* RELATÓRIO TÉCNICO *

ARCO: UM SISTEMA DE GERAÇÃO DE LAYOUT COM ACELERADOR DE ROTEAMENTO

Júlio Salek Aude
Eudes Prado Lopes
Mário Ferreira Martins
Serafim Brandão Pinto

NCE 36/90

Universidade Federal do Rio de Janeiro
Núcleo de Computação Eletrônica
Caixa Postal 2324
20001 - Rio de Janeiro - RJ
BRASIL

* Este artigo foi publicado originalmente nos Anais do XXIII Congresso Nacional de Informática da SUCESU, Rio de Janeiro, em Agosto de 1990.

ARCO: Um Sistema de Geração de Layout com Acelerador de Roteamento

RESUMO

Este trabalho descreve o sistema ARCO de geração de layout de placas de circuito impresso. O sistema possui um acelerador em hardware para execução da etapa de roteamento, baseada no algoritmo de Lee. O software do sistema é composto dos seguintes módulos: interface com o programa de captura de esquemáticos do ORCAD, alocador de componentes, ordenador de sinais para roteamento, gerador de descrição gráfica da placa e sofisticada interface gráfica com o usuário. O sistema ARCO roda em microcomputadores nacionais compatíveis com IBM-PC/XT, AT ou 386

ARCO: A Layout Generation System Using a Hardware Router

ABSTRACT

This work describes ARCO, a PCB layout generation system. The ARCO system uses a hardware router for the execution of Lee's algorithm. The system consists of the following software modules: an interface to ORCAD schematic capture software, a placement tool, a software for automatic ordering of the netlist prior to routing, a software for the generation of a graphics description of the board and a sophisticated graphics user interface. The ARCO system runs on IBM-PC/XT, AT or 386 compatibles.

Introdução

Tipicamente as etapas que constituem o desenvolvimento de projetos eletrônicos, tanto em circuito impresso como em silício, são as seguintes: projeto do sistema, projeto detalhado dos blocos funcionais que compõem o sistema, simulação, alocação dos módulos utilizados e roteamento das conexões entre os módulos O roteamento das conexões e, em geral, uma das etapas que mais consome tempo de máquina no processo de automatização de projetos de circuitos eletrônicos. Mesmo assim, dada uma área fixa para o roteamento dos sinais, os algoritmos existentes nem sempre alcançam índices satisfatórios com relação ao número de conexões automaticamente completadas. Por isso, a intervenção do projetista se torna necessária para completar manualmente as conexões restantes. Este processo é normalmente demorado e complexo.

Em geral, para se obter um maior índice de conexões automaticamente completadas, é necessário exercitar-se os algoritmos de roteamento com diferentes alternativas de alocação dos módulos a serem interconectados e com diferentes ordenações da lista de conexões a serem feitas. Consequentemente, é de grande interesse prático a redução drástica do tempo de execução dos algoritmos de roteamento.

O sistema ARCO tem por objetivo realizar a geração do layout em circuito impresso de sistemas eletrônicos utilizando um acelerador para a execução da etapa de roteamento que implementa em hardware o algoritmo de roteamento proposto por Lee [LEE61]. Este algoritmo garante encontrar um caminho entre dois pontos se tal caminho existe e garante que o caminho encontrado é de tamanho mínimo. O algoritmo, entretanto, é extremamente lento. Considerando-se as qualidades do algoritmo e seu problema de velocidade, a definição

de arquiteturas para sua implementação em hardware torna-se atraente.

Muitas das arquiteturas já propostas para solução deste problema são sistemas com estrutura matricial do tipo SIMD ("Single Instruction Stream - Multiple Data Stream") onde, em princípio, cada elemento de processamento é associado a uma célula da matriz que representa a superfície a ser roteada [BREUE81,IOSUP86,BLANK81]. A vantagem deste tipo de arquitetura é que o potencial paralelismo do algoritmo de Lee é explorado ao máximo. No entanto, o número de elementos de processamento é exageradamente grande e o grau de utilização destes elementos é, muitas vezes, baixo. Além disso, à implementação de arquiteturas deste tipo só é viável com o uso de integração em muito larga escala.

No sistema ARCO a arquitetura do acelerador foi implementada com circuitos integrados SSI, MSI e LSI disponíveis no mercado e funciona como um dispositivo conectável aos microcomputadores nacionais. Em conseqüência, o sistema ARCO viabiliza a construção de estações de trabalho para roteamento poderosas e de baixo custo.

A arquitetura do acelerador é uma extensão da máquina proposta por Won, Sahni e El-Ziq [WON87]. O acelerador tem capacidade para operar com áreas de roteamento de até duas camadas e dobra o paralelismo explorado pela máquina de Won sem subutilizar o hardware adicional necessário. Esta melhoria de desempenho é alcançada com a incorporação na arquitetura da estrutura em "pipeline" utilizada no processador RPS [RUTEN84].

Além do acelerador, o sistema ARCO engloba ainda um conjunto de módulos de software. Tais módulos permitem ao usuário: interfacear com o sistema ORCAD de captura de esquemático, definir

características físicas da área a ser roteada e gerar uma representação gráfica para esta definição, alocar automaticamente os módulos utilizados na superfície a ser roteada, ordenar a lista de conexões a serem roteadas segundo diferentes critérios, rotear as ligações com o uso do acelerador ou com o uso de um roteador implementado em software, obter saída gráfica para visualização e alteração da solução gerada pelo roteador e definir ou alterar bibliotecas descritivas das características físicas dos módulos e dos conectores conhecidos pelo sistema.

Na Seção 2 deste artigo, os principais aspectos do algoritmo básico de Lee são brevemente revisados. As alterações introduzidas neste algoritmo básico são também discutidas nesta seção. A arquitetura do acelerador é descrita na Seção 3. A Seção 4 descreve as características mais importantes dos módulos de software que compõem o sistema ARCO. Finalmente, na Seção 5 o estado atual do projeto é descrito e suas perspectivas futuras são discutidas.

O Algoritmo de Lee

O algoritmo de Lee sempre consegue achar um caminho ligando dois pontos, se este caminho existe, e tal caminho é sempre o menor possível. A implementação do algoritmo, no entanto, requer muita memória para armazenar a descrição da superfície a ser roteada como uma matriz de células quadradas e a execução do algoritmo pode ser bastante demorada quando a área a ser roteada é grande. Estes problemas reduzem consideravelmente as vantagens do algoritmo em implementações em software. Em implementações em hardware, o mesmo não ocorre já que o tempo de execução do algoritmo é drásticamente reduzido e o uso de muita memória não é

tão relevante.

Para cada conexão a ser feita, a execução do algoritmo de Lee passa por três etapas. A primeira delas, a "fase de expansão", é responsável por descobrir se há um caminho interligando os dois pontos. O procedimento utilizado nesta fase consiste na definição, a cada iteração, de uma nova fronteira de células a partir da expansão das células da fronteira atual nas direções norte, sul, leste e oeste. Inicialmente, a fronteira de células é composta de uma única célula, a célula que corresponde ao ponto de origem do caminho a ser traçado que é marcada com o valor 1 . Na próxima iteração cada uma das quatro células vizinhas da célula de início é marcada com o valor 2 se não estiver bloqueada por exemplo por um caminho anteriormente traçado. Todas as células marcadas definem a nova fronteira que é expandida na próxima iteração. A fase de expansão termina quando a célula que representa o ponto destino é atingida e, portanto, um caminho foi encontrado, ou quando não há mais células na fronteira para serem expandidas, significando que não existe caminho entre os dois pontos. Para um caminho de comprimento "l", expresso em número de células atravessadas, o tempo gasto na fase de expansão é proporcional a l**2.

A segunda fase do algoritmo de Lee é a "fase de retraço". Seu objetivo é traçar o caminho entre os dois pontos encontrado na fase de expansão. O procedimento adotado nesta fase começa na célula destino. A cada passo, a próxima célula a ser usada no caminho é escolhida dentre as vizinhas da última célula utilizada. Uma célula só pode ser escolhida se o valor atribuído a ela na fase de expansão é inferior em apenas um ao valor atribuído à última célula usada. O procedimento se encerra quando a célula de início é alcançada. Vários caminhos de comprimento mínimo podem

ser encontrados na fase de retraço. Em geral, o caminho escolhido é aquele que minimiza o número de quebras. O tempo gasto na fase de retraço é proporcional ao comprimento "l" do caminho.

A terceira fase do algoritmo é a "fase de reinicialização". Seu objetivo é preparar a matriz de células que representa a superfície de roteamento para a execução da fase de expansão da próxima conexão. Basicamente, nesta fase, as células utilizadas no caminho definido pelo retraço são transformadas em células de bloqueio para as próximas conexões e os valores marcados nas células não utilizadas são reinicializados. O tempo gasto nesta fase é proporcional a p*q, onde "p" e "q" são as dimensões da matriz de células que representa a superfície a ser roteada.

Três alterações do algoritmo básico de Lee foram implementadas no sistema ARCO. A primeira delas diz respeito a conexão de sinais com mais de dois pontos. Para implementação desta facilidade, a fase de reinicialização deve marcar como células destino, ao invés de células de bloqueio, as células utilizadas nas conexões pertencentes a um mesmo sinal. Desta forma, para cada conexão adicional de um mesmo sinal, uma nova célula origem é definida e todas as células já utilizadas nas conexões anteriores deste mesmo sinal são consideradas células destino. Com este mecanismo, ligações em "T" podem ser produzidas pelo roteador. Para algumas tecnologias muito rápidas isto pode ser indesejável. Para solucionar este problema, a fase de reinicialização adota um procedimento diferente em relação a sinais que devem ser tratados como linhas de transmissão. O roteamento destes sinais é feito em "cadeia" e para tal a fase de reinicialização define que todas as células, com exceção da célula de origem, pertencentes a conexão anterior de um sinal são células de bloqueio para as próximas conexões. A célula de origem é definida como célula destino para a

próxima conexão do sinal.

A segunda alteração implementada visa habilitar o algoritmo a trabalhar com mais de uma camada para roteamento. Esta situação é na verdade a mais comum de se encontrar na prática. Placas de circuito impresso possuem, em geral, pelo menos duas camadas em metal para roteamento. Basicamente, a alteração introduzida no algoritmo consiste em realizar a fase de expansão considerando que a matriz de células é uma matriz tridimensional. Ao se realizar a expansão para uma célula em um dado plano, as células vizinhas situadas em planos inferiores ou superiores são também marcadas. Comumente, em problemas de roteamento em múltiplas camadas, a cada camada é associada uma direção preferencial de roteamento, horizontal ou vertical. No algoritmo implementado no sistema ARCO, a direção preferencial de roteamento determina a ordem de precedência na escolha das células a serem usadas em cada passo do procedimento adotado na fase de retraço. Nesta fase, só se favorece uma mudanca de camada quando não é possível seguir na direção preferencial da camada corrente e existe possibilidade de expansão na direção preferencial da camada para onde se está passando.

A terceira e última alteração introduzida no algoritmo básico de Lee visa melhorar o desempenho do algoritmo, evitando que cada conexão roteada crie bloqueios que certamente dificultarão o roteamento de conexões subsequentes. O mecanismo utilizado para isso foi a introdução de custo diferenciado para as células que compõem a matriz que representa a superfície a ser roteada. Basicamente, a idéia consiste em atribuir custo mais elevado para células vizinhas dos pinos a serem interconectados, custo médio para células sem nenhuma célula vizinha ocupada e custo mais baixo para células vizinhas de células já utilizadas por conexões

anteriores. Com este esquema, busca-se evitar o bloqueio em posições adjacentes aos pinos dos componentes e dos conectores e busca-se, sempre que possivel, realizar-se um "empilhamento" de conexões traçadas próximas umas das outras. Na fase de expansão, células na fronteira a ser expandida que possuam custo diferente de 1 não sofrem expansão. O custo delas é subtraido de 1 e, com este novo custo, elas passam a fazer parte da fronteira a ser expandida na próxima iteração do algoritmo. Com isso, o algoritmo de Lee deixa de gerar o caminho de comprimento mínimo e passa a gerar o caminho de custo mínimo. Na fase de retraço é necessário fazer-se alterações porque o esquema de custo proposto tem um caráter dinâmico: o custo das células é redefinido a medida que o roteamento prossegue. Portanto, a fase de retraço passa a ter também o papel de calcular o custo de cada célula em função dos critérios acima estabelecidos. No sistema ARCO apenas três valores de custo estão sendo utilizados: 1, 2 e 3 e a utilização ou não de custo diferenciado para as células é uma opção do usuário do sistema.

Arquitetura do Acelerador

A arquitetura do acelerador de roteamento do sistema ARCO implementa em hardware as fases de expansão e reinicialização do algoritmo de Lee, que são as fases que mais demandam tempo de processamento em implementações em software. Na arquitetura, a fase de retraço, que é essencialmente um procedimento seqüêncial, é executada em um microprocessador. Com este esquema, o desempenho do roteador não é prejudicado consideravelmente, o hardware se torna mais simples é a implementação da fase de retraço fica mais fléxivel possibilitando que políticas diferentes de custo e de

escolha de caminhos minimos possam ser testadas.

O aspecto fundamental da arquitetura do acelerador de roteamento do sistema ARCO e o uso de um esquema especial de banqueamento para a organização da "memoria de placa", a memoria que armazena as células que compõem a matriz de representação da superficie a ser roteada. A estrutura de banqueamento utilizada é ilustrada na Figura 1, considerando-se apenas uma camada para roteamento. Nesta figura, cada célula está assinalada com o número do banco a que pertence. Como pode ser visto, há 8 bancos distintos. Dois grupos de bancos podem ser definidos: o grupo I, contendo os bancos 0, 1, 2 e 3 e o grupo II, contendo os bancos 4, 5, 6 e 7. As células pertencentes aos bancos do grupo I têm todas as suas células vizinhas em bancos do grupo II e vice-versa. Duas vantagens importantes resultam desta estrutura de banqueamento. Em primeiro lugar, a expansão de cada célula pode ser feita em paralelo, já que as células vizinhas a serem marcadas estão em bancos diferentes que podem ser acessados concorrentemente. Em segundo lugar, durante a fase de expansão do algoritmo, pode-se sempre expandir em paralelo células pertencentes á duas fronteiras subsequentes, já que uma delas causará expansão para células do grupo I e a outra fará expansão para células do grupo II.

Visando explorar as vantagens apontadas acima, duas seqüências de processadores em "pipeline" são usadas na implementação do acelerador para a fase de expansão do algoritmo. Cada seqüência processa a expansão das células pertencentes a uma de duas fronteiras subsequentes. Estas seqüências de processadores em "pipeline" são compostas de 3 estágios. O primeiro estágio lê a próxima célula a ser expandida da memória que armazena a fronteira em expansão. O segundo estágio realiza a expansão propriamente dita, acessando em paralelo até 4 bancos da memória de placa por

camada. Finalmente, o terceiro estágio acrescenta a memoria que armazena a fronteira a ser expandida pela outra sequência de processadores as células marcadas pelo segundo estágio. A Figura 2 mostra a arquitetura básica do acelerador. O protótipo inicial do acelerador foi implementado para trabalhar com placas representadas por matrizes 512x512 em duas camadas. Portanto, como a memória de placa possui 8 bancos por camada, há um total de 16 bancos de 32 kbytes. Cada célula é descrita por um byte. Três bits definem o status da célula: se a célula está bloqueada, se a célula representa um ponto de início, se a célula representa um ponto de destino, se a célula admite furo através dela ou se a célula é parte de um barramento de alimentação. Outros dois bits definem o custo associado àquela célula. Os três bits restantes são usados para definir o valor atribuído à célula durante a fase de expansão. É suficiente utilizar-se apenas três bits para isso porque estes valores são atribuídos ciclicamente de 1 a 7 às células em expansões subsequentes. É necessário se ter 7 valores diferentes para se garantir que, na expansão de uma célula "A", o valor a ser atribuído às suas vizinhas seja necessáriamente diferente do valor atribuído à célula cuja expansão marcou a célula "A". Durante a fase de retraço estes três bits juntamente com os dois bits de custo são utilizados para codificar o tipo de desenho que deve ser feito para descrever o caminho que atravessa a célula. Estes desenhos podem ser, por exemplo: um joelho, um "T", uma cruz, uma linha reta horizontal, etc.

A escrita de células na memória de fronteira pelo terceiro estágio do pipeline é feita em dois passos, um para cada camada. Isto é possível porque a informação a ser armazenada é codificada. A codificação indica as coordenadas da célula que causou a expansão, os valores de custo das células marcadas pela expansão (o valor 0

é gerado se a célula não foi marcada) e, finalmente o valor atribuído a estas células incrementado de 1, que representa o valor a ser usado na expansão destas células. Portanto, cada palavra da memória de fronteira possui 30 bits: 18 bits que definem as coordenada "x" e "y" da célula que causou a expansão, 1 bit que define a camada das células vizinhas, 8 bits que definem os custos das quatro células vizinhas nesta camada, e 3 bits que definem o valor a ser atribuído durante a expansão destas células. A memória de fronteira está organizada em dois conjuntos. Cada conjunto é composto de três bancos. A cada instante de tempo, um destes bancos é selecionado como o banco de leitura para o primeiro estágio de um dos pipelines. O segundo banco é selecionado como o banco de escrita do terceiro estágio do outro pipeline. O terceiro banco trabalha como um buffer que armazena as células que não puderam ser expandidas porque possuiam custo maior do que 1. Com esta organização, conflitos são evitados nos acessos feitos à memória de fronteira pelos dois pipelines, já que as operações de escrita ou leitura nunca se dão em um mesmo banco. As memórias de fronteira foram implementadas como 2 conjuntos de 3 bancos de 8kx30.

A fase de reinicialização do algoritmo também se beneficia da organização da memória de placa. Os dezesseis bancos são acessados em paralelo na operação de atualização dos valores atribuídos às células durante a fase de expansão.

Como já foi dito, a arquitetura do acelerador possui um microprocessador que executa a fase de retraço. Este microprocessador é ainda responsável por executar os passos necessários à inicialização do acelerador, iniciar o roteamento de cada ligação fornecendo as coordenadas dos pontos a serem ligados, e prover facilidades para monitoração e controle da operação do

acelerador para fins de teste.
Uma descrição detalhada dos procedimentos executados em cada fase do algoritmo de Lee pelo acelerador do sistema ARCO foi apresentada por Aude et al. [AUDE88].

O Software do Sistema de Roteamento

O sistema ARCO consta de um acelerador em hardware para execução do algoritmo de Lee e de um conjunto de programas que realizam a interface do sistema com o usuário e complementam a tarefa de geração do layout. Os seguintes programas compõem o sistema ARCO, conforme mostrado no diagrama da Figura 3:

a) Interface com ORCAD:
Esta interface transforma a informação gerada pelo programa de captura de esquemático do sistema ORCAD em informações no formato adequado para uso no restante do sistema ARCO. Tais informações são a lista de sinais e a lista de componentes utilizados no projeto. Na primeira lista, os sinais recebem qualificadores, extraídos dos nomes dos sinais, que caracterizam o sinal como sendo clock, linha de transmissão, alimentação, etc. Na segunda lista, o nome externo do componente é associado com o seu nome interno na biblioteca de componentes ou módulos

b) Gerador de Placa
A função deste programa é transformar uma descrição da superfície a ser roteada sob a forma de texto em um arquivo contendo a matriz de células a ser usada pelo roteador. A descrição textual da placa contém informações tais como: dimensões da placa, número, tipo e localização dos conectores, número de camadas de roteamento e

direção preferencial de roteamento em cada camada.

Para sua operação, o Gerador de Placa utiliza informações armazenadas em três arquivos: o arquivo de saída do programa Alocador, a biblioteca de características físicas dos componentes e conectores conhecidos pelo sistema e o arquivo que contém a descrição textual da placa. Dois arquivos são produzidos como saída pelo Gerador de Placa. O primeiro deles contém as coordenadas na matriz de células que representa a placa de todos os pinos dos componentes e conectores usados no projeto. O segundo arquivo contém a descrição da matriz de células a ser utilizada pelo roteador, o "mapa da placa".

Ao completar a criação do mapa da placa, o programa Gerador de Placa oferece ao usuário a possibilidade de utilizar um editor gráfico para criar barramentos de alimentação na superfície a ser roteada, editar bloqueios, inserir ligações manuais, remover componentes, etc. Todas estas informações editadas manualmente são incorporadas automaticamente ao mapa da placa.

c) Ordenador

Uma das funções deste programa é ordenar os pinos a serem interconectados em um sinal, utilizando critérios que visem a redução do comprimento do caminho final a ser traçado. A outra função deste programa é ordenar os sinais a serem roteados. Esta ordenação é feita segundo um critério escolhido pelo usuário dentre um menu de opções.

Para sua operação, o ordenador lê dois arquivos de entrada. O primeiro deles, gerado pelo Alocador, contém as coordenadas no mapa da placa de todos os pinos de conectores e componentes usados no circuito. O segundo arquivo contém, para cada sinal, uma lista de pinos a serem interconectados. Este arquivo é gerado pela interface com o ORCAD. O ordenador gera como saída um arquivo

contendo a lista de sinais ordenada e, para cada sinal, informação referente ao seu qualificador e às coordenadas no mapa da placa de todos os pinos que compõe o sinal.
O ordenador opera em três etapas. Na primeira delas, os sinais são opcionalmente agrupados conforme suas características elétricas a prioridade entre os grupos é estabelecida pelo usuário. A etapa seguinte consiste em ordenar os pinos em cada sinal. Para sinais que devam ser tratados como linhas de transmissão, o primeiro e último pinos permanecem nesta ordem já que eles representam normalmente os pinos do alimentador e da terminação da linha. Para os demais sinais, os pinos são ordenados de forma que a interconexão deles produza uma árvore mínima. A última etapa consiste em ordenar os sinais dentro de cada grupo segundo uma série de possíveis critérios que incluem a extensão das ligações e o número de pinos.

d) Alocador

Este programa produz automaticamente uma alocação dos módulos a serem interconectados que otimiza a distribuição de sinais sobre a superfície da placa. O algoritmo empregado considera a existência de componentes pré-posicionados e se baseia na técnica de "min-cut" [BREUE77 . O programa Alocador opera usando como dados de entrada informações contidas em quatro arquivos: o arquivo que descreve a placa, o arquivo que contém a lista de sinais, o arquivo que contém a lista de componentes e a biblioteca que armazena informações sobre as características físicas dos módulos e conectores conhecidos do sistema. Como saída, o Alocador produz um arquivo contendo as coordenadas da posição de cada componente e sua orientação. A interface gráfica do Alocador permite que o

usuario modifique o resultado produzido pelo Alocador, movendo os componentes ou posicionando os componentes em outra orientação.

e) Roteador

Fornece a alternativa ao usuário de não usar o acelerador para a execução da etapa de roteamento. O roteador em software funciona de forma idêntica ao roteador em hardware. A interface gráfica do Roteador permite que o usuário trabalhe sobre o resultado final do layout com uma série de facilidades: 4 escalas de zoom que permitem a visualização em detalhe de pequenas regiões do layout; Vôo que permite a observação na tela de placas muito grandes; Visualização em destaque de todas as ligações que compõem um determinado sinal; Apagamento de ligações individuais, de um sinal completo ou de toda uma região do layout; Traçado manual de ligações; Visualização seletiva das camadas que compõem o layout;
O software do sistema ARCO foi implementado em Pascal e roda em microcomputadores nacionais da linha PC-XT/AT e PC-386. O sistema funciona para configurações de máquina com placas CGA, EGA e VGA.

Estágio Atual e Perspectivas Futuras

O hardware do acelerador de roteamento se encontra, no momento, em fase final de teste. Sua implementação se deu basicamente com o uso de memórias MOS estáticas, PALs, microprocessador de 16 bits e alguns circuitos TTL SSI e MSI. Cerca de 600 circuitos integrados compõem o hardware do acelerador. Após a depuração completa deste primeiro protótipo, pretende-se substituir parte dos componentes utilizados por circuitos integrados dedicados do tipo standard-cell ou gate-array.
Experiências de simulação realizadas considerando-se um circuito com 1000 ligações de comprimento médio igual a 200 numa placa de

duas camadas mostram que o acelerador sera capaz de completar este roteamento em cerca de 4 minutos. De acordo com Blank [BLAN81], um processador convencional de médio porte leva cerca de 5 horas para resolver o mesmo problema. Portanto, o acelerador em hardware deve produzir um ganho de velocidade de cerca de 75 em relação a este computador convencional.

A versão 1 do software do sistema ARCO já foi concluida, apresentando todas as facilidades citadas na Seção 4, com exceção da interface gráfica para manipulação dos resultados da alocação. Estas facilidades bem como um editor gráfico para descrição da placa a ser usada farão parte da versão 2 que deverá estar concluída em julho próximo. A versão 1 do sistema foi utilizada no roteamento da placa de interface entre o acelerador e o microcomputador e em algumas experiências de roteamento de canais em layout de circuitos integrados projetados no estilo standard-cell. Resultados de alguns destes roteamentos são mostrados nas Figuras 4 e 5.

Agradecimentos

Os autores agradecem ao CNPq pelo apoio dado na fase inicial do desenvolvimento deste projeto.

Referencias

[AUDE 88] "Acelerador Modular de Roteamento", Aude, J. S., Paiva, E.B., Aude, E.P.L., Lopes, E.P., Martins, M.F., Pinto, S.B., Anais do II Simposio Brasileiro de Arquitetura de Computadores - Processamento Paralelo, Aguas de Lindoia, SP, Outubro de 1988

[BLANK81] "A Parallel Bit Map Processor Architecture for DA Algorithms", Blank,T., Stefik, M. vanCleemput,W., Proceedings of the 18th Design Automation Conference, Junho, 1981, pp. 837-845

[BREUE77] "Min-Cut PLacement", Breuer,M.A., Design Automation and Fault-Tolerant Computing, Outubro, 1977

[BREUE81] "A Hardware Router", Breuer,M.A., Shamsa,K., Journal of Digital Systems, Vol.4, No.4, 1981, pp. 393-408

[IOSUP86] "A Class of Array Architectures for Hardware Grid Routers", Iosupovici,A., IEEE Transactions on Computer-Aided Design, Vol.CAD-5, No.2, Abril, 1986, pp. 245-255

[LEE 61] "An Algorithm for Path Connections and its Applications", Lee,C.Y., IRE Transactions on Electronic Computers, Vol.EC-10, Setembro, 1961, pp. 346-365

[RUTEN84] "A Class of Cellular Architectures to Support Physical Design Automation", Rutenbar,R.A., Mudge,T.N., Atkins,D.E., IEEE Transactions on Computer-Aided Design, Vol.CAD-3, No.4, Outubro, 1984, pp. 264-278

[WON 87] "A Hardware Accelerator for Maze Routing", Won,Y., Sahni,S., El-Ziq,Y., Proceedings of the 24th Design Automation Conference, Junho, 1987, pp. 800-806

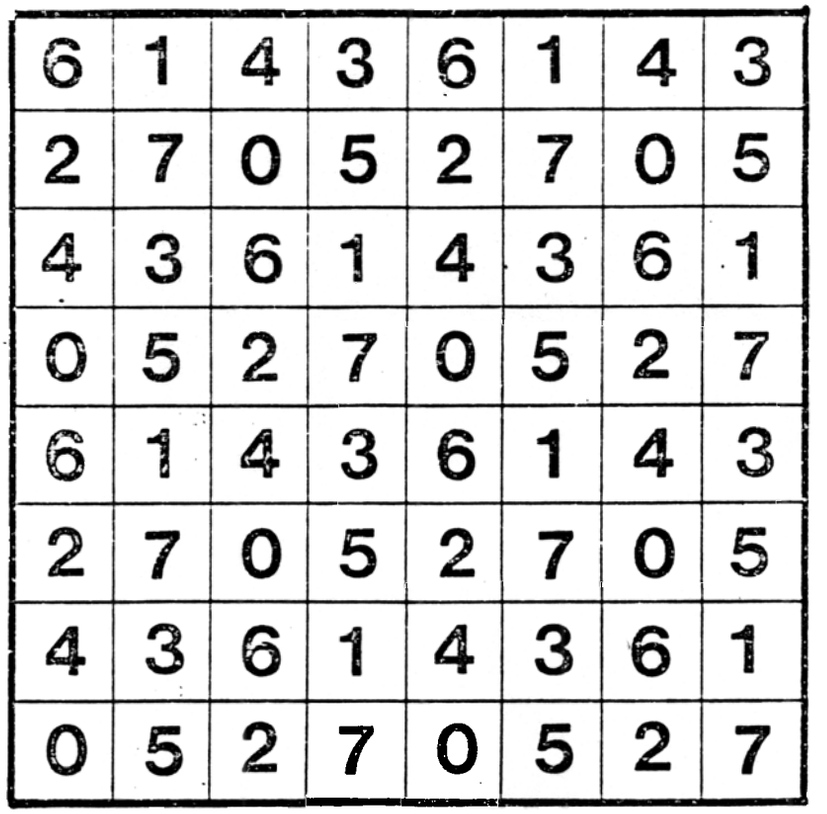

FIG.1 - ESQUEMA DE BANQUEAMENTO DA MEMÓRIA DE PLACA

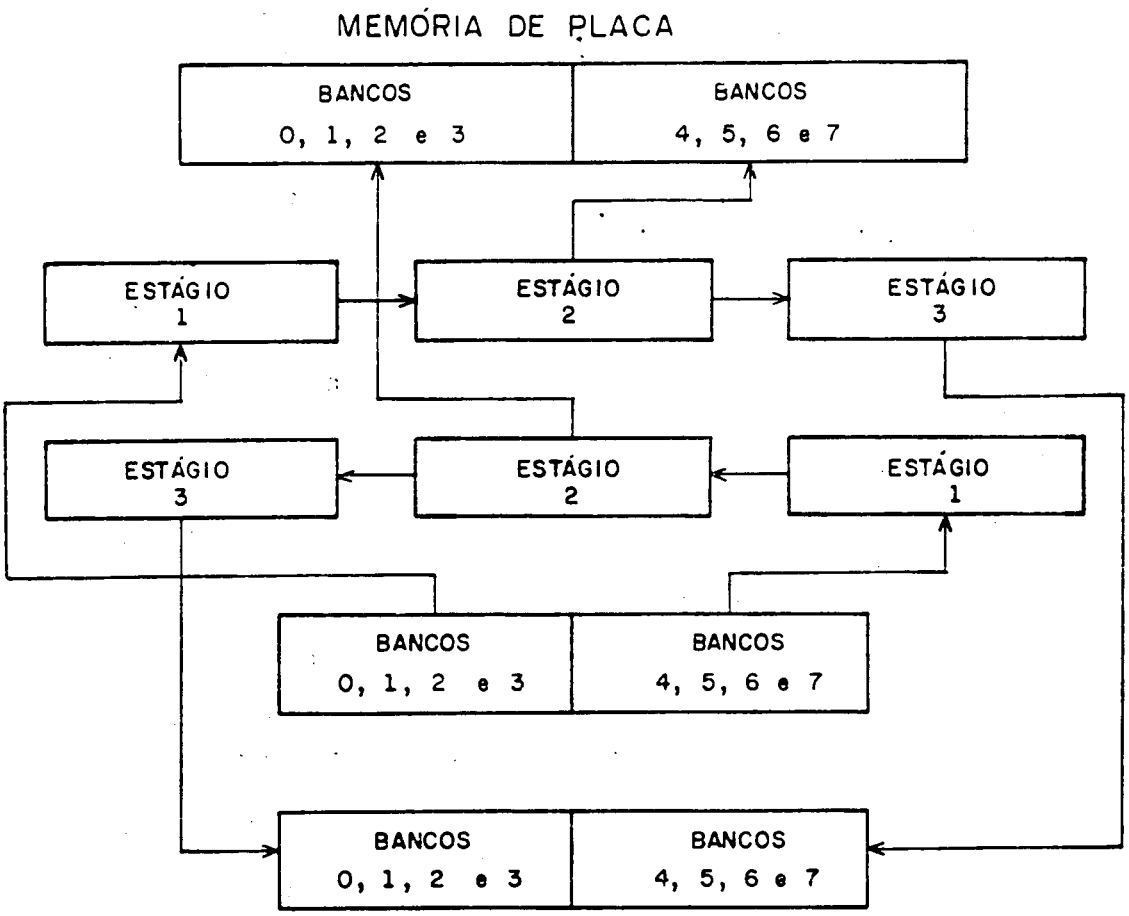

FIG. 2 - MÓDULO BÁSICO DO ACELERADOR DE ROTEAMENTO

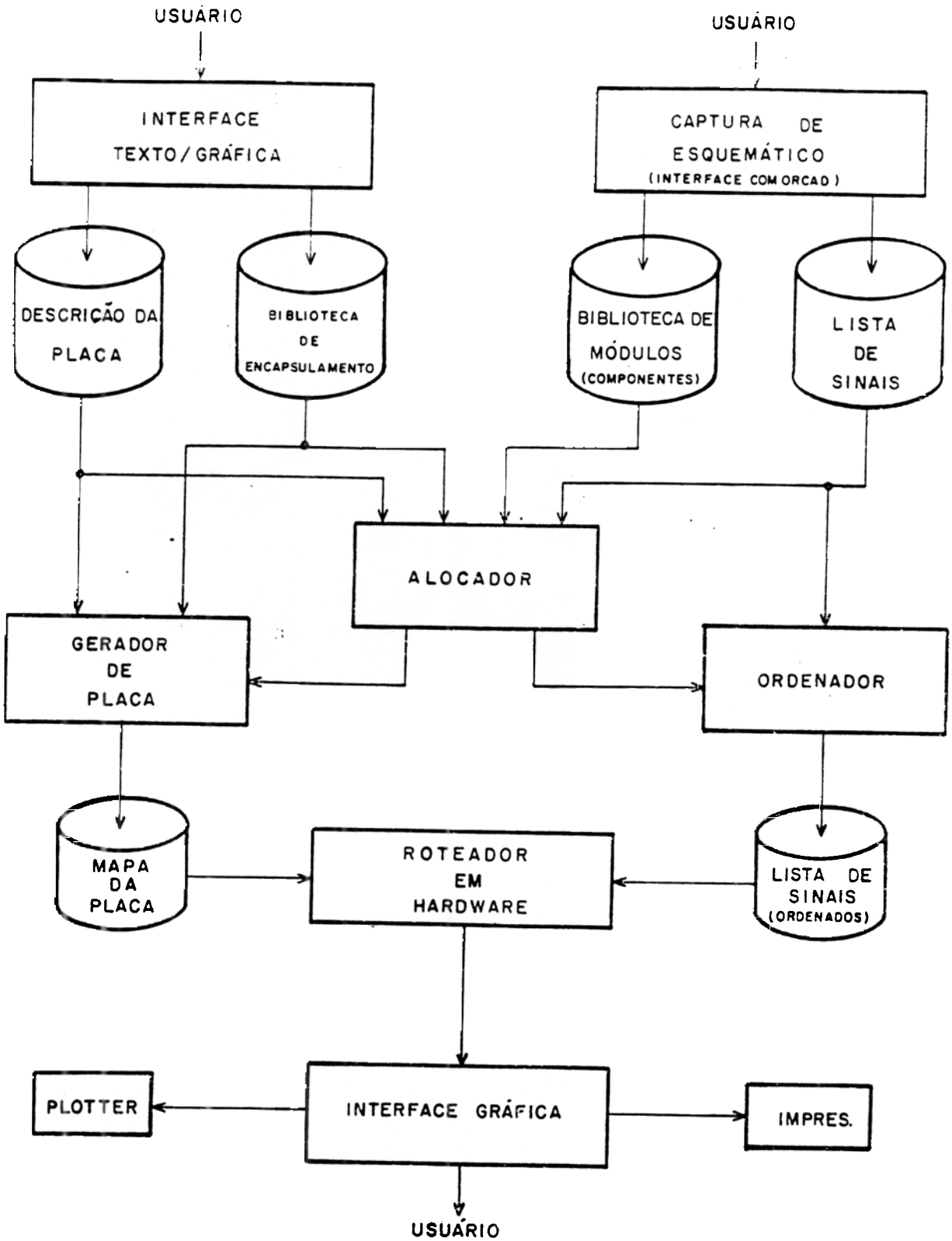

FIG. 3 - SOFTWARE DO SISTEMA

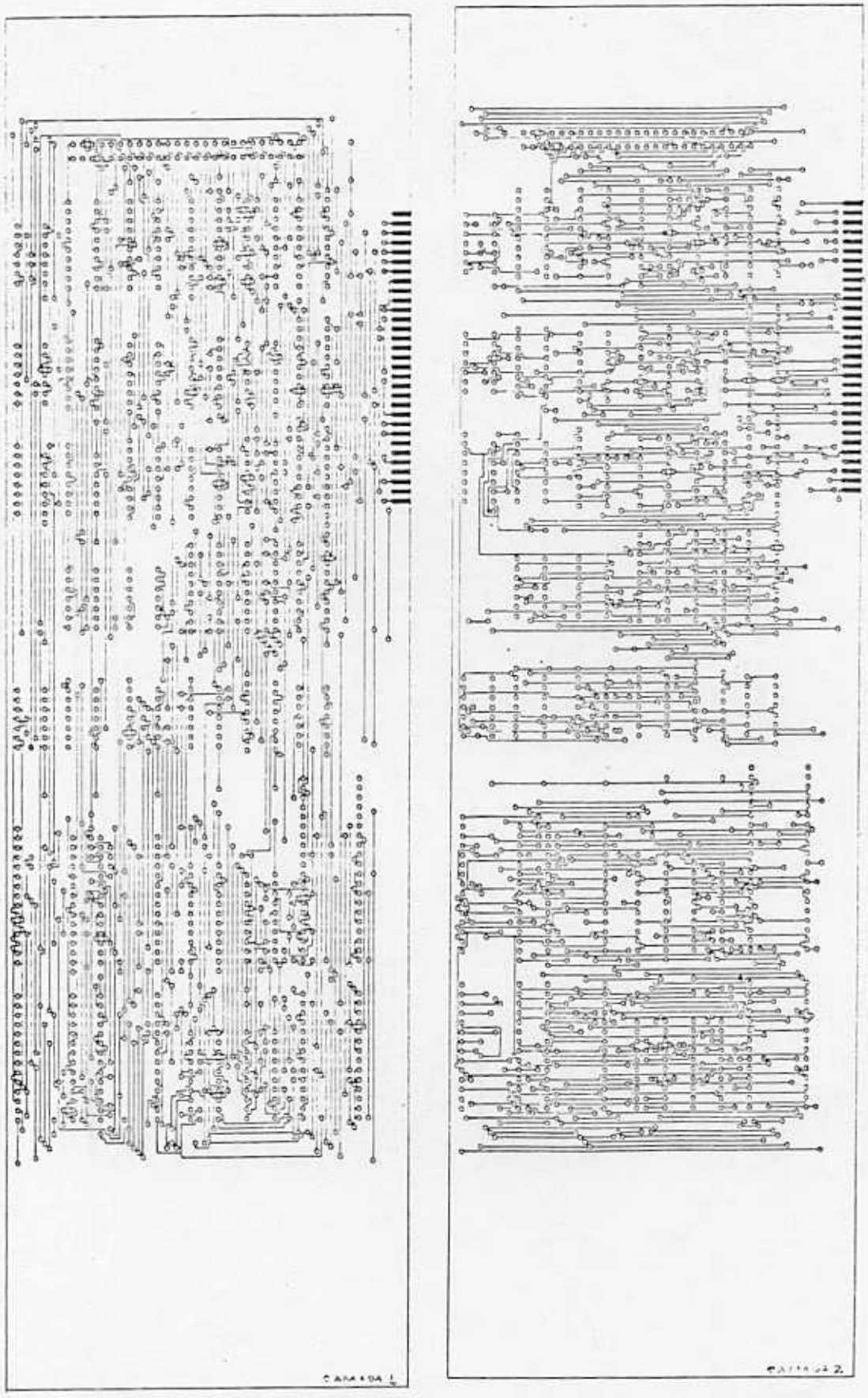

FIGURA 4: ROTEAMENTO DA PLACA DE INTERFACE (SISTEMA ARCO)

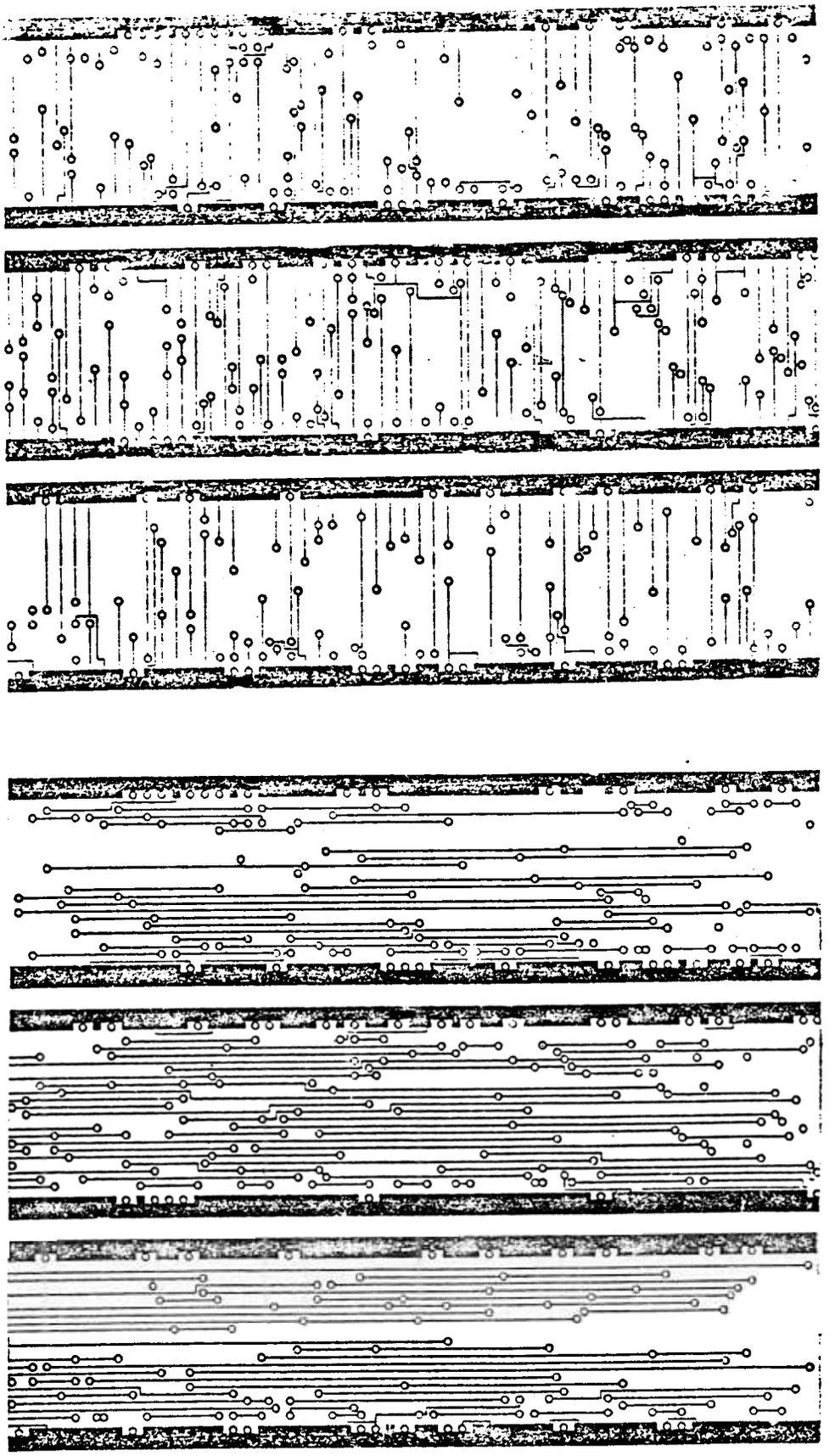

FIGURA 5: ROTEAMENTO DO CANAL "EXEMPLO DIFICIL" DE DEUTZ.