

Roteamento ETSPR para canais QCL baseado em SRRP

Glauco Borges Valim dos Santos¹
Cândice Corradi Kindel²
Marcelo de Oliveira Johann³

Resumo — Este trabalho investiga a redução do número de vias no modelo de canal utilizado nas matrizes *gate array* do projeto ÁGATA, utilizando soluções do Problema de Roteamento de Uma Única Linha (SRRP – *Single Row Routing Problem*), há tempos muito utilizado no roteamento de circuitos impressos. Tal redução no número de vias implica em um modelo de Roteamento Planar de dois Lados Equipotenciais (ETSPR – *Equipotential Two-Sided Planar Routing*), o que permite a aplicação de “traduções” de soluções SRRP. A personalização das matrizes se dá em uma única camada de metal, forma de implementação que é conhecida como QCL (*Quickly Customized Logic*). O primeiro passo foi realizado com bons resultados, gerando soluções SRRP para entradas de ETSPR com boa redução de altura, restando algumas otimizações a serem completadas.

Palavras Chave — roteamento de circuitos integrados, matriz de transistores, problema de roteamento de canal, QCL, ETSPR, SRRP, CAD.

I. INTRODUÇÃO

O modelo QCL (*Quickly Customized Logic*) proposto em [7] vem sendo utilizado com sucesso nos canais de roteamento das matrizes *gate-array* do projeto ÁGATA [1], devido às condições que ele provê para a aplicação do algoritmo LEA (*Left Edge Algorithm*) [5], que produz soluções ótimas em tempo linear. Em QCL há uma camada de metal inferior predefinida e uma camada de metal superior programável para interconexões [2]. O modelo de [7] consiste, assumindo uma posição de referência horizontal para o canal, de condutores verticais fixos localizados na camada inferior, chamados de *underpasses* (representados em cinza claro na Fig.1(a)). Sobre eles, trilhas horizontais de roteamento (em cinza escuro) são posicionadas aos pares, intercaladas com vias de acesso (preto).

Aos *underpasses* são assinaladas redes originárias de bandas de transistores localizadas acima e abaixo do canal. E geralmente, como o caso das matrizes do ÁGATA, um Roteamento em Rio é utilizado para este assinalamento. O modelo criado por [7] necessita de uma trilha de vias entre cada par de trilhas horizontais de roteamento, evitando restrições verticais na aplicação do algoritmo LEA (uma das condições necessárias para a aplicação deste). A redução para

apenas duas trilhas de vias localizadas no limite superior e no limite inferior do canal libera trilhas horizontais para roteamento e/ou diminui a altura do canal, como considerado anteriormente em [3]. Este modelo é conhecido em [6] como *Equipotential Two-Sided Planar Routing* (ETSPR), onde somente um subconjunto das redes é roteado sobre a banda de células, para a redução do congestionamento dos canais adjacentes à mesma, num modelo conhecido como *Over The Cell* (sobre a célula). No caso do ÁGATA, que se trata de QCL, o roteamento de canal não pode se dar, mesmo que em parte, sobre a banda de células, por se tratar de um MPGA (*Masked Programmable Gate Array*) onde as interconexões entre os *gates* da matriz (na banda de células) e as conexões entre os *underpasses* dos canais são realizadas ao mesmo tempo e na mesma camada de metal na hora da personalização do circuito.

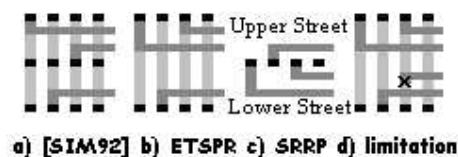


Fig. 1. Modelo [7] (a), ETSPR (b) e SRRP (c), e uma limitação em ETSPR(d).

ETSPR (Fig.1(b)) apresenta similaridade com *Single Row Routing Problem* (SRRP [5]). SRRP é o problema de roteamento de um conjunto de redes de dois ou mais terminais dispostos em uma linha de nodos chamada *single row* (Fig.1(c)). A área sobre a *single row* é chamada de **rua superior** e a área abaixo de **rua inferior**. Note como a rua superior e a rua inferior da representação do SRRP da Fig.1(c) correspondem respectivamente a área inferior e superior do ETSPR representado na Fig.1(b).

Além das similaridades, algumas diferenças devem ser consideradas. Alguns padrões de conexões possíveis em ETSPR não são possíveis em SRRP, como mostra a Fig.2(a). A rua superior e a inferior de um SRRP também não são, necessariamente, diretamente correspondentes às áreas inferiores e superiores de um ETSPR, podendo se intercalar

¹ Glauco Borges Valim dos Santos, Pontifícia Universidade Católica do Rio Grande do Sul – Campus Universitário II – BR 472 Km 7 CEP 97500-970 – Uruguaiana, RS – Brasil, Phone: +55(55)413-1515, Fax: +55(55)413-1280, glauco@puers.campus2.br.

² Cândice Corradi Kindel, Pontifícia Universidade Católica do Rio Grande do Sul – Campus Universitário II – BR 472 Km 7 CEP 97500-970 – Uruguaiana, RS – Brasil, Phone: +55(55)413-1515, Fax: +55(55)413-1280, candy@puers.campus2.br.

³ Marcelo de Oliveira Johann, Pontifícia Universidade Católica do Rio Grande do Sul – Campus Universitário II – BR 472 Km 7 CEP 97500-970 – Uruguaiana, RS – Brasil, Phone: +55(55)413-1515, Fax: +55(55)413-1280, johann@puers.campus2.br.

como mostra a Fig.2(b). De qualquer forma, é ainda possível mapear soluções de SRRP para ETSPR.

Além disso, conexões horizontais tornam *underpasses* mais facilmente inalcançáveis por outras conexões, uma vez que estas devem ser todas planares no modelo ETSPR, bastando duas para que isto aconteça (veja Fig.1(d)). Já no modelo original do ÁGATA elas não representam restrições verticais umas as outras, o que é assegurado pela disponibilidade de uma trilha dedicada a vias para cada trilha dedicada a conexões, seja acima ou abaixo desta.

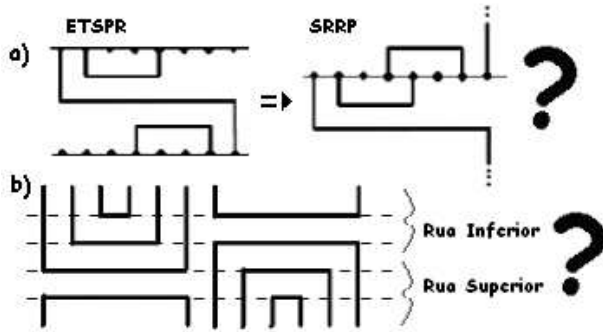


Fig. 2. Diferenças entre ETSPR e SRRP.

Uma abordagem baseada no trabalho de [4] foi escolhida entre três abordagens teóricas definidas para o problema, uma vez que foi assumida como a mais viável devido à existência de trabalhos anteriores baseados também em [4] com modelos semelhantes ao modelo aqui proposto ([8] e [2]).

II. A ABORDAGEM BASEADA EM [4]

Abordagens para ETSPR baseadas em [4] já foram apresentadas antes, como [8] e [2], mas uma vez que [8] não satisfaz os requerimentos para solução de problemas de relativa complexidade (não utiliza *doglegs* em suas conexões para procurar soluções de roteamentos, por exemplo) e [2] não descreve claramente o seu modelo, este trabalho segue o mesmo caminho, mas começando por [4] em vez dos dois trabalhos de ETSPR.

Primeiro, algumas limitações de [4] devem ser consideradas. Este não limita o Congestionamento Entre Nós (C_B - *Between Nodes Congestion*), ou seja, o número de cruzamentos de conexões entre ruas que podem ocorrer entre cada nó da *single row*, o que aumenta fortemente a sua chance de sucesso. Cada vez que um cruzamento é realizado em SRRP, ao ser mapeado para ETSPR resulta na necessidade de um *underpass* a mais no que é chamado de *dogleg*. Na Fig.13(b) e (c) temos a representação em SRRP e ETSPR, para cada uma, de duas situações em um determinado ponto de roteamento, observe que um cruzamento no SRRP resultará em um *underpass* no qual existe uma conexão entrando no limite inferior e uma saindo no limite inferior, ou vice versa, este é o caso em que temos um *dogleg*. *Doglegs* ajudam bastante na realização do roteamento planar e implicam na disponibilidade de *underpasses*. Porém, no mapeamento de *underpasses* assinalados a redes para nós de uma *single row*, alguns destes *underpasses* são adjacentes, não permitindo a

utilização de *underpasses* livres entre eles para *doglegs*. Tais casos limitam C_B através do canal durante o roteamento. Ainda assim pode ser possível reposicionar estes *underpasses* realizando um novo assinalamento de forma a liberar *underpasses* necessários para *doglegs*. Além disso, uma vez obtendo-se resultados satisfatórios na redução de altura do canal com a mudança de modelo, mais trilhas poderão ser usadas para o Roteamento em Rio, responsável por este assinalamento.

Em [4], todas as possíveis ordenações das redes que passam sobre cada ponto de conexão com a *single row* são consideradas. Fig.3(a) mostra as redes que cruzam cada ponto de conexão com a *single row* para um caso hipotético. Em cada nó, uma das redes se conecta a ele, se tratando de um ponto de conexão daquela rede. Cada ponto de conexão de uma rede é classificado como um *start point* (ponto inicial), caso seja o primeiro ponto de conexão desta, como um *finish point* (ponto final), caso seja o último ponto de conexão desta, e um *middle point* (ponto intermediário), no caso de não ser nenhum dos anteriores. As figuras Fig.3(b), Fig.3(c) e Fig.3(d) mostram o procedimento para os *start points*. Neste caso todas as ordenações possíveis são consideradas durante a inserção da rede em roteamento. Para o primeiro somente uma ordem é definida, como mostra a Fig.3(b). Após a primeira rede ser colocada em roteamento, cada nova inserção resultará em diferentes ordenações, como mostra a Fig.3(c) para a Fig.3(b), e a Fig.3(d) para somente uma das ordenações da Fig.3(c) (uma vez que o número de ordenações cresce cada vez que uma nova rede é inserida em roteamento). No caso de *finish points* e *middle points* só se verifica quais ordenações possuem a rede a ser conectada a *single row* adjacente a ela, eliminando as outras, o que resulta em cortes de alguns galhos da árvore de busca. Este exemplo foi mostrado para mostrar que o tamanho da árvore de busca de [4] é bastante grande, e tentar justificar a utilização de uma versão simplificada.

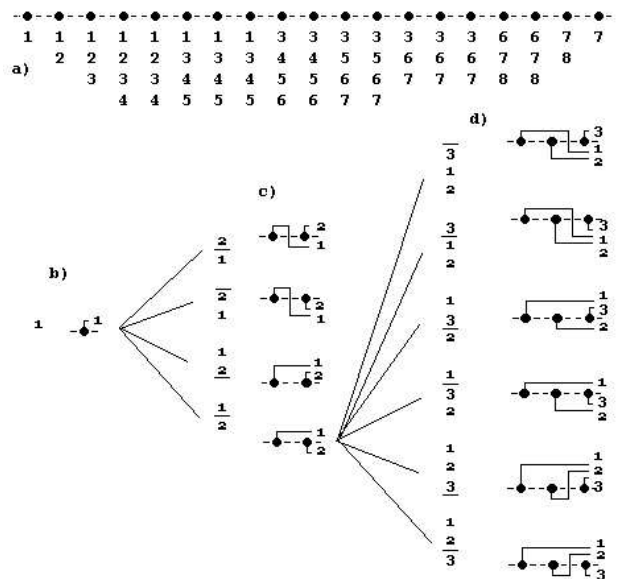


Fig. 3. Exemplo do procedimento de [4].

Uma versão simplificada do roteamento SRRP de [4] foi implementada. Esta versão não usa a árvore de busca de [4], mas uma simplificada, com o que se espera ganhar eficiência, apesar de ser menos completa do que a original. Com um novo conceito de armazenamento dos segmentos das conexões, aliado a utilização de suas listas encadeadas que permitem a criação de, no máximo, dois ramos na árvore de busca de cada nodo a ser processado, espera-se reduzir a necessidade de memória e processamento.

Seguindo [4], a classificação original dos pontos é mantida. Mas novas políticas devem ser utilizadas para os procedimentos em cada tipo de ponto de conexão encontrado durante o roteamento.

Uma iteração lida com cada um dos pontos, inserindo *start points* em roteamento, excluindo *finish points*, e fazendo ambas as coisas para *middle points*. Cada introdução insere a rede do *start point* em uma de duas listas encadeadas correspondentes às posições de redes na rua superior e inferior no nodo corrente da *single row*. Uma política definirá em qual lista encadeada a rede deve ser inserida primeiro, criando um braço na árvore de busca. Não havendo sucesso na continuação do roteamento a rede será inserida na outra rua, o que corresponderá ao segundo braço neste ponto da árvore de busca. A política escolhida até o momento é simples e implica em escolher a rua (ou a lista encadeada) que está menos congestionada, ou seja, que tem o menor número de redes em roteamento no momento.

Cada vez que se trata de um nodo correspondente a um *finish point*, a rede deve ser procurada nas listas encadeadas e, no caso de não ser a primeira rede da lista em que estiver, os nodos anteriores a ela devem trocar de rua (ou seja, de lista encadeada), já que ela deve se conectar a *single row*, o que implicará em cruzamentos correspondentes as redes que trocaram de rua (lista encadeada), se houverem. Os cruzamentos implicam no aumento de C_B entre o nodo corrente e o nodo anterior, como mostra a Fig.13(b) quando um nodo referente a um *finish point* da rede do segmento apontado por uma seta é encontrado. Esta é a diferença mais básica em relação ao roteamento original de [4]. Enquanto aquele gerava cruzamentos ao gerar todas as possibilidades de inserção de redes, a versão deste trabalho gera cruzamentos ao conectar *finish points* a *single row*.

Após todos os nodos serem cobertos pela iteração, o *layout* pode ser obtido através da informação guardada a cada posição no par de listas encadeadas correspondentes as ordenações das redes naquela posição. Faz-se necessário também que o desenho do *layout* leve em consideração a redução de dobras (*bends*) desnecessárias, como mostra a Fig.4. Na Fig.4 *middle points* são referidos como *stafin points*, forma como foram chamados na implementação.

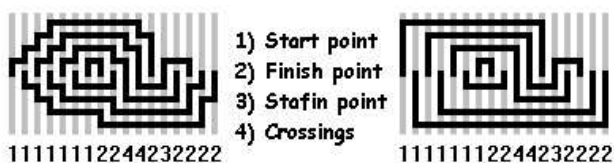


Fig. 4. Exemplo do histórico armazenado durante o roteamentos e o *layout* correspondente.

III. IMPLEMENTAÇÃO

O algoritmo foi completamente implementado em C em um IBM-PC rodando Linux. Foi testado com problemas de roteamento de canal de cinco circuitos de teste: copell, powpad11, powxor11, m8255 e timer14.

Para armazenar as posições das redes sobre cada nodo foi criada uma estrutura chamada **POAP** (*Positions Over A Point*). Esta estrutura contém duas listas encadeadas correspondentes as redes da rua superior e inferior, na devida ordem em relação a *single row*. Um vetor de nodos previamente classificados correspondente a *single row* é criado. A primeira POAP é criada para o primeiro nodo inserindo-o na lista correspondente a rua superior (por praxe, seguindo [4]). Em seguida, uma função recursiva é chamada para gerar as POAPs dos demais nodos. A Fig.5 mostra em pseudocódigo essa função, que recebe por parâmetro a POAP anterior, um vetor de nodos, o índice do próximo nodo, e o tamanho total do vetor de nodos. Enquanto o índice for menor que o total uma nova POAP é gerada como uma cópia da POAP anterior, mas sem os *finish points*, e com um ponteiro para a POAP anterior para que se possa percorrer o histórico do roteamento na hora da geração do *layout*. A POAP corrente é modificada de acordo com o tipo do nodo corrente. Diferentes procedimentos são realizados para cada tipo de nodo, como já descrito na seção anterior, e a função é chamada novamente com a nova POAP sendo enviada por parâmetro e o índice do vetor incrementado.

```
void gen_POAP(POAP lastpoap, NODE **nv, int next, int total) {
    if (next < total) {
        POAP currentpoap = eliminateFinishes(copy of lastpoap);
        NODE currentnode = nv[next];
        switch (currentnode->type) {
            case START: { startProcedure(); break; }
            case FINISH: { finishProcedure(); break; }
            case MIDDLE: { middleProcedure(); break; }
        }
    }
    else {
        /* founded !!! */ generateLayout();
    }
}
```

Fig. 5. Pseudocódigo da função recursiva responsável pelo roteamento.

Foi também implementada uma ferramenta de visualização em JavaTM, que imprime o resultado ao redor de uma linha horizontal centralizada representando a *single row*. Sob cada conexão com a *single row* e cruzamento, é desenhado em cinza claro uma representação de *underpass*, para verificar a necessidade de reposicionamento daqueles adjacentes ou próximos demais. Por simplificação, a ferramenta, por enquanto imprime o *layout* resultante sem dobra (*bend*) alguma, mas algumas devem ser consideradas para permitir o perfeito encaixe da rua superior com a inferior da *single row* em um mapeamento para ETSPR.

Por enquanto, falta a implementação do reposicionamento dos *underpasses* adjacentes ou muito próximos entre os quais haja uma necessidade de *underpasses* livres maior do que o número real de *underpasses* entre eles. Feito isso, o Roteamento em Rio deve ainda ser possível. Outra possibilidade é realizar o assinalamento das redes

provenientes das bandas de células aos *underpasses*, e conseqüentemente o Roteamento em Rio, em conjunto com o roteamento dos *underpasses*, de forma a melhorar as condições para a realização deste. Informações a respeito do número de *underpasses* livres antes e depois de cada nodo podem ser importantes, ou ainda a utilização de zonas que representem todas as possíveis posições de um *underpass* respeitando um Roteamento em Rio válido, como o que foi definido como *patterns* em [2].

IV. RESULTADOS

O algoritmo resolveu todos os problemas de roteamento com resultados satisfatórios para quatro dos cinco circuitos testados. A Fig.6 mostra um diagrama com o resultado dos roteamentos dos canais para cada um dos 5 circuitos. Nas três primeiras colunas estão os nomes dos circuitos, a seqüência de canais e o número total de canais a serem roteados. Nas duas colunas que seguem estão, respectivamente, os resultados de trilhas utilizadas pelo roteamento, o que é igual à altura em Problemas de Roteamento de Canal (CRP – *Channel Routing Problem*), e o número de trilhas adicionado de 2, incluindo as duas trilhas dedicadas à via acima e abaixo do canal, correspondentes aos limites inferiores e superior do canal, como é considerada a altura nos canais do ÁGATA, uma vez que são tratados os pares de trilhas dedicadas a conexão intercalados com as trilhas dedicadas a vias. Na coluna seguinte, chamada de t' , temos o número de trilhas de conexões utilizados no ÁGATA, o que não corresponde a altura e não deve ser considerado como parâmetro para comparação, uma vez que ainda não conta as vias necessárias para tal roteamento, mas sim considera-se as duas colunas que seguem como parâmetro para serem comparadas com as duas colunas de resultados do algoritmo baseado em [4], respectivamente, a primeira com a altura correspondente a CRPs e a segunda considerando os limites superior e inferior dos canais. As três últimas colunas são referentes ao roteamento do ÁGATA completo com os terminais originários das bandas de células e mais outros terminais vindos das laterais dos canais originários de entrada e saída que se localizam na direita e na esquerda do circuito. Estes últimos resultados não são considerados, pois ainda não foram consideradas as redes laterais no roteamento durante esta fase de avaliação da viabilidade do novo modelo. É importante lembrar também que a altura de canal mostrada para cada circuito, seja em um modelo ou outro, é a altura máxima observada nos canais de cada circuito.

A informação que podemos extrair da tabela é uma animadora redução de altura de canal, para quase todos os circuitos - 4 do total de 5. Para isso basta comparar a quarta e a sétima coluna, onde temos, respectivamente, as alturas de canal para os circuitos no novo modelo e no modelo do ÁGATA. Ou ainda, comparar a quinta coluna com a oitava, que são os resultados anteriormente citados acrescidos de 2 para representar a altura do ÁGATA. Podemos observar, por exemplo, que, para o circuito m8255, a altura original do ÁGATA foi reduzida de 16 (oitava coluna) para 12 (quinta coluna) no novo modelo. Ou então, considerando a altura comumente adotada em problemas de CRP, podemos observar uma redução de altura de 14 (sétima coluna) para 9

(quarta coluna), para o circuito powxor11. São apenas dois pontos de vista diferentes, mas o ganho é sempre o mesmo.

Circu it	Chan nels	Chan nels n°	[Sah84] SRR		ÁGATA					
			CRP $t = h$	ÁGA TA h	t'	CRP h'	ÁGA TA h'	t	CRP h	ÁGA TA h
copel 1	1-17	17	14	16	10	14	16	10	14	16
m825 5	5-15	11	10	12	9	14	16	9	14	16
powp ad11	4-15	12	8	10	8	12	14	10	14	16
powx or11	4-15	12	9	11	9	14	16	10	14	16
timer 14	6-13	8	9	11	8	12	14	8	11	13

Fig. 6. Diagrama comparativo de resultados de roteamento do algoritmo baseado em [4] e do roteamento do ÁGATA.

O único circuito com resultados não muito satisfatórios foi o circuito copel1, o mais complexo, com uma taxa de utilização da matriz maior, com a utilização de 17 canais de roteamento. Ainda assim, pode-se notar que, para este circuito, não foi necessária uma altura maior do que a altura dos canais originais.

Alguns resultados representados pela ferramenta de visualização são mostrados da Fig.7 a Fig.12. A ferramenta originalmente usa cores diferentes para a impressão das conexões de diferentes redes, mas para permitir a visualização deste trabalho em tons de cinza, caso assim seja impresso, elas foram mostradas aqui todas em preto.

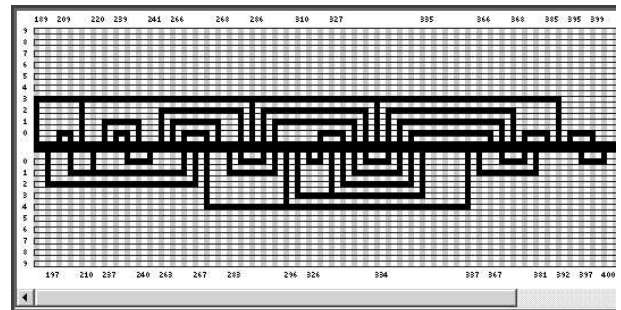


Fig. 7. SRRP para o canal 5 do circuito m8255.

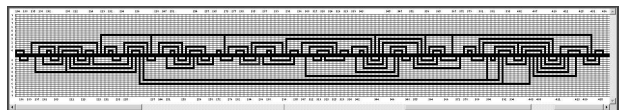


Fig. 8. SRRP para o canal 11 do circuito m8255.

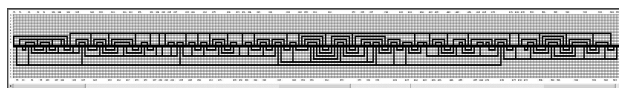


Fig. 9. SRRP para o canal 8 do circuito powpad11.

A Fig.7 e a Fig.11 talvez sejam mais representativas devido a menor complexidade de seus roteamentos que

envolviam menos redes do que os demais, permitindo uma melhor visualização nas figuras.



Fig. 10. SRRP para o canal 14 do circuito powxor11.

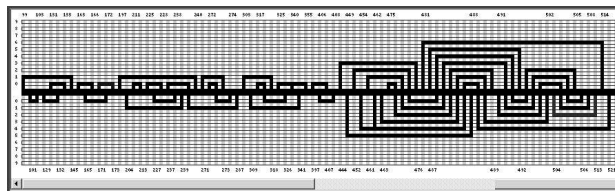


Fig. 11. SRRP para o canal 1 do circuito cope11.

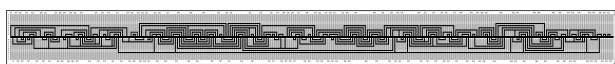


Fig. 12. SRRP para o canal 8 do circuito cope11.

V. CONCLUSÕES E TRABALHOS FUTUROS

O que foi implementado até agora apenas considera conexões possíveis em SRRP, ignorando outras formas de conexões possíveis em ETSPR que podem fortemente melhorar o roteamento planar.

A primeira nova forma de conexão a ser considerada é a mesma que é chamada de *top-bottom crossing* em [6] e *crank-shaped wire* em [2], que se trata de uma conexão que inicia no limite superior do canal e termina no limite inferior, ou vice versa, como a mostrada na Fig.2(a).

Isto significa que, quando um *finish point* é encontrado, como mostra o exemplo da Fig.13(a), durante o roteamento SRRP (à esquerda na figura), o roteamento não procederá de forma usual, reposicionando os segmentos abaixo deste na outra rua, como mostra a Fig.13(b). Neste caso será necessário avaliar se é melhor realizar o procedimento usual ou realizar a nova forma de conexão.

Devemos olhar então, para a imagem a direita da Fig.13(a) e verificar que, no momento em que o *finish point* é encontrado, o segmento da rede a ser finalizada (apontado pela seta) possui apenas dois segmentos acima dele e três abaixo. Então, se uma conexão como a mostrada na Fig.13(c) é feita, teremos menos cruzamentos, o que implica na necessidade de *underpasses* livres, além de uma menor altura, como mostra a imagem a direita da Fig.13(c) com a representação ETSPR desta alternativa.

Acrescentar esta nova possibilidade irá fazer com que aconteça, também no caso de *finish points*, a criação de dois ramos na árvore de busca, o que implicará em quatro ramos no caso de um *middle point*, no qual é realizado tanto o procedimento para *start points*, como para *finish points*. Mesmo assim a redução de cruzamentos e altura é animadora.

Além disso, algumas otimizações na política de inserção de redes em roteamento também remetem à possibilidade de melhoramento na decisão pela primeira rua a criar um ramo na árvore de busca. Esta trata de escolher em segundo lugar a rua em que se encontra a próxima rede a se conectar a *single*

row.

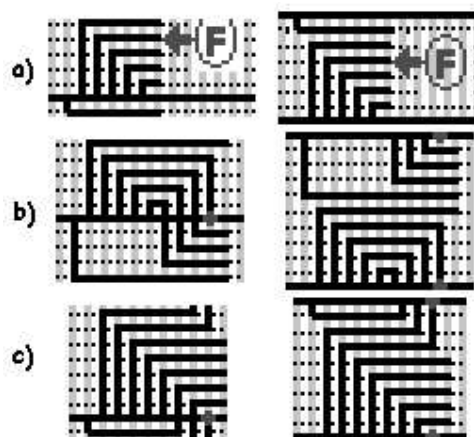


Fig. 13. Justificativa para utilização de novas conexões, para um determinado caso, (a) duas alternativas (b e c) .

Em vista dos resultados obtidos e das possíveis otimizações a serem realizadas torna-se encorajadora a continuação do trabalho. Tanto para que o que falta seja terminado, como a geração do *layout* final em ETSPR e o reposicionamento de terminais para a satisfação dos assinalamentos provenientes das bandas de células, quanto para implementar as otimizações recém descritas.

Abstract.— *This work investigates the reduction of the number of vias in the channel model of the gate arrays of AGATA project, using SRRP (Single Row Routing Problem) solutions. SRRP is well known as a printed board routing technique. Such reduction in the number of vias implies in a ETSPR (Equipotential Two-Sided Planar Routing) model, which accepts the translation of SRRP solutions. The customization of the gate arrays is done in a single layer, in an implementation called QCL (Quickly Customized Logic). The first step is done with good results in channel height reduction, translating SRRP solutions to ETSPR, missing some optimizations to be done.*

Keywords — IC routing, Gate Array, CRP, Quickly Customized Logic, Equipotential Two-Sided Planar Routing, Single Row Routing Problem, Computer Aided Design.

REFERÊNCIAS

- [1] L. Carro et al, "Ambiente ÁGATA de Projeto Versão Beta 2.0". Tercer Workshop Iberchip. Proceedings... México, 1997. p.495-503.
- [2] S-k. Dong, et al, "Two Channel Routing Algorithms for Quickly Customized Logic". EDAC. Proceedings... IEEE, Los Alamitos, 1993, p.122-126.
- [3] M.O. Johann, "Estruturas de Roteamento em Circuitos VLSI", Porto Alegre: PPGC/UFRGS, 1997.
- [4] S. Sahni, R. Raghavan, "The complexity of single row routing", IEEE Trans. on Circuits and Systems, v CAS-31, n 5, 1984.
- [5] N.A. Sherwani, Algorithms For VLSI Physical Design Automation, Kluwer, Massachusetts, 1993.
- [6] N. Sherwani, S. Bhingarde, A. Panyam, Routing in the Third Dimension: From VLSI Chips to MCMs, IEEE Press, Piscataway, 1995.
- [7] S.A. Simões et al, "Matriz gate array cmos avançada, configurável por um único nível de metal". Congresso da SBMICRO. Anais... SBMICRO/USP, São Paulo, 1992, p.281-291.
- [8] Y. Sun, "A Channel Router for Single Layer Customization Technology". ICCAD. Proceedings... 1991. p.436-439.