd/e1,vdd/f1,
    vdd/g1,vdd/h1,vdd/h2,vdd/i1]++)
(+2:[gnd/self,gnd/a1,gnd/a2,gnd/a3,gnd/a4,gnd/b1,gnd/b2,
    gnd/c1,gnd/c2,gnd/d1,gnd/d2,gnd/d3,gnd/e1,gnd/f1,
    gnd/g1,gnd/h1,gnd/h2,gnd/i1]++)
(+2:[p1/a1,p8/a2]++)
(+2:[p2/a1,p7/a2]++)
(+2:[p3/a1,p6/a2]++)
(+2:[p4/a1,p5/a2]++)
(+2:[p1/a2,p8/a3]++)
(+2:[p2/a2,p7/a3]++)
(+2:[p3/a2,p6/a3]++)
(+2:[p4/a2,p5/a3]++)
(+2:[p1/a3,p8/a4]++)
(+2:[p2/a3,p7/a4]++)
(+2:[p3/a3,p6/a4]++)
(+2:[p4/a3,p5/a4]++)
(+2:[p1/b1,p4/b2]++)
(+2:[p2/b1,p3/b2]++)
(+2:[p1/c1,p6/c2]++)
(+2:[p2/c1,p5/c2]++)
(+2:[p3/c1,p4/c2]++)
(+2:[p1/h1,p6/h2]++)
(+2:[p2/h1,p5/h2]++)
(+2:[p3/h1,p4/h2]++)

```

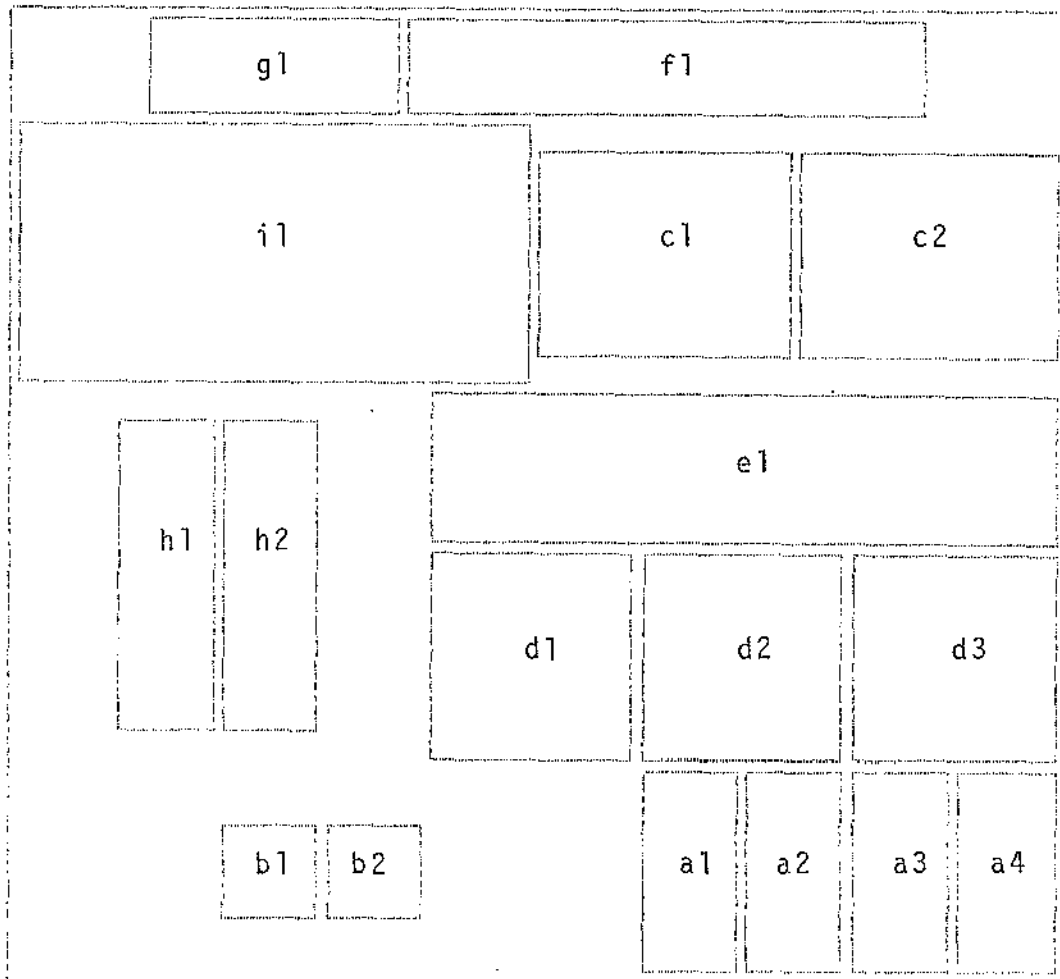
(+2:[p1/d1,p6/d2] +)  
 (+2:[p2/d1,p5/d2] +)  
 (+2:[p3/d1,p4/d2] +)  
 (+2:[p1/d2,p6/d3] +)  
 (+2:[p2/d2,p5/d3] +)  
 (+2:[p3/d2,p4/d3] +)  
 (+2:[p1/a4,p4/b1] +)  
 (+2:[p2/a4,p3/b1] +)  
 (+2:[p1/b2,p5/d1] +)  
 (+2:[p2/b2,p4/d1] +)  
 (+2:[p1/c2,p1/e1,p6/f1] +)  
 (+2:[p2/c2,p5/f1] +)  
 (+2:[p3/c2,p2/e1,p5/f1] +)  
 (+2:[p3/f1,p3/d3] +)  
 (+2:[p1/g1,p6/h1,p6/i1] +)  
 (+2:[p2/g1,p5/h1,p5/i1] +)  
 (+2:[p1/self,p6/d1,p1/i1] +)  
 (+2:[p2/self,p1/d3,p2/i1] +)  
 (+2:[p3/self,p3/i1] +)  
 (+2:[p4/self,p8/a1,p6/c1,p4/e1] +)  
 (+2:[p5/self,p5/a1,p5/c1,p6/g1] +)  
 (+2:[p6/self,p4/c1,p3/e1,p5/g1] +)

\*)

### B.2.2 Saída

Para este exemplo de entrada foram realizadas três execuções da ferramenta, alterando em cada uma delas apenas a razão de aparência desejada.

A seguir são apresentados o gráfico e a expressão de posicionamento relativo correspondente a cada uma das execuções:

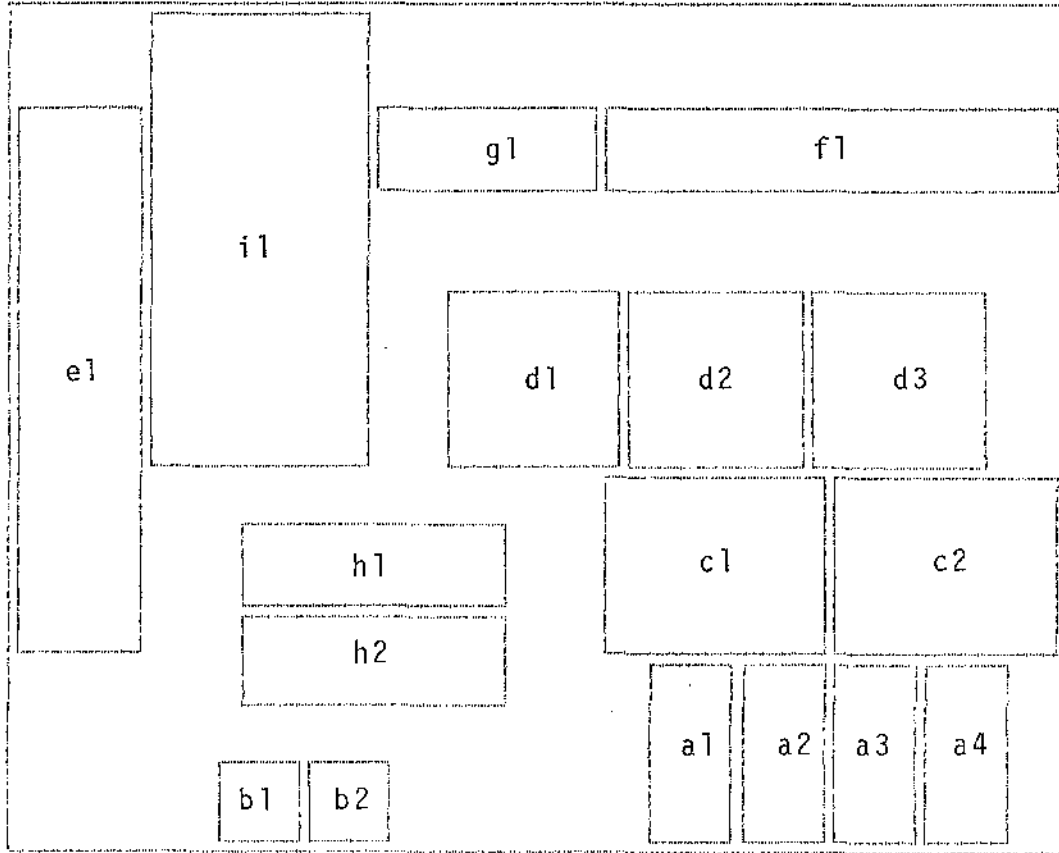


```

(( b1 | b2 ) | ( a1 | a2 | a3 | a4 )) -
(( h1 | h2 ) | ( d1 | d2 | d3 ) - e1 ) -
( i1 | ( c1 | c2 )) - ( R g1 | R f1 )

```

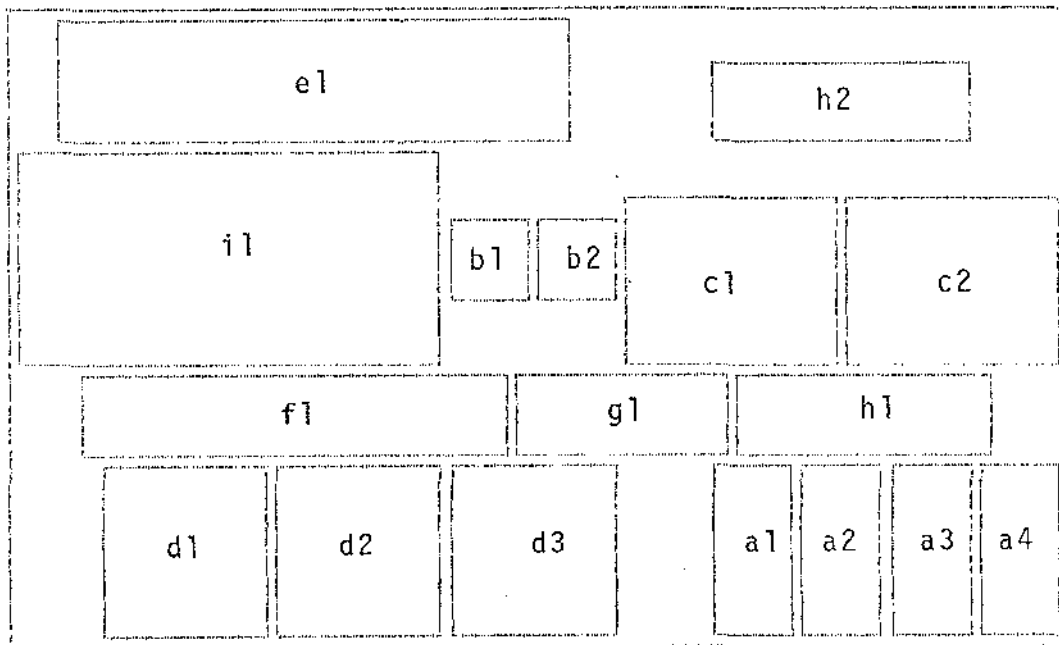
Figura 21. Posicionamento com razão de aparência 1.0



```
( R e1 | ( R ( h1 | h2 ) - ( R i1 |
(( d1 | d2 | d3 ) - ( R g1 | R f1 ))))) &
((( b1 | b2 ) - R ( h1 | h2 )) |
(( a1 | a2 | a3 | a4 ) - ( c1 | c2 )))
```

Figura 22. Posicionamento com razão de aparência 0.8





```

(( d1 | d2 | d3 ) | ( a1 | a2 | a3 | a4 )) -
( R f1 | R g1 | R h1 ) -
((( i1 | ( b1 | b2 )) - e1 ) |
(( c1 | c2 ) - R h2 ))

```

Figura 23. Posicionamento com razão de aparência 0.6

## Índice Remissivo

### A

Abordagens de projeto  
  personalizado 5  
  semi-personalizado 5  
    dispositivos lógicos 8  
    matriz de portas  
      lógicas 5  
    matrizes de componentes  
      analógicos 8  
    sistemas baseados em  
      células 6  
Algoritmo de Steinberg- 19

### C

Célula 6  
  composta 6  
  primitiva 6  
Células primitivas com altura  
  padrão 7  
Canal 16, 40  
Compilação de silício 4, 9  
Compilador de silício 21  
Complexidade de Projeto 1

### D

Descrição estrutural de  
  células 35

### E

Especificação geométrica 10  
Expressões mistas 33  
Expressão de posicionamento  
  relativo 29, 30

### F

Fileira 37

### G

Geradores de posicionamento 17  
  construtivos 17  
  por crescimento  
    epitaxial 17  
  por particionamento 17  
    randômicos 17  
Grafo de interseção de  
  canal 40  
Grafo de posição de canal  
  horizontal 26  
  vertical 26  
Grafo de posicionamento  
  relativo 26  
  propriedades 27

### H

Hierarquia de composição 24

### L

Localização alvo 19

M

Matriz de conexidade 36  
Matriz de conexidade por  
face 38  
Matrizes de portas lógicas  
posicionamento 14  
Melhoradores de  
posicionamento 17  
força dirigida 20  
força dirigida relaxada 19  
relaxação de pares de força  
dirigida 21  
troca de pares na  
vizinhança 18  
troca de pares padrão 18

N

Níveis de abstração 1  
comportamental 1  
elétrico 3  
funcional 2  
geométrico 3  
lógico 3

O

Operação de desmembramento 31  
Operação de redução de  
adjacência  
horizontal 29  
vertical 29

P

Posicionamento 13  
Posicionamento inicial 24  
geração 24, 40

melhoramento 24, 46

R

Razão de aparência 38  
Reconhecimento de  
regularidades 23, 37  
Regras de projeto 3, 14  
Roteamento 13

S

Sinal 13

T

Traçado 4, 13  
posicionamento 4  
absoluto 16  
relativo 16  
roteamento 4

U

Unidades lambda 14

V

Vértice  
fonte 24  
sorvedouro 24  
Valor de mérito 18, 43, 48