# AUTOMAÇÃO DE PROJETOS DE SISTEMAS DIGITAIS

## - ARQUIVO CENTRAL -

ANTONIO MARCOS DE AGUIRRA MASSOLA

Tese de Livre-Docência apresentada

à

ESCOLA POLITÉCNICA DA UNIVERSIDADE

DE SÃO PAULO

BIBLIOTECA
Dept. Eng. ELETRICIDADE
E.P.U.S.P

- JULHO 1976 -

# A G R A D E C I M E N T O S

Ao Professor Doutor Antonio Hélio Guerra Vieira pelo apoio e constante incentivo.

Aos colegas do Laboratório de Sistemas Digitais, pelo apoio e colaboração que sempre me ofereceram.

Ao engenheirando Egmont Y. Shimizu, pelo auxílio na confecção deste trabalho.

As secretárias Maria Inês Mello e Sonia Regina Izarelli, pelos excelentes serviços de datilografia e pela dedicação com que se houveram.

Ao Nelson A. Kobayashi, pelos serviços de desenho.

À todos que, direta ou indiretamente, tenham cooperado para a concretização deste trabalho.

# R E S U M O

Este trabalho apresenta os principais aspectos da Automação de Projetos de Sistemas Digitais com a utilização de computadores.

Apresenta uma visão geral do problema de Automação, suas diferentes fases e como estas se relacionam.

Descreve-se a estrutura e organização de um Arquivo Central de dados e sua importância dentro de um Sistema Integrado de Automação.

Finalmente apresenta-se um esboço de uma linguagem geral a ser utilizada pelas diferentes fases do Sistema.

# S U M M A R Y

The author at this paper deals with some basic of computer-aided automation features on digital systems design.

It is related a general view on automation problem, their different phasis and their relationships, as well.

It is described the main structure of a master-file organization and also includes some explanations in respect of its importance for a computer-aided design for digital Systems.

The paper also includes an general language for the System.

Aos meus pais

# ÍNDICE

CAPÍTULO I

I.1 A AUTOMAÇÃO ..... 01

CAPÍTULO II

II.1 O SISTEMA INTEGRADO DE AUTOMAÇÃO ..... 10

II.2 OPERAÇÃO DO SISTEMA ..... 11

II.3 O CONTROLE DO SISTEMA ..... 19

CAPÍTULO III (Informações necessárias a cada fase)

III.1 BIBLIOTECA ..... 22

III.2 DESCRIÇÃO LÓGICA ..... 33

III.3 SIMULAÇÃO ..... 35

III.4 PARTIÇÃO ..... 43

III.5 POSICIONAMENTO ..... 47

III.6 FIAÇÃO ..... 48

III.7 ROTAS ..... 48

III.8 DOCUMENTAÇÃO ..... 50

CAPÍTULO IV

IV.1 OS COMANDOS DO SUPERVISOR ..... 51

CAPÍTULO V

V.1 OBSERVAÇÕES FINAIS ..... 102

BIBLIOGRAFIA ..... 106

# CAPÍTULO I

REFERÊNCIAS: [2] [5] [8] [14] [15]

## I.1 A AUTOMAÇÃO

O termo "Automação" é comumente usado para designar a utilização de computadores e técnicas computacionais no auxílio de um projeto e no desenvolvimento de sistemas ou de sistemas.

A técnica de automação vem sendo desenvolvida desde a década de 60, porém, nos anos recentes é que adquiriu uma maior importância, decorrente do nível de sofisticação dos projetos, graças ao desenvolvimento tecnológico sempre crescente.

O aparecimento de dispositivos periféricos, especificamente destinados a facilitar a obtenção de resultados gráficos, tais como: "plotters" planares dotados de recursos fotográficos de precisão; "displays" que permitem a interação do projetista com o computador, possibilitaram a utilização de técnicas de automação em áreas onde a participação pessoal e muitas vezes artesanal do projetista tinha aspecto predominante.

A área de microeletrônica, onde o aparecimento dos circuitos em "LSI" vieram revolucionar o campo dos circuitos integrados e a área de sistemas digitais, onde projetos cada vez mais sofisticados são possíveis graças ao aparecimento de novos dispositivos oriundos de um acelerado desenvolvimento tecnológico, são exemplos de campos de aplicação das técnicas de automação de projetos.

Sistemas específicos de automação têm sido desenvolvidos por indústrias produtoras de "LSI" e de sistemas digitais, apesar de seu custo de desenvolvimento, implantação e atualização ser relativamente alto. Visam, entretanto, dar maior segurança, rapidez, padronização e baixo custo por projeto, características indispensáveis no atual estágio da indústria eletrônica.

Neste ponto uma pergunta se põe: por que não adquirir um sistema de automação de projetos já pronto e testado em vez de desenvolver um novo, suscetível a erros e demora de implantação ?

A resposta a essa pergunta pode ser fornecida segundo os seguintes aspectos:

a) muitas vezes um sistema de automação é suficientemente genérico, abrangendo várias áreas de aplicação, a ponto de necessitar, para a sua correta utilização, do treinamento de pessoal técnico, fato esse que faz com que o projetista fique dependente de determinadas pessoas toda vez que necessite utilizar o sistema integrado de automação;

b) os recursos previstos para utilização nos programas que compõem o sistema, ficam condicionados aos recursos tecnológicos existentes no local onde o sistema foi desenvolvido, tornando-se necessário adaptá-lo aos recursos disponíveis onde se está utilizando o sistema;

c) muitos sistemas de automação, embora específicos para determinadas áreas de atividades, por exemplo, sistemas digitais, possuem certas características que não se adaptam às dos projetistas locais, necessitando assim ou da adaptação dos projetistas aos algorítmos utilizados ou da modificação do sistema para adaptá-lo às novas características;

d) o sistema de automação comprado fora muitas vezes (e na maioria delas) não é compatível com os sistemas de computação que se dispõe e isso impõe a necessidade de adaptar os programas que compõem o sistema para permitir o processamento no sistema de computação disponível.

A análise de cada item citado mostra que é bem mais adequado o desenvolvimento de um sistema de automação local que possua os requisitos necessários para atender as diversas fases de um projeto, tendo o seu desenvolvimento compatibilizado com o siste

ma de computação que se dispuser. É claro que esse será um condicionante bastante significativo, dependendo da profundidade do sistema de automação que se deseja implementar.

Aceitando-se a idéia de que deve-se partir para o projeto de um sistema de automação local, passa-se agora à visualização das principais fases que devem compor esse sistema quando o mesmo se destina a cobrir projetos na área de sistemas digitais.

Em linhas gerais, as fases que devem ser vencidas pelo projetista podem ser definidas como sendo as seguintes, para um projeto sem auxílio do computador:

1. Com os dados disponíveis, elaborar o projeto utilizando grandes blocos funcionais.

2. Após definida a constituição em blocos, elaborar, para cada porção significativa do projeto, uma descrição a nível de circuitos, que devem realizar em conjunto as funções desejadas. O sistema nesta parte do projeto já está definido e o projetista poderia então analisar o comportamento do circuito se dispusesse das tecnicas de simulação. Caso contrário, e não dispondo de outros recursos, é obrigado a aguardar a montagem do projeto, em laboratório, para avaliar seu desempenho.

3. Com essas etapas completas - conhecidas como projeto lógico - ele deve estabelecer a divisão dos elementos componentes do circuito em um certo número de placas (de tamanho padronizado), verificando sempre a distribuição dos sinais, para evitar a duplicidade de funções em diferentes placas. No entanto, há casos em que a duplicação de funções é necessária para evitar o uso excessivo de pinos de conectores. Essa divisão é denominada partição.

4. Nesta etapa serão alocados os componentes que fazem parte do circuito. O projetista deve verificar cuidadosamente a posição desses componentes, de forma a facilitar as interligações.

As ligações tipicamente experimentais, utilizando fio enrolado a soquetes - método de conexão conhecido como "wire-wrap" - constituem uma opção para o projetista. Esse tipo de ligação é bastante simples e permite modificar facilmente as ligações com erro. O projetista poderá usar também ligações sob a forma de rotas de circuito impresso. Porém, com esta técnica, é importante posicionar adequadamente os componentes na placa para facilitar o estabelecimento dos caminhos elétricos. Este tipo de interligação, no entanto, dificulta as correções.

5. O sistema, nesta etapa, está pronto para ser testado em laboratório. Cada placa será testada e corrigida em separado e, a seguir, testada em conjunto.

6. Concluída a fase de testes, o projetista deve reunir toda a documentação disponível sobre o projeto, efetuar as atualizações necessárias e editar a documentação final do projeto. Essa documentação deve conter todas as informações que se relacionam com o projeto: diagramas funcionais, diagramas do projeto lógico, partição, colocação de componentes, fiação, resultados dos testes, etc..

A ilustração da figura 1.1 mostra como ficaria o desenvolvimento de um projeto utilizando um sistema de automação. As linhas mais escuras representam cada fase distinta do processo. Cada fase é, na realidade, interdependente.

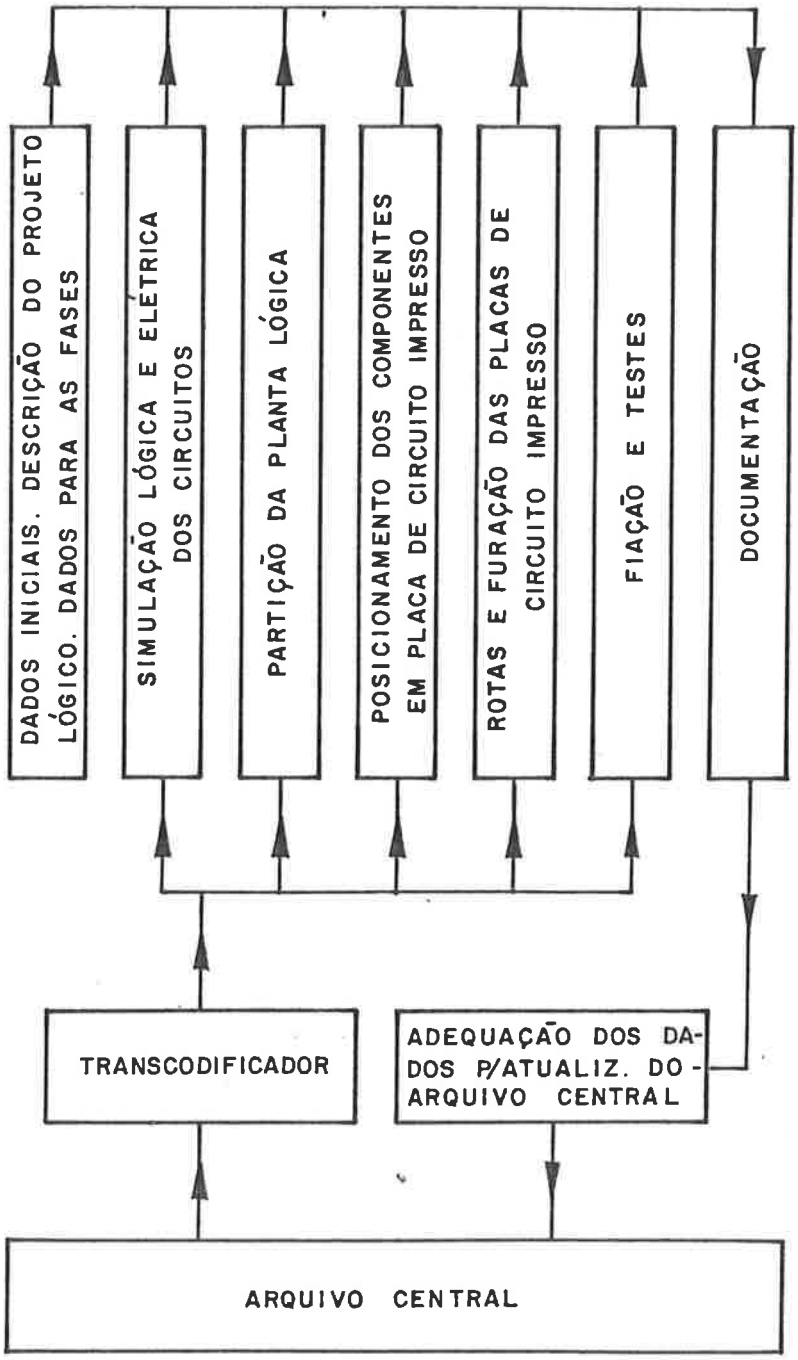



Figura 1.1

Estrutura Simplificada de um Sistema de Automação de Projetos

Com base na ilustração da figura 1.1, tem-se:

Arquivo Central - É um banco de dados alimentado, inicialmente, com informações técnicas relacionadas com o projeto. A perfeita organização desse arquivo é fundamental, pois dele dependem todas as fases da Automação. Cada uma das fases receberá (do arquivo central) informações e dados gerados em fases anteriores, e também dados específicos à sua realização. Por outro lado, cada fase processada guardará, no arquivo central, os resultados obtidos para serem usados posteriormente.

Transcodificador - É um conjunto de programas que, para cada fase, busca as informações necessárias no arquivo central, formatando-as convenientemente para uso específico de cada programa, permitindo assim a modularidade entre os diversos programas da Automação.

É através deste transcodificador que se obterá a informação se os dados para determinada fase existem no arquivo central ou se deverão ser fornecidos pelo projetista.

Descrição do Projeto Lógico - Através de uma linguagem adequada, o projetista alimenta o arquivo central com informações, vínculos e parâmetros para cada um dos programas, bem como descreve a planta do sistema elétrico de um projeto.

Documentação - Todas as vezes em que uma nova informação for gerada e adicionada ao arquivo central, uma documentação gráfica será produzida (gráficos ou relatórios) para auxiliar o projetista e também para registrar o andamento do projeto. Essas plantas e relatórios fornecem a situação de cada etapa do projeto e são de grande valia para a elaboração da montagem do circuito elétrico e também para futuras alterações ou manutenções.

Adequação dos Dados - As informações que serão alimentadas no arquivo central devem ser adequadas por um conjunto de programas auxiliares que tratarão do formato e posicionamento desses novos dados no arquivo, na fase inicial de descrição ou após o pro

cessamento de um programa específico. Esta fase, junto com o trans codificador, será responsável por todas as operações de entrada e saída de dados do arquivo central.

Simulação Lógica e Elétrica do Circuito - A simulação do comportamento de um circuito lógico, ou parte dele, será observa da através da execução desta fase. Ela possibilita ainda a intro dução de correções e simplificações, através da análise, pelo pro jetista, das saídas gráficas e informações fornecidas.

O projetista poderá ainda dispor de outros recursos de simulação, através da análise de tensão e correntes para circui tos discretos, bastando para isso efetuar a análise de tais cir cuitos em simuladores específicos.

Como simulação lógica entender-se-á a análise a nível de "gates" ou através da tabela da verdade ou da equação lógica de ca da componente.

Dessa forma, através de um processo interativo, pode-se atingir um grau de otimização do projeto praticamente impossível de se alcançar sem o emprego dos recursos computacionais.

Partição da Planta Lógica - Uma vez testado o projeto lógico do sistema elétrico, o passo seguinte é a partição da planta lógica como um todo, visando a sua execução física. Para tanto, em prega-se um conjunto de programas, segundo critérios definidos pe lo projetista, para determinar os circuitos que serão realizados em cartões ou placas de circuito impresso. Assim, entre outros da dos, a dimensão de placa de circuito impresso, o tipo de conector (número de contatos elétricos) e a dimensão dos componentes já de vem estar disponíveis.

Posicionamento dos Componentes - Após a escolha adequa da dos componentes elétricos a serem utilizados na construção de uma placa, deverá ser resolvido o problema da distribuição topoló

gica desses componentes sobre essa placa. A definição da posição dos componentes a serem utilizados poderá ser feita automaticamente após a partição. Para a definição da posição de cada componente na placa devem ser consideradas as ligações elétricas entre um componente e os demais. Para esta fase, deve-se dispor da relação dos componentes e da relação de interligação entre esses componentes. Assim que estiver definida a posição de cada componente, estará também definida uma árvore interligando contatos elétricos cujas posições estão agora estabelecidas.

Rotas e Furação das Placas - Quando estiverem determinadas as posições dos componentes elétricos na placa de circuito impresso, será necessário então definir a posição das rotas ou caminhos condutores que irão interligar dois ou mais contatos. Para tanto utiliza-se um conjunto de programas que aplicam algoritmos apropriados e respeitam os vínculos pré-estipulados para a produção de plantas de interconexão entre terminais de conectores, furos metalizados, terminais de circuitos integrados, etc. A complexidade das plantas de interconexão (rotas) dependerá da posição escolhida para os componentes, dos algorítmos usados e dos vínculos impostos, como por exemplo a distância mínima entre rotas. Assim, para uma mesma placa, é possível obter várias soluções em função dessas restrições. Esse fato mostra a inter-relação existente entre partição, posicionamento e as rotas de circuito impresso.

Fiação e Testes - Um conjunto de programas estabelece, nesta fase, as ligações de fiação em painéis de conectores, respeitando os vínculos da tecnologia empregada("wire-wrap", por exemplo). Esses programas fornecem, como resultado, o comprimento de cada fio a ser utilizado, níveis de interconexão e, entre outros, os códigos dos pinos a serem interligados. Um outro conjunto de programas, recebendo os dados referentes a uma placa - esquema lógico, relação dos componentes, descrição das ligações - determinará conjuntos de seqüências de sinais de teste, para determinação de falhas elétricas em placas já montadas.

Cada uma destas fases é constituída de programas independentes que, em função dos resultados apresentados, podem impli

car no reprocessamento de fases anteriores, buscando sempre uma otimização do projeto através de um procedimento interativo.

# CAPÍTULO II

REFERÊNCIAS: [4] [5] [14] [15]

## II.1 O SISTEMA INTEGRADO DE AUTOMAÇÃO

A figura em diagrama em blocos apresentada no capítulo anterior apresenta em seu conjunto as etapas constituintes de um sistema de automação de projeto onde a integração de diferentes fases é o aspecto predominante.

Essa estruturação entretanto não é definitiva, pois, alguns detalhes devem ser acrescentados para permitir uma melhor utilização da estrutura como um todo.

Assim sendo torna-se necessário acrescentar àquela figura alguns blocos que deverão desempenhar a função de selecionadores de uma ou outra alternativa que se apresentam.

A figura 2.1 é então, um diagrama em blocos completo do sistema de automação integrado que se pretende implementar.

A técnica principal a ser utilizada no desenvolvimento de um sistema deste tipo é a de modularizar o sistema, ou seja, defini-lo como vários blocos estanques e que poderão se relacionar entre si através de regras previamente estabelecidas.

Usando-se tais regras é possível restaurar o sistema e reparametrizá-lo para processamento das fases em um tempo mínimo.

Técnicas desse tipo não são muitas vezes utilizadas devido ao planejamento adicional que requerem e ao tempo gasto em implementação. São entretanto, bastante importantes pois permitem estruturar o sistema e implementar as partes individualmente e depois interligá-las como um todo.

Outro aspecto importante que se apresenta quando do planejamento de um sistema como este é a interface projetista - máquina (computador). É esta sem dúvida a parte mais difícil de implementar pois é dela que dependerá a maior ou menor facilidade de adaptação do sistema.

Outros itens também importantes e que devem ser pensados quando do planejamento do sistema, levando-se em consideração que deve-se dispor de estruturas de arquivos para armazenamento de informações essenciais ao sistema, são os seguintes:

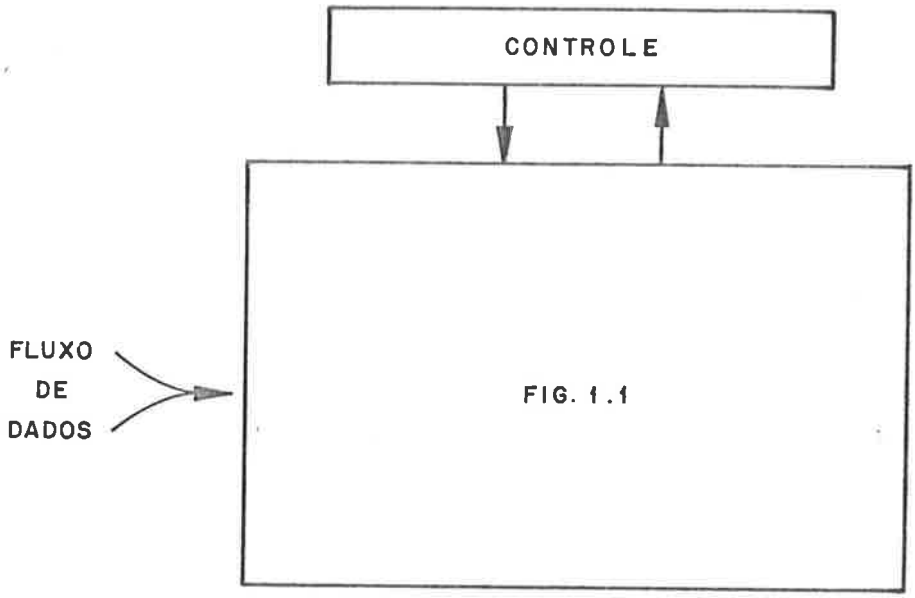
- flexibilidade no manuseio de dados
- adição de novos dados ou informações sem maiores problemas de reorganização de arquivos
- acesso fácil a dados em qualquer circunstância
- possibilidade de acomodação de informações quer na forma fixa ou comprimento variável
- capacidade de deleção ou armazenamento de arquivos sob controle do sistema.

Ênfase especial deve ser dado quanto a eficiência do sistema.

## II.2 OPERAÇÃO DO SISTEMA

O sistema integrado de automação poderá ter sua operação total, comandada pelo próprio sistema ou então poderá ser utilizado em partes e, neste caso, em sua operação deverá haver a intervenção do projetista em cada fase que está sendo utilizada.

Em sua operação, total ou parcialmente comandada pelo próprio sistema, o projetista deverá então fornecer através de




Estrutura do Sistema Integrado de Automação com fase adicional

Figura 2.1

linguagem de comunicação com o sistema, todas as informações necessárias e suficientes à respeito do projeto.

Essas informações, manipuladas de forma conveniente, serão colocadas em arquivos adequados, já definidos e que constituem a estrutura do arquivo central do sistema .

Ver-se-á mais adiante a estruturação e constituição de tais arquivos, objetivo principal deste trabalho, pois constituem a organização central de todo o sistema.

Com estas informações fornecidas pelo projetista e de posse dos blocos constituintes de cada fase da automação integrados ao sistema, pode-se obter agora novos dados através do processamento de cada fase.

É neste ponto que deve-se efetuar uma distinção entre o processamento da automação, totalmente comandado pelo sistema ou então sob comando parcial.

Considera-se, em primeiro lugar, o comando como sendo o sistema, ou seja, todas as fases processadas na seqüência que a seguir se expõe.

O sistema já dispõe de todas informações necessárias para, se assim o usuário o desejar, efetuar a simulação lógica do projeto.

Esta simulação deverá se desenvolver utilizando como informação principal a tabela da verdade de cada componente, disponível no arquivo de funções associada a cada um. Vinculada a esta tabela existirá uma outra constituída de atrasos correspondentes a cada conjunto de sinais de entrada.

O projetista disporá agora de recursos que o permitirá simular o projeto como um todo ou apenas parte dele.

Através de recursos gráficos ou de impressão, o projetista analisará o resultado da simulação e efetuará as modificações ou correções que julgar necessárias, utilizando para tanto, programa que permita a edição do seu arquivo de dados para a simulação.

Após várias interações com o sistema e uma vez atingido o seu objetivo, ou seja testar o projeto antes de implementá-lo, passa-se ao processamento das demais fases que permitirão a realização física do projeto através de informações fornecidas em cada fase do sistema e que constituem sua documentação.

A fase seguinte a ser processada deverá ser a da partição do projeto lógico, visto que, toda informação a respeito do mesmo já foi fornecida ao sistema pelo projetista.

O resultado do processamento desta fase deverá ser o preenchimento de novos arquivos com informações a respeito do número de placas, que deverá compor o projeto, os componentes que deverão estar presentes em cada placa, conecções entre pinos de componentes, sinais gerados ou utilizados por uma determinada placa, sinais disponíveis em conectores e eventuais pinos de testes. Todas essas informações serão necessárias para as diversas fases seguintes do sistema e deverão estar acomodadas nos arquivos em formato tal, que permita ao projetista a sua pronta utilização, bem como sua adequação no caso de achar conveniente qualquer modificação no resultado fornecido pela fase processada. O sistema deve permitir então a edição de dados ou informações existentes nos arquivos.

Para esta fase o projetista poderá, se o desejar, fixar algum critério para a distinção dos componentes por placa, por exemplo, fixando alguns componentes em determinada placa, ou mesmo amarrando sinais em conectores.

Se esse procedimento não for utilizado a partição correrá por conta do algoritmo utilizado no programa, determinando-se assim a distribuição inicial dos componentes por placa e, por conseguinte, estabelecendo o critério para a distribuição final.

A fase seguinte a ser processada será a que permite efetuar o posicionamento de componentes em cada placa.

Para tanto, esta fase deverá ter diponível as informações, de quais os componentes por placa e as ligações entre os mesmos. Nesta fase a proximidade ou conectividade das ligações é importante e será ela que indicará a localização do componente na placa. As dimensões de cada componente também deverão ser significativas, pois ditarão a ocupação de cada placa.

Como resultado desta atividade ter-se-á a inclusão, no arquivo de placas, das informações a respeito da localização de cada componente na placa, inclusive com sua orientação.

Novamente, a exemplo da fase anterior, o projetista poderá ter acesso a esses resultados e adequá-los, se assim o desejar, através da edição das informações contidas nos arquivos de placas.

O critério a ser fixado pelo projetista para esta fase se assim o desejar, será a posição fixa de algum componente. Caso contrário o próprio algoritmo do programa de posicionamento ditará a posição do componente.

Com os resultados já obtidos através da partição e do funcionamento dos componentes, pode-se agora partir para a realização da fase seguinte do projeto, através da automação integrada, que será a elaboração de dados necessários à fiação em "WIRE-WRAP" de cada placa particionada. Esta fase será opcional, pois muitas vezes o projetista poderá não utilizá-la, partindo diretamente para o estabelecimento de uma placa de circuito impresso.

É, entretanto, importante pois permite, se assim se desejar, obter dados de fiação de placas para testes iniciais em bancadas de laboratórios.

Como se vê, esta fase nada introduz como informações para o sistema, pois apenas utiliza dados já existentes.

A fase que permite obter informações para rotas e fiação das placas de circuito impresso pode agora ser processada.

O projetista deverá, antes do processamento desta fase, fornecer algumas informações ao sistema, entre elas: tipo da placa a ser trabalhada, número de camadas desejadas ou disponíveis, tipos de ligações e interagir quanto às prioridades das ligações a serem efetuadas.

Esta fase é bastante complexa e deverá permitir a interação do projetista com o sistema através de dispositivos gráficos convenientes, ou seja, "displays" dotados de canetas magnéticas durante a etapa de estabelecimento das rotas e de "plotters" planares dotados de dispositivos fotográficos na etapa final de obtenção da chamada arte-final.

O processamento desta fase permitirá que o projetista obtenha dados quanto :

- relação das posições das ligações;
- lay-out de cada uma das faces da placa de circuito impresso;
- posição dos furos metalizados necessários à realização das ligações;
- informações adicionais, como o comprimento de cada ligação já definida, relação eventual de ligação não possível de realização através do processo de rotas, tipos de brocas a utilizar, etc.

Através da adequação dessas informações, pode-se também obter a fita para furação da placa de circuito impresso para comando das máquinas de controle numérico que as realizarão.

Aqui também preve-se a intervenção do projetista para adequar as informações, através de manipulação dos dados por programa de edição de arquivos.

A fase de documentação, embora aqui seja descrita como a última, pode ser acionada a qualquer momento, desde que o projetista considere necessário, para documentar o processo, ou melhor, fornecer informações do projeto necessárias para a implementação.

Deve entretanto conter programas que permitam fornecer todas as informações necessárias ao perfeito entendimento do projeto.

Entre os documentos que deverão ser elaborados pela fase de documentação pode-se citar: os diagramas lógicos; lista de componentes com respectivas funções, a localização, tipo e quantidade; diagramas de fiação; sinais associados a pino ou conectores; resultados gráficos de simulações; etc.

Convém salientar que, uma boa estrutura de arquivos permitirá sem problemas a obtenção de qualquer tipo de documentação - que se desejar.

Quando operando sob comando parcial do sistema de automação, o projetista poderá fazer uso de qualquer fase, independentemente de ter ou não processado a fase anterior.

Para tanto será necessário, pois o sistema é modular, introduzir através de linguagens adequadas, as informações requeridas pela fase a ser processada.

É neste ponto que deverá entrar em ação o programa de <u>transcodificação</u> pois este programa deverá investigar, para aquela fase a ser processada, quais os dados necessários e se estes já existem nos arquivos do sistema ou se deverão ser alimentados pelo projetista.

Uma vez processada a fase, o comando voltará ao projetista que decidirá qual a próxima atividade.

Para cada fase deve - se estabelecer um programa de comunicação que seja o mais geral possível e que possa ser utilizado em todas, evitando-se assim dificuldades para o projetista.

Os programas de edição de arquivos deverão também estar disponíveis em cada uma destas fases e se enquadrarão no mesmo esquema de linguagem de comunicação.

Assim sendo, um único programa de comunicação do projetista com o computador deverá preencher os requisitos para todas as fases.

## II.3 O CONTROLE DO SISTEMA

Uma vez estabelecida as principais fases e componentes de um sistema de automação de projetos, verifica-se que todas as fases que o compõe dependem de informações que o projetista deve fornecer para que estas fases possam ser processadas.

Esses dados ou informações ou já existem no sistema, caso em que podem ser acessados através de uma consulta nos arquivos existentes, ou então, não existem e devem ser fornecidos pelo projetista.

Tanto no primeiro como no segundo, é de grande importância a forma como essas informações são fornecidas ao sistema, pois é da sua facilidade de utilização que dependerá, em grande parte, o sucesso do próprio Sistema de Automação.

Muitas vezes o Sistema de Automação apresenta inúmeras características que o tornam de grande valia no auxílio de um projeto, entretanto a forma de entrada de dados é tão complicada e difícil que acaba por desistimular o possível usuário interessado em suas vantagens.

Pode-se entretanto diversificar a forma de entrada de dados, estabelecendo-se distinção entre as formas de entrada de dados que devem ser utilizadas pelo projetista para desenvolver o seu projeto e aquela que é utilizada com menor freqüência e destinada a equipar o sistema de recursos necessários à algumas fases de um projeto.

Entretanto, desde que o sistema seja definido para ser utilizado pelo projetista e assim procedendo esta-se procurando tornar o sistema o mais independente possível de quem o desenvolveu, é claro que o tipo de comunicação homem - máquina, no que tange fornecimento de dados e controle do sistema, deve ser o mais flexível possível.

Dentro do esquema estabelecido para o sistema integrado de Automação de Projetos e representado em blocos na figura 2.1, a comunicação homem-máquina deve-se processar em diversas fases:

a) criação e manutenção de informações no arquivo central;

b) fornecimento de informações para o processamento de fases distintas;

c) fornecimento de dados auxiliares para a documentação;

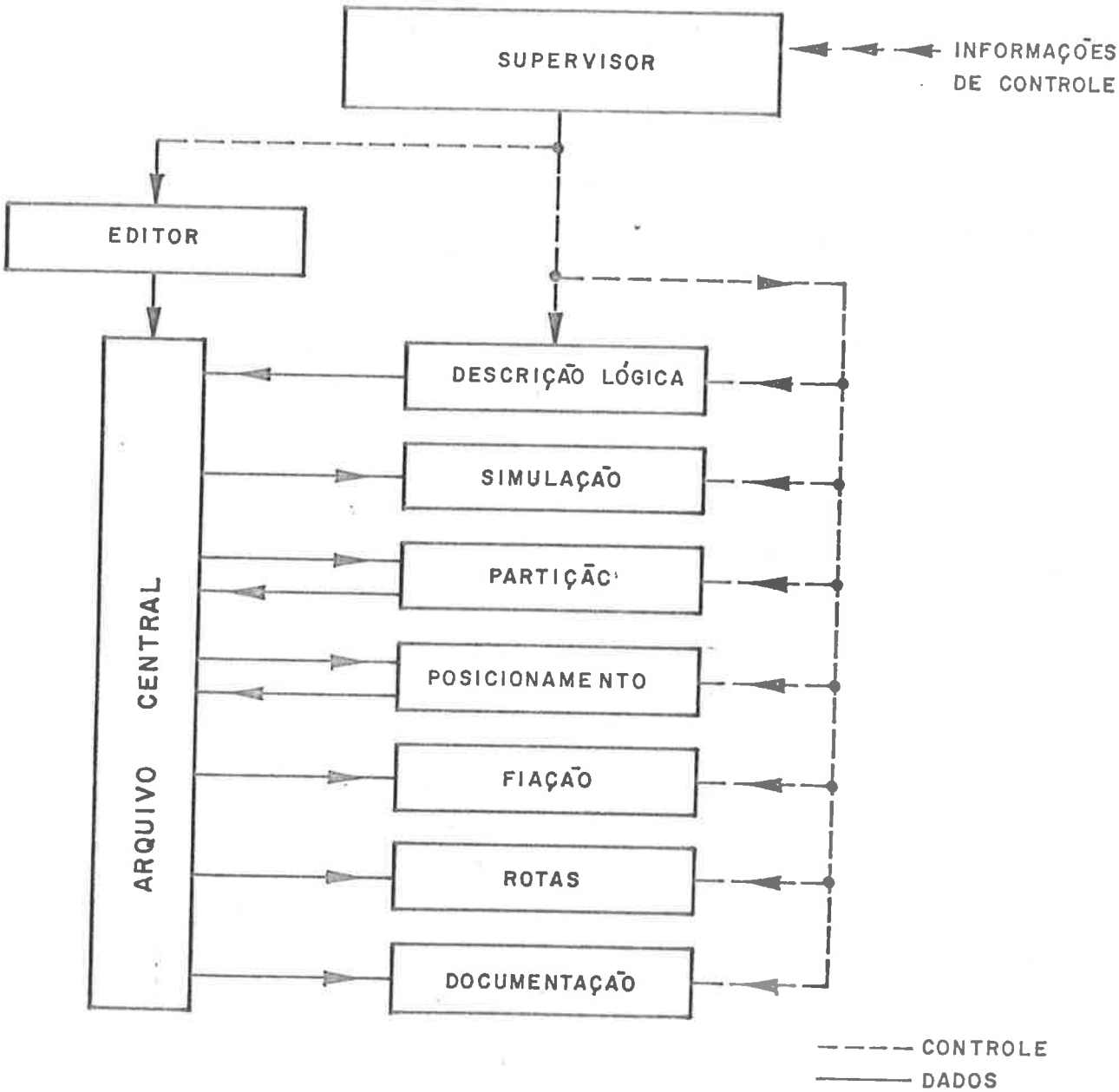
d) fornecimento de informações necessárias para a descrição do projeto como um todo;

f) estabelecimento de comandos para o sistema.

Pelo esquema acima pode-se antecipar que, para o gerenciamento de dados a serem fornecidos ao sistema e utilizados pelas diversas fases, deve-se dispor de um supervisor, cuja função principal será a de entender os comandos emitidos pelo projetista, através da utilização de uma linguagem única, para todas as fases do sistema.

Vê-se então, a partir de agora, que o transcodificador, colocado no diagrama em blocos, nada mais é que um Supervisor do Sistema, pois executa a Supervisão do Sistema através da interpretação da linguagem utilizada pelo projetista.

A figura 2.2 é o diagrama em blocos final do Sistema.




Estrutura do Sistema Integrado de Automação a ser implementado

Figura 2.2

# CAPÍTULO III

REFERÊNCIAS: [1] [3] [6] [7] [9] [10] [11] [12] [13]

A implementação de um sistema Integrado de Automação como descrito no capítulo anterior, baseia-se principalmente na estrutura do arquivo central.

Neste capítulo dever-se-á apresentar as informações que são necessárias à cada fase da automação, como esses dados devem ser alocados nos arquivos, a estrutura desses arquivos e algumas considerações a respeito de algoritmos a serem utilizados.

## INFORMAÇÕES NECESSÁRIAS À CADA FASE

Para as diferentes fases as informações a serem fornecidas são analisadas, em separado, a seguir.

### III-1 BIBLIOTECA

A fase de estruturar uma livraria de componentes, funções e padrões para o Sistema de Automação é denominada de Biblioteca.

Nesta fase o projetista deverá fornecer todas as informações que necessitar no desenvolvimento do seu projeto lógico, ou seja quais as funções que utilizará, os componentes que farão referência a essas funções e os padrões de ocupação e de componentes.

Para descrever as funções de um componente as informações necessárias são as seguintes:

a) Tipo de função, associada a um código que estabelecerá uma ligação entre a função a ser executada e o número do componente fornecido pelo fabricante.

Exemplo: função NOR corresponde ao componente 7402 do fabricante.

b) Sinais simbólicos de entrada, para a execução da função do ítem a.

c) Sinal simbólico de saída, que representa a saída da função descrita em a , utilizando-se as entradas descritas em b.

d) "Fan-in" e "Fan-out", para o componente.

Para se definir o que é "fan-in" ou "fan-out" como utilizado na descrição, deve-se antes de mais nada definir-se o que é unidade de carga.

Unidade de Carga é considerada como uma entrada simples de uma porta lógica de uma família qualquer. Isso significa para o caso da família TTL - 7400, uma carga representada pela corrente de menos 1.6 mA quando no estado lógico "0" e de 40μA quando no estado lógico "1".

"Fan-in" de uma entrada de um elemento lógico é o número obtido através de normalização em relação à definição de unidade de carga.

"Fan-out" de uma saída, é o número máximo de unidades de carga que podem ser ligadas a essa saída, assegurando os níveis "0" e "1".

Para determinados componentes o "fan-in" é muitas vezes o mesmo para as várias entradas do elemento e o "fan-out" pode obedecer essa mesma situação.

Entretanto, havendo distinção entre os "fan-in" das entradas e entre os "fan-out" das saídas, esse fato deverá ser especificado através de uma lista relacionando o "fan-in" de uma deter

minada entrada com o sinal simbólico que a representa. Idem para o "fan-out".

e) Função associada ao elemento

Deverá ser representada na Biblioteca através de uma tabela de verdade onde entrarão como cabeçalho os nomes simbólicos das entradas e saídas e como linhas, os possíveis valores assumidos pelos mesmos durante a execução da função exercida pelo elemento.

Associada a esta tabela deverão estar presentes outras tabelas que serão descritas a seguir.

As figuras III-1A e III-1B são representativas das descrições das funções para o 7402 NOR e para o 7474 FF.

Comparecendo-se elementos nos quais, as entradas encontram-se relacionadas com atrasos do tipo "SETUP", e larguras do tipo do "CLOCK", essas entradas devem ser tratadas através de modelos como representado na figura III-1B.

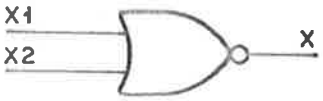
f) Atrasos

Dentre as características associadas a um elemento lógico, uma das principais a ser considerada será o atraso a ele associado. Muitas vezes, como é comum ocorrer, um único atraso não é característica principal de um elemento, pois dependendo do projeto lógico do qual o elemento participa, torna-se necessário levar outros atrasos em consideração.

De uma maneira geral o que se considera, é o atraso de propagação que caracteriza o elemento lógico ao variar sua saída do valor lógico "0" para "1", que será denominado TD01 e o atraso correspondente ao elemento lógico mudar do nível lógico "1" para "0", que será denominado TD10.

Esses atrasos são sempre valores constantes e o que se utiliza, são valores mínimos, nominais e máximos dos atrasos fornecidos pelo próprio fabricante.

7402 NOR




FUNÇÃO

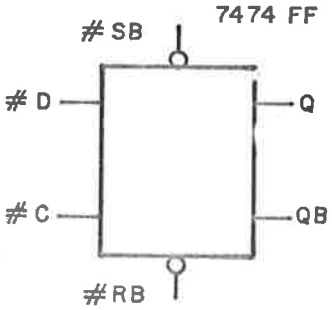
| X1 | X2 | X |
|---|---|---|
| Ø | Ø | 1 |
| 1 | X | Ø |
| X | 1 | Ø |

X = OU Ø
OU 1

ATRASOS

| | MINIMO | | NORMAL | | MÁXIMO | |
|---|---|---|---|---|---|---|
| | X1 | X2 | X1 | X2 | X1 | X2 |
| TDØ1 | 12 | 12 | 12 | 12 | 22 | 22 |
| TD1Ø | 8 | 8 | 8 | 8 | 15 | 15 |

7474 FF



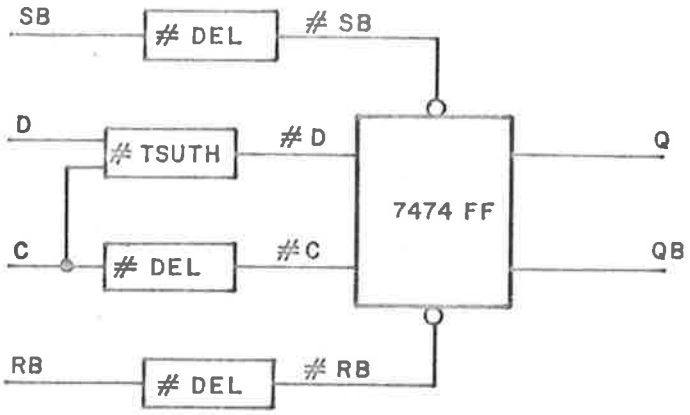

FUNÇÃO

| #SB | #RB | #C | #D | Q |
|---|---|---|---|---|
| Ø | Ø | X | X | ERRO |
| Ø | 1 | X | X | 1 |
| 1 | Ø | X | X | Ø |
| 1 | 1 | U | Ø | Ø |
| 1 | 1 | U | 1 | 1 |

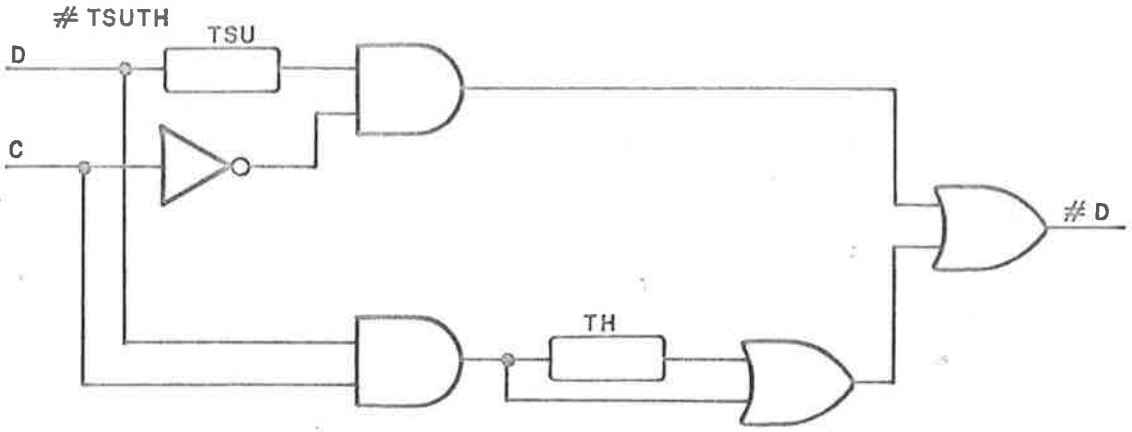
ATRASOS

| | MINIMO | | | NORMAL | | | MÁXIMO | | |
|---|---|---|---|---|---|---|---|---|---|
| | #SB | #RB | #C | #SB | #RB | #C | #S | #R | #C |
| TDØ1 | 25 | 25 | 10 | 25 | 25 | 14 | 25 | 25 | 25 |
| TD1Ø | 40 | 40 | 10 | 40 | 40 | 20 | 40 | 40 | 40 |

FIG. III - 1A




MODELO DO 7474 FF



MODELO DAS FUNÇÕES

**FUNÇÃO**

| D | C | # D |
|---|---|---|
| Ø | X | Ø |
| 1 | X | 1 |

OS "GATES" SÃO IDEAIS

ATRASOS { TSU, TH, DEL } DADOS PELO COMPONENTE

FIG. III - 18

As demais características dos elementos lógicos sensíveis à borda e que são utilizadas na fase de simulação, dizem respeitos aos tempos de "hold" e de "set-up", bem como as larguras do sinais de "CLOCK", "SET" e "RESET".

Todos esses atrasos são fornecidos através de tabelas para o Sistema, como representado na figura III-1B.

O tempo de "hold" é definido como sendo, o intervalo de tempo em que as entradas correspondentes a dados, devem permanecer estáveis, após o pulso de "CLOCK" ter caído abaixo ou se elevado acima de um determinado valor(ordem 50% ou então 1.5V no caso da família TTL) para assegurar operação correta do "flip-flop" , por exemplo.

O tempo de "set-up" é definido como sendo, o intervalo de tempo em que as entradas correspondentes a dados, devem permanecer estáveis, antes da ação do pulso de "CLOCK" no "flip-flop" por exemplo.

A figura ilustra, com a forma de onda, os tempos de "hold" e "set-up".

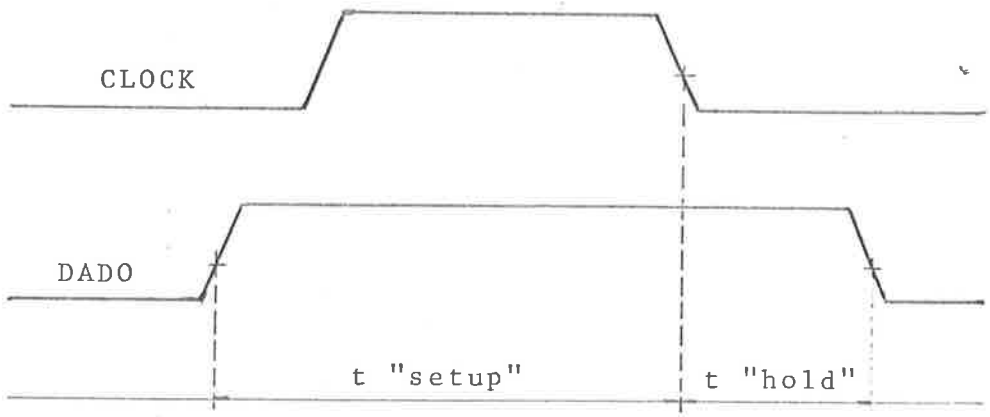



Figura III.2

Para componente propriamente dito, as informações necessárias à elaboração do arquivo são as seguintes:

a) O código de fabricação

É um número que corresponde ao fornecido pelo fabricante. Por exemplo, um circuito NOR poderá ser identificado como sendo o 7402.

b) O número de elementos que o constitue

Corresponde ao número de elementos integrantes do componente. Para o 7402, tem-se quatro elementos NOR.

c) A "pinagem" do componente

Nesta descrição o projetista deverá fornecer o número total de pinos do componente, bem como, detalhar as relações entre os pinos de saída e os correspondentes pinos de entrada. A função de relacionamento deverá estar definida no arquivo de funções.

Na descrição das interligações dos sinais internos que dependem de atraso que foram descritos por funções, os números dos pinos devem ser maiores que o número total de pinos dos componentes.

Na descrição de "pinagem" os pinos de alimentação devem ser explicitados no final da descrição, separadamente.

A figura III-3 mostra um exemplo de ligações para o 7402 e para o 7474.

d) A dimensão do componente

Deve-se fornecer neste caso a dimensão do componente, dando-se uma folga da ordem de 20% para permitir a inserção dos pinos do mesmo no circuito.

No caso do componente ser um soquete, a dimensão será a real do soquete.

e) A dissipação do componente

Fornecido a partir dos dados de manuais e servirá para a contabilização das necessidades energéticas de uma placa, por exemplo.

f) Os padrões disponíveis para este componente

O projetista deverá fornecer quais os códigos de padrões que correspondem a esse componente para permitir sua escolha, quando da colocação do mesmo em placas de circuito impresso.

Para a geração dos padrões de componentes, as informações necessárias são as seguintes:

a) O código pelo qual será conhecido o padrão

Através deste dado pode-se localizar, para cada componente, a respectiva matriz de pontos com permissão ou não para o traçado das rotas.

b) O terminal básico de referência

Corresponderá ao terminal do componente que servirá como "pivot" para a localização do mesmo e para a geração das coordenados dos demais pinos ou terminais.

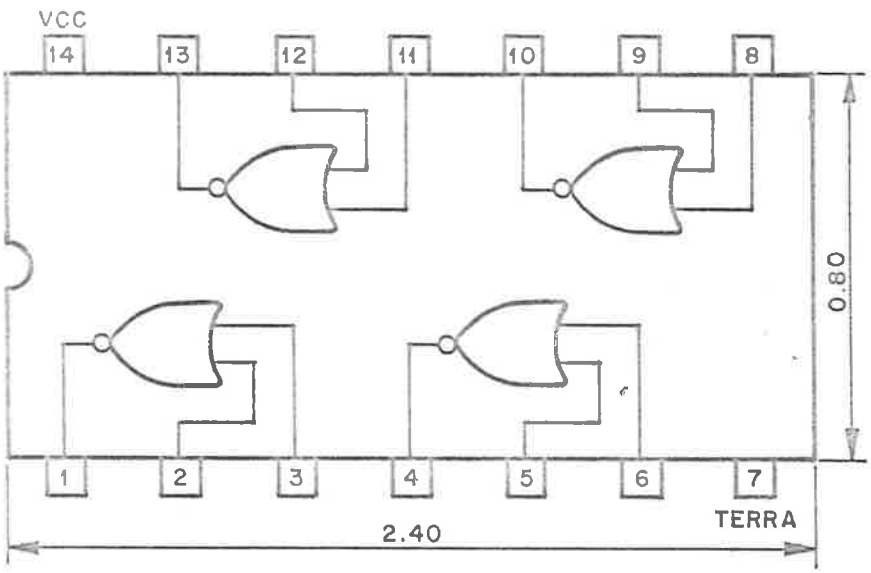
c) O número de terminais do componente

Este número corresponde ao número de pinos que acessam o componente.

d) O tipo de isolação a ser utilizado

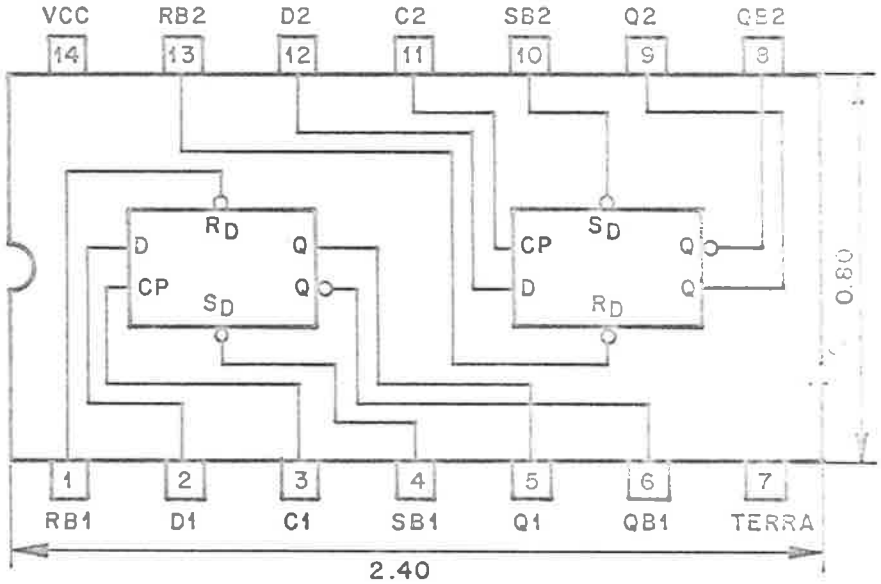
Esta informação permitirá acessar no arquivo correspondente de padrões, o tipo de isolação para cada pino do componente delimitando-se as áreas não permitidas para o traçado de rotas.

## 7402




DIMENSÃO = 0.80x2.40
PADRÃO = I14N,I14V

## 7474




Número de elementos = 4
Pinagem: 14 pinos

| Pinos de Saída | função | pinos de entrada | |
|---|---|---|---|
| 01 | 7402 NOR | 02 | 03 |
| 04 | 7402 NOR | 05 | 06 |
| 10 | 7402 NOR | 09 | 08 |
| 13 | 7402 NOR | 12 | 11 |
| 14 | VCC | | |
| 07 | TERRA | | |

dissipação = 135 mW

Número de elementos = 2
Pinagem: 14 pinos

| Pinos de Saída | função | Pinos de entrada | | | |
|---|---|---|---|---|---|
| 05 | 7474FF(Q) | 15 | 16 | 17 | 18 |
| 06 | 7474FF(QB) | 15 | 16 | 17 | 18 |
| 08 | 7474FF(QB) | 19 | 20 | 21 | 22 |
| 09 | 7474FF(Q) | 19 | 20 | 21 | 22 |
| 14 | VCC | | | | |
| 07 | TERRA | | | | |
| 15 | # DEL | 04 | | | |
| 16 | # DEL | 01 | | | |
| 17 | # DEL | 03 | | | |
| 18 | # TSUTH | 02 | 03 | | |
| 19 | # DEL | 10 | | | |
| 20 | # DEL | 13 | | | |
| 21 | # DEL | 11 | | | |
| 22 | # TSUTH | 12 | 11 | | |

dissipação = 250 mW

FIG. III-3

e) As coordenadas locais do terminal básico

Estas coordenadas correspondem às coordenadas do terminal básico dentro do reticulado padrão utilizado. A partir destas coordenadas será obtida as coordenadas dos demais pinos ou terminais.

f) O incremento em relação ao terminal básico

Através desta informação determinar-se-á em relação às coordenadas locais, a posição dos demais terminais.

g) A ocupação

Através dela definir-se-á se a face ocupada é a superior, a inferior ou ambas.

Para a geração dos padrões de ocupação, as informações necessárias são as seguintes:

a) O código do padrão de ocupação

Através do qual, na descrição do padrão de componentes, deverá ser referenciado.

b) O terminal básico

Que deverá ser um zero para indicar que se trata de um padrão de ocupação.

c) O número de células

Que compõe o padrão de ocupação.

d) Os tipos de ocupação das células

Através desta informação monta-se uma estrutura associadando-se a cada célula um tipo de ocupação, quais as células ocupadas e a face onde deve ser feita a ocupação.

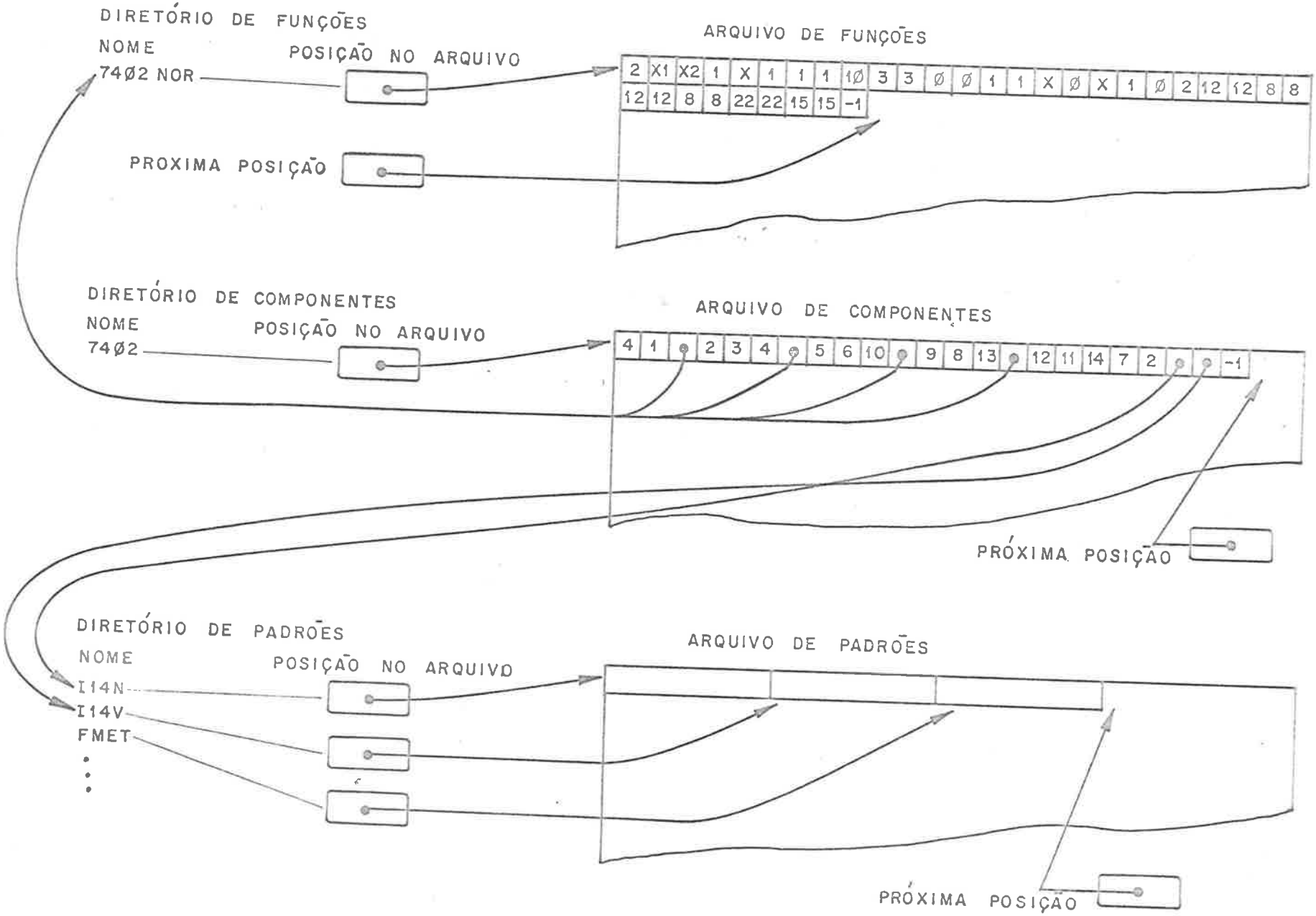



FIG. III-4
BIBLIOTECA

A referência |11| deste trabalho foi utilizada para a obtenção das informações para a geração dos padrões. Todos os dados até aqui descritos, fornecidos pelo projetista, deverão ser armazenados em três arquivos distintos sendo cada um deles, destinado a uma finalidade:

- função - ARQUIVO DE FUNÇÕES
- componente - ARQUIVO DE COMPONENTES
- padrão de ocupação e padrão de componentes - ARQUIVO DE PADRÕES.

Para cada arquivo, existirá um diretório que indicará os elementos presentes no arquivo e respectiva posição dentro do mesmo.

A figura III-4 mostra a geração da função 7402 NOR e do componente 7402 no arquivo, bem como, a estrutura de arquivos utilizada pela Biblioteca.

Convém salientar que, um esboço da linguagem que permite a entrada desses dados, encontra-se descrito no capítulo seguinte.

## III-2 DESCRIÇÃO LÓGICA

Para esta fase o projetista deverá, inicialmente, numerar os componentes de sua planta lógica utilizando-se de um código para cada um e atribuir nomes aos sinais.

Uma vez encerrado esse trabalho o projetista deverá decidir se seus componentes serão classificados como singelos ou complexos.

Admite-se como complexos, àqueles componentes que não tenham função lógica simples (NAND, NOR, OR, AND, etc).

Os demais componentes sempre serão considerados como singelos.

As informações adicionais necessárias são:

a) Descrição de todos os componentes singelos presentes na planta lógica

Para tanto o projetista utilizará os nomes, atribuidos na fase de preparação de dados, para as saídas dos componentes singelos.

A relação entre os sinais de entrada e de saída é feita através da função do singelo.

b) Descrição de todos os componentes complexos presentes na planta lógica

Para tanto o projetista deverá utilizar o número, o código do componente com respectiva função e os pinos associados aos nomes dos sinais atribuidos na preparação de dados.

c) Fornecer os nomes do sinais gerados externamente

Caberá agora ao projetista gerar uma lista de nomes de sinais externos ao seu projeto.

Com essas informações, o Sistema estará de posse de todos os dados necessários para realizar as etapas da Automação.

A medida em que esses dados são fornecidos a fase de Descrição Lógica deverá gerar uma tabela de sinais através da qual serão contabilizadas as cargas normalizadas, utilizando-se para tanto o recurso de somar a carga de "fan-out" e subtrair as cargas "fan-in". Além dessa contabilização serão marcados os sinais gerados, utilizados, e os gerados e utilizados, recorrendo-se para tanto a códigos distintos para cada indicação.

Ao final da descrição, não havendo erros na codificação o Sistema gerará um bloco, dentro do arquivo de Descrição Lógica,

bedecendo-se o mesmo formato em que foi fornecido pelo projetista e indicará em que posição este bloco se localiza.

A tabela de sinais gerada estará anexa à parte final da descrição lógica, com as informações dos tipos de sinais e do balanço das cargas normalizadas.

Necessita-se, agora definir o que é um bloco de informações.

Define-se como bloco, um sub-conjunto de um arquivo. Cada bloco será identificado por um nome e conterá informações relacionadas a uma determinada descrição.

A figura III-6 representa a estrutura utilizada para a descrição lógica da planta lógica da figura III-5.

Um esboço da linguagem de descrição lógica é apresentada no capítulo IV.

## III-3 SIMULAÇÃO

As informações necessárias à simulação do projeto lógico encontram-se no arquivo das Descrições Lógicas e na Biblioteca do Sistema.

O projetista de acordo com sua necessidade, deverá fornecer condicionantes para a realização da sua simulação. Listam-se como tais condicionantes, a imposição de condições iniciais a determinados sinais e a geração de sinais para a excitação dos circuitos a serem simulados.

Para a realização da simulação, esta fase deverá montar - com base nos dados disponíveis no bloco correspondente à descrição lógica em questão, bloco este localizado no arquivo das Descrições Lógicas, e nos arquivos da Biblioteca, uma estrutura para cada componente e interligando-se através de ponteiros.

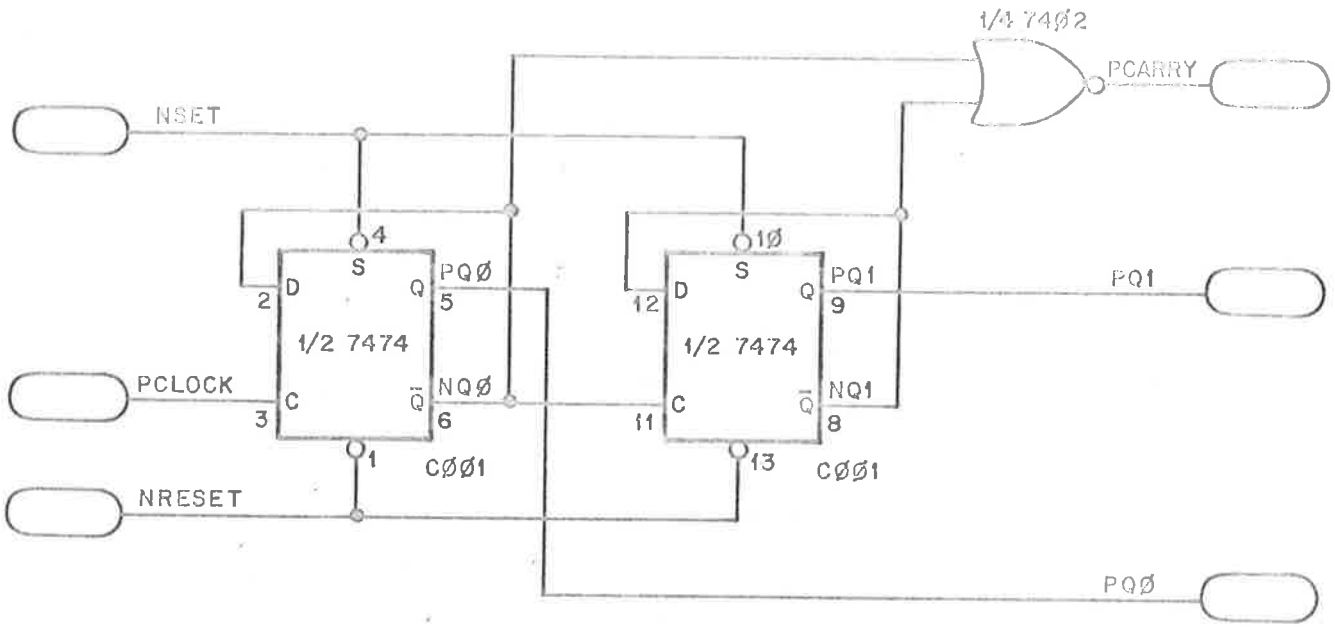
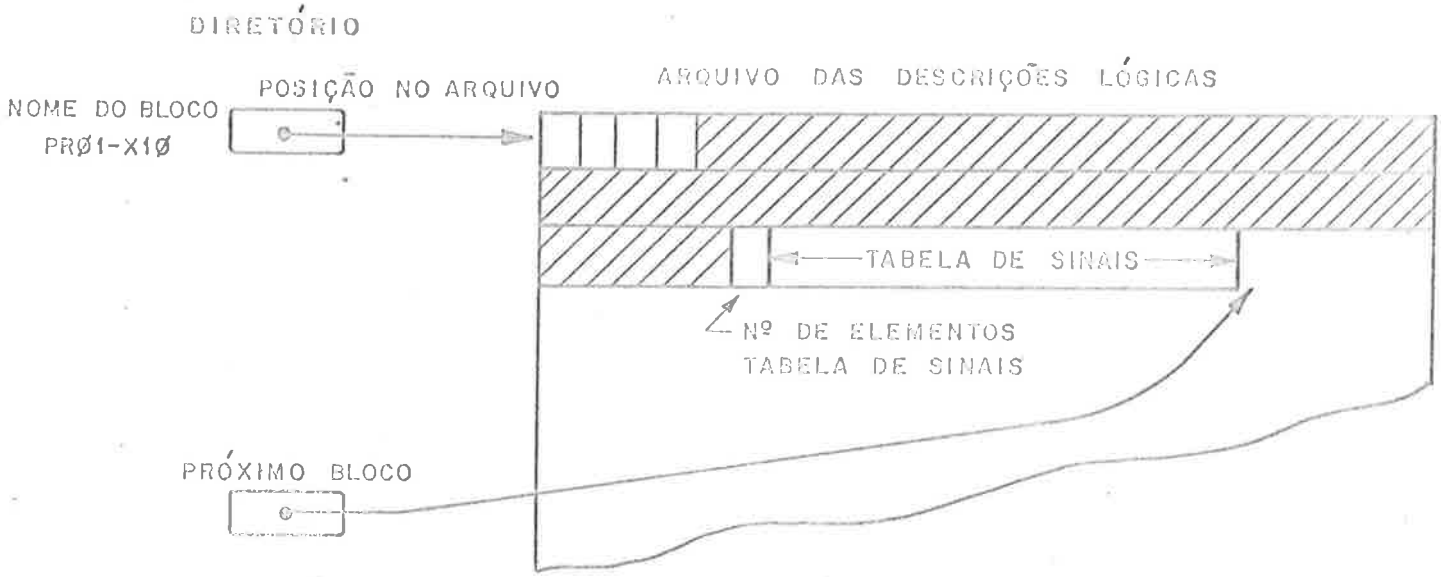

1/4 74Ø2
PCARRY
NSET
4
S
PQØ
2
D
Q
5
1/2 7474
PCLOCK
C
NQØ
3
6
1
CØØ1
NRESET
1Ø
S
PQ1
12
D
Q
9
1/2 7474
C
NQ1
11
8
13
CØØ1
PQ1
PQØ

FIG. III - 5




TIPO = 1 - GERADO
2 - UTILIZADO
3 - GERADO E UTILIZADO

TABELA DE SINAIS

| NOME | TIPO | "FANIN/FANOUT" |
|---|---|---|
| PCARRY | 1 | 1Ø |
| NQØ | 3 | 6 |
| NQ1 | 3 | 6 |
| NRESET | 2 | -6 |
| PCLOCK | 2 | -2 |
| NSET | 2 | -4 |
| PQØ | 1 | 1Ø |
| PQ1 | 1 | 1Ø |

FIG. III-6
DESCRIÇÃO LÓGICA

Cada estrutura conterá:

- ponteiros para as posições da tabela de sinais correspondentes aos sinais referenciados.

- ponteiros para os diretórios de funções ou componentes, diretórios esses que definem o funcionamento do componente através de tabelas da verdade e de atrasos.

- lista de variáveis a serem utilizadas pela simulação com a finalidade de contabilizar atrasos já decorridos desde a última variação das entradas para o elemento correspondente.

- posições destinadas à semáforos e indicadores.

O algoritmo para utilização na simulação consta basicamente de dois elos principais. Cada elo representa uma ação fundamental para cada passo da simulação.

O primeiro elo, que também corresponde ao primeiro passo da simulação, é chamado de elo da execução; o segundo será chamado elo da ativação. A existência desses dois elos é fundamental para o algoritmo e toda simulação será conduzida pela passagem obrigatória através desses elos.

Admitir-se-á por ora, que já se dispõe de uma estrutura de dados numérica, através de listas que permitem a ligação de um determinado conjunto de dados da estrutura com o seguinte e assim sucessivamente. O conjunto final da estrutura deverá apresentar mesma organização, apenas apresentando informação adicional que permita indicar que é o último da lista.

Dentro da estrutura em questão, existem para cada conjunto de dados, duas posições bem definidas e necessárias ao algoritmo. A primeira dessas posições será denominada de contador e a segunda de semáforo. Em cada conjunto de dados, tanto o contador como o

semáforo permitirão indicar qual o elo que deverá ser processado.

O contador deverá ter conteúdo representativo do atraso do elemento que representa na estrutura, e o semáforo será apenas o indicador de atividade para o elemento representado na estrutura. Apresenta-se ainda, uma outra posição denominada indicador de reativação, cuja finalidade será descrita quando de sua utilização na descrição do algoritmo.

Um indicador é posicionado para indicar qual dos elos é para ser processado.

Inicialmente, este indicador conterá a informação de processamento do elo de execução.

Refira-se a figura III-7.

ELO DE EXECUÇÃO

Durante esta fase do processamento, a estrutura de dados é percorrida em lotes correspondentes aos conjuntos numéricos representativos de cada elemento a ser simulado, e verifica-se o conteúdo do contador. Se o valor desse contador é positivo, nada deverá ser alterado na estrutura para o elemento em estudo, e com a consulta ao ponteiro indicador do elemento seguinte, este deverá ser analisado.

Se o valor do contador for negativo ou nulo, isso significará uma ordem de execução do elemento ou seja sua simulação e alteração dos valores numéricos da estrutura a ele relacionados. Após a execução, o contador será substituido por um valor positivo qualquer e o próximo conjunto de dados será analisado após a consulta ao ponteiro conveniente.

Chegando-se ao fim da estrutura de dados, o indicador que estava posicionado para comandar o elo de execução terá seu conteúdo indicando agora o elo de ativação e esse será agora processado.

ELO DE ATIVAÇÃO

Durante esta fase, novamente a estrutura séra percorrida pelo simulador. Para cada conjunto numérico é agora efetuada a análise do semáforo.

Dessa análise, resultam duas alternativas possíveis:

o conteúdo do semáforo é zero

Neste caso o algoritmo deve verificar se o elemento em estudo deverá ser ativado. Esta verificação é efetuada através do estudo dos estados dos sinais de entrada do elemento e do seu valor de saída. Se houver em decorrência, uma necessidade de mudança na saída do elemento, o mesmo deve ser ativado. Isso significa colocar-se no contador o valor do atraso a ser descontado e no semáforo o valor igual a 1 (ativado). Não havendo causa que justifique a ativação, do semáforo, este permanecerá com seu valor igual a 0.

o conteúdo do semáforo é zero

Neste caso, apesar do elemento estar ativado por alguma causa anterior (por exemplo, já descontando o contador) deve-se verificar se o mesmo é reativável, o que é possível através da consulta ao indicador de reativação. Se for reativável, deve-se verificar se o estado atual das entradas provoca a variação no valor da saída do elemento sendo simulado. Se isso realmente acontece, o elemento deve ter seu contador novamente reposicionado com o valor do atraso correspondente ao caso em questão. Caso contrário, o processamento deve prosseguir sem alteração no conteúdo do contador. Se o elemento não é reativável, deve-se prosseguir o processamento, sem alteração no conteúdo do contador permanecendo este com o valor e que deverá ser descontado no momento oportuno.

Em ambos os casos, quer o semáforo contenha ou não zero, o processamento deve prosseguir com a análise do próximo elemento da estrutura.

Durante a fase de teste do semáforo, deve-se descontar uma unidade de atraso do valor existente no contador, para os elementos

que estiverem ativados.

O elo de ativação termina quando a consulta à estrutura indicar seu final; neste caso, o algoritmo deve fornecer as informações correspondnetes à situação dos elementos simulados e atualizar o tempo de simulação.

A simulação, se necessário prosseguir, continuará com o elo de execução, pois o indicador correspondente a esta atividade foi posicionado após o término do elo de ativação.

Ativa-se o algoritmo descrito, através de dados obtidos como se segue:

a) Montar a máscara dos sinais de entrada.

b) Pesquisa-se esta máscara na tabela da verdade da seguinte maneira:

b1 - Para cada linha da matriz, substituir-se as entradas irrelevantes que constam da tabela pelas entradas correspondentes na máscara atual montada em a , obtendo-se uma nova máscara.

b2 - Compara-se a máscara resultante com a máscara atual.

b3 - Se não houver coincidência repete-se o processo para as próximas linhas.

b4 - Quando houver coincidência achou-se a máscara.

c) Constroe-se um conjunto contendo todos os sinais de entrada cujos valores sofreram modificações.

d) Procura-se nas tabelas de atrasos os valores dos atrasos correspondentes a estas entradas e aos valores de saída do elemento alterado, obtendo-se assim um conjunto de atrasos possíveis.

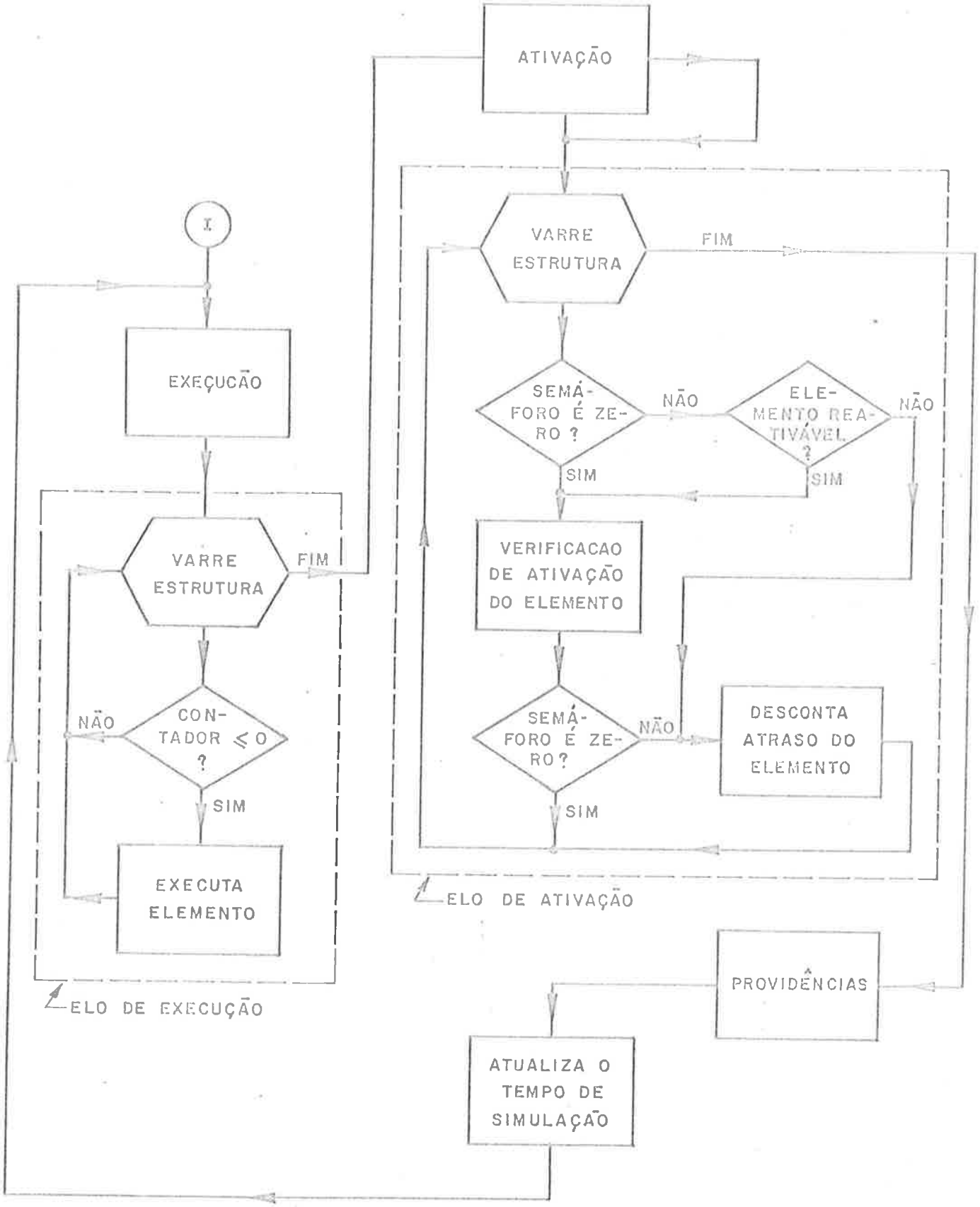

ATIVAÇÃO
VARRE ESTRUTURA
FIM
SEMÁ-FORO É ZE-RO ?
NÃO
ELE-MENTO REA-TIVÁVEL ?
NÃO
SIM
SIM
VERIFICACAO DE ATIVAÇÃO DO ELEMENTO
SEMÁ-FORO É ZE-RO?
NÃO
DESCONTA ATRASO DO ELEMENTO
SIM
ELO DE ATIVAÇÃO
I
EXEÇUCÃO
VARRE ESTRUTURA
FIM
CON-TADOR ≤ 0 ?
NÃO
SIM
EXECUTA ELEMENTO
ELO DE EXECUÇÃO
PROVIDÊNCIAS
ATUALIZA O TEMPO DE SIMULAÇÃO

FIG. III - 7

e) Escolhe-se entre os atrasos constituintes do conjunto obtido em d o de maior valor, o qual define o atraso do elemento nesta situação.

f) Inicia-se o valor da variável de contabilização de atraso do elemento com o valor obtido em c.

g) Com os resultados obtidos em b e f, passa-se à execução do agoritmo anteriormente descrito.

## III-4 PARTIÇÃO

A fase de Partição deverá distribuir os componentes em placas e para cada uma estabelecer quais são as interligações.

Para tanto as informações necessárias já se encontram no arquivo mestre da Descrição Lógica, necessitando-se apenas empregar-se o algoritmo para efetuar a partição.

O critério adotado por este algoritmo deverá ser o de particionar a planta lógica em um número de placas mínimo para acomodar os componentes e respectivas interligações.

Algumas informações entretanto podem ser deixadas a critério do projetista, pois poderá assim interagir com o programa solicitando que a partição seja feita levando-se em consideração que:

- certos sinais devem comparecer em pinos prefixados de conectores;

- certos componentes devem estar alocados em determinadas placas;

- certos componentes singelos devem ser agrupados a seu critério;

- o número de placas é determinado;

- a dimensão física da placa não é a padrão;

- a porcentagem de ocupação dos pinos de conectores é dada;

- o tipo de célula que é o padrão do reticulado;

- etc.

Não havendo essa interação, o algoritmo admitirá as condições padrões através do qual foi desenvolvido.

Para cada bloco da Descrição Lógica a fase de partição gerará tantos blocos quanto o forem o número de placas obtidas ou fornecidas na partição.

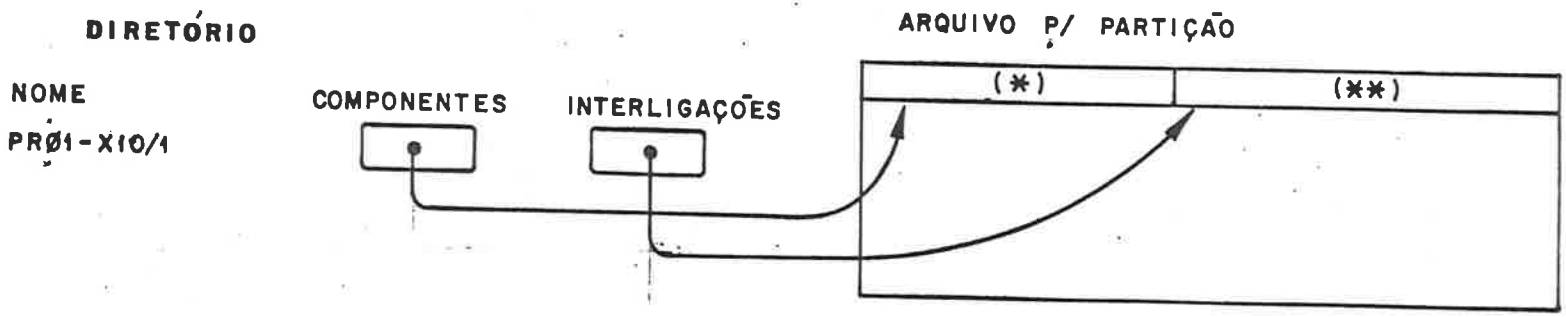
Cada bloco receberá um nome que será constituido do mesmo nome original declarado na descrição lógica e acrescido de um número que o relacionará à placa em questão. Esses nomes aparecem no Diretório da fase de Partição.

Dentro de cada bloco gerado ter-se-á disponível as informações dos componentes que compõe a placa e das respectivas interligações. Para tanto, cada nome de bloco terá dois ponteiros que correspondem à localização do bloco dos componentes e das interligações.

Cada interligação receberá um código através do qual será identificado dentro do bloco correspondente.

A figura III-8 explica melhor a estrutura adotada.

Mostra-se uma placa com reticulado solicitado pelo projetista, na fig. III-9.




APÓS PARTIÇÃO:

O CASO CORRESPONDENTE A TER-SE APENAS UMA PLACA.

(*)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| CØØ1 | 7474 | | | CØØ2 | AMP-M-5Ø | | |
| CØØ3 | 74Ø2 | | | | | | |

(**)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SØØ1 | NRESET | PØØ1 | CØØ1 | PØ13 | CØØ1 | PØØ1 | CØØ2 | SØØ2 | NQØ |
| PØØ2 | CØØ1 | PØØ6 | CØØ1 | PØ11 | CØØ1 | PØØ2 | CØØ3 | SØØ3 | PCLOCK |
| PØØ3 | CØØ1 | PØØ2 | CØØ2 | ... | | | | | |

Figura III.8

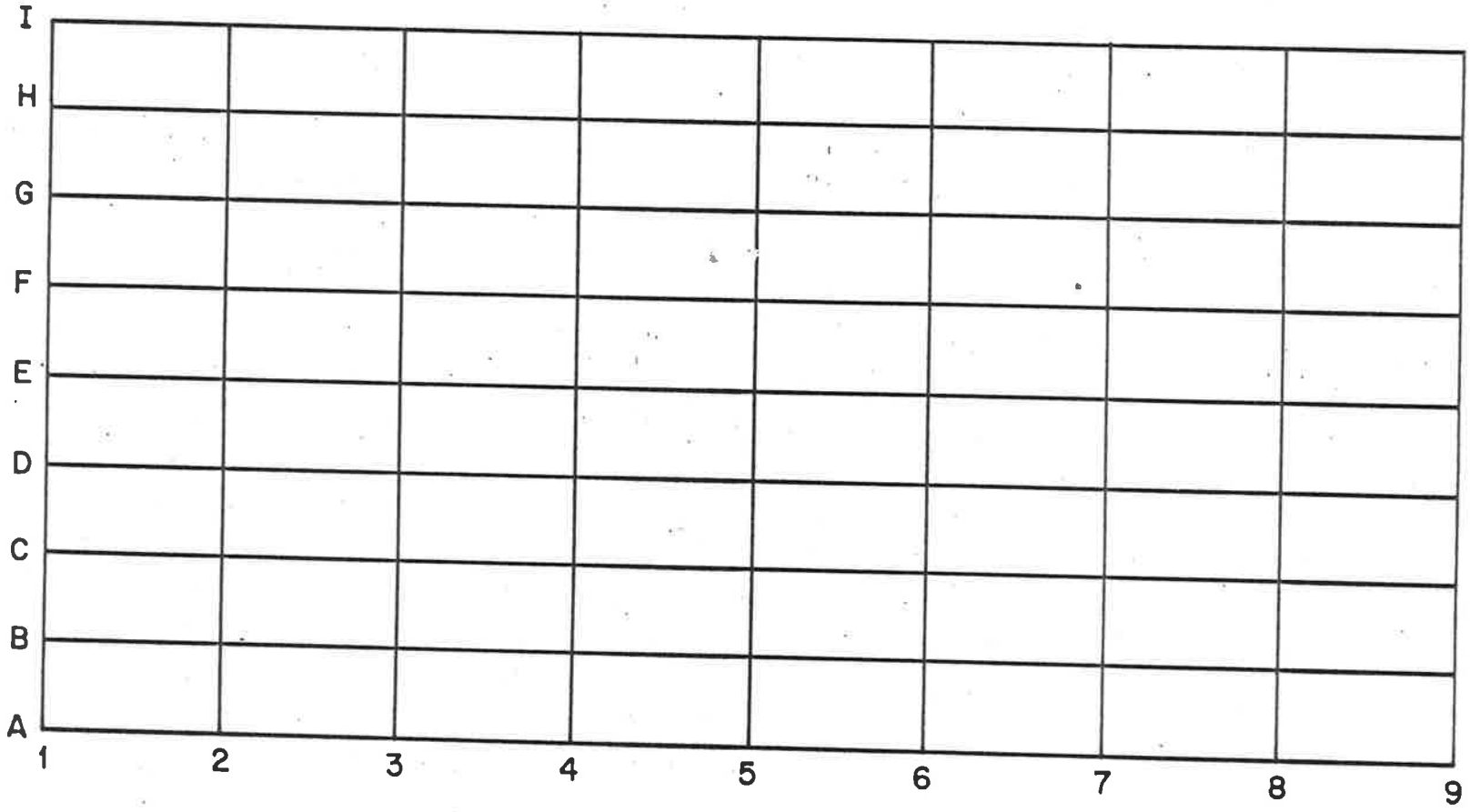



Figura III.9

## III-5 POSICIONAMENTO

A fase de Partição, encarregou-se de separar os elementos constituintes da planta lógica em componentes e interligações.

Cumpre agora à fase de Posicionamento, fornecer a localização exata desses componentes nas placas em que foram designadas.

As informações necessárias a esta fase, já se encontram disponíveis nos arquivos do sistema, a menos que o projetista deseje fixar, como condicionantes, a posição de determinados componentes e respectivos padrões.

Se o projetista está utilizando o Sistema Integrado de Automação e deseja apenas posicionar os componentes, sem ter passado pela fase de Partição, deverá, é claro, fornecer ao Sistema outras informações tais como a dimensão de placa, célula básica, etc, informações essas que deveriam entrar na Partição.

O algoritmo em que se baseia o posicionamento, trabalha levando-se em consideração a conectividade das ligações entre componentes, o espaço existente na placa, e através desses fatos,poderá determinar a localização do mesmo e qual o padrão a ser escolhido, entre àqueles disponíveis na Biblioteca do Sistema.

Após efetuada a escolha do local onde deve ficar o componente dentro da placa e do padrão que obedecerá, essas informações deverão ser inseridas para o componente no bloco que identifica a placa, utilizando-se o ponteiro correspondente à localização dos componentes para esta placa.

Antes entretanto de efetuar o posicionamento, o projetista deverá resolver a partição dos componentes singelos que por ventura deixou a seu encargo.

A estrutura de dados em que o Posicionamento baseia-se é o mesmo da fig. III-8.

Mostra-se uma placa com reticulado, solicitado pelo projetista, e o posicionamento final dos componentes, na figura IV.6.

Mostra-se o esquema do arquivo com posicionamento da figura IV.6 na figura III.10.

## III-6 FIAÇÃO

Para esta fase também já se dispõe de todos os dados necessários. A estrutura na qual se baseia é a mesma do Posicionamento, ou seja, nos blocos gerados pela Partição, que contém os componentes e as interligações.

Para se obter uma saída adequada de dados a ser utilizada pela fiação propriamente dita (pelo técnico), as coordenadas absolutas dos componentes na placa, já disponíveis através do posicionamento, deverão ser transformadas em posições de localização em termos de células, a exemplo do reticulado em células mostrado na figura III-8.

Algumas informações adicionais, para a Fiação, dizem respeito:

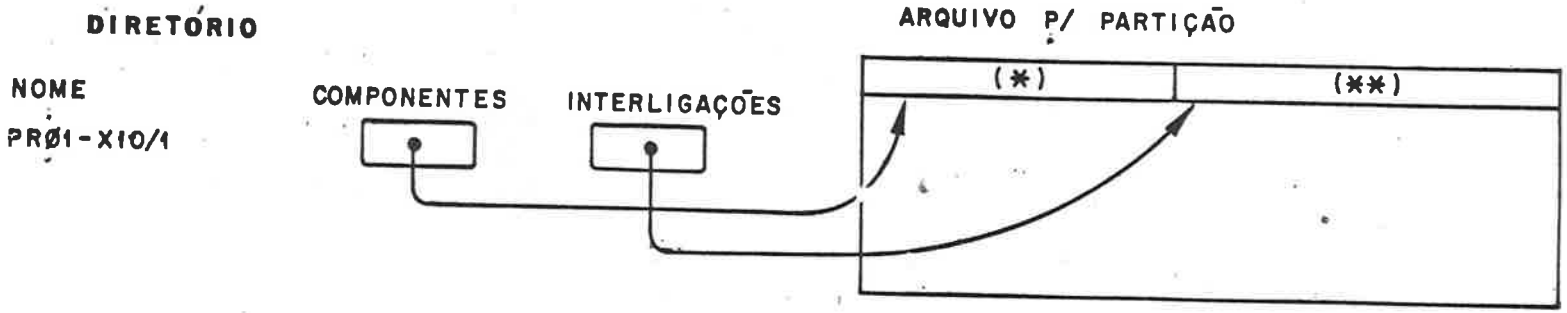
- ao número de níveis de ligações;
- à cor do fio para cada nível;
- ao número do fio a ser utilizado;

informações essas a serem fornecidas pelo projetista.

O algoritmo da Fiação deverá realizar as ligações minimizando o comprimento das mesmas.

## III-7 ROTAS

Para o estabelecimento das rotas de circuito impresso as informações necessárias encontram-se disponíveis nos arquivos do Sistema desde que o projetista se disponha a utilizar as informações




APÓS PARTIÇÃO E POSICIONAMENTO: O CASO CORRESPONDENTE A TER-SE APENAS UMA PLACA.

(*)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| CØØ1 | 7474 | 2.6Ø 1.5Ø | I14N | CØØ2 | AMP-M-5Ø | ØØØ 1.5Ø | CM 5Ø |
| CØØ3 | 74Ø2 | 2.6Ø 6.5Ø | I14N | | | | |

(**)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SØØ1 | NRESET | PØØ1 | CØØ1 | PØ13 | CØØ1 | PØØ1 | CØØ2 | SØØ2 | NQØ |
| PØØ2 | CØØ1 | PØØ6 | CØØ1 | PØ11 | CØØ1 | PØØ2 | CØØ3 | SØØ3 | PCLOCK |
| PØØ3 | CØØ1 | PØØ2 | CØØ2 | | | | | | |

...

Figura III.10

manipuladas pelo algoritmo.

Um dado básico para a fase de Rotas é a disponibilidade dos padrões de ocupação e componentes localizados no arquivo de Padrões.

Havendo necessidade de se fornecer dados a cerca de:

- número de camadas;

- tipo de ligação preferencial por face;

- ligações especiais ou prioritárias;

o projetista deverá fornecê-las através de um meio adequado que dispõe, ou seja, uma linguagem de entrada.

Outras informações necessárias dizem respeito ao controle dos documentos a serem produzidos.

Os algoritmos utilizados na fase de Rotas são compostos de vários métodos entre os quais citam-se: o método de "Lee" e das raias.

Por serem bastante complexos não se relatam aqui os algoritmos, cita-se apenas a referência [11] onde podem ser localizados.

A fase de Rotas deve montar uma estrutura interna que permita a elaboração das rotas e dos mapas de interconexão, possibilitando seu tratamento por rotinas gráficas.

Dessas mesmas estruturas internas resultarão dados para uma fita de furação. Esses dados deverão ser adequados aos códigos da máquina de controle numérico a ser utilizada.

## III-8 DOCUMENTAÇÃO

Os dados de projeto que serão necessários documentar, encontram-se disponíveis em todos os arquivos que constituem o Sistema de Automação, desde que tenham sido gerados, nas fases anteriores.

A fase de Documentação deverá manipular esses dados e formatá-los de acordo com a solicitação do projetista.

Para cada solicitação reserva-se um tipo de relatório, cujos elementos constituintes encontram-se nos arquivos do Sistema.

# CAPÍTULO IV

REFERÊNCIAS: [6] [11] [12] [13] [15]

De acordo com o estabelecido no Capítulo II, o Sistema Integrado de Automação deve ser gerenciado por um Supervisor que aceite comandos do projetista, os interprete e tome para si as ações. Estabelece-se agora quais devem ser esses comandos, tendo-se em vista que a linguagem deverá ser a mais simples possível.

A linguagem adotada terá como condicionante ser a mais próxima possível daquela do próprio projetista, evitando-se assim particularizações e regras que poderão torná-la inviável de utilização.

Os comandos estabelecidos devem observar, entretanto, algumas regras no que tange a sua forma, disposição de seus elementos constituintes e finalização, sem limitar a liberdade de sua utilização.

Efetua-se, a seguir, o esboço de uma linguagem para comandar o sistema.

## IV-1 OS COMANDOS DO SUPERVISOR

O Supervisor deverá ter comandos específicos que permitam a escolha da fase do Sistema que se deseja utilizar. Como esses comandos tratam com os principais elementos constituintes do Sistema, resolveu-se padronizá-los, de forma a distinguí-los dos demais.

Toda vez que um comando deste tipo surgir, o Supervisor decidirá qual fase do Sistema deve ser executada, providenciará a leitura das informações, criação de blocos de dados dentro de arquivos, aceitará condições de contorno para a fase, decidirá se os dados devem ser editados, em fim, procederá a aquisição de informações sobre a ação de outros comandos específicos de cada fase.

Após a interpretação correta desses comandos e colocação dos mesmos na ordem em que eles realmente devem entrar, através de um processo de ordenação hierárquica, o supervisor passará o comando do sistema para a fase selecionada que se encarregará agora do processamento de todas as informações obtidas, executando, assim, a fase selecionada do Sistema Integrado de Automação.

Após a execução de cada fase o controle do sistema deverá retornar ao supervisor.

Os comandos que serão interpretados pelo supervisor selecionarão a seqüência de diretrizes a serem tomadas e serão identificados por uma marca especial que é um # antes do comando.

O formato desses comandos é livre. Devem ter a marca especial e a palavra, ou conjunto de palavras, que o compõe será reservada para uso do próprio sistema.

Todas as palavras que se seguem, e que são consideradas como fornecidas ao Supervisor do Sistema de Automação, não são acentuadas, uma vez que, os periféricos, através dos quais serão fornecidas, não possuem acentuação. A tabela IV-1 representa os comandos de controle do Supervisor.

#INICIO
#CARREGAR
#DESCARREGAR
#EDITAR nome do bloco
#BIBLIOTECA
#DESCRICAO LOGICA
#SIMULACAO
#PARTICAO
#POSICIONAMENTO
#FIACAO
#ROTAS
#DOCUMENTACAO
#FIM

Tabela IV.1

A função de cada comando após sua interpretação pelo Sisma é a seguir relatada:

# INICIO

Indicará para o Supervisor que todos os ponteiros e tabelas necessárias ao desenvolvimento de um projeto pelo Sistema de Automação deverão ter seus valores posicionados nas suas condições iniciais. Significará, assim, uma limpeza total do sistema.

# CARREGAR

Indicará ao Supervisor que o projetista deseja colocar nos arquivos do Sistema, o conteúdo a ser lido de uma unidade armazenadora de dados (fita magnética).

O projetista antes deste comando deverá fornecer o comando # INICIO.

# DESCARREGAR

Indicará ao Supervisor que o projetista deseja copiar todos os arquivos existentes no Sistema, em unidade de armazenamento de dados (fita magnética).

# EDITAR nome do bloco

A interpretação deste comando implicará na chamada de um programa de edição de dados, que atuará no bloco selecionado, eliminando, inserindo ou substituindo dados nele existentes. O novo bloco obtido poderá ser colocado na mesma posição que ocupava inicialmente, se houver disponibilidade de área para tal, ou será colocado na próxima área disponível dentro do arquivo-mestre que o comportava.

Toda vez que o Editor for chamado e ocorrer modificação de dados dentro do bloco, o supervisor deverá executar uma rotina interna para delir blocos subseqüentes que dependam, para sua subexistência, do bloco corrigido.

Uma vez sob ação do Editor, os comandos para correção do bloco serão os mostrados na tabela IV.2.

| | |
|---|---|
| /D, n1 {,n2} | eliminar linhas do bloco entre n1 e n2 (inclusive) |
| /R, n1 {,n2} | substituir linhas do bloco entre n1 e n2 |
| /I, n1 | inserir linha(s) após n1 |
| /E | encerrar correções. |

Tabela IV.2

## # BIBLIOTECA

Sob a ação deste comando, o supervisor deverá selecionar a fase de Biblioteca. Esta atuará sobre os arquivos que a constitui efetuando a geração ou eliminação de elementos presente nos arquivos, bem como permitindo-se efetuar a listagem do conteúdo de um determinado elemento ou nomes dos mesmos nela presentes.

Os sub-comandos que pertencem ao # BIBLIOTECA são os que se mostram na tabela IV.3 e que permitem trabalhar sobre três diferentes membros da Biblioteca: funções de componentes, componentes propriamente ditos e padrões de componentes ou de ocupação.

* GERAR FUNCAO
* GERAR COMPONENTE
* GERAR PADRAO
* LISTAR FUNCAO
* LISTAR COMPONENTE
* LISTAR PADRAO
* DELIR FUNCAO
* DELIR COMPONENTE
* DELIR PADRAO
* DIRETORIO
* FIM

Tabela IV.3

Os sub-comandos de geração (os do tipo * GERAR) possuem algumas palavras reservadas que permitem o fornecimento de informações. Para cada sub-comando as palavras chave ou operadores, que a seguir se apresentam, podem aparecer em qualquer ordem e alguns são opcionais, devendo comparecer quando o componente ou função o exigir.

```
* GERAR FUNCAO
  & comentario
  NOME =
  ENTRADAS =
  SAIDAS =
  FANIN =
  FANIN LISTA = (,),(,), .../
```

FANOUT =
FANOUT LISTA = (,),(,), .../
FUNCAO (m,n) =
ATRASO NORMAL (n) =
ATRASO MAXIMO (n) =
ATRASO MINIMO (n) =

* GERAR COMPONENTE
& comentario
NOME =
NUMERO DE ELEMENTOS =
TEMPO SETUP =
TEMPO HOLD =
LARGURA CLOCK =
LARGURA SET =
LARGURA RESET =
PINAGEM =
DIMENSAO = (,)
DISSIPACAO =
PADRAO =

* GERAR PADRAO
& comentario
OCUPACAO
TIPO =
TERMINAL BASICO =
NUMERO DE CELULAS =
CELULA (tipo) = $(x_1,y_1),(x_2,y_2)$, ... / $\left\{ \begin{matrix} U \\ L \end{matrix} \right\}$

* GERAR PADRAO
& comentario
COMPONENTE
TIPO =
TERMINAL BASICO =
NUMERO DE TERMINAIS =

ISOLACAO =

COORDENADAS LOCAIS = (,)

INCREMENTO = (,)

$\left\{\begin{matrix} U \\ L \\ F \end{matrix}\right\}$

Analisando esses sub-comandos tem-se a observar:

1. O nome de uma função ou de um arquivo tem o seu número de caracteres limitado. Esse número de caracteres deverá ser no máximo 15.

O nome da função é fornecido através do operador:

NOME =

2. Os sinais de entradas e saídas do elemento são designados por nomes simbólicos separados por vírgula, através dos operadores:

ENTRADAS =

SAIDAS =

3. As cargas normalizadoras para as entradas e saídas ("FANIN", "FANOUT") são fornecidas através dos operadores:

FANIN =

FANOUT =

desde que tenham as mesmas unidades para todas as entradas e saídas.

Havendo distinção entre cargas normalizadoras para entradas e/ou saídas, fornece-se a informação através de uma lista con

tendo o nome simbólico dos sinais e respectivas cargas, pelos operadores:

FANIN LISTA = (,),(,), .../
FANOUT LISTA = (,),(,), .../

4. A tabela da verdade que descreve o funcionamento do elemento é fornecida através do operador:

FUNCAO (m,n) = $d_1$, $d_2$, $d_3$ ...

onde: m = número de elementos-linha corresponentes às possíveis situações dos sinais de controle;

n = número de sinais de controle, portanto igual ao número de colunas.

5. Os atrasos de transição de estado são fornecidos através dos operadores do tipo ATRASO. Na falta de um ou mais informações sobre os atrasos a considerar, o sistema preenche as informações faltantes com os dados dos atrasos que foram fornecidos. É obrigatória, entretanto, a presença de pelo menos um operador ATRASO.

O operador ATRASO tem a forma:

$$\text{ATRASO} \begin{Bmatrix} \text{MINIMO} \\ \text{NORMAL} \\ \text{MAXIMO} \end{Bmatrix} (n) = d_1, d_2, \ldots$$

onde: n = número de sinais de entrada para o elemento, sinais esses que implicam no atraso.

6. O número de elementos constituintes de um componente é fornecido através do operador:

NUMERO DE ELEMENTOS = numero

Entende-se como número de elementos de um componente, por exemplo, o número de "gates" singelos que o constitui.

7. O posicionamento físico dos sinais no componente é fornecido através do operador:

PINAGEM = numero

onde número é o total de pinos que o constitui.

Segue-se após esse operador, a descrição dos pinos de saída, relacionados através de uma função dos pinos de entrada.

Assim:

PINAGEM = numero
pino 1 = função (pino 2, pino 3)
.
.
.
.
.
.
numero = ~~b~~ pino VCC b/ pino TERRA.

8. Os operadores LARGURA e TEMPO são opcionais. São utilizados toda vez que o projetista necessite fornecer informações sobre tempos de sinais "Set-up","Hold" e larguras do "CLOCK", "SET" e "RESET".

A forma desses operadores sãoas seguintes:

$$\text{TEMPO} \begin{Bmatrix} \text{SETUP} \\ \text{HOLD} \end{Bmatrix} = \text{numero}$$

$$\text{LARGURA} \begin{Bmatrix} \text{CLOCK} \\ \text{SET} \\ \text{RESET} \end{Bmatrix} = \text{numero}$$

Esses operadores podem ou não comparecer, dependendo do componente.

9. A dimensão física do componente, dado necessário para dispô-lo numa placa de circuito impresso, é fornecida através do operador:

DIMENSAO = (x,y)

onde x e y são as dimensões em centímetros.

10. A dissipação do componente é fornecida através do operador:

DISSIPACAO = numero

onde número é dado em mW.

11. Os padrões deste componente, existentes no arquivo de padrões, são explicitados, através de uma lista, da forma seguinte:

PADRAO = codigo 1, codigo 2, codigo 3, .../

onde: codigo 1, ..., correspondem aos padrões que o projetista poderá utilizar.

12. Os operadores relacionados ao sub-comando *GERAR PADRAO, apesar de colocados em forma mais adequada, obedecem a descrição que pode ser conhecida através da referência [11] deste trabalho e são utilizados para geração de padrões para circuito impresso.

13. Comentários podem ser colocados a qualquer instante, desde que tenham um "&" antes dos mesmos.

Os sub-comandos do tipo * LISTAR permitem efetuar a impressão das funções, dos componentes ou dos padrões relacionados em lista que segue os sub-comandos.

Assim, a forma desse sub-comando será:

$$* \text{ LISTAR} \left\{ \begin{array}{l} \text{FUNCAO} \\ \text{COMPONENTE} \\ \text{PADRAO} \end{array} \right\}$$

$\text{nome}_1$, $\text{nome}_2$, $\text{nome}_n$

O sub-comando * DIRETORIO permitirá efetuar a impressão dos nomes dos elementos presentes em cada arquivo da Biblioteca.

O sub-comando * FIM encerra a descrição de cada elemento a ser gerado.

Para esta fase do sistema, ter-se-á três arquivos, cada um deles indexado por um ponteiro no Diretório respectivo, associado a um nome que identificará o bloco no arquivo. A figura IV.1 é o esquema adotado para a BIBLIOTECA do sistema.

DIRETÓRIO FUNÇÃO

| nome | ponteiro para localização do bloco |
|---|---|
| 7402NOR | [ • ] |
| 74153MUX | [ • ] |
| . | |
| . | |
| . | |

DIRETÓRIO COMPONENTE

| nome | ponteiro para localização do bloco |
|---|---|
| 7402 | [ • ] |
| 74153 | [ • ] |

DIRETÓRIO PADRÃO

| nome | ponteiro para localização do bloco |
|---|---|
| IMET | [ • ] |
| I16N | [ • ] |

Figura IV.1

EXEMPLOS:

a) No caso do 7402 a geração de arquivos é obtida com:

```
#  BIBLIOTECA
*  GERAR FUNCAO
   NOME = 74Ø2 NOR
   ENTRADAS = X1, X2
   SAIDAS = X
   FANIN = 1
   FANOUT = 1Ø
   FUNCAO (3,3) = Ø, Ø, 1
                  1, X, Ø
                  X, 1, Ø
   &
   & O SISTEMA ADOTARA ATRASOS MAXIMOS E
   & MINIMOS IGUAIS AO NORMAL.
   &
   ATRASO NORMAL(2) = 12,12
                      8,8
*  FIM

*  GERAR COMPONENTE
   NOME = 74Ø2
   NUMERO DE ELEMENTOS = 4
   PINAGEM = 14
             Ø1 = 74Ø2NOR(Ø2,Ø3)
             Ø4 = 74Ø2NOR(Ø5,Ø6)
             1Ø = 74Ø2NOR(Ø9,Ø8)
             13 = 74Ø2NOR(12,11)
             14 = Ø/14Ø/Ø8
   DIMENSAO = (0.80, 2.40)
   DISSIPACAO = 135
   PADRAO = I14N, I14V/
*  FIM
#  FIM
```

b) Para o 7474, tem-se:

```
#  BIBLIOTECA
*  GERAR FUNCAO
   NOME = 7474FF(Q)
   ENTRADAS = # SB, # RB, # C, # D
   SAIDAS = Q, QB
   FANIN LISTA =(#SB,2), (# RB,3), (#C,2), (# D,1)/
   &
   &
   FANOUT = 1Ø
   FUNCAO (5,6)= Ø, Ø, X, X, E, E
                 Ø, 1, X, X, 1, Ø
                 1, Ø, X, X, Ø, 1
                 1, 1, U, Ø, Ø, 1
                 1, 1, U, 1, 1, Ø

   ATRASO MINIMO (3) = 25, 25, 1Ø
                       4Ø, 4Ø, 1Ø

   ATRASO NORMAL (3) = 25, 25, 14
                       3Ø, 4Ø, 2Ø

   ATRASO MAXIMO (3) = 25, 25, 25
                       4Ø, 4Ø, 4Ø
* FIM
```

```
*  GERAR COMPONENTE
   NOME = 7474
   NUMERO DE ELEMENTOS = 2
   TEMPO SETUP =  2Ø
   TEMPO HOLD =   5
   LARGURA CLOCK =  3Ø
   LARGURA SET =  3Ø
   LARGURA RESET =   3Ø
   PINAGEM = 14
             Ø5 =  7474FF(Q) (15, 16, 17, 18)
             Ø6 = 7474FF(QB) (15, 16, 17, 18)
             Ø8 = 7474FF(QB) (19, 2Ø, 21, 22)
             Ø9 = 7474FF(Q) (19, 2Ø, 21, 22)
             15 = # DEL (Ø4)
             16 = # DEL (Ø1)
             17 = # DEL (Ø3)
             18 = # TSUTH (Ø2, Ø3)
             19 = # DEL (1Ø)
             2Ø = # DEL (13)
             21 = # DEL (11)
             22 = # TSUTH (12, 11)
             14 =  ¥ 14¥Ø7
   DIMENSAO = (0.80, 2.40)
   DISSIPACAO = 250
   PADRAO = I14N, I14V/
*  FIM
#  FIM
```

c) Para o 7474 são necessárias as funções:

```
#  BIBLIOTECA
*  GERAR FUNCAO
   NOME = # TSUTH
   ENTRADAS = X, Y
   SAIDAS = Z
   FANIN = 1
   FANOUT = 99
   FUNCAO (2,3) = Ø, X, Ø
                  1, X, 1
*  FIM
```

```
*  GERAR FUNCAO
   NOME = # DEL
   ENTRADAS = X
   SAIDAS = Y
   FANIN = 1
   FANOUT = 99
   FUNCAO (2,2) = 1,1
                  Ø,Ø
*  FIM
#  FIM
```

## # DESCRIÇÃO LOGICA

A interpretação deste comando implicará na execução por parte do supervisor de uma série de programas que analisarão os sub-comandos subseqüentes, preparando os dados para a fase de Descrição Lógica do trabalho do projetista.

Os sub-comandos pertencentes a esta fase são os que se apresentam na tabela IV.4.

```
*  INDENTIFICACAO
*  COMPONENTES SINGELOS
*  COMPONENTES COMPLEXOS
*  ENTRADAS PRIMARIAS
*  SAIDAS PRIMARIAS
*  TESTAR CARREGAMENTOS
*  SEPARAR LIGACOES
*  SEPARAR COMPONENTES
*  DIRETORIO
*  AUTOMATIZAR
*  LISTAR nome do bloco
*  DELIR nome do bloco
*  FIM
```

Tabela IV.4

Passa-se, a seguir, a analisar cada sub-comando apresentado na tabela IV.4:

1. * IDENTIFICACAO

Este sub-comando permite a entrada de informações correspondentes ao arquivo relacionado com o projeto em andamento.

Consta dos seguintes operadores:

a) CONTROLE = codigo

O código contido em controle será armazenado junto com as informações iniciais contidas nos operadores pertencentes ao sub-comando de * IDENTIFICACAO, se o bloco referenciado em NOME estiver sendo gerado pela primeira vez.

Caso contrário, será efetuado o confronto entre os dois códigos e havendo coincidência corresponderá a uma atualização do bloco em questão.

b) NOME =

Através do qual fornece-se a indentificação do bloco a ser processado ou em processamento, dependendo da informação contida no operador CONTROLE.

c) DATA = n1/n2/n3

Permite o fornecimento da data de geração do bloco cujo nome foi referenciado pelo operador NOME.

d) DESCRICAO =⟨livre⟩

Permite fornecer dados de descrição a respeito ao bloco que se está gerando.

e) ATUALIZAÇÃO = $\left\{\begin{matrix}\text{INICIO} \\ \text{n1/n2/n3}\end{matrix}\right\}$

Permite entrar com a palavra INICIO ou com a data de atualização do bloco referenciado no operador NOME, dependendo da variável presente no operador CONTROLE.

2. * COMPONENTES SINGELOS

Este sub-comando informará ao sistema que as informações subseqüentes serão relacionadas com sinais de entrada e saída de componentes singelos, incluindo o tipo de função que os relaciona.

As informações deverão ser fornecidas obedecendo a forma seguinte:

Sinal de saída/tipo de função/lista de sinais de entrada separados por vírgulas /
& comentário

3. * COMPONENTES COMPLEXOS

Este sub-comando deverá vir após o sub-comando anterior se o mesmo existir.

Este sub-comando permitirá a descrição dos componentes com funções específicas. Relacionará o número do componente no projeto lógico, com o número de componente fornecido pelo fabricante, sua função e os sinais associados a cada pino.

Havendo necessidade do projetista definir um componente singelo com um número de seu projeto lógico, ele o fará, mas efetuará sua descrição como se fosse um COMPONENTE COMPLEXO.

As informações serão fornecidas obedecendo ao seguinte formato:

num. do componente/nº do fabricante - função / (pinos, sinais), (pinos, sinais), .../
& comentário

No caso em que o componente é um conector as palavras VCC e TERRA são reservadas e indicam a que pinos esses sinais estão ligados.

4. * ENTRADAS PRIMARIAS
   * SAIDAS PRIMARIAS

Descrevem sinais fornecidos externamente ao projeto em questão, ou sinais de saídas que serão utilizados em outros dispositivos não pertencentes ao projeto em análise.

A forma de se transmitir esses sinais será através de uma lista de nomes, separados por vírgulas, logo após o sub-comando.

{ * ENTRADAS PRIMARIAS
  * SAIDAS PRIMARIAS }

$nome_1$, $nome_2$, ... $nome_n$

5. * TESTAR CARREGAMENTOS

Este sub-comando informará ao sistema que o projetista deseja efetuar testes com as cargas normalizadas a fim de detetar possíveis problemas de projeto. Além do teste de cargas, será efetuada a verificação dos possíveis sinais referenciados e não gerados na descrição do projeto lógico.

6. { * SEPARAR LIGACOES
* SEPARAR COMPONENTES }

Este sub-comando efetuará a separação das informações contidas no bloco que descreve a lógica do projeto em outro bloco distinto, com nome acrescido de um índice numérico e associado a dois ponteiros distintos e que deverá ser utilizado pelas fases de Rotas, Pocisionamento e Fiação, que conterá os componentes e as ligações entre os mesmos. Visa facilitar a utilização de fases em separado.

A descrição desta estrutura é efetuada na análise do comando # PARTICAO.

7. * DIRETORIO

Este sub-comando permitirá ao projetista obter a impressão dos nomes dos blocos que estão presentes no arquivo de Descrição Lógica do Sistema.

8. * AUTOMATIZAR

Através deste sub-comando o projetista definirá que toda a Automação deverá ser controlada pelo próprio Sistema. Apenas informará, em cada fase, os tipos de relatórios ou informações que deseja obter.

9. * LISTAR nome no bloco

Este sub-comando permitirá que o projetista obtenha uma impressão das informações contidas no bloco, cujo nome consta do sub-comando.

10. * DELIR nome no bloco

Este sub-comando permitirá ao projetista eliminar um bloco de informações contido no arquivo de Descrição lógica do Sistema.

O próprio Supervisor encarregar-se-á da eliminação dos demais blocos dependentes do bloco inicial eliminado pelo projetista.

11. * FIM

Este sub-comando encerra a descrição das informações para o projeto lógico, devendo aparecer após os sub-comandos ou - * COMPONENTES SINGELOS ou * COMPONENTES COMPLEXOS ou * ENTRADAS PRIMARIAS ou * SAIDAS PRIMARIAS, dependendo de qual vai encerrar a descrição lógica.

Para esta fase do sistema será estabelecida uma estrutura que consistirá de um arquivo mestre, denominado de arquivo das descrições lógicas, e de um diretório que conterá o nome de cada bloco descrito no arquivo, bem como, o ponteiro que indicará sua localização dentro do arquivo mestre.

A figura IV.2 é o esquema dessa estrutura.

DIRETÓRIO

| nome do bloco | ponteiro para a localização do bloco |
|---|---|
| PRØ1-X1Ø | [ • ] |
| PRØ7-X25 | [ • ] |

Figura IV.2

Exemplo: Para a planta lógica da figura III.5, tem-se:

```
#   DESCRICAO LOGICA
*   IDENTIFICACAO
    CONTROLE = Ø61  (por exemplo)
    NOME = PRØ1-X1Ø
    DATA = Ø6/Ø4/76
    DESCRICAO = CONTADOR BINARIO
    ATUALIZACAO = INICIAL
*   COMPONENTES SINGELOS
          PCARRY/74Ø2NOR/NQØ, NQ1/
&
&   DESCREVEU-SE O SINGELO
&
*   COMPONENTES COMPLEXOS
          CØØ1/7474-FF/(Ø1,NRESET), (Ø2,NQØ), (Ø3,PCLOCK),
          (Ø4,NSET), (Ø5,PQØ), (Ø6,NQØ), (Ø8, NQ1), (Ø9,PQ1),
          (1Ø,NSET), (11,NQØ), (12,NQ1), (13,NRESET)/
&
&   DESCREVEU-SE O 7474
&
    CØØ2/AMP-M-5Ø/(5Ø,VCC), (25,TERRA)/
*   ENTRADAS PRIMARIAS
    NSET, PCLOCK, NRESET

*   SAIDAS PRIMARIAS
    PCARRY, PQ1, PQØ

*   FIM
#   FIM
```

# SIMULACAO

A interpretação deste comando implicará na execução da simulação lógica do projeto descrito pelo projetista.

Os sub-comandos pertencentes a esta fase são os apresentados na tabela IV.5.

* SIMULAR nome do bloco
* LISTAR nome do bloco
* DELIR nome do bloco

Tabela IV.5

A análise de cada sub-comando é a seguir apresentada:

1. * SIMULAR nome do bloco

Este sub-comando permite que o projetista solicite a simulação lógica do projeto descrito no bloco cujo nome é fornecido através do sub-comando.

Integram este sub-comando vários operadores:

a) CONDICOES

Através do qual permite-se o fornecimento de valores iniciais aos sinais listados logo a seguir ao operador, de acordo com a forma abaixo:

$$\begin{pmatrix} \text{nome do} & \text{valor} \\ \text{sinal} & (1 \text{ ou } \emptyset) \end{pmatrix}, \begin{pmatrix} \text{nome do} & \text{valor} \\ \text{sinal} & (1 \text{ ou } \emptyset) \end{pmatrix}, \ldots$$

b) GERACAO DE TESTES

Através deste operador pode-se estabelecer o comportamento de sinal de testes através de descrição na forma abaixo:

nome do sinal / estado inicial / tempo, tempo, .../

{ ALTO
BAIXO }

Aparecendo dois valores para tempo e, em seguida, um P, o sinal será periódico e o primeiro valor do tempo será o período e o segundo, a largura do primeiro pulso dentro do período.

c) TEMPO DE SIMULACAO = numero

Este operador permite fornecer a duração da simulação que o projetista deseja.

Este valor deverá estar em acordo com os atrasos dos elementos utilizados.

d) ATRASO { MINIMO
NORMAL
MAXIMO }

Este operador informará ao simulador o tipo de atraso a se considerar para os elementos lógicos durante a simulação.

e) SAIDA GRAFICA { PLOTTER
DISPLAY
IMPRESSORA } , CONTROLE { NORMAL
nome de um sinal
MUDANÇAS }

Através deste operador o projetista indicará qual o dispositivo periférico a ser utilizado para a saída da simulação e o tipo de controle a ser utilizado para amostrar as formas de onda dos sinais.

Havendo a presença da palavra chave:

NORMAL, o simulador fornecerá a saída dos sinais de forma contínua;

⟨nome de um sinal⟩, o simulador fornecerá a saída toda vez que o sinal de controle, definido através de seu nome, estiver no nível "1";

MUNDANCAS, o simulador fornecerá a saída dos sinais toda vez que houver mudança em um dos sinais de interesse do projetista.

A lista de nomes dos sinais é fornecida após o operador.

Os nomes devem vir separados por vírgulas:

sinal 1, sinal 2, ... sinal n

2. A fase de Simulação utiliza a mesma estrutura de arquivo da fase de Descrição Lógica e executa a simulação através da alocação de áreas próprias para tal e localizadas dentro do próprio programa.

Havendo essa identificação de arquivos com a fase de Descrição Lógica, os sub-comandos de * LISTAR nome no bloco e * DELIR nome no bloco exercem aqui as mesmas funções que os citados na fase de Descrição Lógica.

3. * FIM

Este sub-comando é utilizado para encerrar a descrição de operadores na fase de Simulação.

Exemplo:

Considerando-se o projeto descrito como PRØ1-X1Ø anteriormente tem-se:

```
# SIMULACAO
* SIMULAR PRØ1-X1Ø
  CONDICOES
            (PQØ, Ø), (NQØ,1), (PQ1,1), (NQ1,Ø)

  GERACAO DE TESTES
            NSET/ALTO/1Ø, 2Ø/
            PCLOCK/BAIXO/3Ø, 2Ø, P/
            NRESET/ALTO/3ØØ, 2Ø/
  TEMPO DE SIMULACAO = 5ØØ

  ATRASO NORMAL

  SAIDA GRAFICA PLOTTER, CONTROLE = MUDANCAS
      PCARRY, PQ1, PQØ

* FIM

# FIM
```

# PARTICAO

A interpretação deste comando determinará ao supervisorque a fase a ser executada deverá ser a da Partição.

Os sub-comandos pertencentes ao controle desta fase serão os estabelecidos na tabela IV.6.

* PROCESSAR nome do bloco
* LISTAR nome do bloco
* DELIR nome do bloco
* DIRETORIO
* PARAMETRIZAR
* FIM

Tabela IV.6

Analisa-se, a seguir, cada sub-comando apresentado na tabela IV.6:

1. * PROCESSAR nome do arquivo

Este sub-comando permite ao projetista solicitar a partição lógica do projeto, descrito no bloco cujo nome é fornecido através do sub-comando.

Integram este sub-comando, os operadores:

a) NUMERO DE PLACAS = numero

Através deste operador é fornecido o número de placas que se deseja obter após a partição.

b) DIMENSAO DA PLACA = (x,y)

Este operador determinará, para a fase, o número de componentes possíveis de serem alocados em cada placa. Será o espaço considerado útil da placa. As dimensões de uma placa são fornecidas em centímetros.

c) SINAIS = (nome, pino), (nome, pino) .../

Através deste operador o projetista poderá fornecer os sinais que deseja ter nos pinos declarados do conector.

Cada operador SINAIS deverá dar origem a um novo conector.

d) COMPONENTES = (numero do componente, placa), .../

Se o projetista deseja fixar um determinado componente em uma placa, ele utilizará o operador acima, no qual informará o número de componentes em seu projeto lógico (a numeração dada por ele) e em que placa deseja tê-lo, associando, a cada uma, um número de 1 a n.

e) OCUPACAO = numero

Este operador indicará qual a ocupação dos pinos dos conectores em porcentagem.

f) CELULA BASICA = (,)

Através deste operador fornece-se as dimensões de uma célula do reticulado em que a placa deverá ser dividida para se executar a partição. Os valores dessas dimensões são dados em centímetros.

Não havendo os operadores de a a f, o próprio programa alocará os componentes nas placas e os sinais nos conectores. Neste caso o programa utilizará o número mínimo de placas que julgar necessário e uma dimensão padrão para as placas,

A exemplo das fases de Descrição Lógica e de Simulação, esta fase trabalha sobre o mesmo arquivo e, assim sendo, o projetista poderá listar ou delir blocos dentro do arquivo mestre.

Para tanto ele dispõe, nesta fase, dos sub-comandos:

* DELIR nome do bloco
* LISTAR nome do bloco

que exercem as mesmas funções que as citadas na Descrição Lógica e na Simulação.

2. * DIRETORIO

Este comando permite listar os blocos gerados pela Partição contendo os componentes e interligações.

3. * PARAMETRIZAR

Este sub-comando permite ao projetista deixar a seu critério a alocação de singelos num mesmo componente, ou não, quando da partição.

O resultado da partição designará, para cada singelo, uma parâmetro que será um número, em ordem crescente, para cada função, e que ficará a critério do projetista a incorporação ou não desses singelos num mesmo componente.

4. * FIM

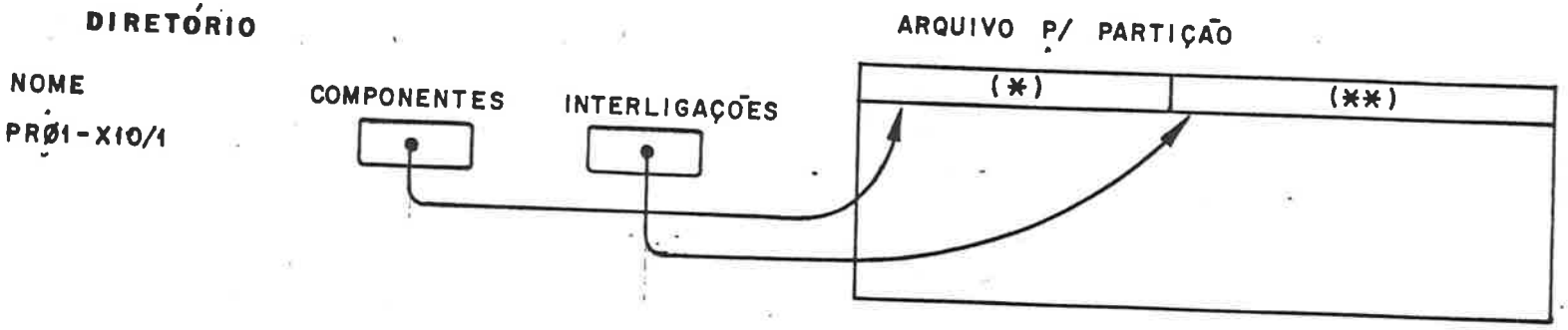
Este operador encerra o fornecimento de condições para a partição.

Ao ser processada a fase de Partição, ela gerará tantos blocos quantos forem necessários, cada um deles correspondendo aos componentes e interligações de cada placa.

Para isso, o nome do bloco utilizado na descrição lógica, e a partir do qual se faz a partição, receberá um índice que varia de 1 a n, onde este índice correponde ao número da placa.

Cada nome de bloco agora receberá dois ponteiros, um deles referenciando os componentes e o outro as interligações. Essas informações encontram-se localizadas no Diretorio da Partição.

A estrutura da figura IV.3 é a representação dos blocos gerados na partição.




APÓS PARTIÇÃO:

O CASO CORRESPONDENTE A TER-SE APENAS UMA PLACA.

(*)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| CØØ1 | 7474 | | | CØØ2 | AMP-M-5Ø | | |
| CØØ3 | 74Ø2 | | | | | | |

(**)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SØØ1 | NRESET | PØØ1 | CØØ1 | PØ13 | CØØ1 | PØØ1 | CØØ2 | SØØ2 | NQØ |
| PØØ2 | CØØ1 | PØØ6 | CØØ1 | PØ11 | CØØ1 | PØØ2 | CØØ3 | SØØ3 | PCLOCK |
| PØØ3 | CØØ1 | PØØ2 | CØØ2 | ... | | | | | |

FIG. IV-3

Exemplo:

Sempre com referência ao projeto descrito como PRØ1-X1Ø:

```
# PARTICAO
* PROCESSAR PRØ1-X1Ø
  NUMERO DE PLACAS = 1
  DIMENSAO DA PLACA = (1Ø, 2Ø)
  CELULA BASICA = (1.25, 2.50)
* FIM
# FIM
```

As informações fornecidas acima levam a um reticulado como mostra a figura IV.4.

O resultado do programa é o seguinte:

Componentes: CØØ1 - 7474
CØØ2 - AMP-M-5Ø
CØØ3 - 7402

Interligações:

| | | | |
|---|---|---|---|
| SØØ1-NRESET | PØØ1 - CØØ1 | PØ13 - CØØ1 | PØØ1 - CØØ2 |
| SØØ2-NQØ | PØØ2 - CØØ1 | PØØ6 - CØØ1 | PØ11 - CØØ1 |
| | PØØ2 - CØØ3 | | |
| SØØ3-PCLOCK | PØØ3 - CØØ1 | PØØ2 - CØØ2 | |
| SØØ4-NSET | PØØ4 - CØØ1 | PØ1Ø - CØØ1 | PØØ3 - CØØ2 |
| SØØ5-PQØ | PØØ5 - CØØ1 | PØØ4 - CØØ2 | |
| SØØ6-NQ1 | PØØ8 - CØØ1 | PØ12 - CØØ1 | PØØ3 - CØØ3 |
| SØØ7-PQ1 | PØØ9 - CØØ1 | PØØ5 - CØØ2 | |
| SØØ8-CARRY | PØØ1 - CØØ3 | PØØ6 - CØØ2 | |
| SØØ9-VCC | PØ14 - CØØ1 | PØ14 - CØØ3 | PØ5Ø - CØØ2 |
| SØ1Ø-TERRA | PØØ7 - CØØ1 | PØØ7 - CØØ3 | PØ25 - CØØ2 |

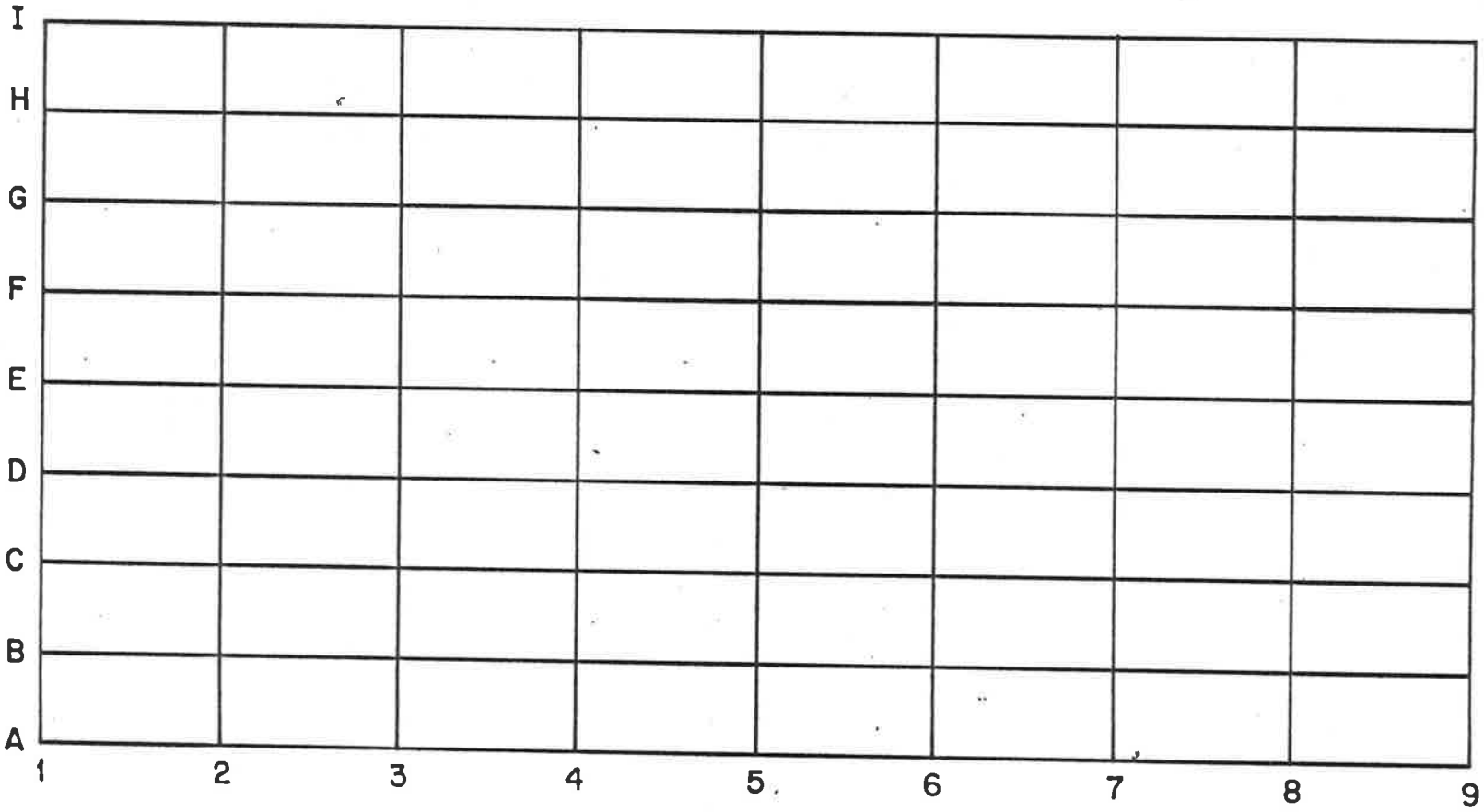



FIG. IV-4

# POSICIONAMENTO

A interpretação deste comando ativará a fase de Posicionamento de componentes nas placas integrantes do projeto em andamento.

Além do posicionamento do componente, resultará, como informação, o tipo de padrão de componente que deverá ser utilizado.

Compõem este comando, os sub-comandos apresentados na tabela IV.7.

* PARAMETRICO
* PROCESSAR nome do bloco
* LISTAR nome do bloco
* DIRETORIO
* FIM

Tabela IV.7

Passa-se, a seguir, à análise de cada sub-comando:

1. * PARAMETRICO

Através deste sub-comando, o projetista decidirá como agrupar os membros de componentes singelos, cuja partição foi comandada através do sub-comando * PARAMETRIZAR, solicitado na fase de Partição.

O projetista deverá, antes de proceder ao posicionamento, resolver os membros que foram parametrizados.

A forma de resolver a parametrização é a seguinte:

(tipo do componente/código de parametrização) = (tipo do componente/ código resolvido)

2. * PROCESSAR nome do bloco

Este sub-comando implica no processamento da fase que determinará o pocisionamento dos componentes.

O nome do bloco deverá ser o nome original declarado na fase de Descrição Lógica.

Os operadores para este sub-comando são:

a) CONDICOES =

Através do qual permite-se a fixação de um determinado componente em uma das placas da partição.

A forma de descrever essas condições é a seguinte:

(nº do componente do projetista, posição X, posição Y, padrão), ...

b) DIMENSAO = (,)

Operador este, análogo ao da Partição.

c) CELULA BASICA = (,)

Operador análogo ao da Partição.

3. * LISTAR nome do bloco

Através deste sub-comando o projetista obterá uma impressão do bloco mencionado, contendo o posicionamento dos componentes.

O nome do bloco deverá ser o da descrição lógica, acrescido do número de seqüência incorporado pelo Supervisor.

4. * DIRETORIO

Permite ao projetista obter uma impressão dos blocos existentes no arquivo da Partição.

5. * FIM

Este sub-comando encerra a descrição das condições.

A fase de Pocisionamento opera sobre os mesmos arquivos da fase de Partição. Apenas incluirá, nos blocos correspondentes às placas particionadas e cujos componentes foram posicionados, a posição destes e o padrão utilizado.

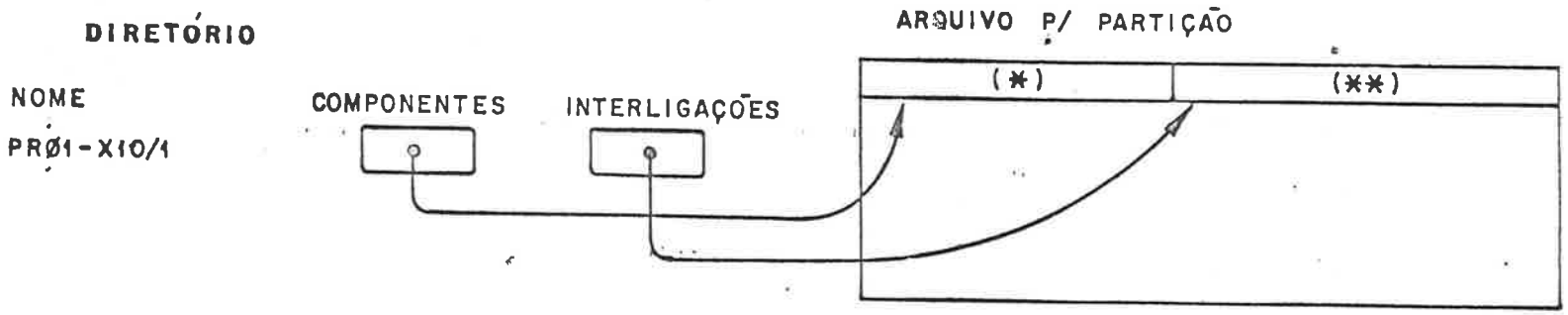
A figura IV.5 é a estrutura utilizada e o Posicionamento dos componentes no bloco.

Exemplo:

Refere-se sempre ao projeto PRØ1-X1Ø:

```
# POSICIONAMENTO
* PROCESSAR PRØ1-X1Ø
  CONDICOES
  (CØØ2, 1, A, CM5Ø), (CØØ1, 2, B, I14N), (CØØ3, 2,F, I14N)
* FIM
# FIM
```

A figura IV.6 é a placa com o reticulado solicitado e com a definição das posições dos componentes.




APÓS PARTIÇÃO E POSICIONAMENTO: O CASO CORRESPONDENTE A TER-SE APENAS UMA PLACA.

(*)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| CØØ1 | 7474 | 2.6Ø 1.5Ø | I14N | CØØ2 | AMP-M-5Ø | ØØØ 1.5Ø | CM 5Ø |
| CØØ3 | 74Ø2 | 2.6Ø 6.5Ø | I14N | | | | |

(**)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SØØ1 | NRESET | PØØ1 | CØØ1 | PØ13 | CØØ1 | PØØ1 | CØØ2 | SØØ2 | NQØ |
| PØØ2 | CØØ1 | PØØG | CØØ1 | PØ11 | CØØ1 | PØØ2 | CØØ3 | SØØ3 | PCLOCK |
| PØØ3 | CØØ1 | PØØ2 | CØØ2 | ... | | | | | |

Figura IV.5

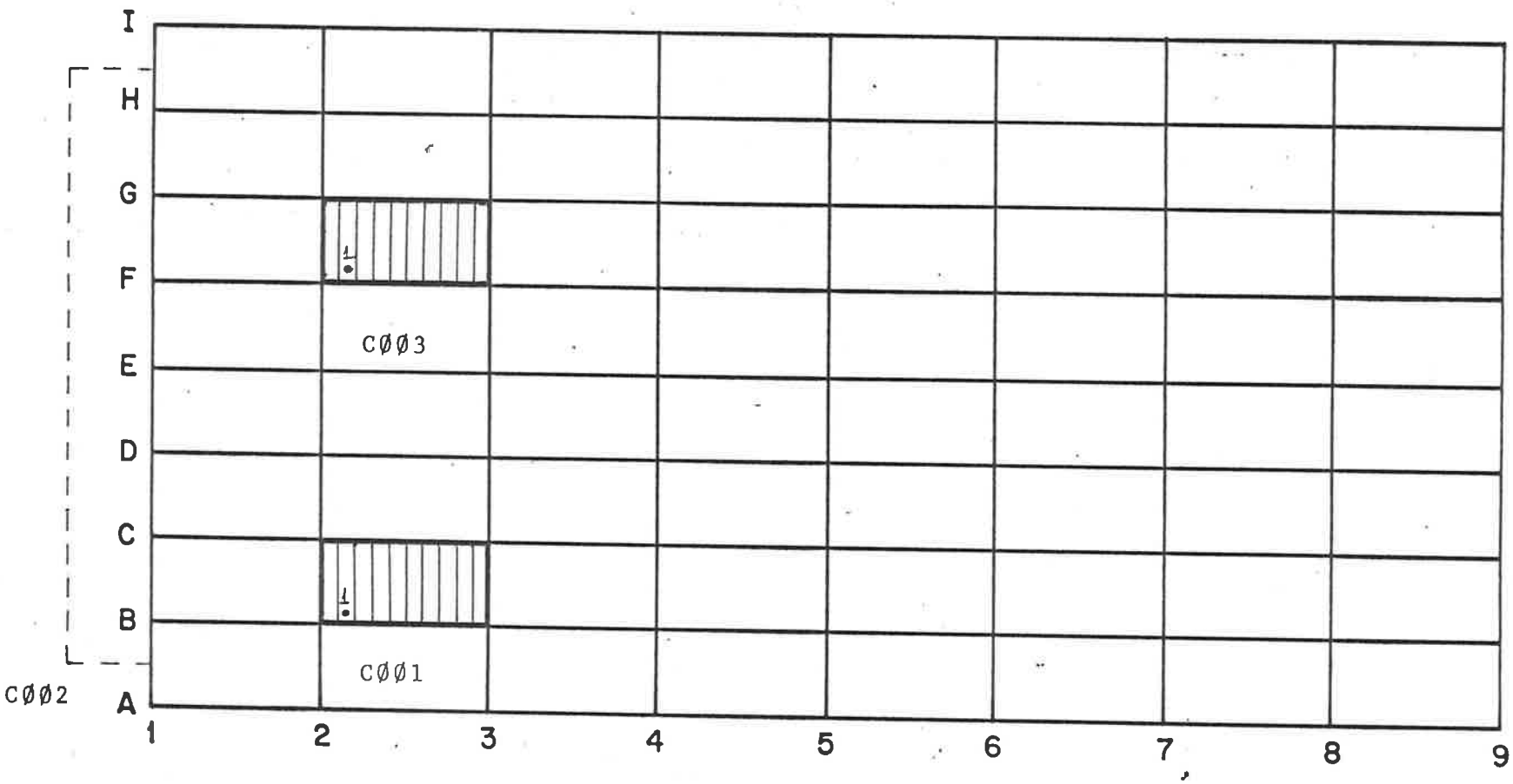



Figura IV.6

# FIACAO

Este comando informará ao Supervisor que deve ser ativada a fase de Fiação.

Mostra-se os sub-comandos constituintes desta fase na tabela IV.8:

* PROCESSAR nome do bloco
* FIM

Tabela IV.8

Analisa-se, a seguir, os sub-comandos da tabela IV.8.

1. * PROCESSAR nome do bloco

Através deste sub-comando efetuar-se-á o processamento da fase de Fiação para o projeto já descrito e contido no bloco de nome mencionado no sub-comando.

O nome do bloco diz respeito ao nome original declarado quando da descrição lógica do projeto.

Compõem este comando, os operadores:

a) NUMERO DE NIVEIS = numero

Através do qual será fornecido o limite máximo de conecções por pino.

b) COR DO FIO = (cor, nivel), (cor, nivel), ...

Através do qual o projetista indica a cor do fio de "WIRE-WRAP" e em que nível ela deve ser utilizada.

c) NUMERO DO FIO = numero

Através do qual é fornecida a bitola do fio a ser utilizado.

2. * FIM

Através deste sub-comando, encerra-se o fornecimento de informações referentes a fase de Fiação.

A fase de Fiação é desenvolvida sobre a mesma estrutura que foi utilizada pela fase de Posicionamento. A figura IV.5 é o resultado a ser utilizado como entrada de informações para esta fase.

Note-se, que para a impressão deve-se transformar a coordenada absoluta dos pinos de referência dos componentes em posição de localização do componente em uma célula do reticulado.

Para o componente conector de borda a informação é dada sobre a forma de S⟨pino⟩, onde S corresponde ao lado da placa oposto aquele ocupado pelos componentes.

Exemplo:

Refere-se sempre ao projeto PRØ1-X1Ø:

```
#FIACAO
* PROCESSAR PRØ1-X1Ø
  NUMERO DE NIVEIS = 2
  COR DO FIO = (BRANCO, 1), (VERDE, 2)
  NUMERO = AWG-24
* FIM
# FIM
```

O resultado do processamento da fase é apresentado em forma impressa na página que segue.

| LIGAÇÃO NÚMERO | NOME | PINOS A INTERLIGAR | | COMPRIMENTO (mm) | NÍVEL | COR | BITOLA |
|---|---|---|---|---|---|---|---|
| ØØ1 | NRESET | Ø2BØ1 | Ø2B13 | 70 | 1 | BRANCO | AWG-24 |
| ØØ2 | NQØ | Ø2BØ2 | Ø2BØ6 | 70 | 1 | BRANCO | AWG-24 |
| ØØ3 | NQØ | Ø2B11 | Ø2FØ2 | 110 | 1 | BRANCO | AWG-24 |
| ØØ4 | PCLOCK | Ø2BØ3 | SØ2 | 90 | 1 | BRANCO | AWG-24 |
| ØØ5 | NSET | Ø2BØ4 | Ø2B1Ø | 70 | 1 | BRANCO | AWG-24 |
| ØØ6 | PQØ | Ø2BØ5 | SØ4 | 90 | 1 | BRANCO | AWG-24 |
| ØØ7 | NQ1 | Ø2BØ8 | Ø2B12 | 70 | 1 | BRANCO | AWG-24 |
| ØØ8 | PQ1 | Ø2BØ9 | SØ5 | 90 | 1 | BRANCO | AWG-24 |
| ØØ9 | PCARRY | Ø2FØ1 | SØ6 | 100 | 1 | BRANCO | AWG-24 |
| Ø1Ø | VCC | Ø2B14 | Ø2FØ7 | 110 | 1 | BRANCO | AWG-24 |
| Ø11 | TERRA | Ø2BØ7 | Ø2FØ7 | 110 | 1 | BRANCO | AWG-24 |
| Ø12 | NRESET | Ø2B13 | SØ1 | 100 | 2 | VERDE | AWG-24 |
| Ø13 | NQØ | Ø2BØ6 | Ø2B11 | 70 | 2 | VERDE | AWG-24 |
| Ø14 | NSET | Ø2B1Ø | SØ3 | 110 | 2 | VERDE | AWG-24 |
| Ø15 | NQ1 | Ø2B12 | Ø2FØ3 | 110 | 2 | VERDE | AWG-24 |
| Ø16 | VCC | Ø2F14 | S5Ø | 90 | 2 | VERDE | AWG-24 |
| Ø17 | TERRA | Ø2FØ7 | S25 | 100 | 2 | VERDE | AWG-24 |

# ROTAS

A interpretação deste comando pelo Supervisor ativará a fase de elaboração de caminhos elétricos em placas de circuito impresso.

Os sub-comandos pertencentes ao comando citado estão apresentados na tabela IV.9.

* PROCESSAR nome do bloco
* FURACAO
* FIM

Tabela IV.9

Analisa-se, a seguir, os sub-comandos acima:

1. * PROCESSAR nome do bloco

Este sub-comando iniciará efetivamente a fase de execução do programa de rotas.

O nome do bloco corresponde ao que foi descrito através do sub-comando * IDENTIFICACAO, quando da Descrição Lógica.

Os operadores que compõem este sub-comando e respectivas funções são:

a) NUMERO DE CAMADAS = numero

Este operador ditará o tipo de "suporte impresso" que a placa deverá obedecer.

No caso do número ser igual a 2, será o de placa de circuito impresso com cobre nas duas faces.

Se for maior que 2, será "multi-layer".

b) TIPO DE LIGACAO = codigo

Indicará, para as placas de duas faces, qual a direção predominânte na face da placa citada. Assim:

IH ou SV: corresponderá à face superior, ligações verticais e face inferior, ligações horizontais.

IH ou SH: idem, ligações horizontais na face superior e verticais na inferior.

c) LIGACOES $\left\{ \begin{array}{l} \text{ESPECIAIS} \\ \text{PRIORITARIAS} \end{array} \right\}$ =

Através deste operador pode-se acrescentar ligações (especiais) ou estabelecer que determinadas ligações devem ser efetuadas antes das demais (prioritárias).

No caso de se acrescentar ligações deve-se utilizar a palavra chave ESPECIAIS.

Querendo se determinar a prioridade deve-se utilizar a palavra chave PRIORITARIAS.

As formas a serem obedecidas para descrição das ligações poderão ser:

$C_1$: (nº da ligação, par de coordenada, $\left\{\begin{matrix} U \\ L \end{matrix}\right\}$, par de coordenada, $\left\{\begin{matrix} U \\ L \end{matrix}\right\}$, ... )

$C_2$: (nome do sinal, pino inicial de ligação, número do componente, $\left\{\begin{matrix} U \\ L \end{matrix}\right\}$, par de coordenada, $\left\{\begin{matrix} U \\ L \end{matrix}\right\}$, ...)

Num mesmo operador poderão aparecer ambas as formas de descrição, desde que separadas por vírgulas. Uma "/" encerra o fornecimento de ligações.

d) SAIDA GRAFICA $\left\{\begin{matrix} \text{DISPLAY} \\ \text{PLOTTER} \end{matrix}\right\}$

Este operador selecionará o dispositivo de saída gráfica que se deseja.

2. * FURACAO

Este operador deverá informar ao programa de rotas que o projetista deseja obter uma fita cujos dados comandarão uma máquina de controle numérico que efetuará a furação da placa.

3. * FIM

Este operador encerra o fornecimento de informações para a fase de Rotas.

Exemplo:

```
# ROTAS
* PROCESSAR PRØ1-X1Ø
  NUMERO DE CAMADAS = 2
  TIPO DE LIGACAO = SV
  LIGACOES ESPECIAIS =
  (VCC, 5Ø, CØØ2, U, 2.6, 9.5, U, 7.5, 9.5, U, 7.5, 3.75),
  (VCC, 14, CØØ3, U, 2.6, 9.5), (VCC, 14, CØØ1, U, 2.6, 3.75),
  (TERRA, 25, CØØ2, L, 6.Ø, 4.75, L, 6.Ø, 6.5),
  (TERRA, Ø7, CØØ3, L, 6.Ø, 6.5),
  (TERRA, Ø7, CØØ1, L, 6.0, 1.50, L, 6.0, 4.75) /

  SAIDA GRAFICA PLOTTER
* FIM
* FURACAO
# FIM
```

A figura IV.7 é o resultado da placa de circuito impresso, levando-se em consideração a forma de descrição das ligações acima, sem as demais ligações efetuadas pela fase de Rotas.

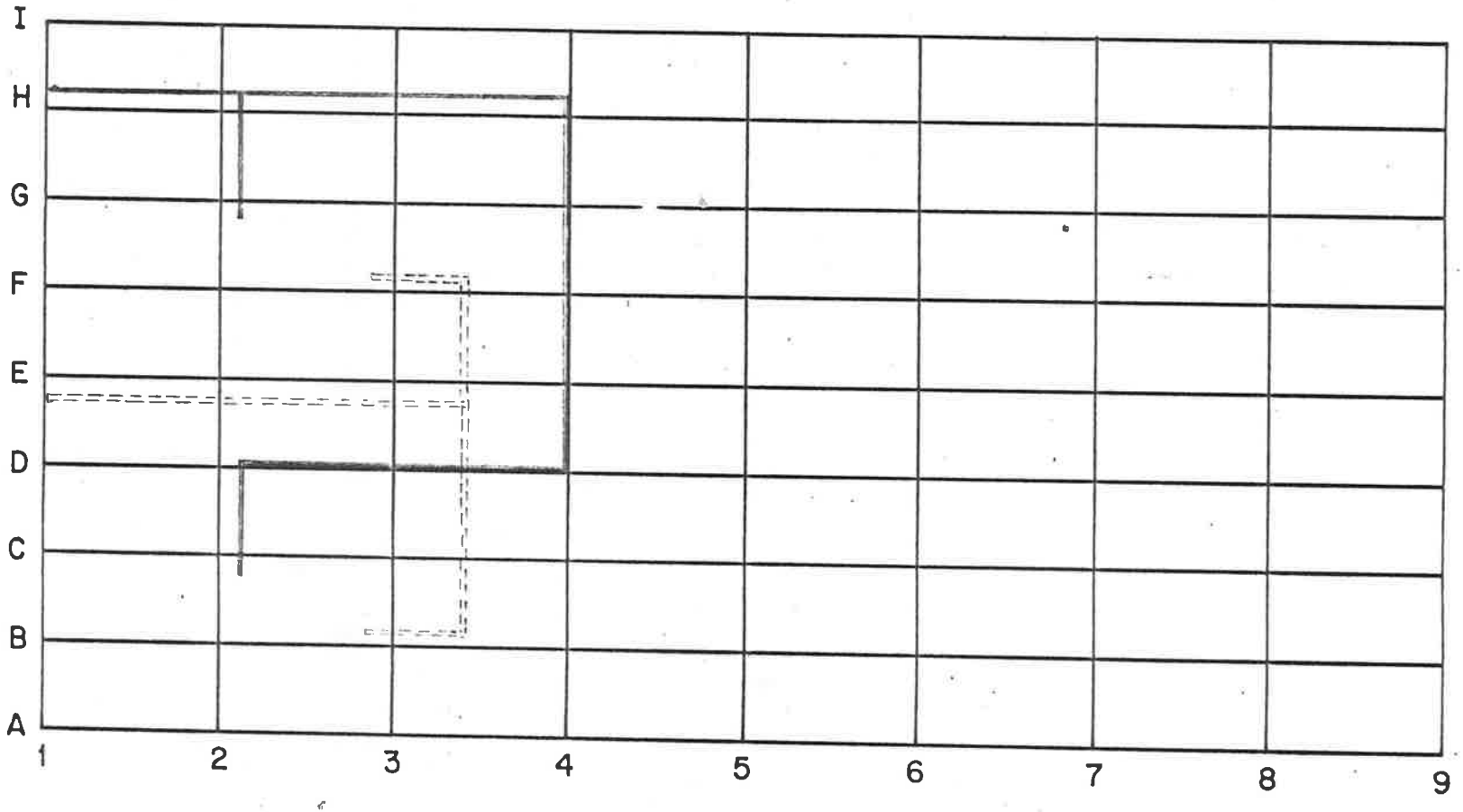



Figura IV.7

# DOCUMENTACAO

Este comando ativará a fase que permitirá obter a documentação final do projeto. É claro que alguns documentos importantes para o projeto já foram obtidos durante o processamento de fases intermediárias.

Os sub-comandos constituintes desta fase estão indicados na tabela IV.10.

* PROCESSAR nome do bloco
* FIM

Tabela IV.10

Analisa-se, a seguir, esses sub-comandos.

1. * PROCESSAR nome do bloco

Através deste sub-comando iniciar-se-á o tratamento dos arquivos do Sistema de Automação, para obtenção dos documentos solicitados através dos operadores:

a) LISTA DE COMPONENTES

Através deste operador mostrar-se-ão tabelas impressas contendo:

a.1 número do componente atribuído no projeto, código de fabricação do componente, função, número da placa no qual está alocado e a posição do mesmo na placa;

a.2 códigos dos componentes e quantidade por placa;

a.3 número da placa respectiva, necessidades energéticas (consumo).

b) DIAGRAMA LOGICO

Será obtida saída impressa, por placa, do esquema lógico fornecido inicialmente como descrição lógica do projeto.

c) DISPOSICAO

Obter-se-á uma saída impressa, por placa, com a disposição dos componentes na placa.

d) FUNCOES DE COMPONENTES

Obter-se-á uma impressão que descreve o componente, pinos associados a sinais com respectiva função.

Haverá marcação especial para pinos não utilizados e respectiva função, permitindo, assim, posterior utilização quando necessário.

e) LISTA DE SINAIS

Obter-se-á uma lista de sinais, por placa, com informações relativas à:

e.1 geração e utilização na placa;

e.2 apenas geração;

e.3 apenas utilização.

Biblioteca da Escola Politécnica
São Paulo
FD-241

# FIM

Este comando, ao ser interpretado pelo supervisor, encerrará a execução da fase.

# CAPÍTULO V

## V.1 OBSERVAÇÕES FINAIS

OBSERVAÇÃO 1. Como já foi mencionado, o projetista poderá optar por executar a Automação do seu projeto lógico, sob total controle do Sistema ou, resolver partes de seu projeto através da execução de fases isoladas da Automação.

Ao se decidir pelo processamento de seu projeto em fases separadas, o projetista deverá verificar que dados ou informações esta fase a ser executada necessitará, e sob qual forma, estes dados deverão ser entregues ao Sistema.

Uma apreciação, sobre o tipo de trabalho que o projetista deseja receber do Sistema e as fases que deverá recorrer para a realização deste trabalho, é feita em seguida.

### 1. APENAS A SIMULAÇÃO

Neste caso deverá descrever os dados da planta lógica a ser simulada, através da fase de Descrição Lógica.

Em seguida poderá executar sua simulação.

### 2. APENAS A PARTIÇÃO

Como o projetista quer se desfazer do trabalho manual de particionar seu projeto, disporá dos recursos interativos da fase de Partição mas, para tal, descreverá seu projeto lógico utilizando o recurso que dispõe da fase de Descrição Lógica.

Em seguida fará a Partição e poderá, através dos recursos que esta fase dispõe, obter uma solução concorde com os critérios que desejar.

3. APENAS O POSICIONAMENTO DOS COMPONENTES

O projetista deverá recorrer a fase de Descrição Lógica e em seguida, utilizando recursos que esta fase dispõe, separar as informações referentes à sua planta lógica em dados referentes aos componentes e suas interligações.

Após esse procedimento, deverá executar a fase de Posicionamento, também sendo-lhe aqui dada a possibilidade de interagir com as resoluções do Sistema.

4. APENAS A FIAÇÃO

O procedimento neste caso será o mesmo que o do ítem anterior.

Após o posicionamento, onde o projetista fez interações com o Sistema para adequá-lo, executará a fase de Fiação.

5. APENAS ROTAS

Também para este item, o procedimento é equivalente ao do ítem 3.

Executará em seguida a fase de Rotas onde utilizará todos os recursos que esta fase lhe proporciona.

A figura V-1 resume esses procedimentos.

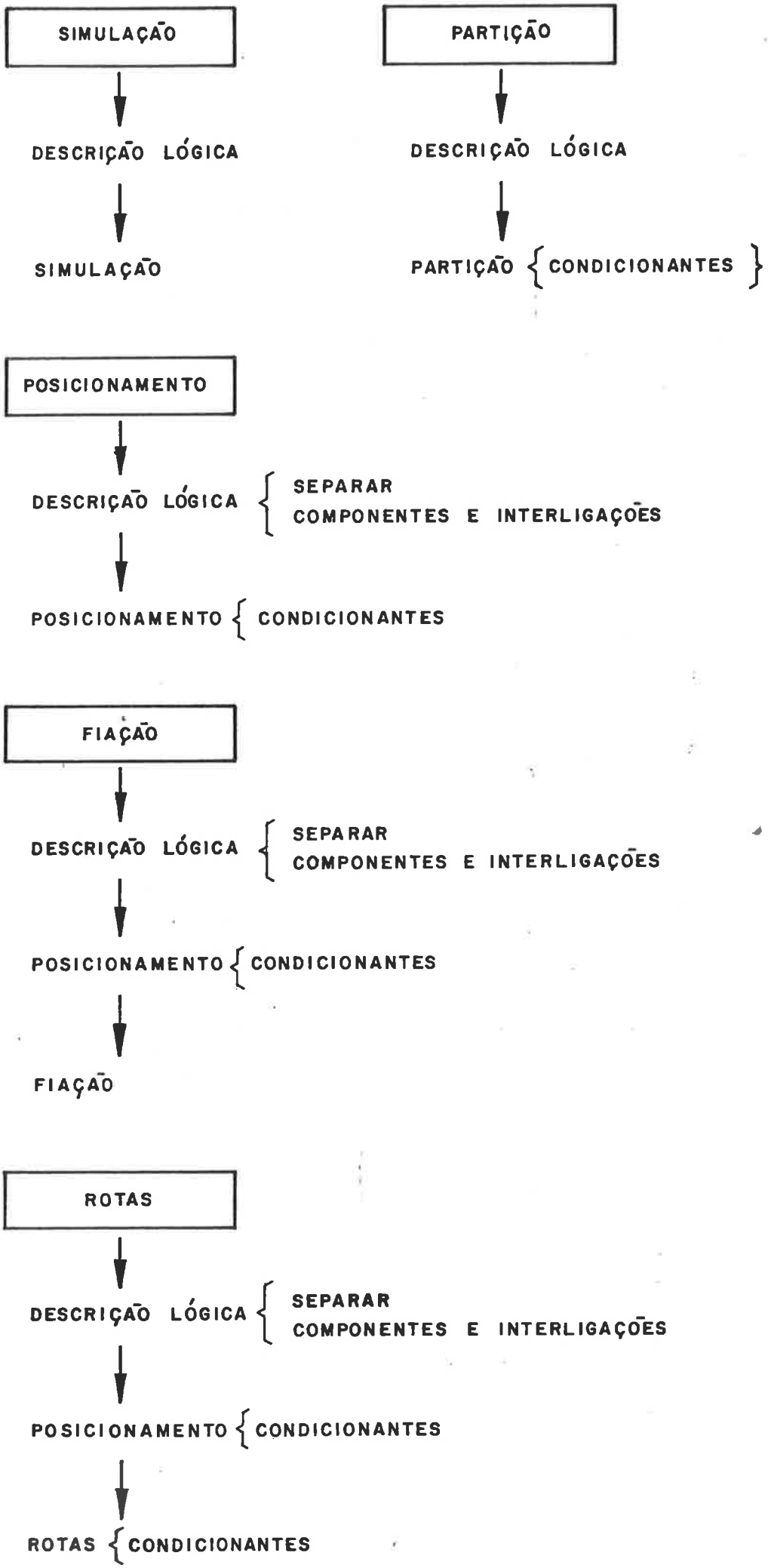

SIMULAÇÃO
DESCRIÇÃO LÓGICA
SIMULAÇÃO
PARTIÇÃO
DESCRIÇÃO LÓGICA
PARTIÇÃO
CONDICIONANTES
POSICIONAMENTO
DESCRIÇÃO LÓGICA
SEPARAR
COMPONENTES E INTERLIGAÇÕES
POSICIONAMENTO
CONDICIONANTES
FIAÇÃO
DESCRIÇÃO LÓGICA
SEPARAR
COMPONENTES E INTERLIGAÇÕES
POSICIONAMENTO
CONDICIONANTES
FIAÇÃO
ROTAS
DESCRIÇÃO LÓGICA
SEPARAR
COMPONENTES E INTERLIGAÇÕES
POSICIONAMENTO
CONDICIONANTES
ROTAS
CONDICIONANTES

FIG. V-1

OBSERVAÇÃO 2. Os arquivos estruturados para apoiar o Sistema Integrado de Automação devem ser manipulados por programas adequados, os quais permitirão a obtenção fácil de uma determinada informação.

Para tanto dispõe-se de programas comerciais de apoio à manipulação de arquivos.

Esses programas devem ser muito bem analisados para se escolher o mais adequado à estrutura de arquivos utilizada.

Sendo a estrutura de dados utilizada nesses arquivos bastante particular, para a solução deste problema específico apresenta-se como mais conveniente a alternativa de desenvolvimento de um conjunto de programas bem definidos para a manipulação dessa particular estrutura de dados.

OBSERVAÇÃO 3. Os programas atualmente disponíveis no Laboratório de Sistemas Digitais devem sofrer uma adequação para serem incorporados so Sistema Integrado de Automação.

Com referência ao Sistema de Computação nos quais foram desenvolvidos, deve-se efetuar a reprogramação de tais programas, uma vez que se destinavam a utilização em máquinas de reduzida capacidade de memória, reduzida capacidade de arquivos em unidades de armazenamento secundários e de baixa velocidade de processamento.

Dispõe-se atualmente de recursos de computação extremamente flexíveis e adequados a suportar um Sistema Integrado de Automação do tipo descrito neste trabalho.

# "B I B L I O G R A F I A"

[1 ] - BERETVAS, LIN, TAYLOR
"A General Purpose Design Automation File System", Proc.4th. Design Automation Workshop - IBM - GDL4, 1966 pág. 20-1 a 20-13.

[2 ] - BREUER, M.A.
"Design Automation of Digital Systems", Englewood Cliffs, New Jersey, Prentice Hall Inc., 1972

[3 ] - BREUER, M.A.
"Functional Partitioning and Simulation of Digital Circuits", IEEE Trans. Comp., Vol. C-19 novembro 1970 - pág. 1038 a 1046.

[4 ] - BREUER, M.A.
"Recent Developments in the Automated Design and Analysis of Systems", Proc. IEEE, Vol. 60, Nº 1, Janeiro 1972 - pág. 12 a 20.

[5 ] - CASE, P.W. e outros.
"Solid Logic Design Automation", IBM Journal, Abril 1964 pág. 127 a 140.

[6 ] - CUNHA, N.M.J.
"Automação de Projetos de Sistemas Digitais - Partição de Circuitos Lógicos", Tese de Mestrado em Engenharia de Eletricidade - Escola Politécnica da Universidade de São Paulo, 1976.

[7 ] - DATE, C.J.
"An Introduction to Database Systems". Addison-Wesley Publishing Company, Inc. - 2nd. Edition, Janeiro 1976.

[8 ] - FITCH, A.E.
"A User Look at D.A. - Yesterday, Today, Tomorrow", Proc. 6º Design Automation Workshop, IBM - Junho 1969, pág. 371 a 382.

[9 ] - KATZAN, H.J.
"Computer Data Management and Database Technology", Van Nostrand Reinhold Company, 1975.

[10] - LAKE, D.W.
"Logic Simulation in Digital Systems", Computer Design, Maio 1970 - pág. 77 a 83.

[11] - MARTINI, J.S.C.
"Automação de Projetos de Sistemas Deigitais - Rotas de Circuito Impresso", Tese de Mestrado em Engenharia de Eletricidade - Escola Politécnica da Universidade de São Paulo, 1976.

[12] - MASSOLA, A.M.A.
"Automação de Projetos de Sistemas Digitais - Simulação em Nível de Portas Lógicas", Tese de Doutoramento em Engenharia de Eletricidade - Escola Politécnica da Universidade de São Paulo, 1974.

[13] - MELNIKOFF, S.S.S.
"Automação de Projetos de Sistemas Digitais - Posicionamento de Componentes em Placas de Circuito Impresso", Tese de Mestrado em Engenharia de Eletricidade (em preparação) - Escola Politécnica da Universidade de São Paulo.

[14] - ROSENBLUTH, W.
"Design Automation Architeture and Applications", Computer Design, Fevereiro 1976 - pág. 12 a 17

[15] - SANDERSON & MILICI
"A Logic and Signal Flow Diagram Subsystem", Sperry Division - Sperry Rand Corp., Internal Publication pág. 250 a 257.