

O MÓDULO DE POSICIONAMENTO RELATIVO DO PROJETO TRANCA

MARCELO LUBASZEWSKI *

Pós-Graduação em Ciência da Computação (PGCC) - UFRGS
Av. Osvaldo Aranha, 99 - CP 1501
90.001 - Porto Alegre - RS - BRASIL
Telefone: (0055) 512-212161
Telex: (051) 2680

RESUMO

Este artigo apresenta o desenvolvimento do módulo de posicionamento relativo do Projeto TRANCA ("TRANSPARENT Cell Approach").

TRANCA trata-se de um sistema de síntese automática de layout, baseado em uma metodologia de projeto que emprega técnicas de roteamento diferentes daquelas em que "standard cell" tradicional se apóia. As interconexões entre células são realizadas sobre as áreas ativas do circuito, com vistas à eliminação de canais de roteamento e à obtenção de rotas menos resistivas.

A partir da descrição estrutural do circuito, de informações topológicas da biblioteca de células e da área reservada em planta baixa, o módulo de posicionamento relativo gera partições de células com aproximadamente a mesma área e com um número mínimo de redes comuns. Cada partição constituirá uma fileira de células na estrutura física. Maximizando o número de conexões realizadas sobre as células e minimizando o número de células de interconexão responsáveis pela comunicação entre fileiras, obtém-se um ganho considerável em área de implementação e velocidade do circuito.

O procedimento para a partição de blocos "standard cell" com transparência fundamenta-se em partição de grafos aplicada ao posicionamento de circuitos. As principais características do módulo sob investigação residem em: tratamento flexível do critério de balanceamento, realização de otimizações locais (baseadas em transferências individuais de células e alterabilidade de pesos de redes) e tempo de processamento reduzido (pela utilização de estruturas de dados redundantes e ordenações diversas).

* Engenheiro Eletrônico (UFRGS - 1986); Mestrando em Ciência da Computação (PGCC - UFRGS).

1. INTRODUÇÃO

A integração de circuitos VLSI tornou-se viável graças ao surgimento de *ferramentas computacionais para auxílio ao projeto*: à medida que aumenta a complexidade do sistema digital que se deseja implementar, tanto mais se faz necessária a *redução do tempo de projeto* e a *garantia de correção do circuito*.

Atendendo à demanda do mercado, sistemas de *síntese automática* passaram a ser desenvolvidos e têm evoluído no sentido de abranger transformações a partir de níveis de abstração cada vez maiores. O objetivo deste processo é chegar a ferramentas que possibilitem a realização física do projeto, partindo-se da especificação comportamental do circuito apresentada sob forma algorítmica.

1.1. O Sistema TRANCA

Dentro do contexto de projeto automático, o sistema TRANCA pretende contribuir com uma ferramenta de *síntese de layout de blocos "standard cell"* a partir de uma descrição a nível lógico do circuito (estrutural ou comportamental).

A figura 1 mostra os módulos que constituem o sistema TRANCA. Os módulos envolvidos por retângulos duplos são aqueles já desenvolvidos [LUB 88a] [LUB 88b] ou em desenvolvimento. As setas em destaque simbolizam os pontos de entrada no sistema.

O *estilo de layout* gerado por este sistema difere de "standard cell" tradicional pelo fato do roteamento ser realizado *sobre a área ativa do circuito*, dispensando o uso de canais dedicados a interconexões. Esta metodologia introduz *compactação de área e minimização de características elétricas indesejáveis* [REI 87] [REI 88].

Não é intenção deste artigo fazer uma análise global dos módulos que constituem o sistema TRANCA. Pretende-se, contudo, descrever as estratégias envolvidas no módulo de posicionamento relativo.

1.2. O Módulo de Posicionamento Relativo

De uma maneira geral, o posicionamento das células que constituem um bloco é dito *relativo*, quando a um conjunto de células, cujas relações de adjacência (interconexões) são conhecidas, se atribui parte de uma área sem a definição de localização de cada célula individualmente [LIE 85].

Os geradores de posicionamento existentes são baseados em diferentes tipos de algoritmos. Os mais apropriados para o problema de *alocação relativa* fundamentam-se em *partição de grafos*.

No sistema TRANCA, o objetivo é dispor de um gerador de posicionamento relativo que forneça uma *partição* de um bloco em conjuntos de células com aproximadamente a mesma área e pouco relacionados entre si. Cada um destes conjuntos constituirá uma fileira (banda) na estrutura física.

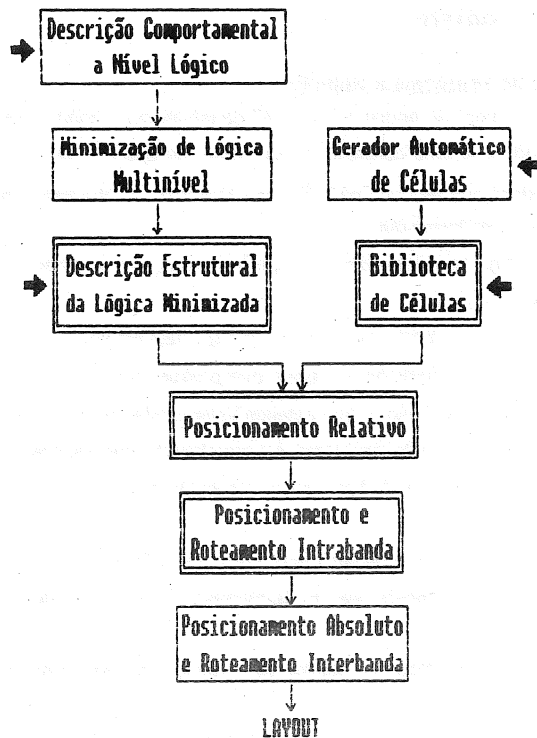


Figura 1 - O Sistema TRANCA.

Basicamente, este trabalho está dividido como se explica a seguir.

Na seção 2, o problema de partição de grafos é analisado em detalhe.

A seção 3 procura estabelecer o vínculo entre partição e posicionamento de circuitos. Analisa-se como realmente a heurística Kernighan-Lin (seção 2) tem sido empregada e como as limitações de dimensões são tratadas.

A seção 4 detém-se na solução para o problema de partição de blocos de "standard cells" com transparência [REI 87] aplicada ao seu posicionamento.

Na seção 5, descreve-se os procedimentos envolvidos na implementação do módulo de posicionamento relativo, as estruturas de dados adotadas e as características relevantes de alguns algoritmos já desenvolvidos.

Comentários relacionados à inclusão de procedimentos para a otimização global das partições aparecem na seção 6.

2. PARTIÇÃO DE GRAFOS

2.1. Definição do Problema e Soluções

Dado um grafo G , com N nodos n_i ($i=1...N$) de tamanho t_i , ligados por A arcos a_{ij} ($j=1...N$) com pesos p_{ij} , e supondo que uma condição de contorno, do tipo tamanho ($T < \text{somatório dos } t_i$) ou cardinalidade ($C < N$) desejada de cada partição, seja fornecida, o processo consiste na divisão de G em subgrafos g_k ($k \leq N$) de forma que:

(a) para qualquer g_k , somatório dos t_i (n_i pertence a g_k) $\leq T$, ou para qualquer g_k , somatório dos n_i (n_i pertence a g_k) $\leq C$; e,

(b) somatório dos p_{ij} (a_{ij} pertence a mais de um g_k) seja mínimo.

O que se deseja é gerar partições do grafo que possuam um número mínimo de arcos comuns com pesos elevados (custo de partição) e que estejam balanceadas segundo algum critério.

Uma solução exata para este problema é normalmente impraticável, uma vez que ele é de natureza combinatoria. Problemas deste tipo - sujeitos à explosão combinatoria, são conhecidos como *NP-completos* [HYA 73].

Já que a obtenção de uma solução ótima neste caso implicaria em elevadíssimo tempo de processamento, *heurísticas* parecem ser bons recursos para a produção de bons resultados rapidamente.

Várias tentativas têm sido feitas no sentido de formular procedimentos para a determinação de partições.

A *geração de soluções aleatórias*, mantendo a melhor de todas aquelas analisadas até um determinado momento e encerrando o processo ao atingir um tempo ou número de tentativas pré-especificado, apresenta uma probabilidade de sucesso, em cada tentativa, extremamente baixa [KER 70].

A *procura de aglomerados de células fortemente relacionadas* geralmente não inclui provisões para a satisfação das limitações de tamanho das partições e não sistematiza a alocação de nodos que não pertencem obviamente a nenhum aglomerado em particular.

A heurística *Kernighan-Lin*, descrita a seguir, além de prover procedimentos para os casos não tratados nas heurísticas referidas acima, apresenta uma probabilidade estatística de sucesso satisfatória. Kernighan & Lin são referenciados em toda a literatura que versa sobre partição aplicada.

2.2. A Heurística Kernighan-Lin

O problema de *partição em múltiplos subgrafos* não conta até o presente momento com nenhum método satisfatório para seu tratamento. O que se tem feito é a aplicação sucessiva de *bipartições*, especificando-se diferentes critérios de balanceamento a cada passo (quando necessário), no sentido de gerar tantos subgrafos quantos se queira e que representem boas partições quando comparados dois a dois.

Neste sentido, a heurística Kernighan-Lin, definida para bipartições, passa a ter um caráter mais abrangente.

A idéia da heurística é, partindo-se de dois subgrafos quaisquer, determinar um conjunto de nodos de cada subgrafo, tal que, transferindo-se cada um destes conjuntos para o outro subgrafo, se obtenha uma redução na inter-relação entre as partições do grafo (figura 2.1). Este procedimento é repetido até que não seja mais possível a minimização do custo.

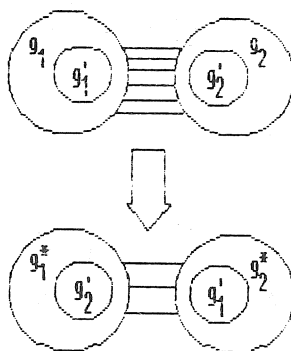


Figura 2.1-A idéia da heurística Kernighan-Lin.

Antes da apresentação detalhada da heurística, a introdução de alguns conceitos é primordial.

Custo Externo de um nodo (E) é entendido como sendo o somatório dos pesos atribuídos aos arcos do nodo que o relacionam com o subgrafo ao qual ele não pertence.

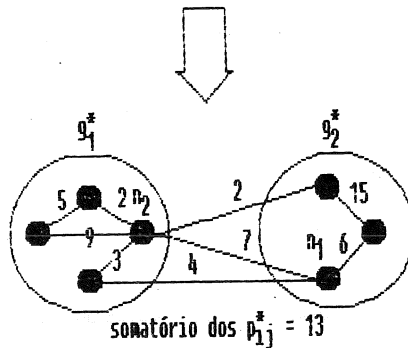
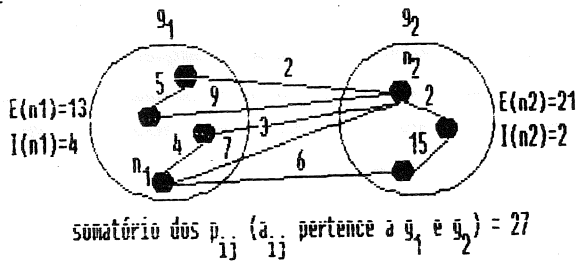
Custo Interno (I), por sua vez, é o somatório dos pesos atribuídos aos arcos que o ligam a nodos pertencentes ao mesmo subgrafo.

Suponha que g_1 e g_2 sejam uma partição do grafo e que n_1 pertencente a g_1 e n_2 pertencente a g_2 sejam dois nodos. Ganho (G) de intercâmbio de n_1 e n_2 é definido como a diferença entre o somatório dos pesos dos arcos comuns a g_1 e g_2 antes e depois da troca dos nodos. Prova-se [KER 70] que o ganho é dado por

$$G = E_{n1} - I_{n1} + E_{n2} - I_{n2} - 2p_{12}$$

onde p_{12} é o peso do arco que liga n_1 a n_2 .

Exemplo:



$$G = \text{somatório dos } p_{1j} - \text{somatório dos } p_{1j}^* = 27 - 13 = 14$$

$$G = E_{n_1} - I_{n_1} + E_{n_2} - I_{n_2} - 2 \cdot p_{12}$$

$$G = 13 - 4 + 21 - 2 - 2 \cdot 7 = 14$$

Figura 2.2 - Exemplo de troca de nós.

Podemos definir ainda o que se entende por custo de um nó (D):

$$D_{ni} = E_{ni} - I_{ni} \quad (i = 1 \dots N).$$

A seguir se apresenta o procedimento global na forma algorítmica:

1. Calcular o custo de todos os nós de g_1 e g_2 .
2. Escolher n_i pertencente a g_1 e n_x pertencente a g_2 tal que

$$G = D_{ni} + D_{nx} - 2p_{ix}$$

seja máximo.

3. Esquecer n_i e n_x temporariamente e chamá-los de n_i' e n_x' respectivamente.

4. Recalcular os custos dos nós pertencentes a $g_1 - \{n_i\}$ e $g_2 - \{n_x\}$ por

$$D_{n_i'} = D_{n_i} + 2p_{yi} - 2p_{yx} \quad n_i' \text{ pertence a } g_1 - \{n_i\}$$

$$D_{n_x'} = D_{n_x} + 2p_{zx} - 2p_{zi} \quad n_x' \text{ pertence a } g_2 - \{n_x\}$$

onde as parcelas somadas correspondem aos pesos dos arcos que passam a integrar também o outro subgrafo, e as subtraídas, aos pesos dos arcos que passam para o interior do subgrafo em questão.

5. Repetir os passos de 2 a 5, não levando em consideração os nodos anteriormente selecionados em 2, até que todos os nodos tenham sido analisados.
6. Escolher dentre os pares (n_i', n_x') a sequência (a partir do primeiro, inclusive) daqueles que, tendo seus ganhos somados, produzem um ganho total máximo.
7. Se somatório dos $G > 0$, então realizar a troca dos nodos selecionados em 6 entre g_1 e g_2 e voltar a 1.
8. Fim do procedimento.

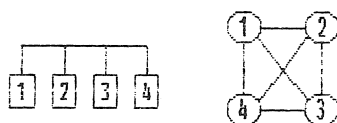
Uma medida razoável da efetividade de um procedimento é dada pela probabilidade de que ele chegue a uma solução ótima em uma única tentativa e pelo tempo de processamento envolvido. [KER 70] discute em detalhes ambos os aspectos com relação à referida heurística, chegando a uma probabilidade de $2^{-N/30}$ e tempo de máquina da ordem de $N^{2.4}$.

3. PARTIÇÃO DE CIRCUITOS

3.1. O Tratamento de Redes Múltiplas

O mapeamento de um circuito integrado em uma estrutura do tipo grafo é imediato: a *nodos* correspondem *células*, a *arcos* correspondem *grupos de conexões*, a *peso de arco* entre dois nodos corresponde o *número de conexões* entre duas células, e a *subgrafos* correspondem *subcircuitos*. Devido a esta correspondência direta, toda a teoria apresentada até então é válida para o problema de partição de circuitos aplicado ao posicionamento destes.

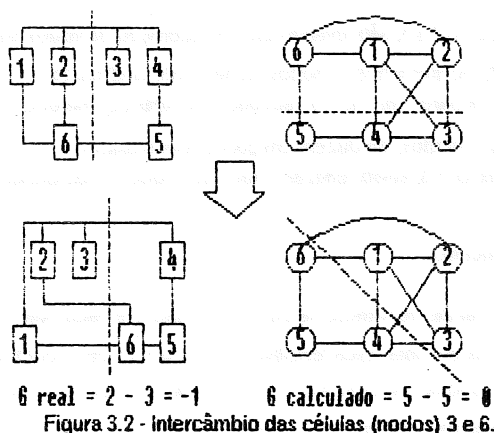
No entanto, a modelagem de circuitos por grafos deixa a desejar no tocante à representação de *conexões multiponto (redes múltiplas)*. Arcos relacionam nodo a nodo de um grafo, de forma que conexões múltiplas seriam representadas por *subgrafos completos* (subgrafos onde cada nodo se liga a todos os outros) nesta estrutura (figura 3.1).



(a)rede multiponto (b)subgrafo completo

Figura 3.1 - Representação de conexões multiponto em grafos.

Ao realizar uma troca de nodos entre partições, se pelo menos um deles faz parte de uma rede múltipla, o ganho calculado nem sempre retrata o ganho real (figura 3.2). Isto decorre do fato de uma conexão do tipo multiponto, que é implementada com um único fio, ser representada por vários arcos de pesos comparáveis àquele que corresponde à ligação física de somente duas células.



A modelagem por grafos favorece, outrossim, a formação de aglomerados de nodos que compartilham a mesma rede múltipla, o que não é sinônimo de uma boa partição (figura 3.3) [SCH 72].

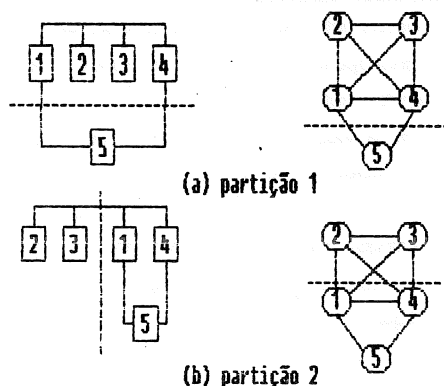


Figura 3.3 - Erro na representação na forma de grafo.

Contudo, quando se quer minimizar o custo de objetos ligados, a melhor solução é adotar uma outra representação gráfica e, principalmente, estabelecer outros critérios para o cálculo de ganho de partição. Neste caso, heurísticas que envolvem *cálculo de ganho por diferença de custos*, como aquela descrita na seção anterior, podem ser facilmente modificadas para suportar o tratamento de redes multiponto.

Para tanto, [SCH 72] propõe uma estratégia de ataque à nova situação.

O ganho passa a ser obtido por:

$$G = D_{ni} + D_{nx} - \text{fator de correção } (n_i, n_x) - \text{fator de correção } (n_x, n_i).$$

onde:

D_{ri} e D_{rx}

Passam a ser calculados segundo as seguintes regras:

Para cada rede r à qual a célula pertence:

1. Se a célula é a *única representante de r na partição*, então a parcela do seu custo relacionada a r é *positiva e de módulo igual ao peso da rede*.

2. Se a célula *não é a única da rede r na partição*, dois casos são possíveis:

(a) Se *não existem células de r na partição complementar*, então a parcela do seu custo relacionada à referida rede é *negativa e de módulo igual ao peso de r* .

(b) Se *existem células de r na outra partição*, então a parcela em questão é *nula*.

fator de correção (n_i, n_x) [fator de correção (n_x, n_i)]

Prevê a possibilidade de n_i e n_x estarem conectados. Se n_i (n_x) é o único elemento de uma determinada rede r em g_1 (g_2), e se n_x (n_i) também está em r , D_{ri} (D_{rx}) contém uma parcela igual ao *peso da rede* associada ao movimento de n_i (n_x) para g_2 (g_1), que claramente não será computada se n_x (n_i) for simultaneamente movido para g_1 (g_2). Este fator simplesmente subtrai o *peso de r* do ganho nestas situações.

O recálculo sugerido pelo passo 4 da heurística Kernighan-Lin (seção 2.3) passa a ser realizado por:

$$D_{ny}' = D_{ny} + [\text{fator de correção } (n_y, n_i) + \text{fator de correção } (n_i, n_y)] \text{ depois da troca} \\ - [\text{fator de correção } (n_y, n_x) + \text{fator de correção } (n_x, n_y)] \text{ antes da troca}$$

onde n_y pertence a $g_1 - \{n_i\}$;

$$D_{nz}' = D_{nz} + [\text{fator de correção } (n_z, n_x) + \text{fator de correção } (n_x, n_z)] \text{ depois da troca} \\ - [\text{fator de correção } (n_z, n_i) + \text{fator de correção } (n_i, n_z)] \text{ antes da troca}$$

onde n_z pertence a $g_2 - \{n_x\}$.

Evidentemente, a implementação de tais critérios é onerosa em termos de tempo de processamento. Todavia, a garantia de melhores resultados [SCH 72] é compensadora.

3.2. Critérios de Balanceamento

Até agora não foi fornecido nenhum detalhe com respeito aos critérios de balanceamento de uma partição. Como já citado, estes podem ser basicamente de dois tipos: *cardinalidade* ou *tamanho das partições*.

Em se tratando de cardinalidade, desde que as trocas de células sejam realizadas aos pares, o equilíbrio inerente às partições iniciais não é jamais perturbado.

Se o critério for tamanho dos subcircuitos, um cuidado maior deve ser dispensado. Neste caso, a cada nova iteração, antes de efetuar o intercâmbio, a condição de equilíbrio deve ser avaliada para tal operação. O que normalmente se faz é estabelecer previamente uma *razão* do tipo

$$|g_1| / (|g_1| + |g_2|) \approx q, \text{ onde}$$

$|g_k|$ é o somatório das larguras das células que constituem o subcircuito k .

Uma vez que a razão na forma como é proposta não é determinística ($\sim$), deve-se especificar o intervalo de tolerância admissível para o desvio da igualdade. [FID 82] sugere

$$q.W - t_{\max} \leq |g_1| \leq q.W + t_{\max}$$

onde $W = |g_1| + |g_2|$ e $t_{\max} = \max(t_i)$, e adverte que este critério funciona bem em casos onde a variação nos tamanhos das células não é muito grande. Desta forma, é permitido a *todas* as células moverem-se à partição complementar, desde que o balanço seja respeitado.

Evidentemente, o fato de respeitar sempre tal critério é mais um fator limitante na busca de uma solução ótima.

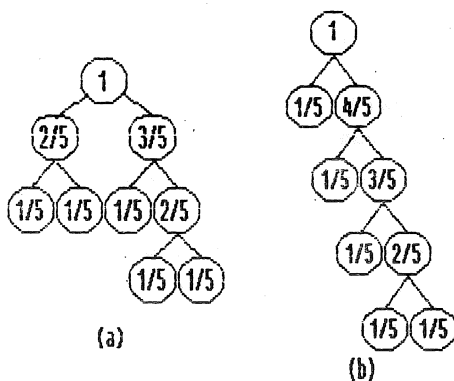
3.3. Formas de Implementação

Providas soluções para as deficiências e indefinições do procedimento de partição de gratos aplicado a circuitos, passemos à análise geral das implementações que se valem dos princípios até agora estabelecidos.

Quando a questão é posicionamento relativo, o problema se limita a quantas partições realizar e como fazê-lo.

Uma solução é partir de tantos subcircuitos quantos desejados e submetê-los *dois a dois* ao procedimento descrito, até que melhorias não sejam mais possíveis em nenhum par. Novamente se recai no problema de explosão do número de tentativas. Se o número de partições imposto é elevado, o número de vezes de realização do processo (no mínimo $OP/2$, onde P é o número de subcircuitos) pode se tornar muito grande.

Outra alternativa - a mais utilizada, seria a realização de *bipartições sucessivas*. Neste caso, o número de execuções do processo é bastante mais reduzido ($P - 1$). Ainda, as formas de implementação podem ser distintas:



5 partições

Figura 3.4 - Formas de implementação de bipartições sucessivas.

Computacionalmente falando, a alternativa (b) é bastante mais fácil de implementar. Com relação à efetividade de uma e outra versão, nem a literatura analisada apresenta estudos, nem tampouco é intuitivo discursar a respeito.

No posicionamento absoluto, o *aspecto geométrico* desponta como dominante. Aqui são definidas sucessivamente *linhas imaginárias* que cortam uma estrutura física, e os esforços são direcionados no sentido de minimizar o número de conexões perpendiculares a tais linhas. O controle do critério de balanceamento torna-se crítico. Este procedimento é conhecido na literatura como posicionamento "*min-cut*". [COR 79] descreve em detalhes esta classe e tece comentários a respeito de resultados obtidos utilizando-se diferentes estratégias.

Da análise dos pontos apresentados nesta seção e daqueles anteriormente estudados, fica evidente que, em se tratando de *partição*, *nada é determinismo, tudo é heurística*.

4. POSICIONAMENTO RELATIVO DE "STANDARD CELLS" COM TRANSPARÊNCIA

4.1. A Metodologia de Projeto

A metodologia de projeto baseia-se em quatro pontos fundamentais: *estrutura de banda*, *transparência horizontal e vertical*, *gerenciamento de trilhas* e *maleabilidade*.

Nesta seção, serão discutidos apenas os aspectos que influenciam o tratamento de posicionamento via *partição*, ou seja, *transparência* aplicada ao roteamento. Detalhes sobre a metodologia podem ser encontrados em [REI 87] e [REI 88].

As células são projetadas de forma a disporem de trilhas livres, paralelas às linhas de alimentação, por onde as conexões externas são implementadas em *metal*. A *transparência horizontal* está relacionada ao número de trilhas que podem atravessar a célula.

O roteamento interno a uma banda é, portanto, realizado sobre as células, dispensando o uso de canais dedicados a conexões (figura 4.1).

O roteamento entre bandas usa células de interconexão [REI 87] para a passagem de sinais. Estas células - implementadas em *polissilício*, são posicionadas (uma em cada banda) entre células funcionais e resgatam os sinais de interesse, passando-os às bandas adjacentes. As conexões são realizadas verticalmente às linhas de alimentação neste caso (figura 4.1).

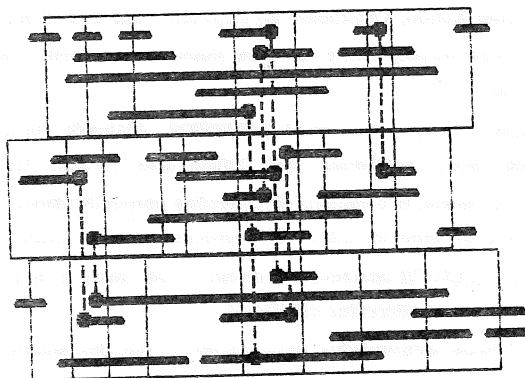


Figura 4.1 - Roteamento sobre as bandas e utilizando células de interconexão.

O objetivo de se utilizar um processo de partição para determinar que células devem compor que bandas reside em:

- (a) maximizar o número de conexões realizadas em metal (sobre a banda), visando a obtenção de rotas menos resistivas; e,
- (b) minimizar o número de células de interconexão, pela redução do número de redes comuns às bandas, visando a otimização de área de implementação.

4.2. Estratégias de Partição

A solução adotada é a de realização de *bipartições sucessivas*. Procura-se compensar as limitações inerentes à ação de divisão-e-conquista associada a este procedimento, pelo acréscimo de provisões para a otimização das partições aos pares. Para tanto, os pesos das redes múltiplas do tipo interface são reavaliados antes que inicie a próxima partição.

Os subcircuitos são gerados *em fatis*, ou seja, o critério de balanceamento utilizado faz com que o resultado de cada partição seja uma *banda* e um *bloco residual*. Este bloco pode ser ou a última banda gerada pelo processo, ou um subcircuito a particionar. Esta estratégia introduz maior simplicidade de processamento, uma vez que inexistem blocos pendentes na fila de espera do processo. Além do mais, somente nestas condições se torna viável a otimização das partições aos pares citada anteriormente e descrita mais adiante.

Com relação a uma bipartição, a otimização é realizada pela permissão de *transferências individuais de células* quando as trocas aos pares não forem mais possíveis.

4.3. Descrição do Procedimento

As informações necessárias para dispor um processo de partição provêm da *descrição estrutural* do circuito [WAG 87b], da *biblioteca de células* [LUB 88b] e do *local de destino do bloco na planta baixa*.

A extração da conectividade é feita a partir da descrição estrutural do circuito. Para cada *célula*, são obtidos *nome*, *redes conexas* e *células vizinhas*; para cada *rede*, devem ser conhecidos o seu *peso* e as *células* que estão a ela conectadas. Estes dados são do tipo *estático*, isto é, permanecem inalterados durante todo o processo.

No decorrer do procedimento, informações *dinâmicas* são geradas e agregadas às referidas estruturas. Alguns exemplos são: *partição* à qual a célula pertence, *custo* de célula, *apontadores de ordenação de custos* e outras.

Da biblioteca deseja-se a informação estática de *largura* de célula, enquanto que, da planta baixa, se quer saber o *formato* aproximado do bloco a posicionar. Este último é fornecido sob a forma de *relação de aspecto (RA)*, ou seja, razão altura/largura do bloco.

De posse de todos os dados, o primeiro passo consiste no cálculo do número de bandas a gerar ($nb = \text{número de bipartições} + 1$). Isto é feito como se mostra a seguir:

$$RA = \text{altura} / \text{largura} = (nb \cdot k_1) / [(1+k_2) \cdot W / nb] \quad (I), \text{ onde}$$

k_1 é uma constante - dependente de tecnologia, função da *altura padrão* das células e da *distância mínima* entre bandas; e,

k_2 é uma constante - função de estudos estatísticos, que fornece o acréscimo percentual de área pela *inclusão de células de interconexão* no circuito.

De (I) vem que:

$$nb = [(1+k_2) \cdot W \cdot RA / k_1]^{1/2} \quad (II).$$

Supondo que o valor de k_2 seja super-estimado para a maioria dos casos, nb - dado por (II), é truncado para resultados fracionários.

A rigor, k_1 pode ser inferior ao valor previsto quando da *partição* resultam bandas idênticas (*regularidade*), pois, neste caso, as linhas de alimentação de mesmo potencial de bandas adjacentes poderiam ser sobrepostas [LUB 88b]. Na falta de previsões, prefere-se assumir que isto não ocorre.

O segundo passo é a determinação da *largura máxima* de célula (l_{max}) para o estabelecimento do critério de balanceamento da bipartição. Este passo é repetido a cada nova bipartição, visando a redução da probabilidade da última banda (gerada como bloco residual) não respeitar o critério (ser maior ou menor que os limites estabelecidos).

Neste ponto, a inequação de equilíbrio fica perfeitamente definida, já que $q = 1 / nb$ e l_{max} é conhecido.

O estabelecimento das *partições iniciais* precede o processo de bipartição propriamente dito. Este é realizado por *pesquisa sequencial*, procurando aproveitar a tendência do projetista de descrever o circuito agrupando as células de intensa conectividade entre si. Esta atitude deve reduzir o número de trocas de células nas etapas seguintes.

Chega-se, então, ao centro do problema: a execução da heurística Kernighan-Lin. Neste passo, são acrescentadas algumas *provisões* e são feitas *modificações*:

(a) a *ordenação de custos* é feita paralelamente ao cálculo destes [FID 82], o que resulta em economia de tempo na pesquisa do par que maximiza o ganho;

(b) o *recálculo dos custos*, após a escolha de cada par, é realizado somente para as células vizinhas daquelas que constituem o par e não envolve procedimentos que determinem fatores de correção;

(c) a *verificação do equilíbrio (balanço)* entre partições é realizada somente quando a proposta do conjunto de pares a serem transferidos está disponível, possibilitando que as trocas se compensem entre si;

(d) a *transferência individual de células* [FID 82], desde que trocas aos pares não sejam mais proveitosas e que o balanço seja mantido, otimiza a partição.

Caso a compensação citada em (c) não ocorra, os *últimos* pares selecionados são retirados sucessivamente do conjunto, até que o critério passe a ser respeitado.

O objetivo básico de (d) é tratar as células de custo positivo, que não foram movidas pelo fato de constituírem um par que não respeitava o equilíbrio, no sentido de transferi-las, uma a uma, realizando o teste do critério de balanceamento para *cada* caso. Em última análise, o que se tenta fazer é um rearranjo dos pares de forma a manter o balanço.

Entre duas bipartições, ocorre a *modificação dos pesos das redes multiponto da interface* (que inicialmente podem possuir pesos superiores a 1, caso sejam consideradas *críticas* por imposição do projeto). A estas redes são atribuídos pesos sucessivamente maiores, à medida que elas se propagam à banda e ao bloco residual. O valor que se acresce ao peso é função do aumento do número de conexões entre as bandas anteriormente geradas e os blocos resultantes da partição, caso a rede não se limite a um subcircuito desta (figura 4.2).

A exceção à regra é a atribuição de peso nulo às redes (simples ou múltiplas) da interface que possuem uma única célula no bloco residual. Neste caso, a existência de tal rede é irrelevante para a próxima bipartição.

Esta atitude aumenta significativamente a probabilidade destas redes se restringirem ou à banda, ou ao bloco residual, o que resulta em minimização do número de conexões multibanda.

Por fim, a *dissociação da banda gerada* é uma ação importante, no sentido dos passos seguintes manipularem estruturas de dados cada vez menores. A informação de peso das redes múltiplas da interface é a única que precisa prosseguir.

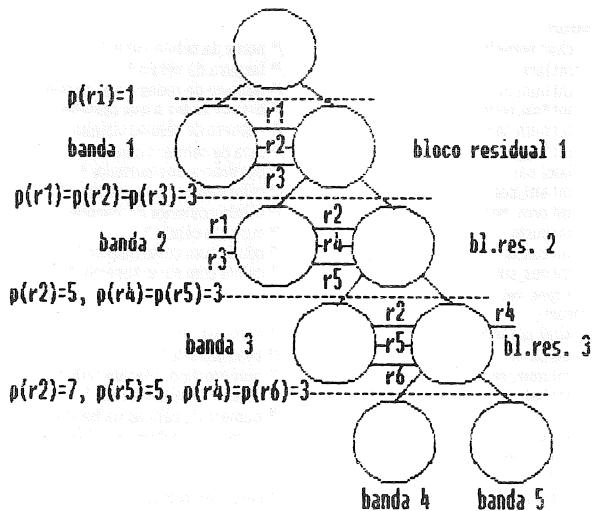


Figura 4.2 - Alteração dos pesos das redes múltiplas da interface.

5. IMPLEMENTAÇÃO

5.1. Estruturas de Dados

A escolha das estruturas apropriadas para o armazenamento de dados em um sistema é fator determinante na performance deste. Tanto que esta escolha deve ser baseada nas operações às quais os dados são submetidos durante o processo.

Em aplicações onde a velocidade de processamento é crítica, a dissociação dos dados em várias estruturas, introduzindo *redundância* de informação, apesar de onerar em termos de memória, reduz significativamente a *profundidade de laços* (laços internos a laços).

Na fase de especificação destas estruturas, a configuração da máquina que se dispõe e a quantidade e tipo de informações que serão manipuladas devem ser analisados cuidadosamente, na busca de um compromisso *memória x velocidade* que minimize as limitações do sistema.

Dentro desta filosofia, optou-se por quatro estruturas de dados para o desenvolvimento do módulo de posicionamento relativo: *vetor de células*, *vetor de redes*, *lista de pares selecionados* e *vetor de bandas geradas*. Os campos que compõem estas estruturas aparecem na figura 5.1.

```

typedef struct {
    char nome[9];           /* nome da célula - gate */
    int larg;               /* largura da célula */
    int num_red;            /* número de redes a que pertence */
    int *ap_redes;          /* lista de redes a que pertence */
    int num_viz;            /* número de células vizinhas */
    int *ap_viz;            /* lista de células vizinhas */
    bool part;              /* partição a que pertence */
    int ant_part;           /* célula anterior da partição */
    int prox_part;          /* célula posterior da partição */
    int custo;              /* custo da célula */
    int cus_inf;            /* célula com custo inferior */
    int cus_sup;            /* célula com custo superior */
} type_cel;

typedef struct {
    char nome[9];           /* nome da rede */
    int peso;               /* peso da rede */
    int num_cel;            /* número de células da rede */
    int *ap_cel;            /* lista de células da rede */
    int kb;                 /* número de células na banda */
    int lr;                 /* número de células no bloco residual */
} type_red;

typedef struct {
    int cel_ban;            /* célula da banda */
    int cel_res;            /* célula do bloco residual */
    int ganho;              /* ganho da troca */
    int dif_larg;           /* diferença de largura da banda */
} type_par;

typedef struct {
    int prim_ban;           /* índice da primeira célula da partição */
    int type_ban;
}

```

Figura 5.1 - Campos das estruturas de dados.

Observe que a redundância de informação está presente em vários níveis. Alguns exemplos são:

- as redes às quais uma determinada célula pertence poderiam ser determinadas percorrendo todo o vetor de redes e verificando se a célula está presente na lista de células de cada rede;
- as células vizinhas seriam obtidas percorrendo-se a lista de células de cada rede à qual a célula de referência pertence.

A partir destes exemplos, pode-se verificar o quão intrincadas seriam as operações de pesquisa. Pode-se, também, ter uma idéia da quantidade de memória que seria necessária para a codificação do procedimento e do tempo de processamento envolvido.

Observe, ainda, que alguns campos são utilizados para o *encadeamento* do vetor de células segundo critérios distintos: ordenação de células e de custos de células que pertencem à mesma partição. A ordenação de custos facilita a pesquisa dos pares que maximizam o ganho, enquanto que a ordenação de células simplifica o vetor de partições geradas, que só necessita a informação de índice da primeira célula da partição.

Neste ponto, cabe salientar que a introdução da lista de redes que estabelecem a vizinhança, para cada célula da lista de vizinhas, reduziria de 1 a profundidade de laços do procedimento, acelerando sobremaneira a execução do processo. No entanto, o número limite de células de um circuito a particionar, que possa ser tratado por uma máquina do tipo IBM-PC compatível, decresceria significativamente.

5.2. Algoritmos

Nesta seção, apresenta-se o algoritmo a nível de sistema e são feitos comentários a respeito de procedimentos relevantes de alguns subsistemas.

A figura 5.2 sintetiza todos os procedimentos envolvidos no módulo de posicionamento relativo.

1. ANÁLISE SINTÁTICA E SEMÂNTICA DA LINGUAGEM DE DESCRIÇÃO DE HARDWARE (LDH) --> GERAÇÃO DAS ESTRUTURAS INTERNAS.
2. CONSULTA À BIBLIOTECA DE CÉLULAS.
3. ENTRADA DA RELAÇÃO DE ASPECTO.
4. DETERMINAÇÃO DO NÚMERO DE PARTIÇÕES.
5. DETERMINAÇÃO DO CRITÉRIO DE BALANCEAMENTO.
6. DETERMINAÇÃO DAS PARTIÇÕES INICIAIS.
7. CÁLCULO E ORDENAÇÃO (DENTRO DA PARTIÇÃO) DOS CUSTOS DE TODAS AS CÉLULAS.
8. FORMAÇÃO DE PARES DE GANHO MÁXIMO E SELEÇÃO DA SEQUÊNCIA DE PARES QUE MAXIMIZA O GANHO GLOBAL.
9. VERIFICAÇÃO DO CRITÉRIO DE BALANCEAMENTO E TRATAMENTO DE DESEQUILÍBRIO.
10. TROCA DAS CÉLULAS ENTRE PARTIÇÕES.
11. REALIZAÇÃO DOS PASSOS DE 7 A 11 ENQUANTO HOUVER PAR COM GANHO POSITIVO.
12. TRANSFERÊNCIAS INDIVIDUAIS DE CÉLULAS.
13. ALTERAÇÃO DOS PESOS DAS REDES DA INTERFACE E DISSOCIAÇÃO DA BANDA GERADA.
14. REALIZAÇÃO DOS PASSOS DE 5 A 14 ENQUANTO HOUVER BLOCOS A PARTICIONAR.
15. SAÍDA DAS BANDAS GERADAS.

Figura 5.2 - Algoritmo a nível de sistema.

A LDH referida no passo 1 trata-se de NILO [WAG 87b], uma linguagem estrutural no nível de portas lógicas, constante no sistema AMPLO [WAG 87a] em desenvolvimento no PGCC-UFRGS. Esta escolha foi fruto, principalmente, do fato de se dispor de outra ferramenta para suporte ao projeto (*simulador lógico*), e de NILO possibilitar uma *descrição hierárquica* no nível de REDES [WAG 87a]. Posteriormente, pretende-se adotar uma outra LDH que permita a descrição mista de portas lógicas e células que se enquadram no nível de transferências entre registradores (flip-flops, multiplexadores, decodificadores, ...).

A determinação das partições iniciais (passo 6) é regida pela definição das células que constituem a banda. A pesquisa sequencial é encerrada quando se atinge o valor mínimo para o módulo da diferença entre o tamanho da banda e o equilíbrio ($1/nb \cdot W$).

A ordenação de custos (passo 7) é realizada por um algoritmo de ordem N^2 . A idéia é evoluir no sentido de ordenar os custos sobre uma *estrutura do tipo árvore*, o que reduziria a complexidade do procedimento para $O(N \log N)$.

A formação de pares de ganho máximo (passo 8) deve levar em consideração os *fatores de correção das células aos pares*. Para tanto, as células que concorrem à seleção são aquelas que possuem custo superior à diferença entre o maior custo da partição à qual pertence e a soma dos fatores

de correção das células de maior custo na banda e no bloco residual. Isto diminui consideravelmente o número de células candidatas à formação do par.

Para o tratamento de desequilíbrio (passo 9), a grandeza *desvio* é definida como sendo a distância entre o tamanho da banda, se todos os pares selecionados fossem trocados, e o limite mais próximo do critério. Cada par selecionado, cuja alteração do tamanho da banda coopera com a existência do desvio, é retirado do conjunto sucessivamente (do último ao primeiro), até que o valor atualizado do desvio se torne nulo ou negativo. Neste ponto, ficam definidos os pares que respeitam o critério de balanceamento.

A estratégia para a realização de transferências individuais de células (passo 12) baseia-se na utilização das mesmas rotinas desenvolvidas para trocas aos pares, incluindo-se provisões para o *teste do critério concorrente à seleção*.

Para maiores informações, o autor coloca a disposição dos interessados os algoritmos a nível de subsistema e o programa fonte do módulo de posicionamento relativo.

5.3. Estado Atual do Projeto

Uma primeira versão do módulo de posicionamento relativo está sendo depurada e deve tornar-se disponível muito em breve.

Esta versão foi escrita em Linguagem C e roda em máquinas do tipo IBM-PC compatível. Durante o projeto do software, teve-se o cuidado de gerar uma versão em C *padrão*, com vistas a manutenção da característica de *portabilidade* da linguagem. A próxima versão deve ser implantada em uma estação de trabalho baseada no supermicro ED680 - UCP MC68000, sistema operacional UNIX.

Alguns circuitos já foram submetidos aos procedimentos descritos. Os resultados são alentadores, tanto em termos de produto do sistema, como em termos de tempo de execução. Contudo, ainda é cedo para apresentar estatísticas que comprovem a boa performance da ferramenta, mesmo porque o projeto encontra-se na fase inicial de validação.

O que se pode afirmar, atualmente, com relação à complexidade do algoritmo implementado, é que esta não é de ordem superior a N^2 . Tão logo o sistema tenha sido validado, uma análise detalhada de complexidade do algoritmo será realizada.

6. FUTURAS VERSÕES

Como a grande maioria das ferramentas baseadas em partição esbarram em soluções do tipo *mínimo local*, faz-se necessária a introdução de uma etapa de *otimização global*, cujo objetivo é atingir uma solução melhor que aquela fornecida pelo gerador de posicionamento.

Algumas propostas interessantes de procedimentos para otimização global são:

(a) repetir o procedimento Kernighan-Lin para várias partições iniciais diferentes e adotar a melhor solução;

(b) realizar bipartições sobre as partições iniciais e recombina as quatro partições geradas em outras duas, submetendo-as, após, ao procedimento Kernighan-Lin [KER 70]; e,

(c) submeter as partições de saída do gerador de posicionamento a um algoritmo de relaxação baseado em "simulated annealing", como aquele apresentado em [SEC 85].

A primeira versão do módulo de posicionamento relativo, que abrange procedimentos para otimizações locais, mantém em aberto a etapa de otimização global das partições. Uma futura versão deve incluir provisões neste sentido. Contudo, muita pesquisa deve ser realizada na área, e um bom guia de estudos parece ser aquele indicado pelas alternativas citadas acima.

7. CONCLUSÕES

As contribuições científicas do projeto TRANCA como um todo residem em:

(a) utilizar uma nova metodologia voltada para projeto automático e baseada em um estilo de layout intermediário entre "standard cell" tradicional e "full-custom";

(b) introduzir novos conceitos e concepções em termos de posicionamento e roteamento de células, advindos das vantagens e restrições da própria metodologia de projeto em uso;

(c) avançar no sentido de possibilitar o projeto automático em níveis cada vez mais abstratos de especificação.

Dentro do contexto TRANCA, o módulo de posicionamento relativo contribui de maneira significativa com o item (b) citado acima. Mais especificamente, a colaboração deste módulo apóia-se em:

(a) busca de soluções que reduzam o tempo de processamento envolvido na execução do procedimento de bipartição;

(b) tratamento flexível do critério de balanceamento;

(c) introdução de otimizações locais na heurística Kernighan-Lin adaptada ao tratamento de circuitos;

(d) evolução no sentido de agregar ao sistema procedimentos para otimização global.

8. AGRADECIMENTOS

Este trabalho está sendo realizado sob a orientação do Professor Dr. Ricardo Reis, com o suporte de programação de André Baggio e com o apoio financeiro do CNPq - órgão brasileiro de fomento à pesquisa.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- [COR 79] CORRIGAN, L.I.. A Placement Capability Based on Partitioning. In: 16th Design Automation Conference, 1979, pp. 408-13.
- [FID 82] FIDUCCIA, C.M. & MATHEYES, R.M.. A Linear-Time Heuristic for Improving Network Partitions. In: 19th Design Automation Conference, 1982, pp. 175-81.
- [HYA 73] HYAFIL, L. & RIVEST, R.L.. Graph Partitioning and Constructing Optimal Decision Trees are Polynomial Complete Problems. In: Rep. 33, IRIA-Laboria, 1973.
Apud: DUNLOP, A. E. & KERNIGHAN, B. W.. A Procedure for Placement of Standard-Cell VLSI Circuits. In: IEEE Transactions on CAD, Vol. CAD-4, No. 1, January 1985, pp. 92-8.
- [KER 70] KERNIGHAN, B.W. & LIN, S.. An Efficient Heuristic Procedure for Partitioning Graphs. In: Bell System Technical Journal, Vol. 49, Feb. 1970, pp. 291-308.
- [LIE 85] LIESENBERG, H.K.D.. A Layout Module for a Silicon Compiler. Thesis submitted to the University of Newcastle-Upon-Tyne for the degree of doctor of philosophy. Electrical and Electronic Engineering, University of Newcastle-Upon-Tyne, April 1985, pp. 72-104.
- [LUB 88a] LUBASZEWSKI, M.S. et alii. Uma Ferramenta para o Projeto de Circuitos "Standard Cell". In: III Simpósio Brasileiro de Concepção de Circuitos Integrados, Gramado, RS, Brasil, 13-15 Abril 1988, pp. 173-82.
- [LUB 88b] LUBASZEWSKI, M. S. et alii. A Biblioteca de Standard Cells do Projeto TRANCA. In: III Congresso da Sociedade Brasileira de Microeletrônica, São Paulo, SP, Brasil, Julho 1988.
- [REI 87] REIS, R.A.L.. A New Standard Cell CAD Methodology. In: IEEE Custom Integrated Circuits Conference, Portland, USA, May 1987.
- [REI 88] REIS, R.A.L., GOMES, R.F. & LUBASZEWSKI, M.S.. An Efficient Design Methodology for Standard Cell Circuits. In: IEEE International Symposium on Circuits and Systems '88, Helsinki, Finland, June 1988.
- [SCH 72] SCHWEIKERT, D.G. & KERNIGHAN, B.W.. A Proper Model for the Partitioning of Electrical Circuits. In: 9th Design Automation Workshop, June 1972, pp. 57-62.
- [SEC 85] SECHEN, C. & SANGIOVANNI-VINCENTELLI, A.. The TimberWolf Placement and Routing Package. In: IEEE Journal of Solid-State Circuits, Vol. SC-20, No. 2, April 1985, pp. 510-22.
- [WAG 87a] WAGNER, F. R., FREITAS, C. M. D. S. & GOLENDZINER, L. G.. Linguagens de Descrição de Hardware para Suporte à Integração do Processo de Projeto em AMPLO. In: Relatório de Pesquisa 65, Pós-Graduação em Ciência da Computação/UFRGS, Porto Alegre, RS, Brasil, Março 1987.
- [WAG 87b] WAGNER, F. R. & FREITAS, C. M. D. S.. NILO - Uma Linguagem para a Descrição de Hardware no nível de Portas Lógicas. In: Relatório de Pesquisa 66, Pós-Graduação em Ciência da Computação/UFRGS, Porto Alegre, RS, Brasil, Março 1987.