

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA DE CAMPINAS
DEPARTAMENTO DE AUTOMATIZAÇÃO

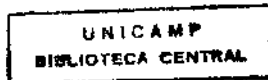
IMPLEMENTAÇÃO E TESTE DE
UM SISTEMA PARA TRAÇADO DE
CIRCUITOS IMPRESSOS

Pêricles Gama Negri

Orientador: Nelson Castro Machado

Tese de Mestrado apresentada
à Faculdade de Engenharia de
Campinas da Universidade Es-
tadual de Campinas.

Janeiro - 1974



BIBLIOTECA DA AREA DE ENGENHARIA
UNICAMP

50
FEC
et. 1

AGRADECIMENTOS

Agradeço ao Dr. Nelson Castro Machado, meu orientador, pelo incentivo necessário para a realização desse trabalho e à Srta. Eliana Fussako Tsuboka, pelo esforço no trabalho de datilografia.

ÍNDICE

INTRODUÇÃO	1
1 - O PROBLEMA DO TRAÇADO DOS CIRCUITOS IMPRESSOS	3
1.1 - ALGORITMOS "SÉRIE"	4
1.1.1 - ALGORITMOS DE LEE E CONGÊNERES	4
1.1.2 - ALGORITMOS HEURÍSTICOS	5
1.2 - ALGORITMOS "PARALELO"	7
1.2.1 - ALGORITMOS CELULARES	7
1.2.2 - ALGORITMO DE SIMULAÇÃO TOPOGRÁFICA	7
1.2.3 - ALGORITMO DA ABERTURA DESCENDENTE	8
1.2.4 - ALGORITMO DOS CANAIS	9
2 - IMPLEMENTAÇÃO DO ALGORITMO DOS CANAIS	13
2.1 - DESIGNAÇÃO DOS ESPAÇOS	16
2.2 - DESIGNAÇÃO DOS CANAIS VERTICAIS	21
2.3 - RELOCAÇÃO DOS SEGMENTOS VERTICAIS	23
2.4 - DESIGNAÇÃO DOS CANAIS HORIZONTAIS	26
2.5 - RELOCAÇÃO DOS SEGMENTOS HORIZONTAIS	28
3 - UTILIZAÇÃO DO SISTEMA	31
3.1 - ENTRADAS	33
3.2 - SAÍDAS	36
3.2.1 - PARÂMETROS GEOMÉTRICOS DO CARTÃO	36
3.2.2 - SEGMENTOS VERTICAIS	37
3.2.3 - SEGMENTOS HORIZONTAIS	38
3.2.4 - FUROS METALIZADOS	39
3.2.5 - MATRIZES DO ALGORITMO	40

4 - TESTE DE PERFORMANCE DO ALGORITMO	42
4.1 - NÚMERO MÉDIO DE FALHAS	42
4.2 - COMPRIMENTO MÉDIO DAS CONEXÕES	43
4.3 - TEMPO DE EXECUÇÃO	43
5 - CONCLUSÕES	46
BIBLIOGRAFIA	48

INTRODUÇÃO

O surgimento dos circuitos integrados provocou, como consequência natural, um aumento muito grande na densidade de conexões entre os componentes de um cartão de circuito impresso. Qualquer pessoa que tenha se dedicado alguma vez ao projeto de um desses cartões, nas suas diversas fases, sabe que essa é uma tarefa que pode passar de divertida a impossível à medida que o número de conexões aumenta. É perfeitamente normal encontrar-se hoje em dia, principalmente em computadores digitais, cartões de 2 faces com até 50 circuitos integrados digitais do tipo "dual-in-line". O projeto de um desses cartões pode levar até 6 semanas para ser completado. O custo envolvido nesse tempo é suficiente para justificar a necessidade da automação, senão total, pelo menos parcial do projeto de circuitos impressos.

É costume dividir-se em 3 partes distintas o projeto de circuitos impressos:

- * distribuição dos componentes entre os diversos cartões
- * localização dos componentes em um cartão
- * traçado das conexões em um cartão

Entre as três, a última tarefa é sem dúvida a mais demorada, tediosa e à que mais tempo de pesquisa tem sido dedicado nos últimos anos. Esse trabalho se refere justamente àquela terceira fase do projeto e pretende ser o núcleo de um possível sistema automatizado de produção de circuitos impressos a ser implementado no Laboratório de de Campinas.

No capítulo 1 é realizado um estudo geral dos algoritmos utilizados atualmente para o traçado de circuitos impressos sendo ressaltadas as vantagens e desvantagens de cada um. No capítulo 2, o algoritmo dos Canais, foi escolhido sendo mostradas as idéias utilizadas na sua implementação. No capítulo 3 são dadas as diretrizes para utilização de um programa que implementa no PDP-10 o algoritmo dos canais, sendo apresentado um exemplo real. No capítulo 4 o sistema foi testado de modo a medir a sua performance em três parâ-

metros, quais sejam, número de falhas, comprimento médio das conexões e tempo de execução. Finalmente no capítulo 5 são feitas críticas e sugestões visando a implantação de um sistema automático - para traçado de circuitos impressos.

1 - O PROBLEMA DO TRAÇADO DE CIRCUITOS IMPRESSOS

Basicamente o conjunto de conexões que constituem um circuito impresso pode ser considerado como um grafo cujos vértices ("nodes") são representados pelos terminais dos componentes e cujas arestas ("edges") são representadas pelas conexões propriamente. Assim o problema do traçado de um circuito impresso poderia ser equacionado do seguinte modo:

- * divisão do grafo em subgrafos planares
- * traçado dos subgrafos planares, sem interseção entre as arestas, em um ou mais planos (representados pelas faces do cartão de circuito impresso), conectados entre si pelas arestas de corte ("cutedges" - implementadas através de furos metalizados).

Para a primeira fase existem diversos algoritmos programáveis, geralmente bastante complexos. Para a segunda fase, porém, não existe nenhuma solução prática (a definição de um grafo planar não é construtiva). A ordem escolhida para o traçado das conexões pode determinar uma impossibilidade no problema. De acordo com o algoritmo utilizado no traçado das conexões, existem algumas idéias - acerca da ordem em que as conexões devem ser traçadas [1, 2]. Assim é mais ou menos óbvio que as conexões mais longas devem ser deixadas para o fim (com o inconveniente de se tornarem mais longas e complexas). Porém, estas idéias não são longas e construtivas uma solução. Desse modo o equacionamento do problema do traçado de circuitos impressos apresentado acima se torna artificial e de pouca utilidade. Na prática os algoritmos utilizados eliminam a divisão em subgrafos planares (na realidade a fazem implicitamente) já que ela não apresenta nenhuma vantagem.

Para efeito didático os algoritmos de traçado de circuitos impressos serão divididos em dois grupos, algoritmos "série" e algoritmos "paralelo". São chamados algoritmos "série" aqueles em que cada conexão é traçada independentemente, segundo uma ordem predeterminada e sem levar em conta possíveis interferências com futuras conexões a serem traçadas. Os algoritmos "paralelo" são aqueles em

que as conexões são planejadas e ajustadas conjuntamente de modo a interferirem entre si o mínimo possível.

1.1 - ALGORITMOS "SÉRIE"

1.1.1 - ALGORITMO DE LEE E CONGÊNERES

Deve-se a Lee [3] o primeiro algoritmo prático capaz de traçar num plano uma conexão entre dois pontos. Nele, o plano é dividido num reticulado de quadrados, chamados células, cujo lado é igual à distância mínima permitida entre duas conexões. As conexões são constituídas de segmentos horizontais e verticais que passam pelos centros das células. Os pontos a serem conectados são colocados em células que são marcadas com Origem (O) e Terminal (T). O processo de traçado de uma conexão se inicia marcando-se as células vizinhas da célula O (não diagonalmente) com o número 1. As células vizinhas àquelas marcadas com o número 1 são por sua vez marcadas com o número 2 e assim por diante em ordem crescente até que a célula T seja alcançada. Traça-se agora a conexão simplesmente tomando-se uma sequência de células numeradas em ordem decrescente a partir da célula T. Observe-se que existem mais de uma sequência possível de ser tomada, pois o algoritmo determina todos os caminhos possíveis entre as células O e T. Todas sequências possuem o mesmo comprimento que é mínimo. Escolhe-se uma sequência adotando-se um critério qualquer, como por exemplo o de minimizar o número de segmentos. As células utilizadas na conexão são marcadas como ocupadas e não são mais utilizadas no processo de traçado de outras conexões.

A grande vantagem do algoritmo de Lee reside no fato de que se existir um caminho possível entre 2 pontos, ele fatalmente será encontrado. Porém, ele possui desvantagens que na maioria das vezes o tornam proibitivo. A primeira delas consiste na grande quantidade de memória que ele exige ao ser programado, já que, sendo o reticulado muito fino, um grande número de células tem que ser representado.

A segunda desvantagem está no fato de que ele é intrinsecamente lento, já que, conforme foi dito anteriormente, todos os caminhos

possíveis entre dois pontos são determinados, o que implica num grande número de operações. Finalmente a terceira desvantagem está na dificuldade em se fazer correções manualmente, pois no caso de se ter que completar conexões não realizadas pelo algoritmo, tem-se normalmente que remover um grande número de conexões já traçadas a fim de se poder acrescentar novas linhas [4]. Na figura 1.1 é mostrado o desenvolvimento do algoritmo de Lee para o traçado da conexão entre as células 0 e T. As células marcadas com um "c" constituem uma conexão previamente traçada. Akers [5] mostrou que realmente bastam 2 símbolos, "1" e "2" por exemplo, para numerar as células (na sequência 11221122 ...)

O algoritmo Lee é aplicável somente a cartões de circuito impresso de 1 face. Heiss [6] e Geyer [7] estenderam o algoritmo de Lee para cartões multifaces. No algoritmo de Heiss o procedimento de Lee é aplicado em cada plano separadamente e de modo que o conjunto de células vizinhas de uma célula qualquer inclui não somente as 4 células coplanares como também as 2 células dos planos imediatamente superior e imediatamente inferior. No algoritmo de Geyer as células são consideradas multidimensionais e cada célula se comunica com uma vizinha por qualquer plano. O procedimento de Lee é executado simultaneamente em todos os planos. No processo de traçado da conexão, após os ciclos de numeração crescente, esta pode se manter num plano ou passar de um plano para outro. A célula em que há a mudança de plano constituir-se-á num furo metalizado. O algoritmo de Geyer além de mais rápido que o de Heiss apresenta uma grande flexibilidade que permite que se faça conexões entre um plano e uma reta e conexões do tipo ortogonal em que os segmentos horizontais e verticais de uma conexão são colocados em planos diferentes. Porém ambos os algoritmos empregam fundamentalmente o procedimento de Lee com suas vantagens e desvantagens

1.1.2 - ALGORITMOS HEURÍSTICOS

A alternativa para os algoritmos baseados no procedimento de Lee é um conjunto de algoritmos fundamentalmente semelhantes chamados de heurísticos ou ainda de algoritmos de linhas. Basicamente eles consistem em traçar linhas a partir dos pontos a serem conectados e fazer com que essas linhas se encontrem diretamente ou indiretamen

te a partir de linhas auxiliares. No algoritmo de Hightower [8] - passa-se uma linha por cada ponto do par a ser conectado, sendo es - sas duas linhas perpendiculares entre si. Se as duas linhas se in- tersetarem a conexão estará terminada, caso contrário são traçadas perpendiculares (linhas de escape) ãquelas linhas, para as quais - agora se procura uma interseção. O processo continua, traçando-se linhas de escape perpendiculares às linhas de escape anteriores - até que elas se encontrem ou até que não seja mais possível traçá- las. No algoritmo de Aramaki, Arimoto e Kawabata [1] é traçada 1 - linha horizontal a partir de cada ponto do par a ser interconecta- do e em direção ao outro ponto. Tenta-se então achar uma conexão - vertical entre as duas linhas, dentro da área compreendida entre - os dois pontos. Se não for possível desloca-se as horizontais para baixo segundo uma distância pré-fixada e tenta-se de novo. Se se conseguir as extremidades das horizontais são ligadas aos pontos a - través de segmentos verticais, caso contrário a conexão é abandona- da. Esses algoritmos podem ser facilmente estendidos para cartões de 2 ou mais faces, simplesmente dividindo-se as conexões entre os diversos planos ou, de uma maneira mais eficiente, usando-se cone- xões do tipo ortogonal em que os segmentos horizontais e verticais de uma conexão são separados, colocados em planos distintos e in- terconectados através de furos metalizados. Esse procedimento mi - nimiza o problema de cruzamento entre as conexões e permite uma - maior densidade já que num mesmo plano, nenhuma linha interrompe - outra.

Esses algoritmos se caracterizam principalmente por sua rapidez e pela pouca memória que utilizam. O pequeno requisito de memória se justifica pelo fato das entidades representadas serem retas e não pontos como no algoritmo de Lee, o que obviamente reduz o número - de entidades a serem representadas. A sua rapidez se explica pelo pequeno número de operações necessárias para o traçado de uma cone - xão. São essas duas características que tornam esses algoritmos - muito mais práticos que os algoritmos baseados no procedimento de Lee. Além do mais as conexões não completadas por esses algoritmos podem ser facilmente feitas a mão porque as direções das conexões são predeterminadas apresentando uma regularidade no seu traçado [4]. Em contraposição, essa rigidez na definição das conexões im-

plica em algumas desvantagens. A primeira delas está no fato de - que muitas conexões que seriam completadas pelo algoritmo de Lee não o são por esses algoritmos pois não há liberdade no traçado dessas conexões. A segunda é que o comprimento das conexões geralmente aumenta pois muitas delas que poderiam ser traçadas diretamente são quebradas num número maior de segmentos horizontais e verticais. Finalmente, quando se adota conexões ortogonais para cartões multifaces o número de furos metalizados normalmente é grande pois sendo as conexões quebradas em vários segmentos horizontais e verticais, estes tem que ser interligados através daqueles.

1.2 - ALGORITMOS "PARALELO"

1.2.1 - ALGORITMOS CELULARES

Esses algoritmos associam uma certa idéia de paralelismo ao algoritmo de Lee [9, 10]. Neles o plano é dividido em células cujo tamanho sendo bem maior que o das células do algoritmo de Lee, incluem várias dessas permitindo que mais de uma conexão passe por uma célula. Obviamente o tamanho das células determina sua capacidade. O procedimento de Lee é aplicado para cada conexão, porém ele não determina o seu traçado final mas sim as células pelas quais a conexão passará. Somente depois que isso foi feito para todas as conexões é que elas são traçadas de modo que não haja interferência entre elas dentro das células.

Os algoritmos celulares reduzem no algoritmo de Lee a interferência destrutiva que cada conexão exercia sobre outras que fossem traçadas depois. Além do mais o requisito de memória e o número de operações são reduzidos pois o número de células diminui. Não foi apresentada uma extensão eficiente dos algoritmos celulares para cartões multifaces. A idéia de Cooper e Brow consiste em colocar furos metalizados em posições pré-determinadas o que não é eficiente.

1.2.2 . ALGORITMO DE SIMULAÇÃO TOPOGRÁFICA

Esse algoritmo se deve a uma idéia bastante engenhosa baseada na construção de uma estrutura topográfica [11]. Nele, como no algoritmo de Lee, o plano é dividido num reticulado de células quadra-

das. A estrutura topográfica é implementada associando-se à cada célula uma altura, formando-se assim "picos", "cordilheiras" e "vales". Inicialmente é atribuída uma altura máxima às células que contêm os terminais dos componentes e às células da periferia formando-se assim "picos" e uma "cordilheira" respectivamente. Às demais células são atribuídas alturas que variam inversamente com a distância à(s) célula(s) de altura máxima mais próxima(s). Isso define "vales" e "bacias" na estrutura. Cada conexão avança um segmento por vez, e o próximo segmento só é traçado depois que todas as conexões avançaram um segmento. Assim todas as conexões são traçadas simultaneamente. As conexões são traçadas a partir de ambas as extremidades. Um segmento é determinado "afundando-se" a célula alvo (último segmento traçado da extremidade oposta) e com ela toda a estrutura harmonicamente. O segmento então é traçado a partir da atual célula origem na direção de maior declive. Após isso a estrutura volta à posição original. À medida que as conexões vão sendo traçadas novas "cordilheiras" vão sendo definidas e toda a estrutura vai sendo modificada dinamicamente. Observe-se que esse processo faz com que as conexões tendam a se repelir.

Apesar da vantagem do paralelismo do algoritmo, ele sofre dos mesmos males do algoritmo de Lee no que se refere à memória e tempo de processamento, acrescidos de uma complexidade maior de programação. Os autores não especificam como estender o algoritmo para cartões multifaces.

1.2.3 - ALGORITMO DA ABERTURA DESCENDENTE

Esse algoritmo é tipicamente um algoritmo de linhas com conexões ortogonais ao qual foi acrescentada uma capacidade de paralelismo e que se aplica somente para cartões de 2 faces [12]. Nele o plano é preenchido com um padrão de furos metalizados de dois tipos, furos de terminais (P) e furos de conexões entre os planos (V). Os furos são dispostos em colunas do mesmo tipo na sequência PVPVVPV PVV... Entre cada 2 colunas de furos existe uma linha vertical sobre a qual são traçadas segmentos verticais das conexões, e entre cada 2 linhas de furos existem 2 linhas horizontais sobre as quais são traçadas segmentos horizontais das conexões. Cada conexão é constituída de um segmento vertical e de um segmento horizontal. O

segmento vertical se inicia no furo terminal superior, ocupa qualquer uma das linhas verticais adjacentes e se estende até a linha de furos do furo terminal inferior, terminando em qualquer um dos furos V adjacentes. O segmento horizontal se inicia no fim do segmento vertical, ocupa qualquer das 4 linhas horizontais adjacentes e se estende até o furo terminal inferior. Todas as conexões são definidas dessa maneira sem se preocupar com possíveis interferências entre elas. Depois que todas as conexões são definidas percorre-se o plano passo a passo de cima para baixo com uma abertura suficiente para abranger somente uma linha de furos por vez (e as 4 linhas horizontais adjacentes). Dentro dessa abertura então resolve-se os conflitos que existirão entre as diversas conexões como coincidência entre os segmentos horizontais, os segmentos verticais e os furos do tipo V. Se alguma conexão não puder ser completada dentro da posição atual da abertura, ela é transferida para a próxima posição. Se ainda assim não for possível ela é eliminada.

O algoritmo apresenta as vantagens dos algoritmos heurísticos, - quais sejam rapidez e pequeno requisito de memória acrescidas do - fato das conexões serem planejadas em conjunto. Porém ele perde em flexibilidade (estrutura rígida de furos) e rapidez para o algoritmo dos canais a ser examinado em seguida.

1.2.4 - ALGORITMO DOS CANAIS

O algoritmo dos canais [13], como o algoritmo anterior, é um algoritmo heurístico com conexões ortogonais, para circuitos de 2 e 4 fios e aplicação para circuitos integrados digitais de 2, 4, 6, 8 e 16 linhas. Os circuitos integrados (CIs) são dispostos no cartão segundo uma matriz retangular. O cartão é dividido horizontalmente e verticalmente em regiões chamados espaços (v. figura 1.2). Um espaço vertical compreende uma coluna de CIs mais metade, de cada lado, do espaço entre as colunas dos CIs. Um espaço horizontal compreende metade da largura de um CI mais metade do espaço entre as linhas dos CIs. Os espaços verticais e horizontais são divididos em canais verticais e horizontais respectivamente que são linhas que se estendem por toda a extensão do espaço e nas quais são traçados os segmentos verticais e horizontais das conexões. O espaçamento - entre os canais verticais é de 1/20 de polegada entre os CIs e de

1/10 de polegada dentro dos CIs. O espaçamento entre os canais horizontais é de 1/20 de polegada.

Observe-se que os pinos do(s) conector(es) são divididos em grupos de modo que cada grupo equivale a um meio-CI.

O traçado das conexões consiste de 3 fases distintas:

- * designação de espaços
- * designação de canais
- * relocação de segmentos

Na designação de espaços todas as conexões são definidas e são atribuídos espaços aos seus segmentos. Existem 2 tipos de conexões (ver figura 1.4). Uma conexão entre pinos situados no mesmo espaço horizontal é constituído de 3 segmentos, 1 segmento horizontal e 2 segmentos verticais que conetam o horizontal aos pinos. Observe-se que os segmentos verticais são "implícitos" isto é, eles não ocupam nenhum canal vertical estando completamente definidos. Uma conexão entre pinos quaisquer é constituída de 5 segmentos, 2 segmentos horizontais, 1 segmento vertical que coneta os segmentos horizontais entre si, e 2 segmentos verticais que conectam os segmentos horizontais aos pinos.

Na segunda fase do processo, que é a responsável pela eficiência do algoritmo, cada espaço em separado é analisado e aos seus segmentos são atribuídos canais de tal modo que o número mínimo possível de segmentos é colocado, dentro de cada canal e sem que haja coincidência entre eles.

Pode acontecer que na segunda fase algum espaço utilize um número de canais maior do que o número máximo e um espaço adjacente utilize poucos canais. Na terceira fase então, esses casos são analisados e há uma redistribuição dos segmentos entre os espaços de modo a minimizar o número de conexões que não possam ser traçadas.

O algoritmo dos canais se caracteriza sobretudo pela sua rapidez e facilidade de programação

4	3	2	3	4	5	6	7	8	9	10	11
3	2	1	2	3	4	5	C	9	10	11	12
2	1	0	1	2	3	4	C	10	11	12	13
3	2	1	2	3	4	5	C	11	12	13	14
4	3	2	3	4	5	6	C	12	13	14	
5	4	3	4	5	6	7	C	13	14		
6	5	4	5	C	C	C	C	14	T		
7	6	5	6	C	10	11	12	13			
8	7	6	7	8	9	10	11	12			
9	8	7	8	9	10	11	12	13			

FIGURA 1.1 - Desenvolvimento do algoritmo de Lee

Esp. Vertical 1

Esp. Vertical 2

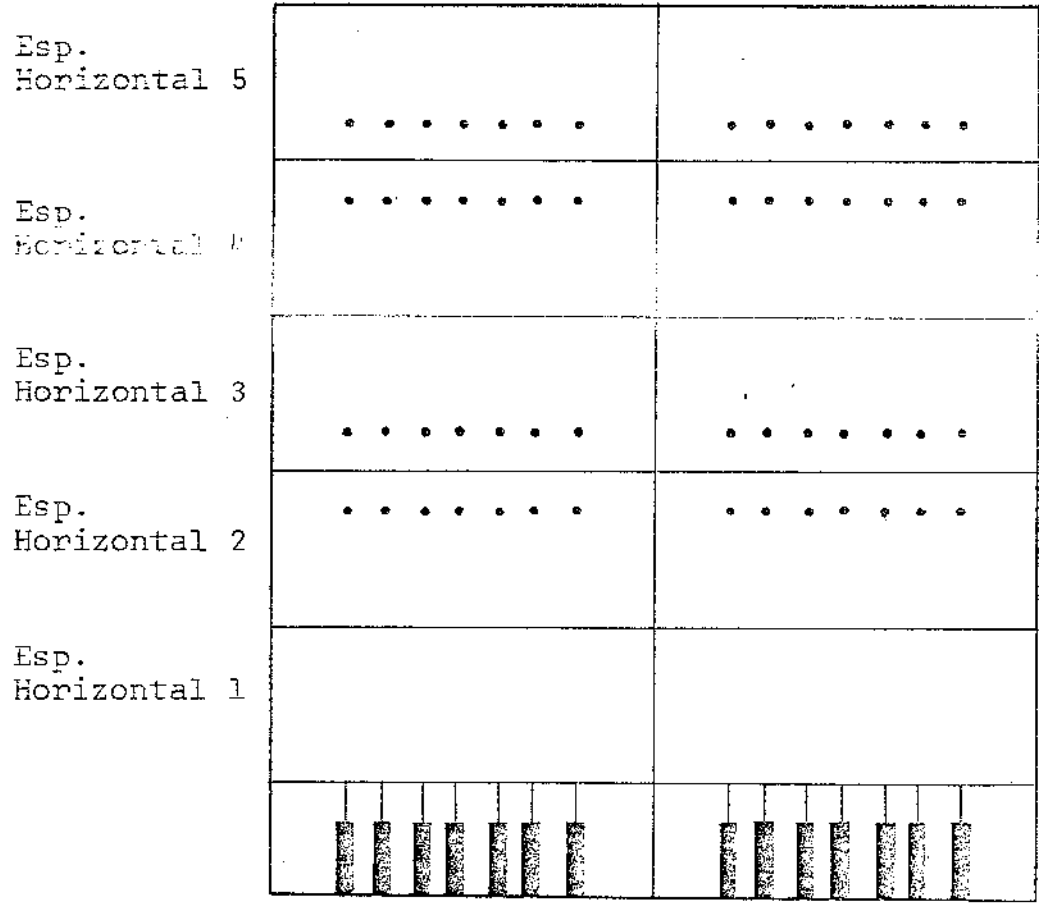


FIGURA 1.2 - Divisão em espaços

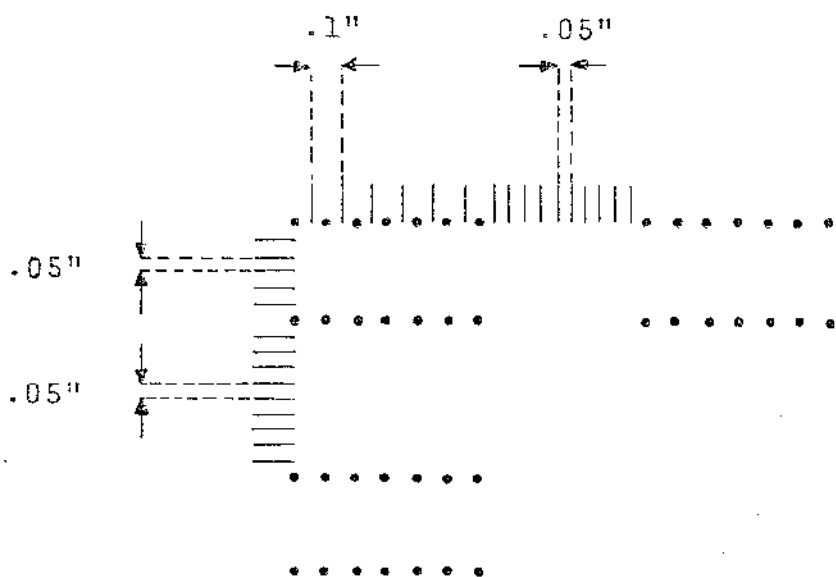


FIGURA 1.3 - Distribuição dos canais

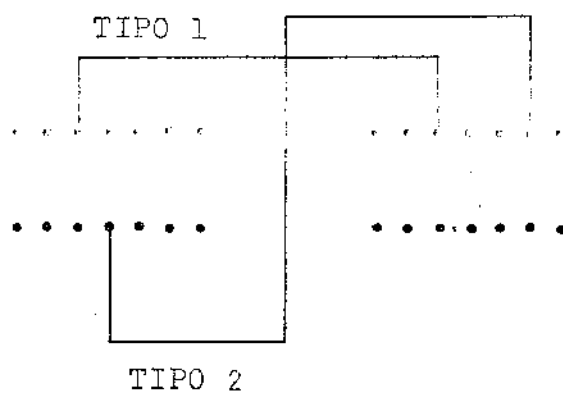


FIGURA 1.4 - Tipos de conexões

2. IMPLEMENTAÇÃO DO ALGORITMO DOS CANAIS

Do capítulo anterior depreende-se que nenhum algoritmo é capaz de por si só resolver integralmente o problema do traçado de circuitos impressos. Porém é mais ou menos óbvio que qualquer sistema para isso planejado deverá ter como base um algoritmo heurístico. Isso se deve às suas características de rapidez e pequeno requisito de memória que são fatores determinantes de custo. Porém como esses algoritmos não garantem que sempre que for possível uma conexão será traçada, é desejável que o algoritmo seja do tipo "paralelo" de modo a minimizar o número de conexões não concluídas. Isso restringe a uma escolha entre o algoritmo da abertura descendente e o algoritmo dos canais. Apesar de fundamentalmente semelhantes, decidiu-se por esse último devido principalmente a sua maior facilidade de programação e maior flexibilidade, apesar dele ser específico para circuitos integrados digitais do tipo "dual-in-line."

O algoritmo foi implementado em 5 fases distintas e na seguinte ordem:

- * Designação dos Espaços
- * Designação dos Canais Verticais
- * Relocação dos Segmentos Verticais
- * Designação dos Canais Horizontais
- * Relocação dos Segmentos Horizontais

É necessária esta ordem devido ao fato seguinte. Suponha-se na figura 2.1 que os canais horizontais já tenham sido designados, os segmentos horizontais relocados, os canais verticais designados, e que agora os segmentos verticais vão ser relocados de mo-

do que o segmento A passará a ocupar a posição indicada na figura 2.2. Isso fará com que os segmentos horizontais B e C passem a coincidir o que não pode acontecer. Logo os canais horizontais te não que ser redesignados e todo o trabalho anterior será inútil. Por isso os segmentos verticais tem que ser processados antes dos segmentos horizontais.

Todo o desenvolvimento do algoritmo gira em torno de 2 matrizes de 2 dimensões, MSH (matriz de segmentos horizontais) e MSV (matriz de segmentos verticais). Cada linha dessas matrizes corresponde a um espaço (vertical ou horizontal dependendo da matriz) e cada elemento de uma linha descreve um segmento do espaço correspondente. Além de descrever um segmento, um elemento de uma matriz indica através de apontadores, os elementos da outra matriz que a ele estão conectados. Cada elemento é formado de 8 campos de 9 bits cujo significado é mostrado abaixo.

Elemento da Matriz de Segmentos Horizontais

CANALH	IH	ESVE	ASVE	OSVE	ISVT	ASVT	OSVT
--------	----	------	------	------	------	------	------

CANALH - Canal Horizontal. Contém o canal horizontal para o qual o segmento foi designado.

IH - Indicador Horizontal. Indica se o segmento já foi ou não designado para algum canal.

ESVE - Espaço do Segmento Vertical à Esquerda. Se o segmento se conectar à esquerda com um segmento vertical indica o espaço desse segmento. Se se conectar com um pino indica Circuito Integrado ou o conector que o con-

tém. Se for conector ele é representado em excesso de 100.

- ASVE - Apontador do Segmento Vertical à Esquerda. Se o segmento se conectar à esquerda com um segmento vertical aponta para ele na MSV. Se se conectar com um pino contém o número 511.
- CSVE - Canal do Segmento Vertical à Esquerda. Se o segmento se conectar à esquerda com um segmento vertical, contém o canal para o qual o segmento foi designado. Se se conectar com um pino, contém o número do pino.
- ESVD - Espaço do Segmento Vertical à Direita. Semelhante à ESVE.
- ASVD - Apontador do Segmento Vertical à Direita. Semelhante a ASVE.
- CSVD - Canal do Segmento Vertical à Direita. Semelhante à CSVE.

Elemento da Matriz de Segmentos Verticais

CANALV	IV	ESHS	ASHS	CSHS	ESHI	ASHI	CSHI
--------	----	------	------	------	------	------	------

- CANALV - Canal Vertical. Contém o canal vertical para o qual o segmento foi designado.
- IV - Indicador Vertical. Indica se o segmento já foi ou não designado para algum canal.
- ESHS - Espaço do Segmento Horizontal Superior. Indica o espaço do segmento horizontal ao qual o segmento se conecta na extremidade superior.

- ASHS - Apontador do Segmento Horizontal Superior. Aponta para o elemento de MSH que se conecta ao segmento na extremidade superior.
- CSHS - Canal do Segmento Horizontal Superior. Contém o canal para o qual foi designado o elemento de MSH que se conecta ao segmento na extremidade superior.
- ESHI - Espaço do Segmento Horizontal Inferior. Semelhante a ESHS.
- ASHI - Apontador do Segmento Horizontal Inferior. Semelhante a ASHS.
- CSHI - Canal do Segmento Horizontal Inferior. Semelhante a CSHS.

Observe-se que para muitos dos campos acima definidos, não seriam necessários 9 bits. Devido a dificuldades de manipular "bytes" de tamanhos diferentes na linguagem utilizada, fez-se todos os campos com comprimentos iguais.

Associadas às matrizes MSH e MSV, existem 2 matrizes, colunas APH e APV que possuem tantas linhas quantos são os espaços horizontais e verticais respectivamente, e cujos elementos indicam o número atual de segmentos dos espaços correspondentes.

2.1 - DESIGNAÇÃO DOS ESPAÇOS

Nessa fase do algoritmo cada conexão é lida e classificada em 3 tipos:

- * Conexão Direta (ver figura 2.3) - os segmentos não ocupam nenhum canal.

- * Conexão do tipo 1 (ver figura 1.4) - entre pinos situados no mesmo espaço horizontal.
- * Conexão do tipo 2 (ver figura 1.4) - entre pinos situados em espaços horizontais diferentes.

Se a conexão for direta ela é simplesmente colocada numa lista de conexões diretas e não entra na formação de MSH e MSV. Se a conexão for do tipo 1 ou 2 ela é dividida no número de segmentos correspondentes e estes são colocadas nas matrizes MSH e MSV. Uma conexão é definida por:

- TIP00 - Indica se o pino origem da conexão é de conector ("CO") ou de circuito integrado ("CI").
- NO - Indica o número do circuito integrado ou do conector ao qual o pino origem pertence.
- PO - Indica o número de pino origem.
- TIP0T - Indica se o pino terminal da conexão é de conector ("CO") ou de circuito integrado ("CI").
- NT - Indica o número do circuito integrado ou do conector ao qual o pino terminal pertence.
- PT - Indica o número do pino terminal.

Definição das variáveis envolvidas no algoritmo:

CONDIR - Matriz onde são colocadas as conexões diretas. À cada linha corresponde uma conexão que é descrita por 4 colunas. A primeira coluna contém o número de circuito integrado ou conector que contém o pino origem. No caso de conector ele é representado em excesso de 100. A segunda coluna contém o número de pino origem. As outras duas co

lunas descrevem o pino terminal de maneira análoga às duas primeiras.

- APCD - Apontador que indica o número de conexões que CONDIR contém.
- ERO - Espaço horizontal do pino origem.
- EVO - Espaço vertical do pino origem.
- EHT - Espaço horizontal do pino terminal.
- EVT - Espaço vertical do pino terminal.
- EHD - Espaço horizontal do pino da direita.
- EHE - Espaço horizontal do pino da esquerda.
- EVD - Espaço vertical do pino da direita.
- EVE - Espaço vertical do pino da esquerda.
- EHS - Espaço horizontal do pino superior.
- EHI - Espaço horizontal do pino inferior.
- EV - Espaço vertical do segmento vertical de uma conexão do tipo 2.
- PD - Pino da direita.
- PE - Pino da esquerda.
- ND - Número do circuito integrado ou do conector do pino da direita.
- NE - Número do circuito integrado ou do conector do pino da esquerda.
- TABELA - Matriz que descreve os circuitos integrados. À cada linha corresponde um CI que é descrito por 3 colunas. A primeira coluna contém a linha em que o CI está colocado, a segunda contém a coluna em que o CI está colocado e a terceira o número de pinos do CI.

Descrição do algoritmo

- PASSO 1 - Leia próxima conexão: TIPOO, NO, PO, TIPOT, NT, PT. Se não houverem mais conexões a serem feitos, FIM.
- PASSO 2 - Determine EVO, EHO, EVT, EHT a partir de TABELA.
- PASSO 3 - Se a conexão for direta coloque-a em CONDIR e vá para o passo 1.
- PASSO 4 - Determine EVD e EVE
- EVD = máximo (EVO, EVT)
- EVE = mínimo (EVO, EVT)
- PASSO 5 - Determine EHS e EHI
- EHS = máximo (EHO, EHT)
- EHI = mínimo (EHO, EHT)
- PASSO 6 - Determine EHD e EHE
- Se EVD=EVO então EHD=EHO, EHE=EHT
- senão EHD=EHT, EHE=EHO
- PASSO 7 - Determine PD, PE, ND e NE a partir de TABELA
- PASSO 8 - Se EHS $\neq$ EHI (conexão do tipo 2) então vá para o passo 9 senão (conexão do Tipo 1):
- defina o segmento horizontal na matriz MSH:
- APH(EHS)=APH(EHS)+1
- ASVE(EHS,APH(EHS))=511
- CSVE(EHS,APH(EHS))=PE
- Se PE é pino de conector
- então ESVE(EHS,APH(EHS))=NE+100
- senão ESVE(EHS,APH(EHS))=NE
- ASVD(EHS,APH(EHS))=511
- CSVD(EHS,APH(EHS))=PD
- Se PD é pino de conector
- então ESVD(EHS,APH(EHS))=ND+100

senão ESVD(EHS,APH(EHS))=ND

- vá para o passo 1

PASSO 9 - EHS $\neq$ EHI (conexão do tipo 2):

- determinar EV tal que $EVE \leq EV \leq EVD$ e APV(EV) seja mínimo
- definir o segmento vertical EV na matriz MSV:

APV(EV)=APV(EV)+1

APH(EHS)=APH(EHS)+1

APH(EHI)=APH(EHI)+1

ESHS(EV,APV(EV))=EHS

ASHS(EV,APV(EV))=APH(EHS)

ESHI(EV,APV(EV))=EHI

ASHI(EV,APV(EV))=APH(EHI)

- definir os segmentos horizontais na matriz MSH:

ESVE(EHD,APH(EHD))=EV

ASVE(EHD,APH(EHD))=APV(EV)

se PD é pino de conector

então ESVD(EHD,APH(EHD))=ND+100

senão ESVD(EHD,APH(EHD))=ND

ASVD(EHD,APH(EHD))=511

CSVD(EHD,APH(EHD))=PD

se PE é pino de conector

então ESVE(EHE,APH(EHE))=NE+100

senão ESVE(EHE,APH(EHE))=NE

ASVE(EHE,APH(EHE))=511

CSVE(EHE,APH(EHE))=PE

ESVD(EHE,APH(EHE))=EV

ASVD(EHE,APH(EHE))=APV(EV)

- vá para o passo 1

2.2 - DESIGNAÇÃO DOS CANAIS VERTICAIS

Na fase de designação de espaços as conexões foram divididas em segmentos e estes colocados nos espaços respectivos. Porém não foi determinado em que canal do espaço o segmento seria traçado. Nessa fase isso será feito para os segmentos verticais e de modo a utilizar um número mínimo de canais, isto é, cada canal deverá conter um número máximo de segmentos. O processo é repetido para cada espaço vertical e da seguinte maneira. Cada segmento tem uma extremidade superior e uma extremidade inferior definidas pelos segmentos horizontais que a ele se conectam. Como para esses segmentos horizontais não foram atribuídos canais, aquelas extremidades são definidas somente pelos espaços que esses segmentos ocupam. O processo se inicia tomando-se todos os segmentos de um espaço e colocando-os numa lista auxiliar. Depois disso, retira-se da lista um segmento cuja extremidade superior seja máxima e coloca-se no primeiro canal. Retira-se agora da lista o segmento cuja extremidade superior seja máxima mas que seja menor que a extremidade inferior do segmento colocado anteriormente no canal e coloca-se-o também. O processo se repete até que nenhum segmento da lista preencha aqueles requisitos. Passa-se então para o próximo e assim por diante até que todos os segmentos sejam retirados da lista.

Definição das variáveis envolvidas no algoritmo

E - Espaço vertical sendo analisado

NLINHAS - Número de linhas de CIs

NEH - Número de espaços horizontais. $NEH=2 \times NLINHAS+1$

NCOL - Número de colunas de CIs.

NEV - Número de espaços verticais. $NEV=NCOL$

FILA - Lista auxiliar que contém os segmentos do espaço sendo analisado.

APF - Apontador da FILA. Indica quantos elementos FILA contém.

S - Segmento retirado de FILA.

LIMSUPE - Indica a extremidade inferior do último segmento colocado no canal sendo designado.

NMCV - Matriz coluna, que possui tantas linhas quantos são os espaços verticais. Cada elemento indica o número de canais utilizado no espaço vertical respectivo.

Descrição do algoritmo

PASSO 1 - $E=1$

PASSO 2 - Se $APV(E)=0$ então vá para o passo 8

PASSO 3 - Coloque os segmentos de E em FILA

$APF=APV(E)$

$NMCV(E)=1$

PASSO 4 - $LIMSUPE=NEH+1$

PASSO 5 - Se não existe em FILA um segmento S tal que $ESHS(E,S) < LIMSUPE$ e $ESHS(E,S)$ seja máximo, então pegue novo canal
 $NMCV(E)=NMCV(E)+1$ e vá para o passo 2.

PASSO 6 - Retire S de FILA e coloque-o no canal $NMCV(E)$

$APF=APF-1$

$CANALV(E,S)=NMCV(E)$

PASSO 7 - Se $APF=0$ (FILA está vazia) vá para o passo 9

PASSO 8 - Se $E=NEV$ (todos os espaços foram analisados)

então FIM

senão tome próximo espaço, $E=E+1$, e vá para o passo 2.

PASSO 9 - $LIMSUPE = ESHI(E,S)$, vá para o passo 5.

2.3 - RELOCAÇÃO DOS SEGMENTOS VERTICAIS

Nessa fase verifica-se para cada espaço vertical E o número de canais utilizados na fase anterior. Se esse número for maior que o número de canais de um espaço vertical, os segmentos que ocupam os canais em excesso são transferidos para o espaço $E+1$. Isso é feito deslocando-se os segmentos dos canais inferiores para os canais superiores no espaço $E+1$ e preenchendo-se os canais inferiores com os segmentos transferidos do espaço E . É óbvio que essa transferência pode afetar os espaços $E+2$, $E+3$, etc. Observe-se que não existe nenhum limite para o número de canais do espaço $E+1$ a serem ocupados por segmentos do espaço E , porém, verificou-se na prática, que, se não se impuser um limite, os segmentos horizontais tenderão a ficar muito longos, o que aumentaria o número de falhas no traçado das conexões. Assim decidiu-se permitir que somente os canais localizados antes do CI no espaço $E+1$, sejam acessíveis aos segmentos a serem transferidos do espaço E .

Pode acontecer, devido aos processos de distribuição de canais verticais e de relocação de segmentos verticais, que os segmentos horizontais tenham sua orientação trocada. Assim um segmento horizontal que estivesse se conectando à esquerda com um pino e à direita com um segmento vertical passaria a se conectar à esquerda com o segmento e à direita com o pino. Devido a isso, depois que um segmento vertical é definitivamente colocado num canal, os segmentos horizontais que a ele se interconectam são examinados e se necessário reorientados.

Definição das variáveis envolvidas no algoritmo

E - Espaço vertical sendo analisado

S - Um segmento vertical do espaço E

ESH - Espaçamento horizontal entre CIs. É dado por um número inteiro de vigésimos de polegadas.

NMP - Número de pinos do CI que possui o maior número de pinos.

NCEV - Número de canais por espaço vertical.

$$NCEV = NMP / 2 + ESH - 2$$

LIMITE - Número de canais do espaço E+1 para os quais podem ser transferidos os segmentos dos canais em excesso do espaço E.

CANV - Variável temporária que indica o canal vertical para o qual o segmento S será transferido.

DESLOC - Número de canais do espaço E+1 ocupados por segmentos transferidos do espaço E.

IMPOSS - Matriz onde são colocadas as conexões que não foram traçadas. À cada linha corresponde uma conexão. A primeira coluna contém o número de CI ou conector (representado em excesso de 100) que contém o pino origem. A segunda coluna contém o número de pino origem terminal. A terceira e quarta colunas descrevem o pino terminal.

API - Apontador que contém o número de elementos de IMPOSS.

Descrição do algoritmo

PASSO 1 - Faça IV=0 para todos os elementos de MSV.

PASSO 2 - Inicialização

$$E=1, S=0, DESLOC=0$$

PASSO 3 - Se E=NEV então LIMITE=0

$$\text{senão } LIMITE = \text{inteiro}((ESH-1)/2)$$

- PASSO 4 - Tome o próximo segmento do espaço E
 $S=S+1$
- PASSO 5 - Se $S \leq APV(E)$ vá para o passo 12
- PASSO 6 - Se $IV(E,S)=1$ (os segmentos restantes a serem analisados foram transferidos do espaço E-1 e portanto já foram relocados) vá para o passo 12.
- PASSO 7 - $CANV=CANALV(E,S)+DESLOC$
- PASSO 8 - Se $CANV \leq NCEV$ (o segmento S não é transferido para o espaço E+1) então:
- $CANALV(E,S)=CANV$
 - $IV(E,S)=1$
 - Indique em $MSH(ESHS,ASHS)$ e $MSH(ESHI,ASHI)$ o novo valor de $CANALV(E,S)$ e reoriente esses segmentos se necessário.
 - vá para o passo 4
- PASSO 9 - $CANV > NCEV$ (o segmento S é transferido para o espaço E+1)
 $CANV=CANV-NCEV$
- PASSO 10 - Se $CANV > LIMITE$ (o segmento S iria ocupar em E+1 um canal além do limite permitido), a conexão é interrompida.
- Coloque a conexão definida por S, $MSH(ESHS(E,S),ASHS(E,S))$ e $MSH(ESHI(E,S),ASHI(E,S))$ em IMPOSS
 - Retire S, $MSH(ESHS(E,S),ASHS(E,S))$ e $MSH(ESHI(E,S),ASHI(E,S))$ de MSV e MSH
 - vá para o passo 4
- PASSO 11 - $CANV \leq LIMITE$
- $CANALV(E,S)=CANV$
 - $IV(E,S)=1$

- Transfira S de E para E+1

$APV(E+1) = APV(E+1) + 1$

$MSV(E+1, APV(E+1)) = MSV(E, S)$

- Indique em $MSH(ESHS(E, S), ASHS(E, S))$ e

$MSH(ESHI(E, S), ASHI(E, S))$ o novo valor de

$CANALV(E, S)$ e reoriente esses segmentos se necessário.

- Retire S de E

- vá para o passo 4

PASSO 12- $E = E + 1$

PASSO 13- Se $E > NEV$

então FIM

senão $-DESLOC = \text{máximo}(0, \text{mínimo}(\text{LIMITE}, DESLOC + NMOV(E) - NCEV)$

-vá para o passo 3.

2.4 - DESIGNAÇÃO DOS CANAIS HORIZONTAIS

O processo de designação dos canais horizontais é em tudo semelhante ao processo de designação dos canais verticais. A única diferença está no fato de que, estando os segmentos verticais completamente definidos em espaço e canal, as extremidades dos segmentos horizontais ficam determinadas apenas pelo espaço. Isso permite também. Isso permite então que se faça um ordenamento mais refinado dos segmentos horizontais e consequentemente se obtém uma maior densidade de segmentos por canal.

Definição das variáveis envolvidas no algoritmo

E - Espaço horizontal sendo analisado

FILA - Lista auxiliar que contém os segmentos do espaço sendo analisado.

APF - Apontador da FILA. Indica quantos elementos FILA contém.

S - Segmento retirado de FILA

LIMSUPE - Indica o espaço do segmento vertical que se conecta com a extremidade esquerda do último segmento horizontal colocado no canal sendo designado.

LIMSUPC - Indica o canal do segmento vertical que se conecta com a extremidade esquerda do último segmento horizontal colocado no canal sendo designado.

NMCH - Matriz coluna que possui tantas linhas quantos são os espaços horizontais. Cada elemento indica o número de canais utilizado no espaço horizontal respectivo.

Descrição do algoritmo

PASSO 1 - $E=1$

PASSO 2 - Se $APH(E)=0$ então vá para o passo 8

PASSO 3 - Coloque os segmentos de E em FILA

$APF=APH(E)$

$NMCH(E)=1$

PASSO 4 - $LIMSUPE=NEV+1$

$LIMSUPC=NCEV+1$

PASSO 5 - Se não existe em FILA um segmento S tal que $ESVD(E,S) > LIMSUPE$, ou $ESVD(E,S)=LIMSUPE$ e $CSVD(E,S) > LIMSUPC$, e $ESVD(E,S)$ e $LIMSUPC$ sejam máximos, então pegue novo canal $NMCH(E)=NMCH(E)+1$, e vá para o passo 4.

PASSO 6 - Retire S de FILA e coloque-o no canal $NMCH(E)$

$APF=APF-1$

$CANALH(E,S)=NMCH(E)$

PASSO 7 - Se $APF=0$ (FILA está vazia) vá para o passo 8

PASSO 8 - Se $E=NEH$ (todos os espaços foram analisados)

então FIM

senão tome próximo espaço, $E=E+1$, e vá para o passo 2

PASSO 9 - $LIMSUPE=ESVE(E,S)$

- $LIMSUPC=CSVE(E,S)$

- vá para o passo 5

2.5 - RELOCAÇÃO DOS SEGMENTOS HORIZONTAIS

Também a relocação dos segmentos horizontais é bastante semelhante à relocação dos segmentos verticais. Contudo, aqui o deslocamento de um segmento de um espaço para outro não implica em nenhuma reorientação de segmentos. Além disso foi acrescentada uma característica que permite algumas vezes uma economia de canais. Suponha-se que não foi possível relocar um segmento horizontal que faz parte de uma conexão do tipo 2. Sendo assim ele deve ser retirado de MSH e juntamente com ele, o 2º segmento horizontal da conexão. Suponha-se ainda que esse 2º segmento pertença a um espaço horizontal situado acima do espaço do 1º segmento horizontal e que esse 2º segmento ocupe sozinho um canal no seu espaço. Ao ser retirado, esse segmento provocaria um desperdício de 1 canal. Assim, para evitar esse fato, toda vez que isso acontecer, todos segmentos situados em canais acima daquele são rebaixados de 1 canal. Observe-se que esse fato só é relevante se o espaço do 2º segmento for superior ao do 1º segmento, pois os espaços inferiores já tiveram seus segmentos relocados.

Definição das variáveis envolvidas no algoritmo

E - Espaço horizontal sendo analisado

S - Um segmento horizontal do espaço E

SV- Segmento vertical que se conecta ao segmento horizontal S

- S2 - Segmento horizontal que constitui com S e SV uma conexão do tipo 2.
- E2 - Espaço de E2
- ESV - Espaçamento vertical entre CIs. É dado por um número inteiro de vigésimos de polegadas.
- NCEH - Número de canais por espaço horizontal $NCEH = (ESV + 4) / 2$
- LIMITE - Número de canais do espaço E+1 para os quais podem ser transferidos os segmentos dos canais em excesso do espaço E.
- CANH - Variável temporária que indica o canal horizontal para o qual o segmento S será transferido.
- DESLOC - Número de canais do espaço E+1 ocupados por segmentos transferidos do espaço E.

Descrição do algoritmo

- PASSO 1 - Faça $IH = 0$ para todos os elementos de MSH
- PASSO 2 - Inicialização
 $E = 1, S = 0, DESLOC = 0$
- PASSO 3 - Se $E = NEH$
 então LIMITE = 0
 senão para E par LIMITE = 3
 para E ímpar LIMITE = $ESV / 2$
- PASSO 4 - Tome o próximo segmento do espaço E
 $S = S + 1$
- PASSO 5 - Se $S > APH(E)$ vá para o passo 12
- PASSO 6 - Se $IH(E, S) = 1$ (os segmentos restantes a serem analisados foram transferidos do espaço E-1 e portanto já foram relocados) vá para o passo 12.
- PASSO 7 - $CANH = CANALH(E, S) + DESLOC$.

PASSO 8 - Se $CANH \leq NCEH$ (o segmento S não é transferido para o espaço E+1) então:

- $CANALH(E, S) = CANH$

- $IH(E, S) = 1$

- Se S fizer parte de uma conexão do tipo 2, indique em SV o novo valor de $CANALH(E, S)$

- vá para o passo 4

PASSO 9 - $CANH > NCEH$ (O segmento S é transferido para o espaço E+1)

$CANH = CANH - NCEH$

PASSO 10 - Se $CANH > LIMITE$ (o segmento S iria ocupar em E+1 um canal além do limite permitido), a conexão é impossível:

- Coloque a conexão definida por S e eventualmente SV e S2 em IMPOSS

- Retire S e eventualmente SV e S2 de MSH e MSV

- Se a conexão do qual S faz parte for do tipo 2, $E2 > E$ e S2 for o único segmento de $CANALH(E2, S2)$, então para todo segmento SX de E2 tal que

$CANALH(E2, SX) > CANALH(E2, S2)$

Faça $CANALH(E2, SX) = CANALH(E2, S2)$

- vá para o passo 4

PASSO 11 - $CANH \leq LIMITE$

- $CANALH(E, S) = CANH$

$IV(E, S) = 1$

- Transfira S de E para E+1

$APH(E+1) = APH(E+1) + 1$

$MSH(E+1, APH(E+1)) = MSH(E, S)$

- Se S faz parte de uma conexão do tipo 2 indique em S2 o novo valor de $CANALH(E, S)$

- Retire S de E
- vá para o passo 4

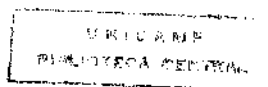
PASSO 12 - $E = E + 1$

PASSO 13 - Se $E > NEH$

então FIM

senão $-DESLOC = \text{máximo}(0, \text{mínimo}(\text{LIMITE}, DESLOC +$
 $NMCH(E) - NCEH))$

-vá para o passo 3



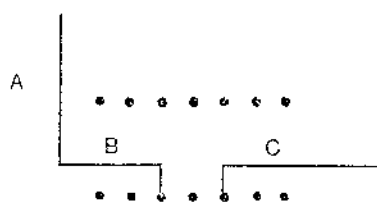


FIGURA 2.1

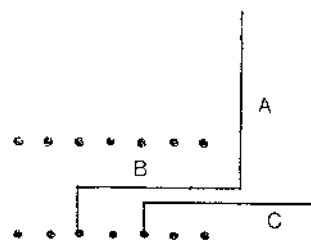


FIGURA 2.2

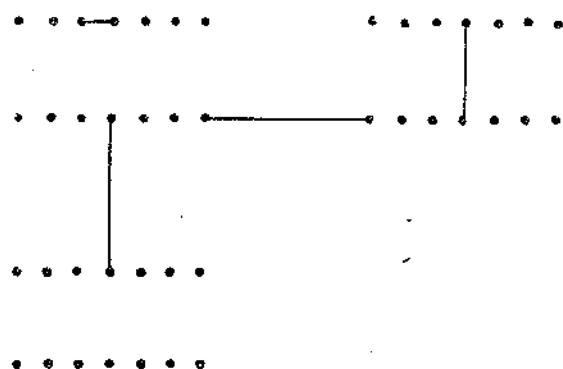


FIGURA 2.3 - Conexões diretas

3 - UTILIZAÇÃO DO SISTEMA

A implementação do algoritmo dos canais foi feita através de um programa em ALGOL para o PDP-10 do Centro de Processamento de Dados da UNICAMP. O programa consiste do bloco principal e de sete PROCEDURES. O bloco principal somente lê os parâmetros de entrada e chama os PROCEDURES. Estes executam o algoritmo e dão saída aos resultados. Abaixo temos uma descrição dos PROCEDURES:

- * ESPACO - implementa a designação dos espaços
- * SETCANALV - implementa a designação dos canais verticais
- * RELOCAV - implementa a relocação dos segmentos verticais
- * SETCANALH - implementa a designação dos canais horizontais
- * RELOCAH - implementa a relocação dos segmentos horizontais
- * IMAGEM - gera um arquivo de saída em disco contendo as matrizes MSH, MSV, CONDIR e IMPOSS
- * RESULT - gera vários arquivos de saída em disco contendo as coordenadas dos segmentos horizontais, dos segmentos verticais, dos furos metalizados, e os parâmetros geométricos do cartão.

O programa foi implementado na área [3001, 3001] em duas versões, TRACE1 e TRACE2. A única diferença entre eles é que no TRACE2 o PROCEDURE IMAGEM não foi implementado de modo que o arquivo de saída contendo as matrizes não é gerado.

3.1 - ENTRADAS

Todas as entradas para o programa são fornecidas em um arquivo ar-

mazenado em disco, construído pelo usuário. O arquivo se chama DADOS e é dividido em 3 partes:

* Parâmetros

* Descrição dos CIs

* Conexões

Os parâmetros são 12 números inteiros que são fornecidos na ordem abaixo:

- NCONEX - número de conexões a serem traçadas
- NCHIP - número de CIs do cartão
- NLINHAS - número de linhas de CIs no cartão
- NCOL - número de colunas de CIs no cartão
- NMP - número de pinos do CI que possui o maior número de pinos
- NSEG - número de segmentos previstos para cada espaço das matrizes MSH e MSV
- ESH - espaçamento horizontal entre CIs dado em vigésimos de polegada
- ESV - espaçamento vertical entre os CIs dado por um número par de vigésimos de polegada
- LC - comprimento dos terminais dos conectores dado em vigésimos de polegada
- X0 - distância em vigésimos de polegada entre a borda esquerda do cartão e o primeiro canal do primeiro espaço vertical
- Y0 - distância em vigésimos de polegada entre os terminais dos conectores e o primeiro canal do primeiro espaço horizontal
- YF - distância em vigésimos de polegada entre o último canal do último espaço horizontal e a borda superior do cartão.

Os parâmetros LC, X0, Y0 e YF estão indicados na figura 3.1 .

A descrição dos CIs consiste de NCHIP linhas de modo que cada linha corresponde a um CI. Cada linha contém 3 números que descrevem, na ordem, a linha em que o CI está localizado, a coluna em que o CI está localizado e o número de pinos do CI. Observe-se que supõe-se que os CIs estão numerados de 1 a NCHIP e de modo que a primeira linha corresponde ao CI 1, e segunda ao CI 2 e assim por diante.

Cada conexão é descrita por dois conjuntos de dados, onde cada conjunto descreve um pino de conexão. O conjunto é formado por 2 letras entre espas ("CI" ou "CO") que descrevem se o pino pertence a um CI ou a um conector, seguidas do número do CI ou conector e do número do pino. Esses dados são separados entre si por um ou mais espaços em branco.

As 2 páginas seguintes contêm a listagem do arquivo DADOS para o cartão W708 do PDP-10. A primeira linha contém os parâmetros, as oito seguintes a descrição dos CIs e as restantes as conexões. O único requisito é que os dados sejam fornecidos naquela ordem, porém o formato é livre.

56	8	6	2	14	128	10	6	10	7	4	6
3	1	14									
3	2	14									
4	1	14									
4	2	14									
5	1	14									
5	2	14									
6	1	14									
6	2	14									
"CI"	1	1	1	"CI"	1	12					
"CI"	1	2	2	"CI"	2	2					
"CI"	1	6	6	"CI"	1	8					
"CI"	1	8	8	"CI"	5	7					
"CI"	5	7	7	"CI"	5	1					
"CI"	1	10	10	"CI"	3	9					
"CI"	3	9	9	"CI"	7	12					
"CI"	7	12	12	"CI"	7	9					
"CI"	1	4	4	"CI"	1	11					
"CI"	1	1	1	"CI"	3	8					
"CI"	3	8	8	"CI"	3	10					
"CI"	3	10	10	"CI"	3	11					
"CI"	3	11	11	"CI"	3	12					
"CI"	3	12	12	"CI"	3	14					
"CI"	3	14	14	"CI"	3	13					
"CI"	3	3	3	"CI"	3	5					
"CI"	3	8	8	"CI"	5	11					
"CI"	5	11	11	"CI"	7	5					
"CI"	7	5	5	"CI"	7	8					
"CI"	7	8	8	"CI"	7	11					
"CI"	7	6	6	"CI"	3	13					
"CI"	3	4	4	"CI"	5	4					
"CI"	5	4	4	"CI"	7	4					
"CI"	5	5	5	"CI"	7	1					
"CI"	5	12	12	"CI"	4	3					
"CI"	6	3	3	"CI"	6	5					
"CI"	6	8	8	"CI"	7	10					
"CI"	6	9	9	"CI"	4	13					
"CI"	6	10	10	"CI"	6	8					
"CI"	6	11	11	"CI"	6	10					
"CI"	6	12	12	"CI"	6	11					
"CI"	6	13	13	"CI"	6	12					
"CI"	6	9	9	"CI"	6	5					
"CI"	8	5	5	"CI"	2	11					
"CI"	8	5	5	"CI"	6	9					
"CI"	7	4	4	"CI"	8	7					
"CI"	8	7	7	"CI"	8	9					
"CI"	8	9	9	"CI"	8	1					
"CI"	8	2	2	"CI"	6	12					
"CI"	8	3	3	"CI"	6	10					
"CI"	5	10	10	"CI"	6	3					
"CI"	8	4	4	"CI"	6	4					
"CI"	5	4	4	"CI"	4	12					
"CI"	4	12	12	"CI"	4	4					
"CI"	4	4	4	"CI"	2	4					
"CI"	2	14	14	"CI"	2	7					

\$

"CI"	6	14	"CI"	5	2
"CI"	6	2	"CI"	2	12
"CI"	6	1	"CI"	6	6
"CI"	4	2	"CI"	2	13
"CI"	6	13	"CI"	4	14
"CI"	4	14	"CI"	4	11
"CI"	1	9	"CO"	1	6

3.2 - SAÍDAS

As saídas do programa são colocadas em 5 arquivos diferentes que contém respectivamente, as matrizes do algoritmo, os parâmetros geométricos do cartão gerado, as coordenadas dos segmentos verticais, as coordenadas dos segmentos horizontais e as coordenadas dos furos metalizados.

3.2.1 - PARÂMETROS GEOMÉTRICOS DO CARTÃO

Esses parâmetros são gerados num arquivo em disco chamado CARD. A página seguinte contém a listagem desse arquivo para o cartão W708 especificado anteriormente. A primeira linha contém, nessa ordem, o comprimento horizontal do cartão, o comprimento vertical, o número de CIs, o número de conectores, o número máximo de pinos que possui um CI e o comprimento dos terminais dos conectores. As 8 linhas seguintes contém, em ordem, a coordenada horizontal do pino 1 do CI, a coordenada vertical do pino 1 do CI e o número de pinos do CI. Como antes, a linha 1 corresponde ao CI 1, a linha 2 ao CI 2, etc. Finalmente, as 2 últimas linhas contém as coordenadas horizontais dos terminais 1 dos conectores 1 e 2 nessa ordem.

57	97	8	2	14	10
11	47	14			
33	47	14			
11	59	14			
33	59	14			
11	71	14			
33	71	14			
11	83	14			
33	83	14			
11					
33					

3.2.2 - SEGMENTOS VERTICAIS

Os segmentos verticais são colocados num arquivo em disco chamado VERT. As duas páginas seguintes contêm a listagem desse arquivo para o cartão W708 especificado anteriormente. Cada linha corresponde a um segmento e contém, nessa ordem, a coordenada horizontal, do segmento, a coordenada vertical de uma extremidade do segmento e a coordenada vertical da outra extremidade.

21	44	49
11	48	47
45	45	47
24	44	47
11	47	46
41	46	47
39	45	47
45	44	47
13	52	53
23	56	53
23	51	53
49	54	53
36	50	53
33	52	53
37	51	53
21	52	53
13	56	59
15	59	57
19	57	58
21	56	59
17	58	59
37	58	59
37	60	59
29	56	59
39	57	59
21	66	65
39	62	63
49	65	63
23	63	69
23	64	63
15	63	63
15	65	64
15	65	63
37	67	65
33	65	62
33	65	66
39	66	65
27	69	71
17	71	73
17	71	73
47	73	71
41	73	71
37	71	70
35	71	69
25	70	71
33	71	72
42	72	71
41	65	68
15	68	65
37	68	65
17	79	77
13	77	76
21	77	79
45	79	77
33	77	78
25	77	76
17	74	73
39	74	73

16	75	74
15	83	84
45	84	82
17	86	77
39	77	82
43	82	77
35	77	88
39	88	77
37	82	77
41	77	88
23	58	89
17	89	94
32	91	89
11	89	94
15	91	89
33	89	91
41	91	89
43	98	89
19	85	83
19	87	83
17	88	83
33	87	83
35	86	83
36	86	83
8	58	48
7	56	45
16	68	44
9	69	51
22	66	54
24	62	46
8	59	54
12	66	58
8	98	87
16	63	56
18	73	58
14	98	74
16	52	14
26	75	58
17	73	56
16	79	67
29	58	46
38	85	87
32	98	82
44	88	78
36	86	74
31	75	67
34	68	56
31	57	45
38	52	44
46	78	68
32	78	51
48	76	62
17	47	53
45	83	89
35	59	53

3.2.3 - SEGMENTOS HORIZONTAIS

Os segmentos horizontais são colocados num arquivo em disco chamado HORIZ. As duas páginas seguintes contêm a listagem desse arquivo para o cartão W708 especificado anteriormente. Cada linha corresponde a um segmento e contém, nessa ordem, a coordenada horizontal de uma extremidade do segmento, a coordenada vertical do segmento e a coordenada horizontal da outra extremidade.

\$

16	14	21
8	48	11
7	45	15
10	44	21
11	46	14
29	46	41
31	45	39
30	44	45
8	52	13
10	50	23
9	51	23
12	54	19
29	50	39
30	52	33
32	51	37
16	52	21
7	56	13
15	57	19
16	56	21
10	58	17
30	58	37
29	60	37
34	56	39
31	57	39
12	66	21
14	62	23
19	63	23
8	64	23
13	63	16
25	64	28
35	63	40
31	67	37
33	62	40
33	66	39
9	69	23
11	70	23
10	73	17
29	73	41
37	76	46
1	68	46
30	70	35
33	72	43
11	68	15
34	68	37
8	79	17
13	76	30
21	79	38
40	79	45
33	78	46
35	76	48
14	74	17
36	74	39
31	75	39
15	84	45
12	80	17
38	82	43
35	80	39
32	82	37

4

41	80	44
8	90	23
17	91	23
11	91	15
33	91	41
30	90	43
12	86	19
8	87	19
14	88	17
30	87	33
32	86	35
36	86	39
19	65	17
17	65	15
11	65	13

3.2.4 - FUROS METALIZADOS

Os furos metalizados são difinidos num arquivo em disco chamado FUROS. As três páginas seguintes contêm a listagem desse arquivo para o cartão W708 especificado anteriormente. Cada linha corresponde a um furo e contém, nessa ordem, a coordenada horizontal e a coordenada vertical do furo.

\$

16	14
21	14
8	40
11	40
7	45
15	45
10	44
21	44
11	46
14	46
29	46
41	46
31	45
39	45
30	44
45	44
8	52
13	52
10	50
23	50
9	51
23	51
12	54
19	54
29	50
39	50
30	52
33	52
32	51
37	51
16	52
21	52
7	56
13	56
15	57
19	57
16	56
21	56
10	58
17	58
30	58
37	58
29	60
37	60
34	56
39	56
31	57
39	57
12	66
21	66
14	62
23	62
19	63
23	63
8	64
23	64
13	63
16	63

35	64
38	64
35	63
40	63
31	67
37	67
33	62
48	62
33	66
39	66
9	69
23	69
11	70
23	70
10	73
17	73
29	73
41	73
37	70
44	70
35	69
46	69
32	70
35	70
33	72
43	72
11	68
15	68
34	68
37	68
8	79
17	79
13	76
30	76
21	79
38	79
46	79
45	79
11	78
10	78
35	76
48	76
14	74
17	74
36	74
39	74
31	75
39	75
15	84
45	84
12	80
17	80
39	82
43	82
35	80
39	80
32	82
37	82

4

41	80
44	80
8	90
23	90
17	91
23	91
11	91
15	91
33	91
41	91
30	90
43	90
12	86
19	86
8	87
19	87
14	88
17	88
30	87
33	87
32	86
35	86
36	86
39	86

3.2.5 - MATRIZES DO ALGORITMO

As matrizes do algoritmo, MSH, MSV, IMPOSS e CONDIR, estão definidas num arquivo em disco chamado MATRIX que contém as imagens dessas matrizes definidas no capítulo 2. As páginas seguintes contêm uma listagem desse arquivo para o cartão W708 especificado anteriormente.

\$

MATRIZ DE SEGMENTOS HORIZONTAIS

ESPACO 1

1	1	1	13	7	101	511	6	1
---	---	---	----	---	-----	-----	---	---

ESPACO 2

ESPACO 3

ESPACO 4

ESPACO 5

ESPACO 6

4	1	1	1	2	1	511	1	1
2	1	1	2	1	1	511	3	2
1	1	1	3	4	1	511	6	3

\$

3	1	1	511	1	1	6	6	4
3	1	2	5	1	2	511	5	5
2	1	2	12	3	2	511	4	6
1	1	2	13	2	2	511	7	7

ESPRADO 7

3	1	1	1	2	1	511	13	1
1	1	1	3	4	1	511	8	2
2	1	1	4	3	1	511	8	3
4	1	1	5	5	1	511	10	4
1	1	2	5	1	2	511	11	5
3	1	2	13	2	2	511	14	6
2	1	2	15	4	2	511	12	7
3	1	1	13	7	1	511	9	8

ESPRADO 8

1	1	1	2	1	3	511	2	1
2	1	3	511	3	3	511	5	2
1	1	1	10	7	3	511	6	3
3	1	1	11	4	3	511	4	4
3	1	2	1	2	4	511	3	5
4	1	2	2	1	4	511	3	6
1	1	2	11	5	4	511	4	7
2	1	2	12	3	4	511	4	8

ESPRADO 9

4	1	1	5	5	3	511	9	1
1	1	1	6	6	3	511	8	2
2	1	3	511	10	3	511	8	3
3	1	1	7	2	3	511	8	4
2	1	3	511	13	1	10	7	5
3	1	4	511	13	2	3	7	6
2	1	4	511	13	2	4	8	7
5	1	2	10	3	4	511	12	8
1	1	4	511	14	2	16	12	9
4	1	4	511	14	4	511	11	10

\$

ESPACO 10

2	1	1	4	3	5	511	7	1
3	1	5	511	1	5	511	7	2
5	1	1	11	4	5	511	4	3
5	1	2	2	1	6	511	5	4
3	1	6	511	3	2	8	10	5
2	1	6	511	2	2	14	11	6
3	1	2	15	4	6	511	2	7
4	1	6	511	1	6	511	6	8
1	1	3	511	14	3	511	12	9
1	1	2	11	5	4	511	12	10

ESPACO 11

5	1	1	7	2	5	511	11	1
3	1	5	511	13	2	1	2	2
5	1	5	511	9	2	3	7	3
5	1	2	4	8	6	511	8	4
4	1	6	511	14	2	14	11	5
3	1	6	511	13	2	16	12	6
1	1	1	12	6	5	511	4	7
1	1	2	9	6	6	511	4	8
2	1	2	10	3	6	511	4	9

ESPACO 12

4	1	7	511	3	8	511	7	1
1	1	1	8	9	5	511	11	2
3	1	6	511	11	6	511	9	3
1	1	6	511	13	6	511	11	4
3	1	2	7	4	6	511	12	5
1	1	6	511	10	2	8	10	6

ESPACO 13

4	1	1	9	2	7	511	8	1
5	1	7	511	11	7	511	8	2
5	1	7	511	14	7	511	12	3
5	1	8	511	14	8	511	10	4
4	1	2	7	2	8	511	9	5

#

1	1	1	8	5	7	511	5
2	1	1	9	2	7	511	5
3	1	1	12	6	7	511	4
2	1	2	6	2	8	511	1
1	1	2	7	4	8	511	2
1	1	2	9	6	8	511	4

6
7
8
9
10
11

MATRIZ DE SEGMENTOS VERTICAIS

ESPACO 1

2	1	7	1	3	6	1	4
1	1	8	1	1	6	2	2
4	1	7	2	1	6	3	1
3	1	10	1	2	7	3	2
5	1	9	1	4	7	4	4
6	1	9	2	1	6	4	3
2	1	11	1	5	9	4	3
5	1	13	6	1	12	2	1
2	1	13	7	4	13	12	2
7	1	9	5	2	8	3	1
4	1	10	3	5	8	4	3
6	1	13	8	3	11	7	1
7	1	7	8	3	1	1	1

1
2
3
4
5
6
7
8
9
10
11
12
13

ESPACO 2

2	1	11	2	3	8	5	3
1	1	10	4	5	8	6	4
7	1	11	3	5	9	6	3
8	1	11	4	5	9	7	2
1	1	7	5	1	6	5	3
2	1	13	9	8	13	14	2
4	1	13	10	4	12	5	3
10	1	12	6	1	10	5	3
6	1	13	11	1	11	8	1
3	1	11	9	2	9	8	5
5	1	10	10	1	8	7	1
3	1	8	8	2	6	6	2
2	1	7	6	3	6	7	1

1
2
3
4
5
6
7
8
9
10
11
12
13

#

11	1	11	5	4	10	6	2	
4	1	10	7	3	7	7	2	14
12	1	11	6	3	9	9	1	15
								16

NUMERO DE CONEXOES NAO CONCLUIDAS 9

6	9	2	5
6	9	8	5
6	10	8	3
7	1	5	5
7	13	3	9
7	13	7	9
7	10	5	8
7	12	7	6
8	11	6	9

NUMERO DE CONEXOES DIRETAS 6

1	4	1	11
3	10	3	11
3	11	3	12
3	14	3	13
8	7	8	8
7	7	7	10

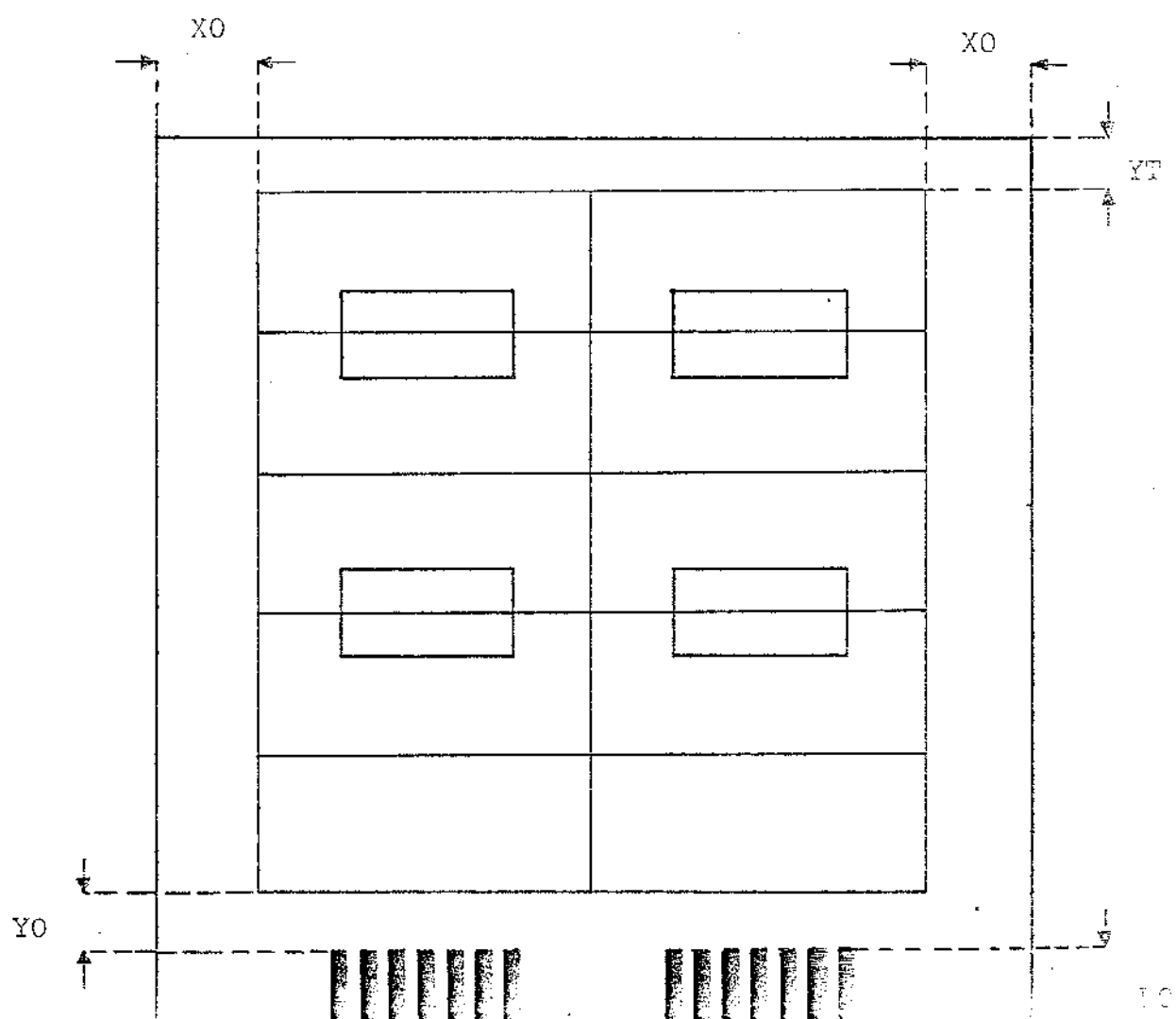


FIGURA 3.1 - Parâmetros geométricos do cartão

4. TESTE DE PERFORMANCE DO ALGORITMO

A performance do algoritmo foi medida segundo três parâmetros, quais sejam, número médio de falhas (conexões não traçadas), comprimento médio das conexões e tempo de execução. As medidas foram feitas gerando-se cartões com vários números de conexões e aplicando-se o algoritmo a esses cartões. O cartão tomado como padrão contém 9 CIs de 14 pinos distribuídos em 3 linhas e 3 colunas com espaçamento horizontal entre CIs de 10 vigésimos de polegada e espaçamento vertical de 8 vigésimos de polegada. Esses espaçamentos são os utilizados no cartão Philips 432202639882 que é um cartão pré fabricado para montagem com CIs de 14 pinos que são interligados entre si por fios e portanto permite uma densidade muito maior de conexões do que um cartão de dimensões idênticas para interligações com circuito impresso. Esse espaçamento foi tomado pequeno propositalmente de modo a saturar o cartão rapidamente sem ter-se que gerar um número muito grande de conexões. As conexões foram geradas aleatoriamente e de modo a simular conexões reais, isto é, o grafo das conexões foi gerado sem caminhos cíclicos, sendo impossível partir-se de um vértice (terminal) e voltar-se a ele próprio tomando-se uma sequência de arestas (conexões) interligadas.

4.1 - NÚMERO MÉDIO DE FALHAS

O número médio de falhas foi medido para cartões com 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380, 390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490, 500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 610, 620, 630, 640, 650, 660, 670, 680, 690, 700, 710, 720, 730, 740, 750, 760, 770, 780, 790, 800, 810, 820, 830, 840, 850, 860, 870, 880, 890, 900, 910, 920, 930, 940, 950, 960, 970, 980, 990, 1000 conexões. Para cada número de conexões foram geradas 1000 amostras. Na figura 4.1 vê-se as curvas do número médio de falhas e do número médio de conexões traçadas. Observe-se que esse último satura em torno de 55 conexões. Na prática o número de conexões presentes em um cartão varia em torno da metade do número total de pinos. No caso esse número é de 126 (9x14) o que dá 63 conexões. Assim é bastante boa a performance do algoritmo principalmente se levarmos em conta o pequeno espaçamento entre CIs adotado e também o fato que na prática as conexões estão bem longe de serem aleatórias já que normalmente os CIs são posicionados de modo a minimizar o comprimento das conexões. Como o comprimento dos segmentos é fator fundamental para a eficiência do algoritmo

é de se supor que para cartões reais a sua tendência é de melhorar. Na figura 4.2 vê-se que o desvio padrão do número médio de falhas tem sempre um valor bastante pequeno, o que seria de se esperar, já que sendo as conexões uniformemente distribuídas, o seu comprimento médio deve-se manter aproximadamente constante, não devendo a configuração das conexões influenciar o número de falhas.

4.2 - COMPRIMENTO MÉDIO DAS CONEXÕES

O comprimento médio das conexões foi também medido tomando-se 100 amostras de cartões para cada número de conexões. Como era de se esperar (ver figura 4.3) o comprimento médio das conexões se mantém bastante constante já que elas são distribuídas uniformemente. Contudo é interessante notar que mesmo para um número médio de conexões, que é onde o processo de relocação de segmentos é mais aplicado, o comprimento médio das conexões se manteve constante.

4.3 - TEMPO DE EXECUÇÃO

Devido a dificuldades práticas de se medir o tempo de execução do algoritmo tomou-se somente 1 amostra para cada número de conexões. Porém isso não tem muita importância pois a disposição das conexões não deve influir no tempo de execução, mas sim o número delas. Além do mais o que interessa no caso é a ordem de grandeza e não o valor absoluto. Pela figura 4.4 pode-se facilmente concluir que o tempo de execução é sem dúvida a principal característica do algoritmo dos canais. Vê-se que ele varia linearmente com o número de conexões com uma derivada de aproximadamente 70 milissegundos por conexão. Essa velocidade por si só justifica a utilização do algoritmo dos canais mesmo nos casos em que ele possua limitações, pois tempo no caso significa custo.

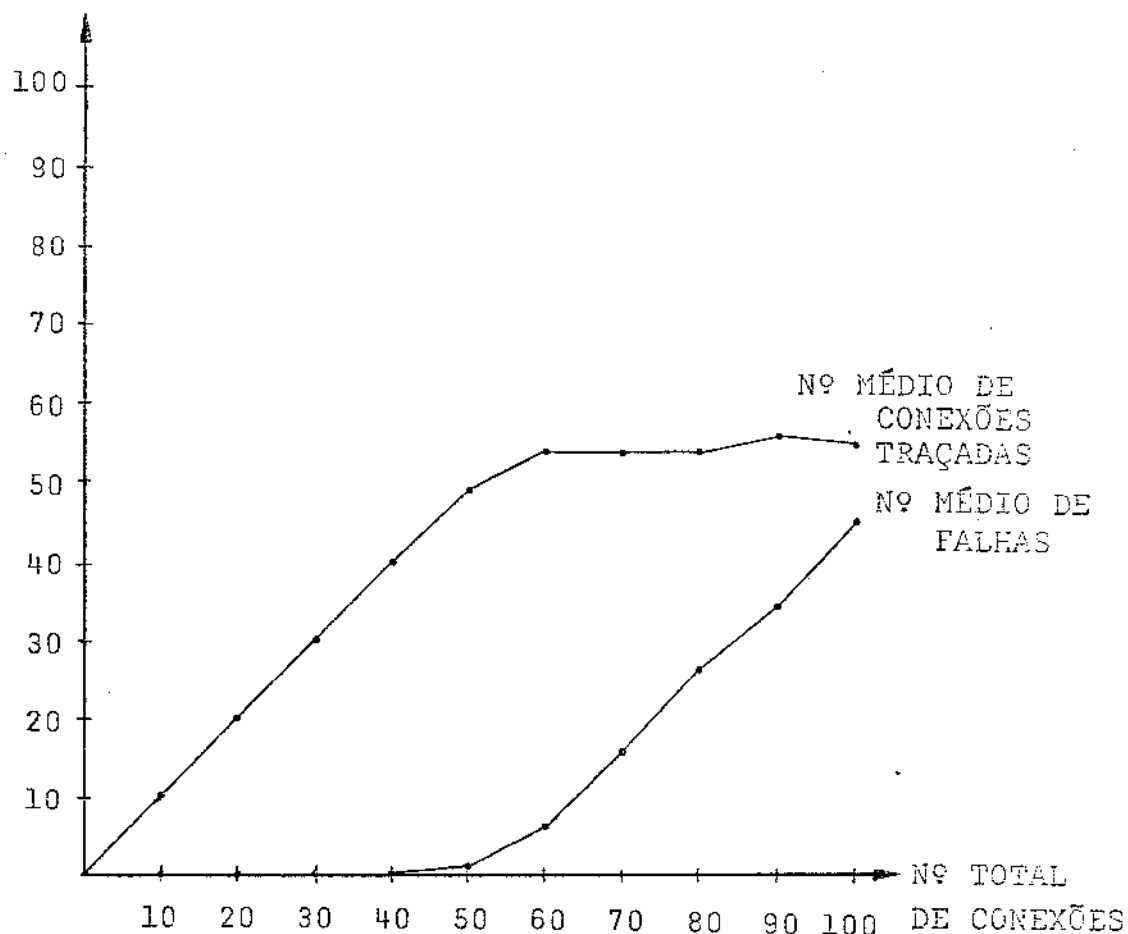


FIGURA 4.1

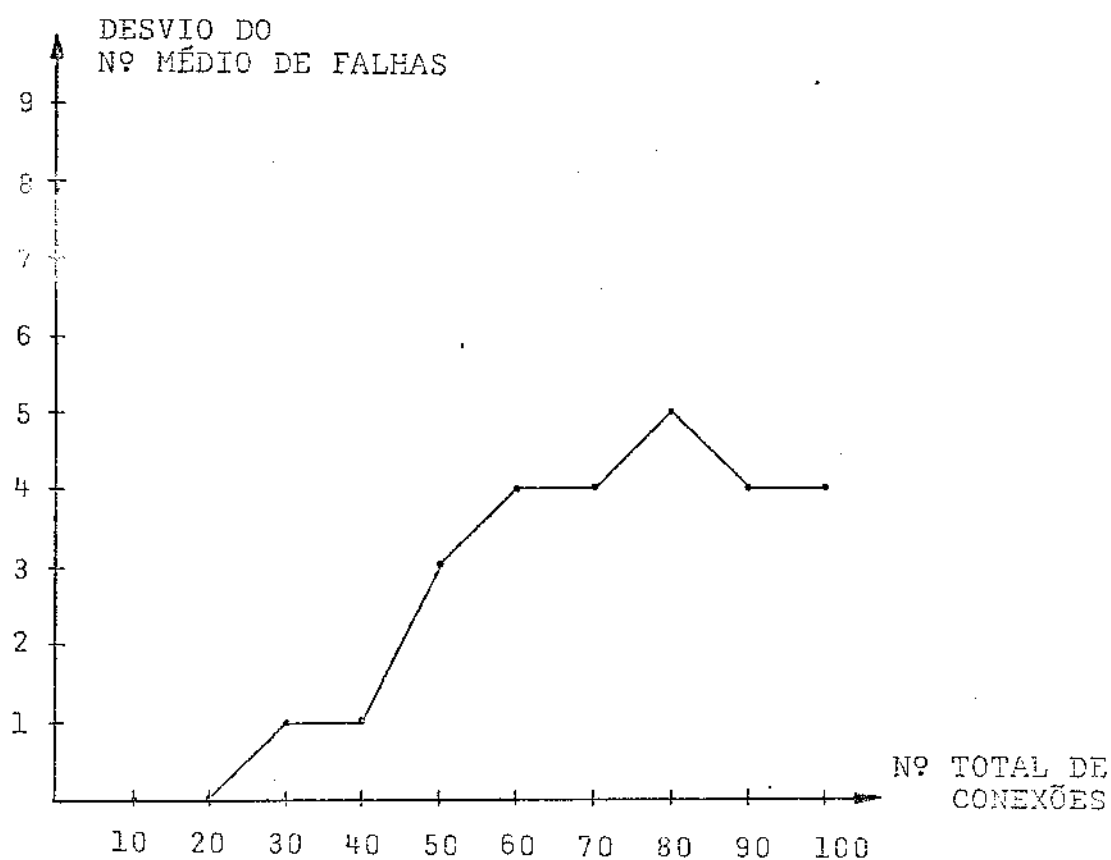


FIGURA 4.2

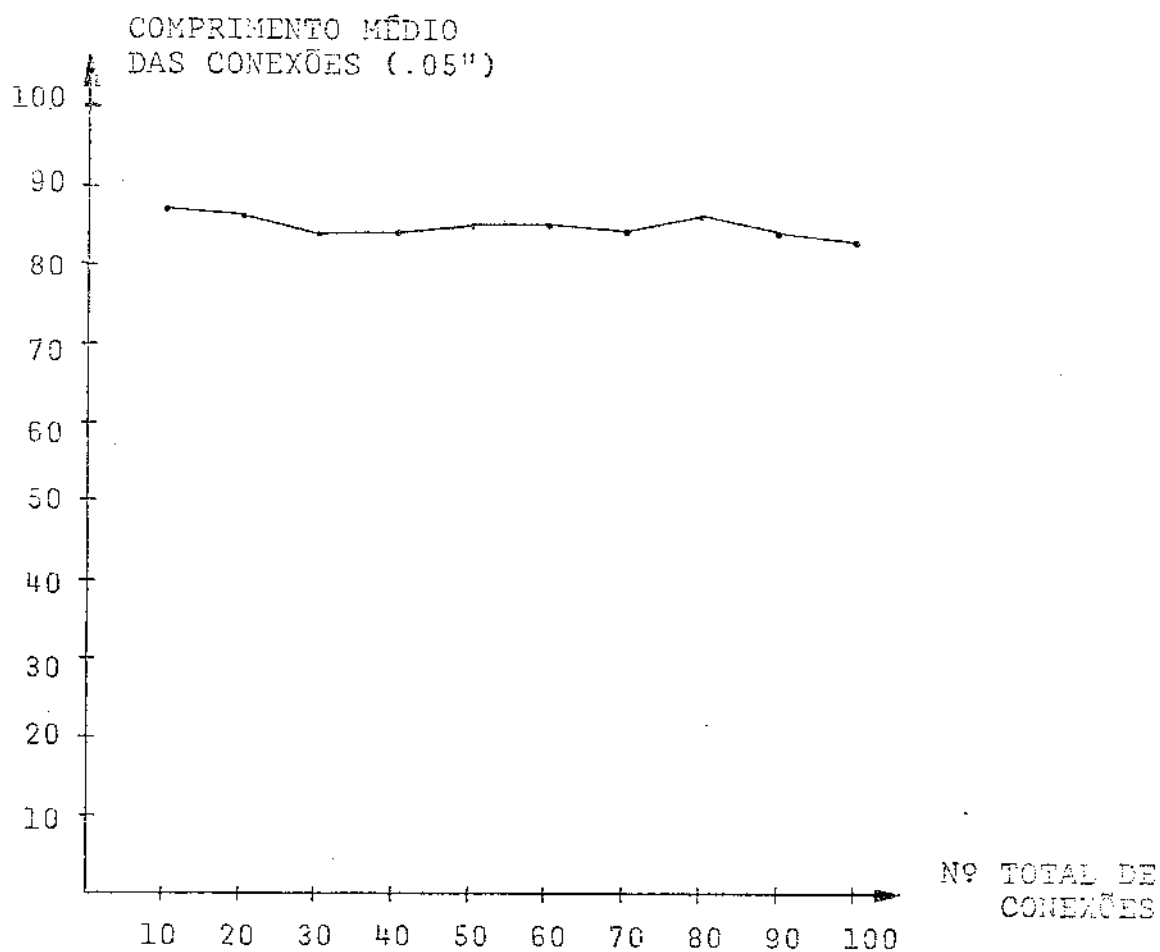


FIGURA 4.3

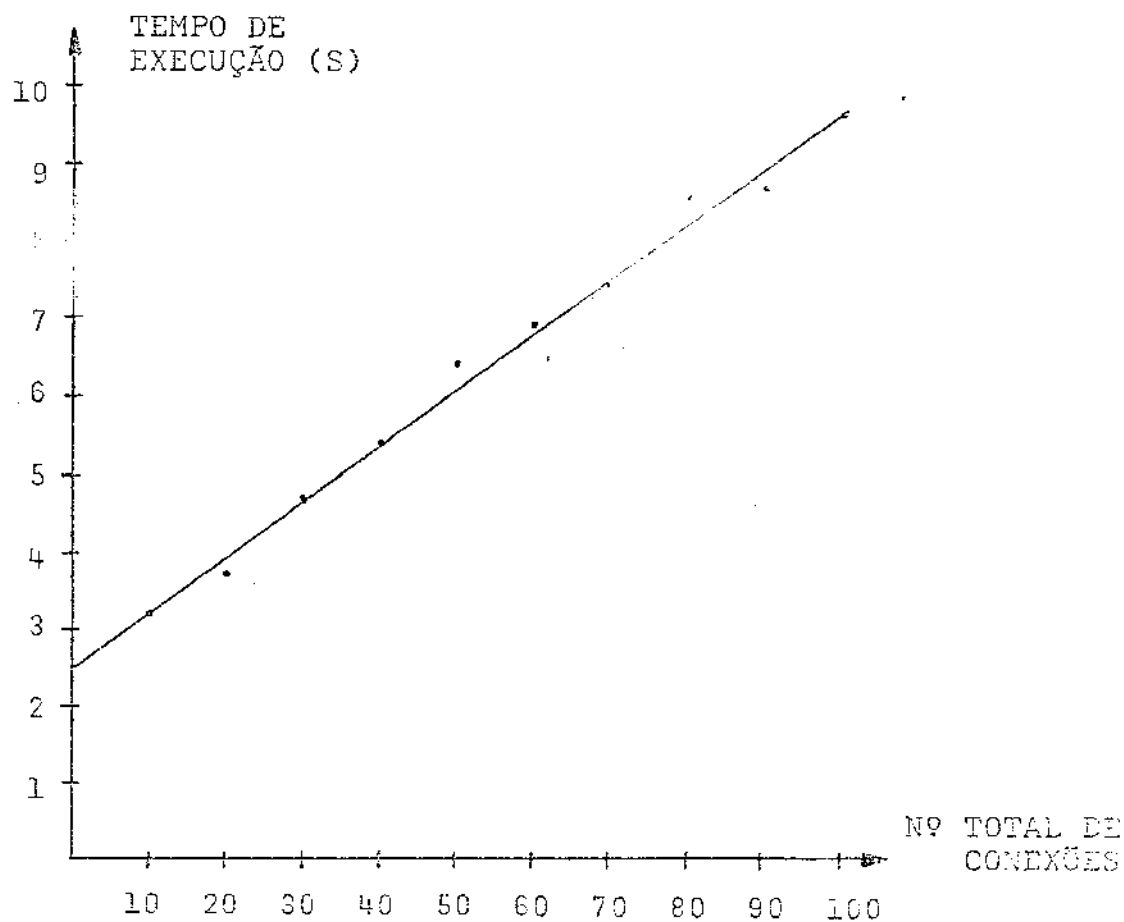


FIGURA 4.4

5 - CONCLUSÕES

Da análise do resultados dos testes realizados com o algoritmo dos canais depreende-se que as suas características de alta velocidade e baixo requisito de memória justificam e aconselham a sua utilização como núcleo de um sistema automático para traçado de circuitos impressos. Porém, ele possui algumas limitações práticas que devem ser levados em consideração no projeto de um desses sistemas e que enumeramos abaixo

- (a) Devido à rigidez com que são definidas as conexões algumas delas não são realizadas devido não à falta de espaço físico no cartão mas sim à forma geométrica predeterminada das conexões.
- (b) Devido ainda a forma predeterminada das conexões, muitas delas tomam um comprimento excessivo e se quebram num número de segmentos que talvez pudesse ser evitado. Lembre-se que muitos segmentos significam muitos furos metalizados.
- (c) O algoritmo se aplica somente a circuitos integrados do tipo "dual-in-line". Na prática todos os cartões de circuito impresso possuem pelo menos alguns componentes discretos.
- (d) O algoritmo realiza somente conexões ponto-a-ponto. Muitas vezes, conexões ponto-a-reta são necessárias, como por exemplo as conexões dos pinos de terra e Vcc dos CIs aos respectivos pontos de terra e Vcc respectivamente, que normalmente circuncosam todo o cartão de circuito impresso.
- (e) O algoritmo designa um conector para cada coluna de CIs e de modo que cada conector possui tantos terminais quantos são os pinos do CI de maior número de pinos. O número de terminais que assim se obtém pode muitas vezes não ser suficiente. Além do mais são utilizados os terminais de um dos lados dos conectores.

Para o item (a) podem ser realizadas algumas modificações no algoritmo de modo a tentar diminuir o número de falhas. Assim, por exemplo, poder-se-ia fazer a relocação dos segmentos nos dois sentidos, isto é, permitir que segmentos do espaço E fossem transferi-

dos para o espaço $E-1$ bem como para o $E+1$. Uma outra hipótese sugerida pelo autor do algoritmo é permitir que os segmentos horizontais de um espaço possam ser transferidos para o outro espaço não somente até a linha de pinos mas mais além desde que isso não provoque conflitos. Isso equivale a estabelecer uma linha quebrada - como limite entre dois espaços. No entanto acreditamos que as melhorias obtidas não seriam dramáticas. Para os demais itens seria muito difícil, senão impossível, obter-se soluções sem modificar profundamente o algoritmo. Assim nos parece que muito mais eficiente que alterar o algoritmo seria estabelecer um sistema de 2 fases. Na primeira fase seria executado o algoritmo dos canais que traçaria a maioria das conexões. Na segunda fase seria executado um algoritmo baseado no procedimento de Lee, provavelmente o de Geyer, que realizaria as conexões restantes. Observe-se que nessa segunda fase todos aqueles problemas citados anteriormente seriam solucionados pois esse algoritmo encontra sempre o caminho mais curto entre 2 pontos, pode realizar conexões ponto-a-reta e não pressupõe nenhuma estrutura para os pontos a serem conectados. Como a segunda fase se aplicaria poucas vezes, o grave problema do procedimento de Lee que é o tempo de execução seria minimizado. Contudo se se pensar num sistema completamente automático, a sua eficiência poderia cair bastante devido à necessidade de interligação entre as duas fases, principalmente no caso de se querer refazer conexões que estivessem com um comprimento excessivo. Assim nos parece que a melhor solução seria um sistema semi-automático em que o usuário seria auxiliado para a conexão dos pontos. Através de um terminal gráfico interativo o usuário poderia examinar o "lay-out" produzido pelo algoritmo dos canais e através da "light-pen" indicaria as conexões a serem realizadas ou refeitas pelo algoritmo baseado no procedimento de Lee. Assim acreditamos que seria obtido um sistema com máxima rapidez e eficiência.

BIBLIOGRAFIA

- 1 - I. Aramaki, T. Kawabata, K. Arimoto, "Automation of Etching-Pattern Lay-Out", Communications of the ACM, Novembro de 1971, volume 14, número 11, páginas 720-730.
- 2 - G.C. Evans, M.W. Gribble, "Automatic Interconnection System for Electronic Components", Proceedings IEEE, Dezembro de 1969, páginas 1992-1998.
- 3 - C.Y. Lee, "An Algorithm for Path Connections and Its Applications", IRE Transactions on Electronic Computers, Setembro de 1961, volume EC-10, páginas 346-365.
- 4 - K. Shostack, "Computerized Methods for The Routing of Printed Circuit Boards", Computer Design, Julho de 1970, volume 9, número 7, páginas 80-87.
- 5 - S.B. Akers, "A Modification of Lee's Path Connection Algorithm", IEEE Transactions on Electronic Computers, Fevereiro de 1967, volume EC-16, páginas 97-98.
- 6 - S. Heiss, "A Path Connection Algorithm for Multi-Layer Boards", Proceedings Share/ACM, IEEE Design Automation Workshop, 1968, páginas 6.1-6.11.
- 7 - J.M. Geyer, "Connection Routing Algorithm for Printed Circuit Boards", IEEE Transaction on Circuit Theory, Janeiro de 1971, volume CT.18, número 1, páginas 95-100.
- 8 - D.W. Hightower, "A Solution to Line-Routing Problems on The Continuous Plane", Proceedings Design Automation Workshop, 1969, páginas 1-24.
- 9 - R.B. Hitchcock, "Cellular Wiring and The Cellular Modeling Technique", Proceeding Design Automation Workshop, 1969, páginas 25-41.

- 10 - J.F. Cooper, J.A. Brown, "Automated Partitioning and Placement in The WRAP System", Proceedings Technical Program of National Electronics Packaging and Production Conference, Janeiro de 1968, páginas 85-93.
- 11 - C.J. Fisk, D.L. Caskey, L.I. West, "ACCEL : Automated Circuit Card Etching Layout", Proceedings IEEE, Novembro de 1967, páginas 1971-1982
- 12 - S.E. Lass, "Automated Printed Circuit Routing with A Stepping Aperture", Communications of the ACM, maio de 1969, volume 12, número 5, páginas 262-265.
- 13 - J.E. Stevens, Jr., "Fast Heuristic Techniques for Placing and Wiring Printed Circuit Boards", CAC Document n958, University of Illinois at Urbana - Champaign, Outubro de 1972, páginas 53-76.