UNIVERSIDADE ESTADUAL DE CAMPINAS - UNICAMP

FACULDADE DE ENGENHARIA ELÉTRICA - FEE

"SISTEMA GRAFICO DE ENTRADA DE DADOS PARA SIMULADORES LÓGICOS"

Ricardo Fannain

Orientadora: Profa. Dra. Alaide Pellegrini Mammana
Co-orientador: Prof. Dr. Carlos Ignácio Z. Mammana

Tese apresentada à Faculdade de Engenharia Elétrica, da Universidade Estadual de Campinas - UNICAMP, como parte dos requisitos exigidos para obtenção do título de MESTRE EM CIÊNCIAS.

- Agosto de 1988 -

Este exemplar corresponde à redação de tese defendida por Ricardo Vannain e aprovada pela Comissão julgadora em 9 de agosto de 1988

16/8/88

Aloísio Pharmacio

Ao meu pai, o passado, e ao meu filho,
o futuro.

Agradeço à minha esposa Ione, pelo apoio e compreensão nas frequentes ausências.

Não posso deixar de agradecer também ao Centro Tecnológico para Informática e a todos que de um modo ou outro contribuiram para o desenvolvimento deste trabalho, em especial à colega Nilsa Aparecida Barreto, que colaborou na digitação deste texto.

Agradecimentos especiais aos Profs. Dra. Alaide Pellegrini Mammana e Dr. Carlos Ignácio Z. Mammana, pela determinação, competência e auxílio na realização desta tese.

ÍNDICE

RELAÇÃO DAS TABELAS E FIGURAS

APRESENTAÇÃO

O trabalho desta tese originou-se da necessidade do desenvolvimento de um sistema gráfico para descrição dos elementos de circuitos e de suas interligações - "NET-LIST", durante a fase de projeto de sistemas digitais [1].

Este sistema gráfico consiste de dois programas, o Programa de Entrada Esquemática - PESQA - e o Editor Gráfico de Estímulos Externos - EDTES, que auxiliam na descrição dos arquivos de dados de entrada usados em simuladores lógicos, tais como o SIMUL [2,3] e o HILO-3 [4]. O SIMUL é um simulador lógico com atraso unitário, desenvolvido no Centro Tecnológico para Informática, que usa os valores lógicos 0, 1 e X. O HILO-3 é um sistema de simulação lógica, desenvolvido pela GenRad Inc., formado por um simulador lógico com atraso atribuído, um simulador de falhas e um gerador automático de vetores de teste.

O Programa de Entrada Esquemática - PESQA - e o Programa Editor de Estímulos Externos - EDTES - foram desenvolvidos visando sua utilização em sistemas computacionais de baixo custo, ou seja, sistemas baseados em microcomputadores de 16 bits com monitor de vídeo 640 X 200 pontos, teclado alfanumérico, um disco rígido de 10 M bytes e pelo menos uma unidade de disco flexível, possibilitando assim sua disseminação visando principalmente seu uso em ensino e treinamento [5,6].

Esses programas fazem parte do Sistema Didático de Projeto [7], utilizado no Curso de Laboratório de Microeletrônica da II e III Escola Brasileiro-Argentina de Informática - EBAI, e vêem sendo utilizados, em carater experimental, no projeto de circuitos integrados no Instituto de Microeletrônica do Centro Tecnológico para Informática.

Neste texto descrevemos detalhadamente os dois programas, discutindo em que nos baseamos para seu desenvolvimento, quais as opções para sua estrutura de dados e como procedemos para o seu desenvolvimento e documentação, tendo em vista a preocupação de torná-los simples, de fácil uso e utilizáveis em computadores de baixo custo. Além do uso desses programas, em microcomputadores, para o ensino e treinamento, ele servirá também como protótipo para o desenvolvimento de um sistema gráfico mais eficiente, utilizando computadores maiores, com terminais gráficos de alta resolução, e tendo como núcleo gráfico, pacotes comerciais, tais como o PLOT 10 da Tektronix ou o "Advanced Graphics Packaged" da Hewlett-Packard.

De forma a tornar este texto didático, procuramos estruturá-lo em capítulos. No capítulo I discutimos alguns aspectos importantes no desenvolvimento de ferramentas de projeto de circuitos integrados digitais auxiliado por computador - PAC, iniciando com uma introdução aos conceitos básicos do projeto de circuitos integrados, seguida de uma breve discussão sobre as estruturas de dados mais utilizadas nesses programas de PAC. Nos capítulos II e III descrevemos detalhadamente as rotinas de transformações gráficas e as funções dos programas PESQA e EDTES, respectivamente, com base na norma IEEE para Especificação de Requisitos de Software [8]. O capítulo IV foi reservado para a análise do desempenho dos protótipos desenvolvidos, com base na utilização experimental no projeto de sistemas integrados digitais de relativa complexidade. Em particular nos referimos à capacidade de armazenamento de dados, comentando as principais limitações observadas, com base nas quais são propostas sugestões para melhorias futuras. É também apresentado um exemplo simples de uso dos programas no projeto de um circuito de um contador.

I. CONCEITOS

1. Projeto Auxiliado por Computador para Circuitos Eletrônicos Digitais

1.1. Introdução

Os computadores vem sendo usados como elementos auxiliares no projeto de sistemas digitais, desde sua primeira geração. No começo seu uso se restringia a tarefas simples, como a de confecção de artes finais - "layout" - utilizadas na fabricação de circuitos impressos e na confecção de esquemas de ligação de circuitos. O projeto lógico em si, era feito manualmente, sempre baseado no conhecimento da teoria dos circuitos lógicos e na experiência do projetista.

Com o advento dos circuitos integrados, esta situação de quase artesanato teve que ser alterada, pela própria complexidade desses circuitos, que inviabiliza o projeto manual e a realização de testes com protótipos.

O projeto de circuitos integrados de grande escala de integração - LSI, e de muito grande escala de integração - VLSI, devido à sua alta complexidade, implica na manipulação de uma grande quantidade de dados, relativos aos parâmetros elétricos dos elementos de circuito, conexões, etc., que só o computador é capaz de tratar e manipular.

Para trabalhar com tão grande volume de dados, é desejável que esses programas de auxílio a projetos, chamados de ferramentas de projeto, sejam de uso fácil e confortável ao projetista, fazendo com que a interação entre o homem e o computador seja próxima e natural.

1.2. Projeto Auxiliado por Computador

Entende-se por Projeto Auxiliado por Computador - PAC - a atividade de projeto de circuitos digitais efetuada com a ajuda de computadores digitais e técnicas computacionais.

A terminologia **Projeto Auxiliado por Computador** foi primeiramente usada para descrever as aplicações de computadores nos cálculos das equações que caracterizam um circuito. Atualmente este termo significa um sistema interativo onde o projetista e o computador trabalham juntos desde a especificação do projeto até a confecção do "layout" do circuito, passando pelas etapas de validação e testes.

Como podemos ver na figura 1.1, um sistema de PAC para projeto de circuitos integrados digitais deverá conter, basicamente, ferramentas tais como, simulador lógico, gerador automático de vetores de teste, simulador de falhas, extrator de circuito, e simulador de circuitos, usadas para a validação e testes do circuito, e ferramentas como editor gráfico, roteador, alocador, verificador de regras de projeto e verificador de regras elétricas, usadas para a confecção de "layout". Todas essas ferramentas, detalhadas no item 2.5, ligadas a uma base comum de dados e a um programa gerenciador, que ao mesmo tempo administra este banco de dados e cuida da interface com o usuário, formam um sistema de PAC integrado.

O PAC é imprescindível se se pretende reduzir custos e tempo de desenvolvimento de sistemas digitais.

1.3. Desenvolvimento de Sistemas de PAC

No desenvolvimento de sistemas de projeto de circuitos digitais auxiliados por computador, deve-se ter em mente tanto velocidade de operação e capacidade aritmética, quanto a capacidade de manipulação de dados, pois quanto mais complexos os circuitos a serem projetados maior o número de dados a serem processados.

Assim, é desejável que o sistema de PAC tenha as sequintes características:

a) Ser capaz de estruturar e manipular grandes quantidades de dados;

b) Permitir alocação dinâmica de memória, pois projetos de circuitos integrados requerem uma grande e imprevisível quantidade de memória;

c) Minimizar o tempo de execução dos programas, diminuindo os custos de projeto;

d) Ser utilizável em diversos computadores, para o que deve ser composto de programas transportáveis para diferentes sistemas computacionais;

e) Ser modular para facilitar modificações e adições de novas funções ou módulos;

f) Ter entradas e saídas eficientes de dados, utilizando, sempre que possível, interfaces gráficas.

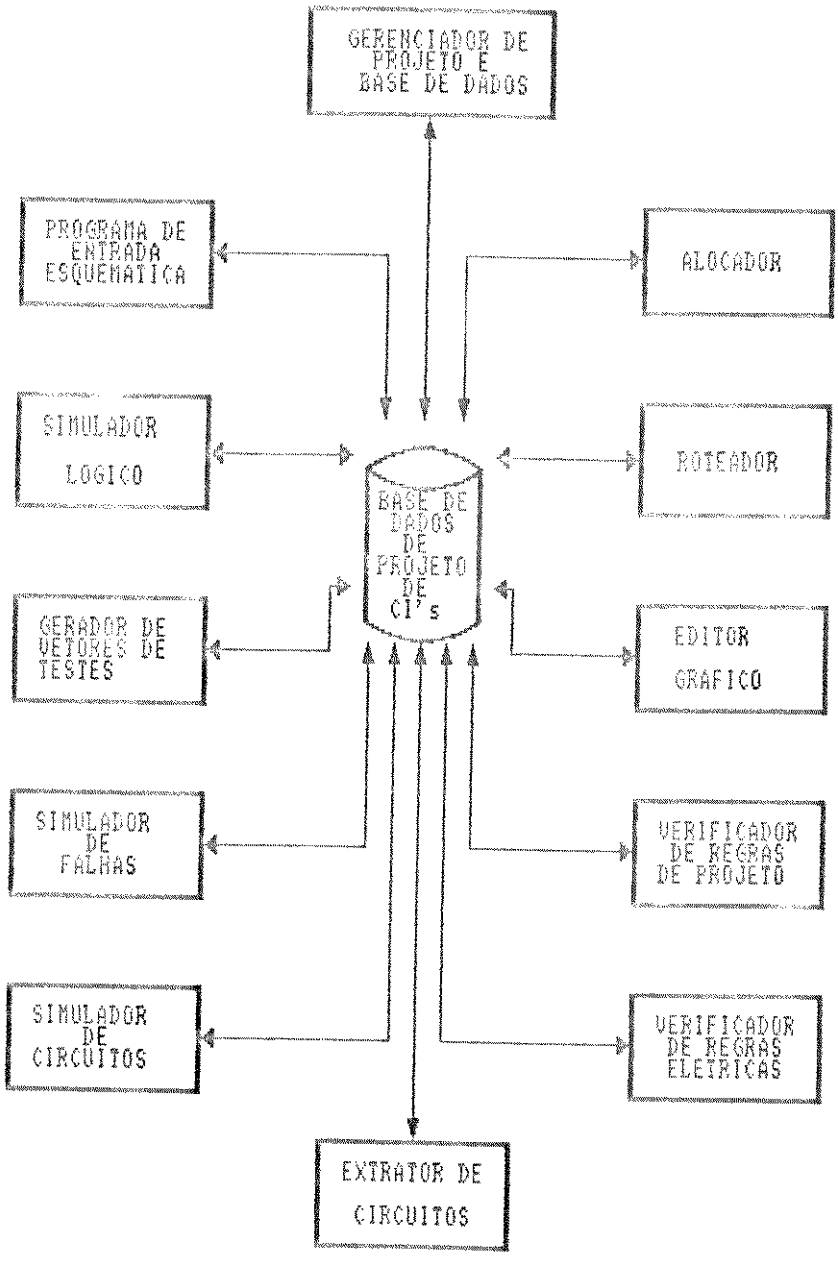


Figura 1.1. Sistema integrado de PAC.

A organização dos dados em PAC é muito importante pois é com eles que traduzimos o mundo real para os modelos utilizados para simular o comportamento dos elementos dos circuitos. Por exemplo, numa simulação lógica, os circuitos são modelados usando-se redes estruturadas em que os elementos de circuito são substituidos por seus modelos lógicos, levando-se em conta suas funções lógicas, tempos de atraso, tempos de subida e descida, número de elementos a serem alimentados (fan-out), etc., além da informação sobre a conectividade de cada elemento. Tem-se então um conjunto de dados para cada elemento de circuito.

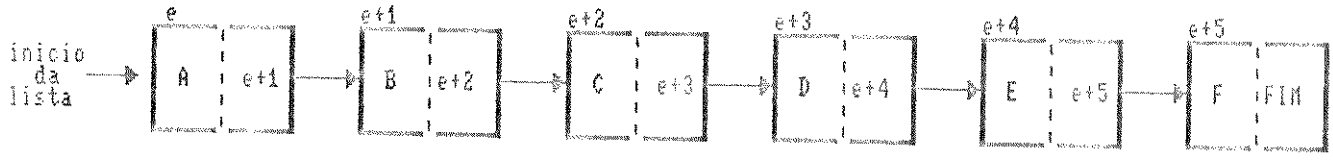
No circuito com muitos elementos, estes conjuntos individuais de dados devem estar ligados através de estruturas que facilitem sua manipulação, devendo ser armazenados de tal forma que possam ser acessados e processados eficientemente. Essas estruturas devem também ser adequadas à inserção e à supressão de dados em função das necessidades do projeto. Nesses programas, geralmente são usadas estruturas de dados em forma de listas, sendo que na figura 1.2 estão ilustrados alguns tipos de listas mais frequentemente usadas [9,10].

A figura 1.2.a mostra uma lista ligada, onde cada célula desta lista é formada pelo dado, ou conjuntos de dados, e pelo apontador para a próxima posição da lista. Na última célula, o apontador indica o fim da lista. A figura 1.2.b mostra uma lista ligada onde cada célula é composta por dados e apontadores, e em determinadas células, o dado é um endereço para outra lista ligada, ou seja, é um apontador para uma sub-lista. A figura 1.2.c mostra uma lista ligada, mas com o apontador da última célula contendo o endereço da primeira, formando uma lista circular. Na figura 1.2.d temos uma lista duplamente ligada, onde cada célula é formada pelos dados e dois apontadores, e cada apontador contém endereços de células diferentes da lista. A figura 1.2.e mostra uma estrutura em anel onde temos várias sub-listas circulares, ligadas entre si também de forma circular.

A escolha do tipo de estrutura de dados depende da particular aplicação do programa pois, para cada aplicação, podemos ter diferentes tipos de dados, e de como queremos acessar esses dados. Para um sistema com vários programas, é importante usar a mesma estrutura para o mesmo tipo de dado, para facilitar a integração desses programas.

Dentre as características desejáveis para um sistema de PAC, a eficiência de comunicação entre o homem e a máquina, expressa pelas facilidades de entrada e saída, exige que os seguintes aspectos sejam considerados:

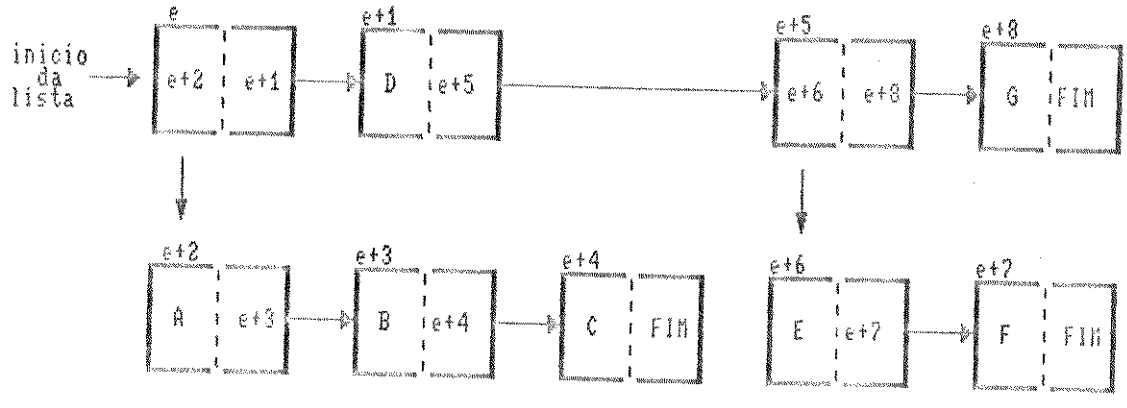
a) A linguagem de interação entre o usuário e o sistema deve ser conversacional, ou seja, fácil de se usar e fácil de se entender;



ENDEREÇO	CONTEÚDO	APONTADOR
e	A	e+1
e+1	B	e+2
e+2	C	e+3
e+3	D	e+4
e+4	E	e+5
e+5	F	FIM

(a) lista ligada

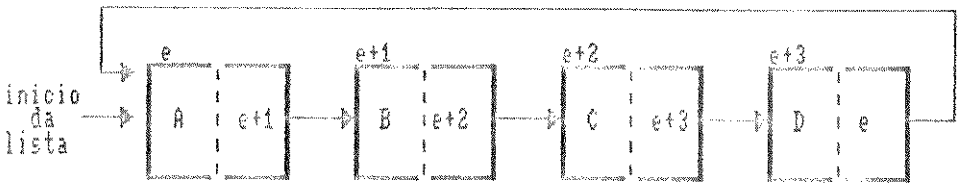
Figura 1.2. Estruturas de dados.



ENDEREÇO	CONTEÚDO	APONTADOR
e	e+2	e+1
e+1	D	e+5
e+2	A	e+3
e+3	B	e+4
e+4	C	FIM
e+5	e+6	e+8
e+6	E	e+7
e+7	F	FIM
e+8	G	FIM

(b) lista ligada com sub-listas

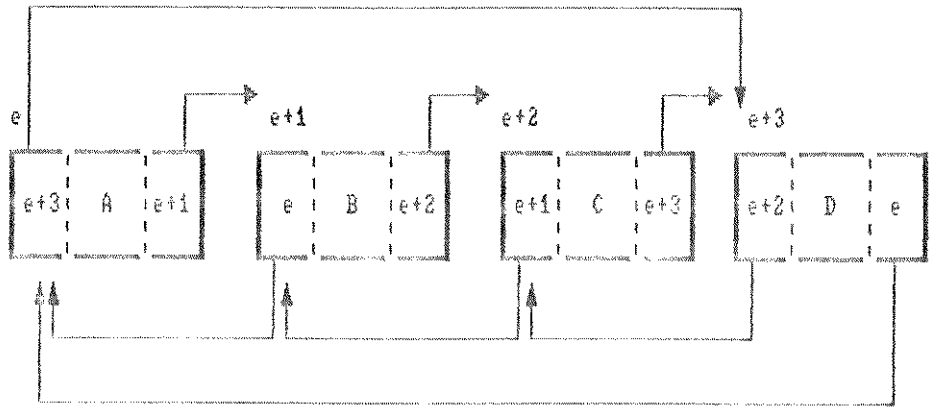
Figura 1.2. Estruturas de dados (continuação).



ENDEREÇO	CONTEÚDO	APONTADOR
e	A	e+1
e+1	B	e+2
e+2	C	e+3
e+3	D	e

(c) lista circular ligada

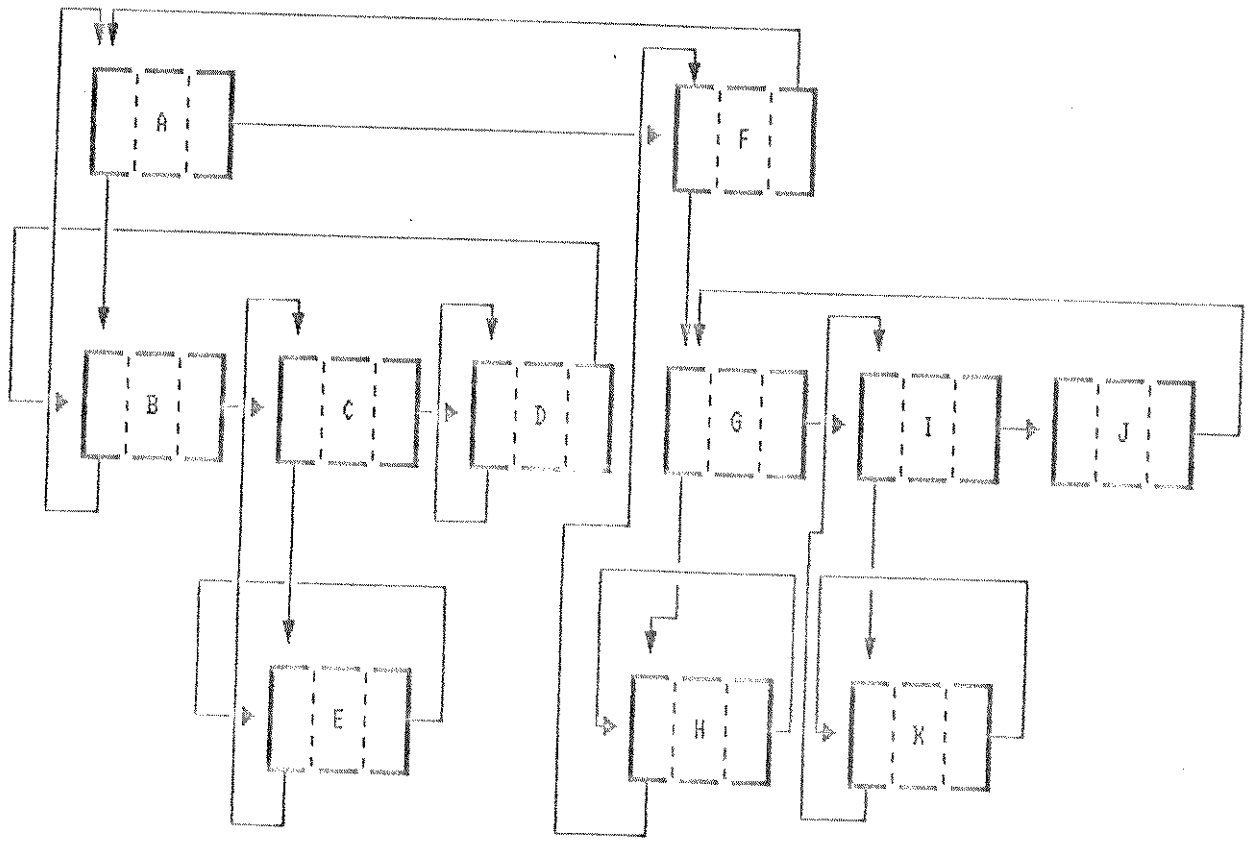
Figura 1.2. Estruturas de dados (continuação).



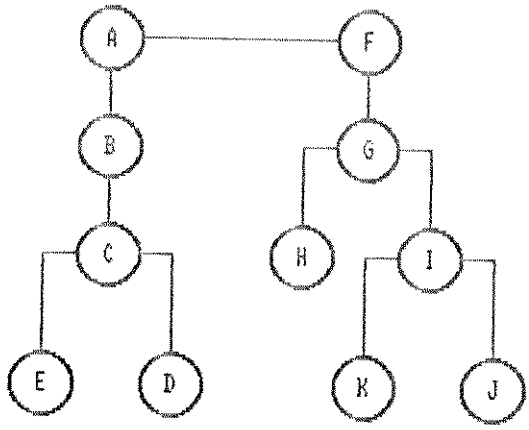
ENDEREÇO	CONTEUDO	APONTADOR 1	APONTADOR 2
e	A	e+1	e+3
e+1	B	e+2	e
e+2	C	e+3	e+1
e+3	D	e	e+2

(d) lista duplamente ligada

Figura 1.2. Estruturas de dados (continuação).



(e.1) Estrutura em anel



(e.2) Representação da árvore equivalente

(e) estrutura em anel

Figura 1.2. Estruturas de dados (continuação).

b) Os resultados devem ser apresentados de maneira clara e rápida;

c) A deteção e o tratamento de erros devem ser eficientes;

d) As técnicas de "software" usadas, devem ser invisíveis, ou seja, para o usuário não interessa saber como o programa foi feito;

e) O acesso a comandos do sistema operacional, tais como os comandos de manipulação, criação e deleção de arquivos, deve ser facilitado.

2. Projeto de Circuitos Integrados Digitais

2.1. Introdução

O projeto de circuitos integrados digitais pode ser dividido em duas fases:

- projeto lógico
- projeto e preparação das artes finais - "layout" - e das máscaras

O projeto lógico inicia-se com a etapa de especificação funcional do circuito e termina com a sua validação através de simulações.

O projeto do "layout" consiste na edição dos vários níveis de máscaras necessárias para o processamento do circuito.

Na figura 1.3 podemos ver o fluxograma seguido no projeto de circuitos integrados digitais.

Existem três metodologias de projeto de circuitos integrados:

- semi-dedicados em "gate-arrays"
- semi-dedicados em células padrão - "standard cells"
- totalmente dedicados - "full custom"

De uma maneira geral, essas três metodologias têm em comum a fase de projeto lógico, diferindo apenas na fase de projeto de "layout", como poderemos ver nos itens seguintes.

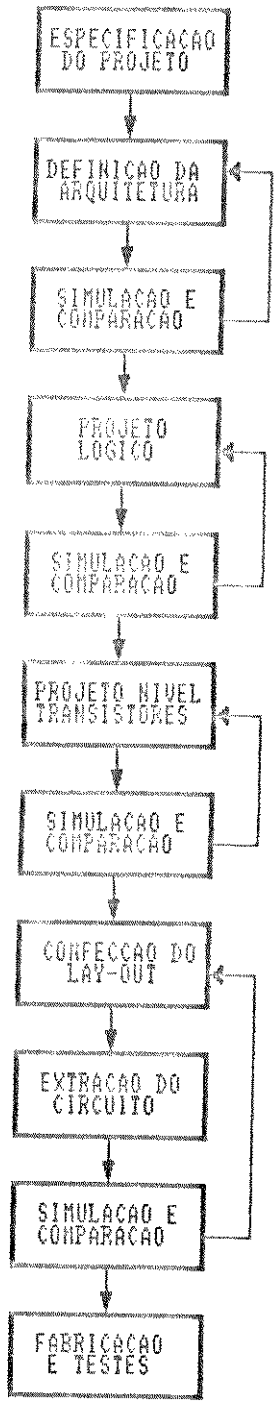


Figura 1.3. Fluxograma de projeto de circuitos integrados.

Com base na especificação funcional do sistema, projeta-se o circuito lógico (descrito em diagrama esquemático). Este diagrama é simulado para se verificar se ele realmente desempenha as funções descritas na especificação. Para analisar o comportamento do circuito em presença de falhas, é feita ainda a simulação de falhas do circuito. Por último é gerado um conjunto de estímulos capazes de detetar eventuais falhas de fabricação. Estes vetores são utilizados na fase de testes do circuito acabado.

A etapa descrita acima, consiste na fase de projeto lógico, semelhante para as três metodologias.

2.2. Metodologia de Projeto de Circuitos Integrados Semi-dedicados com "Gate-arrays"

Um "gate-array" é constituido por uma matriz de células elementares, circundada por células de entrada e saída. Na tecnologia CMOS, por exemplo, as células elementares são compostas por pares de transistores tipo N e tipo P e por tiras de polissilício usadas para interconexão entre as células elementares. Na figura 1.4 podemos ver uma célula elementar com três transistores tipo N e três transistores tipo P.

A matriz é processada até a etapa de metalização, ficando por completar apenas as interconexões, que são individualizadas para cada projeto. Assim o projetista configura o circuito lógico interligando os transistores disponíveis nas células elementares, tendo como grau de liberdade apenas a configuração da camada de metal e o uso dos passantes de polissilício. Os "gate-arrays" têm grande popularidade na implementação de circuitos LSI. Isto advém de uma série de fatores tais como: permitir o uso intensivo de ferramentas de PAC durante o projeto; disponibilidade de blocos lógicos - ou "macro-células" - compatíveis com circuitos TTL e CMOS padrão; curto ciclo de projeto; rápida obtenção de protótipos; e baixo custo de desenvolvimento. Dependendo da demanda e da aplicação do circuito, sua integração em "gate-array" torna-se menos cara e mais rápida, comparativamente às outras metodologias.

Na figura 1.5 podemos ver o fluxo de projeto na metodologia em "gate-array". Este fluxo é dividido em fase de projeto lógico, fase de "layout" e fase de fabricação.

Após a fase de projeto lógico, discutido no item 2.1, entramos na fase de projeto das máscaras de metalização. Esta fase inicia-se com a disposição e interligação dos blocos lógicos fornecidos pelo fabricante, ou seja, alocação e roteamento das macro-células, seguindo o diagrama esquemático obtido na fase de projeto lógico.

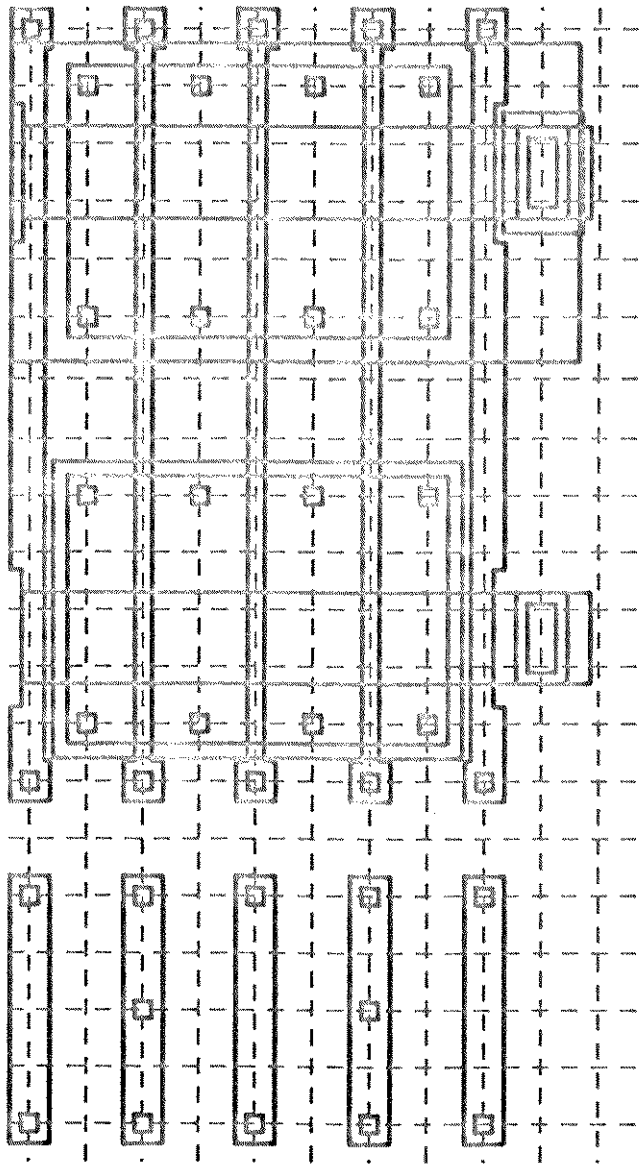


Figura 1.4. Célula elementar de um "gate-array".

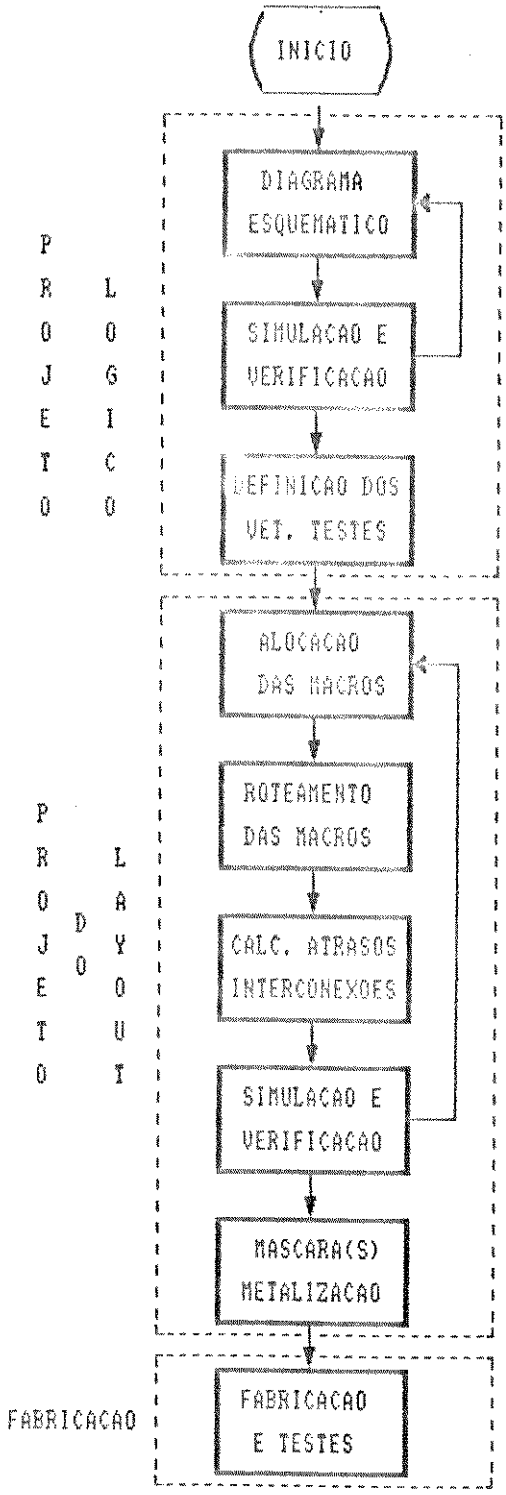


Figura 1.5. Fluxo de projeto na metodologia "gate-array".

Depois, é necessário extrair os atrasos devido às linhas de interconexão, para posterior simulação. Esta simulação é feita para verificar se os atrasos impedem que o circuito atenda às especificações. Se os atrasos não interferirem no funcionamento do circuito, temos as máscaras de metalização prontas para a fabricação, caso contrário, são necessárias alterações na alocação e roteamento das macro-células, voltando-se assim para o início desta etapa.

Na fase de fabricação é feita a metalização das matrizes, o encapsulamento e os testes dos circuitos.

2.3. Metodologia de Projeto de Circuitos Integrados Semi-dedicados com Células Padrão

Um projeto na metodologia de células padrão baseia-se numa biblioteca de células previamente simuladas, testadas e homologadas. Estas células tanto podem ser dispositivos simples, tais como portas lógicas e "latches" ou componentes mais complexos como RAMs, ROMs e PLAs.

Esta metodologia difere da anterior porque nela é necessária a confecção de todos os níveis de máscaras para cada projeto, enquanto na anterior só era necessária a confecção dos níveis de metalização.

O projetista tem à disposição um grau de liberdade maior do que na metodologia de "gate-array". O circuito integrado resultante é otimizado em área e desempenho, o que implica em maiores velocidades e menor consumo de potência. Com esta metodologia é mais fácil o projeto de circuitos complexos, devido a disponibilidade de células de funções como memórias, ULAs, decodificadores em PLAs, etc.

Esta metodologia é mais cara comparada com a de "gate-array", devido ao tempo maior de projeto. A escolha entre as duas metodologias deve levar em conta a quantidade de circuitos a serem produzidos e as características destes circuitos.

O fluxo de projeto na metodologia de células padrão pode ser vista na figura 1.6. A diferença essencial deste fluxograma para o descrito anteriormente é que neste todos os níveis de máscaras devem ser descritos e fabricados.

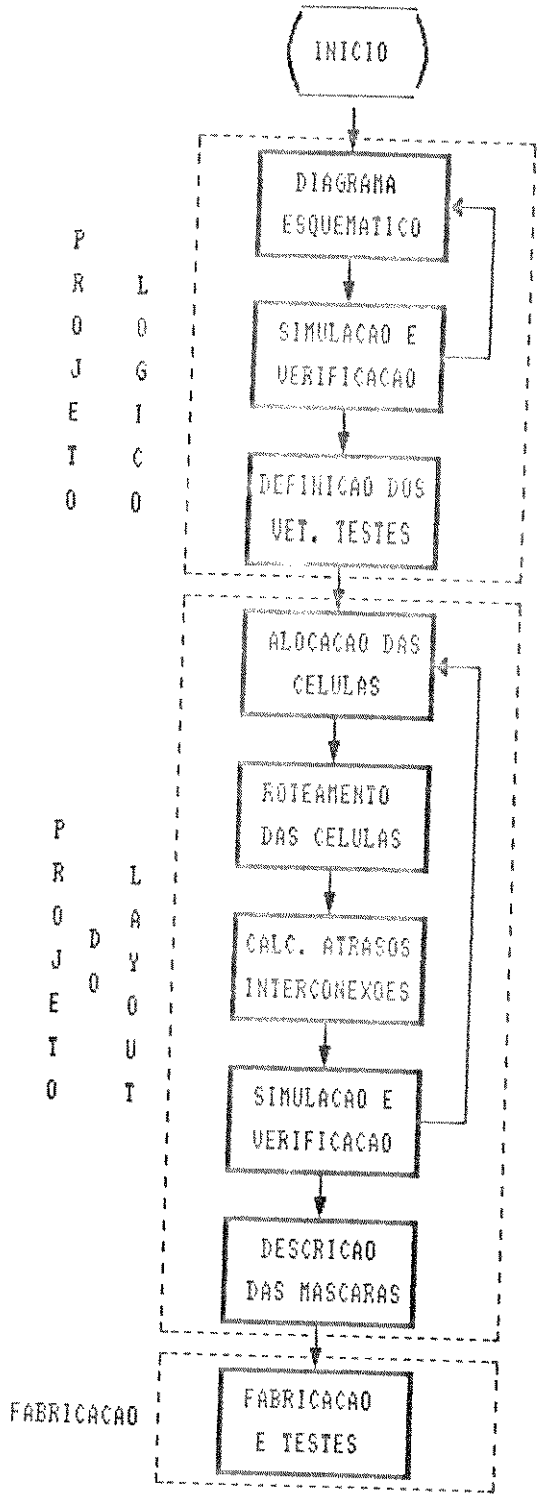


Figura 1.6. Fluxo de projeto na metodologia de células padrão

2.4. Metodologia de Projeto Totalmente Dedicado

Os projetos totalmente dedicados são os que demandam mais tempo para se obter um protótipo, pois os dispositivos do circuito têm que ser projetados. Por outro lado, para um circuito complexo é a metodologia recomendada por otimizar a área de silício e o desempenho do circuito apesar do elevado custo de projeto e de fabricação.

Esta metodologia é principalmente aplicada nos casos em que é grande o volume de produção do circuito, podendo-se assim diluir os custos de projeto e fabricação. Muitas vezes, é a única alternativa disponível quando se trata de circuitos de alta performance ou para aplicações especiais.

O fluxo de projeto na metodologia "full custom" pode ser visto na figura 1.7. A grande diferença deste fluxograma para os anteriores está na fase de projeto de "layout". Após a obtenção do diagrama esquemático, inicia-se a edição dos dispositivos que irão compor os blocos lógicos. O conjunto desses blocos lógicos editados e interligados irá compor os diversos níveis de máscaras. Há a necessidade de se verificar se as dimensões do "layout" não estão violando as regras geométricas de projeto. Se alguma regra for violada, é necessário retornar à edição dos blocos e interconexões.

Após o "layout" ser editado e verificado, é extraido seu circuito equivalente, que é submetido a uma simulação, de forma a se determinar se cumpre as especificações do projeto. Se isto não ocorrer, será necessário retornarmos à ediçào de blocos e interconexões, iniciando-se o ciclo novamente. Se o circuito cumprir as especificações, podemos comsiderar pronto o projeto das máscaras.

2.5. Ferramentas de Software para Projeto de Circuito Integrados

Para qualquer metodologia adotada é importante termos um conjunto de ferramentas de software que nos auxilie no projeto de um circuito integrado. Um conjunto mínimo destas ferramentas é descrito a seguir [1,11,12,13].

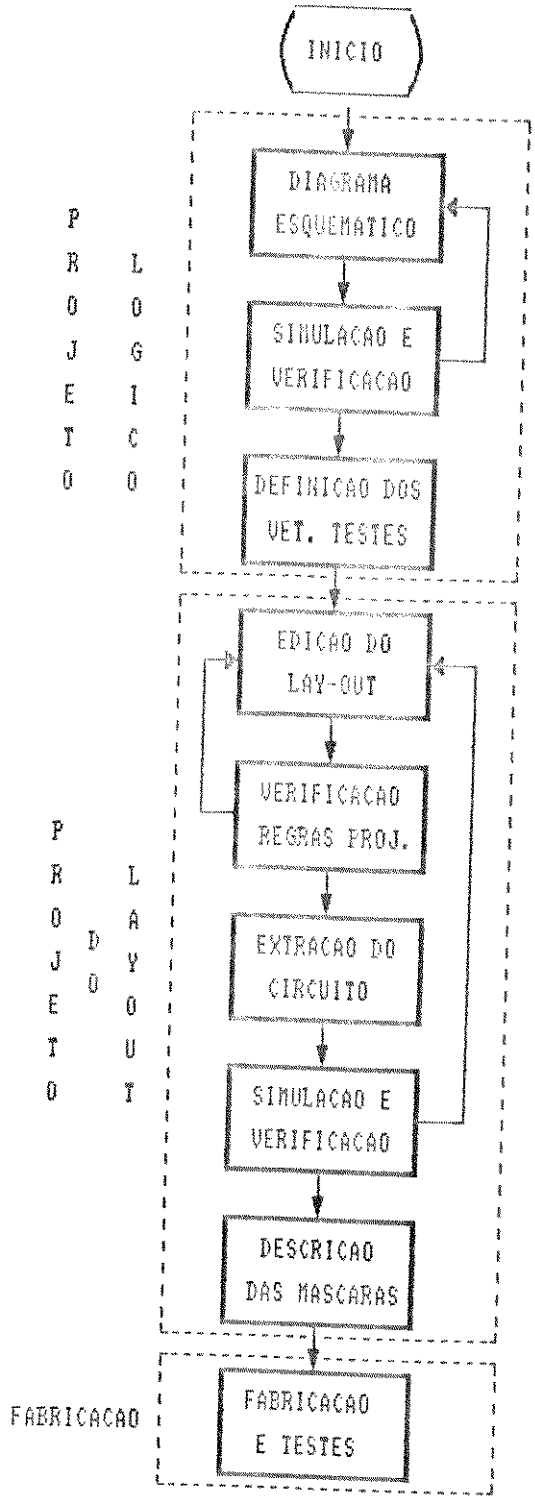


Figura 1.7. Fluxo de projeto na metodologia totalmente dedicado.

a) Editores esquemáticos

A primeira tarefa no projeto de um circuito integrado consiste em descrever a topologia do circuito a partir de seu diagrama lógico, relacionando-se as conexões entre seus elementos. Isto tanto pode ser feito textualmente, como pelo uso de um editor esquemático. Os editores esquemáticos permitem a composição dos dados mais facilmente e com menor incidência de erros. Essas informações são utilizadas nas fases de simulação e composição do "layout" das máscaras dos circuitos. Os editores esquemáticos têm ainda bibliotecas de macro-células de "gate-arrays", de células padrão (para a metodologia de células padrão), e de circuitos integrados comerciais para o projeto de placas de circuitos impressos.

b) Simuladores de circuitos

Os simuladores de circuitos são programas para análise e verificação de circuitos eletrônicos digitais e analógicos, que usam as técnicas convencionais de análise de circuitos, ou seja, a solução de um sistema de equações que relacionam tensões, correntes e resistências (ou condutâncias) dos circuitos. A simulação é efetuada substituindo-se os dispositivos não lineares por modelos que podem ser basicamente de dois tipos: os modelos analíticos, expressos por relações matemáticas entre as variáveis de tensão e corrente em seus terminais, e os circuitos equivalentes, construídos com elementos passivos, geradores de corrente e tensão. Este tipo de programa é caracterizado pela alta precisão de seus resultados e pelo longo tempo de simulação. Ele é usado para verificar em detalhes, pequenos circuitos ou partes críticas de circuitos complexos, com o objetivo de se verificar resultados obtidos com simuladores mais eficientes, mas menos precisos, como é o caso dos simuladores lógicos. O programa de simulação de circuitos mais utilizado é o " Simulation Program with Integrated Circuit Emphasys " - SPICE - desenvolvido no Electronics Research Laboratory da Universidade da Califórnia, em Berkeley [14].

c) Simuladores de temporização

É possível fazermos simulações de circuitos mais rápidas do que as mencionadas acima, às expensas da precisão nos resultados do comportamento dos circuitos. Para isto, utilizam-se simuladores cujos modelos matemáticos dos elementos de circuito são mais simples do que os empregados nos simuladores de circuitos. Em muitos casos, esta simplificação torna a simulação duas ordens de magnitude mais rápida do que a realizada com um simulador de circuito.

d) Simuladores lógicos

Os simuladores lógicos usam como modelos as portas lógicas NOT, AND, OR, NAND e NOR. Parâmetros de tempo, tempos de atraso e tempos de subida e descida podem ser atribuídos às portas lógicas. Atrasos devido a efeitos parasita podem ser também atribuídos aos modelos se se tiver um conhecimento prévio do comportamento dos elementos através de uma simulação a nível de circuitos.

Como num circuito lógico é difícil que todas as portas sejam ativadas simultaneamente, os simuladores lógicos geralmente utilizam o método de controle de eventos (event driver), ou seja, os eventos lógicos são dispostos em uma fila e apenas as portas que apresentam mudança de estado lógico em pelo menos uma de suas entradas, são avaliadas.

e) Simuladores a nível de chave

Os simuladores a nível de chave utilizam como primitivas o modelamento simples de transistores, considerando-os como chaves, e reconhecendo apenas os níveis lógicos alto, baixo e indeterminado (1, 0, X). Esses simuladores são usados para se assegurar se o comportamento do circuito extraído do "layout" corresponde ao esperado, por comparações com os resultados de simulações previamente feitas. A utilização de técnicas de relaxação na avaliação dos estados lógicos dos nós torna este tipo de simulação mais rápida, se comparada às descritas anteriormente.

f) Simuladores de falhas e geradores de vetores de testes

Uma grande preocupação no desenvolvimento de projetos de circuitos integrados digitais com mais de mil portas equivalentes é com respeito ao emprego de técnicas de testabilidade, pois um teste funcional exaustivo torna-se proibitivo pelo longo tempo consumido. Para isto, na fase de projeto lógico, é gerado um conjunto especial de estímulos externos - vetores de teste - capazes de detectar falhas no circuito final, decorrentes dos processos de fabricação. Geralmente, esses vetores são capazes de detectar falhas únicas que redundem em nós com valor lógico constante, ou seja, estado lógico baixo (stuck-at-zero) ou alto (stuck-at-one), em geral devidas a curto-circuitos ou interrupção de fitas de alumínio de interconexões. Estatisticamente sabe-se que estes tipos de falhas cobrem aproximadamente 90% das falhas ocorridas normalmente na fabricação dos circuitos. Esses vetores de teste são gerados pelo Gerador Automático de Vetores de Teste e aplicados nos circuitos fabricados pelos equipamentos automáticos de testes (ATE).

Os simuladores de falhas servem tanto para avaliar o comportamento dos circuitos com a presença de falhas, como também para avaliar os vetores de teste quanto à sua capacidade de detectar falhas.

e) Editores de "layout"

Os editores de "layout" proporcionam ao projetista, através de facilidades gráficas de desenho, a geração, de modo interativo, do "layout" de um circuito integrado. Esses editores devem permitir a manipulação de polígonos em vários níveis, com operações tipo janela, cópia, rotação, deslocamento e eliminação.

f) Alocadores e Roteadores

Os alocadores são programas que fazem automaticamente a disposição das células padrão em um "layout", ou das macro-células em um "gate-array", levando em conta critérios de conectividade, definidos no diagrama esquemático do circuito. Esses programas geralmente trabalham em conjunto com os roteadores, que tem a função de interligar essas células ou macro-células. Convém observar que estes tipos de programas dificilmente conseguem definir 100% da alocação e do roteamento das células ou macro-células do circuito, sendo necessária a intervenção do projetista.

g) Verificadores de Regras de Projeto

Para verificar se o "layout" não transgrediu as regras de projeto, utilizamos os verificadores de regras de projeto. Alguns editores de "layout" fazem a verificação das regras quando da edição.

h) Extratores de Circuito

Os extratores de circuito examinam a inter-relação dos níveis de máscaras a fim de reconhecer os transistores e outros dispositivos que formam o circuito representado por este conjunto de máscaras. A maioria dos algorítmos usa intersecções geométricas de formas para o reconhecimento de dispositivos ativos. Podem também ser extraídos do "layout", resistências, capacitâncias e transistores parasitas, que tendem a prejudicar a performance ou mesmo inviabilizar funcionalmente o circuito. O circuito extraído é novamente simulado para verificar se seu comportamento corresponde às especificações.

2.6. Compiladores de Silício

Um compilador de silício é uma ferramenta automática que converte a descrição funcional de um circuito na descrição a nível de máscaras. Os compiladores de silício, ainda hoje, se baseiam em configurações fixas, na qual o projetista tem liberdade limitada para explorar o projeto. Geralmente esses sistemas utilizam células padrão, matrizes de "gate-arrays" e suas macro-células ou PLAs, em conjunto com um pacote integrado de ferramentas computacionais. Esses sistemas não são voltados para qualquer tipo de arquitetura de circuitos. Não existem ainda, sistemas capazes de projetar automaticamente um circuito VLSI totalmente dedicado.

3. Objetivos deste trabalho.

Como foi visto, para cada etapa do fluxo do projeto é essencial a utilização de ferramentas de auxílio projeto, sem as quais não seria possível a confecção de um circuito integrado. Dentre essas ferramentas podemos salientar aquelas que auxiliam o projetista na descrição do circuito para simuladores. Essas ferramentas, por serem editores esquemáticos gráficos, são importantes pois permitem a redução de erros quando da formação dos arquivos de entrada para simuladores lógicos e para alocadores e roteadores, se utilizarmos as metodologias de "gate-array" ou de células padrão.

O objetivo deste trabalho foi o desenvolvimento do Programa de Entrada Esquemática - PESQA - e do Editor Gráfico de Estímulos Externos - EDTES - que se originou da necessidade de termos estes tipos de ferramentas integradas no Sistema Didático de Projeto - SDP - [14] e no Sistema de Projeto Lógico - SPL - do IM/CTI, mencionados no capítulo II, pois ainda hoje, não existem editores esquemáticos nacionais comerciais, encontrando-se apenas programas oriundos de outros países. Os programas utilizados pelo SDP servirão de protótipos para o desenvolvimento de ferramentas mais eficientes, que comporão o SPL.

Para se chegar a estes protótipos foi necessário o desenvolvimento de um pequeno núcleo gráfico, contendo principalmente transformações geométricas (deslocamento, rotação, etc.) e capacidade de desenho de círculos, pois o compilador utilizado só permite as primitivas **ponto** e **reta** .

Como já foi dito, versões mais enriquecidas, a serem utilizadas no SPL, serão implementadas em computadores com terminais de alta resolução, periféricos tipo "tablet", "plotter" e "mouse", utilizando-se um pacote gráfico que facilite a implementação de características importantes de edição gráfica, tais como, "zoom", "pan", "windows", etc..

II. PROGRAMA DE ENTRADA ESQUEMÁTICA E EDITOR GRÁFICO DE ESTÍMULOS EXTERNOS

1. Desenvolvimento dos Programas PESQA e EDTES.

1.1. Introdução

Para um desenvolvimento rápido e organizado dos programas PESQA e EDTES, procuramos empregar uma metodologia de projeto que fosse simples o suficiente, para minimizarmos o tempo de projeto, e completa o suficiente para que o trabalho fosse bem conceituado e documentado. Para isto o trabalho foi dividido em três fases:

- especificação de requisitos funcionais do "software",
- projeto físico e codificação,
- testes,

detalhadas nos itens seguintes.

A metodologia adotada visou a especificação completa dos programas, para, se necessário, permitir a integração de outras pessoas no decorrer do desenvolvimento do projeto. Esta especificação foi importante também na fase de projeto físico e codificação pois todas as funções do programa foram detalhadas, facilitando suas implementações.

1.2. Especificação funcional de requisitos do "software"

Para o desenvolvimento de um sistema de software é essencial a existência de uma especificação de requisitos funcionais completa, pois além de mostrar o que o sistema é capaz de executar, ela auxilia no projeto físico,na codificação e na documentação, sendo usada como parâmetro de comparação na fase de testes.

Ao especificar os programas PESQA e EDTES, procuramos inicialmente situá-los no contexto a que se destinam, isto é, o sistema ao qual serão integrados, definindo-se claramente seus objetivos, suas aplicações e os benefícios que deles decorrerão. Definimos também termos, abreviações e referências que serão usados no decorrer da especificação.

Descrevemos a seguir, de uma forma geral, os programas fazendo um sumário das funções que deverão compo-los, mostrando estas funções e suas inter-relações através de um diagrama de blocos. Procuramos ainda descrever as características dos usuários potenciais e as restrições de projeto, tais como, limitações de "hardware", limitações da linguagem na qual seriam implementados, interfaces para outros aplicativos, etc. .

Na especificação de requisitos, procuramos detalhar as funções dos programas, cujas descrições são capazes de proporcionar conceitos necessários à fase do projeto físico e codificação. Utilizamos aqui Diagramas de Fluxos de Dados (DFDs) [15], para melhor ilustrar o fluxo de informações através das funções.

Depois da descrição das funções, mostramos como será a estrutura de dados dos programas, descrevendo todas as principais listas, vetores ou matrizes que eles utilizarão.

No que se refere ao desempenho foram especificados os requisitos numéricos estáticos, tais como, números de usuários simultâneos, números de arquivos e quantidades de dados manipulados pelos programas.

Por fim descrevemos a configuração do "hardware" para o qual os programas foram projetado e suas interfaces externas com o usuário, de "hardware" e de "software".

A especificação de requisitos funcionais de "software" é um documento que deve ser frequentemente atualizado durante o desenvolvimento do produto, de forma a que, no fim do projeto, retrate fielmente o programa e suas funções.

A seguir apresentamos os documentos de especificação escritos, para os programas PESQA e EDTES, com base no padrão IEEE [8]. O padrão IEEE foi utilizada por ser uma norma internacional que atende todos os requisitos de especificação de "software" mencionados acima.

1.3. Especificação funcional do Programa de Entrada Esquemática

1.3.1. Introdução

1.3.1.1. Propósito do documento

Este documento tem como finalidade a especificação do Programa de Entrada Esquemática de circuitos digitais do Sistema Computacional de Projeto Lógico do CTI/IM.

1.3.1.2. Escopo do programa

O projeto lógico é a fase inicial de um projeto de circuitos integrados digitais. Essa etapa se inicia com a especificação do circuito a nivel de blocos funcionais, e termina com o projeto a nível de portas lógicas e chaves. Ela exige o uso intensivo de ferramentas computacionais, que além de acelerar o projeto, permitem certificar se um circuito está correto do ponto de vista da função lógica que ele deve desempenhar , eliminando assim, em muitos casos, a necessidade de se montar um protótipo do circuito, com componentes discretos. O fluxograma da fase de projeto lógico pode ser visto na figura 2.1.

Para o projetista de circuitos digitais, torna-se mais interessante projetar e analisar circuitos por meio de diagramas esquemáticos. O Programa de Entrada Esquemática tem a função de auxiliar o projetista na descrição de circuitos lógicos. O circuito é descrito para o computador utilizando recursos gráficos de entrada, tais como, elementos primitivos, elementos de circuitos pré-editados e facilidades gráficas de edição. Estas facilidades gráficas permitem ao usuário do programa, fazer rotações, repetições ou eliminações de partes do desenho.

Após a edição esquemática do circuito, é extraída automaticamente sua descrição lógica, gerando-se um arquivo no formato de entrada para um específico simulador. Esta ferramenta acelera a entrada de dados, diminui a probabilidade de erros e auxilia na documentação do projeto.

O Programa de Entrada Esquemática a ser desenvolvido irá se chamar "PESQA". Ele servirá como interface entre o projetista de circuitos integrados e programas de simulação tais como SIMUL [2,3], HILO-3 [4], SIMUFA [16].

1.3.1.3. Definições e abreviações

- HILO-3 - Programa de simulação lógica com atraso atribuído, simulação de falhas e geração automática de vetores de testes, desenvolvido pela GenRad Inc [4].
- SIMUL - Programa de Simulação Lógica a nível de portas, desenvolvido no CTI/IM [2,3].
- SIMUFA - Programa de Auxílio à Verificação de Vetores de Testes, desenvolvido no CTI/IM [16].
- EDTES - Programa Editor Gráfico de Estímulos Externos, desenvolvido no CTI/IM [esta tese, 6].

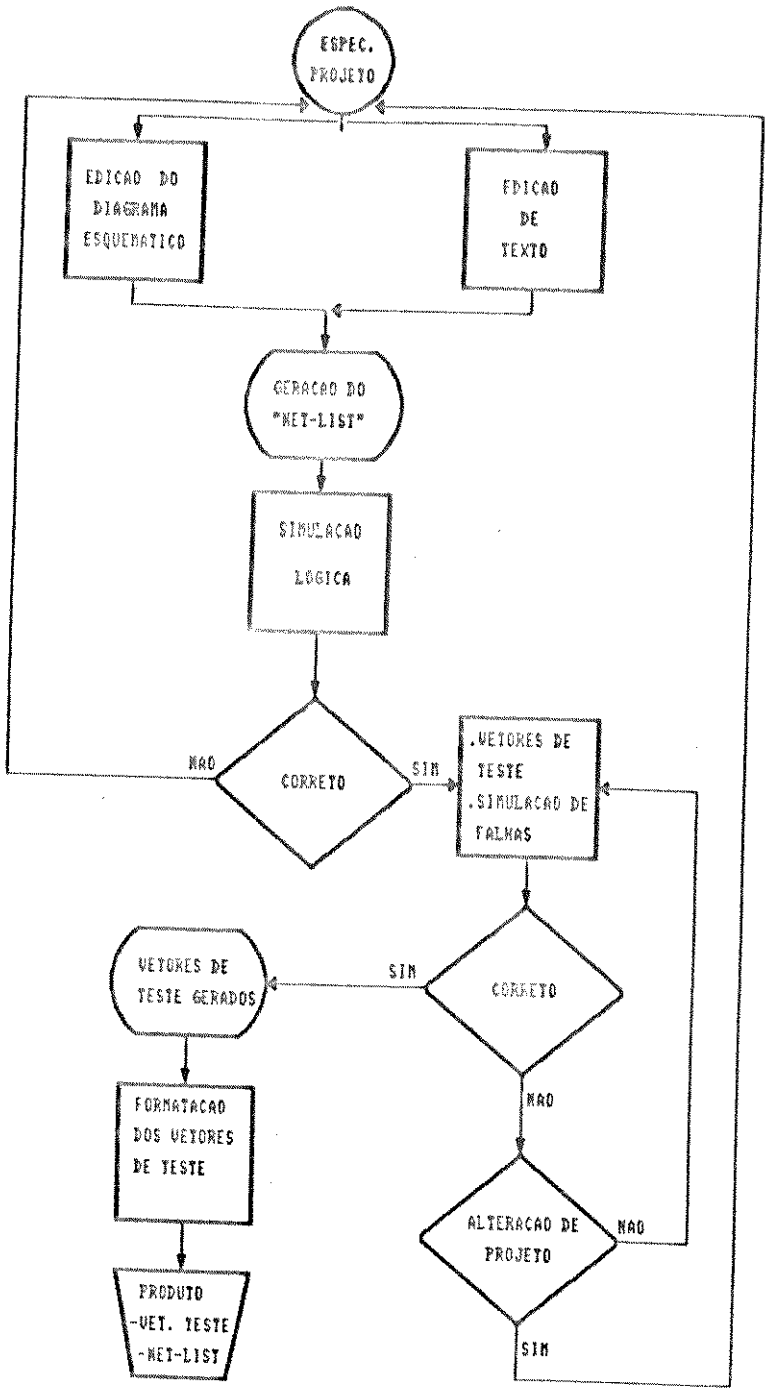


Figura 2.1. Fluxograma da fase do projeto lógico.

. SPL - Sistema Computacional de Projeto Lógico do CTI/IM.

. NET-LIST - Descrição lógica de um circuito na forma de texto, mostrando os elementos e suas interconeções.

1.3.1.4. Referências

. Especificação Funcional do Sistema Computacional de Projeto Lógico (SPL), documento em fase de preparação, CTI/IM.

. Especificação Funcional do Programa Gerenciador de Projetos e Base de Dados, documento em fase de preparação, CTI/IM.

. Especificação Funcional do Programa Formatador de Vetores de Teste, documento em fase de preparação, CTI/IM.

. Especificação Funcional do Programa Editor Gráfico de Estímulos Externos, descrito no ítem seguinte.

. Manual de Usuário do Programa HILO-3, GenRad Inc., October 1985 [4].

. Manual de Operação do Programa SIMUL, Centro Tecnológico para Informática, Julho 1986 [2,3].

. Manual de Operação do Programa SIMUFA, Centro Tecnológico para Informática, Julho 1986 [16].

1.3.1.5. Generalidades

Este documento contém, ainda, as funções do Programa de Entrada Esquemática, especificação de requisitos funcionais e desempenho, especificação de interfaces externas, restrições de projeto e características do computador, onde será implementado, e seus periféricos.

1.3.2. Descrições gerais

1.3.2.1. Perspectiva do produto

O Programa de Entrada Esquemática irá compor o Sistema Computacional de Projeto Lógico (SPL) do Instituto de Microeletrônica da Centro Tecnológico para Informática e o Sistema Didático de Projeto (SDP) utilizado nas Escolas Brasileiro-Argentina de Informática - EBAI - [7]. O SPL será o resultado do desenvolvimento e implantação de ferramentas de auxílio ao projeto lógico em um computador, constituindo assim, um sistema integrado de ferramentas, tendo como objetivo agilizar e organizar as tarefas referentes a esta fase de projeto.

Fazem parte do SPL, além do Programa de Entrada Esquemática, um Editor Gráfico de Estímulos Externos, um Simulador Lógico com atraso atribuido, um Simulador de Falhas, um Gerador Automático de Vetores de Teste, os três últimos integrados no programa HILO-3, um Programa Gerenciador de Projetos e Base de Dados e um Programa Formatador de Vetores de Teste.

Na figura 2.2 podemos ver o Diagrama do Sistema Computacional de Projeto Lógico.

1.3.2.2. Funções do programa

No programa de entrada esquemática a composição do diagrama esquemático é feita mediante facilidades gráficas de desenho (transformações, símbolos, linhas, identificação de elementos e sinais, etc.).

A partir dos diagramas esquemáticos, o programa extrai o "net-list" do circuito, o qual contém a descrição dos elementos lógicos, tais como, AND, NAND, NOR, OR, EXOR, EXNOR, FFD e FFJK, e das interconexões que o compõem. O "net-list" é usado para gerar a entrada de simuladores e também como informação para a fase de projeto de "layout".

Para realizar este trabalho, o Programa de Entrada Esquemática tem como partes principais:

a. interface com o usuário;

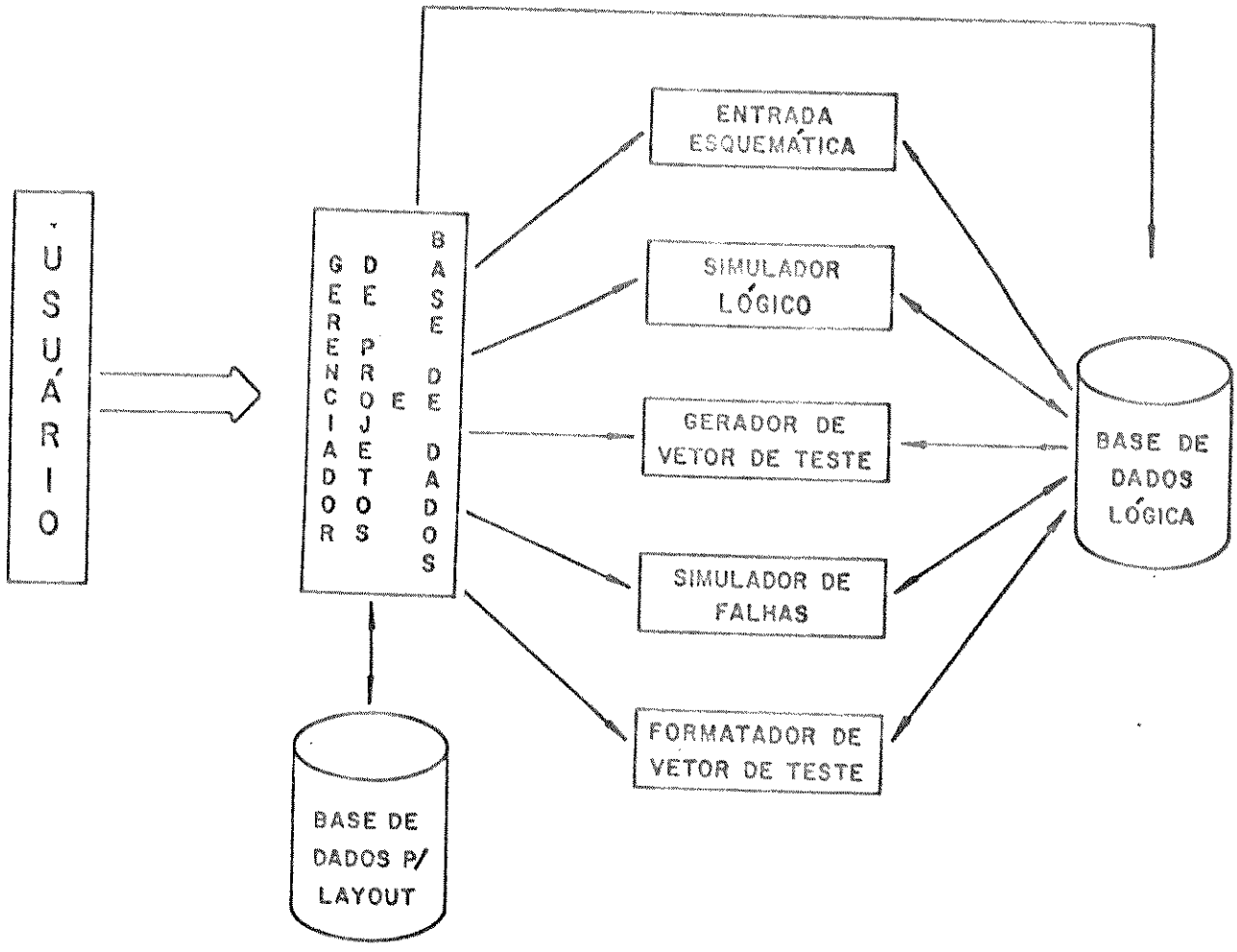


Figura 2.2. Diagrama do Sistema Computacional de Projeto Lógico.

b. edição do circuito :

- edição dos circuito por páginas,
- chamada de células primitivas,
- roteamento manual entre as células,
- verificação de "fan-out" e "fan-in";

c. geração de "net-list" com interface para:

- HILO-3,
- SIMUL,
- SIMUFA;

d. geração de documentação:

- lista de componentes,
- desenho do circuito em plotter ou impressora semi-gráfica.

Na figura 2.3 podemos ver o diagrama de blocos do Programa de Entrada Esquemática.

1.3.2.3. Características dos usuários

O Programa de Entrada Esquemática destina-se aos projetistas de circuitos digitais.

1.3.2.4. Suposições e dependências

O sistema foi desenvolvido supondo que a disponibilidade de "hardware" e "software" fossem as seguintes:

- microcomputador de 16 bits da linha PC;
- sistema operacional compatível com DOS;
- linguagem Turbo Pascal versão 3.01.

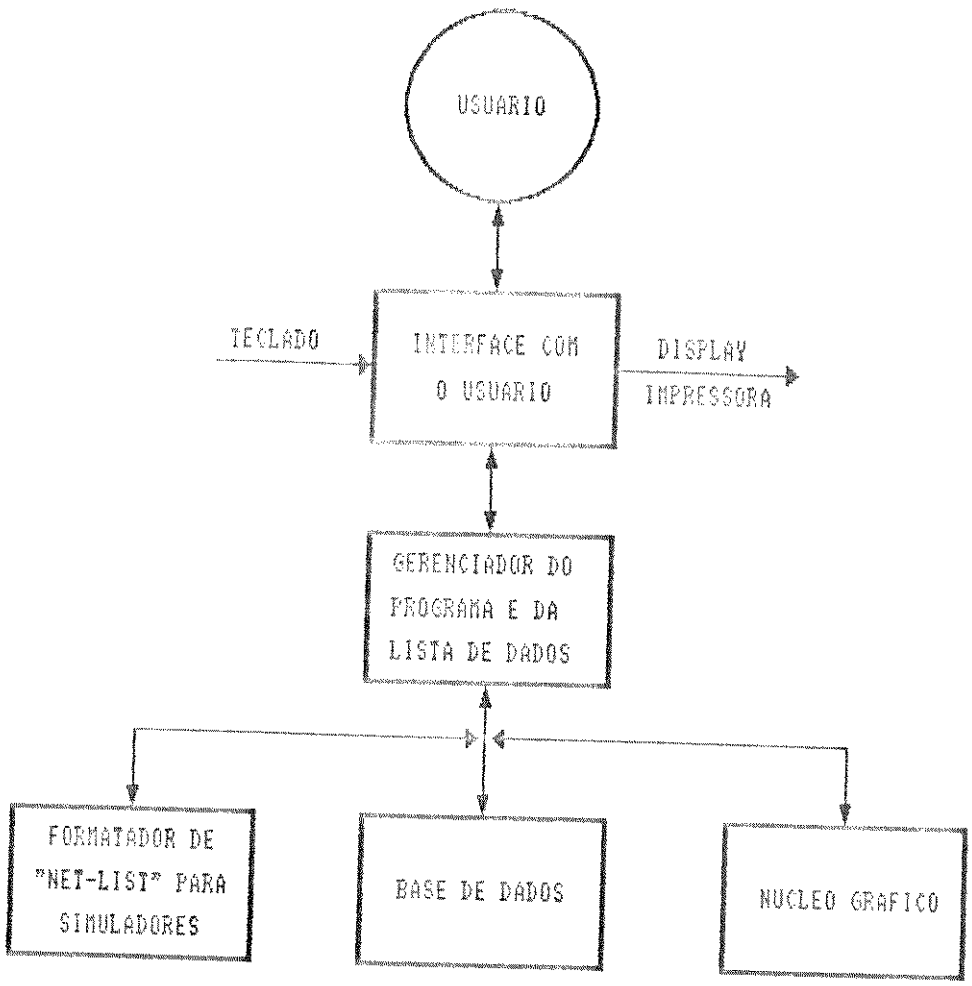


Figura 2.3. Diagrama de blocos do Programa de Entrada Esquemática.

1.3.3. Especificação de requisitos

1.3.3.1. Diagrama de fluxo de dados

No programa PESQA, as funções são classificadas em três níveis hierárquicos:

- funções primárias;
- funções de edição de circuito;
- funções de edição de elementos.

Nas figuras 2.4, 2.5, 2.6 podemos ver os DFDs das funções primárias, das funções de edição de circuito e das funções de edição de elementos, respectivamente.

1.3.3.2. Requisitos funcionais

1.3.3.2.1. Organização de edição

No PESQA, a edição do diagrama esquemático de um circuito é organizada em folhas e páginas. Uma folha de desenho é formada por nove páginas. Cada página tem o tamanho da tela do monitor. A organização em folhas e páginas pode ser vista na figura 2.7. A visualização de uma folha é feita através de janelas como ilustra a figura 2.8.

1.3.3.2.2. Entidades

As funções graficas (rotação, eliminação, repetição) e a função de identificação de elementos e sinais podem ser aplicadas a três tipos de entidades. Toda vez que uma destas funções é acionada, o programa irá perguntar a que tipo de entidade deve ser aplicada a função. Essas entidades são:

- elementos;
- sinais;
- blocos.

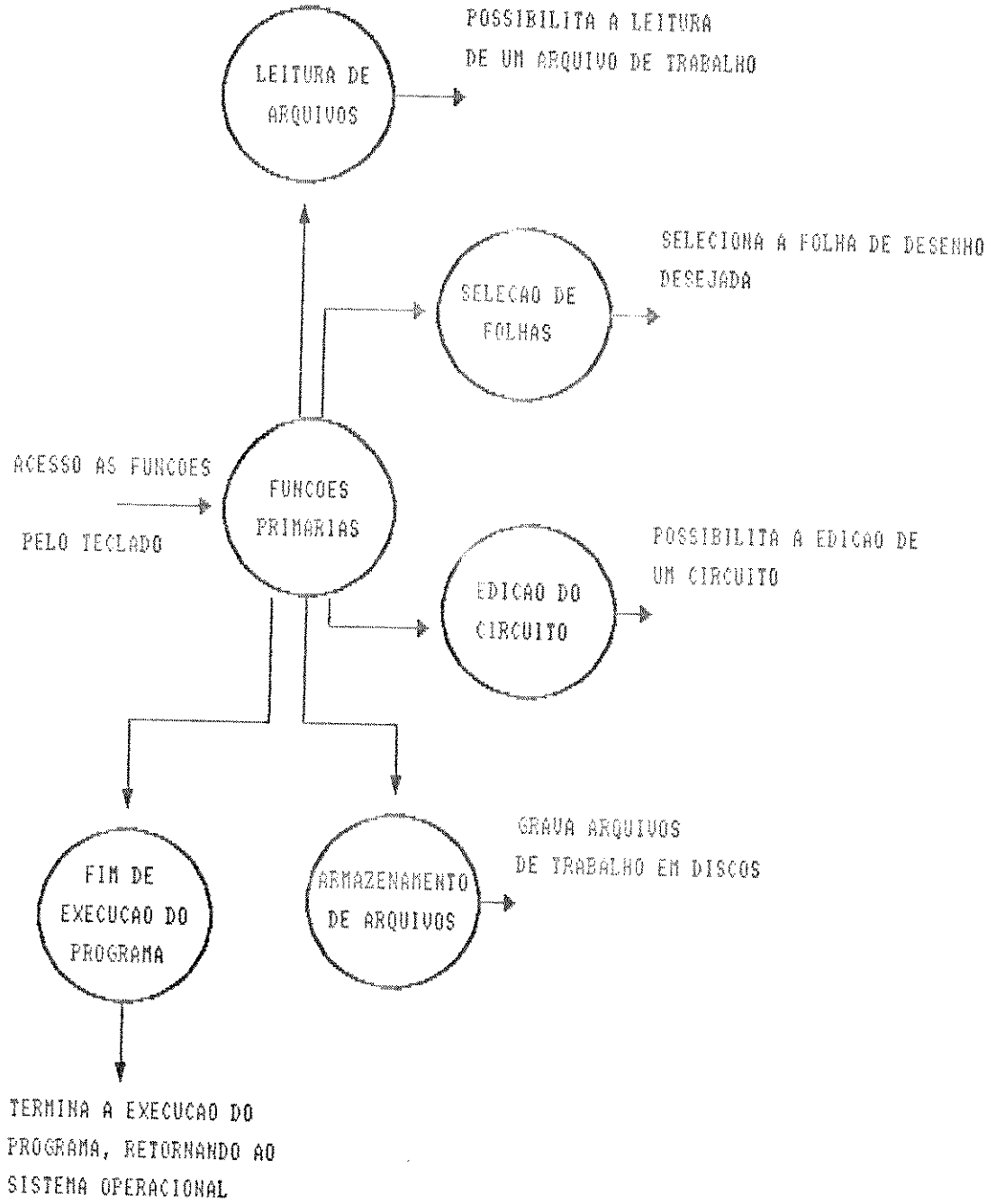


Figura 2.4. Diagrama de fluxo de dados das funções primárias.

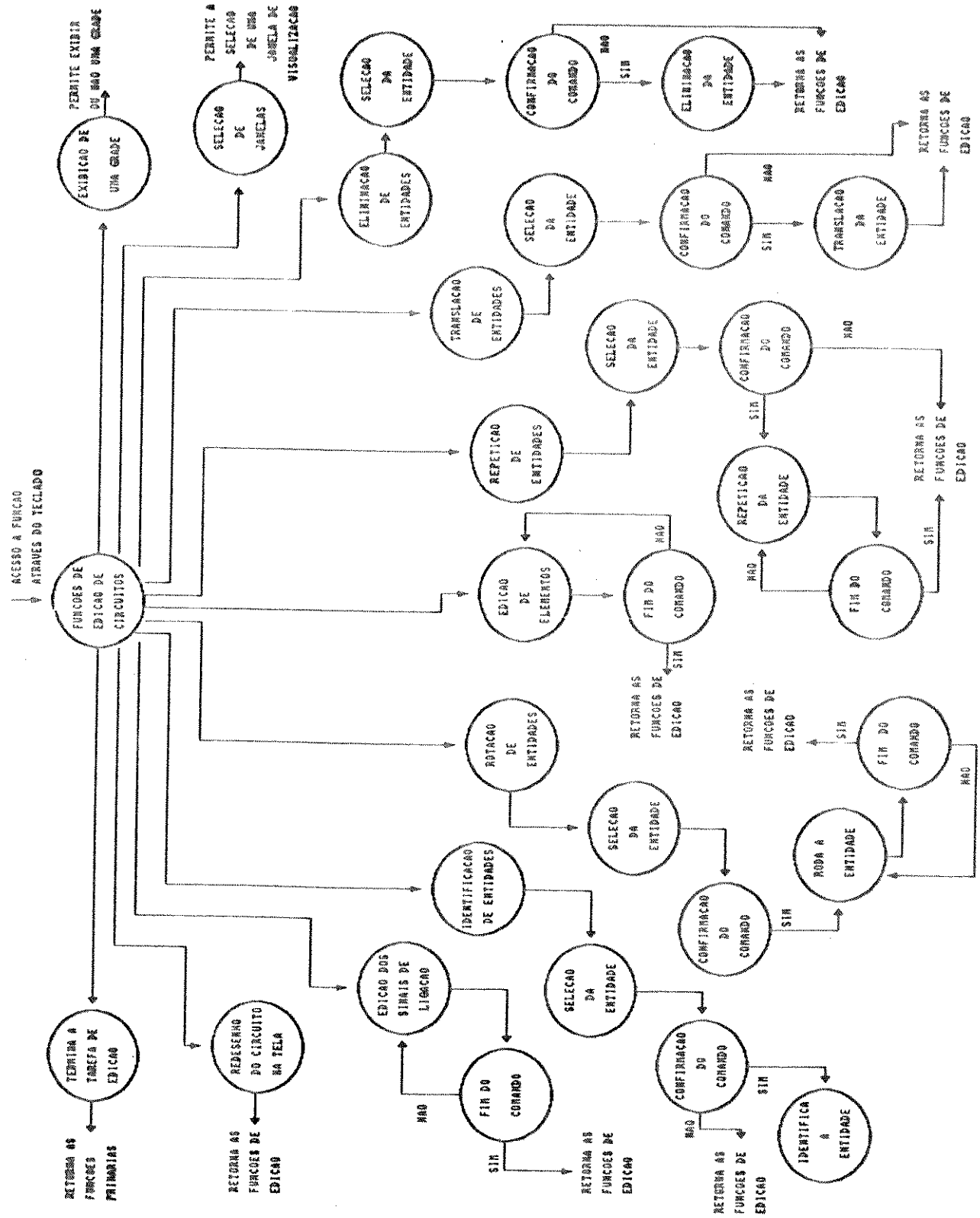


Figura 2.5. Diagrama de fluxo de dados das funções de edição.

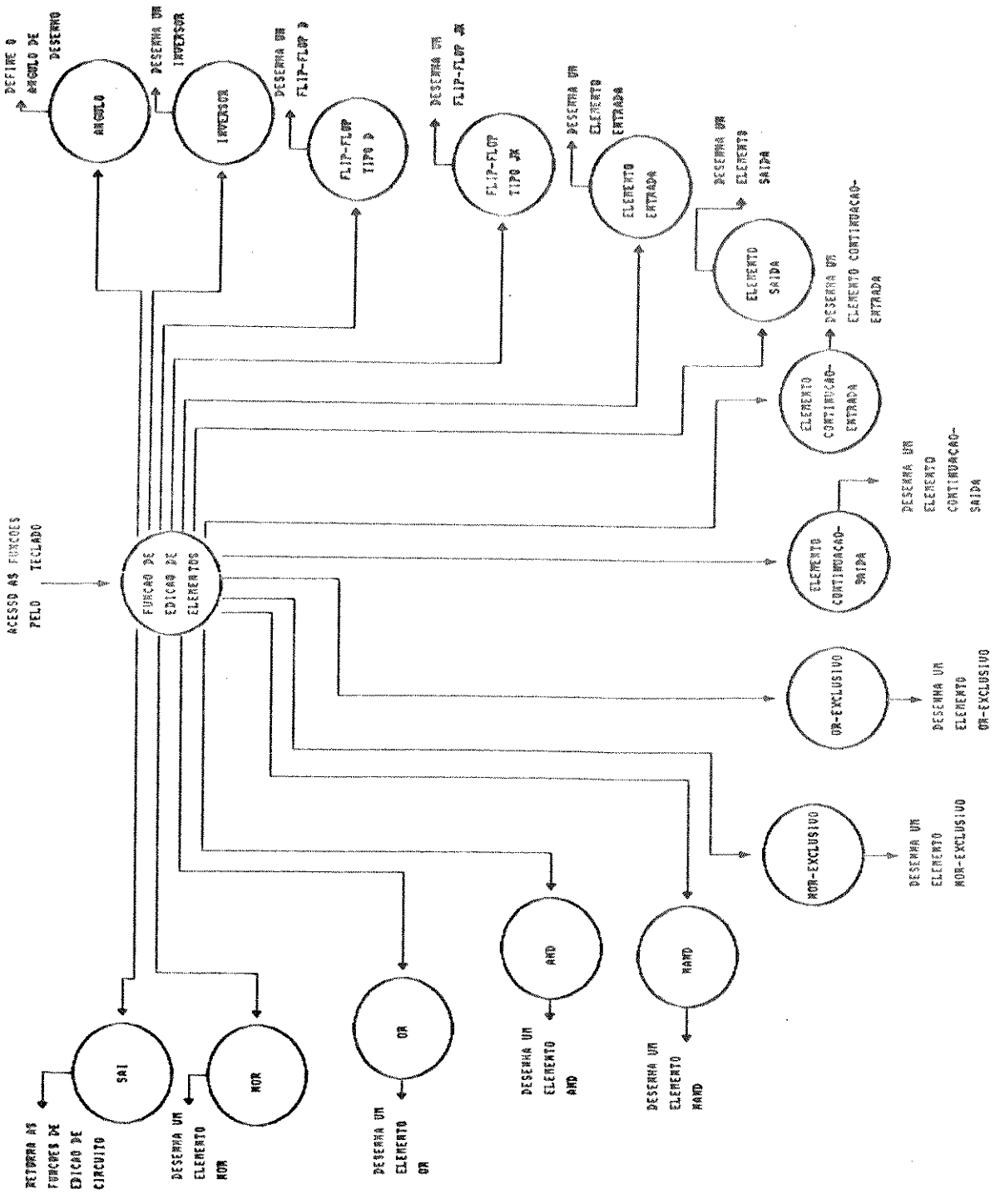


Figura 2.6. Diagrama de fluxo de dados da funções edições de elementos.

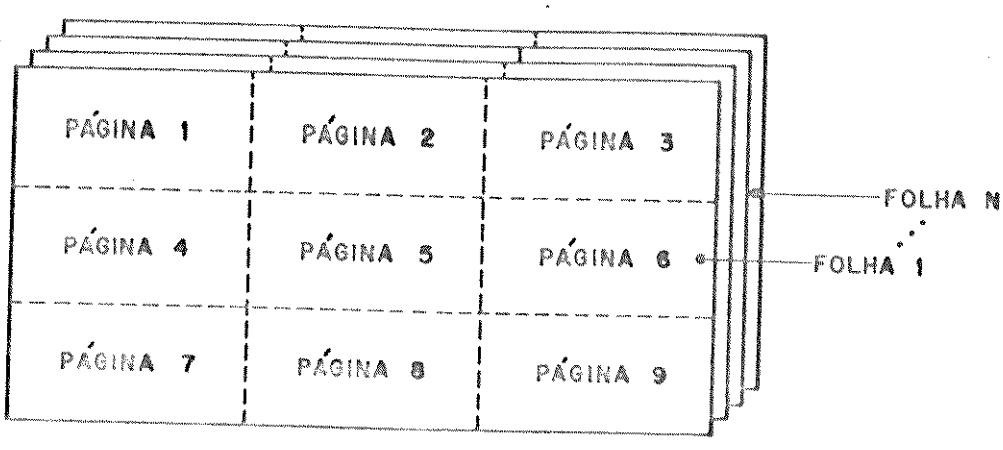


Figura 2.7. Organização em folhas e páginas de um diagrama esquemático.

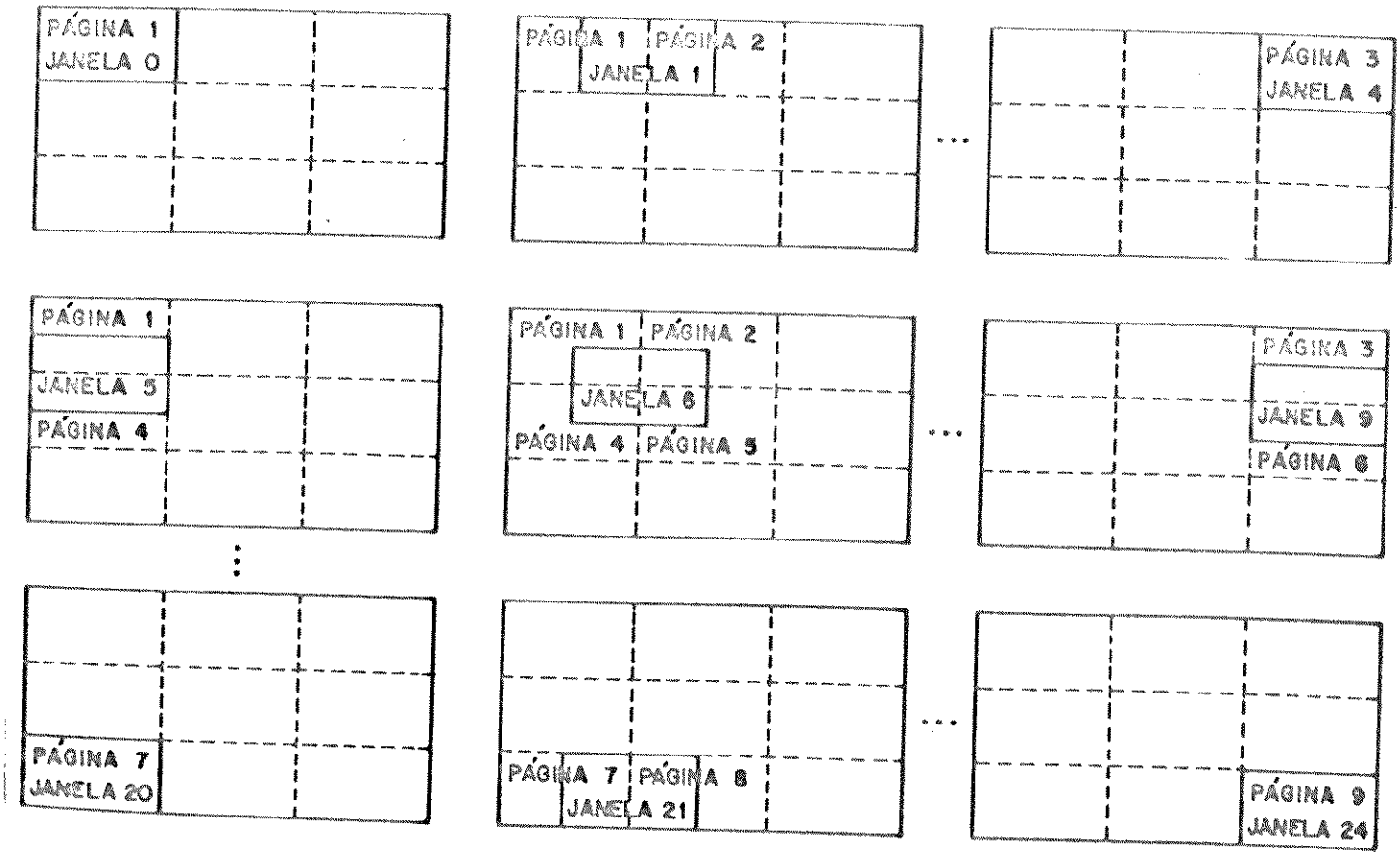


Figura 2.8. Janelas de visualização.

a. Elementos

São as primitivas lógicas, compatíveis com as primitivas usadas nos simuladores lógicos, e as primitivas auxiliares. Primitivas auxiliares são aquelas que não têm função lógica, servindo apenas para indicar entrada, saída, continuação de entrada e continuação de saída de sinais.

A seleção de um elemento para aplicação das funções gráficas é feito posicionando-se o cursor num ponto correspondente a uma de suas entradas.

b. Sinais

São as ligações entre os elementos primitivos.

A seleção de um sinal para aplicação das funções gráficas é feita posicionando-se o cursor em qualquer ponto definido do sinal.

c. Blocos

São conjuntos de um ou mais elementos ou sinais. A seleção de um bloco para aplicação das função gráficas é feito através de um retângulo (janela de seleção) que envolve os elementos ou sinais. Este retângulo é definido através da seleção de dois vértices opostos de uma de suas diagonais.

1.3.3.2.3. Organização da tela de trabalho

A tela do PESQA está dividida em:

- área de cabeçalho;
- área de menu;
- área de diálogo;
- área de trabalho.

a. Área de cabeçalho

É reservada para as informações do nome do projeto, número da folha em que se está trabalhando, número da janela de visualização e posição do cursor em termos de coordenadas x e y.

b. Área de menu

Nesta área são mostradas as funções disponíveis para cada modo de trabalho. O acesso às funções do menu é feito teclando-se uma letra que identifica a função. Esta letra é diferenciada das outras, aparecendo em maiúscula, no menu, à direita na tela. Por exemplo, a função para rotação é acessada pela letra "T", e aparecerá no menu como "roT".

c. Área de diálogo

Esta área é subdividida em:

c.1 Área de mensagem, onde temos as mensagens de advertência do programa.

c.2 Área de entrada de dados, onde são requisitados dados necessários ao programa.

d. Área de trabalho

Área reservada para a edição do diagrama esquemático. Nesta área temos à disposição um cursor em forma de cruz, que se movimenta segundo uma grade, podendo esta grade ser ou não visível. A movimentação do cursor é feita através de teclas, como ilustra a Tabela 2.1. A figura 2.9 mostra a divisão da tela em áreas.

1.3.3.2.4. Funções do programa

As funções disponíveis no Programa de Entrada Esquemática são as seguintes:

1.3.3.2.4.1. Funções de menu principal

a. EDCIR

Esta função tem como propósito criar ou modificar um circuito. A chamada do circuito é feita através da área de diálogo O acesso a essa função é feito pela tecla "E", de "Edição".

b. L ARQ

Esta função tem como propósito ler arquivos de circuitos pré-editados. Toda vez que esta função é acionada, o programa pergunta o nome do arquivo, através da área de diálogo. O acesso a essa função é feito pela tecla "L", de "Ler".

Tabela 2.1.

Teclas para movimentação do cursor

Tecla	Função
↑	O cursor se desloca um passo para cima.
↓	O cursor se desloca um passo para baixo.
-->	O cursor se desloca um passo para a direita.
<--	O cursor se desloca um passo para a esquerda.
PG UP	O cursor se desloca um passo na diagonal, para cima e para a direita.
PG DN	O cursor se desloca um passo na diagonal, para baixo e para a direita.
HOME	O cursor se desloca um passo na diagonal, para cima e para a esquerda.
END	O cursor se desloca um passo na diagonal, para baixo e para esquerda.
+	Aumenta o passo de deslocamento do cursor.
-	Diminui o passo de deslocamento do cursor.

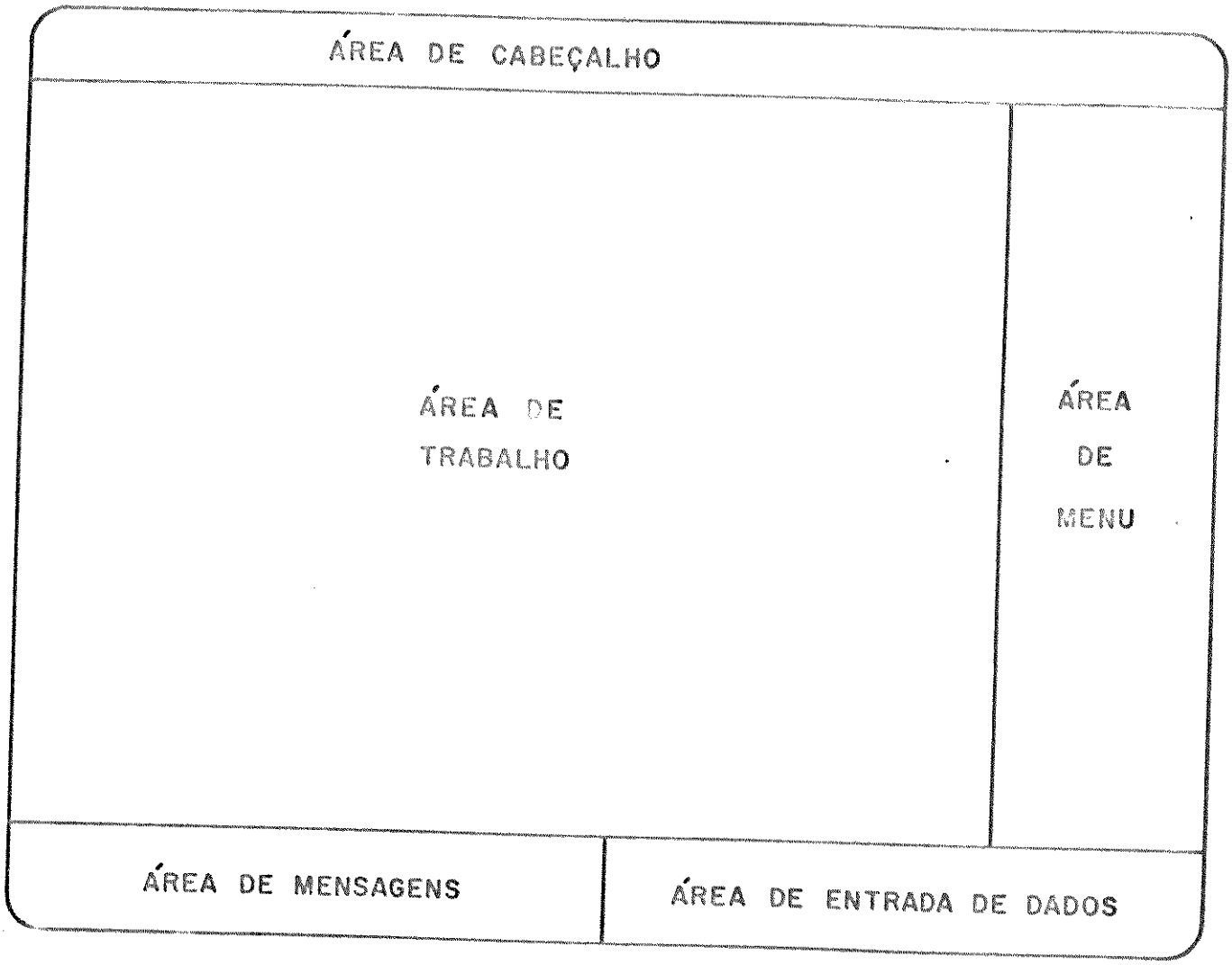


Figura 2.9. Divisão da tela em áreas.

O programa adotará a extensão .PSQ para o arquivo de trabalho passível de leitura. Os arquivos com extensão .PSQ, são arquivos com o formato intermediário PESQA do circuito.

c. IMPRI

Esta função tem como propósito imprimir, em uma impressora gráfica, o diagrama esquemático do circuito. O acesso a essa função é feito através da tecla "R", de "impRessão".

d. FOLHA

Esta função tem como objetivo selecionar uma determinada folha do diagrama esquemático do circuito. O acesso a essa função é feito através da tecla "F", de "Folha". O número da folha desejada é requisitado pelo programa através da área de diálogo.

e. ARMAZ

Esta função tem como objetivo armazenar o diagrama esquemático editado em um arquivo. O nome do arquivo será requisitado através da área de diálogo. O acesso a essa função é feito pela tecla "A", de "Armazenar". A extensão dos nomes dos arquivos pode ser vista no item b.

f. SAI

Esta função tem como objetivo retornar ao sistema operacional, finalizando assim o uso do programa PESQA. O acesso a essa função é feito pela tecla "S", de "Sair".

1.3.3.2.4.2. Menu de Edição de Circuito

a. GRADE

Esta função permite a exibição, na tela, de uma grade. A grade é uma matriz de pontos equidistantes, utilizada para auxiliar o usuário na alocação dos elementos e sinais. A função é acessada através da tecla "G", de "Grade".

b. JANEL

Esta função permite escolher uma janela dentro da folha de trabalho. A função é acessada através da tecla "J", de "Janela", e a escolha da janela é feita através de teclas, como ilustra a Tabela 2.2 .

Tabela 2.2.

Teclas para movimentação da janela de visualização

Tecla	Função
0	Vai para a janela 0
1	Vai para a janela 1
2	Vai para a janela 2
3	Vai para a janela 3
4	Vai para a janela 4
5	Vai para a janela 5
6	Vai para a janela 6
7	Vai para a janela 7
8	Vai para a janela 8
9	Vai para a janela 9
A	Vai para a janela 10
B	Vai para a janela 11
C	Vai para a janela 12
D	Vai para a janela 13
E	Vai para a janela 14
F	Vai para a janela 15
G	Vai para a janela 16

Tabela 2.2.

Teclas para movimentação da janela de visualização (continuação)

Tecla	Função
H	Vai para a janela 17
I	Vai para a janela 18
J	Vai para a janela 19
K	Vai para a janela 20
L	Vai para a janela 21
M	Vai para a janela 22
N	Vai para a janela 23
O	Vai para a janela 24
-->	Anda uma janela para direita
<--	Anda uma janela para esquerda
↑	Anda uma janela para cima
↓	Anda uma janela para baixo
PG UP	Anda uma janela na diagonal, para cima e para direita
PG DN	Anda uma janela na diagonal, para baixo e para direita
HOME	Anda uma janela na diagonal, para cima e para esquerda
END	Anda uma janela na diagonal,para baixo e para esquerda

c. APAG

Esta função permite apagar uma entidade selecionada. Antes de apagar, o programa pede a confirmação do comando através da área de diálogo. Se a resposta for afirmativa, ela elimina a entidade da estrutura de dados e a apaga da tela. A função é acessada através da tecla "P", de "aPagar".

d. DESL

Esta função permite a translação de uma entidade para uma nova posição. A função é acessada pela tecla "D", de "Deslocar". Antes de deslocar, o programa pede a confirmação. Se a resposta for positiva, o programa espera o novo posicionamento do cursor e a digitação da tecla "RET".

Para o deslocamento de elementos é utilizado apenas um ponto de referência. Este ponto, definido pelo par de coordenadas (X,Y), corresponde à entrada do elemento localizada mais à esquerda e embaixo, com o elemento desenhado com ângulo de 0 graus. Sabendo o tipo do elemento e sua nova posição, o programa o apaga de sua posição original, elimina-o da estrutura de dados e o desenha na nova posição (X1,Y1). A figura 2.10 ilustra o deslocamento de um elemento.

Para o deslocamento de sinal o ponto de referência é o ponto inicial do sinal (Xi, Yi). Com a nova posição deste ponto (X1, Y1) podemos calcular os deslocamentos no eixo X e no eixo Y, que serão aplicados aos pontos do sinal, como:

$$(2.1) \qquad Dx = X1 - Xi$$

$$(2.2) \qquad Dy = Y1 - Yi$$

Para o deslocamento dos demais pontos do sinal, usamos:

$$(2.3) \qquad [Xgf\ ,\ Ygf,\ 1] = [Xgi,\ Ygi,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Dx & Dy & 1 \end{vmatrix}$$

onde:
(Xgi, Ygi) representam os pontos do sinal antes da translação e
(Xgf, Ygf) representam os pontos do sinal após a translação.

Com todos os pontos transladados, o programa apaga o sinal da posição original, elimina-o da estrutura de dados e desenha-o na nova posição. A figura 2.11 ilustra o deslocamento de um sinal.

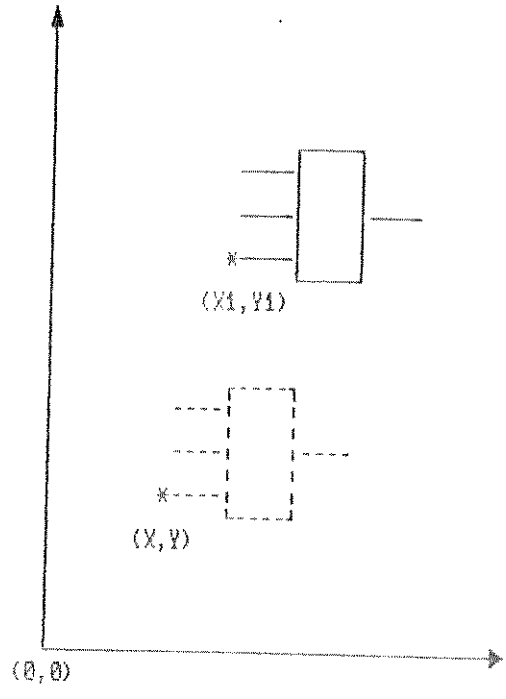


Figura 2.10. Exemplo de deslocamento de elementos.

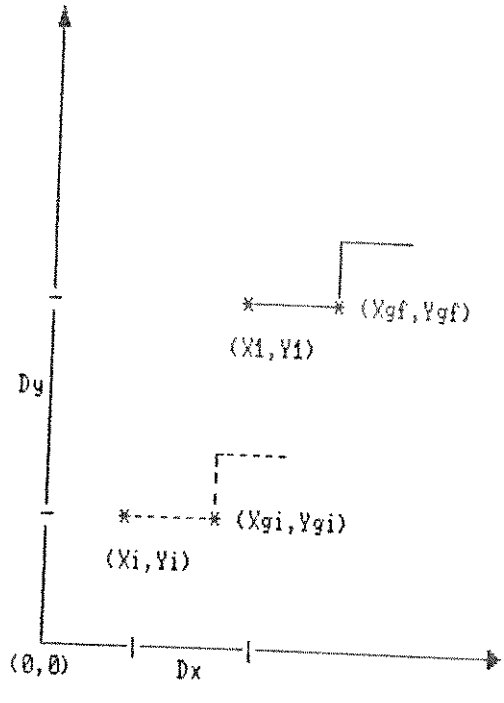


Figura 2.11. Exemplo de deslocamento de sinais.

Para o deslocamento de um bloco, o ponto de referência (Xi, Yi) é o vértice esquerdo inferior do retângulo de seleção de bloco. Seja (Xf, Yf) o ponto da nova posição do bloco, ou seja, o novo ponto do vértice de referência do retangulo, Dx e Dy os deslocamentos correspondentes à translação desse vértice para a nova posição e Dgx e Dgy os deslocamentos correspondentes à translação de pontos de referência de elementos ou sinais para o vértice de referência.

Na translação de um elemento pertencente a este bloco, primeiro desloca-se o ponto de referência (Xgi, Ygi) desse elemento, para o ponto (Xi, Yi) do retângulo. Depois translada-se este ponto para a nova posição (Xf, Yf), referente ao retângulo transladado e por fim, desloca-se o ponto de referência do elemento para sua posição (Xgf, Ygf) relativa dentro do bloco.

Para a translação de sinais, pertencentes a este bloco, procede-se da mesma forma, com a diferença de que todos os pontos (Xgi, Ygi) do sinal devem ser primeiro deslocados para o vértice de referência do retângulo. Esta sequência é equacionada do seguinte modo:

$$(2.4) \quad [X1gi,\ Y1gi,\ 1] = [Xgi,\ Ygi,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -Dgx & -Dgy & 1 \end{vmatrix}$$

$$(2.5) \quad [X2gi,\ Y2gi,\ 1] = [X1gi,\ Y1gi,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Dx & Dy & 1 \end{vmatrix}$$

$$(2.6) \quad [Xgf,\ Ygf,\ 1] = [X2gi,\ Y2gi,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Dgx & Dgy & 1 \end{vmatrix}$$

Com os novos posicionamentos dos elementos, ou sinais, o programa apaga-os da posição original, desenha-os nas novas posições, atualizando as céiulas da estrutura de dados com as novas posições. A figura 2.12 ilustra o deslocamento de um bloco.

e. REPT

Esta função permite a repetição de uma entidade. A função é acessada pela tecla "R", de "Repetir". Antes de repetir, o programa pede a confirmação. Se a resposta for positiva, o programa espera o novo posicionamento do cursor e a digitação da tecla "RET". Esta função termina digitando-se "F", de "Fim".

Aqui utilizamos o mesmo algoritmo descrito no item d, mas não eliminamos a entidade da posição inicial.

f. ROT

Esta função permite a rotação de uma entidade. O ângulo de rotação pode ser 0, 90, 180 ou 270 graus. A função é acessada pela tecla "T", de "roTação". Antes de aplicar a rotação, o programa pede a confirmação. Se a resposta for positiva, o programa pede o ângulo a ser rodado e espera o novo posicionamento do cursor e a digitação da tecla "RET".

Para a rotação de um elemento o ponto de referência corresponde à entrada do elemento localizada mais à esquerda e abaixo. Como temos o ponto de referência, o tipo de elemento e o ângulo sob o qual está desenhado, a rotação deste elemento consiste em repetir o elemento na nova posição com a nova direção. Esta direção é definida como:

(2.7) $$\text{ângulo\_novo} = (\text{ângulo\_inicial} + \text{ângulo\_rotação}) - 360$$

A figura 2.13 ilustra a rotação de um elemento, segundo um ângulo de 90 graus.

Para a rotação de um sinal toma-se como ponto de referência o ponto inicial do sinal (Xi, Yi). Aplicamos uma translação ao sinal para a origem do sistema de referência (0,0), como visto no ítem anterior. Na origem aplicamos uma rotação em todos os pontos do sinal e os transladamos para a nova posição. Esta seqüência pode ser equacionada do seguinte modo:

(2.8) $$Dx = Xi,\ Dy = Yi$$

(2.9) $$[Xig,\ Yig,\ 1] = [Xg,\ Yg,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -Dx & -Dy & 1 \end{vmatrix}$$

$$(2.10) \qquad [X2g,\ Y2g,\ 1] = [X1g,\ Y1g,\ 1] \begin{vmatrix} \cos\theta & -\text{sen}\theta & 0 \\ \text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$(2.11) \qquad [Xgf,\ Ygf,1] = [X2g,\ Y2g,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Dgx & Dgy & 1 \end{vmatrix}$$

onde:

(Xi, Yi) é o ponto de referência do sinal,
(Xg, Yg) representam as coordenadas dos pontos do sinal e
(Xgf,Ygf) representam as novas coordenadas dos pontos.

O programa irá, então, desenhar o novo sinal na nova posição e com a nova direção dada pelo ângulo de rotação. A figura 2.14 ilustra a rotação de um sinal segundo um ângulo de 90 graus.

Para a rotação de um bloco, o ponto de referência (Xi, Yi) é o vértice esquerdo inferior do retângulo de seleção de bloco. As etapas de rotação de um bloco, equacionada a seguir, iniciam-se com uma translação do bloco para a origem do sistema (0,0), equação 2.13, como visto no item anterior. Depois para cada elemento ou sinal é feita uma translação para o novo ponto de referência do bloco e, equação 2.14. Em seguida, é aplicada a rotação desejada, equação 2.15, e o elemento ou sinal é novamente transladado para sua posição relativa no bloco, equação 2.16, e o ponto de referência transladado para sua nova posição, equação 2.17, junto com os elementos ou sinais.

$$(2.12) \qquad Dx = Xi-0 \quad ; \quad Dy = Yi-0$$

$$(2.13) \qquad [X1g,\ Y1g,\ 1] = [Xg,\ Yg,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -Dx & -Dy & 1 \end{vmatrix}$$

$$(2.14) \qquad [X2g,\ Y2g,\ 1] = [Xg,\ Yg,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -Dxg & -Dyg & 1 \end{vmatrix}$$

$$(2.15) \qquad [X3g,\ Y3g,\ 1] = [X2g,\ Y2g,\ 1] \begin{vmatrix} \cos\theta & -\mathrm{sen}\theta & 0 \\ \mathrm{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$(2.16) \qquad [X4g,\ Y4g,\ 1] = [X3g,\ Y3g,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Dxg & Dyg & 1 \end{vmatrix}$$

$$(2.17) \qquad [Xgf,\ Ygf,\ 1] = [X4g,\ Y4g,\ 1] \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Dx & Dy & 1 \end{vmatrix}$$

onde:
(Xg, Yg) representam os pontos dos elementos ou sinais antes da rotação e
(Xgf, Ygf) representam os pontos dos elementos ou sinais após a rotação.

O programa irá então, desenhar os novos elementos ou sinais na nova posição e com a nova direção dada pelo ângulo de rotação. A figura 2.15 ilustra a rotação de um bloco segundo um ângulo de 90 graus.

Esta função termina digitando-se "F", de "Fim".

g. NOM

Esta função permite dar nomes às entidades selecionadas. Se a entidade for BLOCO, ela irá gerar nomes padronizados para todos os elementos e sinais que ainda não identificados. A função é acessada pela tecla "N", de "Nomes". Esta função termina digitando-se "F", de "Fim".

As letras "E" e "S" não podem ser usadas para iniciarem nomes, por estarem reservadas para os nomes padronizados gerados pelo programa.

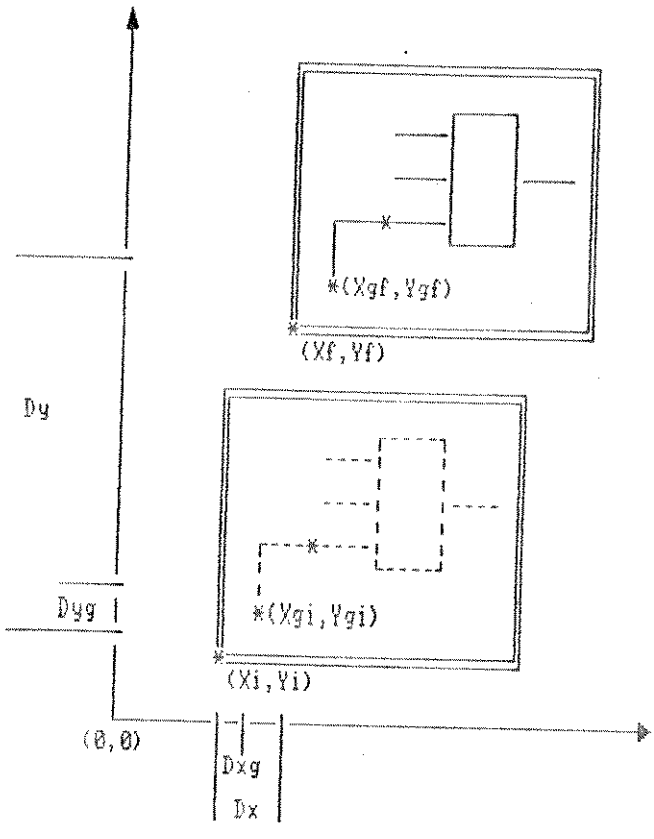


Figura 2.12. Exemplo de deslocamento de blocos.

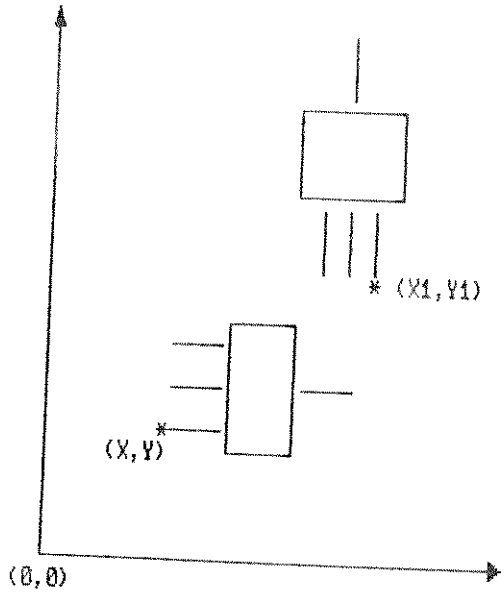


Figura 2.13. Exemplo de rotação de elemento, segundo um ângulo de 90 graus.

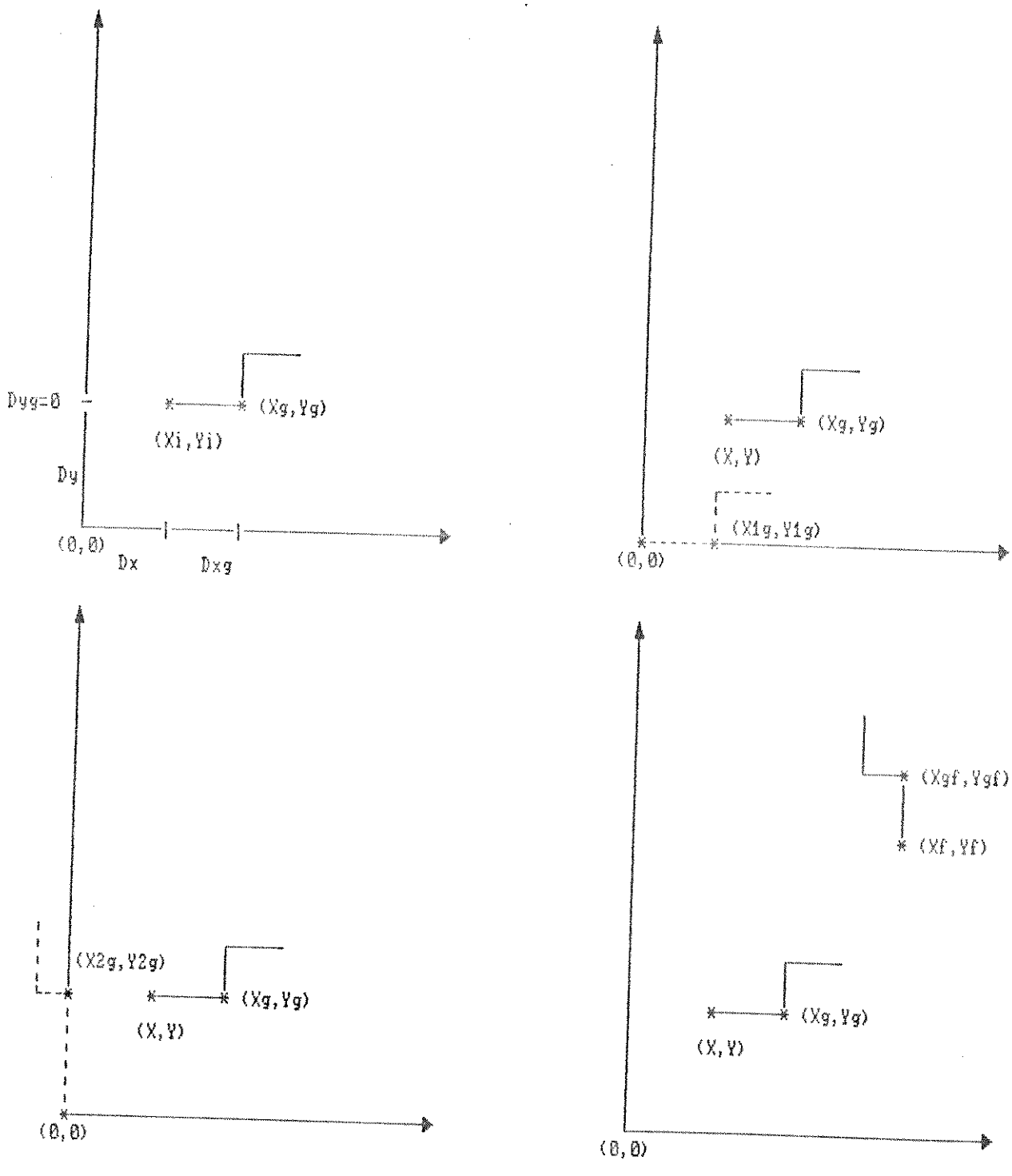


Figura 2.14. Exemplo de rotação de sinal, segundo um ângulo de 90 graus.

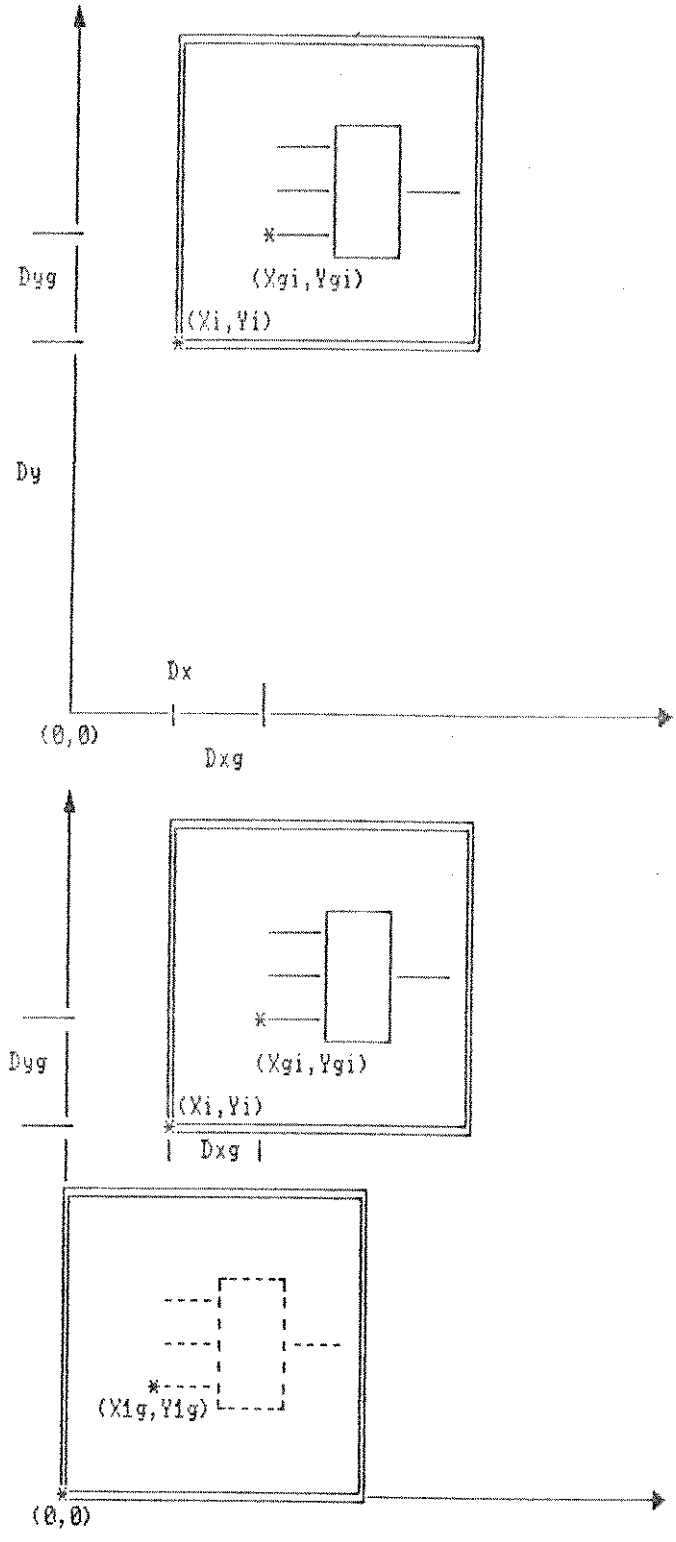


Figura 2.15. (a). Rotação de blocos - Translação do bloco para a origem.

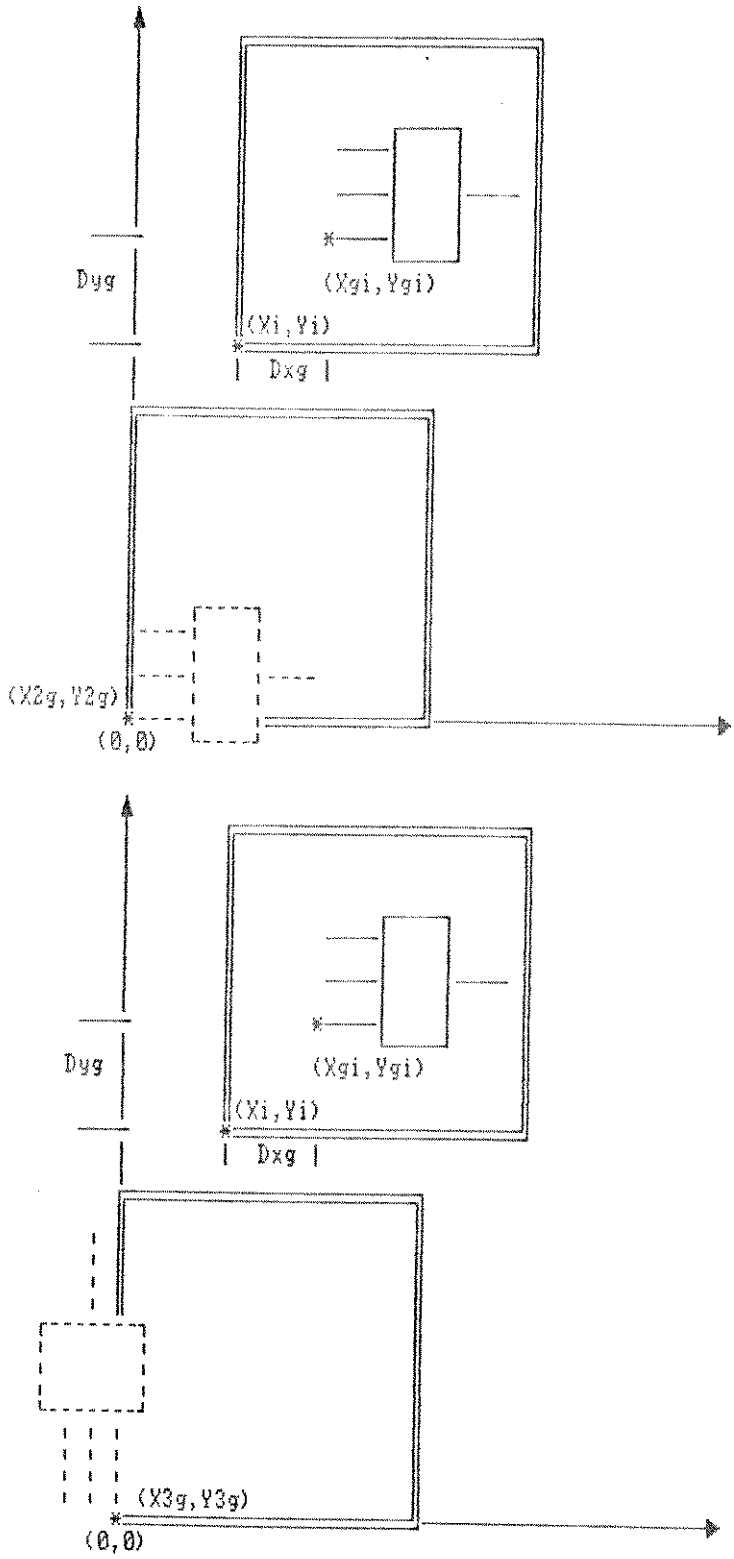


Figura 2.15. (b). Rotação de blocos - Translação dos elementos/ sinais para a origem do bloco e rotação dos elementos/sinais.

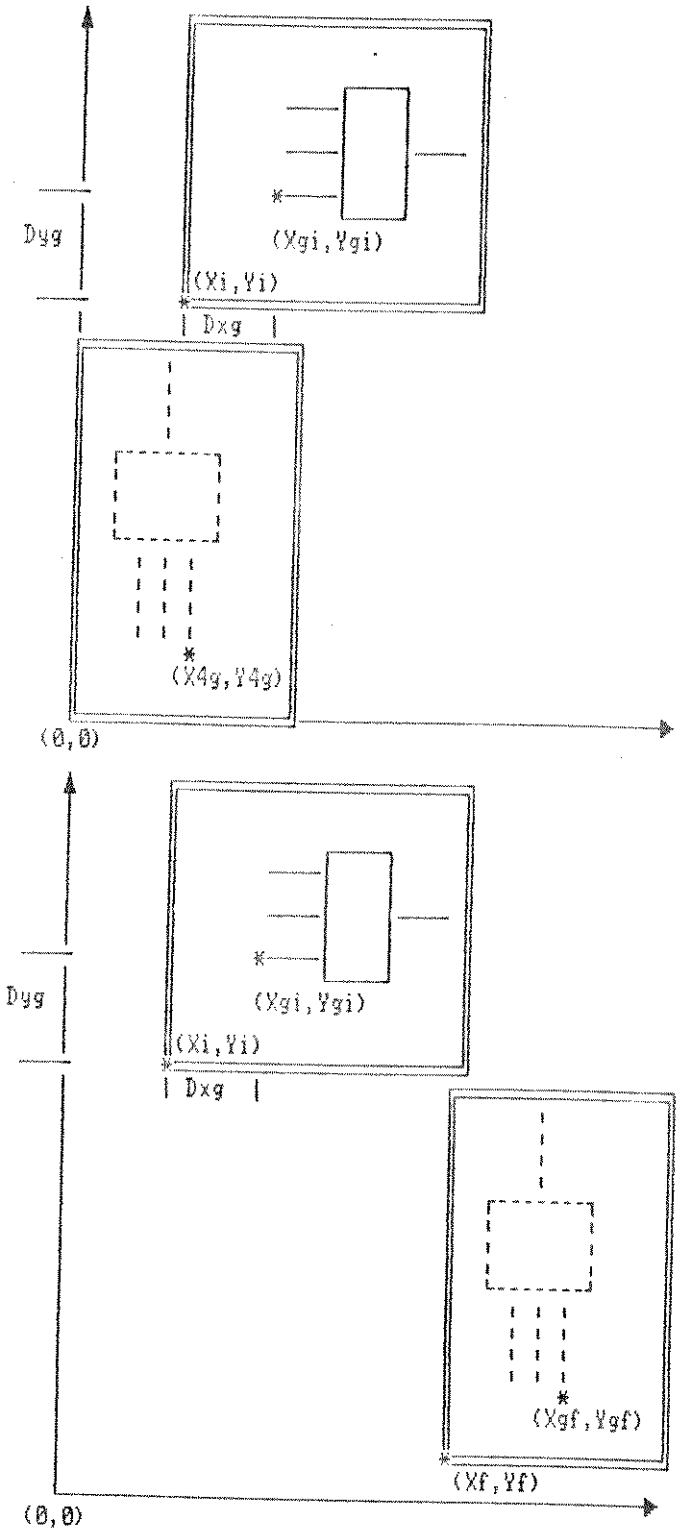


Figura 2.15. (c). Rotação de blocos - Translação dos elementos/sinais para a posição relativa no bloco e translação do bloco para sua nova posição .

h. LIGA

Esta função permite fazer as ligações (sinais) entre os elementos sendo acessada digitando-se "L", de "Ligações". Existem dois modos de se editar um sinal. O primeiro é através da entrada dos pontos na área de diálogo. Para usar este modo, deve-se digitar "P", de "Ponto", e o programa pedirá os pares de coordenadas dos pontos que compõem o sinal. Após a entrada da última coordenada, deve-se teclar "F", de "Fim", para indicar que não há mais pontos a serem introduzidos. O segundo modo é indicar os pontos através do cursor. Para isso, leva-se o cursor para o ponto onde se deseja começar o sinal e digita-se "RET" ou "X". Será usado "X" se o sinal começar em um ponto de outro sinal e se os dois sinais estiverem em contato. Caso contrário deve-se usar a tecla "RET". Os pontos intermediários serão marcados por "X" se houver cruzamento e contato com outro sinal, caso contrário deve-se usar a tecla "C". O ponto final do sinal é marcado com a tecla "RET".

Esta função termina digitando a tecla "F", de "Fim", após o término da edição de sinal.

i. EDELM

Esta função permite o acesso aos elementos primitivos. Ela é acessada pela tecla "M", de "eleMentos". Associada a essa função, temos as seguintes opções:

i.1. ANGUL

Esta função permite a escolha do ângulo sob o qual deve ser desenhado o elemento (0, 90, 180 ou 270 graus). O valor por omissão (default) usado pelo programa é 0 graus. A função é acessada pela letra "L", de "ânguLo".

i.2. INV

Esta função desenha um inversor na posição do cursor e segundo uma direção escolhida. Ela é acessada pela tecla "I", de "Inversor".

i.3. FFD

Esta função desenha um flip-flop tipo D, na posição do cursor e segundo uma direção escolhida. Ela é acessada pelas teclas "F" e "D", acionadas sequencialmente.

i.4. FFJK

Esta função permite o desenho de um flip-flop JK na posição do cursor e segundo uma direção escolhida. Ela é acessada através das teclas "J" e "K", acionadas em sequência.

i.5. ENTRA

Esta função permite o desenho de um elemento que representa uma entrada do circuito, na posição do cursor e segundo uma direção escolhida. Ela é acessada através do tecla "R", de "entRada".

i.6. SAIDA

Esta função permite o desenho de um elemento que representa a saída do circuito, na posição do cursor e segundo uma direção escolhida. Ela é acessada através da tecla "D", de "saíDa".

i.7. CONTE

Esta função permite o desenho de um elemento, que indica que um sinal é a continuação de outro sinal desenhado na pagina anterior, na posição do cursor e segundo uma direção escolhida. Ela é acessada através das teclas "C" e "E", de "Continuação Entrada", acionadas em sequência.

i.8. CONTS

Esta função permite o desenho de um elemento, que indica que um sinal continua em uma das páginas posteriores, na posição do cursor e segundo uma direção escolhida. Ela é acessada através das teclas "C" e "S", de "Continução Saída", acionadas em sequência.

i.9. RESTE

Esta função permite redesenhar todos os elementos já editados. Ela é acessada teclando-se "E" duas vezes, de "rEstaura Elemento".

i.10. SEGUE

Esta função permite mostrar o outro menu de edição de elementos. Ela é acessada pela tecla "U", de "segUe".

i.11. EXOR

Esta função permite desenhar um OR-Exclusivo, na posição do cursor e segundo uma direção escolhida. Ela é acessada pela teclas "X" e "O", de "Exclusive Or", acionadas em seguida.

i.12. EXNOR

Esta função permite desenhar um NOR-Exclusivo, na posição do cursor e segundo uma direção escolhida. Ela é acessada pelas teclas "X" e "N", de "Exclusive Nor", acionadas seguidamente.

i.13. AND N

Esta função permite desenhar, na posição do cursor e segundo uma direção escolhida, um AND de N entradas, sendo que N pode variar de 2 a 5. Ela é acessada pela tecla "A" e pela tecla correspondente ao número de entradas.

i.14. NAN N

Esta função permite desenhar, na posição do cursor e segundo uma direção escolhida, um NAND de N entradas, sendo que N pode variar de 2 a 5. Ela é acessada pelas teclas "N" e "A" e pela tecla correspondente ao número de entradas.

i.15. OR N

Esta função permite desenhar, na posição do cursor e segundo uma direção escolhida, um OR de N entradas, sendo que N pode variar de 2 a 5. Ela é acessada pela tecla "O" e pela tecla correspondente ao número de entradas.

i.16. NOR N

Esta função permite desenhar na posição do cursor e segundo uma direção escolhida, um NOR de N entradas, sendo que N pode variar de 2 a 5. Ela é acessada pelas teclas "N" e "O" e pela tecla correspondente ao número de entradas.

i.17. VOLTA

Esta função permite voltar ao outro menu de edição de elementos. Ela é acessada pela tecla "V".

i.18. SAI

Esta função permite sair do modo de edição de elemento. Ela é acessada pela tecla "S".

j. REST

Esta função permite redesenhar sinais ou elementos. Ela é acessada pela tecla "E" seguida respectivamente pela tecla "S", para redesenhar sinais, "E" para redesenhar elementos e "B" para redesenhar elementos e sinais (blocos).

k. SAI

Esta função permite sair do Programa de Entrada Esquemática e retornar ao sistema operacional. Ela é acessada pela tecla "S", de "Sai".

1.3.3.3. Estruturas de dados

No programa PESQA adotamos a filosofia de lista ligada, conforme descrevemos no item 1.3 do Capitulo I. Para armazenar os dados, utilizamos quatro listas de diferentes tipos:

- de coodenadas (x, y);
- de elementos;
- de sinais;
- de folhas.

A lista de coordenadas faz parte da célula da lista de elementos e da lista de sinais. Estas duas listas, por sua vez, fazem parte da célula da lista de folhas.

1.3.3.3.1. Lista ligada para coordenadas

Nesta estrutura, vista na figura 2.16, armazenamos todos os pontos em termos de suas coordenadas. Sua célula básica é composta por:

- coordenada x : variável inteira;
- coordenada y : variável inteira;
- apontador para a próxima célula;
- apontador para a célula anterior.

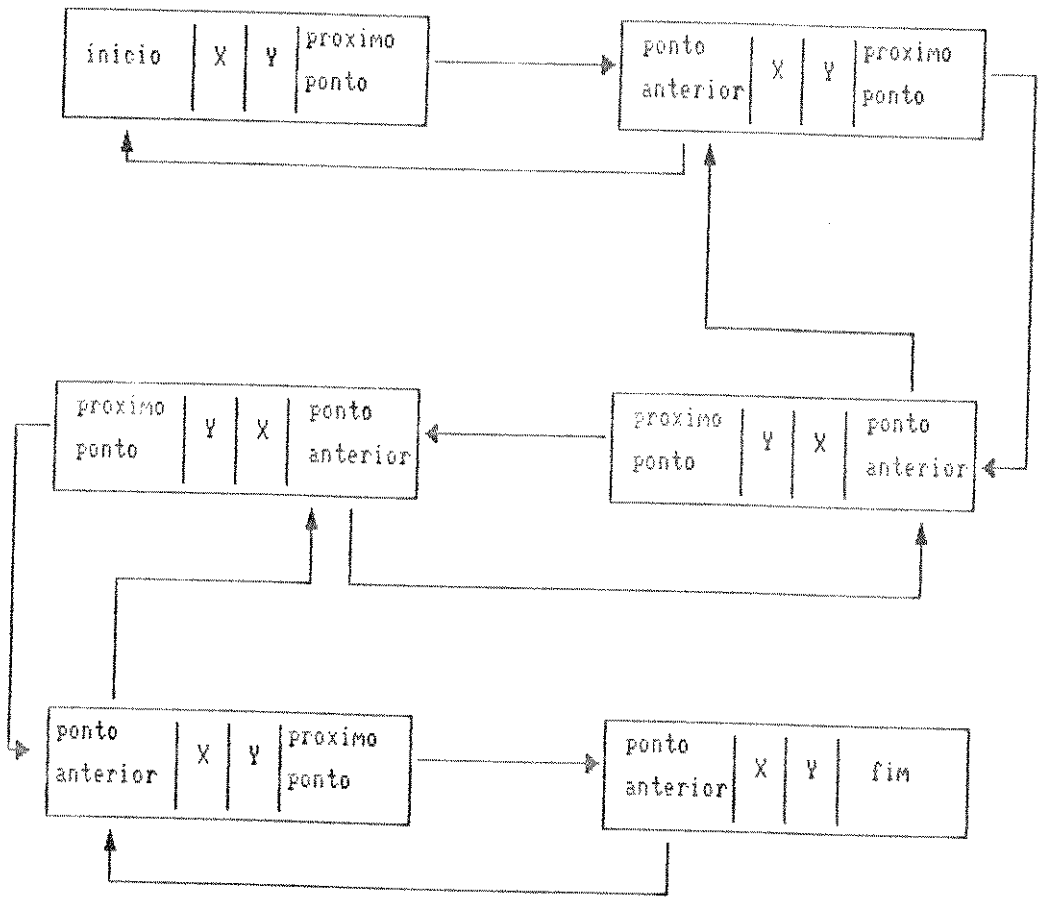


Figura 2.16. Lista ligada para coordenadas.

1.3.3.3.2. Lista ligada para elementos

Neste tipo de lista, ilustrada na figura 2.17, são armazenadas todas as informações referentes aos elementos. A célula básica deste tipo de lista é formada por:

- nome do elemento : variável tipo cadeia de 5 caracteres;
- tipo do elemento : variável inteira;
- ângulo de desenho : variável inteira;
- lista ligada de coordenadas para os pontos das entradas do elemento;
- lista ligada de coordenadas para os pontos das saídas do elemento;
- apontador para a próxima célula;
- apontador para a célula anterior.

1.3.3.3.3. Lista ligada para sinais

Neste tipo de lista armazenamos os pontos que compõem um sinal e as informações que caracterizam este sinal. A célula básica deste tipo de lista é formada por:

- lista ligada de coordenadas para os pontos do sinal;
- lista ligada de coordenadas que armazenam os pontos de cruzamento com contato com outros sinais;
- variável tipo booleana que indica se este sinal é cruzado, com contato, por outro sinal;
- nome do sinal : variável tipo cadeia de 5 caracteres;
- apontador para a próxima célula;
- apontador para a célula anterior.

A figura 2.18 ilustra a estrutura desta lista.

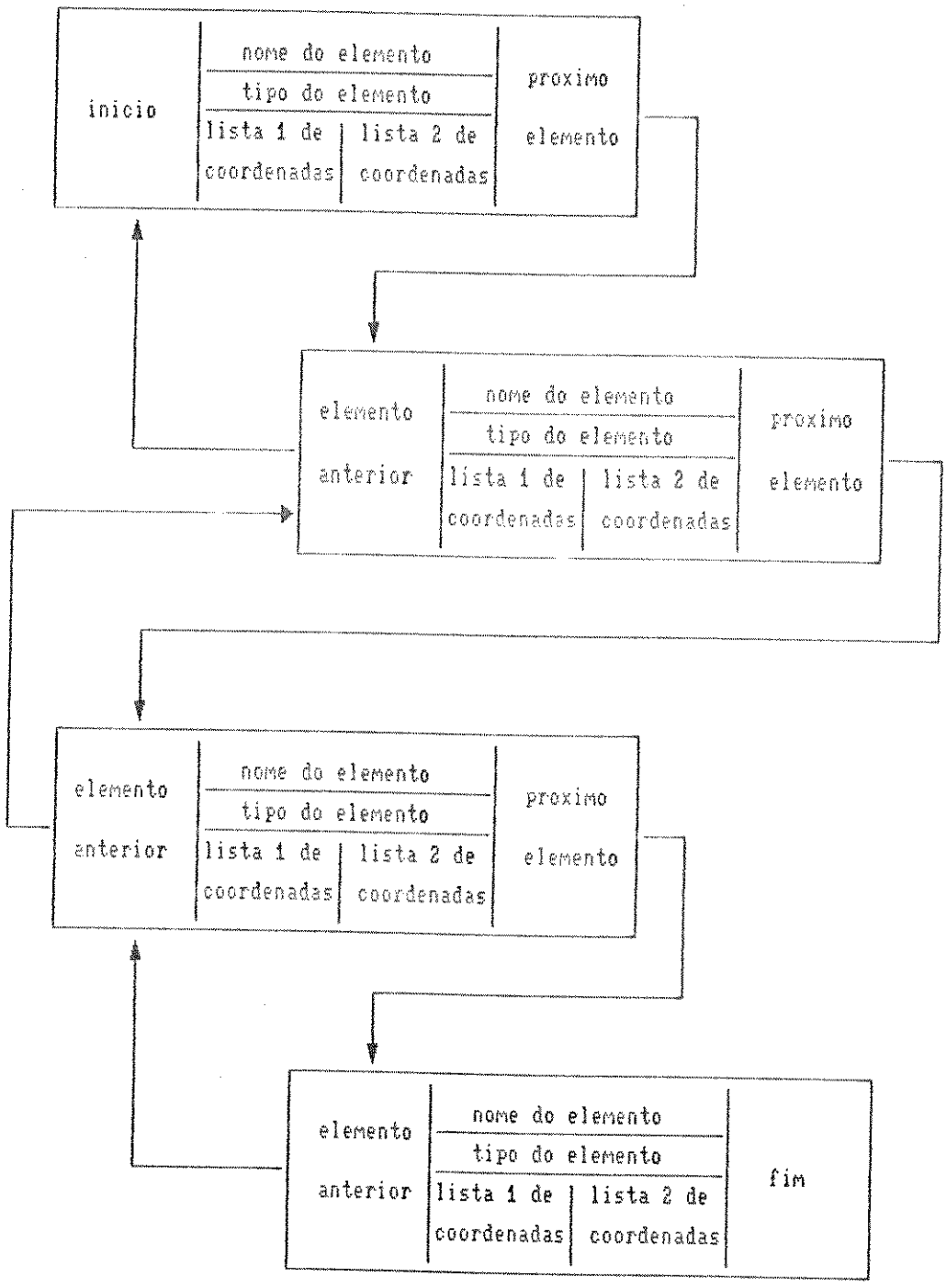


Figura 2.17. Lista ligada para elementos.

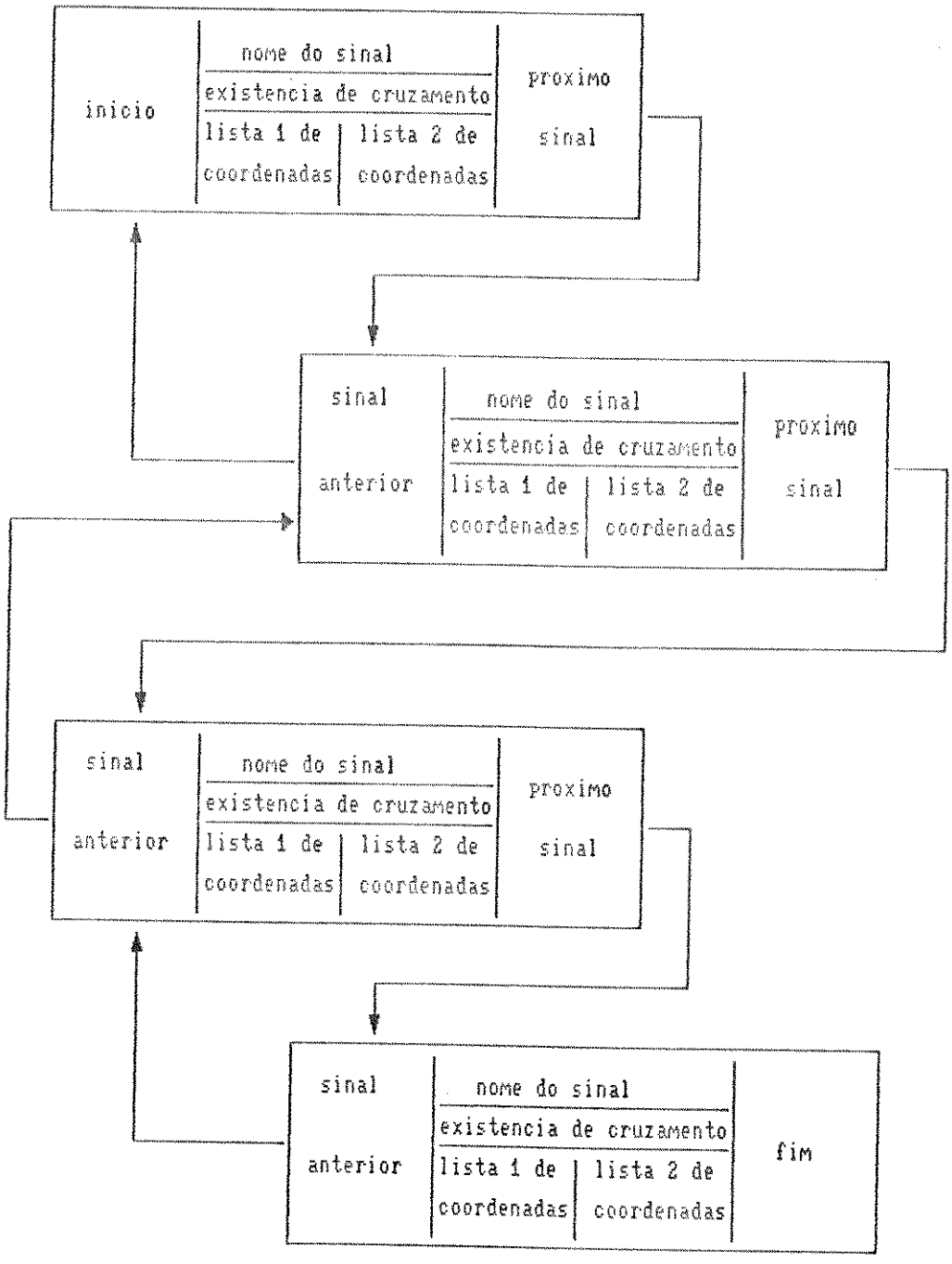


Figura 2.18. Lista ligada para sinais.

1.3.3.3.4. Lista ligada para folhas

Neste tipo de lista, armazenamos todas as informações contidas nas folhas, ou seja, as informações de edição do circuito. A célula básica deste tipo de lista é formada por:

- número de identificação da folha : variável inteira;
- apontador para a lista ligada de elementos;
- apontador para a lista ligada de sinais;
- apontador para a próxima célula;
- apontador para a célula anterior;

A figura 2.19 ilustra a estrutura geral de dados do programa PESQA.

1.3.3.4. Requisitos de desempenho

1.3.3.4.1. Mono-usuário

O desenvolvimento do Programa de Entrada Esquemática foi feito em microcomputador de 16 bits compatível com IBM-PC, sob um Sistema Operacional compatível com DOS, estando assim restrito a um usuário por máquina.

1.3.3.4.2. Arquivos

O Programa de Entrada Esquemática trabalha com os seguintes arquivos:

. Arquivo.PSQ - contém o NET-LIST do circuito.

. Arquivo.HLO - contém a descrição do circuito no formato HILO.

. Arquivo.SIM - contém a descrição do circuito no formato SIMUL ou SIMUFA.

. Arquivo.ELE - contém a lista de elementos de circuitos usados no projeto.

. Arquivo.EXT - contém a lista dos sinais a serem editados pelo EDTES.

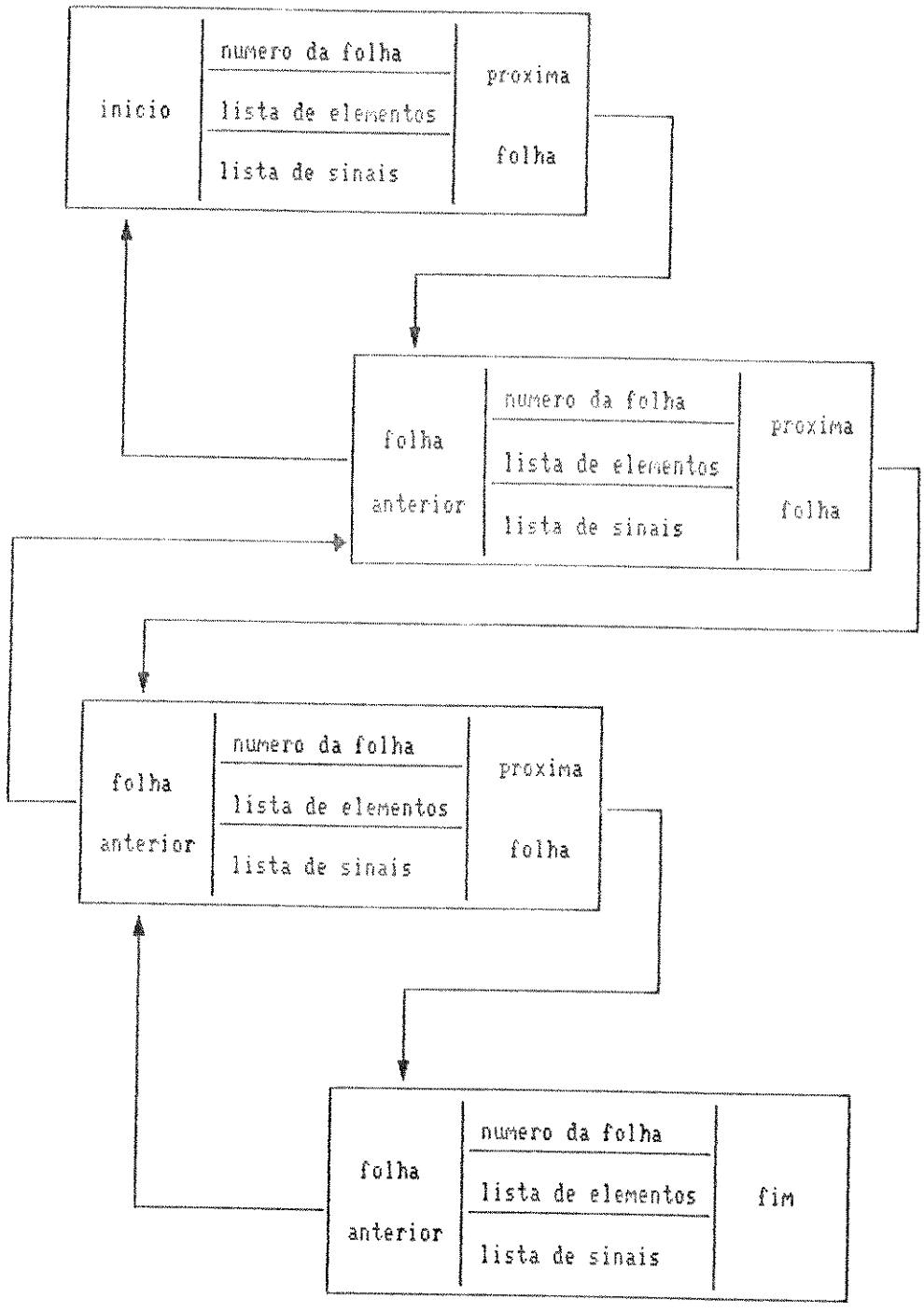


Figura 2.19. Estruturas geral de dados do programa PESQA.

1.3.3.4.3. Quantidade de dados

O Programa de Entrada Esquemática tem a capacidade máxima de 100 elementos por folha, com no máximo 20 folhas, perfazendo um total de 2.000 elementos.

1.3.3.5. Característica do sistema

1.3.3.5.1. Característica de "hardware"

- CPU 8086 e 512K bytes de memória principal;
- 10 Mbytes de memória secundária;
- monitor gráfico de 640 x 200 pontos;
- teclado alfanumérico;
- impressora semi-gráfica de 132 colunas.

1.3.3.5.1. Caracteristica de "software"

- compilador Turbo Pascal versão 3.01;
- área máxima de código : 64k bytes;
- área máxima de dados : 64k bytes.

1.3.3.6. Requisitos de interface externa

1.3.3.6.1. Interfaces com o usuário

- teclado alfanumérico;
- monitor gráfico de 640 x 200 pontos;
- impressora semi-gráfica de 132 colunas.

1.3.3.6.2. Interfaces de "hardware"

. interface serial RS-232;

. interface paralela Centronics.

1.3.3.6.3. Interface de "software"

. interface com o gerenciador de projetos e base de dados;

. interface com os simuladores SIMUL, SIMUFA e HILO-3;

. interface com o Editor Gráfico de Estímulos Externos - EDTES.

1.4. Especificação Funcional do Programa Editor Gráfico de Estímulos Externos

1.4.1. Introdução

1.4.1.1. Propósito do documento

Este documento tem como finalidade a especificação do Programa Editor Gráfico de Estímulos Externos, usado na edição de sinais de estímulos de circuitos digitais, do Sistema Computacional de Projeto Lógico do CTI/IM.

1.4.1.2. Escopo do programa

O Programa Editor Gráfico de Estímulos Externos chama-se "EDTES". Ele servirá como interface entre o projetista e os programas de simulação tais como HILO-3 [4], SIMUL [2,3], SIMUFA [16]. O projetista definirá os estímulos externos de um circuito digital, desenhando a forma de onda utilizando um editor gráfico dedicado.

1.4.1.3. Definições e abreviaturas

. HILO-3 - Programa de simulação com atraso atribuído, simulação de falhas e geração automática de vetores de testes. Produto da Genrad Inc [4].

. SIMUL - Programa de Simulação Lógica a nível de portas, com atraso unitário, desenvolvido no CTI/IM [2,3].

. SIMUFA - Programa de Auxílio à Verificação de Vetores de Testes, desenvolvido no CTI/IM [16].

. PESQA - Programa de Entrada Esquemática, desenvolvido no CTI/IM [este tese,5,6].

. SPL - Sistema Computacional de Projeto Lógico do CTI/IM.

1.4.1.4. Referências

. Especificação Funcional do Sistema Computacional de Projeto Lógico, documento em fase de preparação, CTI/IM .

. Especificação Funcional do Programa Gerenciador de Projetos e Base de Dados, documento em fase de preparação, CTI/IM.

- Especificação Funcional do Programa Formatador de Vetores de Testes, documento em fase de preparação, CTI/IM.
- Especificação Funcional do Programa de Entrada Esquemática, descrita no item anterior.
- Manual do Usuário do Programa HILO-3, GenRad Inc., October 1985 [4].
- Manual de Operação do Programa SIMUL, Centro Tecnológico para Informática, julho 1986 [2,3].
- Manual de Operação do Programa SIMUFA, Centro Tecnológico para Informática, julho 1986 [16].

1.4.1.5. Generalidades

Este documento contém, ainda, as funções do Editor Gráfico de Estímulos Externos, especificação de requisitos funcionais e desempenho, especificação de interfaces externas, restrições de projeto e características do computador e seus periféricos, onde será implementado.

1.4.2. Descrições gerais

1.4.2.1. Perspectiva do produto

Este programa faz parte do Sistema de Projeto Lógico (SPL) do Instituto de Microeletronica do Centro Tecnológico para Informatica e tem como objetivo complementar a tarefa executada pelo Programa de Entrada Esquemática PESQA, isto é, completar a geração gráfica de dados utilizados em simuladores. Ele recebe como dados de entrada um arquivo gerado pelo PESQA, contendo todos os sinais externos de um circuito editado, eliminando assim a possibilidade de não definição de um sinal de entrada.

1.4.2.2. Funções do programa

Como o PESQA, o Programa Editor Gráfico de Estímulos Externos tem o objetivo de facilitar a definição de sinais externos, através de facilidades gráficas de desenho, que serão usados como dados de entrada de simuladores lógicos. Esta ferramenta acelera a entrada de dados, diminui a probabilidade de erros e auxilia na documentação do projeto.

Para a realização desta tarefa, o Programa Editor Gráfico de Estímulos Externos é constituído de:

a. interface com o usuário;

b. núcleo de edição dos sinais:

. edição dos sinais,

. eliminação de sinais editados erroneamente;

c. gerador de arquivo de dados para:

. HILO-3,

. SIMUL,

. SIMUFA;

d. gerador de documentação:

. desenho dos sinais de entrada de um circuito em "plotter" ou impressora semigráfica.

Na figura 2.20 podemos observar o diagrama de blocos do Programa Editor de Estímulos Externos.

1.4.2.3. Características dos usuários

O Programa Editor de Estímulos Externos destina-se a projetistas de circuitos digitais e deve ser usado como complemento do programa PESQA.

1.4.2.4. Suposições e dependências

O programa foi desenvovido supondo que a disponibilidade de "hardware" e "software" fossem as seguintes:

- microcomputador de 16 bits da linha PC;
- sistema operacional compatível com DOS;
- linguagem Turbo Pascal versão 3.01.

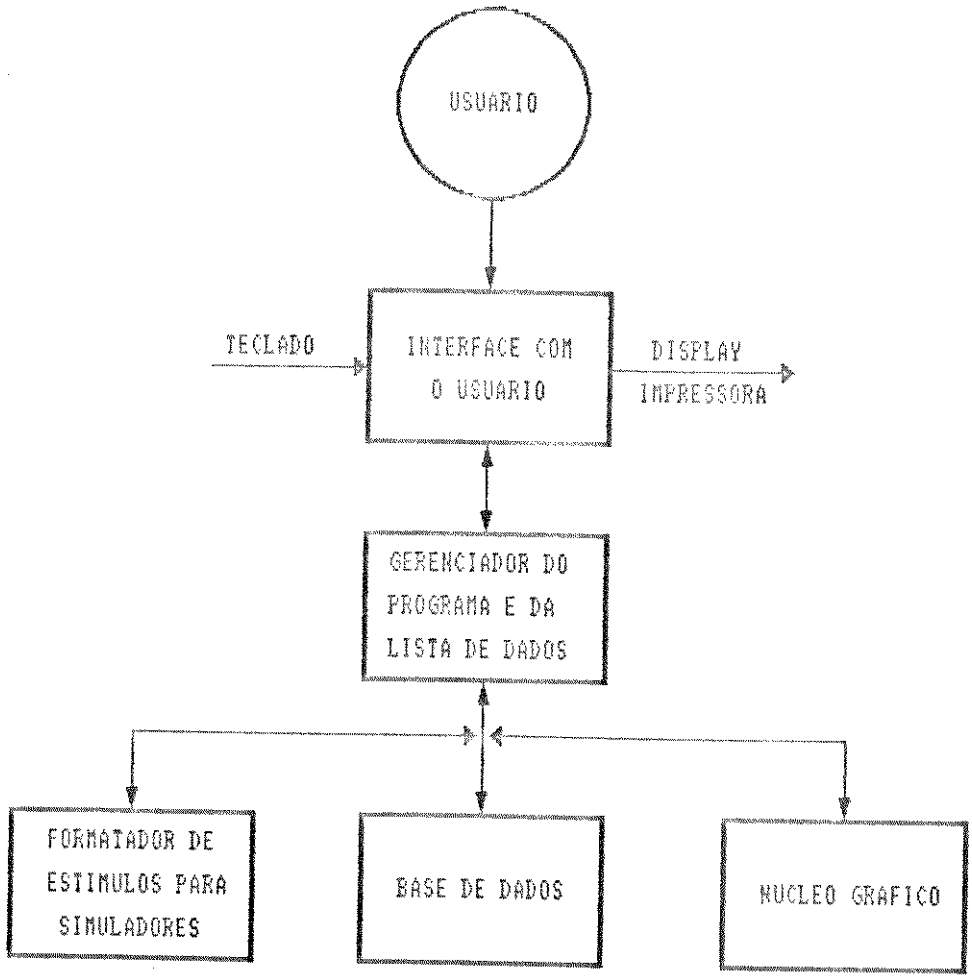


Figura 2.20. Diagrama de blocos do Programa Editor de Estímulos Externos.

1.4.3. Especificação de requisitos

1.4.3.1. Diagrama de fluxo de dados

Utilizamos uma técnica de representação gráfica para mostrar o fluxo de informações (dados) do programa Editor Gráfico de Estímulos Externos, isto é, o programa foi inicialmente estruturado através de um Diagrama de Fluxo de Dados - DFD.

No programa EDTES, os processos (funções) foram divididos em :

- funções de seleção;
  - . seleção de folhas de trabalho;
  - . seleção de janelas de visualização;
  - . seleção de parâmetros de tempo.
- funções de edição;
  - . composição de sinais;
  - . deleção de sinais.
- funções de entrada e saída.
  - . leitura de arquivos;
  - . armazenamento de arquivos;
  - . impressão dos sinais em uma impressora gráfica.

Podemos ver o Diagrama de Fluxo de Dados do programa EDTES na figura 2.21.

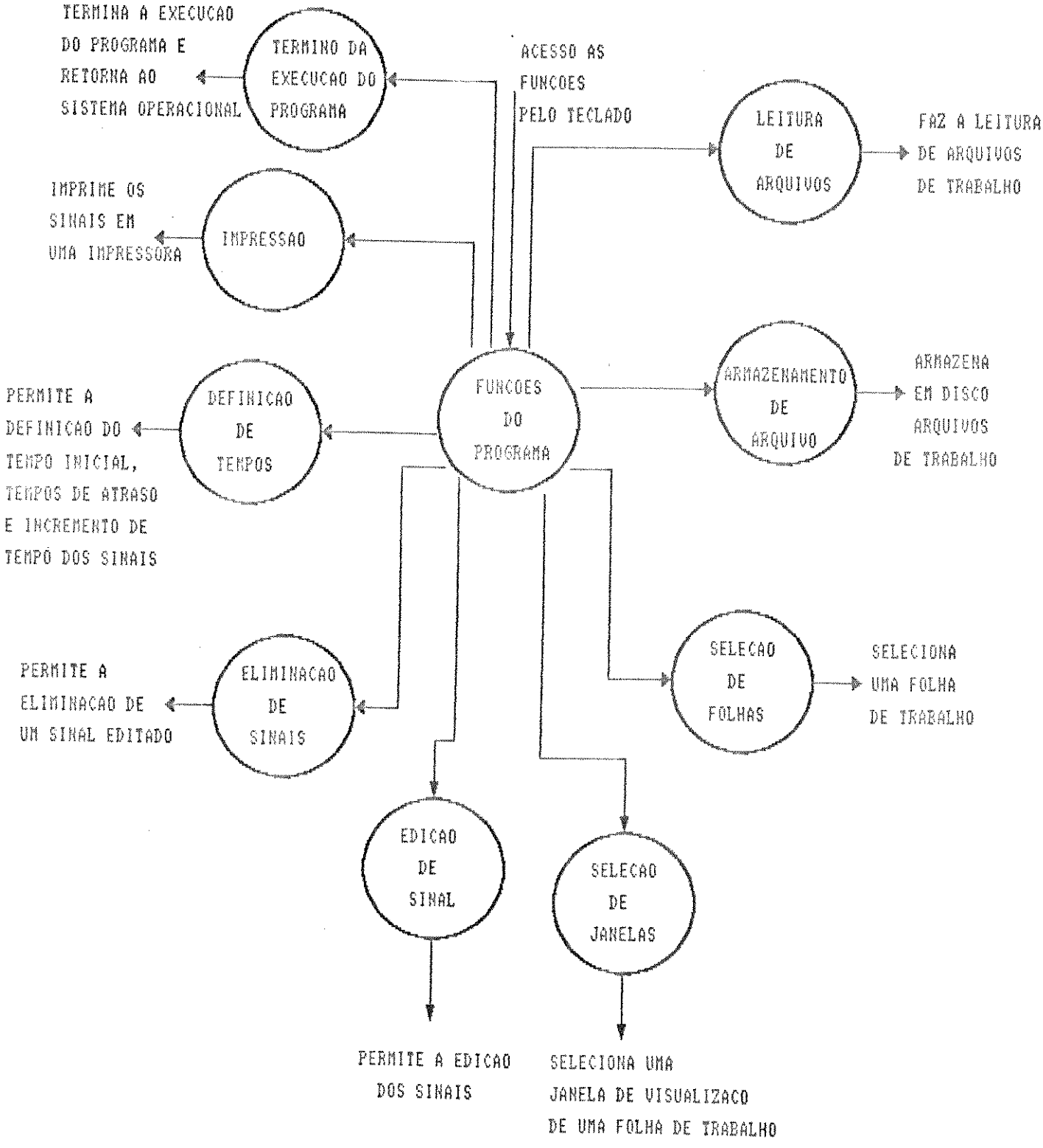


Figura 2.21. Diagrama de fluxo de dados do programa Editor de Estímulos Externos.

1.4.3.2. Requisitos Funcionais

1.4.3.2.1. Organização de Edição

No EDTES, a edição dos sinais externos é organizada em folhas e páginas. Cada folha é dividida em 10 páginas e cada página tem o tamanho da tela do monitor. Em cada folha pode se editar no máximo 13 formas de onda em função do tempo (sinais). A organização em folhas e páginas pode ser vista na figura 2.22. A visualização de uma folha é feita através de janelas, como mostra a figura 2.23.

1.4.3.2.2. Organização da tela de trabalho.

A tela do EDTES está dividida em:

- área de cabeçalho;
- área de menu;
- área de diálogo;
- área de trabalho

a. Área de cabeçalho

Nessa área são apresentadas as informaçòes do número da folha em que se está trabalhando, número de janelas de visualização e a posição do cursor em termos de coordenadas x e y.

b. Área de menu

Nesta área são apresentadas as funções disponíveis ao usuário. O acesso às funções é feita teclando a letra que indica a função. Esta letra é diferenciada das outras por estar indicada em maiúscula no menu, à direita na tela.

c. Área de diálogo

c1. Área de mensagem, onde temos as mensagens de advertência do programa.

c2. Área de entrada de dados, onde são requisitados dados necessários ao programa.

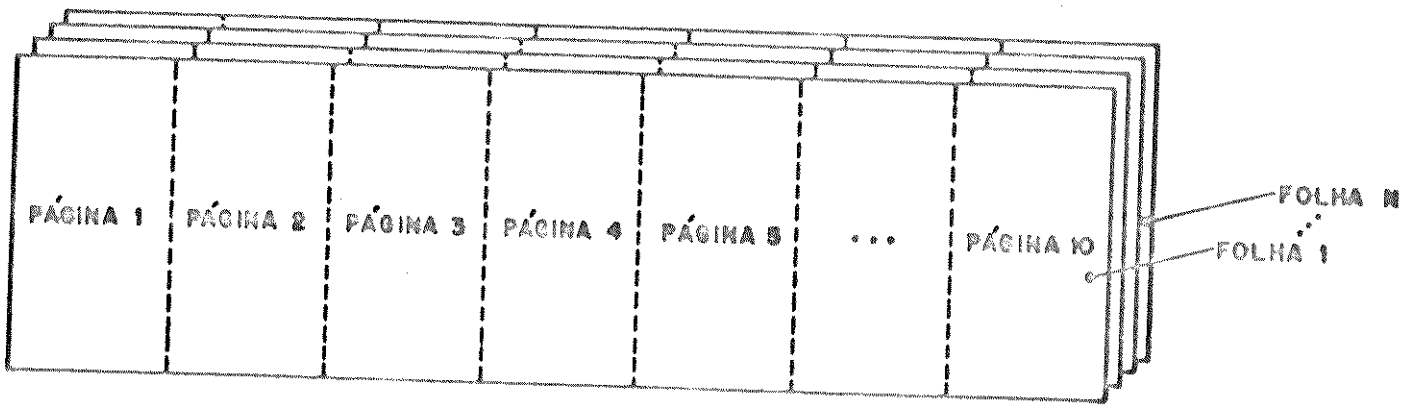


Figura 2.22. Organização do desenho dos sinais em folhas e páginas.

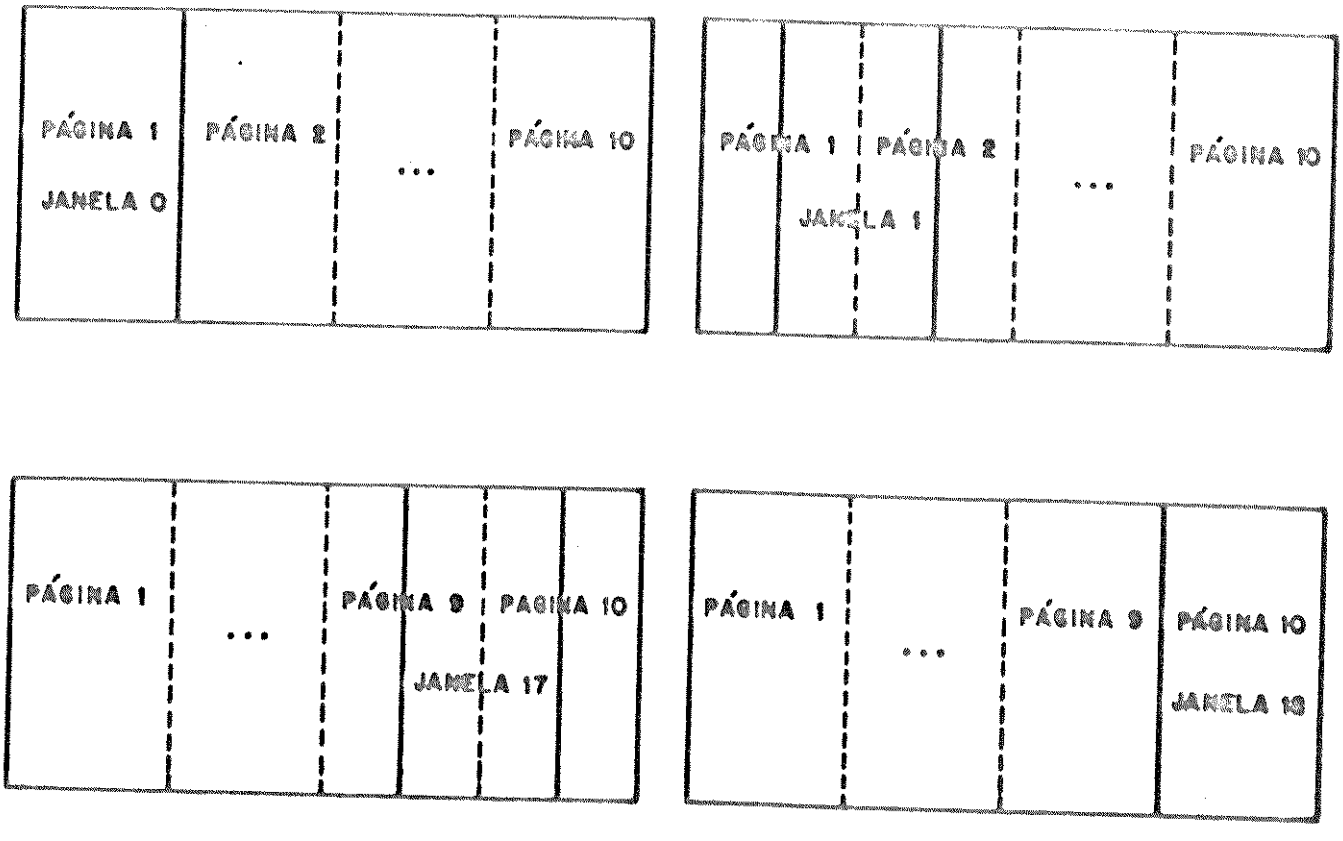


Figura 2.23. Janelas de visualização.

d. Área de trabalho

É a área reservada para a edição dos sinais. Nesta área temos à disposição um cursor, em forma de cruz, que se movimenta segundo uma grade. A movimentação do cursor é feita através de teclas, como mostra a Tabela 2.3. Esta área apresenta duas réguas, uma no eixo "X" e a outra no eixo "Y". A régua do eixo "X" representa o eixo dos tempos. Cada unidade da grade corresponde ao valor do incremento, que se não for indicado pelo usuário, terá o valor "default" igual a uma unidade de tempo. Pela função TEMPO o usuário poderá definir quaisquer valores para o incremento, e um valor para o tempo inicial, cujo "default" é zero. Esses valores poderão ser no máximo iguais a 32.000, por limitação do compilador. A régua do eixo "Y" representa o eixo dos valores lógicos onde os sinais devem ser editados. A figura 2.24 mostra como são definidos os valores lógicos dentre as três possibilidades: "1", "0" e "X", valores alto, baixo e indeterminado, respectivamente.

1.4.3.2.3. Funções do programa

As funções disponíveis no Editor Gráfico de Estímulo Externo são as seguintes:

a. FOLHA

Esta função tem como objetivo selecionar uma determinada folha para edição dos sinais. Ela é acessada através da tecla "F", de "Folha". O número da folha desejada é requisitado pelo programa através da área de diálogo.

b. JANEL

Esta função permite escolher uma janela de visualização dentro da folha de trabalho. A função é acessada através da tecla "J", de "Janela", e a escolha da janela é feita através de teclas, como mostra a tabela 2.4.

c. APGSI

Esta função é acessada pela tecla "P", de "aPagar". Ela permite apagar sinais já editados ou sinais lidos de um arquivo pré-editado.

Antes de apagar o sinal, o programa pede a confirmação do sinal selecionado, através da área de diálogo. Se a resposta for afirmativa, ela elimina o sinal da estrutura de dados. A seleção do sinal é feita levando o cursor para o ponto inicial do sinal e teclando "RET". Esta função termina digitando-se "F", de "Fim". Após o término desta função, o programa automaticamente fará com que os sinais apagados sejam novamente editados.

Tabela 2.3.

Teclas para movimentação do cursor

Tecla	Função
↑	O cursor se desloca um passo para cima.
↓	O cursor se desloca um passo para baixo.
-->	O cursor se desloca um passo para a direita.
<--	O cursor se desloca um passo para a esquerda.
PG UP	O cursor se desloca um passo na diagonal, para cima e para a direita.
PG DN	O cursor se desloca um passo na diagonal, para baixo e para a direita.
HOME	O cursor se desloca um passo na diagonal, para cima e para a esquerda.
END	O cursor se desloca um passo na diagonal, para baixo e para esquerda.
+	Aumenta o passo de deslocamento do cursor.
-	Diminui o passo de deslocamento do cursor.

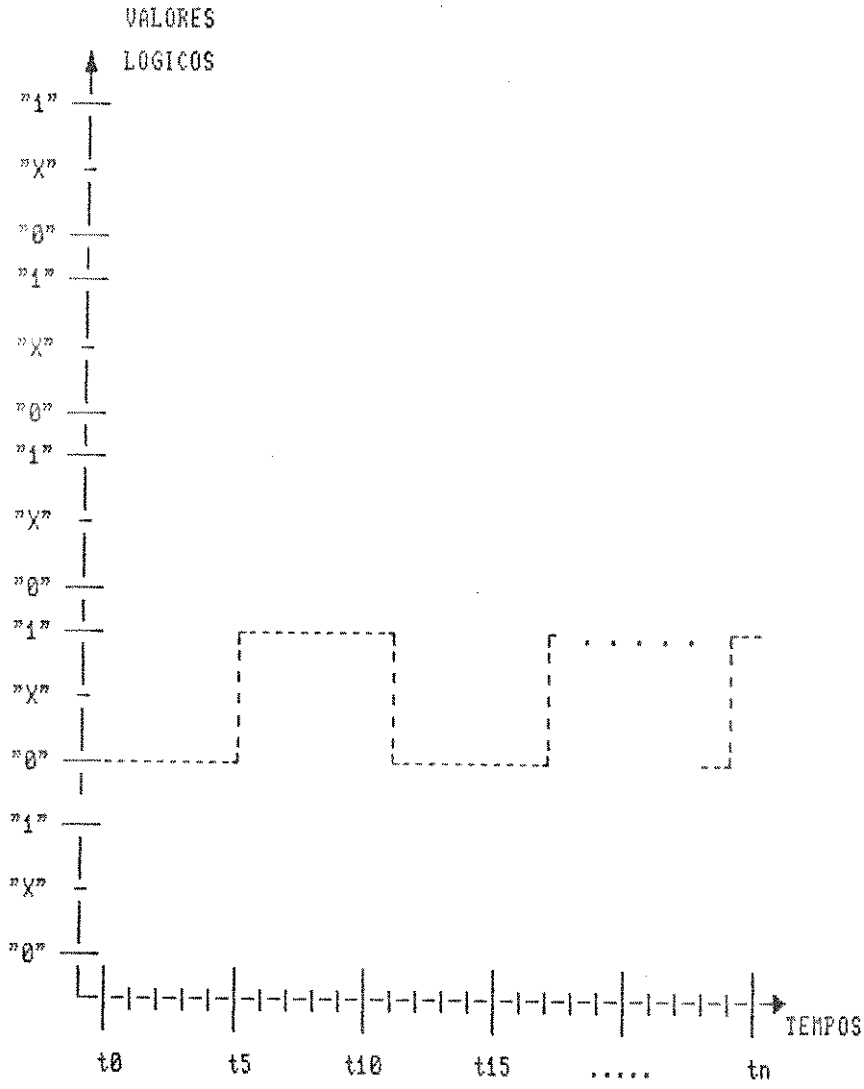


Figura 2.24. Definição de valores na régua de valores lógicos.

Tabela 2.4.

Teclas para movimentação da janela de visualização

Tecla	Função
0	Vai para a janela 0
1	Vai para a janela 1
2	Vai para a janela 2
3	Vai para a janela 3
4	Vai para a janela 4
5	Vai para a janela 5
6	Vai para a janela 6
7	Vai para a janela 7
8	Vai para a janela 8
9	Vai para a janela 9
A	Vai para a janela 10
B	Vai para a janela 11
C	Vai para a janela 12
D	Vai para a janela 13
E	Vai para a janela 14
F	Vai para a janela 15
G	Vai para a janela 16
H	Vai para a janela 17
I	Vai para a janela 18

d. EDSIN

Esta função permite fazer a edição dos sinais provenientes de uma arquivo com extensão .EXT, gerado pelo programa PESQA. Esta função é acessada pela tecla "E".

O desenho da forma de onda do sinal é feito da seguinte maneira:

- digita-se "RET" para o primeiro ponto do sinal;
- para cada mudança de nível lógico digita-se "C";
- digita-se "RET" para terminar o desenho do sinal.

O cursor posiciona-se automaticamente no ponto que representa tempo inicial e o valor lógico inicial de cada sinal.

Após a edição de todos os sinais, o programa encerrará automaticamente a função EDSIN.

e. TEMPO

Esta função é acessada pela tecla "T", de "Tempo". Ela permite a definição de um novo valor para o tempo inicial, cujo "default" é zero, e um novo valor para o incremento de tempo, cujo "default" é uma unidade de tempo. A requisição é feita pela área de diálogo.

f. IMPRI

Esta função tem como propósito imprimir, em uma impressora gráfica as formas de onda dos sinais. O acesso a essa função é feito através da tecla "R", de "impRessão".

g. ARMAZ

Esta função é acessada pela tecla "A", de "Armazenar". Ela tem o objetivo de armazenar arquivos de trabalho com três formatos de dados. O nome do arquivo será solicitado através da área de diálogo, e deverá ter no máximo 8 caracteres.

São os seguintes os formatos disponíveis:

g1. SIMUL

Se for digitado "M" de "siMul", o programa armazenará um arquivo que terá o formato de entrada dos programas SIMUL ou SIMUFA.

g2. HILO

Se for digitado "H", de "Hilo", o programa armazenará um arquivo que terá o formato de entrada do HILO-3.

g3. EDTES

Se for digitado "T", de "eDtes", o programa armazenará um arquivo que será entrada do próprio EDTES.

h. L ARQ

Esta função é acessada pela tecla "L", de "Ler arquivo". Ela tem o propósito de ler arquivos contendo ou os nomes de sinais vindos da edição de um circuito pelo programa PESQA, ou sinais já editados anteriormente pelo próprio programa EDTES. Toda vez que esta função for acionada, o programa perguntará o nome do arquivo, com no máximo 8 caracteres, através da área de diálogo.

Os arquivos possíveis de serem lidos são:

h1. PESQA

Se a tecla "Q", de "pesQa", for acionada, o programa lerá um arquivo gerado pelo programa PESQA. Este arquivo tem a extensão .EXT.

h2. EDTES

Se a tecla "T", de "edTes", for acionada, o programa lerá um arquivo gerado pelo próprio EDTES.

i. SAI

Esta função permite sair do Programa Editor de Estímulos Externos. Ela é acessada pela tecla "S", de "Sai".

1.4.3.3. Estrutura de dados

No programa EDTES os dados são armazenados em listas ligadas. Para isto utilizamos quatro listas de diferentes tipos:

- de coordenadas (x, y);
- de sinais;
- de folhas;
- de estímulos.

A célula da lista de folhas é composta pela lista de sinais que por sua vez tem na sua célula a lista de coordenadas.

1.4.3.3.1 Lista ligada para coordenadas

Na lista para coordenadas armazenamos as coordenadas dos pontos que compõem os sinais. Sua célula básica é composta por:

- coordenada x : variável inteira;
- coordenada y : variável inteira;
- apontador para o próximo ponto;
- apontador para o ponto anterior.

A figura 2.25 mostra a estrutura desta lista.

1.4.3.3.2 Lista ligada para sinais

Esta lista, mostrada na figura 2.26, armazena as informações dos sinais editados. Sua célula básica é composta por:

- lista ligada para coordenadas dos pontos dos sinais;
- tipo de sinal: variável tipo cadeia de 5 caracteres;
- nome do sinal: variável tipo cadeia de 5 caracteres;
- apontador para o próximo sinal;
- apontador para o sinal anterior.

1.4.3.3.3 Lista ligada para folhas

Na lista para folhas, ilustrada na figura 2.27, armazenamos as informações das folhas do EDTES, ou seja, as informações de edição dos estímulos. A célula básica desta lista é composta por:

- número de identificação da folha: variável inteira;
- apontador para a lista ligada de sinais;
- apontador para a próxima folha;
- apontador para a folha anterior;

1.4.3.3.4 Lista ligada para estímulos

Os dados dos estímulos relacionados com o simulador lógico, são armazenados na lista para estímulos, como mostra a figura 2.28. Sua célula básica é composta por:

- tipo do sinal: variável tipo cadeia de 5 caracteres;
- nome do sinal: variável tipo cadeia de 5 caracteres;
- valor inicial do sinal: variável tipo caracter;
- tempo de chaveamento do estímulo: variável tipo vetor inteiro;
- apontador para o próximo estímulo;
- apontador para o estímulo anterior.

1.4.3.4. Requisitos de desempenho

1.4.3.4.1. Mono-usuário

O Programa Editor Gráfico de Estímulos Externos foi desenvolvido para microcomputadores de 16 bits compatíveis com IBM-PC, sob um Sistema Operacional compatível com DOS, estando assim restrito a um usuário por máquina.

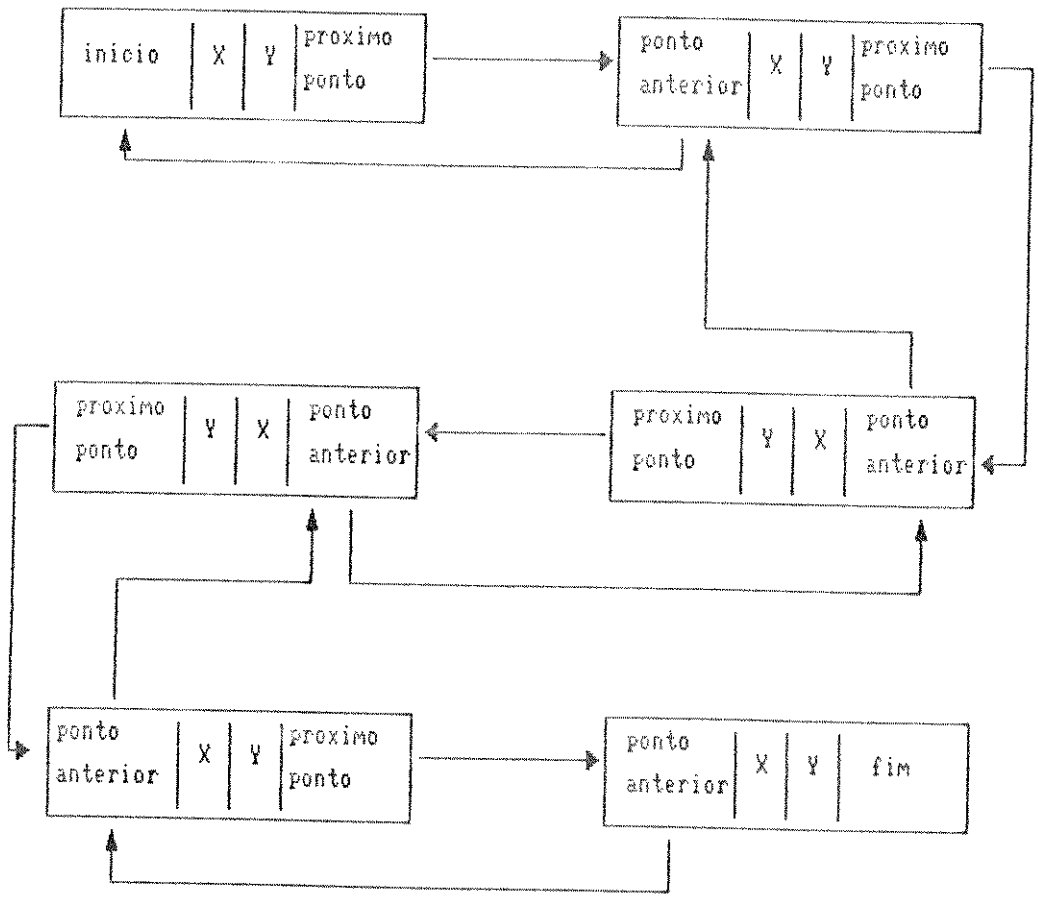


Figura 2.25. Lista ligada para coordenadas.

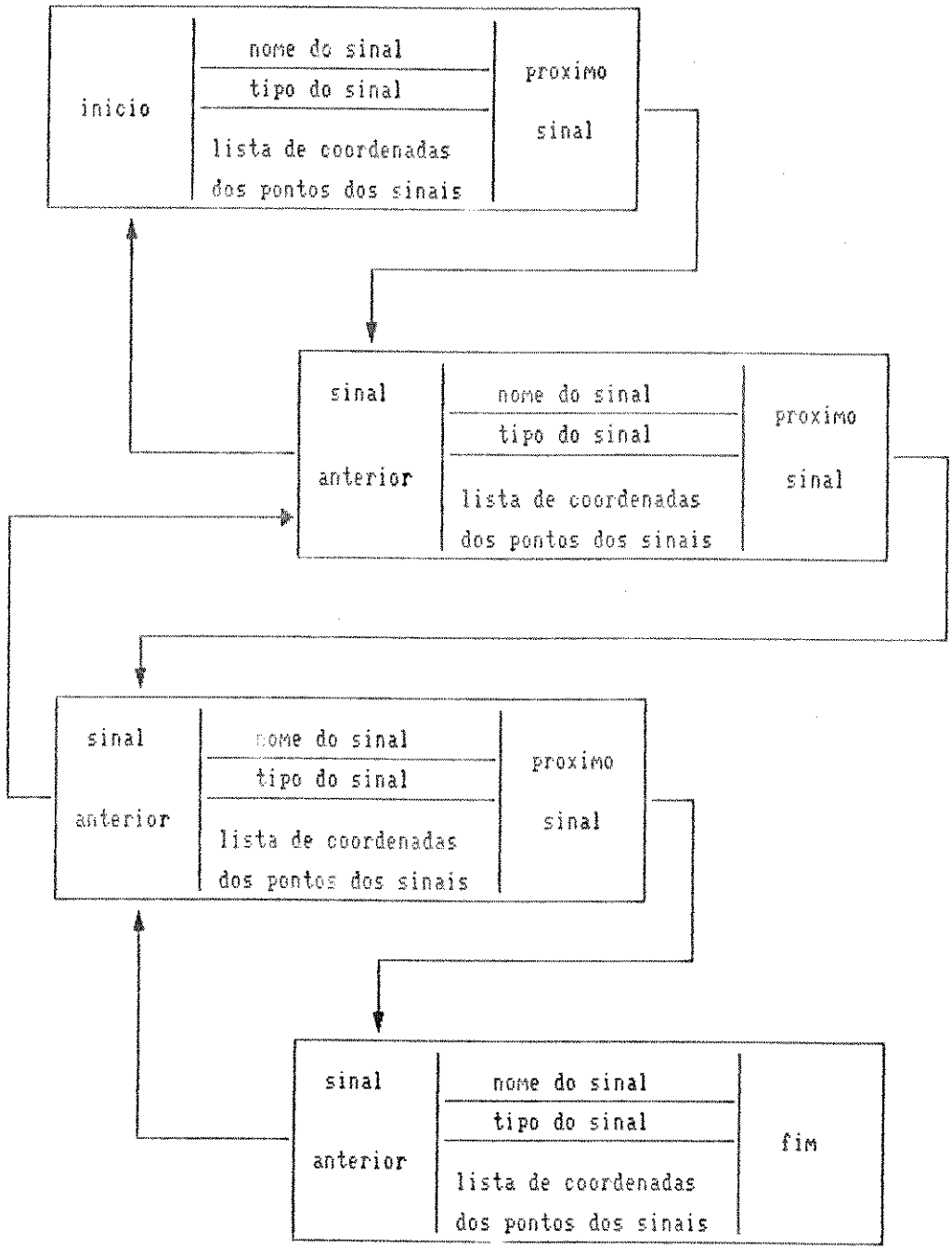


Figura 2.26. Lista ligada para sinais.

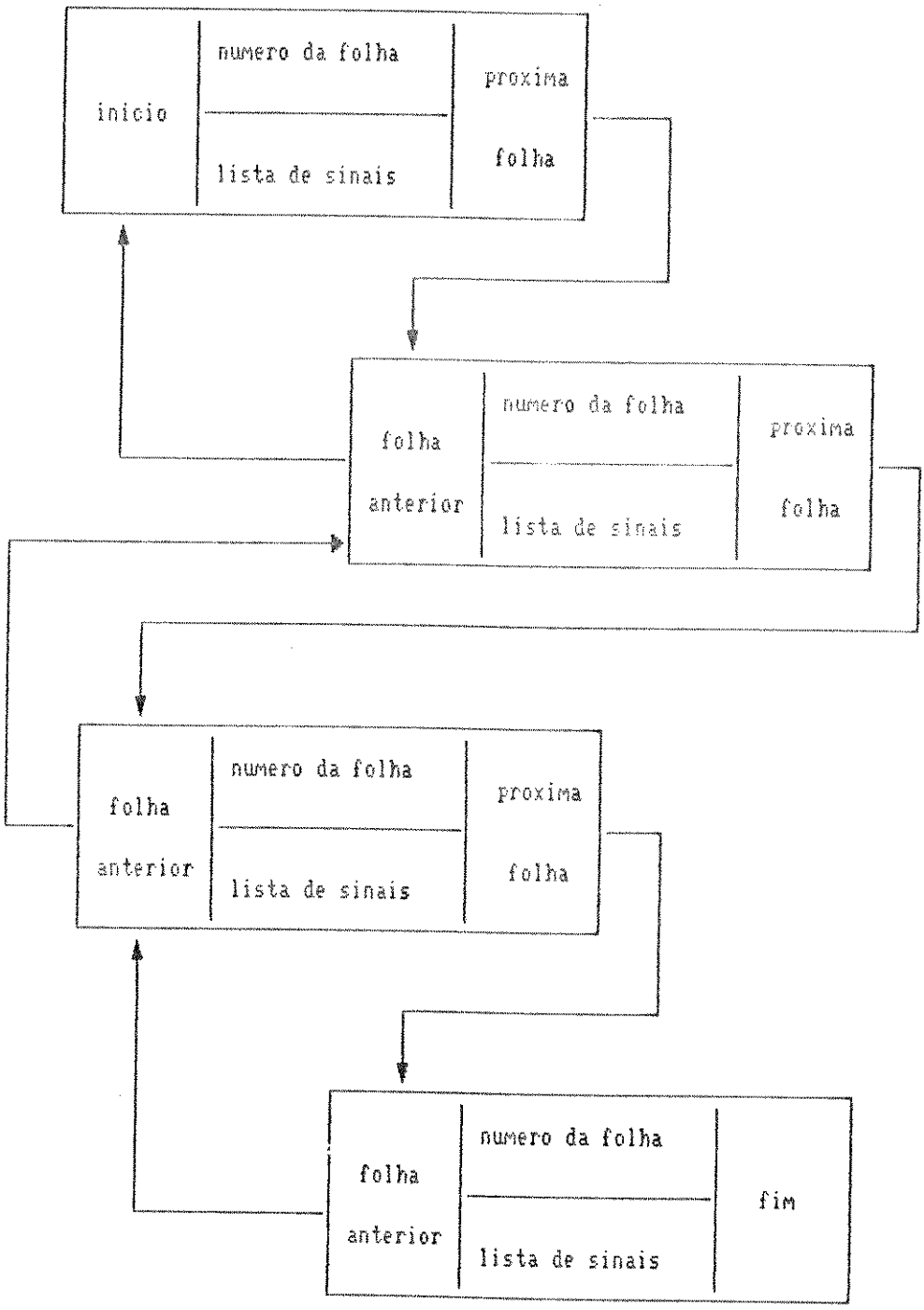


Figura 2.27. Lista ligada para folhas.

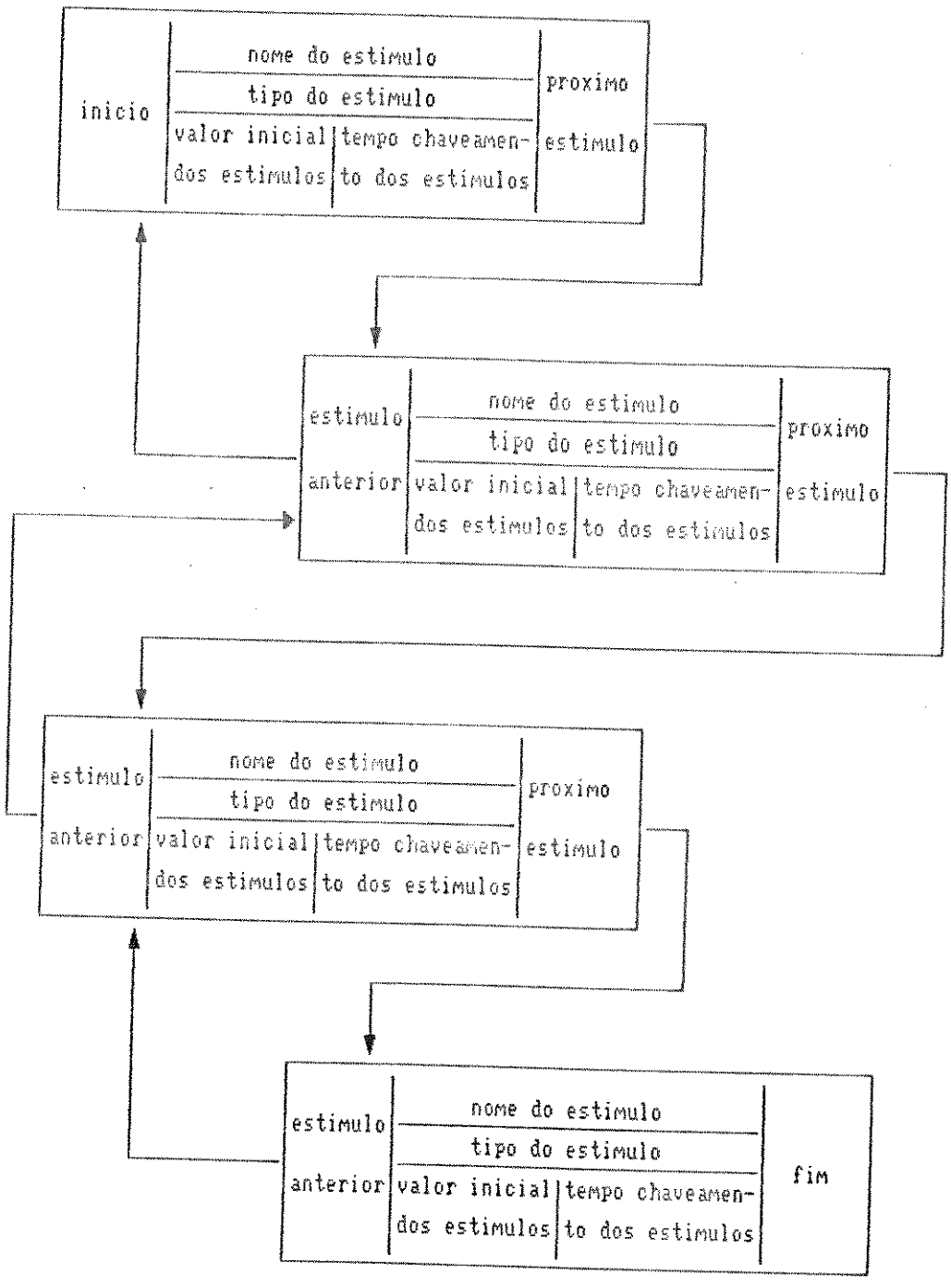


Figura 2.28. Lista ligada para estímulos.

1.4.3.4.2. Arquivos

O Programa Editor Gráfico de Estímulos Externos trabalha com os seguintes arquivos:

- Arquivo.EXT - contém a lista dos sinais a serem editados. Arquivo gerado pelo FESQA.
- Arquivo.EST - contém a definição do estímulo externo para os simuladores SIMUL e SIMUFA.
- Arquivo.HST - contém a definição do estímulo externo para o simulador HILO-3.
- Arquivo.EZT - contém o formato EDTES.

1.4.3.5. Característica do sistema

1.4.3.5.1. Característica do "hardware"

- CPU 8086 e 512K bytes de memória principal;
- 10 Mbytes de memória secundária;
- monitor monocromático de 640 x 200 pontos;
- teclado alfanumérico;
- impressora semi-gráfica de 132 colunas.

1.4.3.5.2. Característica do "software"

- Compilador Turbo Pascal versão 3.01;
- Área máxima de dados : 64 kbytes;
- Área máxima de código: 64 Kbytes.

1.4.3.5. Requisitos de interface externa

1.4.3.5.1. Interfaces com o usuário

- teclado alfanumérico;
- monitor gráfico de 640 x 200 pontos;
- impressora semi-gráfica de 132 colunas.

1.4.3.5.2. Interface de "hardware"

- interface serial RS-232;
- interface paralela Centronics.

1.4.3.5.3. Interface de "software"

- interface com o gerenciador de projetos e base de dados;
- interface com os simuladores SIMUL, SIMUFA e HILO-3;
- interface com o programa de entrada esquemática PESQA.

1.5. Projeto Físico e Codificação

Com base na especificação do programa, iniciamos a fase de projeto físico e codificação. Esta fase compreende a implementação e codificação das rotinas principais, descritas na especificação, e de rotinas secundárias que auxiliam o funcionamento do programa, tais como, rotinas de manipulação de listas, rotinas de leitura, rotinas de escritas, rotinas gráficas para desenhos de primitivas, etc.

Se a especificação é completa e bem definida, é possível que diferentes pessoas ou grupos possam desenvolver partes distintas (módulos) do programa.

Os programas foram projetados levando-se em conta três módulos básicos:

- módulo de entrada e saída de dados;
- módulo de gerenciamento e manipulação da estrutura de dados;
- núcleo gráfico.

O módulo de entrada e saída é responsável pela parte de diálogo entre o programa e o usuário e pela parte de leitura e escrita de arquivos. O módulo de gerenciamento e manipulação da estrutura de dados tem a tarefa de executar as operações que envolvem as listas ligadas, tais como, procura, comparação, ordenação, inserção e supressão de dados. O núcleo gráfico é responsável pelo desenho de todas as portas lógicas, pelo desenhos dos sinais (interconexões) e por todas as funções de transformação descritas nas especificações dos programas.

Foi utilizado o compilador TURBO PASCAL para o desenvolvimento dos programas PESQA e EDTES, para garantirmos a perfeita integração desses programas ao SDP [14]. Se por um lado este compilador ajuda a depuração dos erros, ele limita o tamanho da área reservada para código em 64K bytes. Isto faz com que tenhamos que utilizar alguns artifícios, como o uso de unidades que só ocupam a área de códigos quando utilizadas. Com esta troca de unidades (overlay) entre a memória principal e o disco rígido, o programa torna-se lento.

O protótipo do programa PESQA tem 4.815 linhas de código e as unidades compiladas têm um total de 88k bytes, enquanto que o protótipo do programa EDTES tem 1.972 linhas de código e a unidade compilada tem 35k bytes.

A duração do desenvolvimento, incluindo especificação e projeto físico, foi de quatro meses, com uma pessoa em dedicação exclusiva.

1.6. Testes

Para verificar se os programas projetados atendem as especificações, são necessários testes que permitam verificar o funcionamento e o desempenho. Optamos por realizar nossos testes através do uso do programa, o que nos permite avaliar sua capacidade quanto à quantidade de dados manipulados, a velocidade de processamento desses dados e principalmente quanto a facilidade de uso, além de apontar possíveis erros que necessitem ser corrigidos.

Nossa avaliação de desempenho dos programas foi efetuada através de dois projetos lógicos de circuitos integrados de baixa e média complexidade, juntamente com o programa SIMUL [3], realizados no Instituto de Microeletrônica do Centro Tecnológico para Informática. Os resultados desses testes são mostrados no capítulo III.

III. ANALISE E CONCLUSAO

1. Análise

O desenvolvimento dos programas PESQA e EDTES teve dois objetivos:

- desenvolvimento de ferramentas que, integradas ao Sistema Didático de Projeto [14], auxiliassem no aprendizado de projetos de circuitos integrados, desempenhando a função de descrição dos circuitos lógicos a serem simulados.

- desenvolver, em um curto espaço de tempo, um primeiro protótipo de sistema gráfico de entrada de dados para simuladores lógicos, que servisse como base no desenvolvimento de uma ferramenta gráfica profissional para integrar o Sistema de Projeto Lógico. Este sistema gráfico encontra-se em desenvolvimento no Centro Tecnológico para Informática.

Levando em conta principalmente o primeiro objetivo, procuramos desenvolver os programas em ambientes de baixo custo, visando sua disseminação em universidades, que em geral, têm recursos computacionais limitados . Este sistema subentende:

- microcomputador de 16 bits da linha PC com 512k bytes de RAM;

- monitor monocromático de 640 X 200 pontos;

- teclado alfanumérico;

- acionadores de discos flexíveis;

- disco rígido tipo Winchester de 10M bytes;

- impressora gráfica de 132 colunas;

- sistema operacional compatível com DOS.

Os programas foram desenvolvidos seguindo-se a metodologia de projeto citada nos capítulos I e II, em ambiente Turbo Pascal. O compilador Turbo Pascal limita a área de dados em 64k bytes e a área de códigos em 64k bytes por unidade. Mas como usamos alocação dinamica de memória, a quantidade de dados está limitada pela memória principal do microcomputador onde os programas são executados.

Foi ainda necessária a implementação de um pequeno núcleo gráfico para suportar algumas características de edição, tais como, transformações, desenho de círculos, etc., visto que o compilador só oferece as primitivas de desenho de ponto e reta.

Os programas foram testados na II e III EBAI - Escola Brasileiro-Argentina de Informática, no curso de Laboratório de Microeletrônica, onde foram usados por alunos de diferentes níveis de formação, evidenciando a rapidez no aprendizado e a facilidade de uso, na edição de circuitos integrados relativamente simples.

No CTI/IM, eles também foram testados em dois projetos de circuitos integrados , um com 1200 portas equivalentes e 9 sinais de entrada e outro com 140 portas equivalentes e 8 sinais de entrada. A descrição do circuito com 1200 portas, consumiu um total de 8 páginas de desenho. O arquivo com o formato de dados PESQA, referente a esta descrição, ficou com 73K bytes. A descrição das formas de onda de entrada consumiu 1 página de desenho e o arquivo com o formato de dados EDTES ficou com 8,5K bytes. A descrição do circuito com 140 portas consumiu uma página de desenho, ficando com 7k bytes o arquivo no formato PESQA . A descrição das formas de onda desse circuito utilizou 1 página de desenho e o arquivo com o formato EDTES ficou com 5k bytes.

O simulador lógico usado nestes exemplos foi o SIMUL [3]. A interface com o HILO-3 [4] está ainda em fase de testes, pois só recentemente este simulador ficou disponível no CTI/IM.

Devido às restrições impostas principalmente pelo ambiente onde foram desenvolvidos, os programas não se mostraram eficientes para projetos com mais de 200 elementos de circuito, se comparados a uma descrição feita em um editor de textos. Para projetos de grande porte, seria necessário o desenvolvimento desses programas em computadores com monitores coloridos de média ou alta resolução, tendo disponíveis pacotes gráficos como o PLOT 10 ou o AGP , para podermos implementar funções gráficas que agilizassem o processo de edição, tais como , "zoom", "pan", "windows", etc. Nesse ambiente, o uso de periféricos dedicados à entrada gráfica de dados, como o "mouse" ou o "tablet", aumentaria a velocidade de entrada de dados, tornando os programas ainda mais eficientes.

2. Exemplo do uso dos programas

Para ilustrar uso dos programas de Entrada Esquemática e do Editor Gráfico de Estímulos Externos segue um exemplo da ediçãO e simulaçãO de um circuito, onde são mostrados as telas editadas e também os formatos dos arquivos de saída dos programas PESQA e EDTES e os formatos de dos arquivos de entrada do SIMUL, gerados pelo PESQA e EDTES.

2.1 Arquivo de saída com o formato PESQA

```
TESEMST
1
ELE
e1
22 270
24 4 ;
32 24 ;
e2
1 0
80 24 ;
152 24 ;
e3
1 0
200 24 ;
272 24 ;
e4
22 270
16 44 ;
24 64 ;
e5
1 270
192 52 ;
192 88 ;
e6
3 270
264 52 296 52 ;
280 84 ;
e7
3 270
336 52 368 52 ;
352 84 ;
e8
3 270
416 52 448 52 ;
432 84 ;
```

```
e9
3 270
480 52 512 52 ;
496 84 ;
e10
2 0
72 64 72 48 ;
128 56 ;
e11
21 0
416 116 416 132 416 124 464 92 464 156 ;
512 116 512 132 ;
e12
21 0
256 116 256 132 256 124 304 92 304 156 ;
352 116 352 132 ;
e13
23 270
376 144 ;
376 164 ;
e14
23 270
536 144 ;
536 164 ;
e15
23 270
568 144 ;
568 164 ;
e16
22 270
88 104 ;
96 124 ;
FIM
SINAIS
entr1
32 24 80 24 ;
48 24 ;
entr1
48 24 48 48 72 48 ;
48 24 ;
entr2
24 64 72 64 ;
;
s3
152 24 200 24 ;
;
s4
272 24 416 24 416 52 ;
288 24 ;
```

```
s5
128 56 128 40 512 40 512 52 ;
448 40 368 40 296 40 192 40 ;
s5
448 52 448 40 448 40 ;
448 40 ;
s5
368 52 368 40 368 40 ;
368 40 ;
s5
296 52 296 40 296 40 ;
296 40 ;
s4
264 52 264 32 288 32 288 24 288 24 ;
288 24 ;
s5
192 52 192 40 192 40 ;
192 40 ;
s6
280 84 208 84 208 156 304 156 ;
280 84 ;
s6
280 84 320 84 320 52 336 52 ;
280 84 ;
s7
192 88 192 132 256 132 ;
192 116 192 88 ;
s7
256 116 192 116 192 116 ;
192 116 ;
clk
96 124 256 124 ;
;
out1
352 116 376 116 376 144 ;
;
s10
352 132 352 148 408 148 408 124 416 124 ;
;
s11
304 92 352 92 352 84 ;
;
s7
192 88 384 88 384 132 416 132 ;
192 88 384 116 ;
s7
416 116 384 116 384 116 ;
384 116 ;
```

```
s12
464 92 496 92 496 84 ;
;
s13
432 84 400 84 400 156 464 156 ;
432 84 ;
s13
480 52 464 52 464 84 432 84 432 84 ;
432 84 ;
out2
512 132 536 132 536 144 ;
;
out3
568 144 568 116 512 116 ;
;
FIM
.
```

2.2 Arquivo editado pelo Pesqa com o formato SIMUL

```
ELE
NAND   ENTR2 ENTR1 0     0     0     s8    0
NAND   ENTR1 0     0     0     0     s7    0
NAND   s7    0     0     0     0     s9    0
NAND   s8    0     0     0     0     s5    0
NAND   s2    s8    0     0     0     s1    0
NAND   s9    s8    0     0     0     s2    0
NAND   s4    s8    0     0     0     s3    0
NAND   s9    s8    0     0     0     s4    0
JKFF   s5    s5    s6    s3    s4    OUT3  OUT2
JKFF   s5    s5    CLK   s1    s2    OUT1  s6
FIM
SAI
ENTR1 ENTR2 CLK
OUT3  OUT2  OUT1
FIM
```

2.3 Arquivo de saída do PESQA usado como dado para o EDTES

```
EST
ENTR1 ENTR2 CLK
FIM
```

2.4 Arquivo de saída do EDTES com formato EDTES

```
0 1
1
EST
A ENTR1
16 156 96 156 96 148 576 148 ;
A ENTR2
16 136 576 136 ;
P CLK
16 132 56 132 56 124 96 124 ;
FIM
.
```

2.5 Arquivo editado pelo EDTES com formato SIMUL

```
EST
A ENTR1 0 0 10 0 0 0 0 0
A ENTR2 1 0 0 0 0 0 0 0
P CLK 0 0 5 0 0 0 0 0
FIM
```

2.6 Telas de editadas dos programas PESQA e EDTES

A figura 3.1 mostra um divisor de frequência editado pelo PESQA, a figura 3.2 mostra a edição dos estímulos feito pelo EDTES e a figura 3.3 mostra o resultado da simulação feito com o simulador lógico SIMUL.

3.Conclusões

Os programas PESQA e EDTES desenvolvidos como objeto deste trabalho, atenderam as especificações, principalmente no que se refere aos objetivos de utilização como ferramentas de auxílio ao aprendizado do projeto de circuitos integrados, integrando-se com o Sistema Didático de Projeto [14], diminuindo a probabilidade de erros na descrição de circuitos para a simulação lógica e facilitando a documentação do projeto. No que se refere a circuitos de alta complexidade eles se mostraram limitados, mas importantes para o desenvovimento de sistemas gráficos mais eficientes.

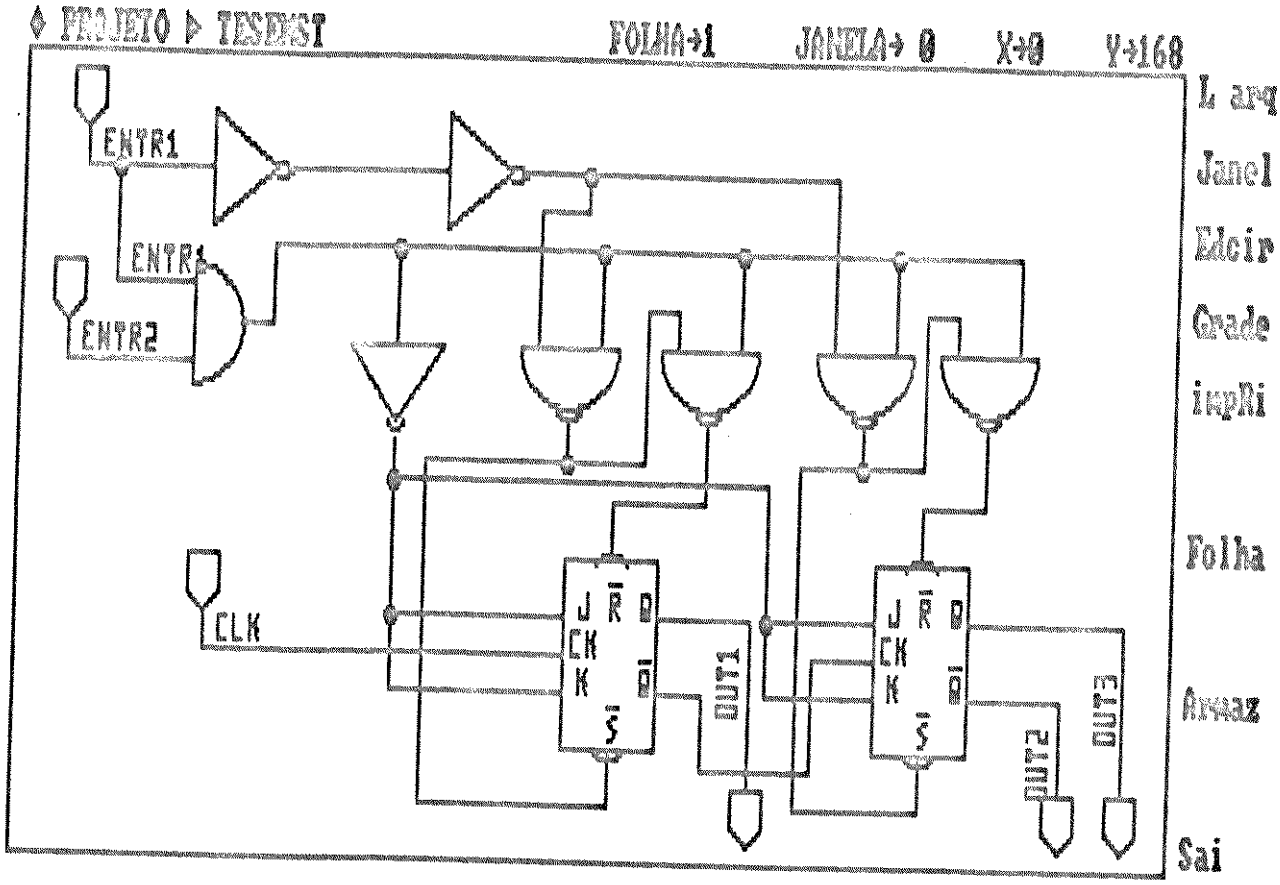


Figura 3.1. Edição de um circuito usando o PESQA.

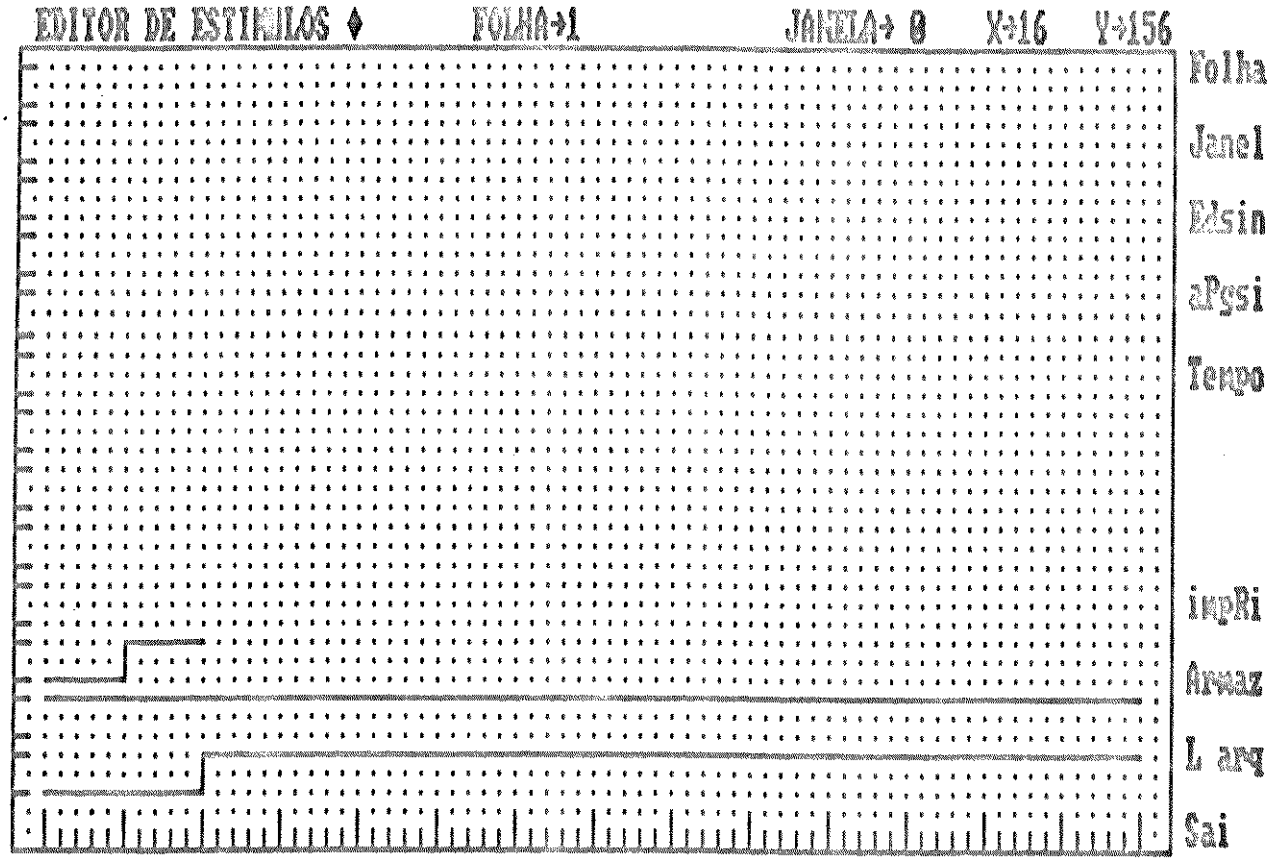


Figura 3.2. Edição dos estímulos usando o EDTES.

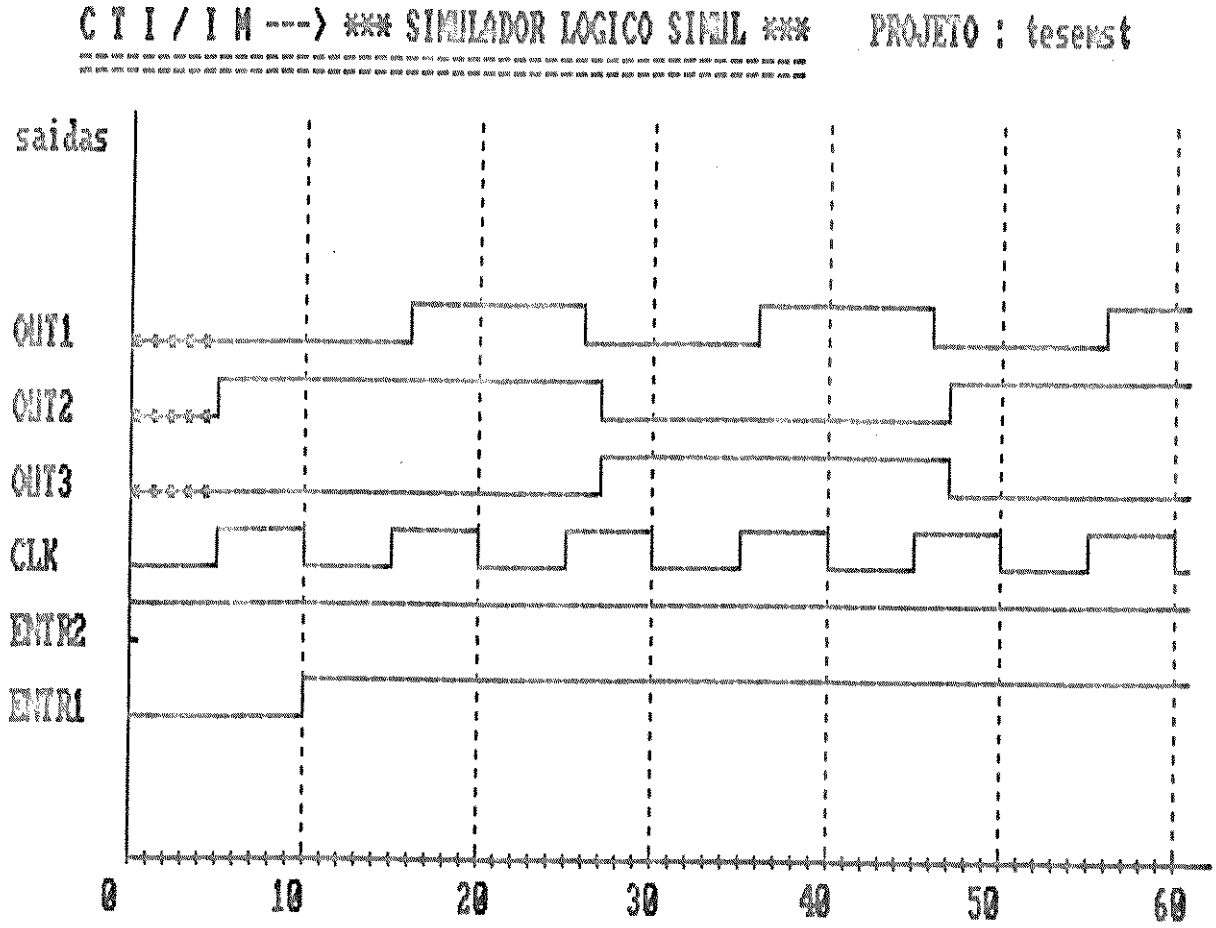


Figura 3.3. Resultado da simulação utilizando o SIMUL.

Os testes de uso mostraram que os programas são eficientes para circuitos pequenos, isto é, de até 200 portas equivalentes. Para circuitos maiores, pelas próprias restrições do ambiente em que foram desenvolvidos, os programas tornam-se ineficientes. Para o uso com circuitos complexos, estes programas estão sendo implementados em ambientes que permitam o uso de periféricos especiais de posicionamento, tais como o "mouse" e o "tablet", usando pacotes gráficos para agilização de rotinas gráficas, tais como, "zoom", "pan", etc.

A metodologia de projeto adotada, permitiu que o desenvolvimento desses programas se desenrolasse de forma rápida e organizada, podendo ser também adotada com sucesso no desenvolvimento de sistemas maiores, onde o número de pessoas envolvidas no projeto é grande.

Como conclusão final podemos salientar que os programas demonstraram ser eficientes como ferramentas didáticas de auxílio a projeto e apesar das limitações impostas pelo ambiente de desenvolvimento, eles mostraram ser confiáveis na descrição de circuitos complexos. Além disto serviram como protótipos no desenvolvimento de ferramentas gráficas mais eficientes.

Em continuação a este trabalho já estamos integrando os dois programas em um único sistema gráfico de entrada de dados utilizando o seguinte ambiente:

. "hardware"

.. computador HP 9000 com 3M bytes de RAM;

.. monitor colorido de 560 X 450 pontos;

.. "tablet" para entrada de dados;

.. "plotter" para saída de dados;

.. disco rígido de 67M bytes.

. "software"

.. sistema operacional HP-UX;

.. linguagem C;

.. pacote gráfico AGP da Hewlett - Packard.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Carlos I. Z. Mammana e Alaíde P. Mammana, "Introdução ao Projeto de Circuitos Integrados", II Escola Brasileiro - Argentina de Informática, 1987; III Escola Brasileiro - Argentina de Informática, 1988.

2. Ricardo Pannain, " Manual de operação do programa SIMUL ", Documento Interno - CTI/IM, Campinas 1986.

3. Ricardo Pannain, " Manual de usuário do programa SIMUL ", Documento Interno - CTI/IM, Campinas 1986; II Escola Brasileiro - Argentina de Informática, 1987; III Escola Brasileiro - Argentina de Informática, 1988.

4. Genrad Inc., " Hilo-3 User's Manual ", Santa Clara, October 1985.

5. Ricardo Pannain, " PESQA - Programa de Entrada Esquemática", II Congresso da Sociedade Brasileira de Microeletrônica - São Paulo, julho 1987.

6. Ricardo Pannain, "Manuais dos Programas PESQA e EDTES", II Escola Brsileiro - Argentina de Informática, 1987; III Escola Brasileiro - Argentina de Informática, 1988.

7. Carlos I. Z. Mammana e Silvia H. O. Machado, " SDP - Sistema Didático de Projeto ", II Escola Brasileiro - Argentina de Informática, 1987; III Escola Brasileiro - Argentina de Informática, 1988; II Congresso da Sociedade Brasileira de Microeletrônica, julho de 1987.

8. Software Engineering Technical Committee of IEEE Computer Society, " Guide to Software Requirements Specification ", julho 1984.

9. Douglas Lewin, " Computer-Aided Design of Digital Systems", Crane, Russak & Co., 1977

10. Paulo Veloso, Clésio Santos, e Antônio Furtado, " Estruturas de Dados ", Editora Campus, Rio de Janeiro 1983.

11. Melvin A. Breuer, "Design Automation of Digital Systems", Prentice - Hall International, 1972.

12. Melvin A. Breuer e Arthur D. Friedman, "Diagnosis & Reliable of Digital Systems", Computer Science Press, 1976.

13. Neil Weste e Kamran Eshraghian, "Principles of CMOS VLSI Design - A System Perspective ", Addison Wesley Publishing Co., 1985.

14. Vladimirescu, A. et all - " Spice 2G5 User's Guide ", University of California - Berkeley.

15. R. S. Pressman, " Software Engineering: A Practitioners Approach ", MacGraw Hill Inc., 1982.

16. Ricardo Pannain, " Manual de operação do programa SIMUFA ", Documento Interno - CTI/IM, Campinas 1986.

17. Willian M. Newman e Robert F. Sproul, " Principles of Interative Computer Graphics " , MacGraw Hill, 1979.

18. J. D. Foley e Van Dam, " Fundamentals of Interative Computer Graphics ", Addison Wesley Publishing Co. 1982.

19. Steven Harrington, " Computer Graphics - A Programming Approach", MacGraw Hill, 1983.

20. I. Sommerville, " Software Engineering ", Addison-Wesley Publishing Company, 1982.

APÊNDICE A

PROGRAMA DE ENTRADA ESQUEMATICA

PESQA - V.00

MANUAL DE USUARIO

A1. INTRODUÇÃO

O objetivo deste manual é apresentar ao usuário as principais características do programa de entrada esquemática PESQA.

Este programa foi desenvolvido na Divisão de Suporte Computacional do Instituto de Microeletrônica do Centro Tecnológico para Informática, em Campinas, São Paulo, Brasil.

A2. DESCRIÇÃO DO SISTEMA

A2.1 Função

Para o projetista de circuitos digitais, torna-se mais interessante projetar e analisar circuitos por meio de diagramas esquemáticos. O programa de entrada esquemática tem como objetivo auxiliar o projetista na composição do diagrama esquemático, mediante facilidades gráficas de desenho (símbolos, linhas, identificação de elementos e sinais, etc.).

A partir do diagrama esquemático, o programa extrai o "NET-LIST" do circuito, contendo a descrição dos elementos lógicos e as interconexões que o compõem. O "NET-LIST" é usado como entrada de simuladores lógicos e também como entrada de programas de alocação e roteamento.

O desenho de um circuito, deverá ser dividido em folhas. Cada folha é dividida em páginas, sendo que cada página corresponde a uma tela do monitor. A figura A1 mostra a organização em folhas e páginas. A visualização de uma folha na tela é feita por janelas de visualização, conforme mostra a figura A2.

A2.2.1 Menu principal

Ao se iniciar o programa, aparecerá uma tela introdutória, onde aparecerá o nome do programa e sua origem. Depois, esta tela se apagará e aparecerá a tela inicial do programa. A figura A3 mostra a tela inicial. A direita da tela, encontramos as funções do menu principal. Todas elas podem ser acessadas, digitando-se a letra que aparece em maiúscula no nome de cada função.

O primeiro parâmetro requisitado pelo programa será o nome de identificação do projeto. Esse nome poderá ter no máximo 15 caracteres alfanuméricos. Após a digitação do nome do projeto, o programa perguntará qual o número da folha que se deseja editar. O número máximo de folhas editáveis é 9.999. Automaticamente, ele irá para a janela 1 da folha desejada.

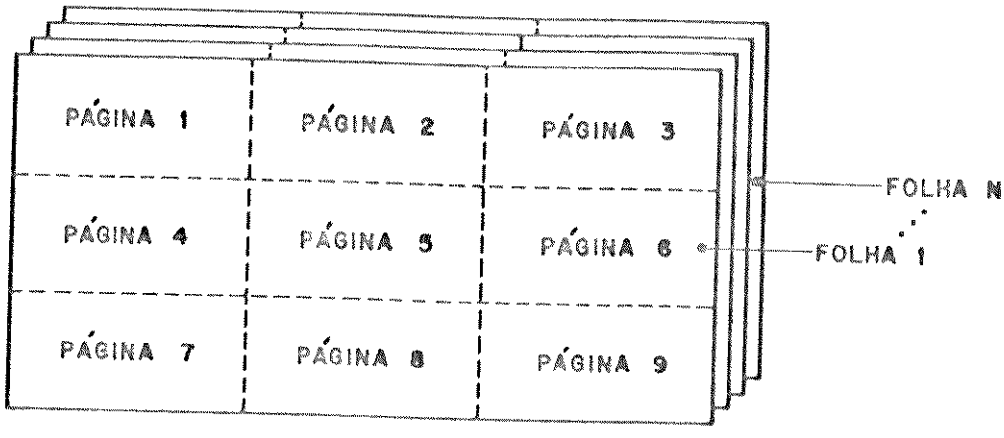


Figura A1. Organização em folhas e páginas de um diagrama esquemático.

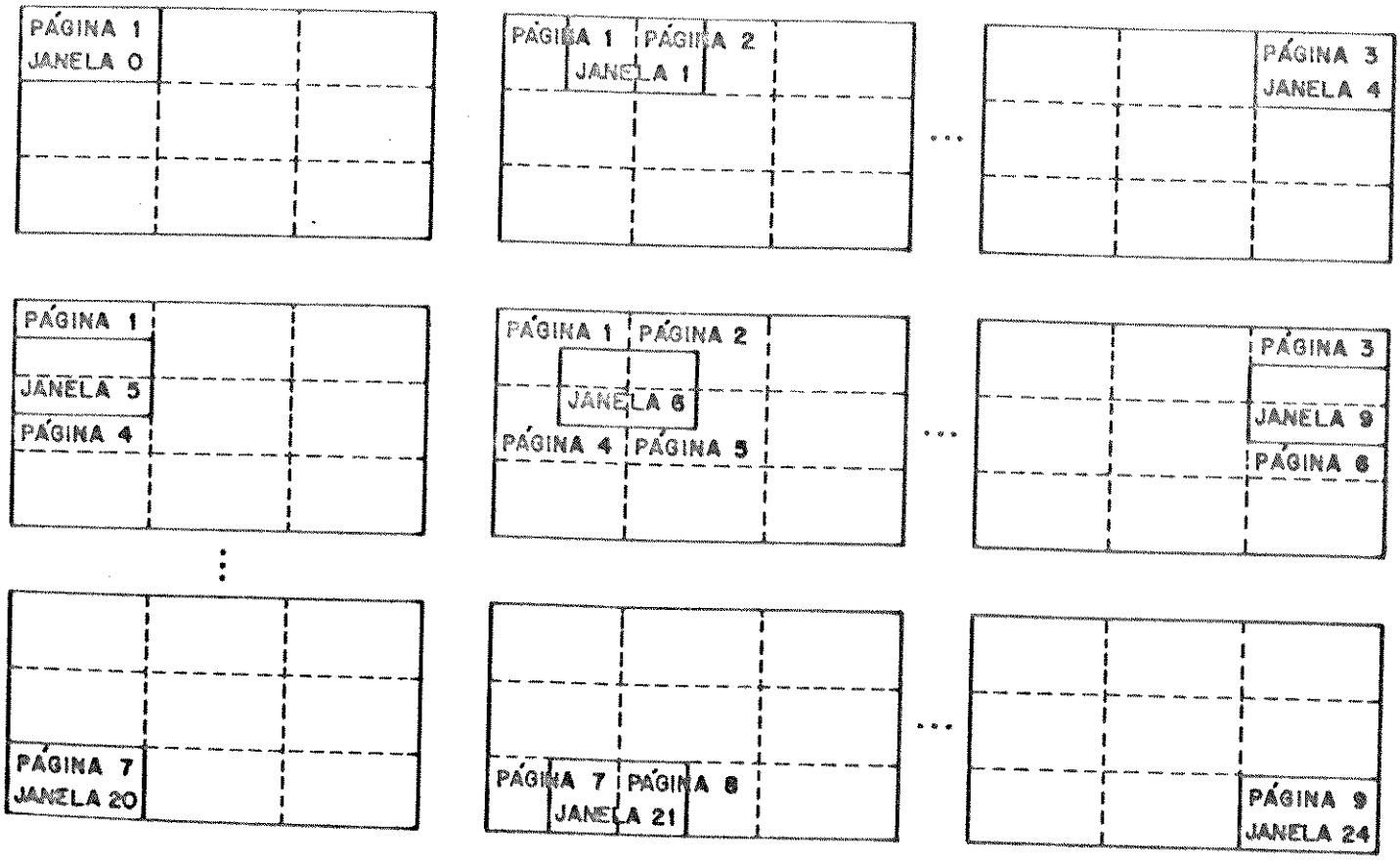


Figura A2. Janelas de visualização.

Funções do menu principal

a) L arq

Quando digitamos "L", o programa pedirá o nome do arquivo a ser lido. Esse nome deverá ter no máximo 8 caracteres. Se ele não achar o arquivo dará a mensagem de "ARQUIVO INEXISTENTE".

b) Edcir

Ao digitarmos "E", o programa estará apto a editar um circuito. As funções de edição de circuito estão descritas no item A2.2.3.

c) impRi

Se digitarmos "R", sairá uma cópia da tela na impressora gráfica.

d) Folha

Quando digitarmos "F", o programa pedirá o número da folha a ser editada e mostrará, se houver, o conteúdo da janela 1 desta folha.

e) Armaz

Ao digitarmos "A", o programa pedirá o nome do arquivo onde armazenará o circuito. Esse nome deverá ter no máximo 8 caracteres. Se for um arquivo novo, ele abrirá este arquivo e dará a mensagem: "ARQUIVO NOVO". Se já existir o arquivo, ele dará a mensagem: "ARQUIVO EXISTENTE", fará uma cópia deste arquivo, com o mesmo nome e extensão ".BCK", e usará o arquivo, cujo nome foi digitado, para armazenar os dados do circuito. Além dessas, esta função apresenta as mensagens "ARMAZENAMENTO DADOS", enquanto estiver guardando os dados em um arquivo, e "FIM" quando esta operação acabar.

f) Sai

Se digitarmos "S", o programa voltará para o sistema operacional.

A2.2.3 Menu de edição de circuito

Quando, no menu principal, digitarmos "E", o programa estará pronto para a edição de um circuito. A figura A4 mostra a tela de edição de circuito. A direita da tela temos o menu com as funções disponíveis.

O posicionamento de elementos e sinais na edição de circuitos, é feito através de um cursor na forma de cruz. O cursor se movimenta na tela através de algumas teclas, como mostra a tabela A1.

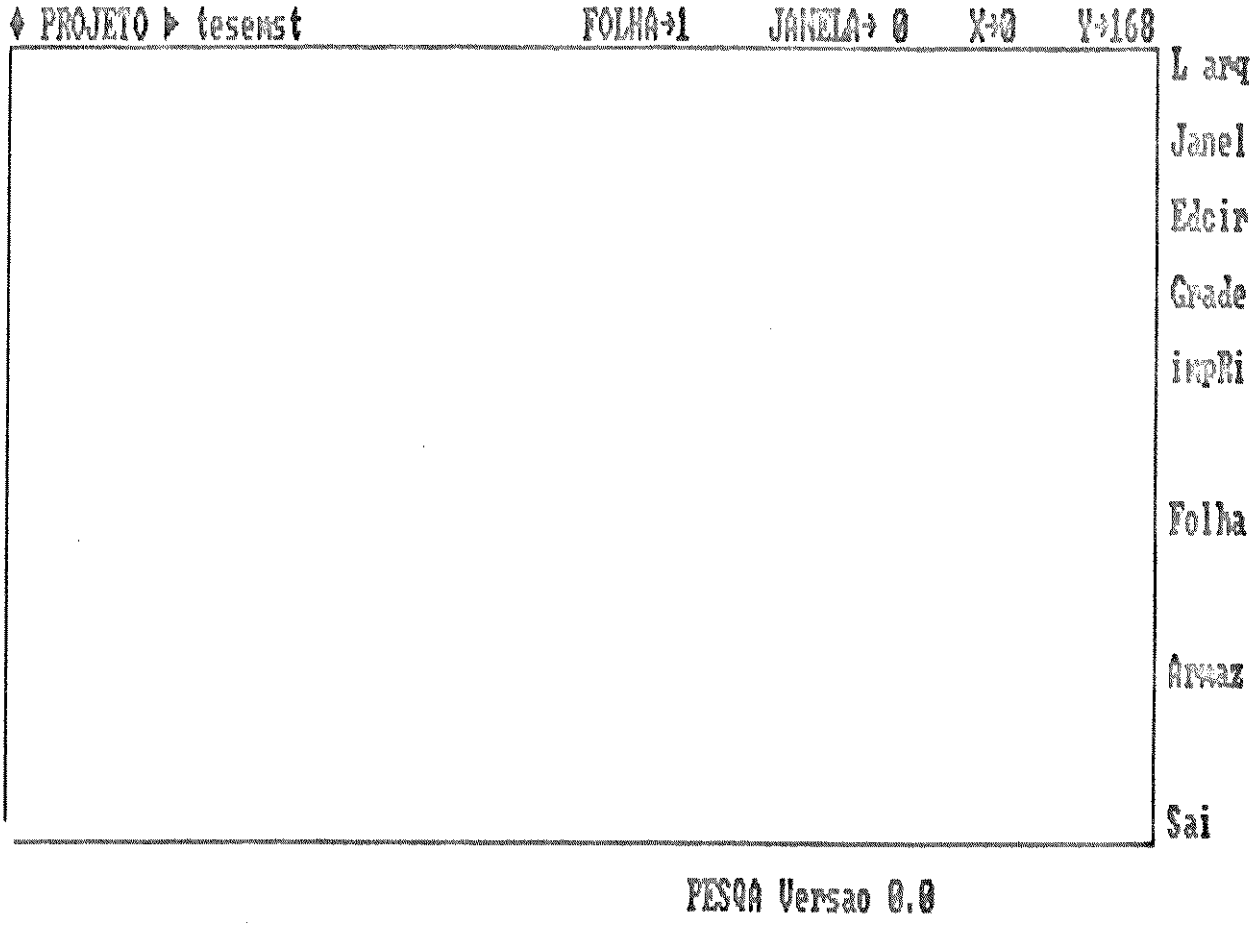


Figura A3. Menu principal do Programa de Entrada Esquemática.

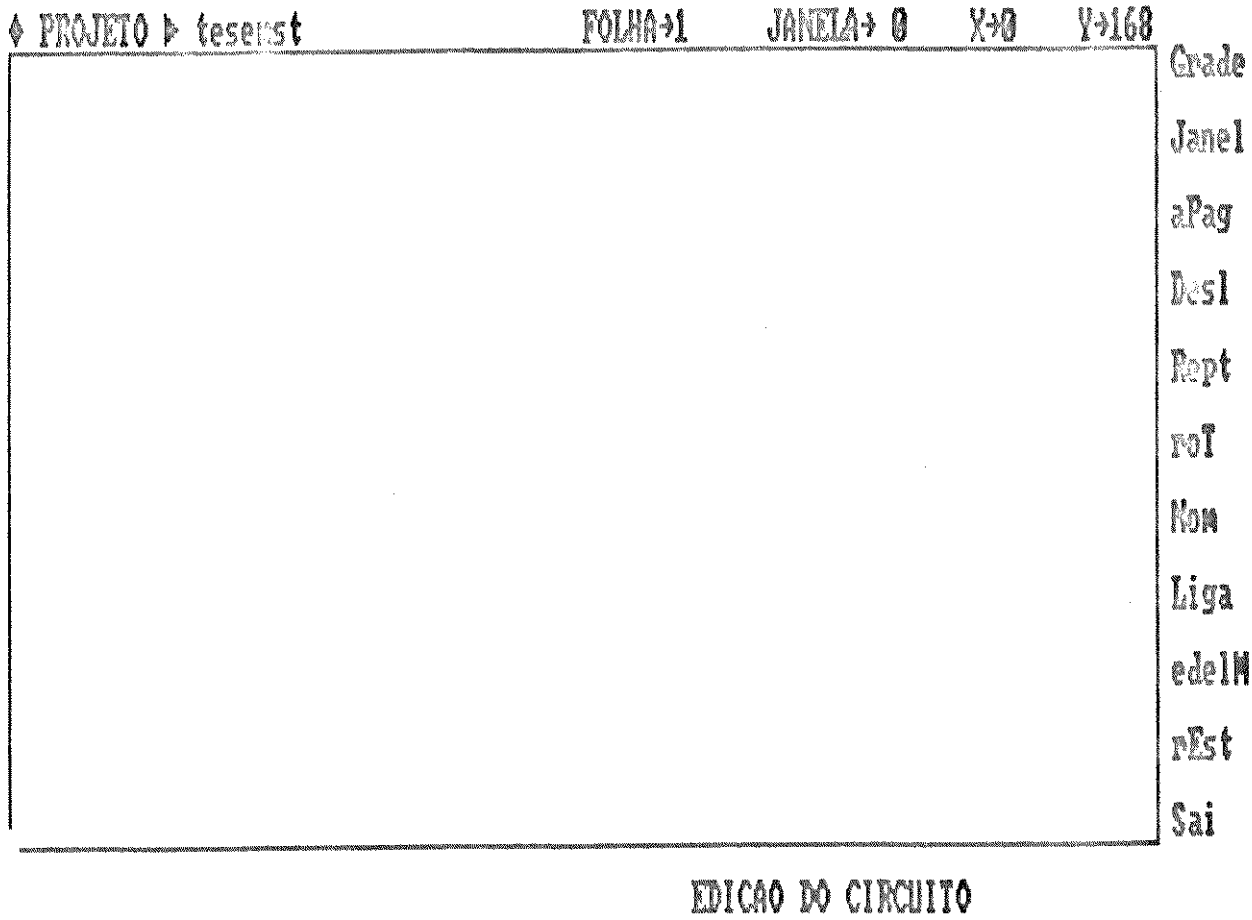


Figura A4. Menu de edição de circuito.

Tabela A1.

Teclas para movimentação do cursor

Tecla	Função
↑	O cursor se desloca um passo para cima.
↓	O cursor se desloca um passo para baixo.
-->	O cursor se desloca um passo para a direita.
<--	O cursor se desloca um passo para a esquerda.
PG UP	O cursor se desloca um passo na diagonal, para cima e para a direita.
PG DN	O cursor se desloca um passo na diagonal, para baixo e para a direita.
HOME	O cursor se desloca um passo na diagonal, para cima e para a esquerda.
END	O cursor se desloca um passo na diagonal, para baixo e para esquerda.
+	Aumenta o passo de deslocamento do cursor.
-	Diminui o passo de deslocamento do cursor.

a) Grade

A tecla "G" permite tanto chamar a grade para exibição na tela como apagá-la.

b) Janel

Ao digitarmos "J", aparecerá o menu indicando as opções de mudança de janela. A figura A5 mostra o menu com as opções de mudança de janela. A tabela A2 mostra as teclas que devem ser digitadas para movimentação das janelas e as respectivas janelas.

c) Funções de transformação

Todas as funções de transformação quando selecionadas, farão aparecer no canto inferior esquerdo do vídeo a mensagem:

"ELEMENTO SINAL BLOCO".

Digitando "E", a função será aplicada a um elemento, "S" a um sinal e "B" a um bloco, ou seja, conjunto de elementos e/ou sinais.

Para a seleção de um elemento, o usuário deverá posicionar o cursor na coordenada mais à esquerda de qualquer entrada e digitar "RET".

Para a seleção de um sinal, o usuário deverá posicionar o cursor na coordenada mais à direita de qualquer ponto do sinal.

Para a seleção de um bloco o usuário deverá ressaltar o bloco com um retângulo. Para isto deve-se marcar os pontos, opostos inferior e superior, da diagonal do retângulo digitando "RET" em cada ponto.

Em todas as funções de transformação, o programa pedirá confirmação do elemento, sinal, ou bloco selecionado.

São os seguintes os pontos de referência para cada transformação:

Elemento: ponto mais a esquerda do primeiro sinal de entrada.

Sinal: ponto inicial do sinal.

Bloco: ponto inferior da diagonal do retângulo de seleção.

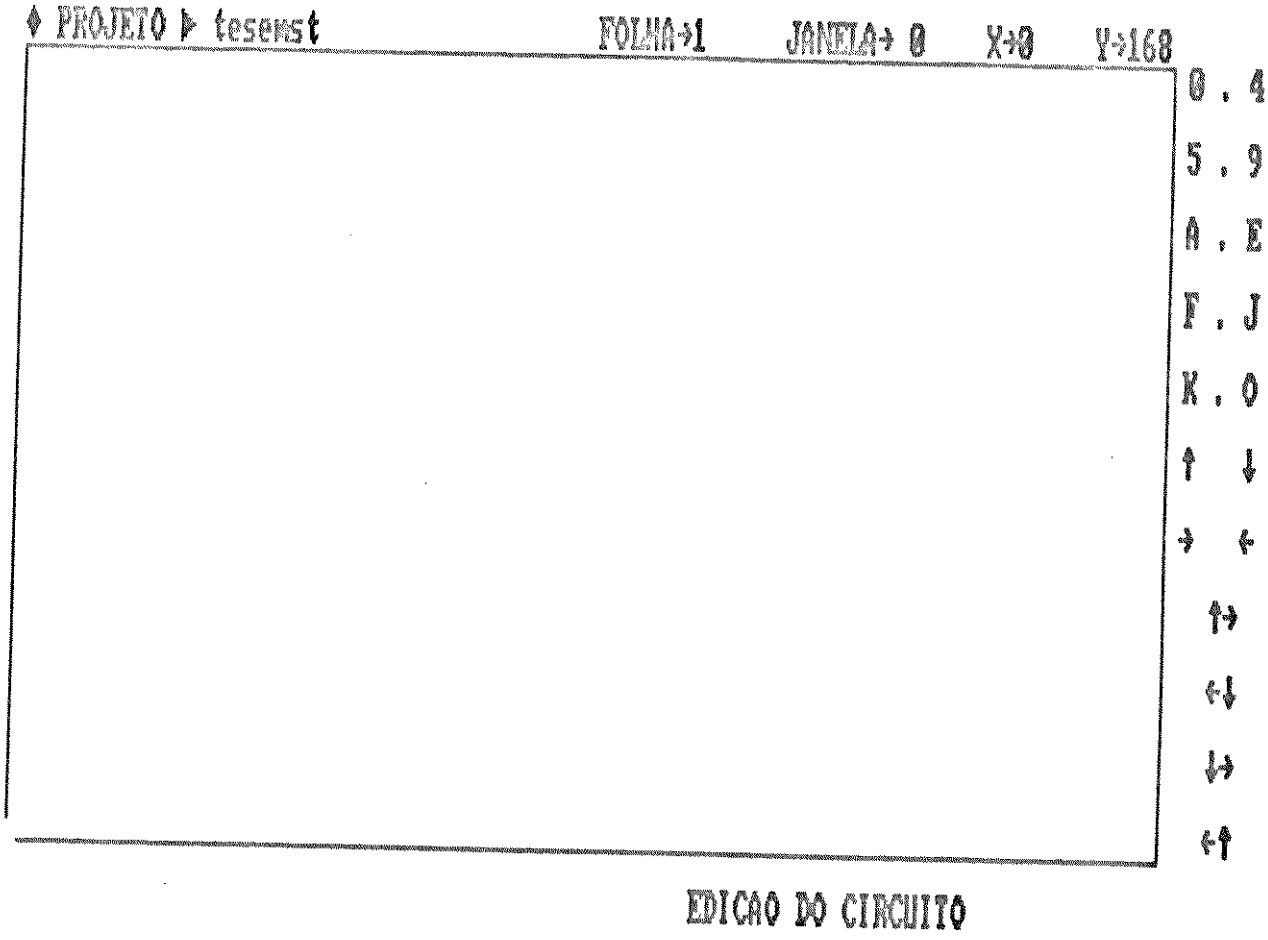


Figura A5. Opção de mudança de janela.

Tabela A2.

Teclas para movimentação da janela de visualização

Tecla	Função
0	Vai para a janela 0
1	Vai para a janela 1
2	Vai para a janela 2
3	Vai para a janela 3
4	Vai para a janela 4
5	Vai para a janela 5
6	Vai para a janela 6
7	Vai para a janela 7
8	Vai para a janela 8
9	Vai para a janela 9
A	Vai para a janela 10
B	Vai para a janela 11
C	Vai para a janela 12
D	Vai para a janela 13
E	Vai para a janela 14
F	Vai para a janela 15
G	Vai para a janela 16

Tabela A2.

Teclas para movimentação da janela de visualização (continuação)

Tecla	Função
H	Vai para a janela 17
I	Vai para a janela 18
J	Vai para a janela 19
K	Vai para a janela 20
L	Vai para a janela 21
M	Vai para a janela 22
N	Vai para a janela 23
O	Vai para a janela 24
-->	Anda uma janela para direita
<--	Anda uma janela para esquerda
↑	Anda uma janela para cima
↓	Anda uma janela para baixo
PG UP	Anda uma janela na diagonal, para cima e para direita
PG DN	Anda uma janela na diagonal, para baixo e para direita
HOME	Anda uma janela na diagonal, para cima e para esquerda
END	Anda uma janela na diagonal,para baixo e para esquerda

Funções de transformação do programa

c1) aPag

Ao digitarmos "P", o programa selecionará a função "APAGAR". Após selecionado o que se deseja apagar, o programa irá pedir confirmação através da mensagem "CONFIRMA (S/N)". Se digitarmos "S", ele apagará o que foi selecionado e sairá da função, caso contrário apenas sairá da função aPag.

c2) Desl

Ao digitarmos "D", o programa selecionará a função "DESLOCAR". Após selecionado o que se deseja deslocar, o programa irá pedir confirmação através da mensagem "CONFIRMA (S/N)". Se for digitado "S", a nova posição do que foi selecionado deverá ser dada, através do posicionamento do cursor e da tecla "RET". O programa então apagará o que foi selecionado da posição original desenhando-o na nova posição e sairá da função. Se a resposta da confirmação não for "S", ele sairá da função Desl.

c3) Rept

Ao digitarmos "R", o programa selecionará a função "REPETIR". Após selecionado o que se deseja repetir, o programa irá pedir confirmação através da mensagem "CONFIRMA (S/N)". Se digitarmos "S", para cada nova posição marcada, ele repetirá o que foi selecionado. Esta função terminará quando digitarmos "F". Se a tecla digitada não for "S", ele sairá da função Rept.

c4) roT

Ao digitarmos "T", o programa selecionará a função "RODAR". Após selecionado o que se deseja rodar, o programa irá pedir confirmação através da mensagem "CONFIRMA (S/N)". Se digitarmos "S", o programa perguntará o ângulo de rotação, com a mensagem "ANGULO (90, 180, 270)". Deve-se digitar o ângulo desejado: 90, 180 ou 270 graus. Qualquer número diferente de 90, 180 ou 270, fará com que o ângulo seja 0. Após selecionado o ângulo, para cada nova posição marcada, aparecerá o desenho transformado. Esta função terminará quando digitarmos "F". Se a resposta da confirmação não for "S", ele sairá da função roT.

d) Nom

Ao digitarmos "N", o programa selecionará a função "NOME", e aparecerá:

"ELEMENTO SINAL BLOCO".

d1) elemento: se for digitado "E", deveremos posicionar o cursor em cada elemento (através de uma entrada) e teclar "RET" para selecioná-lo. O programa irá, então, perguntar o nome do elemento. Esse nome deverá ter no máximo 5 caracteres alfanuméricos e não poderá começar pela letra "e". Este comando termina digitando "F".

d2) sinais: se for digitado "S", deveremos posicionar o cursor em cada sinal (através de um ponto do sinal) e digitar "RET", para selecioná-lo. O programa irá, então, perguntar o nome do sinal. Esse nome deverá ter no máximo 5 caracteres alfanuméricos e não poderá começar pela letra "s". Esta função se repetirá até ser digitado "F".

d3) bloco: se for digitado "B", o programa gerará nomes para os sinais e elementos que ainda não foram identificados.

e) Liga

Ao digitarmos L, o programa entrará na função de edição de sinais. O primeiro ponto de um sinal pode ser marcado por "RET" ou "X". Será marcado por "X" se este sinal começar em um ponto de outro sinal, e se os dois sinais devem estar em contado. Será marcado por RET se o sinal começar em qualquer ponto que não seja cruzamento com contato de sinais. O ponto final de um sinal deverá ser marcado por um "RET". Os pontos intermediários são marcados por "C" se não houver cruzamento com contato de sinais e por "X" se houver.

Os sinais só podem ter ramos horizontais ou verticais. Um ramo é um segmento entre dois pontos adjacentes.

A função Liga terminará após o fim do último sinal digitado e teclando "F".

f) edelM

Ao digitarmos "M", o programa entrará na função de edição de elementos. Para editar um elemento, deveremos escolher o ângulo com o qual será desenhado (0, 90, 180, 270), posicionar o cursor e digitar a letra correspondente a cada elemento, mostrada no menu da tela de edição de elementos. Na figura A6 e figura A7 podemos ver os menus de edição de elementos.

Funções do menu de edição de elementos

f1) anguL

Ao digitarmos "L", o programa perguntará qual o ângulo que deverá ser desenhado o elemento, através da mensagem "ANGULO (90,180,270)?". Se esta função não for escolhida, todos os elementos serão desenhados a 0 graus. Quando escolhemos esta função e determinamos um ângulo, todos os elementos serão desenhados neste ângulo, até um novo pedido de alteração.

f2) Inv

Ao digitarmos "I", o programa desenhará um inversor na posição selecionada (pelo movimento de cursor e digitando "RET") e no ângulo desejado (função anguL).

f3) FfD

Ao digitarmos "F" e "D", o programa desenhará um "flip-flop" tipo D, na posição e ângulo desejados.

f4) FfjK

Ao digitarmos "F" e "K", o programa desenhará um "flip-flop" tipo JK, na posição e ângulo desejados.

f5) entRa

Ao digitarmos "R", o programa desenhará um elemento que representa uma entrada do circuito, na posição e ângulo desejados.

f6) saiDa

Ao digitarmos "D", o programa desenhará um elemento que representa uma saída do circuito, na posição e ângulo desejados.

f7) ContE

Ao digitarmos "C" e "E," o programa desenhará um elemento que representa que o sinal é continuação da página anterior.

f8) ContS

Ao digitarmos "C" e "S", o programa desenhará um elemento que representa que o sinal continua na próxima folha.

f9) rEstE

Ao digitarmos "E" duas vezes, o programa restaurará os elementos já desenhados.

f10) segUe

Ao digitarmos "U", o programa apresentará o outro menu de edição de elementos (figura A7).

f11) eXOr

Ao digitarmos "X" e "O", o programa desenhará uma porta tipo "exclusive-or", na posição e ângulos desejados.

f12) eXNor

Ao digitarmos "X" e "N", o programa desenhará uma porta tipo "exclusive-nor",na posição e ângulo desejados.

f13) And n

Ao digitarmos "A", seguido de um número "n" pertencente ao intervalo [1..5], o programa desenhará uma porta tipo "and" de "n" entradas, na posição e ângulo desejados.

f14) NAn n

Ao digitarmos "N", "A", seguido de um número "n" pertencente ao intervalo [1..5], o programa desenhará uma porta tipo "nand" de "n" entradas, na posição e ângulo desejados.

f15) Or n

Ao digitarmos "O", seguido de um número "n", pertencente ao intervalo [1..5], o programa desenhará um OR de "n" entradas, na posição e ângulo desejados.

f16) NOr n

Ao digitarmos "N", "O", seguido de um número "n", pertencente ao intervalo [1..5], o programa desenhará um NOR de "n" entradas, na posição e ângulo desejados.

f17) Volta

Ao digitarmos "V", o programa voltará ao primeiro menu da edição do elemento. (Figura A6)

f18) Sai

Ao digitarmos "S", o programa sairá do modo de edição de elemento.

Na figura A8, podemos ver os vários elementos.

g) rEst

Ao digitarmos "E", o programa perguntará se desejamos redesenhar :

"ELEMENTO SINAL BLOCO"

g1) Elemento

Se digitarmos "E", o programa redesenhará todos os elementos já editados, da janela visível.

g2) Sinal

Se digitarmos "S", o programa redesenhará todos os sinais já editados, da janela visível.

g3) Bloco

Se digitarmos "B", o programa redesenhará todos os elementos e sinais já editados, da janela visível.

h) Sai

Ao digitarmos "S", o programa sairá do modo de edição.

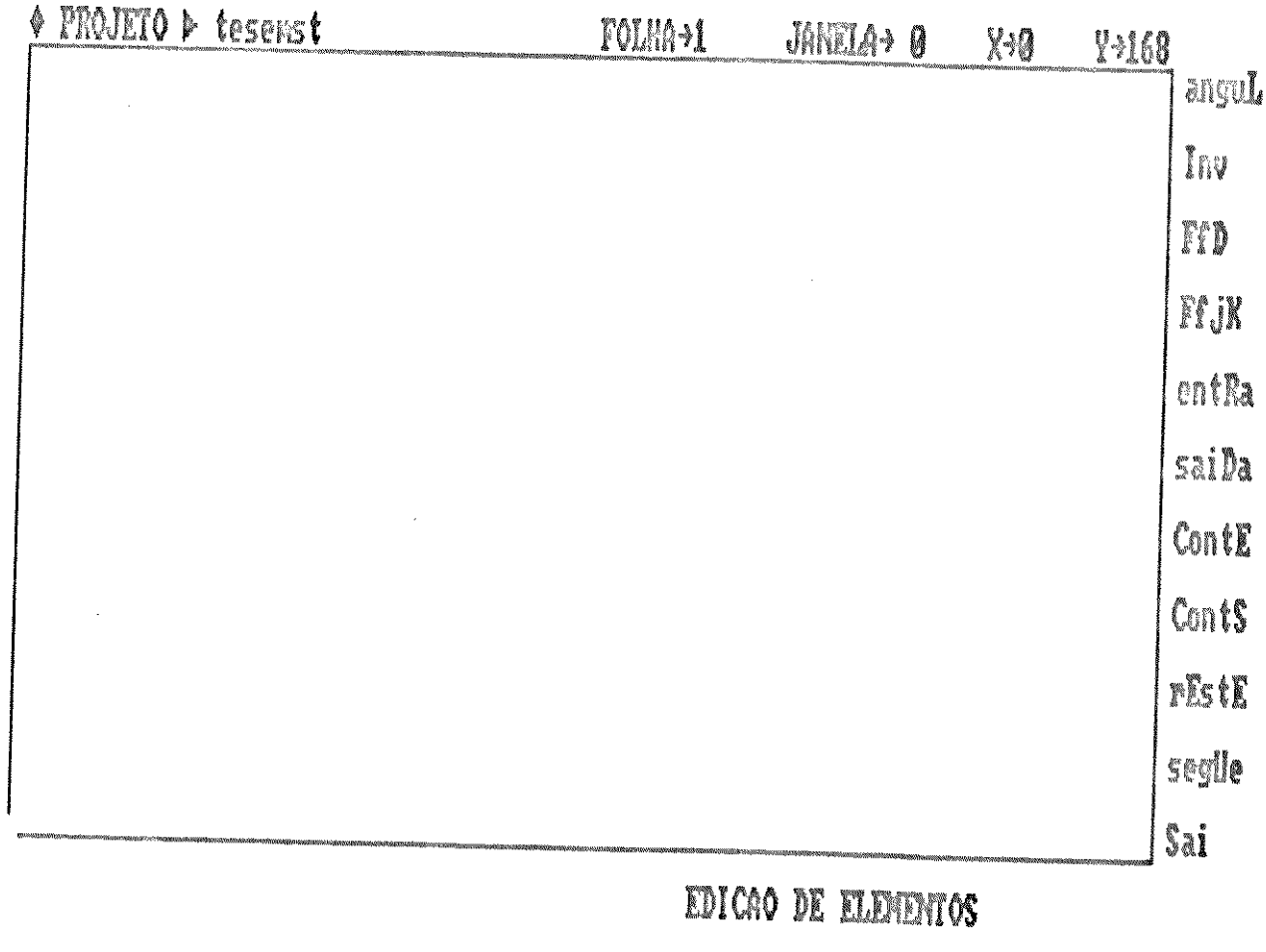


Figura A6. Tela de edição de elementos.

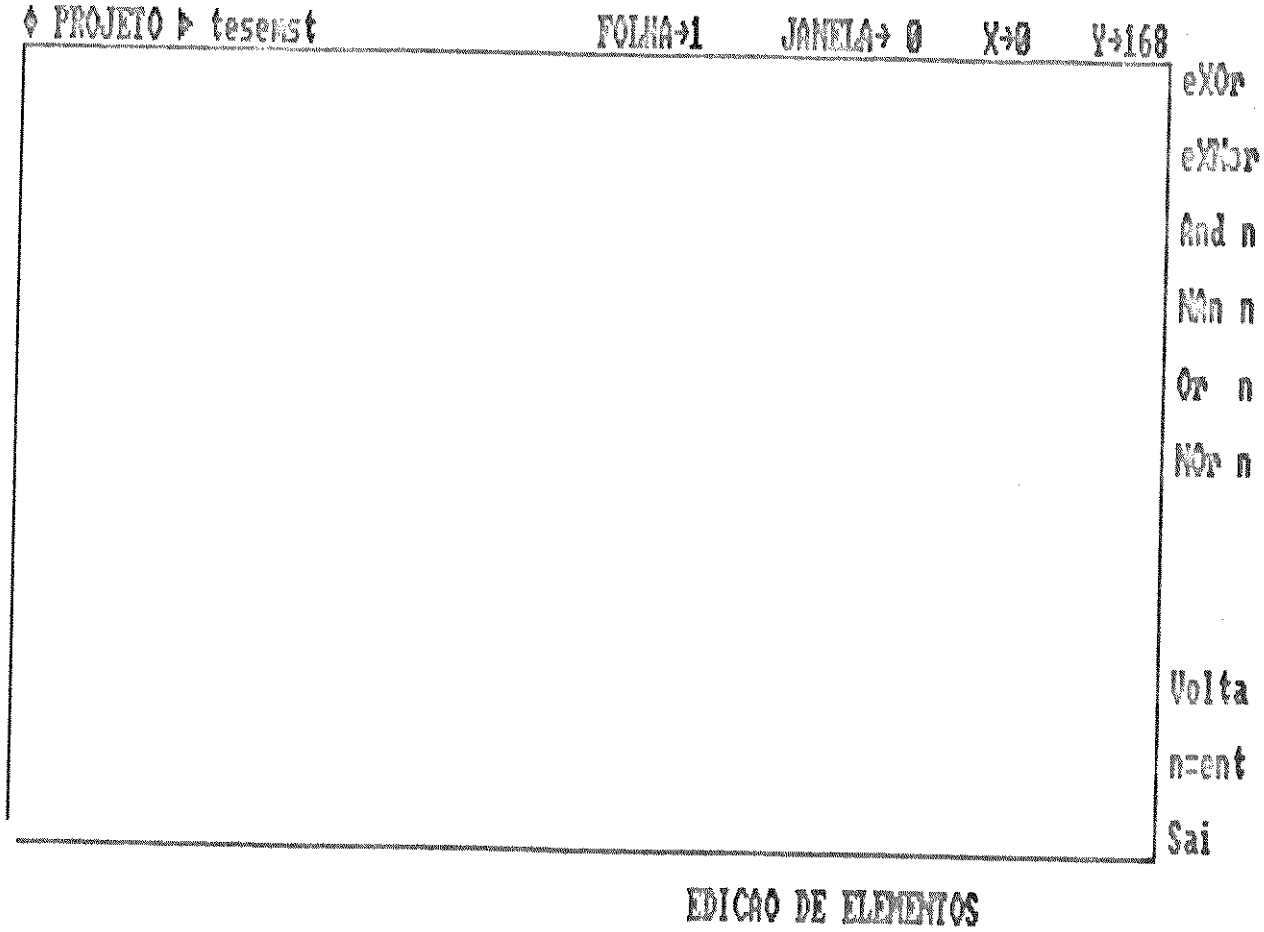
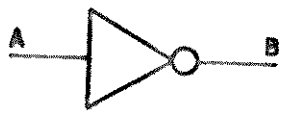


Figura A7. Tela de edição de elementos.



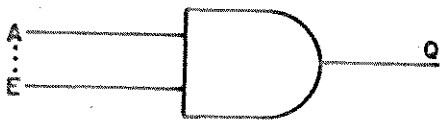
Inversor

Entrada

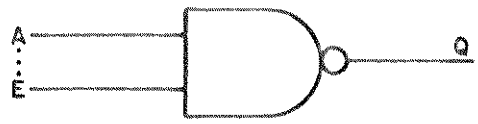
Saída

Continuação-Entrada

Continuação-Saída



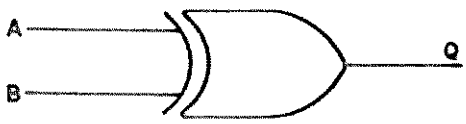
And



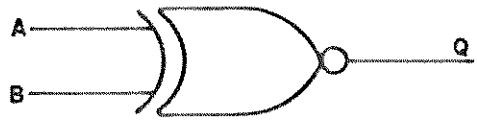
Nand

Or

Nor

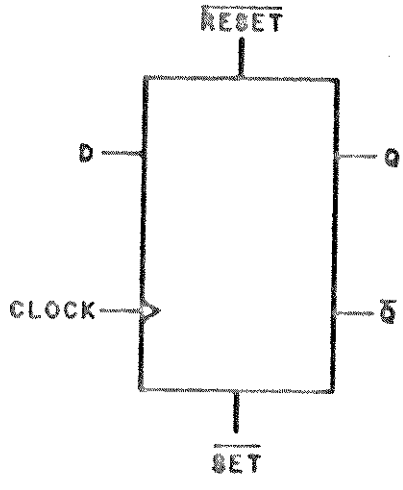


Exor

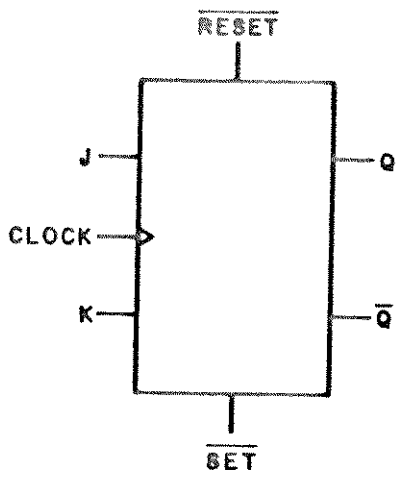


Exnor

Figura A8. Elementos primitivos.



Ffd



Ffjk

Figura A8. Elementos primitivos - (continuação).

APÊNDICE B

PROGRAMA EDITOR DE ESTÍMULOS EXTERNOS

EDTES V. 00

MANUAL DE USUARIO

B1. Introdução

O objetivo deste manual é apresentar ao usuário as principais características do Programa de Edição de Estímulos Externos - EDTES.

Este programa foi desenvolvido na Divisão de Suporte Computacional do Instituto de Microeletrônica, do Centro Tecnológico para Informática, em Campinas, São Paulo, Brasil.

B2. Descrição do sistema

B2.1 Função

Em complementação ao uso do Programa de Entrada Esquemática - PESQA, o Programa de Edição de Estímulos Extenos é necessário para auxiliar o projetista de circuitos digitais, na definição dos estímulos externos, usados na simulação lógica.

No programa EDTES, o desenho dos sinais externos é organizado em folhas, e uma folha pode ter no máximo 10 sinais. Cada folha é dividida em páginas e cada página representa a tela do monitor. A figura B1 mostra essa organização.

A visualização de uma folha na tela do monitor é feita por janelas de viasualização, conforme mostra a figura B2.

B2.2 Descrição do Sistema

B2.2.1 Menu Principal

Ao se iniciar o programa, aparecerá a tela de edição conforme mostra a figura B3. Esta tela apresenta duas réguas, uma no eixo "X" e outra no eixo "Y". A régua do eixo "X" representa o eixo dos tempos. A régua do eixo "Y" representa o eixo dos valores lógicos onde os sinais devem ser editados. A figura B4 mostra como são definidos os valores lógicos.

Cada unidade de grade corresponde ao valor do incremento, que é igual a uma unidade de tempo, exceto se o usuário especificar outro valor usando a função "TEMPO". A função TEMPO também especifica o tempo inicial, ou seja, o valor do primeiro traço da régua de tempos. Se não for especificado, o programa assume o valor zero.

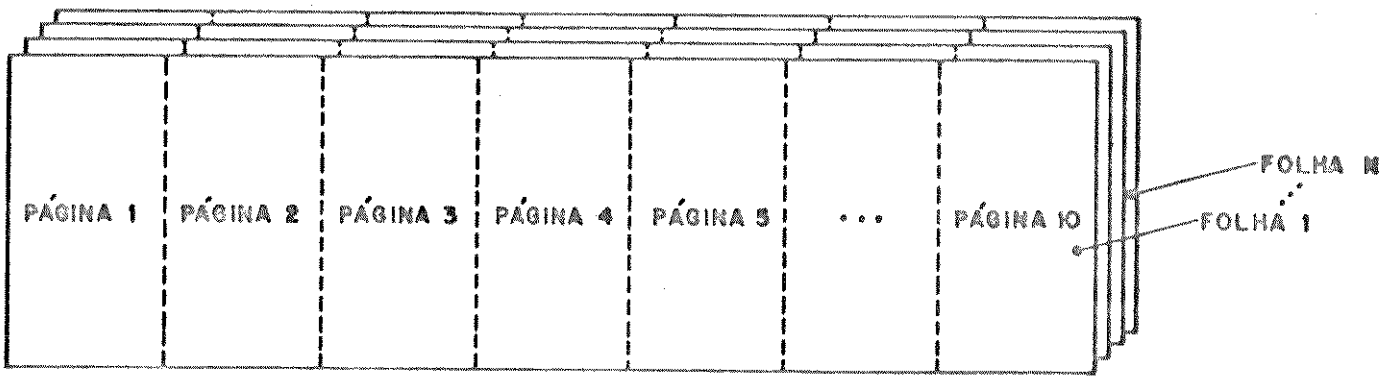


Figura B1. Organização do desenho dos sinais.

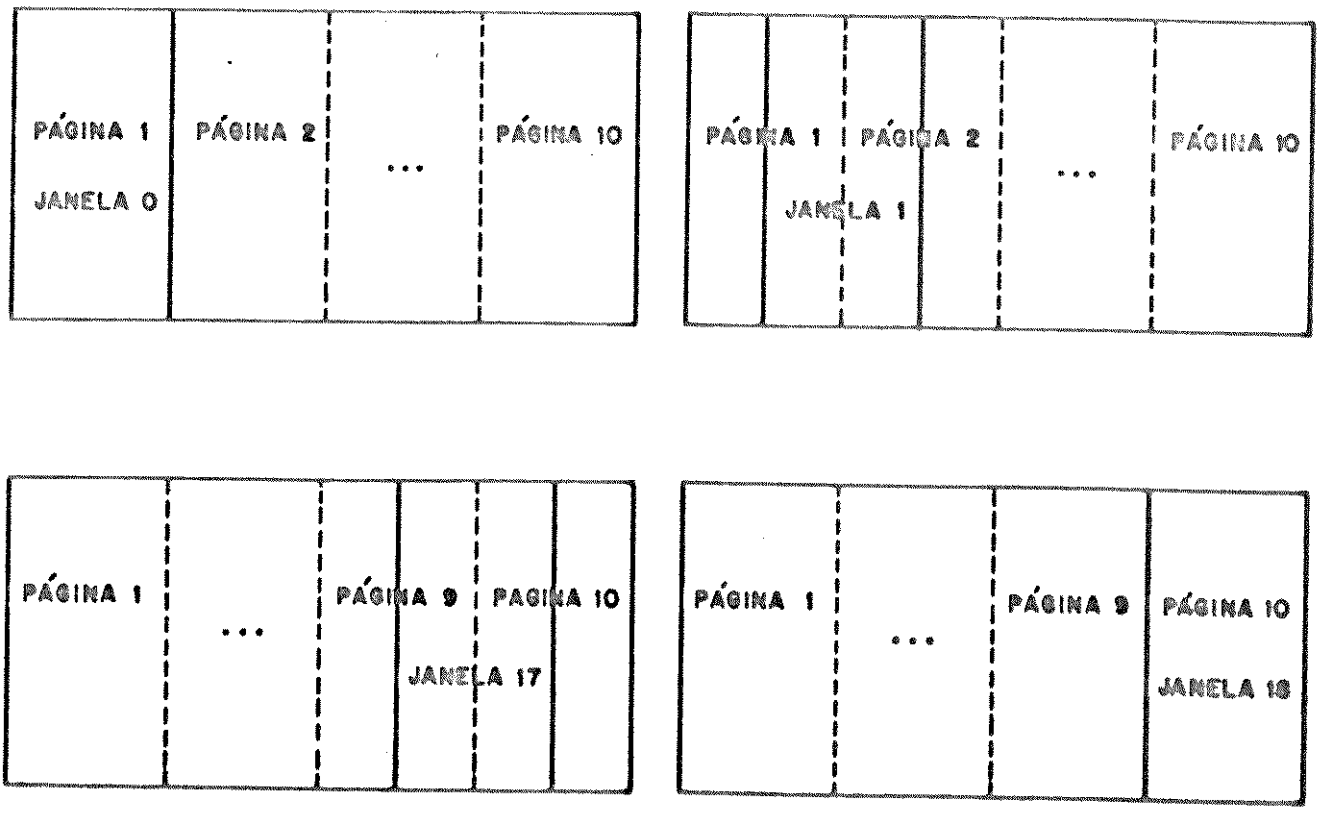


Figura B2. Janelas de visualização.

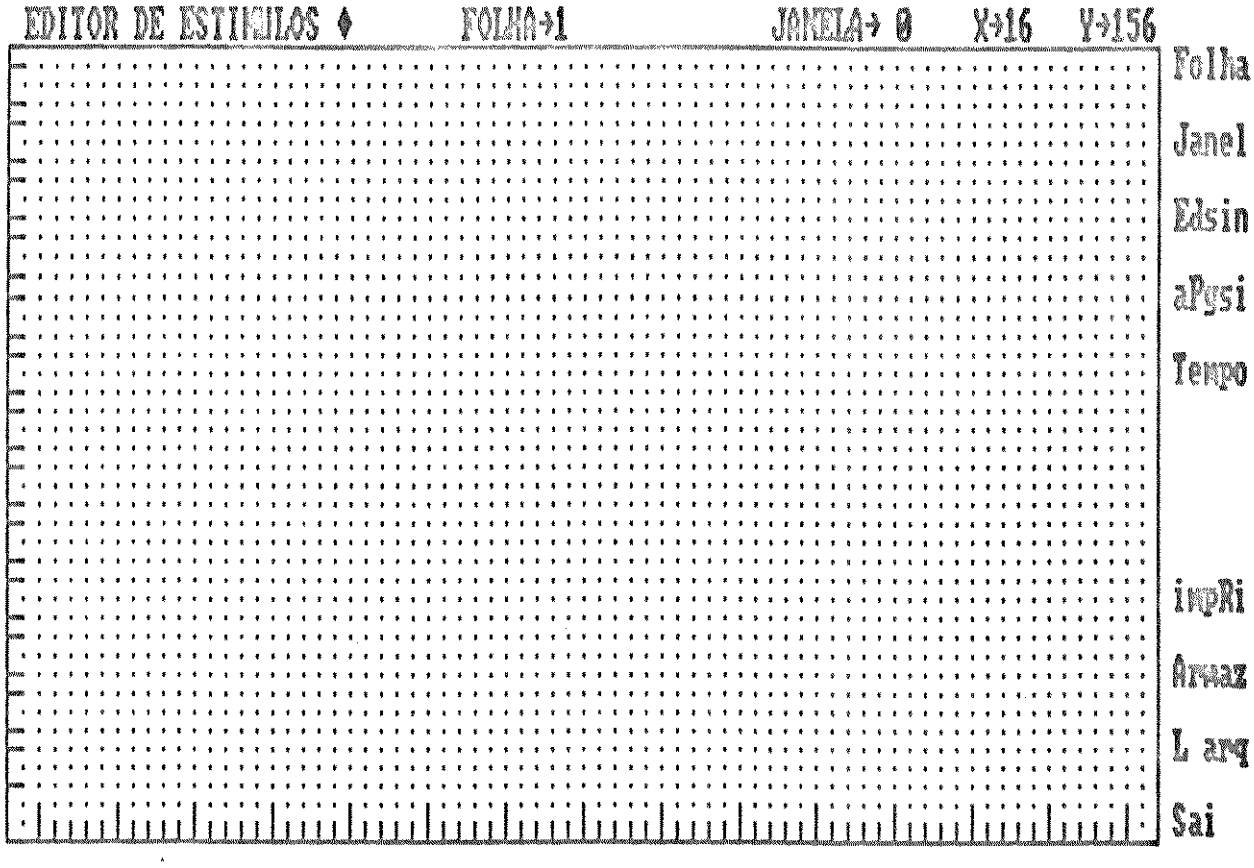


Figura B3. Tela do programa editor de estímulos.

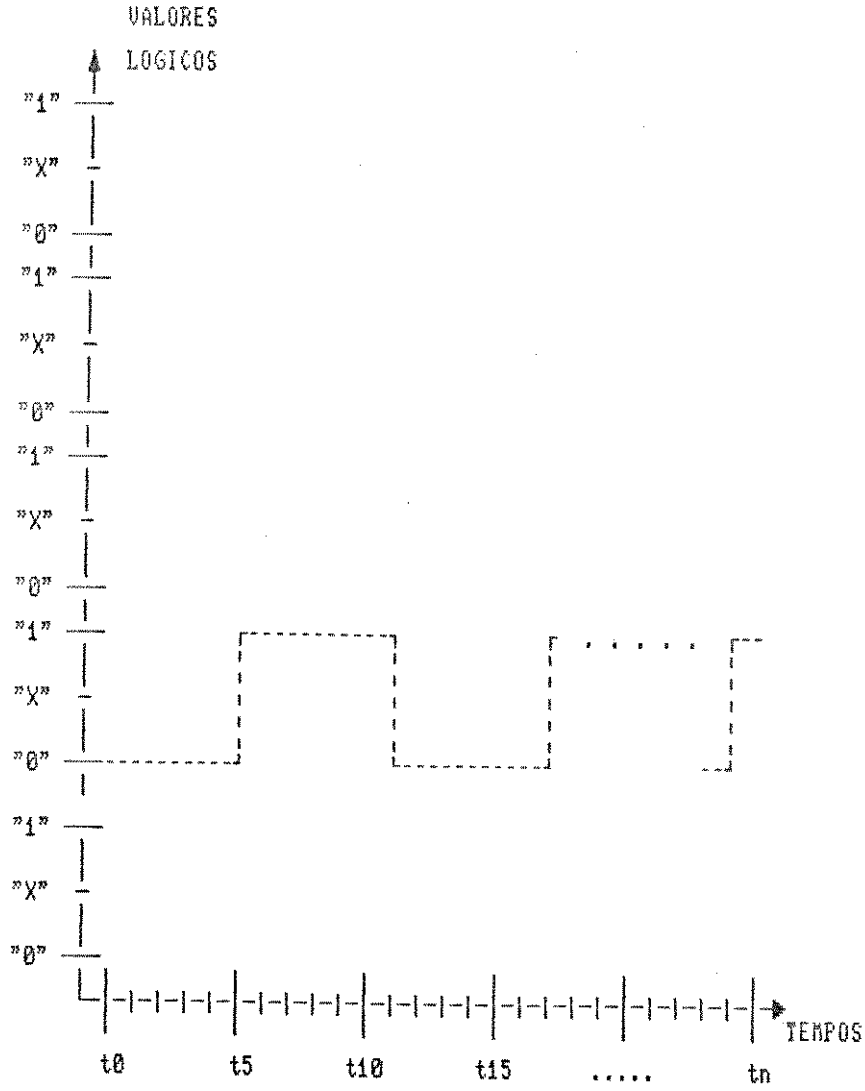


Figura B4. Régua de definição dos valores lógicos.

O posicionamento dos sinais na edição é feito através de um cursor na forma de cruz. Sempre quando se vai editar um sinal, o cursor se posiciona automaticamente no nível lógico zero. O movimento do cursor é feito através de teclas, como mostra a tabela B1.

Para começar a edição dos sinais, primeiro o usuário deverá usar a função "LE ARQUIVO". Existem dois tipos de arquivos:

- saída do programa de entrada esquemática;

- saída do programa editor de estímulos.

O primeiro contém as entradas de um circuito, editado pelo programa PESCA, para alteração. O segundo tem os estímulos já editados, de um circuito, para alteração.

Funções do menu principal

A direita da tela do programa temos um menu com as funções:

a) Folha

Quando digitamos "F", o programa pedirá o número da folha a ser editada, e mostrará, se houver, o conteúdo da janela 1 desta folha.

b) Janel

Ao digitarmos "J", aparecerá o menu indicando as opções de mudança de janela (figura B5).

As opções e as teclas que devem ser acionadas para selecionar uma janela, podem ser vistas na tabela B2.

c) Edsin

Ao digitarmos "E", o programa estará pronto para editar os sinais de entrada originados da saída do programa de entrada esquemática PESCA. Ele mostrará o nome do sinal a ser editado e perguntará o seu tipo através da mensagem:

- "TIPO DO SINAL ".

Os tipos de sinais permitidos são, conforme o Manual de Usuário do Programa SIMUL, A, P ou R.

Tabela B1.

Teclas para movimentação do cursor

Tecla	Função
↑	O cursor se desloca um passo para cima.
↓	O cursor se desloca um passo para baixo.
-->	O cursor se desloca um passo para a direita.
<--	O cursor se desloca um passo para a esquerda.
PG UP	O cursor se desloca um passo na diagonal, para cima e para a direita.
PG DN	O cursor se desloca um passo na diagonal, para baixo e para a direita.
HOME	O cursor se desloca um passo na diagonal, para cima e para a esquerda.
END	O cursor se desloca um passo na diagonal, para baixo e para esquerda.
+	Aumenta o passo de deslocamento do cursor.
-	Diminui o passo de deslocamento do cursor.

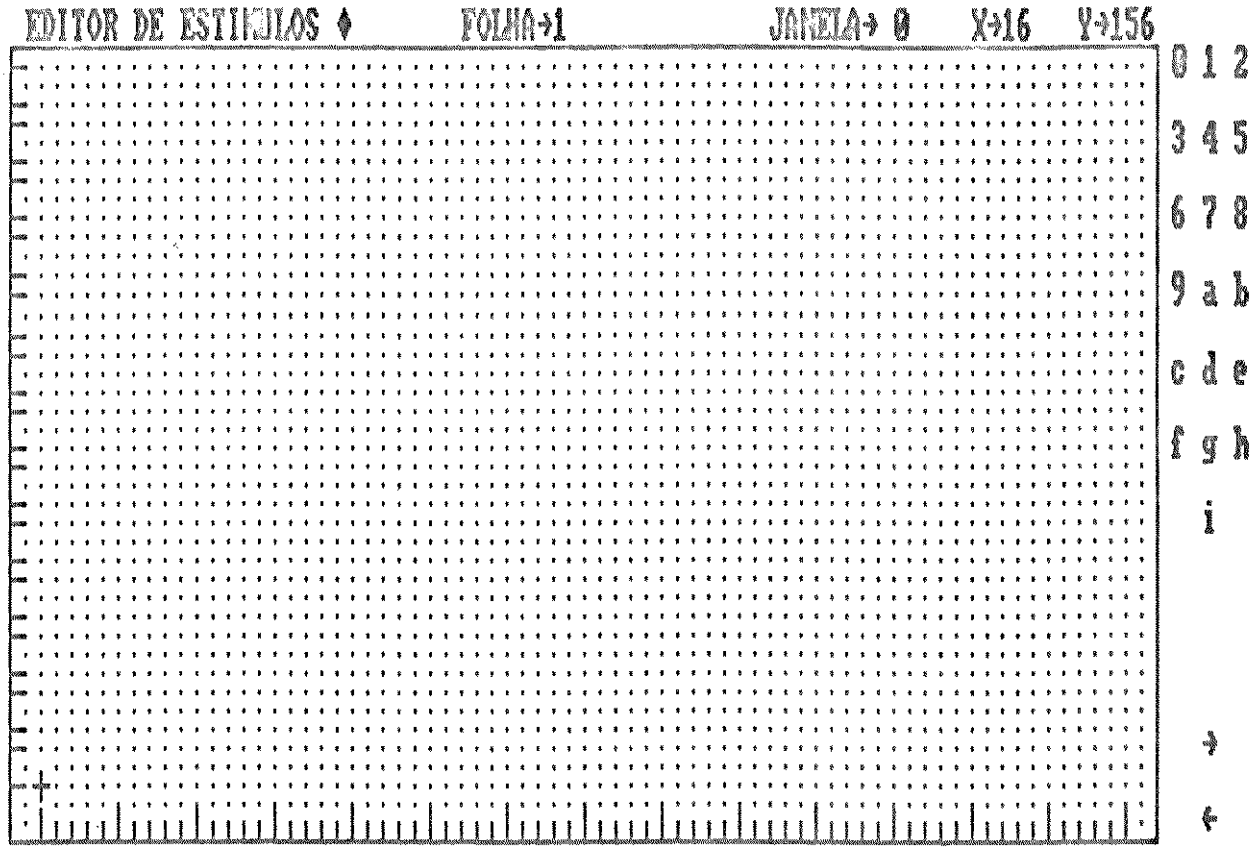


Figura B5. Opções de mudança de janela.

Tabela B2.

Teclas para movimentação da janela de visualização

Tecla	Função
0	Vai para a janela 0
1	Vai para a janela 1
2	Vai para a janela 2
3	Vai para a janela 3
4	Vai para a janela 4
5	Vai para a janela 5
6	Vai para a janela 6
7	Vai para a janela 7
8	Vai para a janela 8
9	Vai para a janela 9
-->	Anda uma janela para direita
<--	Anda uma janela para esquerda
↑	Anda uma janela para cima
⇓	Anda uma janela para baixo
PG UP	Anda uma janela na diagonal, para cima e para direita
PG DN	Anda uma janela na diagonal, para baixo e para direita
HOME	Anda uma janela na diagonal, para cima e para esquerda
END	Anda uma janela na diagonal, para baixo e para esquerda

O desenho do sinal é feito da seguinte maneira:

- digitar "RET" para o primeiro ponto do sinal;
- para cada mudança de nível lógico digitar "C";
- digitar "RET" para terminar a edição do sinal.

Após editados todos os sinais, o programa encerrará a função Edsin.

d) aPgsi

Ao digitarmos "P", o programa estará apto a apagar sinais já editados.

Devemos posicionar o cursor no início do sinal que se quer apagar e digitar "RET". O programa então, pedirá a confirmação através da mensagem:

- "CONFIRMA (S/N)?".

Se a resposta for "S", o sinal será apagado, caso contrário não. Esta função terminará digitando "F".

Após o término dessa função, o programa automaticamente fará com que os sinais apagados sejam editados.

e) Tempo

Ao digitarmos "T", o programa pedirá:

- "TEMPO INICIAL ="

Tempo inicial é o valor que corresponde ao primeiro traço da régua de tempos.

- "INCREMENTO ="

Incremento é o valor de uma divisão na régua de tempos.

Os valores máximos para cada um desses parâmetros é 32.000.

f) impRi

Ao digitarmos "R", sairá uma cópia da tela na impressora gráfica.

g) Armaz

Quando digitamos "A", o programa poderá armazenar dois tipos de saídas.

g1) siMul

Se for digitado "M", o programa armazenará um arquivo que será entrada do programa SIMUL.

Esta função solicita o nome do arquivo. Esse nome deverá ter no máximo 8 caracteres.

g2) edesT

Se for digitado "T", o programa armazenará um arquivo que será entrada do próprio editor de estímulos.

Esta função solicita o nome do arquivo. Esse nome deverá ter no máximo 8 caracteres.

h) L arq

Quando digitamos "L", o programa poderá ler dois tipos de arquivos.

h1) pesQa

Se for digitado "Q", ele lerá um arquivo que é saída do programa de entrada esquemática PESQA.

Esta função solicita o nome do arquivo. Esse nome deverá ter no máximo 8 caracteres.

h2) edesT

Se for digitado "T", ele lerá um arquivo que é saída do próprio editor de estímulos.

Esta função solicita o nome do arquivo. Esse nome deverá ter no máximo 8 caracteres.

i) Sai

Ao digitarmos "S", o programa terminará sua execução.