

177484-9

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**O Sistema TENTOS for WINDOWS -
Um gerenciador de ferramentas para Microeletrônica**

por
LUIZ GUSTAVO GALVES MÄHLMANN

Dissertação submetida à avaliação, como requisito parcial para
a obtenção do grau de Mestre em
Ciência da Computação

Prof. Dr. Ricardo Augusto da Luz Reis
Orientador

Porto Alegre, outubro de 1996.

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

334 6 A 76
Máhlmann

ORIGEM: II
FUNDOS: II

CIP - Catalogação na Publicação

Mählmann, Luiz Gustavo Galves

O Sistema TENTOS for WINDOWS - Um gerenciador de ferramentas para Microeletrônica / Luiz Gustavo Galves Mählmann. - Porto Alegre: CPGCC da UFRGS, 1996.

93 p.: il.

Dissertação (Mestrado)- Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BR - RS, 1996. Orientador: Reis, Ricardo Augusto da Luz.

1. Gerenciador de Ferramentas. 2. Microeletrônica. 3. CAD para PC. I. Reis, Ricardo Augusto da Luz, orient. II. Título.

Universidade Federal do Rio Grande do Sul

Reitor: Prof. Hégio Casses Trindade

Pró-Reitor de Pesquisa e Pós-Graduação: Prof. Cláudio Scherer

Diretor do Instituto de Informática: Prof. Roberto Tom Price

Coordenador do CPGCC: Prof. Flávio Wagner

Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira



Agradecimentos

Agradeço a todos aqueles que tornaram este trabalho possível: ao professor Ricardo Augusto da Luz Reis, pela orientação, apoio e amizade. A Fernando Moraes, pelas valiosas idéias, paciência e amizade. A Marcus Zart, bolsista de Aperfeiçoamento, Rafael Sparta e João Leonardo Fragoso, bolsistas de Iniciação Científica, pela dedicação demonstrada durante os trabalhos de implementação realizados ao longo do período.

Agradeço aos colegas e professores deste curso de pós-graduação, em especial àqueles que formam o grupo de microeletrônica, pela amizade e comentários realizados, que em muito auxiliaram na definição do rumo a ser seguido no decorrer deste trabalho. Dentre estes é indispensável destacar os colegas Marcus Kindel e Marcelo Johann pelos testes de campo que realizaram com o TENTOS, ministrando cursos e demonstrações em congressos internacionais, pois somente desta forma foi possível constatar determinadas correções, que tinham que ser feitas, a fim de melhorar o uso do Sistema.

Foi também muito valiosa a colaboração dos professores que constituíram a banca do seminário de andamento deste trabalho, Prof. Ricardo A. L. Reis, Prof. Tiaraju V. Wagner, Fernando G. Moraes e Fabian Vargas, ao realizarem comentários fundamentais para definição dos trabalhos de finalização necessários, e ao transmitir o apoio e estímulo necessários à conclusão deste trabalho.

Agradeço também ao CNPq pelo auxílio financeiro, recebido em forma de bolsa, durante o Curso de Pós-Graduação.

Finalmente agradeço, mas de modo algum por último, a minha namorada Karen, aos meus pais, irmãs, amigos e familiares, pelo apoio sempre certo e espontâneo.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Sistema de Biblioteca da UFRGS

621.38-181.4(043)
M2148

13245
INF
1997/177484-9
1997/01/31

Sumário

Lista de Figuras.....	7
Lista de Tabelas	9
Lista de Abreviaturas	10
Resumo.....	11
Abstract	12
1 Introdução	13
1.1 Motivação	13
1.2 Objetivos.....	14
2 Gerenciadores de Projeto: uma Revisão	16
2.1 Ambientes Convencionais de Projeto - Conceito.....	16
2.2 Ambiente Integrados e Abertos - Conceito.....	16
2.3 Frameworks - Conceito	17
2.4 Apresentação das Ferramentas	17
2.5 Aplicação das Ferramentas.....	18
2.6 Integração de novas ferramentas.....	19
2.7 Conclusão.....	20
3 Histórico do Sistema TENTOS	21
3.1 A Primeira versão	21
3.2 A Primeira Versão para ambiente MS-WINDOWS™	23
3.3 Conclusão.....	25
4 O Sistema TENTOS <i>for Windows</i>	26
4.1 As Características do Sistema.....	26
4.1.1 A independência da configuração do Sistema.....	27
4.1.2 Associando novas Ferramentas ao TENTOS.....	28
4.1.2.1 Lista de Variáveis	30
4.1.3 Associando novos arquivos de tecnologia ao TENTOS.....	31
4.1.4 Associando novas extensões de circuitos ao TENTOS	32
4.2 A Estrutura do Sistema TENTOS	33
4.2.1 Submenu <i>Arquivo</i>	34
4.2.1.1 Submenu <i>Arquivo</i> - Configuração.....	36
4.2.1.2 Submenu <i>Arquivo</i> - Configuração - Opções.....	36
4.2.1.3 Submenu <i>Arquivo</i> - Configuração - Opções - Tecnologia	36
4.2.1.4 Submenu <i>Arquivo</i> - Configuração - Idioma	37
4.2.2 Submenus <i>Descrições, Sintetizar, Ferramentas e Simuladores</i>	37
4.2.3 Submenu <i>Instalar</i>	37
4.3 A Configuração <i>Standard</i> do TENTOS	37
4.3.1 Menu <i>Descrições</i>	38
4.3.1.1 Editor de Leiaute - EMA	38

4.3.1.2	Editor de Leiaute - SELA	39
4.3.1.3	Edição Textual do Circuito	40
4.3.1.4	Editor de Esquemáticos	40
4.3.1.5	Editor de Símbolos	41
4.3.1.6	Editor Simbólico - SIMBY	42
4.3.2	Menu <i>Sintetizar</i>	43
4.3.2.1	Trago	43
4.3.2.2	Tropic	45
4.3.3	Menu <i>Ferramentas</i>	47
4.3.3.1	Conversor Formato Leiaute SEL p/ Leiaute RS	47
4.3.3.2	Conversor Formato Esquemático ↔ Formato Spice	48
4.3.3.3	Verificador das Regras de Projeto	48
4.3.3.4	Conversor de Formato Leiaute RS p/ Formato Leiaute CIF	48
4.3.3.5	Extrator Elétrico	48
4.3.3.6	Planificador (Expansor)	49
4.3.3.7	Compactador - SYLC	49
4.3.3.8	Extrator Lógico (Extralo)	49
4.3.3.9	Comparador de NetLists - COTONET	50
4.3.4	Menu <i>Simuladores</i>	51
4.3.4.1	Simulador Spice	51
4.4	Executando os aplicativos.....	51
4.4.1	Execução com Caixa de Diálogo	51
4.4.2	Execução Direta	52
4.5	Conclusão.....	52
5	Implementações Futuras	53
5.1	Recursos Fornecidos pelo MS-WINDOWS™	53
5.1.1	Gerenciador de Impressão	53
5.1.2	DDE (Intercâmbio Dinâmico de Dados)	54
5.1.3	OLE (Incorporação e Vinculação de Objetos).....	54
5.1.4	DLL (Bibliotecas de Ligações Dinâmicas).....	54
5.1.5	Área de Transferência.....	55
5.1.6	Demais recursos	55
5.2	Próximos Passos	55
5.2.1	Editor de Esquemáticos	55
5.2.2	Editor de Leiautes.....	56
5.2.3	Gerador de Relatórios.....	57
5.2.4	Conversão Automática de Formatos.....	57
5.2.5	Arquivo de Projeto	57
5.3	Conclusão.....	58
6	Projeto de um Circuito Integrado Utilizando o Sistema TENTOS	59
6.1	Projeto de um Circuito Utilizando o Sistema TRAGO	60
6.2	Projeto de um Circuito Utilizando o Sistema TROPIC	71

6.3 Tutorial de edição manual do leiaute.....	78
6.4 Usando o Sistema TENTOS no Futuro.....	78
6.5 Conclusão.....	79
7 Conclusões	80
Anexo 1 Listagem do Arquivo <i>TENTOS.DAT</i>	81
Anexo 2 Listagem do Arquivo <i>TENTOS.INI</i>	82
Anexo 3 Listagem do Arquivo <i>TENTOSUS.CFG</i>	86
Bibliografia.....	90

Lista de Figuras

FIGURA 1.1 - Processo evolutivo hardware e software	13
FIGURA 3.1 - Interface do TENTOS para DOS	22
FIGURA 3.2 - Estrutura do sistema TENTOS	22
FIGURA 3.3 - Interface do TENTOS para MS-WINDOWS™	24
FIGURA 4.1 - A Nova Interface do Sistema TENTOS for Windows	26
FIGURA 4.2 - Janela de Instalação/Configuração das Ferramentas	29
FIGURA 4.3 - Janela de Instalação/Configuração dos Arquivos de Tecnologia	32
FIGURA 4.4 - Janela de Instalação/Configuração dos Tipos de Extensões	33
FIGURA 4.5 - A estrutura de Menus do Sistema TENTOS	34
FIGURA 4.6 - Caixa de Diálogo - Projeto (Informações do Projeto)	35
FIGURA 4.7 - A disposição das ferramentas nos menus do Sistema TENTOS	38
FIGURA 4.8 - Interface do Editor de Máscaras - EMA	39
FIGURA 4.9 - Interface do Editor de Máscaras - SELA	40
FIGURA 4.10 - Interface do Editor de Esquemáticos	41
FIGURA 4.11 - Interface do Editor de Símbolos	42
FIGURA 4.12 - Interface do Editor Simbólico	43
FIGURA 4.13 - Seqüência de síntese do TRAGO	44
FIGURA 4.14 - Interface do Sistema Trago	44
FIGURA 4.15 - Seqüência de síntese do TROPIC	45
FIGURA 4.16 - Interface do Sistema Tropic	47
FIGURA 4.17 - Interface do Conversor de Formatos SEL ↔ RS	48
FIGURA 4.18 - Interface do Extrator Lógico (EXTRALO)	50
FIGURA 4.19 - Interface do Comparador de NetLists (COTONET)	50
FIGURA 4.20 - Janela de Apresentação dos Parâmetros antes da Execução das Ferramentas	51
Figura 5.1 - Fluxo de dados com comunicação via DDE	54
FIGURA 6.1 - Fluxo das etapas de projeto com geração automática do leiaute	59
FIGURA 6.2 - Descrição do circuito <i>c:\tentosus\adder_tg.sim</i>	60
FIGURA 6.3 - Janela de Edição Textual com o arquivo <i>c:\tentosus\adder_tg.sim</i>	62
FIGURA 6.4 - Sistema TRAGO com os parâmetros certos para a geração do processo de síntese do arquivo <i>c:\tentosus\adder_tg.sim</i>	63
FIGURA 6.5 - Sistema TRAGO - preparação do circuito <i>c:\tentosus\adder_tg.sim</i>	64
FIGURA 6.6 - Sistema TRAGO - compilação do circuito <i>c:\tentosus\adder_tg.sim</i>	64
FIGURA 6.7 - Sistema TRAGO - particionamento do circuito <i>c:\tentosus\adder_tg.sim</i>	65
FIGURA 6.8 - Sistema TRAGO - síntese do circuito <i>c:\tentosus\adder_tg.sim</i>	65
FIGURA 6.9 - Leiaute do circuito <i>c:\tentosus\adder_tg.sim</i> visualizado pelo EMA	66
FIGURA 6.10 - Interface do conversor de formatos SEL↔RS	67
FIGURA 6.11 - Leiaute do circuito <i>c:\tentosus\adder_tg.sim</i> visualizado pelo SELA	68
FIGURA 6.12 - Planificação do circuito <i>c:\tentosus\adder_tg.sim</i> gerado pelo TRAGO	69
FIGURA 6.13 - Extração da <i>Netlist</i> do circuito <i>c:\tentosus\adder_tg.rs</i> gerado pelo TRAGO	69
FIGURA 6.14 - Extração lógica do circuito <i>c:\tentosus\adder_tg.sim</i> gerado pelo TRAGO	70
FIGURA 6.15 - Verificação das regras de projeto do circuito <i>c:\tentosus\adder_tg.sim</i>	70
FIGURA 6.16 - Comparação dos <i>netlists</i> referentes ao circuito <i>c:\tentosus\adder_tg.sim</i>	71
FIGURA 6.17 - Janela de Edição Textual com o arquivo <i>c:\tentosus\adder.sim</i>	73

FIGURA 6.18 - Sistema TROPIC com os parâmetros certos para a geração do processo de síntese do arquivo <i>c:\tentosus\adde.sim</i>	74
FIGURA 6.19 - Sistema TROPIC - extração das portas lógicas do circuito <i>c:\tentosus\adder.sim</i>	75
FIGURA 6.20 - Sistema TROPIC - posicionamento do circuito <i>c:\tentosus\adder.sim</i>	75
FIGURA 6.21 - Sistema TROPIC - mensagens apresentadas no momento da síntese ...	76
FIGURA 6.22 - Sistema TROPIC - mensagem apresentada no momento da síntese	76
FIGURA 6.23 - Edição simbólica do arquivo gerado	77
FIGURA 6.24 - Compactação do circuito <i>c:\tentosus\adder.lds</i>	77

Lista de Tabelas

TABELA 2.1 - Aplicação de cada ferramenta	18
TABELA 2.2 - Aplicação de cada ferramenta	19
TABELA 4.1 - Características visuais da interface do Gerenciador do TENTOS	26
TABELA 4.2 - Lista dos Arquivos de Configuração necessários à execução do TENTOS.....	27
TABELA 4.3 - Lista dos elementos da janela de Instalação/Configuração das ferramentas.....	29
TABELA 4.4 - Lista das variáveis que podem ser usadas como parâmetros	30
TABELA 4.5 - Elementos da janela de Instalação/Configuração dos Arquivos de Tecnologia.....	32
TABELA 4.6 - Elementos da janela de Instalação/Configuração dos Tipos de Extensões.....	33
TABELA 4.7 - Itens da caixa de diálogo - Projeto (Informações do Projeto)	35
TABELA 4.8 - Lista dos diretórios manipulados pelo Sistema TENTOS.	36
TABELA 4.9 - Lista dos elementos pertencentes a Janela de Parâmetros.....	52

Lista de Abreviaturas

CI	circuito integrado
CIF	Formato de arquivo para descrição geométrica de circuitos integrados (Caltech Intermediate Format)
CIR	Formato de arquivo na descrição spice
CMOS	Complementary Metal Oxide Silicon
CPGCC	Curso de Pós-Graduação em Ciência da Computação
EDA	Electronic Design Automation
ESQ	Formato de arquivo para manipulação no editor de esquemático
GME	Grupo de Microeletrônica
PAC	Projeto assistido por computador
PC	Personal Computers, computadores pessoais
RS	Formato de arquivo para descrição geométrica de circuitos integrados

Resumo

Este trabalho apresenta um gerenciador de ferramentas para projeto de circuitos integrados, o sistema TENTOS, agora desenvolvido para o ambiente MS-WINDOWS™. O ambiente TENTOS é um sistema aberto, isto é, permite a fácil inclusão de novas ferramentas em tempo de execução do gerenciador, tornando-o desta forma sempre atual em relação às ferramentas existentes.

Inicialmente será feita uma breve descrição de alguns dos gerenciadores existentes, tanto os desenvolvidos com finalidades comerciais como os do meio acadêmico. Em seguida, será apresentado um histórico sobre a evolução do sistema TENTOS, da sua versão inicial até a sua versão atual.

Em uma etapa seguinte será descrito o estado atual do sistema TENTOS, isto é, suas características principais; a estrutura dos menus; os arquivos de configuração do sistema; como incluir novas ferramentas, arquivos de tecnologia; a configuração standard do sistema, quais ferramentas acompanham o TENTOS; como funciona a execução das ferramentas. Concluída a apresentação do sistema TENTOS, serão apresentados exemplos que ilustram as etapas de desenvolvimento de um projeto de circuito integrado utilizando o sistema TENTOS.

PALAVRAS-CHAVE: AMBIENTE INTEGRADO DE PROJETO, CIRCUITO INTEGRADO, FERRAMENTAS PARA PROJETOS ASSISTIDOS POR COMPUTADORES(PAC), MICROELETRÔNICA.

Title: “The TENTOS Systems for Windows - A Tools Manager for Microelectronic”**Abstract**

This dissertation presents a tool manager for integrated circuit design, the TENTOS system, now developed for the MS-WINDOWS™ environment. The TENTOS package is an open system, that allows an easy inclusion of new tools in the execution time of the manager, allowing an easy and constant updating of tools that are integrated into the package.

Firstly, a short description of existing frameworks will be shown by including commercial and academics systems. Secondly, a brief historical of TENTOS evolution system will be presented.

Following that description the present state of the TENTOS system will be described which comprises: its main characteristics; the structure of menus; system configuration files; how to include new tools and technology files; the standard system configuration, which tools are available into the TENTOS and how they are executed.

Finally some examples on how to use the TENTOS system will be shown.

KEY WORDS: FRAMEWORK, INTEGRATED CIRCUIT, COMPUTER AIDED DESIGN (CAD) TOOL, MICROELECTRONICS.

1 Introdução

1.1 Motivação

Com a evolução dos computadores (aumento da velocidade de processamento, aumento do tamanho da memória principal, discos rígidos cada vez maiores, melhor resolução de vídeo, recursos multimídia) surgem cada vez mais novos sistemas operacionais ou novas versões destes, que buscam explorar da melhor forma possível todos estes avanços. Esta relação acaba gerando um processo contínuo de que a cada novo equipamento disponibilizado pelas indústrias, novos sistemas estarão sendo projetados para tirar o máximo de proveito dos novos recursos oferecidos. A figura 1.1 ilustra esta relação existente entre equipamentos e sistemas.

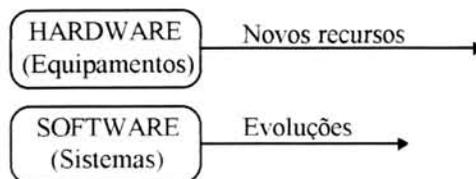


FIGURA 1.1 - Processo evolutivo hardware e software

A crescente exigência de informatização e automatização de tarefas humanas e/ou repetitivas leva ao desenvolvimento de ferramentas específicas onde um variado número de recursos/informações são incorporados ao processo, propiciando, desta forma, uma complementação aos conhecimentos e recursos técnicos dos usuários. Isto tudo faz com que processos que anteriormente requeriam uma demanda extra de mão-de-obra e de tempo sejam suprimidas.

Na microeletrônica não é diferente. Com o surgimento de novos equipamentos e sistemas, o trabalho do projetista foi simplificado. Por exemplo: através da evolução das ferramentas de síntese obtém-se circuitos cada vez mais complexos (com um maior número de transistores, níveis de metal, tecnologia submicrônica); os editores gráficos de leiaute permitem a fácil correção de medidas incorretas ou a adaptação de um leiaute a uma nova tecnologia.

Processos já informatizados são melhorados, a fim de poder explorar ao máximo os novos recursos computacionais que surgem. E é dentro desta filosofia que o sistema TENTOS se insere, ou seja, o aproveitamento de novos recursos computacionais para aprimorar o sistema de PAC, tais como:

- Novos sistemas operacionais (MS-WINDOWS™,...);
- Novas ferramentas de programação (Visual Basic™, Visual C™,...);
- Novas técnicas de programação (programação orientada a objetos);
- Novos conceitos de gerenciadores de projeto para microeletrônica (incorporação de ferramentas sem alteração no código fonte do gerenciador).

Com o advento destes novos recursos surgiu a necessidade de se adaptar o sistema TENTOS a todas estas novas tecnologias, pois as máquinas tipo IBM-PC estão se tornando verdadeiras estações de trabalho, com processadores cada vez mais poderosos, por um preço muito baixo se comparado com as estações de trabalho atuais.

Se não bastassem os motivos enumerados acima, existem outros que já justificariam um investimento no desenvolvimento desta nova versão do sistema TENTOS, a saber:

- Maior difusão da ferramenta, pois os laboratórios das universidades do Brasil e da América Latina são na sua maioria compostos por computadores IBM-PC compatíveis. Convém salientar que o TENTOS, na sua versão MS-DOS™, encontra-se em uso em vários centros de pesquisa Latino-Americanos;
- Fornecer um pacote de ferramentas para microeletrônica de baixo custo;
- O MS-WINDOWS™ aproximou ainda mais os PCs das estações de trabalho, possibilitando por exemplo, a execução de processos em paralelo sem perda de performance, que com ferramentas MS-DOS™, se não era impossível era de difícil implementação.

O sistema TENTOS, objeto desta dissertação, foi projetado para ser um ambiente eficiente, auxiliando o projetista de ferramentas a desenvolver programas que poderão ser facilmente integrados ao sistema, bem como fornecendo aos projetistas de circuitos integrados um ambiente no qual eles poderão desenvolver da melhor forma o projeto de CIs, explorando ao máximo os recursos oferecidos por estas novas tecnologias.

1.2 Objetivos

O Grupo de Microeletrônica da UFRGS visa, entre outros objetivos, pesquisar e desenvolver ferramentas de *software* que auxiliem o projeto de circuitos integrados.

Seguindo os motivos apresentados, tais como o de ter uma ferramenta de baixo custo para o desenvolvimento de projetos para microeletrônica, em computadores tipo IBM-PC, com ambiente MS-WINDOWS™, decidiu-se pela realização de uma nova versão do sistema TENTOS para MS-WINDOWS™. O principal objetivo deste trabalho foi a criação de um ambiente de software para o gerenciamento de ferramentas voltadas ao projeto de circuitos integrados em computadores tipo IBM-PC, utilizando-se como plataforma de trabalho o MS-WINDOWS™. Este sistema possui características, que foram apresentadas com maiores detalhes em [MÄH 94] e [MÄH 95]. O novo ambiente deve atender os seguintes objetivos:

- Incorporar ao TENTOS características de *framework* [WAG 94], isto é, permitir a agregação de novas ferramentas e arquivos de tecnologia, pelo próprio usuário, durante o uso do sistema [WIR 94];
- Maior interação com o usuário, possibilitando interferência do projetista em cada etapa de execução (principalmente nas ferramentas de síntese);
- Configuração visual, definindo os dados em caixas de diálogos.

As características apresentadas acima estão vinculadas diretamente ao gerenciador TENTOS. Muitos outros objetivos, em termos de sistema, também foram executados, são eles:

- Oferecer um pacote de ferramentas voltado ao ensino, eficiente e bem documentado, que atendessem às várias fases do projeto de um circuito integrado;
- Implementar exemplos de execução do gerenciador e das ferramentas através de tutoriais, visando uma maior divulgação e aprendizado da microeletrônica nos meios acadêmicos, útil para a realização de circuitos integrados de pequeno/médio porte;
- Construir um manual do usuário *on-line*;

Com o objetivo de ampliar o sistema, algumas outras tarefas foram executadas, tais como:

- A inclusão de uma nova ferramenta de síntese, com novos conceitos de síntese automática, o TROPIC [MOR 94], ferramenta que estava operacional para o ambiente UNIX. Este software teve seus fontes adaptados para que funcionasse em ambiente MS-WINDOWS™;
- A inclusão de uma ferramenta para a comparação de *netlists*, o COTONET [FRE 92], ferramenta que também estava operante em ambiente UNIX. Assim como o sistema TROPIC, os seus fontes foram adaptados para que funcionasse em ambiente MS-WINDOWS™;
- A inclusão da nova versão do extrator lógico, EXTRALO, parte integrante do Sistema TROPIC, que pode ser usado separadamente do TROPIC;
- A inclusão do editor simbólico SIMBY [SOT 93];
- A inclusão do compactador de leiautes SILC [SOT 94];

Como o sistema TENTOS é uma ferramenta de livre distribuição para universidades brasileiras e latino-americanas também foi objeto de trabalho desta dissertação:

- Um programa para realizar a configuração de dados no momento em que um novo usuário for utilizar o TENTOS;
- A criação de um programa para a instalação do TENTOS no disco rígido do usuário, semelhante aos programas de instalação dos produtos Microsoft™;
- A criação de um arquivo com informações referentes ao grupo de microeletrônica da UFRGS. Neste arquivo estariam informações referentes aos integrantes do GME, como por exemplo: suas áreas de interesse, projetos desenvolvidos, e-mail;
- Múltiplas linguagens (Português, Inglês, Espanhol e Francês) na apresentação da interface do Gerenciador com o usuário.

2 Gerenciadores de Projeto: uma Revisão

Neste capítulo são citados alguns dos sistemas para projetos de circuitos integrados disponíveis no mercado. Estes ambientes são comerciais ou acadêmicos, sendo a apresentação destes feita através dos seguintes enfoques:

- *Apresentação das ferramentas*: qual empresa ou universidade é a responsável pelo desenvolvimento dos sistemas e em qual plataforma de trabalho deve ser utilizada a ferramenta;
- *Aplicação das ferramentas*: a que etapa do projeto de circuitos integrados se destina;
- *Integração de novas ferramentas*: forma de integração de outras ferramentas.

2.1 Ambientes Convencionais de Projeto - Conceito

De um modo geral, considera-se ambiente convencional de projetos aquele onde as ferramentas de diferentes fabricantes, são chamadas a partir de um gerenciador de trabalho comum. Este padrão de ambiente apresenta uma série de inconveniências que retardam a conclusão do projeto, são elas [WAG 94]:

- *Falta de um modelo uniforme de representação de dados*: Tendo em vista o fato de existirem diversos fabricantes de ferramentas para projetos de circuitos integrados, identifica-se a falta de um modelo (definição dos dados) e/ou formato (estrutura de dados do modelo) comum, uniforme, na representação dos dados;
- *Inconsistência de dados*: A existência de diversas alternativas ou vistas de um mesmo objeto, manipuladas por diversas ferramentas, pode trazer problemas de inconsistência de dados (certos dados comuns a diferentes alternativas ou vistas podem estar armazenados de forma redundante em mais de um arquivo e/ou também existe a possibilidade de existir relações entre dados armazenados em diferentes arquivos);
- *Interface-usuário não homogênea*: Cada ferramenta é construída com uma linguagem de interação própria;
- *Dificuldade na integração de novas ferramentas*: Para cada nova ferramenta incluída no ambiente pode ser necessária a construção de novos conversores de formato;
- *Ausência de recursos para gerência de dados*: Os dados de projeto estão armazenados em um grande número de arquivos, sob controle exclusivo do usuário;
- *Ausência de recursos para gerência do processo de projeto*: Os usuários podem ativar as ferramentas em qualquer ordem desejada.

2.2 Ambiente Integrados e Abertos - Conceito

Um ambiente é integrado quando as ferramentas atuam de forma harmoniosa de modo a evitar problema de inconsistência, de falta de homogeneidade e de falta de controle sobre os dados e sobre o projeto.

Os problemas indicados na seção anterior são resolvidos pela construção de ambientes integrados e abertos. A seguir serão apresentadas as principais funcionalidades destes ambientes [WAG 94]:

- *Consistência de dados*: Os dados de projeto manipulados pelas ferramentas seguem um modelo conceitual uniforme;
- *Representação interna única*: Para uma dada representação num ponto do processo de projeto, pode ser opcionalmente adotada uma representação interna única para todas as ferramentas, evitando-se a necessidade de conversores de dados;
- *Interface-usuário homogênea*: Além da uniformidade léxica e sintática, as ferramentas apresentam uma interface-usuário semanticamente uniforme;
- *Recursos de gerência de dados*: O ambiente oferece recursos para gerência de versões e configurações de objetos de projeto;
- *Recursos de gerência do processo de projeto*: O ambiente oferece recursos para controle do seqüenciamento de tarefas do projeto;
- *Trabalho num ambiente distribuído*: Os dados armazenados em um servidor devem estar disponíveis em todos os nodos da rede, sendo feito o controle dos acessos concorrentes a eles;

2.3 Frameworks - Conceito

Para EDA, *frameworks* são plataformas para a construção de ambientes integrados e abertos de projetos de sistemas eletrônicos. Um framework não é um ambiente em particular para uma determinada aplicação. Ele compreende um conjunto de serviços de propósitos gerais que podem ser configurados para uma grande diversificação de aplicações (trata-se de um determinado conjunto de ferramentas que suporta um dado processo de projeto). Um framework tem que ser independente das tecnologias e processos de projeto bem como da fabricação de sistemas e circuitos eletrônicos, de modo a adequar-se a qualquer conjunto de ferramentas de projeto.

Um framework permite aos construtores de ambientes de projeto:

- integrar novas ferramentas ao ambiente;
- definir processos e metodologias de projeto; e
- definir políticas de uso de ferramentas e dados.

Um framework permite aos usuários finais dos ambientes:

- disparar e gerenciar a execução das ferramentas;
- criar, organizar e gerenciar dados;
- visualizar (graficamente) todo o processo de projeto; e
- realizar tarefas diversas de gerência de projeto, inclusive relativas ao trabalho em equipe.

2.4 Apresentação das Ferramentas

A análise que irá ser apresentada a seguir foi elaborada a partir dos seguintes sistemas:

- *Cadence Opus: Framework* desenvolvido pela empresa norte-americana Cadence Design Systems, Califórnia. Trata-se de um sistema para ser executado em estações de trabalho compatíveis com o padrão UNIX, tais como Sun, DEC, HP, Apollo, Intergraph, NEC e Sony.
- *Mentor Graphics*: Este *framework* foi desenvolvido pela empresa norte-americana Mentor Graphics Corporation, Oregon. Trata-se de uma ferramenta

desenvolvida para ser executada em estações de trabalho compatíveis com o padrão UNIX, tais como DEC, Sun, HP, IBM, NEC, Solbourne e Sony.

- *Compass*: Trata-se de um framework desenvolvido pela empresa norte-americana Compass Design Automation, Califórnia. Trata-se de uma ferramenta desenvolvida para ser executada em estações de trabalho compatíveis com o padrão UNIX.
- *Synopsys*: [<http://www.synopsys.com/>] Framework desenvolvido pela empresa norte-americana Synopsys, Califórnia. Trata-se de uma ferramenta desenvolvida para ser executada em estações de trabalho compatíveis com o padrão UNIX.
- *Alliance*: [GRE 92] Sistema desenvolvido pelo MASI-Laboratório da Universidade de Pierre et Marie Curie (UPMC) de Paris-França. Trata-se de um conjunto de ferramentas desenvolvidas para serem utilizadas em estações de trabalho compatíveis com o padrão UNIX, tais como estações Sun e DEC.
- *Ocean*: [GRO 93] [PIL 95] Sistema desenvolvido pela universidade holandesa Delft University of Technology - Electrical Engineering. Este sistema possui versões tanto para estações de trabalho, SUN ou HP por exemplo, como para computadores IBM-PC compatíveis.
- *Octtools*: [CAS 91] Conjunto de ferramentas que foram desenvolvidas pelo Electronics Research Laboratory da Universidade da Califórnia, Berkeley - EUA. Trata-se de um sistema desenvolvido para ser utilizado em estações de trabalho compatíveis com o padrão UNIX
- *Tedmos*: [OLI 94] Ferramenta de projeto de circuitos integrados desenvolvida pela Universidade Federal do Rio de Janeiro. O TEDMOS foi desenvolvido para ser executado em computadores tipo IBM-PC com MS-WINDOWS™.

2.5 Aplicação das Ferramentas

Dentre os sistemas de CAD comerciais em questão, pode-se citar que, de um modo geral, os sistemas Mentor Graphics, Cadence, Compass e Synopsys são destinados a todas as etapas do projeto, isto é, possuem ferramentas que manipulam o circuito tanto a nível comportamental, estrutural como a nível físico. No que se refere a sistemas acadêmicos, o Alliance é o único que engloba ferramentas destinadas a todas as etapas do projeto. Já as ferramentas Tedmos, Ocean e Octools são ferramentas voltadas para a síntese do circuito a partir de uma descrição estrutural.

A seguir será apresentada uma tabela (tabela 2.1) onde será apresentado um quadro resumido da finalidade de cada sistema de CAD analisado, observando os seguintes aspectos: aplicação (a que etapa do projeto se destina o sistema) e um campo de observações, onde o pacote será classificado em framework (a integração das ferramentas é feita através de uma base de dados única), ambiente (quando a transferência dos dados entre as ferramentas é feita através de arquivos intermediários) ou conjunto de ferramentas (existe o conceito de base de dados, porém não há a integração das ferramentas, isto é, não existe um software para gerenciar o sistema).

TABELA 2.1 - Aplicação de cada ferramenta

Ferramenta	Aplicação	Observação
Cadence	Engloba todas as etapas de projeto de circuitos integrados	Framework
Mentor	Sistema completo, integrado por ferramentas	Framework

Graphics	que abrangem todas as operações relacionadas aos diferentes níveis de descrição do circuito.	
Synopsys	Direcionado para a síntese lógica a partir de VHDL. O resultado da síntese será uma descrição estrutural, utilizada por outro sistema, como por exemplo, Compass.	Framework
Compass	Integrado por ferramentas que trabalham em todos os níveis de descrição do circuito.	Framework
Alliance	Conjunto de ferramentas que abrange todas as etapas de projeto, desde a descrição comportamental, até a física.	Conjunto de Ferramentas
Ocean	Ambiente integrado por ferramentas que agem entre as etapas de descrição estrutural e descrição física.	Ambiente
Octtools	Voltada para aplicações a nível de descrição estrutural e física.	Ambiente
Tedmos	Inclui ferramentas para a síntese do circuito a partir de uma descrição estrutural.	Ambiente

2.6 Integração de novas ferramentas

Dentre os 4 (quatro) sistemas comerciais aqui observadas, Cadence, Mentor Graphics, Compass e Synopsys, duas estão abertas à inclusão de outras ferramentas, Cadence e Mentor Graphics. A forma de agregar ferramentas ao Cadence, por exemplo, é através de um linguagem de programação, desenvolvida pela própria Cadence, o SKILL. Com esta linguagem, o projetista tem acesso à base de dados do sistema, podendo manipular objetos (portas lógicas, fios, polígonos, etc.) e assim integrar novas ferramentas. Ainda é possível com SKILL acrescentar, remover ou alterar os menus e janelas de passagem de parâmetros internas do próprio framework, possibilitando ao usuário ter o sistema Cadence configurado da melhor forma que lhe convir.

Dentre as ferramentas acadêmicas, aqui citadas, a ferramenta Octools, desenvolvida em Berkeley, apresenta o recurso de agregar novas ferramentas ao ambiente através da utilização de rotinas em C, fornecidas com o sistema, que permitem acessar, e acrescentar informações à base de dados do sistema. Na tabela 2.2 é apresentado um quadro resumido de como integrar ou não ferramentas aos sistemas analisados.

TABELA 2.2 - Aplicação de cada ferramenta

Ferramenta	Observação
Cadence	Permite a inclusão de novas ferramentas através de especificações feitas com a linguagem de programação SKILL.
Mentor Graphics	Permite a inclusão de novas ferramentas à base de dados FALCON através do sistema OPENDOOR.
Synopsys	Sistema fechado.
Compass	Sistema fechado.
Alliance	Conjunto de ferramentas, não possui um gerenciador de ferramentas.
Ocean	Sistema fechado.

Octtools	Permite agregar novas ferramentas ao ambiente através da utilização de rotinas em C, fornecidas com o sistema.
Tedmos-DOS	Sistema fechado.
Tedmos-Windows	Permite agregar novas ferramentas no estilo Windows, permitindo comunicação via DDE, DLLs e futuramente OLE. No momento atual ainda não existe um gerenciador com facilidades para incluir e excluir ferramentas.

2.7 Conclusão

As ferramentas desenvolvidas com finalidade comercial, aqui mencionadas, representam, quase que a totalidade do mercado mundial de software para projeto de circuitos integrados. Todas estas ferramentas encontram-se desenvolvidas para ambiente de estações de trabalho padrão UNIX, porque são sistemas que requerem muita memória RAM, processadores extremamente rápidos e um sistema operacional adequado aos diversos tipos de operações que devem ser executados, tais como, processamento paralelo de dois ou mais programas, tudo isso são características que limitam o seu uso em computadores tipo IBM-PC.

Já as ferramentas acadêmicas, por não visarem atender a projetos muito complexos e também porque são ferramentas não tão *pesadas*, possuem, em alguns casos, versões para computadores tipo IBM-PC.

3 Histórico do Sistema TENTOS

Dentre as atividades do Grupo de Microeletrônica (GME) do Instituto de Informática da UFRGS está a criação de ferramentas CAD para o projeto de circuitos integrados (CIs). Como produto de vários anos de pesquisa e desenvolvimento, o grupo possui uma quantidade razoável de ferramentas que atendem às várias fases do projeto físico, desde a captura esquemática, até o leiaute em nível de máscara.

Dada a quantidade de ferramentas, bem como a diversidade de formatos exigidos pelas mesmas, decidiu-se partir para uma etapa de integração, na qual foi definido um conjunto de recursos de software que permitisse tanto o projeto *full custom*, como a síntese automática. O resultado desta etapa deveria solucionar as principais dificuldades que os novos projetistas encontravam, devido ao uso independente das ferramentas, como por exemplo:

- que programa utilizar em cada etapa do projeto;
- onde encontrar tais programas;
- quais arquivos de tecnologia devem ser utilizados, e onde encontrá-los;
- que parâmetros e informações são necessários em cada etapa;
- quais formatos e extensões padrão dos arquivos;
- qual a versão mais recente de um programa.

A integração propriamente dita foi realizada de forma a permitir que novos programas pudessem ser integrados ao pacote, no qual o conceito chave foi AMBIENTE INTEGRADO DE PROJETOS (semelhante ao encontrado na interface do Macintosh™). O pacote que integra as diversas ferramentas, eliminando as dificuldades citadas acima, foi chamado TENTOS [MOR 91] [MOR 91a], tendo sido inicialmente desenvolvido para o ambiente MS-DOS™ em computadores PC compatíveis.

3.1 A Primeira versão

A primeira versão do TENTOS, [MOR 91] [MOR 91a], foi desenvolvida para o ambiente MS-DOS™, implementado em máquinas tipo IBM-PC. A sua característica principal era uma interface única com o usuário, cujas principais funções eram:

- execução direta das ferramentas;
- gerenciar os diretórios;
- controlar o formato dos arquivos.

Com isto o projetista de CIs vê o conjunto de ferramentas de CAD como um sistema integrado único, onde todos os aplicativos são acessados através de comandos internos ao sistema, ficando toda a comunicação entre as ferramentas transparente ao usuário.

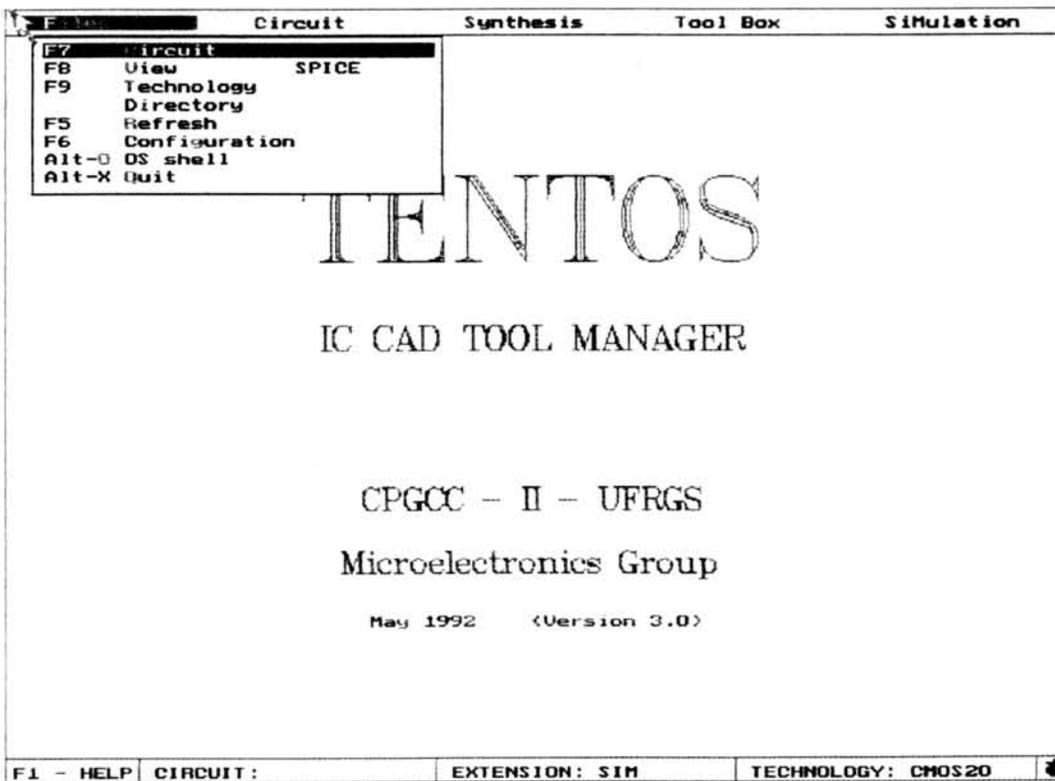


FIGURA 3.1 - Interface do TENTOS para DOS.

A interface gráfica é bastante amigável (figura 3.1). O usuário não necessita saber onde se encontram as ferramentas, os arquivos de tecnologia e, principalmente, como acessar os programas. Basta o usuário executar o TENTOS e seguir os passos de projeto através dos menus existentes na interface do sistema, no qual é possível escolher o circuito de trabalho, definir formatos e executar as ferramentas necessárias de acordo com o desenvolvimento do projeto. A interface permite também que todos estes dados sejam monitorados pelo usuário, mostrando-os na linha inferior da tela.

Na figura 3.2 é apresentada a disposição das ferramentas nos menus.

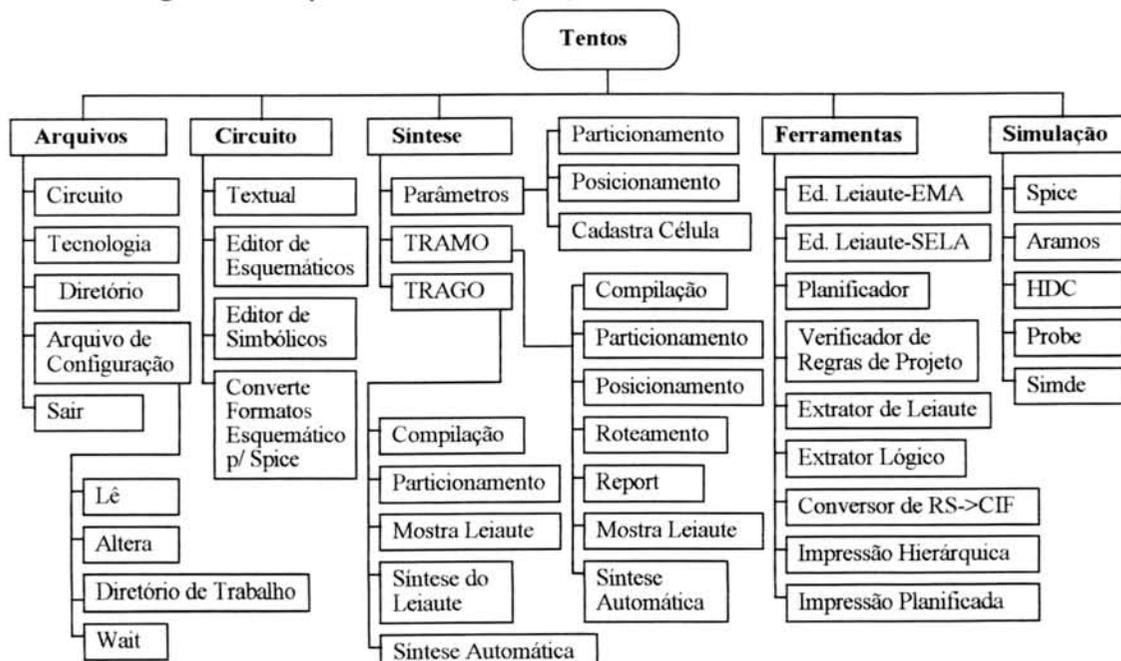


FIGURA 3.2 - Estrutura do sistema TENTOS

Cada um dos submenus (Arquivos, Circuito, Síntese, Ferramentas e Simulação) engloba um determinado conjunto de ferramentas, com características próximas entre si.

- O Submenu **Arquivos** controla as operações relacionadas a arquivos, tais como: selecionar o nome do circuito, o arquivo de tecnologia, os diretórios de trabalho e o arquivo de configuração;
- O Submenu **Circuito** engloba as ferramentas que permitem criar as descrições de circuitos no nível lógico e estrutural (*netlist*), são elas: Editor Textual, Editor de Esquemáticos, Editor de Símbolos e um Conversor de arquivos do formato de esquemático (ESQ) para o formato *Spice* (SIM, CIR);
- O Submenu **Síntese** é integrado por ferramentas para síntese automática de leiautes que utilizam a metodologia TRANCA (TRAMO e TRAGO);
- O Submenu **Ferramentas** inclui uma miscelânea de ferramentas de apoio ao projeto físico, tais como: Editor de Leiautes - EMA e SELA, Planificador, Verificador de Regras de Projeto, Extrator de Leiautes, Extrator Lógico, Conversor de arquivos do formato RS para formato CIF, Impressor Hierárquico e Impressor Planificado;
- Já no Submenu **Simulação** estão as ferramentas relacionadas com a simulação do circuito, são elas: Spice, Aramos, HDC, Probe e Simde.

Convém salientar que esta versão do Sistema TENTOS, em MS-DOS™, encontra-se em uso em vários centros de pesquisa Latino-Americanos. Entre estes citamos o caso do Grupo de Microeletrônica da Escola Politécnica do Equador. Conforme [MON 92] o grupo desenvolveu dois circuitos dentro do Projeto Multiusuário Iberoamericano (PMU), utilizando o TENTOS. O primeiro circuito foi um *Médio Somador*, enviado para a fabricação em dezembro de 1991. Os testes sobre os protótipos, que retornaram em maio de 1992, mostraram que os circuitos eram funcionalmente corretos. O segundo circuito implementado foi um codificador/decodificador de linha programável cujos protótipos ainda não haviam retornado quando da publicação do artigo citado. Outro trabalho que merece referência é o projeto de um circuito integrado para controle de motores de passo desenvolvido pelo Grupo de Microeletrônica da PUC do Peru. Este trabalho recebeu o primeiro prêmio no XI Concurso de Engenharia Mecânica, Elétrica e Ramos Afins (XI CONIMERA) em abril de 1993.

3.2 A Primeira Versão para ambiente MS-WINDOWS™

Como o ambiente MS-WINDOWS™ torna-se cada vez mais um padrão para computadores PC compatíveis, tornou-se interessante oferecer uma versão do TENTOS para MS-WINDOWS™. Nesta primeira fase da adaptação, apenas a interface foi modificada (ver figura 3.3), permanecendo o mesmo conjunto de ferramentas existentes na versão MS-DOS™.

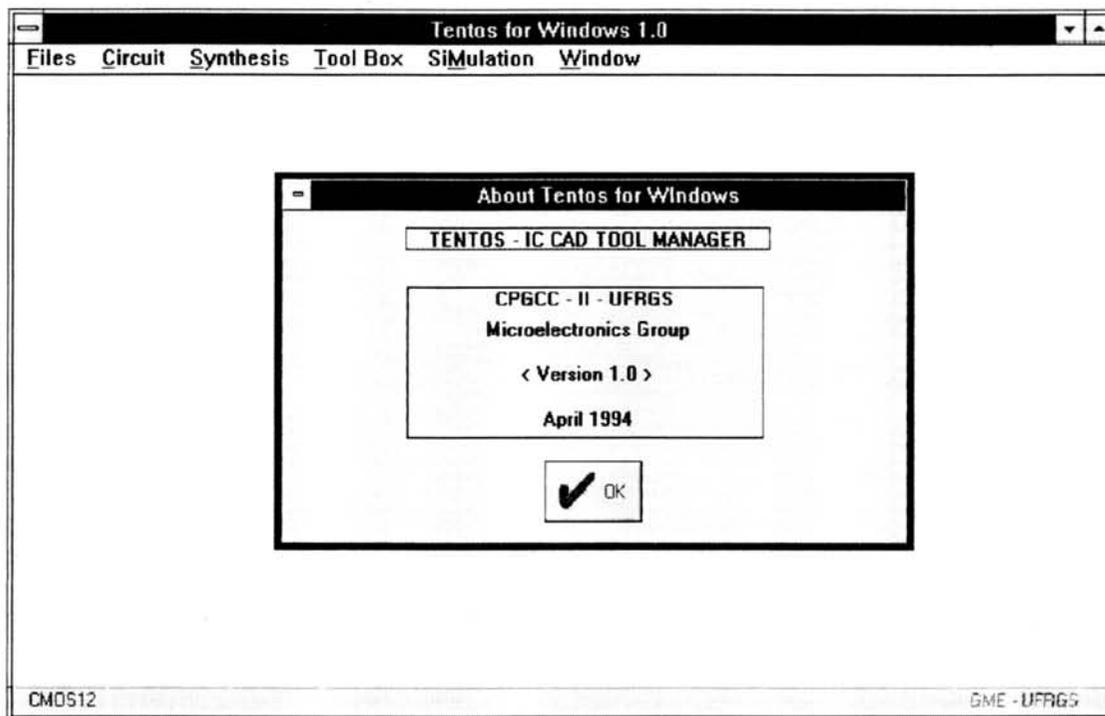


FIGURA 3.3 - Interface do TENTOS para MS-WINDOWS™.

A interface da primeira versão para MS-WINDOWS™ (figura 3.3) é muito similar à versão MS-DOS™, já que a disposição dos menus foi organizada da mesma forma, mas com novos recursos, tais como:

- maior facilidade na escolha do circuito que será utilizado, sendo que o usuário não necessita mais associar a extensão do circuito ao tipo;
- fácil adaptação para novos usuários, uma vez que a ferramenta segue um padrão MS-WINDOWS™ de interface.
- multitarefa, recurso este obtido através do próprio MS-WINDOWS™;
- uso de memória alta, gerenciado pelo MS-WINDOWS™;
- simplificação do procedimento de integração de novas ferramentas.

Algumas dificuldades foram encontradas, haja visto que uma nova forma de abordagem do sistema (Programação Orientada a Objetos) implicou na reescrita de todo o programa gerenciador. Salienta-se que a filosofia de programação se altera completamente. Outra alteração efetuada foi relativa à ferramenta utilizada para a implementação do TENTOS. Na versão para MS-DOS™ foi utilizada a ferramenta GIM [BAG 89], desenvolvida na UFRGS, e na versão para MS-WINDOWS™ foi utilizado o Visual Basic, ferramenta desenvolvida pela Microsoft. Outro fato importante é que as ferramentas acessadas pelo TENTOS não foram alteradas e continuaram na versão antiga para MS-DOS™, ocasionando, assim, chamadas a arquivos com extensão ".PIF" (um formato de arquivo que deve ser criado para que possam ser executadas, em ambiente MS-WINDOWS™, tarefas previstas para ambiente MS-DOS™). Estas chamadas a arquivos ".PIF" causam grandes transtornos ao MS-WINDOWS™, que trabalha de forma compartilhada; contudo não era objetivo inicial alterar tais ferramentas.

Assim como na versão para MS-DOS™ a característica principal do TENTOS para WINDOWS é uma interface única com o usuário. Desta forma, o usuário vê o sistema de forma integrada.

Como já foi visto na figura 3.3, na linha superior da janela, abaixo da linha de título, é apresentado o menu Principal que contém as chamadas para os submenus Files, Circuit, Synthesis, Tool Box, Simulation e Window. Esta primeira versão do Sistema TENTOS apresenta a mesma disposição das ferramentas nos menus que a versão para MS-DOS™, ver figura 3.2.

3.3 Conclusão

Neste capítulo foram apresentadas as versões iniciais do Sistema TENTOS, a primeira para MS-DOS™ e a segunda para MS-WINDOWS™. Visando suprir a deficiência maior destas versões iniciais do sistema, que foi a simplificação no momento da integração de novas ferramentas ao sistema, será apresentado no capítulo seguinte o estado atual do Sistema TENTOS.

4 O Sistema TENTOS *for Windows*

A figura 4.1 apresenta a atual interface do sistema TENTOS para MS-WINDOWS™.

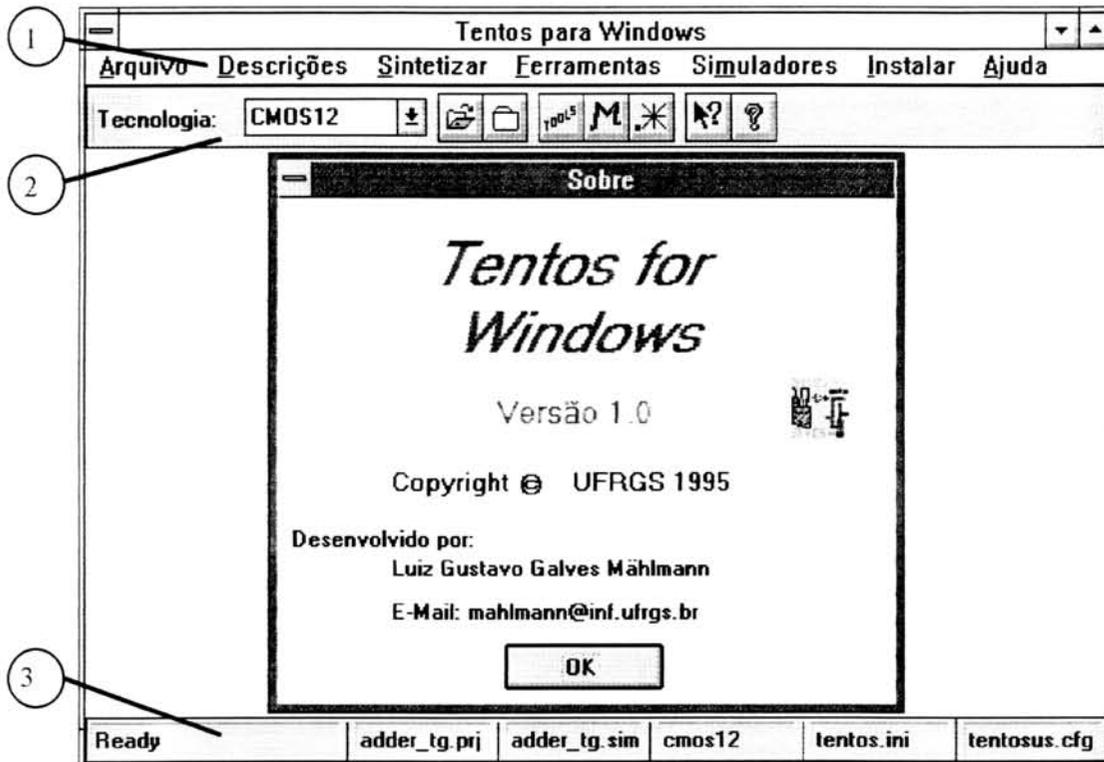


FIGURA 4.1 - A Nova Interface do Sistema TENTOS for Windows

A tabela 4.1 apresenta a descrição dos itens ilustrados na figura 4.1.

TABELA 4.1 - Características visuais da interface do Gerenciador do TENTOS

Itens	Descrição
1	<i>Menu:</i> Caminho para todas as opções presentes no Sistema TENTOS.
2	<i>Barra de Ferramentas:</i> Botões de atalho para as tarefas mais usuais no TENTOS.
3	<i>Barra de Status:</i> Apresenta informações referentes a configuração atual do Sistema.

Para maiores detalhes a respeito do funcionamento do sistema TENTOS, deve-se consultar [MÄH 96].

4.1 As Características do Sistema

Dentre as principais características do sistema TENTOS deve-se citar as seguintes:

- Configuração do ambiente de trabalho independente para cada usuário do Sistema;
- A fácil inclusão de novas ferramentas no ambiente;
- A fácil inclusão de novos arquivos de tecnologias no sistema TENTOS;
- A inclusão de novas descrições de circuitos, caracterizadas pela extensão do nome do arquivo que descreve o circuito ;

- Toda a interação com o usuário é feita através de caixas de diálogos, propiciando desta forma um uso sem erros da ferramenta;
- O Sistema pode apresentar as mensagens da sua interface em quatro idiomas (português, inglês, espanhol e francês) através de uma simples seleção do idioma via menu;
- Sistema com manual do usuário *on-line* [MÄH 96];
- Exemplos de aplicação *on-line* (exemplos apresentados no capítulo 5);
- A troca de informação entre ferramentas é feita através de arquivos gerados pelas mesmas.

4.1.1 A independência da configuração do Sistema

Por se tratar de um gerenciador de programas para microeletrônica, buscou-se dotar o sistema TENTOS de uma completa independência dos dados e programas usados por cada usuário da ferramenta. Desta forma cada projetista teria a sua configuração do sistema, com as suas ferramentas, os seus diretórios de trabalho e os seus dados.

Esta independência da configuração foi obtida através do uso de arquivos de configuração para o sistema. São dois os arquivos que guardam as configurações dos usuários. O primeiro é o arquivo de inicialização (*.INI*) do sistema TENTOS, o qual armazena os dados referentes ao ambiente TENTOS, e o segundo é o arquivo de configuração (*.CFG*) que possui os dados de configuração para algumas das ferramentas incorporadas ao TENTOS, tais como a ferramenta de síntese TRAGO.

Juntando-se a estes dois arquivos já citados, que podem ser configurados pelo usuário, existe um terceiro arquivo de dados básicos do sistema, *TENTOS.DAT*, com nome fixo.

A tabela 4.2 apresenta a lista dos tipos de arquivos de configuração necessários à execução do Sistema TENTOS.

TABELA 4.2 - Lista dos Arquivos de Configuração necessários à execução do TENTOS

Tipo do Arquivo	Finalidade (conteúdo)
Dados Básicos TENTOS.DAT	<ul style="list-style-type: none"> • Nome do arquivo de inicialização em uso; • Editor utilizado para alterar o arquivo de configuração (<i>.CFG</i>); • Posicionamento das janelas de diálogo utilizadas pelo Gerenciador.
Inicialização *.INI	<ul style="list-style-type: none"> • Ferramentas incorporadas aos menus; • Arquivos de tecnologia incorporados ao menu; • Formatos possíveis de circuito; • Diretórios configurados pelo projetista; • Nome do projeto em uso; • Nome do circuito em uso; • Nome do arquivo de tecnologia em uso; • Nome do arquivo de configuração (<i>.CFG</i>) em uso; • Se deve ser mostrada ou não a barra de ferramentas; • Se deve ser mostrada ou não a barra de status; • Se deve ser mostrada ou não a janela de parâmetros; • O diretório onde foi selecionado o circuito em uso;

	<ul style="list-style-type: none"> • O diretório onde foi selecionado o arquivo de inicialização em uso; • O diretório onde foi selecionado o arquivo de configuração em uso.
Configuração *.CFG	<ul style="list-style-type: none"> • Informações pertinentes às ferramentas que acompanham a versão <i>standard</i> do sistema TENTOS.

O Anexo 1 apresenta o conteúdo do arquivo TENTOS.DAT que acompanha a versão *standard* do TENTOS. Já o Anexo 2 mostra as informações que estão incluídas no arquivo de inicialização (TENTOS.INI) que é gerado no momento de instalação do TENTOS. O Anexo 3 lista o arquivo de configuração (TENTOSUS.CFG) que é criado também no momento de instalação do Gerenciador.

4.1.2 Associando novas Ferramentas ao TENTOS

Como a computação é uma ciência em permanente evolução, e por se tratar o TENTOS de um gerenciador de ferramentas para projeto de CIs, constatou-se a necessidade de fazer com que a inclusão de novas ferramentas não ficasse a cargo de um gerente do TENTOS, pelos seguintes motivos:

- A necessidade de lançar novas versões do sistema com uma maior periodicidade, onde se incluíssem novas evoluções das ferramentas, arquivos de tecnologia, etc.
- Manutenção nos códigos fontes para cada nova inclusão;
- Permitir que os usuários agreguem ferramentas do seu interesse ao sistema.

Deste modo, considerou-se de fundamental importância, para que se obtivesse um melhor aproveitamento do TENTOS, que a manutenção (inclusão, alteração e exclusão) de ferramentas, arquivos de tecnologia e extensões de nome dos circuitos, ficasse a cargo de cada projetista que fosse usar o sistema.

Sendo um dos objetivos deste trabalho a fácil utilização do sistema TENTOS, verificou-se que a melhor forma de utilizar as ferramentas, integradas ao sistema, seria incorporá-las aos menus de trabalho do TENTOS. A implementação deste recurso foi facilitada pelo próprio MS-WINDOWS™, pois o objeto *menu* possui propriedades que permitem incluir ou excluir itens da sua lista de opções.

A figura 4.2 apresenta a caixa de diálogo onde serão manipuladas as ferramentas já incorporadas ou em fase de inclusão ao Gerenciador.

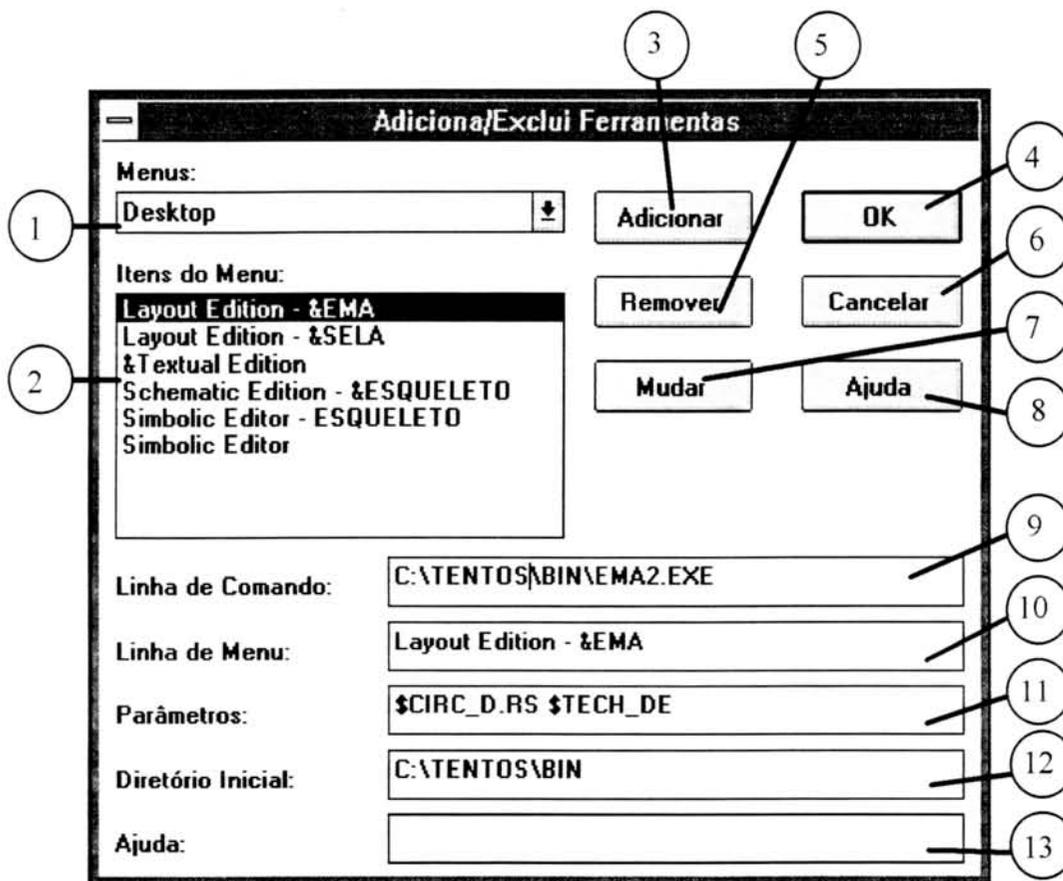


FIGURA 4.2 - Janela de Instalação/Configuração das Ferramentas

Na figura 4.2 nota-se que no campo '10' é utilizado um '&'. A finalidade deste caracter é que quando for apresentado o menu, a letra seguinte ao caracter '&' estará sublinhada. Desta forma, será fornecido ao usuário um *atalho* quando este utilizar o teclado. O gerenciamento desta informação é feito pelo MS-WINDOWS™.

A tabela 4.3 apresenta a descrição de cada um dos componentes da caixa de diálogo apresentada na figura 4.2.

TABELA 4.3 - Lista dos elementos da janela de Instalação/Configuração das ferramentas

Campo	Descrição
1	Indica que menu está sendo manipulado.
2	Indica os itens de menu pertencentes ao menu selecionado no campo '1'.
3	Adiciona ao menu selecionado a ferramenta que se encontra discriminada nos campos '9', '10', '11', '12' e '13'.
4	Encerra a execução da caixa de diálogo gravando no arquivo de inicialização as alterações realizadas.
5	Exclui o item de menu (campo '2') selecionado. Antes de realizar uma exclusão, o usuário deve selecionar no menu qual item deverá ser excluído. No momento em que realizar isto, todas as informações referentes àquela ferramenta serão apresentadas nos campos '9', '10', '11', '12' e '13'.
6	Encerra a execução da caixa de diálogo desconsiderando alterações realizadas.
7	Altera o item de menu (campo '2') selecionado. Antes de realizar uma alteração, o usuário deve selecionar no menu qual item deverá ser alterado. No momento em que realizar isto, todas as informações referentes àquela

	ferramenta serão apresentadas nos campos '9', '10', '11', '12' e '13', a partir disto realizam-se as alterações necessárias e aciona-se este botão.
8	Aciona um Help referente a esta caixa de diálogo.
9	Apresenta a linha de comando da execução da ferramenta, isto é o seu path seguido do nome do executável da ferramenta.
10	Apresenta a descrição da ferramenta que irá constar do menu do TENTOS.
11	Lista de parâmetros que devem acompanhar a linha de comando no momento de execução da ferramenta. Para maiores informações, consultar a seção 4.1.2.1.
12	Diretório inicial da ferramenta. Utilizando-se a variável \$DIR_WK na lista de parâmetros, está se fazendo uma referência ao diretório inicial da ferramenta.
13	Linha de Help da ferramenta. Esta linha será apresentada na barra de status.

As informações fornecidas pelo usuário no momento da inclusão/alteração de ferramentas ficam armazenadas no arquivo de inicialização (*.INI) que está sendo utilizado pelo sistema no momento da operação. Portanto, no momento em que é inicializado o TENTOS, o arquivo de inicialização é lido e o sistema tem a configuração dos seus menus ordenadas conforme disposição estabelecida pelo projetista no momento em que adicionava/alterava/excluía ferramentas.

Cabe salientar que as ferramentas a serem incluídas no sistema podem necessitar de parâmetros ou dados para serem inicializadas. O sistema TENTOS permite que estes parâmetros sejam definidos e repassados às ferramentas no momento da sua execução na linha de comando. Por exemplo, a configuração para o editor de leiautes EMA, apresentada na figura 4.2, será executada da seguinte forma: C:\TENTOS\BIN\EMA2.EXE \$CIRC_D.RS \$TECH_DE, onde \$CIRC_D e \$TECH_DE referem-se ao nome do circuito e ao arquivo de tecnologia, respectivamente, que foram selecionados através do sistema TENTOS. Portanto, para se obter o melhor desempenho deste sistema, as ferramentas incorporadas devem permitir que os seus parâmetros sejam configurados na linha de comando.

4.1.2.1 Lista de Variáveis

Como parâmetros de ferramentas no momento da sua execução, poderemos ter constantes ou variáveis. A tabela 4.4 ilustra quais variáveis poderão ser utilizadas como parâmetros no momento de configurar a execução das ferramentas.

TABELA 4.4 - Lista das variáveis que podem ser usadas como parâmetros

Variáveis	Descrição	Exemplos
Nome do Circuito	Indica o nome do circuito selecionado na opção "Arquivo-Circuito"	
\$CIRC_DE	Diretório + Nome + Extensão	C:\TENTOSUS\INV.RS
\$CIRC_D	Diretório + Nome	C:\TENTOSUS\INV
\$CIRC_E	Nome + Extensão	INV.RS
\$CIRC	Nome	INV
Arquivo de tecnologia	Indica o nome do arquivo de tecnologia selecionado na opção "Arquivo-Configuração-Opções-Tecnologia".	
\$TECH_DE	Diretório + Nome + Extensão	C:\TENTOS\TEC\CMOS12.TEC
\$TECH_D	Diretório + Nome	C:\TENTOS\TEC\CMOS12
\$TECH	Nome	CMOS12

Diretórios de Trabalho	Indicam o caminho para os diretórios de trabalho selecionados na opção "Arquivo-Configuração-Opções-Diretórios".	
\$DIR_IN	Diretório de Entrada	C:\TENTOSUS
\$DIR_OU	Diretório de Saída	C:\TENTOSUS
\$DIR_US	Diretório dos Arq. do Usuário	C:\TENTOSUS
\$DIR_EX	Diretório dos Aplicativos	C:\TENTOS\BIN
\$DIR_TC	Diretórios dos Arq. Tecnologia	C:\TENTOS\TEC
\$DIR_LB	Diretório das Bibliotecas	C:\TENTOS\LIB
Diretório Inicial	Indica o caminho para o diretório inicial das ferramentas indicado no momento de instalação das ferramentas no <i>gerenciador</i> .	
\$DIR_WK		C:\TENTOS\BIN
Arquivo de Configuração	Indica o nome do arquivo de configuração selecionado na opção "Arquivo-Configuração-Lê ".CFG"".	
\$CONF_DE	Diretório + Nome + Extensão	C:\TENTOSUS\TENTOSUS.CFG
\$CONF_E	Nome + Extensão	TENTOSUS.CFG
\$CONF	Nome	TENTOSUS
Arquivo de Inicialização	Indica o nome do arquivo de inicialização selecionado na opção "Arquivo-Configuração-Lê ".INI"".	
\$INIC_DE	Diretório + Nome + Extensão	C:\TENTOSUS\TENTOS.INI
\$INIC_E	Nome + Extensão	TENTOS.INI
\$INIC	Nome	TENTOS

Depois de vista a lista de variáveis, um exemplo ilustrará bem a diferença entre constantes e variáveis.

Exemplo:	
A execução do editor de leiautes (EMA) [STE 89b][STE 89c] poderia ser feita de dois modos:	
<ul style="list-style-type: none"> Com constantes: 	
C:\TENTOS\BIN\EMA2.EXE	(linha de comando)
C:\MORAES\MODEM.RS C:\TENTOS\TEC\CMOS15.TEC	(parâmetros: circuito e arquivo de tecnologia)
Deste modo o Gerenciador sempre executaria o mesmo arquivo.	
<ul style="list-style-type: none"> Ou com variáveis: 	
C:\TENTOS\BIN\EMA2.EXE	(linha de comando)
\$CIRC_D.RS \$TECH_DE	(parâmetros: circuito e arquivo de tecnologia)
Deste modo, a cada execução do editor de leiaute, seria passado como parâmetro para a sua execução o nome do circuito selecionado no presente momento (com extensão .RS), bem como o nome do arquivo de tecnologia que se encontra selecionado na hora da execução.	

4.1.3 Associando novos arquivos de tecnologia ao TENTOS

A figura 4.3 apresenta a caixa de diálogo onde serão manipulados os arquivos de tecnologia já incorporados ou em fase de inclusão ao Gerenciador.

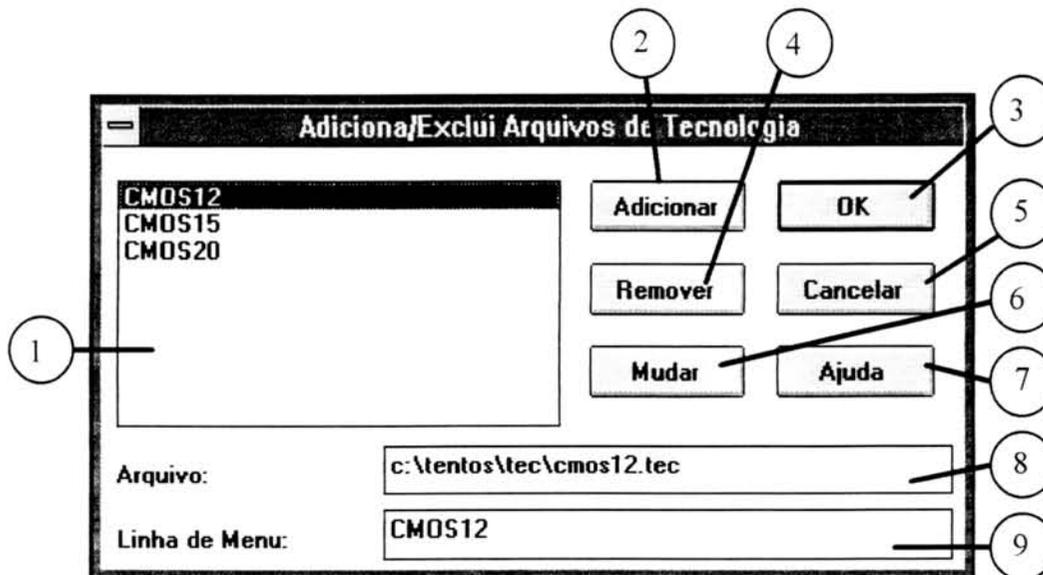


FIGURA 4.3 - Janela de Instalação/Configuração dos Arquivos de Tecnologia

A tabela 4.5 apresenta a descrição de cada um dos componentes da caixa de diálogo apresentada na figura 4.3.

TABELA 4.5 - Elementos da janela de Instalação/Configuração dos Arquivos de Tecnologia

Campo	Descrição
1	Indica os arquivos de tecnologia já incluídos ao TENTOS.
2	Adiciona o arquivo de tecnologia que está discriminado nos campos '8' e '9'.
3	Encerra a execução da caixa de diálogo, gravando no arquivo de inicialização as alterações realizadas.
4	Exclui o arquivo de tecnologia selecionado no campo '1', antes de realizar uma exclusão o usuário deve selecionar no campo '1' qual item deverá ser excluído, no momento em que realizar isto todas as informações referentes àquele arquivo de tecnologia serão apresentadas nos campos '8' e '9'.
5	Encerra a execução da caixa de diálogo desconsiderando alterações realizadas.
6	Altera o arquivo de tecnologia selecionado no campo '1', antes de realizar uma alteração o usuário deve selecionar no campo '1' qual item deverá ser alterado, no momento em que realizar isto todas as informações referentes àquele arquivo de tecnologia serão apresentadas nos campos '8' e '9', a partir disto realiza-se as alterações necessárias e aciona-se este botão.
7	Aciona um Help referente a esta caixa de diálogo.
8	Apresenta o nome completo do arquivo de tecnologia, isto é o seu <i>path</i> seguido do nome do arquivo.
9	Apresenta a descrição do arquivo de tecnologia que irá constar do menu do TENTOS.

4.1.4 Associando novas extensões de circuitos ao TENTOS

A figura 4.4 apresenta a caixa de diálogo onde serão manipulados os tipos de extensões que serão oferecidos ao projetista no momento de seleção do circuito.

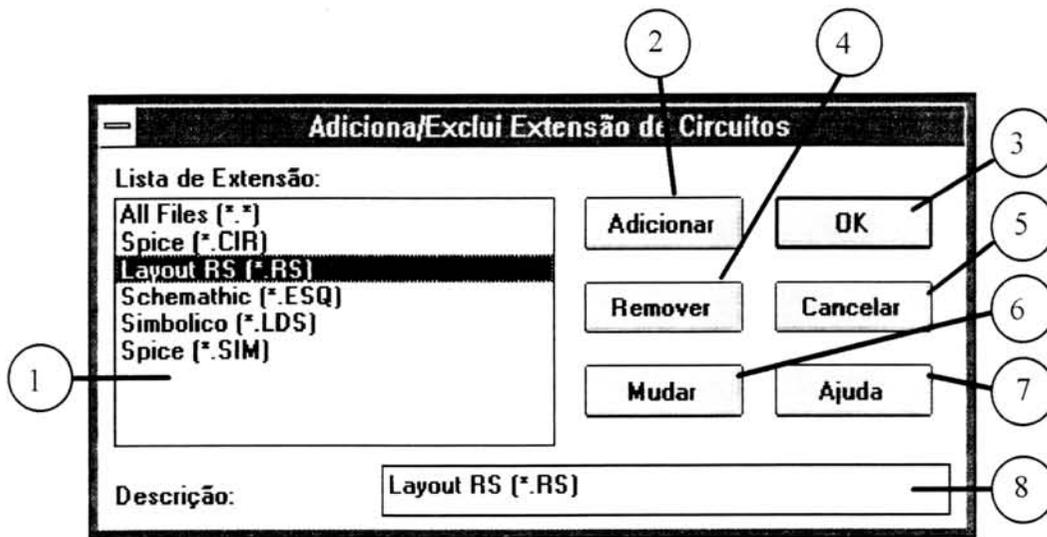


FIGURA 4.4 - Janela de Instalação/Configuração dos Tipos de Extensões

A tabela 4.6 apresenta a descrição de cada um dos componentes da caixa de diálogo apresentada na figura 4.4.

TABELA 4.6 - Elementos da janela de Instalação/Configuração dos Tipos de Extensões

Campo	Descrição
1	Indica as extensões de circuitos já incluídas ao TENTOS.
2	Adiciona a extensão de circuito que encontra-se discriminada no campo '1'.
3	Encerra a execução da caixa de diálogo gravando no arquivo de inicialização as alterações realizadas.
4	Exclui a extensão de circuito selecionada no campo '1', antes de realizar uma exclusão o usuário deve selecionar no campo '1' qual item deverá ser excluído, no momento em que realizar isto, a descrição será apresentada no campo '8'.
5	Encerra a execução da caixa de diálogo desconsiderando alterações realizadas.
6	Altera a extensão de circuito selecionada no campo '1', antes de realizar uma alteração o usuário deve selecionar no campo '1' qual item deverá ser alterado, no momento em que realizar isto, a descrição será apresentada no campo '8', a partir disto realiza-se as alterações necessárias e aciona-se este botão.
7	Aciona um Help referente a esta caixa de diálogo.
8	Apresenta a extensão de circuito, isto é, o seu significado seguido de um parênteses, onde entre estes parênteses deverá estar o formato da extensão a que diz respeito a descrição.

Deve-se observar que uma descrição só será correta, ou seja, só terá utilidade, se o projetista indicar a extensão entre parênteses, por exemplo: *Layout RS (*.RS)*, *Descrição Spice (*.CIR)*. Caso não forem informadas as extensões entre parênteses, de nada adianta a descrição da extensão.

4.2 A Estrutura do Sistema TENTOS

A figura 4.5 apresenta a estrutura de menus do sistema TENTOS *for Windows* na versão 1.00.

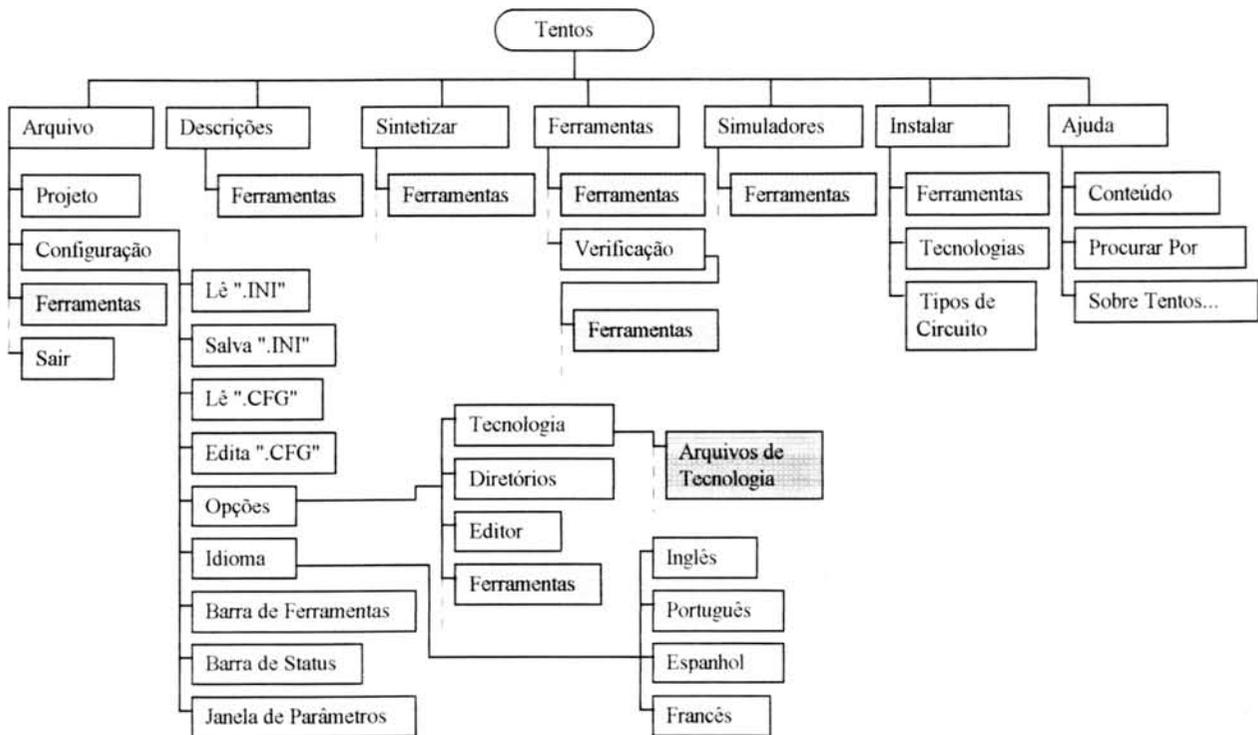


FIGURA 4.5 - A estrutura de Menus do Sistema TENTOS

Os pontilhados existentes na figura 4.5 representam o fato de o menu não possuir um número fixo de elementos naquela posição. Os retângulos, cujos interiores são de cor cinza, claro ou escuro, exercem apenas uma função ilustrativa, pois nestas posições é que serão incorporadas as ferramentas (vide seção 4.1.2) e os arquivos de tecnologia (vide seção 4.1.3).

Na seção seguinte será detalhado cada um dos itens que compõe a estrutura de menus apresentada na figura 4.5.

4.2.1 Submenu *Arquivo*

Projeto: Seleciona o nome do projeto que será manipulado pelas ferramentas integradas ao Gerenciador. A figura 4.6 apresenta o leiaute da janela onde deverá ser indicado o nome do projeto a ser manipulado pelo projetista.

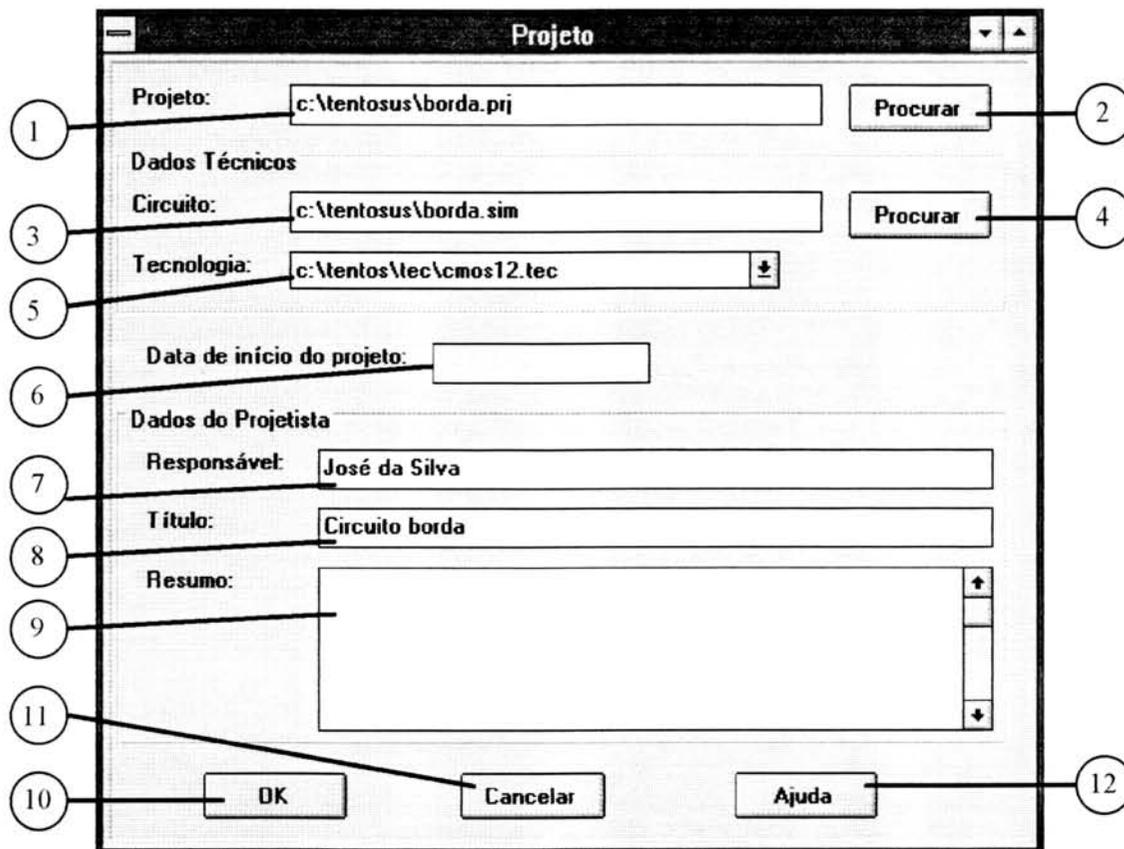


FIGURA 4.6 - Caixa de Diálogo - Projeto (Informações do Projeto).

TABELA 4.7 - Itens da caixa de diálogo - Projeto (Informações do Projeto)

Itens	Descrição
1	Onde deverá ser escrito o nome completo (<i>diretório + nome</i>) do projeto a ser manipulado.
2	Pesquisa o nome do projeto na árvore de diretórios.
3	Onde deverá ser escrito o nome completo (<i>diretório + nome</i>) do circuito integrante do projeto descrito no campo '1'. Esta informação poderá ser utilizada pelo projetista através das variáveis \$CIRC_DE, \$CIRC_D, \$CIRC_E e \$CIRC (a finalidade de cada uma destas variáveis é apresentada na tabela 4.4).
4	Pesquisa o nome do circuito na árvore de diretórios.
5	Seleção do arquivo de tecnologia pertinente ao projeto.
6	Data de início do projeto. Informada pelo Sistema.
7	Nome do responsável pelo projeto.
8	Título do projeto.
9	Resumo do projeto.
10	Conclui operação de seleção do projeto.
11	Cancela operação de seleção do projeto.
12	Mostra texto de Ajuda.

As informações apresentadas nos campos '3', '5', '6', '7', '8' e '9' fazem parte do arquivo projeto.

Sair: Encerra o uso do sistema TENTOS.

4.2.1.1 Submenu Arquivo - Configuração

Lê “.INI”: Seleciona o nome do arquivo de inicialização, ver ANEXO 2, que será utilizado como configuração do sistema. As ferramentas poderão utilizar esta informação através das variáveis, alteradas por este item de menu, \$INIC_DE, \$INIC_E e \$INIC (a finalidade de cada uma destas variáveis é apresentada na tabela 4.4).

Salva “.INI”: Salva a configuração atual do sistema em um arquivo selecionado pelo usuário.

Lê “.CFG”: Seleciona o nome do arquivo de configuração, ver ANEXO 3, que será utilizado pelas ferramentas que acompanham o sistema TENTOS. As ferramentas poderão utilizar esta informação através das variáveis, alteradas por este item de menu, \$CONF_DE, \$CONF_E e \$CONF (a finalidade de cada uma destas variáveis é apresentada na tabela 4.4).

Edita “.CFG”: Permite a edição do arquivo de configuração selecionado na opção *Arquivos - Configuração - Lê “.CFG”*. Utilizando-se do editor de textos selecionado através da opção *Arquivo - Configuração - Opções - Editor*.

Barra de Ferramentas: Habilita ou desabilita o aparecimento da barra de ferramentas (vide figura 4.1) durante a execução do gerenciador.

Barra de Status: Habilita ou desabilita o aparecimento da barra de *status* (vide figura 4.1) durante a execução do gerenciador.

Janela de Parâmetros: Habilita ou desabilita o aparecimento da janela de parâmetros (vide figura 4.20) no momento de execução das ferramentas.

4.2.1.2 Submenu Arquivo - Configuração - Opções

Diretórios: Serve para configurar os caminhos dos diretórios manipulados diretamente pelo gerenciador. A tabela 4.8 apresenta os diretórios que são manipulados nesta opção.

TABELA 4.8 - Lista dos diretórios manipulados pelo Sistema TENTOS.

Diretórios	Variáveis
Diretório de Entrada.	\$DIR_IN
Diretório de Saída.	\$DIR_OU
Diretório do Usuário.	\$DIR_US
Diretório dos Arquivos de Tecnologia.	\$DIR_TC
Diretório das Ferramentas.	\$DIR_EX
Diretório das Bibliotecas.	\$DIR_LB

A tabela 4.4 apresenta melhores detalhes de cada uma das variáveis apresentadas acima.

Editor: Serve para indicar qual editor de textos deve ser utilizado pelo projetista para alteração de dados no arquivo de configuração (.CFG) selecionado.

4.2.1.3 Submenu Arquivo - Configuração - Opções - Tecnologia

Arquivos de Tecnologia: Seleciona qual será o arquivo de tecnologia, dentre os inseridos no gerenciador através da opção *INSTALAR - TECNOLOGIAS* (vide seção 4.1.3). As ferramentas poderão utilizar esta informação através das variáveis, alteradas por este item de menu, \$TECH_DE, \$TECH_D e \$TECH (a finalidade de cada uma

destas variáveis é apresentada na tabela 4.4). Para o arquivo de projeto, a alteração do arquivo de tecnologia, através desta opção, poderá ser temporária ou definitiva.

4.2.1.4 Submenu *Arquivo - Configuração - Idioma*

Inglês: Todas as mensagens do Sistema TENTOS serão apresentadas em inglês.

Português: Todas as mensagens do Sistema TENTOS serão apresentadas em português.

Espanhol: Todas as mensagens do Sistema TENTOS serão apresentadas em espanhol.

Francês: Todas as mensagens do Sistema TENTOS serão apresentadas em francês.

4.2.2 Submenus *Descrições, Sintetizar, Ferramentas e Simuladores*

Estes submenus não possuem nenhum item vinculado ao TENTOS. Todas as opções que venham a ser incorporadas a eles serão feitas através do módulo INSTALAR - FERRAMENTAS (vide seção 4.1.2). Desta forma são menus que possuem uma função de propiciar ao usuário uma melhor distribuição das ferramentas, de acordo com as suas finalidades.

4.2.3 Submenu *Instalar*

Ferramentas: Opção de menu que habilitará a janela de configuração/instalação de ferramentas. Para maiores informações ver seção 4.1.2.

Tecnologias: Opção de menu que habilitará a janela de configuração/instalação de arquivos de tecnologia. Para melhores informações ver seção 4.1.3.

Tipos de Circuito: Opção de menu que habilitará a janela de configuração/instalação dos formatos possíveis de arquivos de descrição de circuitos que serão acessados na opção de menu Arquivos - Circuitos. Para melhores informações ver seção 4.1.4.

4.3 A Configuração *Standard* do TENTOS

Esta seção tem por objetivo apresentar as ferramentas que compõe a configuração básica do Sistema TENTOS.

A figura 4.7 apresenta a disposição das ferramentas nos menus do Sistema TENTOS na versão 1.00.

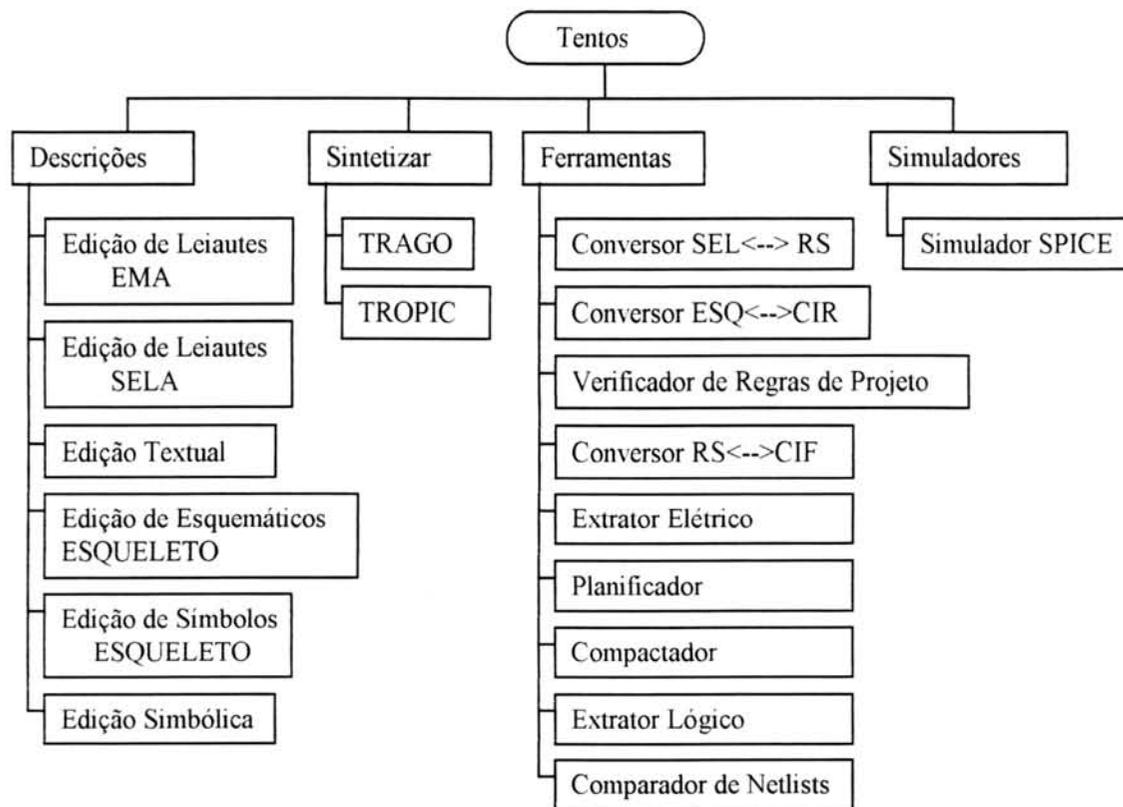


FIGURA 4.7 - A disposição das ferramentas nos menus do Sistema TENTOS

4.3.1 Menu *Descrições*

4.3.1.1 Editor de Leiaute - EMA

Trata-se de um editor gráfico de máscaras para circuitos integrados que permite a montagem do leiaute de uma célula, assim como visualizar e modificar a descrição hierárquica de um leiaute [STE 89b] [STE 89c]. Com o uso deste editor pode-se criar novas definições de células, percorrer livremente as diversas células que compõem a descrição de um circuito, e criar instâncias de células, editando interativamente as transformações que a célula deve sofrer ao ser chamada.

A figura 4.8 apresenta a interface gráfica do editor de máscaras EMA, desenvolvida para ambiente MS-DOS™.

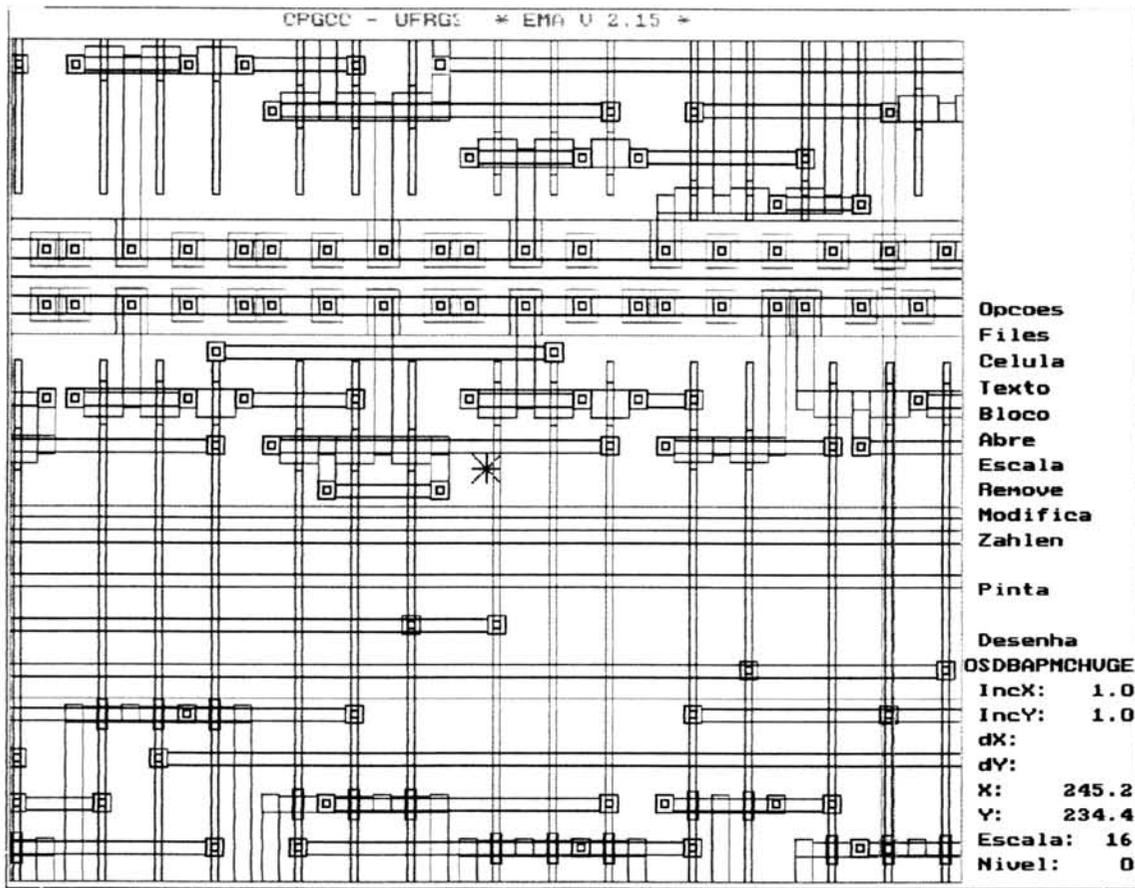


FIGURA 4.8 - Interface do Editor de Máscaras - EMA.

4.3.1.2 Editor de Leiaute - SELA

É um editor de leiaute desenvolvido pelo CPGCC/UFRGS através de convênio SID Microeletrônica - CPGCC [CAA 91]. Este editor visa oferecer uma estação de baixo custo para desenho de leiaute de circuitos integrados baseado em equipamento do tipo IBM-PC™. A sua construção viabiliza a manipulação de circuitos contendo milhares de elementos geométricos.

As características desse editor são: rápida exibição de grandes circuitos; uso de diversas primitivas geométricas como polígonos, círculos, fios e retângulos; possibilidade de uso de geometrias Manhattan, 45 graus e qualquer ângulo; interface com o usuário extremamente amigável com o uso de bindkeys e ícones além de menus mais flexíveis.

Com o objetivo de alcançar um alto desempenho nas operações que envolvem apenas uma área específica do circuito, foi adotada a estrutura de quad-tree.

A figura 4.9 apresenta a interface gráfica do editor de leiautes SELA, desenvolvida para ambiente MS-DOS™.

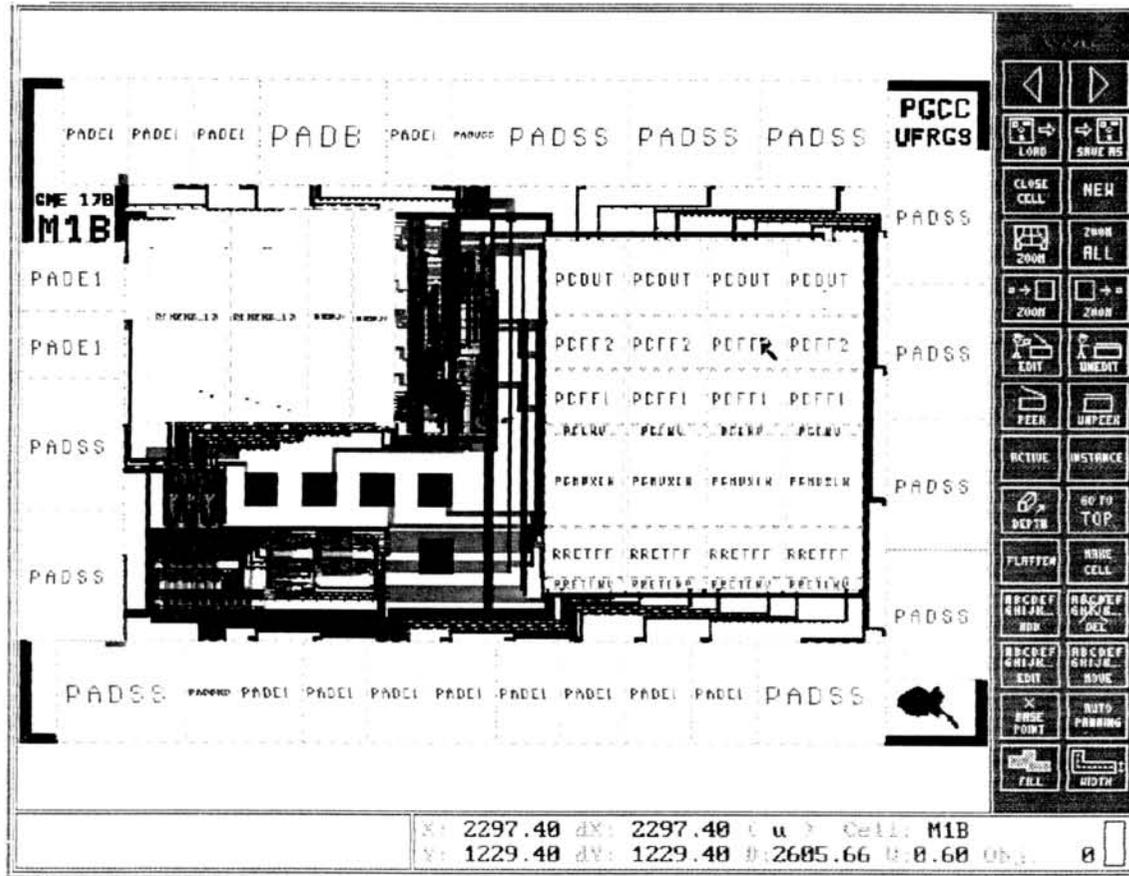


FIGURA 4.9 - Interface do Editor de Máscaras - SELA.

4.3.1.3 Edição Textual do Circuito

É utilizado um editor de texto cuja finalidade é possibilitar as descrições textuais de circuitos, isto é, neste editor não existe configuração de layout de página, de parágrafo, como meio de descrição dos circuitos.

É utilizado o editor que acompanha a versão básica do MS-WINDOWS™.

4.3.1.4 Editor de Esquemáticos

Consiste de um software para geração de diagramas elétricos de maneira gráfica interativa (ESQUELETO) [GME 92] [SAC 90]. Este editor manipula símbolos, armazenados em bibliotecas, e permite ao usuário a criação de esquemas lógicos e elétricos, segundo uma hierarquia.

Um esquemático será representado internamente como um grafo dirigido sem ciclos, contendo um nodo inicial, denominado raiz do esquemático e um número arbitrário de nodos adicionais. Cada nodo do esquemático denomina-se esquema. As arestas do grafo estão associadas a primitivas especiais, denominadas blocos funcionais. Estes determinam as interconexões hierárquicas entre esquemas de um esquemático.

Deste modo, um esquema consiste de símbolos e blocos funcionais unidos por conexões, formando um módulo funcional de dimensões e complexidade determinadas pelo usuário, quando de sua criação.

A figura 4.10 apresenta a interface gráfica do editor de esquemáticos ESQUELETO, desenvolvida para ambiente MS-DOS™.

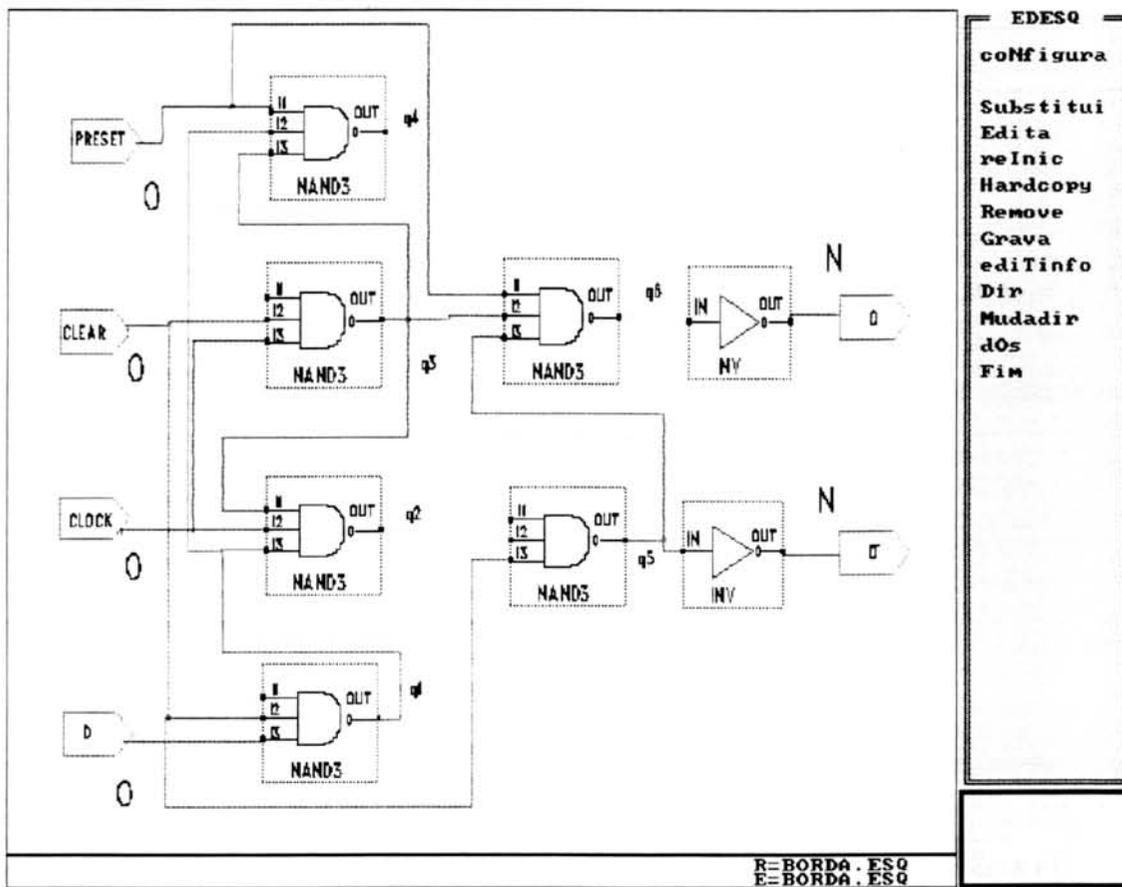


FIGURA 4.10 - Interface do Editor de Esquemáticos.

4.3.1.5 Editor de Símbolos

Esta ferramenta é um módulo auxiliar, mas de fundamental importância para o editor de esquemáticos. A sua função é a de criar novos símbolos que serão incorporados à biblioteca de símbolos utilizadas pelo ESQUELETO [VAS 90].

É fornecida ao usuário uma biblioteca básica, com diversas portas lógicas, registradores e contadores. Através da manipulação destes elementos são criados novos símbolos para serem utilizados pelo usuário no editor de esquemáticos.

A figura 4.11 apresenta a interface gráfica do editor de símbolos, desenvolvida para ambiente MS-DOS™.

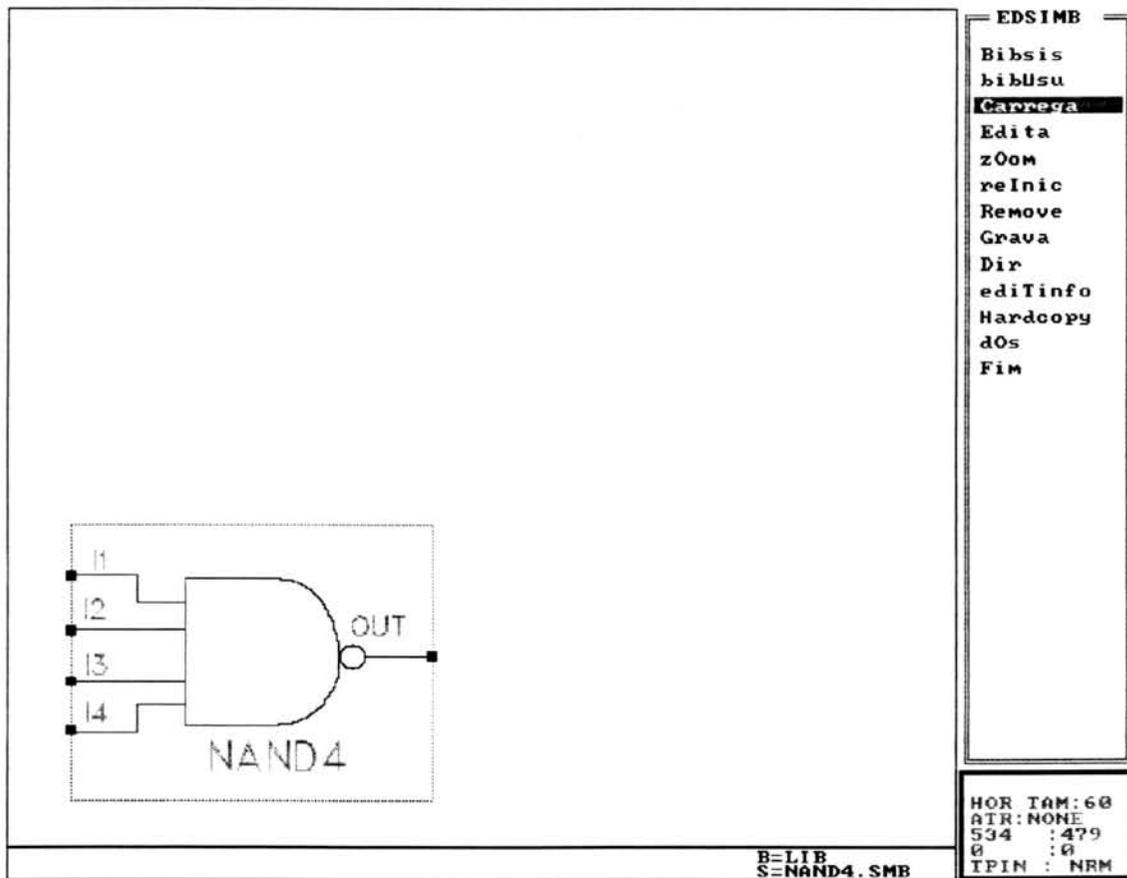


FIGURA 4.11 - Interface do Editor de Símbolos.

4.3.1.6 Editor Simbólico - SIMBY

O SIMBY [SOT 93] é um editor simbólico capaz de descrever graficamente uma célula de um circuito através de elementos simbólicos, como transistores, fios e contatos, e também inserir parâmetros nesses elementos, como largura de um fio e largura do gate de um transistor em função das regras mínimas da tecnologia.

Cada célula descrita no SIMBY é armazenada num arquivo que contém a sua descrição. Essa descrição está em LDS (Linguagem de Descrição Simbólica), linguagem que pode ser utilizada posteriormente como entrada para outras ferramentas do sistema, como o compactador SYLC.

Como o SIMBY é simbólico, o projetista pode abstrair-se completamente das regras geométricas, preocupando-se especificamente com a topologia das células.

A figura 4.12 apresenta a interface gráfica do editor simbólico, desenvolvida para ambiente MS-WINDOWS™.

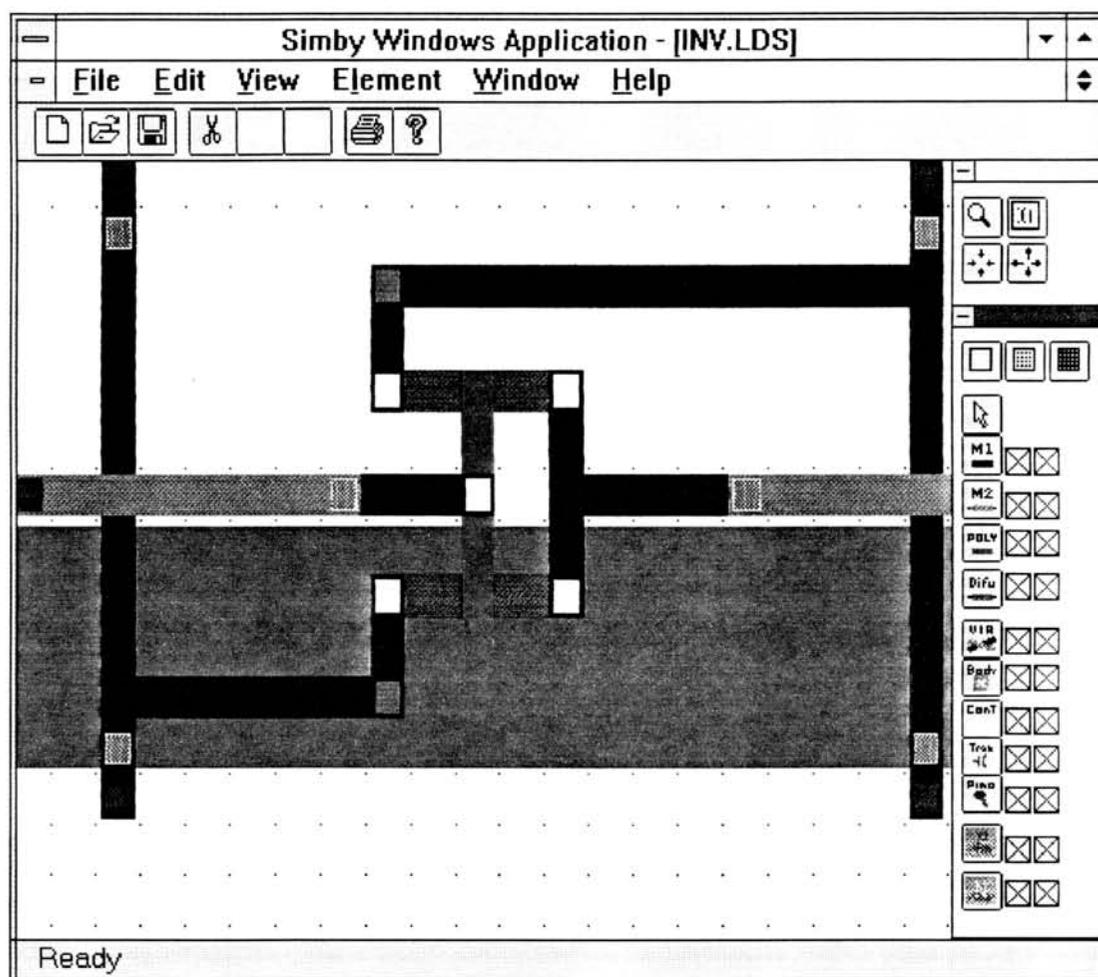


FIGURA 4.12 - Interface do Editor Simbólico.

4.3.2 Menu *Sintetizar*

4.3.2.1 Trago

Sistema para síntese automática de circuitos em lógica aleatória [MOR 89a] [MOR 90] [MOR 90a] [MOR 90b]. A descrição do circuito poderá ser feita através de um editor de esquemáticos ou textualmente. Ao invés de termos uma biblioteca geométrica (previamente desenhada) teremos uma biblioteca estrutural; com circuitos descritos em formato *SPICE*. Mudando a tecnologia, simula-se novamente a célula alterando as dimensões dos transistores, não necessitando todo o redesenho manual das células. E ao invés de arquivos que descrevem a topologia das células, há apenas um arquivo de regras de desenho para cada tecnologia.

Este sistema de síntese explora características tais como:

- **LEIAUTE SIMBÓLICO:** permite descrição independente de regras de desenho, pois todas as larguras e distâncias mínimas são unitárias.
- **BIBLIOTECA ESTRUTURAL:** descrição das células a nível de transistores, e não mais a nível de retângulos.
- **ESTRUTURA EM BANDAS:** o circuito será dividido em bandas, porém estas não terão altura fixa, como em *standard cells* tradicional. Elas terão altura variável, conforme necessidade de roteamento.

- **DESCRIÇÃO GRÁFICA DO CIRCUITO:** será utilizado um editor de esquemas que forneça a descrição estrutural do circuito.
- **AUSÊNCIA DE CANAIS DE ROTEAMENTO:** o roteamento será sobre as células.

Depois de feita a edição do circuito (textual ou gráfica), que independem do TRAGO, os passos de projeto para a geração do leiaute são compostos da compilação da descrição e do particionamento, além de módulos para visualização dos resultados. A figura 4.13 ilustra o fluxo do projeto descrito.

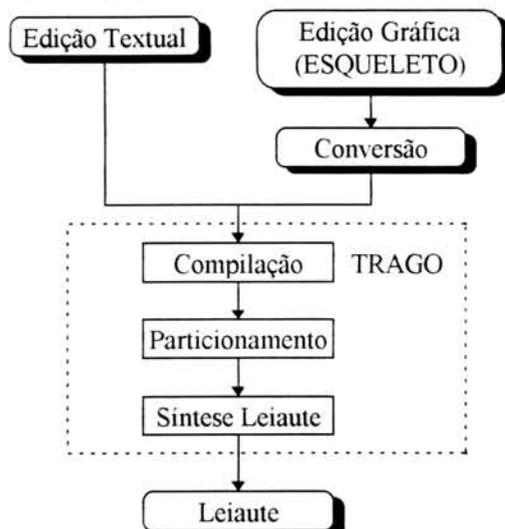


FIGURA 4.13 - Sequência de síntese do TRAGO.

A figura 4.14 apresenta a interface da ferramenta de síntese automática TRAGO.

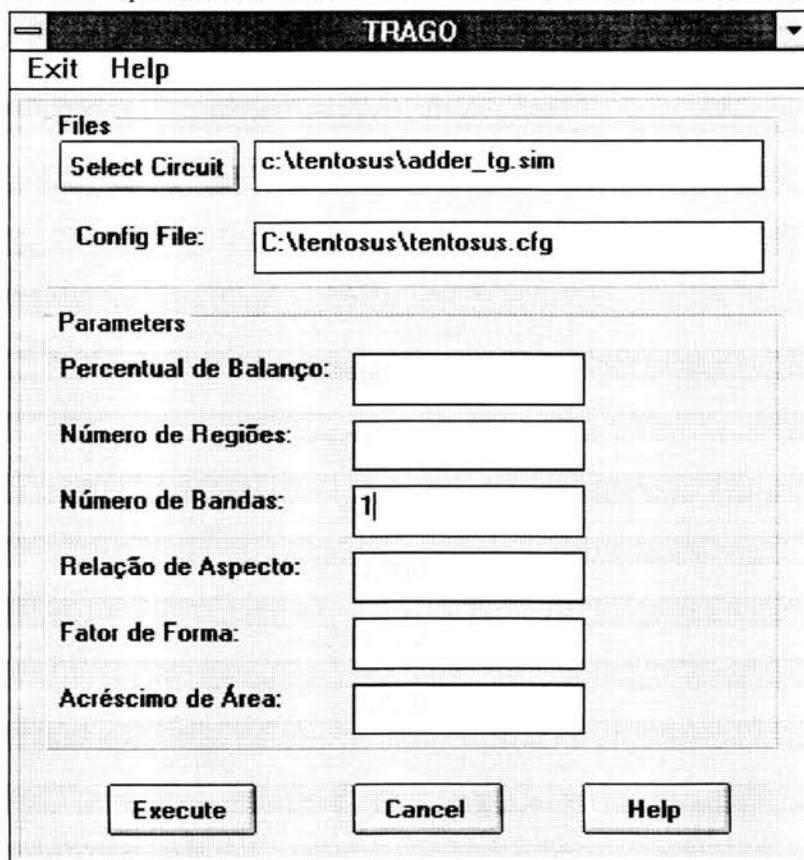


FIGURA 4.14 - Interface do Sistema Trago.

4.3.2.2 Tropic

É uma ferramenta de síntese de leiaute para circuitos em lógica aleatória (nível de portas lógicas). O sistema não utiliza biblioteca de células pré-caracterizadas, permitindo assim a independência em relação às regras de projeto. O circuito é particionado em linhas horizontais de células, chamadas bandas, não havendo canais de roteamento entre as mesmas. O roteamento horizontal é realizado com o primeiro nível metal e o roteamento vertical com o segundo nível de metal. O estilo de leiaute adotado é o linear-matrix multi-banda.

No TROPIC como o simbólico gerado vai ser compactado após a síntese, isto implica em um melhor aproveitamento da área, propiciando desta forma um circuito mais denso que o obtido pelo TRAGO. Outro ganho do TROPIC em relação ao TRAGO é que como o estilo de implementação do TROPIC é linear-matrix (alimentação horizontal de transistores) existem menos capacitâncias parasitas, propiciando, desta forma, um melhor desempenho elétrico do circuito gerado com o TROPIC.

O método adotado por este sistema, para o posicionamento dos transistores, tem a vantagem de reduzir sensivelmente as capacitâncias parasitas, obtendo assim um ótimo desempenho elétrico. Outra vantagem do sistema TROPIC é a alta densidade de integração, pois após a síntese topológica (posicionamento e roteamento) utiliza-se um compactador de leiaute.

A figura 4.15 ilustra o fluxo de projeto utilizando a ferramenta TROPIC.

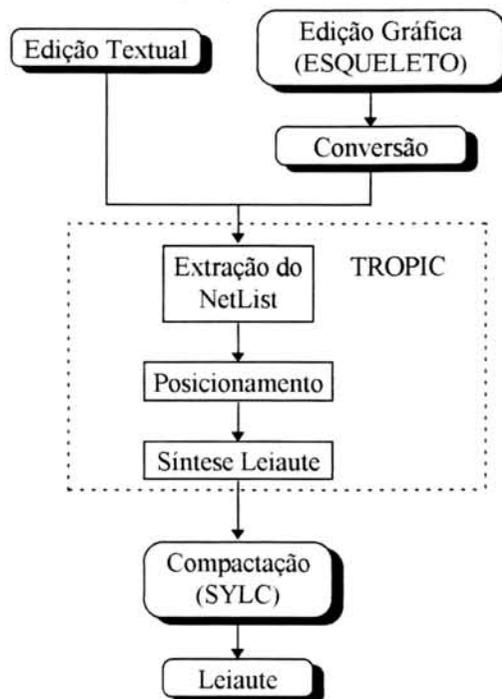


FIGURA 4.15 - Seqüência de síntese do TROPIC.

O sistema TROPIC [MOR 94] possui uma biblioteca de sub-circuitos em formato *SPICE*. Para usar as células existentes nesta biblioteca basta colocar no início da descrição *Spice* o comando **“.include librairie”** (arquivo em *\$GME/cfg*).

Exemplo de descrição *Spice* para o TROPIC :

.include librairie

X1 a b c carry sum vcc adder

```
*interface: a      orient o
*interface: b      orient o
*interface: c      orient o
*interface: carry  orient s
*interface: sum    orient s
```

```
*****
```

O comando “interface” especifica os pinos da macro-célula, e o parâmetro “orient” a borda de um dado pino (norte, sul, leste, oeste)

As portas disponíveis na biblioteca de sub-circuitos são:

```
INV in out
INV2 in out (invert with 4 transistors)
BUFFER in out
TRG in out h nh (transmission gate)
MUX2TO1 i1 i2 C out (multiplexeur : if (C==1) out=i1 else out=i2)

NAND (2 a 9 entradas) : in1 in2 ... ink out
AND (2 a 9 entradas) : in1 in2 ... ink out
NOR (2 a 9 entradas) : in1 in2 ... ink out
OR (2 a 9 entradas) : in1 in2 ... ink out
XOR (2 entradas) : in1 in2 out
NXOR (2 entradas) : in1 in2 out
NXOR_NAND : duas funções: p = in1 nxor in2 e y = in1 nand in2 - pins : in1 in2 p y
XORGATE (2 entradas) : in1 in2 out (xor with nands)

LATCH : data Q load (storage at load=1)
DFF : data Q load (storage at load=1) (with transimtion gates)
DFFG : dff implemented with nand gates
DFFCLEAR : data clear clk q nq
ADD1 : sum = a + b + 1 . carry = a OR b - pins : a b carry sum
ADD0 : sum = a + b . carry = a AND b - pins : a b carry sum
ADDER : sum = a + b + cin. carry = ab + cin(a+b) - pins : a b cin carry sum

CLA1, CLA2, CLA3, CLA4 - specific gates to the carry look-ahead
ADDER4 - adder 4 bits, architecture carry look-ahead
pins: ncin a0 a1 a2 a3 b0 b1 b2 b3 s0 s1 s2 s3 ncout
ADDER16 - adder 16 bits, architecture carry look-ahead
pins: ncin a0...a15 b0...b15 s0...s15 ncout
```

A figura 4.16 apresenta a interface do Sistema TROPIC, desenvolvida para ambiente MS-WINDOWS™.

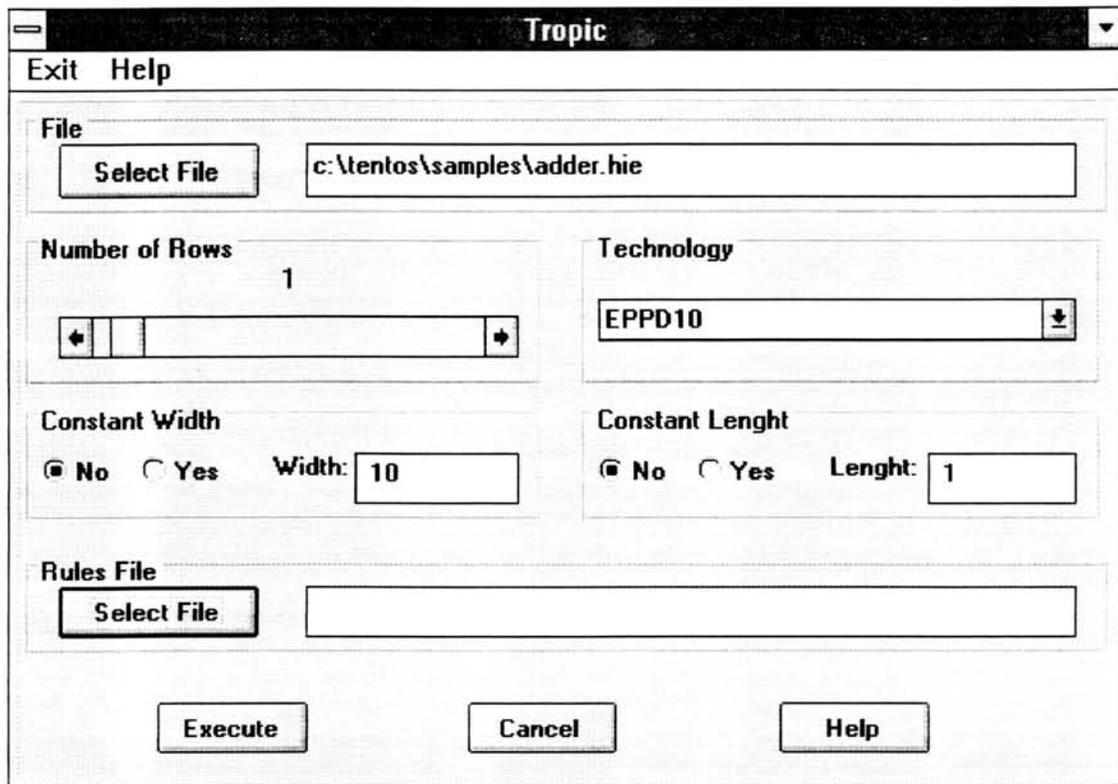


FIGURA 4.16 - Interface do Sistema Tropic.

4.3.3 Menu *Ferramentas*

4.3.3.1 Conversor Formato Leiaute SEL p/ Leiaute RS

Conversor de formatos de circuito. Tanto o circuito de entrada como o circuito de destino poderão ser do tipo: Leiaute SEL, Leiaute RS ou Leiaute CIF.

A figura 4.17 apresenta a interface gráfica do conversor de formatos, desenvolvida para ambiente MS-DOS™.

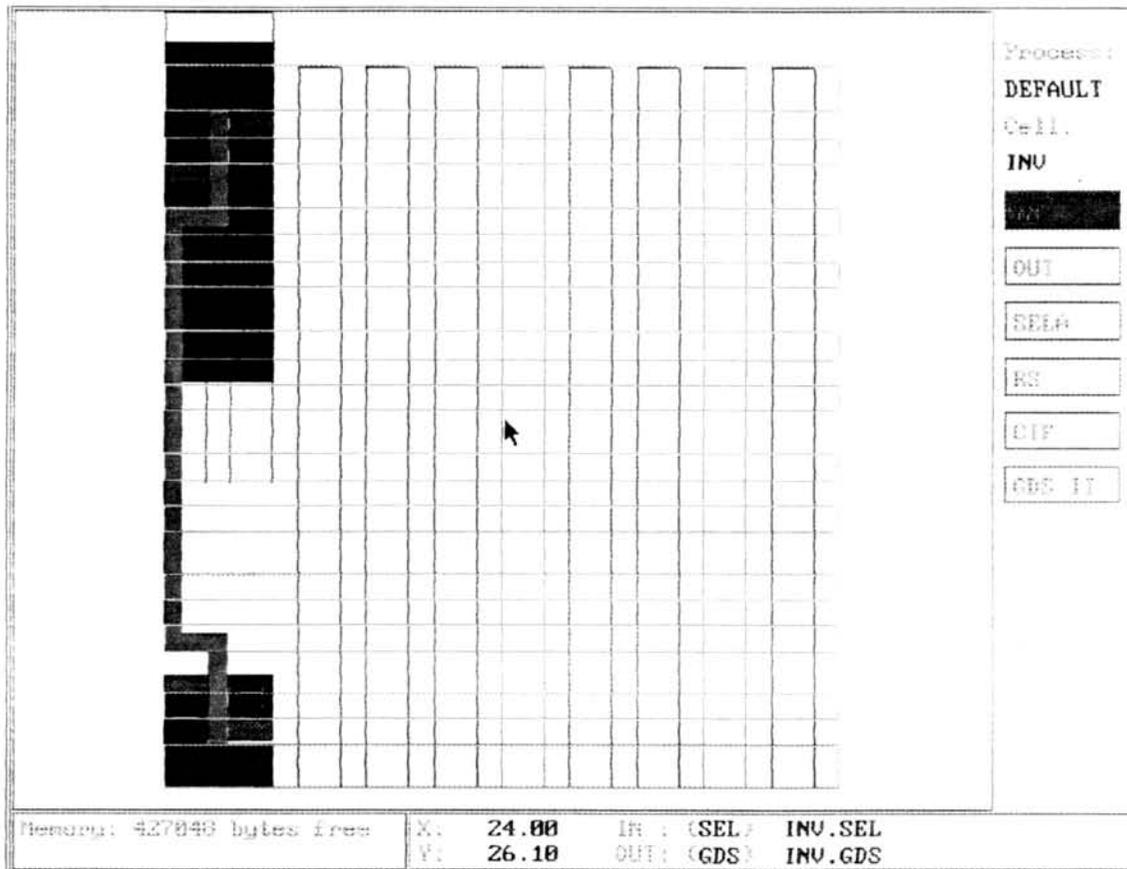


FIGURA 4.17 - Interface do Conversor de Formatos SEL ↔ RS.

4.3.3.2 Conversor Formato Esquemático ↔ Formato Spice

Este software é um conversor de arquivos de descrição de circuitos. A partir de um arquivo descrito em formato ESQ, isto é, para ser manipulado no editor de esquemáticos, cria-se uma versão do mesmo circuito em formato *netlist SPICE*.

4.3.3.3 Verificador das Regras de Projeto

Este aplicativo tem por função a verificação das regras de projeto de um leiaute, tais como distâncias e larguras mínimas [GOM 88]. Em sua execução é imperativo que o circuito já tenha sido planejado, só assim as regras de desenho poderão ser verificadas.

4.3.3.4 Conversor de Formato Leiaute RS p/ Formato Leiaute CIF

Este software é um outro conversor de arquivos de descrição de circuitos. A partir de um arquivo descrito em formato RS, formato de descrição de leiaute utilizado na UFRGS, cria-se uma versão do mesmo circuito em formato CIF, formato de descrição de leiaute padrão.

4.3.3.5 Extrator Elétrico

Este aplicativo foi desenvolvido com o intuito de se obter um arquivo de dados com a lista de transistores, em formato compatível com o simulador SPICE [VLA 81], a partir da descrição geométrica das máscaras (conjunto de retângulos) do circuito integrado, juntamente com um arquivo que descreve as características da tecnologia. O EXTRIBO [STE 89] [STE 89a] [STE 90] lê a descrição do leiaute do circuito, no

formato RS juntamente com um arquivo de descrição da tecnologia e gera uma lista de transistores que pode servir como dados de entrada para o simulador SPICE. Este programa além de poupar o trabalho da elaboração do arquivo de simulação, também é útil para detectar alguns tipos de erros de leiaute. O programa EXTRIBO gera uma saída gráfica do leiaute do circuito, mostrando no desenho os números de nós atribuídos pelo extrator.

Os algoritmos para a identificação das características geométricas das células nesta versão baseiam-se na pesquisa seqüencial de uma lista encadeada de retângulos, classificada por níveis e ordenada segundo uma das coordenadas. A obtenção de uma lista de transistores envolve basicamente dois procedimentos: a avaliação da conectividade e a identificação dos transistores.

4.3.3.6 Planificador (Expansor)

Este programa cria uma descrição plana de um circuito, através da conversão de um circuito descrito em leiaute hierárquico.

4.3.3.7 Compactador - SYLC

O compactador [SOT 94] parte de um leiaute simbólico, gerado pelo Editor Simbólico ou pelo TROPIC, e gera um leiaute em nível de máscara. Sendo assim, o compactador é independente de regras de projeto para a família tecnológica CMOS.

O SYLC trabalha sobre um grafo de restrições montado a partir da posição dos elementos simbólicos da célula. As restrições são geradas de acordo com um arquivo de tecnologia, que contém as regras. A saída do compactador é um leiaute em nível de máscara compactado e livre de erros (em formato leiaute RS).

Também como saída do compactador obtém-se um arquivo que especifica a interface da célula compactada, contendo as posições e larguras dos terminais dessa célula.

Resumidamente, o compactador de células-folha SYLC é de uma dimensão (1D) e trabalha em três etapas principais:

- Geração do grafo de restrições: um grafo representando todas as restrições de afastamento e conectividade do leiaute do circuito (para X ou Y) é montado de acordo com a posição dos elementos simbólicos e conforme as regras de projeto;
- Compactação do grafo: um algoritmo para encontrar o caminho mais longo (longest path) é utilizado para compactar o grafo;
- Minimização de fios (*Wire Length Minimization*): um algoritmo heurístico para minimização dos fios e conseqüente redução de resistências e capacitâncias parasitas é utilizado sobre o grafo já compactado.

Essas etapas são executadas sobre o circuito alternadamente em X e Y até que não seja mais possível compactá-lo.

4.3.3.8 Extrator Lógico (Extraló)

Visa auxiliar o projetista na etapa de verificação do projeto. Concluídas as máscaras, utiliza-se o extrator elétrico - EXTRIBO - para obter-se o *netlist* SPICE do circuito. Obtido o *netlist* a nível de transistores, chama-se o EXTRALO [MOR 89] [MOR 90c] para obter-se a descrição a nível de portas lógicas do circuito, isto é, extrai

do *netlist Spice* todas as células básicas contidas, como: inversores, *nands*, *nors*, *transmission-gates* e portas complexas (AOI). Esta passagem do nível elétrico para lógico facilita a verificação, além de ser uma forma mais natural de "visualizar" o circuito.

A figura 4.18 apresenta a interface gráfica do Extrator Lógico, desenvolvida para ambiente MS-WINDOWS™.

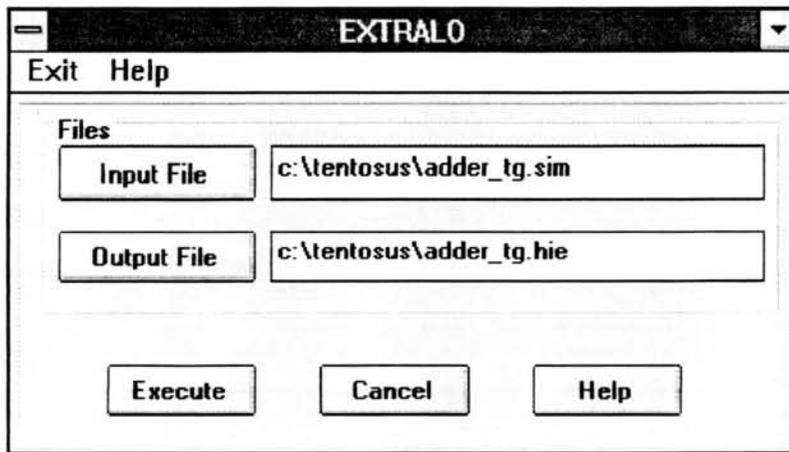


FIGURA 4.18 - Interface do Extrator Lógico (EXTRALO).

4.3.3.9 Comparador de NetLists - COTONET

Trata-se de uma ferramenta que permite um eficiente teste para a verificação do leiaute de um projeto de circuito VLSI, pois possibilita a comparação do *netlist* gerado da representação esquemática do circuito com o *netlist* extraído a partir do leiaute. Inevitavelmente, as duas descrições empregam nomes diferentes para os nós e transistores do circuito e listam estes elementos em diferentes ordens. O COTONET [FRE 92] testa se as duas *netlists* representam o mesmo circuito e produz um mapeamento que associa cada objeto em uma *netlist* com o seu correspondente na outra. O programa informará as eventuais diferenças entre os *netlists*.

A figura 4.19 apresenta a interface gráfica do Comparador de NetLists, desenvolvida para ambiente MS-WINDOWS™.

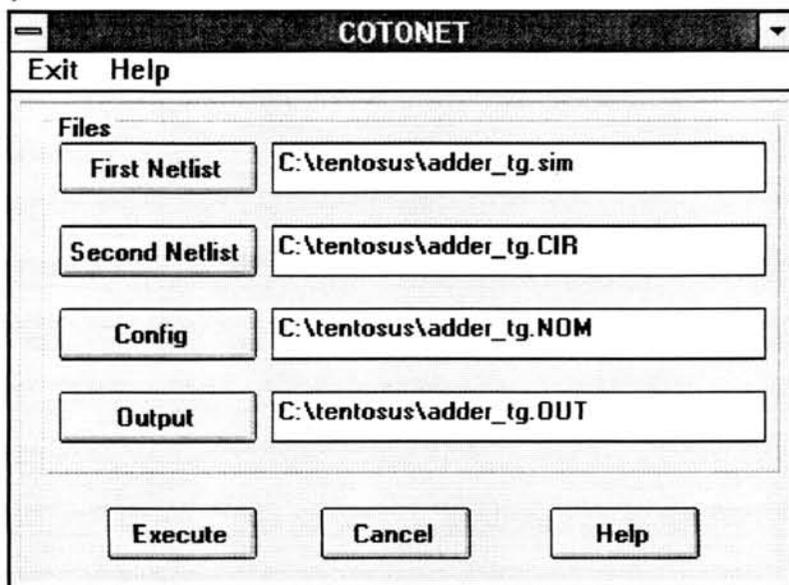


FIGURA 4.19 - Interface do Comparador de NetLists (COTONET).

4.3.4 Menu *Simuladores*

4.3.4.1 Simulador Spice

Trata-se de um simulador a nível elétrico (transistores, capacitores e resistências) [VLA 81]. Para a sua execução é necessário um equipamento com coprocessador numérico em PCs 386. Nas máquinas atuais (PCs 486, Pentium) o coprocessador está embutido na CPU.

4.4 Executando os aplicativos

Os aplicativos são executados de dois modos distintos:

- Com o surgimento, antes da execução, de uma caixa de diálogo onde serão apresentados os parâmetros necessários à execução da ferramenta, vide figura 4.20;
- Execução direta, sem mostrar a caixa de diálogo.

Convém salientar que o surgimento, ou não, desta caixa de diálogo será definido na opção de menu *Arquivo - Configuração - Janela de Parâmetros*, onde o usuário habilitará/desabilitará o surgimento da janela de parâmetros no momento de execução da ferramenta. Portanto a execução desta janela de parâmetros independe da ferramenta.

4.4.1 Execução com Caixa de Diálogo

A figura 4.20 apresenta a caixa de diálogo onde serão apresentados os dados a serem passados como parâmetros no momento de execução das ferramentas.

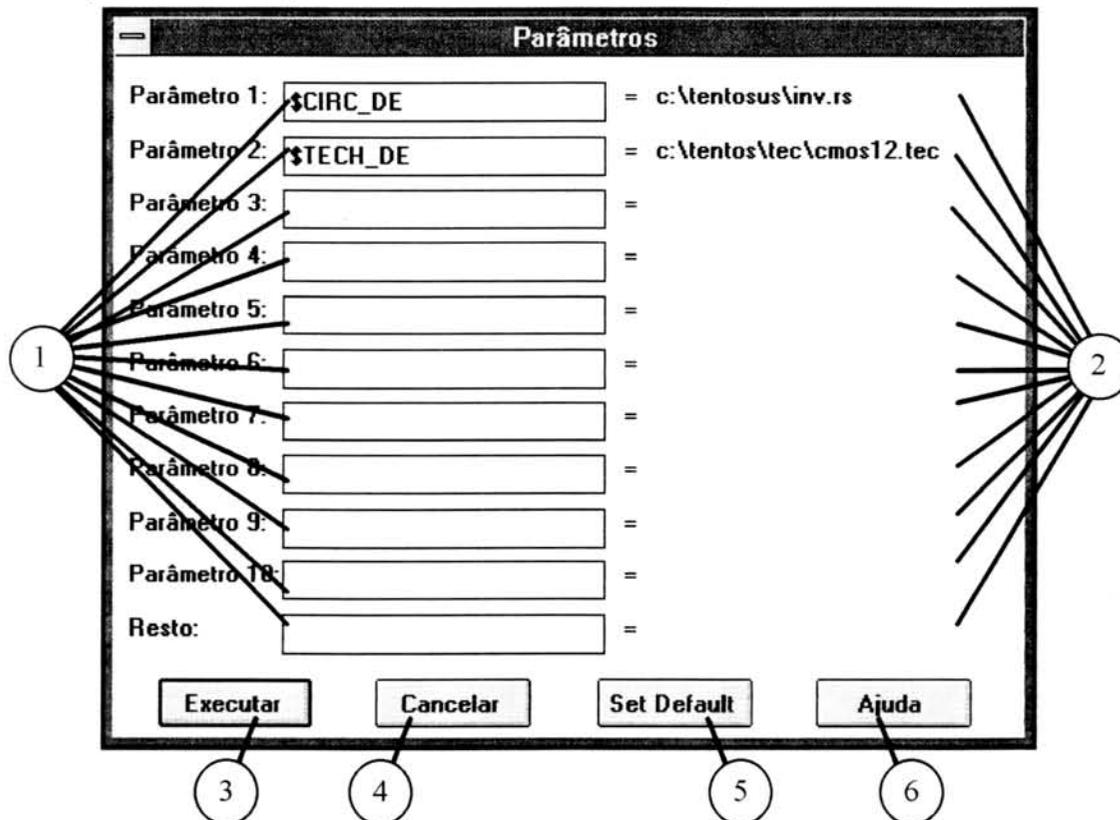


FIGURA 4.20 - Janela de Apresentação dos Parâmetros antes da Execução das Ferramentas

A tabela 4.9 apresenta a descrição de cada um dos componentes da caixa de diálogo apresentada na figura 4.20.

TABELA 4.9 - Lista dos elementos pertencentes a Janela de Parâmetros

Campo	Descrição
1	Indica o formato que os parâmetros devem estar gravados no arquivo de inicialização. Estes campos podem ser alterados.
2	Se o projetista utilizar variáveis (por exemplo: \$TECH_DE, \$CIRC_DE) no interior do campo '1', será apresentado neste campo o significado de cada uma destas variáveis, no presente momento. Caso o dado do campo '1' seja uma constante, o campo '2' terá a mesma informação que no campo '1'.
3	Executa a ferramenta com os parâmetros apresentados nos campos '2'.
4	Cancela a execução da ferramenta.
5	Grava no arquivo de inicialização os dados apresentados nos campos '1' como parâmetros da ferramenta.
6	Aciona um <i>Help</i> referente a esta caixa de diálogo.

4.4.2 Execução Direta

Neste caso o projetista executará a ferramenta tal como ela foi especificada no item de menu *Instalar - Ferramentas*. Não será possível visualizar que parâmetros estão sendo passados para a ferramenta.

Para os projetistas já acostumados com a estrutura do Sistema TENTOS, esta forma agilizará o uso do Gerenciador com as suas ferramentas.

4.5 Conclusão

O pacote TENTOS possui um grande potencial para tornar-se bastante utilizado tanto em instituições de ensino e pesquisa, nacionais ou estrangeiras, como em indústrias que trabalham com microeletrônica, pois trata-se de uma ferramenta de fácil operação e de baixo custo.

Uma característica que orientou este trabalho foi a de tornar a ferramenta mais ágil, que não seja tão sensível às mudanças tecnológicas que estão sempre ocorrendo. Com a proposta de tornar o TENTOS um gerenciador aberto, em uma versão para MS-WINDOWS™, dota-se o sistema de recursos que permitirão a incorporação de novas ferramentas e novos arquivos de tecnologia, sem a necessidade de ter que manipular/alterar o código fonte do TENTOS, para cada nova ferramenta que venha a ser incorporada ao sistema.

5 Implementações Futuras

O sistema TENTOS em sua versão atual pode ser utilizado para realizar todas as etapas de um projeto do leiaute de um circuito, a partir de uma descrição *Spice*, sem ter que trocar de ambiente de projeto.

Observa-se no TENTOS que algumas ferramentas que integram o pacote deveriam possuir uma versão mais atualizada, pois:

- Com a troca do sistema operacional base, de MS-DOS™ para MS-WINDOWS™, novas técnicas de programação poderiam ser utilizadas, os programas poderiam ser mais poderosos, por exemplo, fazer um melhor uso da memória;
- Criar interfaces em MS-WINDOWS™ para pequenos programas que rodam em DOS (por exemplo: conversor de formato RS↔CIF, planificador, verificador de regras de projeto), tal como é feito com a ferramenta de síntese TRAGO.

Algumas tarefas também poderiam ser elaboradas, em relação ao gerenciador propriamente dito, tais como dotar o TENTOS de um gerador de relatórios, gerar automaticamente a conversão dos formatos de arquivos.

Através da implementação de novas versões dos editores de esquemático e de leiaute seriam fornecidas ao usuário maiores facilidades, principalmente no que se refere à parte de editoração gráfica dos circuitos. A implementação de um gerador de relatórios forneceria ao usuário uma poderosa ferramenta que emitiria relatórios onde seriam listados todos os arquivos intermediários criados a partir da execução das ferramentas.

Todas estas novas implementações poderão explorar ao máximo os novos recursos que foram introduzidos pelo MS-WINDOWS™, em comparação com o MS-DOS™, recursos estes que são apresentados na seção a seguir.

Na seção subsequente serão apresentadas com melhores detalhes e motivos algumas sugestões para serem implementadas, num futuro próximo, a fim de dotar o TENTOS de um ambiente mais robusto para o projeto de circuitos integrados.

5.1 Recursos Fornecidos pelo MS-WINDOWS™

Pode-se discutir a validade da inclusão desta seção neste capítulo, porém deve-se considerar que muitas dos trabalhos futuros estarão orientados pelos novos recursos que são propiciados pelo MS-WINDOWS™ [CRA 94][HOL 94][PET 93][WEB 96]. Nas subseções a seguir serão detalhados estes recursos.

5.1.1 Gerenciador de Impressão

Quando da implementação de programas em MS-DOS™, o programador tinha que se preocupar em criar *drivers* de impressão para cada impressora que pudesse ser utilizada pelo usuário. Agora, com o conceito de Gerenciador de Impressão Externo, embutido no MS-WINDOWS™, o programador só se preocupa com impressão uma vez, pois os *drivers* para as impressoras são fornecidos pelo MS-WINDOWS™, o programa torna-se independente de qual impressora será utilizada.

5.1.2 DDE (Intercâmbio Dinâmico de Dados)

Uma das vantagens do ambiente operacional MS-WINDOWS™ é que ele é multitarefa; ou seja, ele pode rodar várias aplicações diferentes ao mesmo tempo. Esta possibilidade permite que dois programas sejam capazes de se comunicar entre si, em tempo de execução, passando dados de um para o outro, por exemplo. Usando DDE pode-se coordenar a transferência de dados entre uma série de aplicações do MS-WINDOWS™. Por exemplo, dois programas MS-WINDOWS™ dialogam entre si enviando mensagens um ao outro. Esses dois programas são conhecidos como *servidor* e *cliente*.

Existem duas maneiras de abrir uma ligação entre programas, como uma ligação automática ou como uma ligação manual. Em uma ligação automática os dados são atualizados no cliente sempre que ele é alterado no servidor. Já na ligação manual os valores só são atualizados se solicitados. De um modo geral o fluxo das informações é do servidor para o cliente, ver figura 5.1, embora seja possível que o cliente também envie dados de volta.

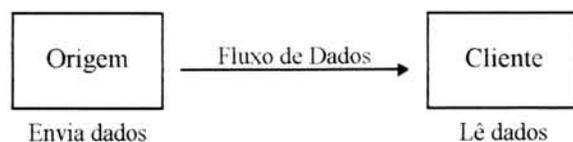


Figura 5.1 - Fluxo de dados com comunicação via DDE

Está baseado no sistema de mensagens embutido do MS-WINDOWS. Dois programas WINDOWS dialogam entre si enviando mensagens um ao outro. Esses dois programas são conhecidos como *servidor* e *cliente*.

5.1.3 OLE (Incorporação e Vinculação de Objetos)

Através da incorporação e vinculação de objetos, as informações podem ser transferidas e compartilhadas entre diversos aplicativos.

Com a incorporação de objetos, um objeto pode ser qualquer tipo de informação (texto, desenho, som, ...), o usuário pode aproveitar um recurso já existente noutra ferramenta para a manipulação de dados no seu aplicativo *mestre*.

Utilizando-se da vinculação de objetos, pode-se utilizar a mesma informação de um objeto em diversos documentos diferentes. Este processo traz uma grande economia de tempo, pois uma vez feita alteração em um objeto, ele se alterará, automaticamente, nos demais.

5.1.4 DLL (Bibliotecas de Ligações Dinâmicas)

As bibliotecas de ligações dinâmicas são um dos mais importantes elementos estruturais do MS-WINDOWS™. A maior parte dos arquivos em disco associados ao MS-WINDOWS™ são módulos do programa ou módulos da biblioteca de ligações dinâmicas. Estas bibliotecas são arquivos independentes que contêm funções que podem ser chamadas por programas e outras DLL para executar determinadas tarefas. Uma biblioteca de ligações dinâmicas é trazida à atividade quando outro módulo chama uma das funções na biblioteca.

O termo *ligação dinâmica* refere-se ao processo que o MS-WINDOWS™ usa para ligar a chamada a uma função em um módulo à função verdadeira no módulo da biblioteca. *Ligação estática* ocorre quando executa-se o LINK para criar um arquivo

.EXE a partir de vários módulos objetos (.OBJ) e arquivos de biblioteca (.LIB). A ligação dinâmica ocorre durante a execução.

5.1.5 Área de Transferência

A área de transferência do MS-WINDOWS™ permite que os dados sejam transferidos de um programa para outro. Esta transferência se dá através da permuta de blocos de memória entre os programas.

O seu mecanismo é relativamente simples, seja no programa que passa os dados, seja no que mais tarde vai receber.

Quando o usuário insere dados que estão na área de transferência, o programa tem que determinar se o formato dos dados pode ser aceito.

5.1.6 Demais recursos

Além dos recursos comentados acima, o MS-WINDOWS™ fornece aos desenvolvedores de solução *for Windows* outros recursos, tais como:

- *Maior naturalidade no uso da memória estendida e expandida:* apesar de ser possível o desenvolvimento de programas em MS-DOS™ que utilize estas memórias, eram poucos os programadores com conhecimento para tanto, tal era a complexidade de desenvolvimento;
- *Multitarefa:* recurso permitido apenas em MS-WINDOWS™. A troca de informações via DDE é uma consequência direta deste recurso;
- *Telas gráficas:* no MS-WINDOWS™ todas as telas possuem uma apresentação gráfica. Isto gera uma melhor apresentação dos programas. Outra fato é que em MS-WINDOWS™ não existe as *piscadas de tela*, comuns quando da troca de telas texto para telas gráficas nos programas em MS-DOS™;
- *Drivers:* nas aplicações para MS-DOS™ o programador tinha que se preocupar com tudo, que monitor iria poder ser utilizado para o programa, que mouse, leiaute do teclado,... . Já na programação para MS-WINDOWS™ todas estas tarefas são executadas pelo MS-WINDOWS™, o programador só tem que se dedicar ao seu programa, a configuração do ambiente é gerenciada pelo MS-WINDOWS™.

5.2 Próximos Passos

A seguir serão apresentados algumas sugestões de trabalhos futuros que poderão ser desenvolvidos.

5.2.1 Editor de Esquemáticos

Conforme análise realizada, em [MÄH 94] [MÄH 95], a respeito do editor de esquemáticos ESQUELETO, [GME 92] [SAC 90], que hoje se encontra incorporado ao TENTOS, constataram-se vários motivos que levariam a se desenvolver um editor de esquemáticos para MS-WINDOWS™, são eles:

- *Uso da memória estendida:* O MS-WINDOWS™ permite um gerenciamento transparente da memória estendida, deste modo será possível a manipulação de circuitos com um grande número de portas lógicas;

- *Interação entre ferramentas:* Como a forma de execução de ferramentas MS-DOS™ em ambiente MS-WINDOWS™ é muito simples, não existe atualmente nenhuma interação de dados entre a ferramenta e o gerenciador, isto é, caso seja necessária uma melhor interação entre as ferramentas, com o formato atual isto não é possível;
- *Novo ambiente de trabalho:* A necessidade de uma evolução da ferramenta para um novo ambiente de trabalho, o MS-WINDOWS™, que ofereça várias vantagens adicionais em relação ao meio anterior, ou seja, a migração do software para uma nova tecnologia disponível.
- *Interface gráfica:* Ferramenta desenvolvida para ambiente MS-DOS™ com interface gráfica, com intensa interação com o usuário. Em MS-WINDOWS™ interface gráfica é uma conseqüência natural de programação, o que não acontece em MS-DOS™.
- *Providências pré-uso:* Desobrigar o projetista de ter que tomadas medidas prévias antes da execução do MS-WINDOWS™ a fim de poder utilizar na totalidade os recursos da versão atual, por exemplo, o *driver* de uso do mouse deve ser instalado antes da execução do ESQUELETO, portanto antes que o MS-WINDOWS™ seja executado.
- *Utilidade de um editor de esquemáticos:* Trata-se de um programa de grande utilidade para o projeto de circuitos, pois é o único meio gráfico para projeto de circuitos a partir do nível lógico-estrutural disponível no sistema.
- *Passagem de parâmetros:* Na versão atual, não é possível passar o nome do circuito que deverá ser editado, pois o ESQUELETO não foi feito para receber esta informação, deste modo, sempre que o projetista for utilizar este editor ele terá que indicar o nome do circuito, que será editado, através de um item de menu do editor ESQUELETO.

Deste modo, torna-se recomendável a criação de uma versão MS-WINDOWS™ para o editor de esquemáticos. Deve-se observar que esta nova versão não deverá ser obrigatoriamente igual à versão atual, e sim uma versão em que todos os recursos oferecidos pelo MS-WINDOWS™ deverão ser explorados, bem como também poderão ser agregados novos recursos ao programa. Isto é, recomenda-se que se construa um novo editor de esquemáticos, em versão para MS-WINDOWS™, que possua todos os recursos oferecidos pelo ESQUELETO e tudo o mais que for útil que não conste da versão atual (por exemplo: permitir ao usuário a impressão gráfica do circuito).

5.2.2 Editor de Leiautes

Apesar da existência de dois editores de leiaute incorporados ao TENTOS, o EMA, [STE 89b] [STE 89c], e o SELA, [CAA 91], constatou-se a necessidade de se desenvolver um editor de leiautes para MS-WINDOWS™, não tão complexo quanto o SELA, porém propiciando ao usuário TENTOS uma forma de editar/visualizar o seu leiaute através de um editor gráfico desenvolvido para MS-WINDOWS™. Desta forma, todos os motivos apresentados para a realização de um novo editor de esquemáticos, com exceção do último, respaldam a necessidade de desenvolver um novo editor de leiautes para MS-WINDOWS™.

Portanto, torna-se recomendável a criação de uma versão MS-WINDOWS™ para o editor de leiautes. Deve-se observar que esta nova versão não será tão complexa quanto é o editor de leiautes SELA, mas deverá ter no mínimo os mesmos recursos que

são oferecidos pelo EMA. isto é, propiciar ao usuário TENTOS uma forma de editar/visualizar o seu leiaute através de um editor gráfico, de fácil utilização. Neste novo editor de leiautes dever-se-á tomar cuidado para que todos os recursos oferecidos pelo MS-WINDOWS™ sejam explorados.

Algumas melhorias, já relacionadas [DOS 91], deverão ser implementadas neste novo editor de leiautes, como por exemplo:

- Não permitir a gravação de retângulos com dimensões nulas;
- Permitir a edição de polígonos a 45°.
- Que os passos de grade sejam proporcionais às dimensões da tecnologia que se está utilizando;
- Informar na tela o nome do arquivo e da célula que estão sendo editados;
- Escrever o nome da célula dentro de seu retângulo envolvente quando esta não for expandida.

Outro recurso que também deverá ser fornecido ao projetista é no que se refere à impressão gráfica do leiaute do circuito.

5.2.3 Gerador de Relatórios

A geração de uma documentação que contenha os passos gerados e os resultados obtidos na elaboração do projeto. Desta forma será fornecida uma ferramenta poderosa no que se refere à documentação de projetos de circuitos integrados. Esta ferramenta deverá ser projetada de modo que monitore todas as etapas executadas no projeto, armazenando as descrições iniciais, intermediárias e finais do circuito. A implementação deste gerador de relatórios também fornecerá novos recursos para usuário, por exemplo: Facilidade para a busca por suporte visando a solução de problemas que ocorram.

5.2.4 Conversão Automática de Formatos

Permitir que a conversão de formatos dos arquivos seja feita de forma automática no sistema. No momento em que o usuário executar uma ferramenta o seu resultado será convertido automaticamente para os demais formatos existentes no sistema, observando-se a compatibilidade dos formatos.

A implementação deste recurso dotaria o TENTOS de uma característica bem própria de *frameworks*. Com esta facilidade sendo transparente para o usuário, o projeto ganha em agilidade e reduz margens de erros no encaminhamento do projeto.

5.2.5 Arquivo de Projeto

Na versão corrente, o TENTOS possui um arquivo de projeto com informações básicas do projeto.

Seria interessante que este arquivo de projeto fosse o projeto, propriamente dito. Este arquivo iria conter todas as descrições de circuito. O projetista iria editar o projeto a partir dele. Com a utilização de OLE, o usuário poderia disparar procedimentos do próprio documento.

Um arquivo de projetos com estas características facilitaria bastante na implementação do gerador de relatórios.

5.3 Conclusão

Com as implementações sugeridas, levando-se em conta os recursos oferecidos pelo MS-WINDOWS™, o sistema TENTOS tornar-se-ia mais homogêneo, pois com a utilização de editores de esquemáticos e de leiautes *for* Windows o usuário realizaria toda a sua editoração em um mesmo ambiente de trabalho, sem alterações de MS-WINDOWS™→MS-DOS™→MS-WINDOWS™. Isto sem falar no fato que tais editores são ferramentas gráficas e o MS-WINDOWS™ é um ambiente gráfico, portanto muito mais adequado para este tipo de aplicação. Já o desenvolvimento de interfaces em MS-WINDOWS™ para as ferramentas sem interface tornaria o trabalho dos projetistas mais amigável do que é hoje e a implementação de um gerador de relatórios forneceria ao usuário um recurso de grande utilidade prática.

Através da utilização dos protocolos de comunicação existentes no MS-WINDOWS™ seriam implemetadas ferramentas para serem executadas em paralelo com outras ferramentas. Nestes processos poderiam estar sendo trocadas informações entre os programas que gerariam novos resultados.

6 Projeto de um Circuito Integrado Utilizando o Sistema TENTOS

Este capítulo tem por objetivo mostrar exemplos de utilização do Sistema TENTOS. Deste modo serão apresentados dois exemplos onde são utilizadas ferramentas de síntese automática de leiaute e um tutorial onde é apresentado um exemplo de concepção de um circuito integrado através de um processo manual.

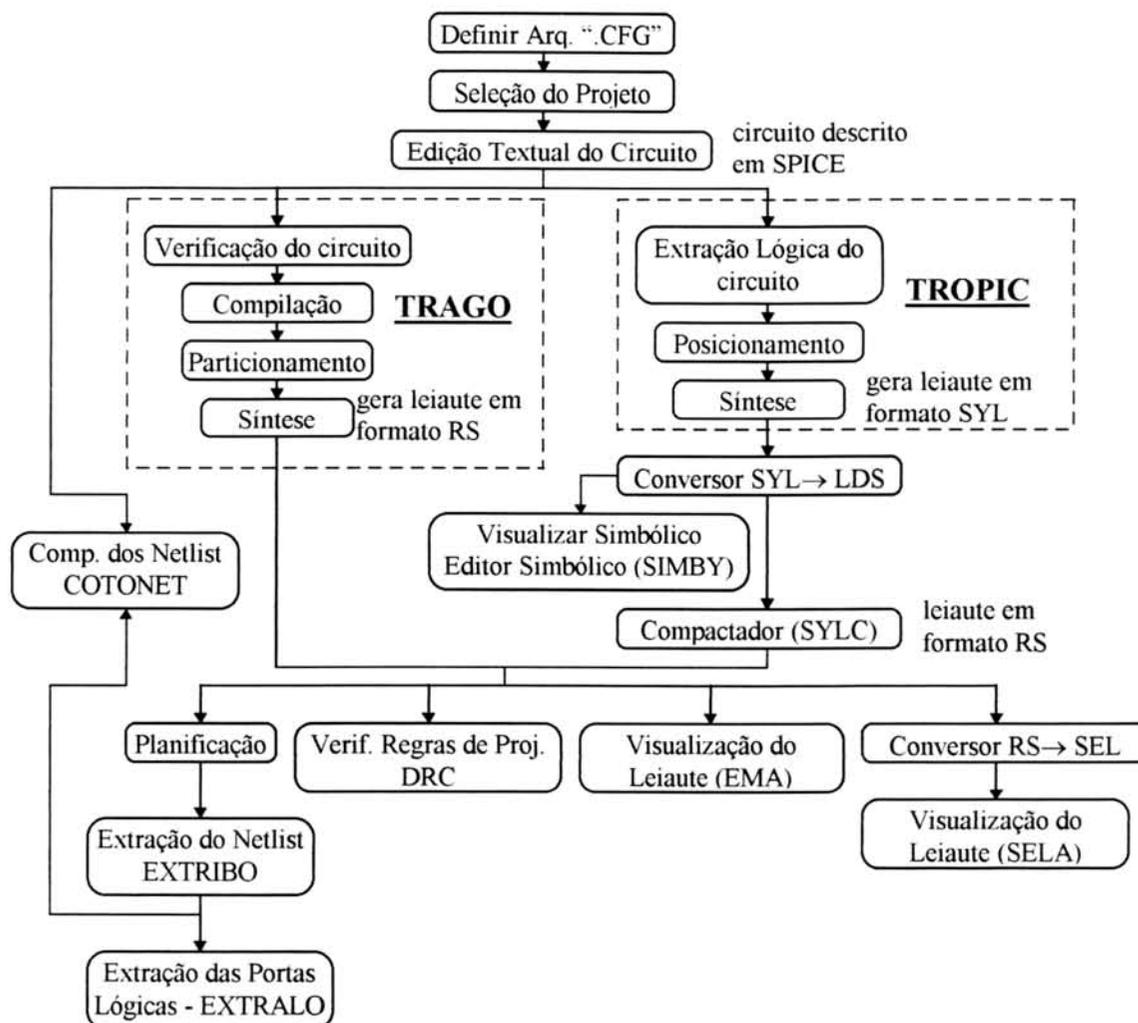


FIGURA 6.1 - Fluxo das etapas de projeto com geração automática do leiaute

Todas as ferramentas envolvidas nestes exemplos encontram-se disponíveis na versão *standard* do Gerenciador. A figura 6.1 apresenta o fluxo das etapas de projeto a serem cumpridas nos tutoriais, de síntese automática do leiaute, que serão apresentados a seguir.

Os exemplos de uso do Sistema TENTOS, que serão apresentados a seguir, terão as suas informações configuradas tendo como referência a instalação do Sistema TENTOS no diretório C:/TENTOS.

Por último será apresentado um exemplo de uso do sistema TENTOS considerando-se a implementação dos trabalhos futuros sugeridos.

6.1 Projeto de um Circuito Utilizando o Sistema TRAGO

Este exemplo tem por objetivo apresentar a síntese de um circuito utilizando a ferramenta TRAGO.

Será utilizado como circuito exemplo um somador cujas características estão apresentadas na figura 6.2.

Arquivo:	C:\TENTOSUS\ADDER_TG.SIM																																													
Descrição:	Somador completo de 2 bits implementado através de NANDs, NORs e INVERSORES																																													
Diagrama Funcional:																																														
Tabela verdade:	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Cin</th> <th>Si</th> <th>Cout</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Cin	Si	Cout	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1
A	B	Cin	Si	Cout																																										
0	0	0	0	0																																										
0	0	1	1	0																																										
0	1	0	1	0																																										
0	1	1	0	1																																										
1	0	0	1	0																																										
1	0	1	0	1																																										
1	1	0	0	1																																										
1	1	1	1	1																																										

FIGURA 6.2 - Descrição do circuito c:\tentosus\adder_tg.sim

A seguir serão apresentados os passos necessários para a realização deste processo.

1. Escolha do arquivo de configuração;

Ir no menu **Arquivo - Configuração - Lê “.CFG”**

Selecionar o arquivo C:\TENTOSUS\TENTOSUS.CFG

2. Seleção do Projeto que será manipulado

Ir no menu **Arquivo - Projeto** ou na barra de ferramentas e clicar sobre o ícone que representa **Projeto**

Selecionar o arquivo de c:\tentosus\adder_tg.prj

Este arquivo já existe, e o seu conteúdo é o seguinte:

Circuito: c:\tentosus\adder_tg.sim
Tecnologia: CMOS12

A seguir encontra-se a descrição *spice* do circuito c:\tentosus\adder_tg.sim.

```
.subckt nor2 i1 i2 out vcc
MP1 _2 i1 vcc vcc PMOS L=1.2U W=5.0U
MP2 out i2 _2 vcc PMOS L=1.2U W=5.0U
MN3 out i1 0 0 NMOS L=1.2U W=5.0U
MN4 out i2 0 0 NMOS L=1.2U W=5.0U
.ends nor2
```

```
.subckt nor3 i1 i2 i3 out vcc
MP1 _2 i1 vcc vcc PMOS L=1.2U W=5.0U
MP2 _3 i2 _2 vcc PMOS L=1.2U W=5.0U
MP3 out i3 _3 vcc PMOS L=1.2U W=5.0U
MN4 out i1 0 0 NMOS L=1.2U W=5.0U
MN5 out i2 0 0 NMOS L=1.2U W=5.0U
MN6 out i3 0 0 NMOS L=1.2U W=5.0U
.ends nor3
```

```
.subckt nand2 i1 i2 out vcc
MP1 out i1 vcc vcc PMOS L=1.2U W=5.0U
MP2 out i2 vcc vcc PMOS L=1.2U W=5.0U
MN3 out i1 _2 0 NMOS L=1.2U W=5.0U
MN4 _2 i2 0 0 NMOS L=1.2U W=5.0U
.ends nand2
```

```
.subckt nand3 i1 i2 i3 out vcc
MP1 out i1 vcc vcc PMOS L=1.2U W=5.0U
MP2 out i2 vcc vcc PMOS L=1.2U W=5.0U
MP3 out i3 vcc vcc PMOS L=1.2U W=5.0U
MN4 out i1 _4 0 NMOS L=1.2U W=5.0U
MN5 _4 i2 _2 0 NMOS L=1.2U W=5.0U
MN6 _2 i3 0 0 NMOS L=1.2U W=5.0U
.ends nand3
```

```
.subckt inv in out vcc
MP1 out in vcc vcc PMOS L=1.2U W=5.0U
MN2 out in 0 0 NMOS L=1.2U W=5.0U
.ends inv
```

```
X1 a b m1 vcc nor2
X2 m1 m2 vcc inv
X3 m2 c m3 vcc nand2
X4 m3 m4 carry vcc nand2
X5 a b m4 vcc nand2
X6 a b c m5 vcc nor3
X7 a b c m8 vcc nand3
X8 carry m5 m6 vcc nor2
X9 m6 m7 vcc inv
X10 m7 m8 sum vcc nand2
```

```
*interface: a * orientacao=O *
*interface: b * orientacao=O *
*interface: c * orientacao=O *
*interface: carry * orientacao=L *
*interface: sum * orientacao=L *
.end
```

Caso o arquivo de projeto não existisse, o usuário após especificar o nome do projeto deveria especificar o circuito e o arquivo de tecnologia que serão manipulados.

3. Manipulação do arquivo com a descrição *spice* do circuito

Ir no menu **Descrições - Textual Edition**

Para este exemplo será apresentada a interface de edição textual do circuito ilustrada pela figura 5.3.

```

Bloco de Notas - ADDER_TG.SIM
Arquivo Editar Localizar ?
|.subckt nor2 i1 i2 out vcc
MP1 _2 i1 vcc vcc PMOS L=1.2U W=5.0U
MP2 out i2 _2 vcc PMOS L=1.2U W=5.0U
MN3 out i1 0 0 NMOS L=1.2U W=5.0U
MN4 out i2 0 0 NMOS L=1.2U W=5.0U
.ends nor2

.subckt nor3 i1 i2 i3 out vcc
MP1 _2 i1 vcc vcc PMOS L=1.2U W=5.0U
MP2 _3 i2 _2 vcc PMOS L=1.2U W=5.0U
MP3 out i3 _3 vcc PMOS L=1.2U W=5.0U
MN4 out i1 0 0 NMOS L=1.2U W=5.0U
MN5 out i2 0 0 NMOS L=1.2U W=5.0U
MN6 out i3 0 0 NMOS L=1.2U W=5.0U
.ends nor3

.subckt nand2 i1 i2 out vcc
MP1 out i1 vcc vcc PMOS L=1.2U W=5.0U
MP2 out i2 vcc vcc PMOS L=1.2U W=5.0U
MN3 out i1 _2 0 NMOS L=1.2U W=5.0U
MN4 _2 i2 0 0 NMOS L=1.2U W=5.0U
.ends nand2

.subckt nand3 i1 i2 i3 out vcc
MP1 out i1 vcc vcc PMOS L=1.2U W=5.0U
MP2 out i2 vcc vcc PMOS L=1.2U W=5.0U

```

FIGURA 6.3 - Janela de Edição Textual com o arquivo *c:\tentosus\adder_tg.sim*

Se o circuito já estiver correto esta é uma fase que pode ser suprimida. No entanto, caso falem informações no circuito, ou o circuito ainda não exista, esta é a etapa onde tais informações deverão ser acrescentadas à descrição *SPICE* do circuito.

4. Geração do processo de síntese utilizando a ferramenta TRAGO

Ir no menu **Sintetizar - TRAGO**

A figura 6.4 apresenta a interface do Sistema TRAGO com os parâmetros apropriados.

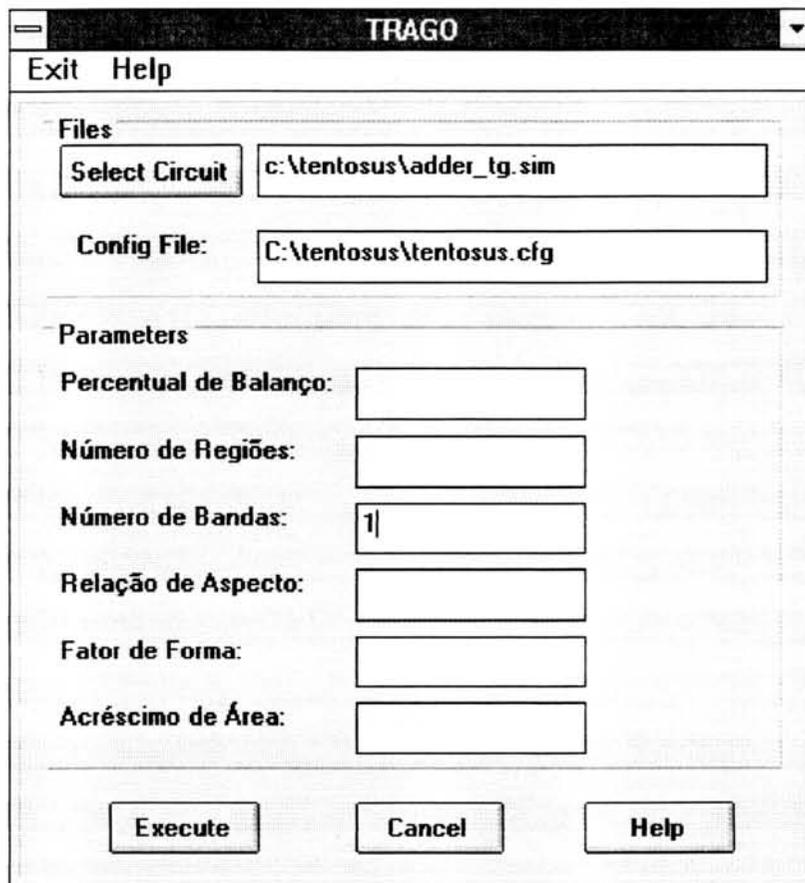


FIGURA 6.4 - Sistema TRAGO com os parâmetros certos para a geração do processo de síntese do arquivo *c:\tentosus\adder_tg.sim*

Para a execução do sistema TRAGO apenas 3 dados podem ser configurados internamente à ferramenta, são eles:

- Nome do circuito;
- Nome do arquivo de configuração;
- Número de Bandas.

As informações, apresentadas na janela do Sistema TRAGO, percentual de balanço, número de regiões, relação de aspecto, fator de forma e acréscimo de área tem os seus valores constantes.

No momento em que o projetista selecionar o botão *Execute* o TRAGO executará os seguintes passos:

- a) Preparação do circuito

Nesta etapa, vide figura 6.5, do Sistema TRAGO é feita a verificação do circuito.

```

C:\TENTOS\BIN>trago.exe c:\tentosus\adder_tg.sim C:\tentosus\tentosus.cfg -1

Arquivo de entrada: adder_tg

Gerando: nor2
numero de celulas: 1
celulas_posicionadas:1 Trilhas:2 Minimo: 2

C:\TENTOS\BIN>pause
Pressione qualquer tecla para continuar . . .

```

FIGURA 6.5 - Sistema TRAGO - preparação do circuito *c:\tentosus\adder_tg.sim*

b) Compilação do circuito

Nesta etapa, vide figura 6.6, do Sistema TRAGO é feita a preparação do *netlist SPICE* para o particionador, inserindo informações relativas a área e a posição dos pinos das células.

```

C:\TENTOS\BIN>cs.exe adder_tg adder_tg C:\tentosus\tentosus.cfg

*** COMPILADOR SPICE***
Entrada = \TENTOSUS\adder_tg.sim
Saída = \TENTOSUS\adder_tg.spic

Execucao OK

C:\TENTOS\BIN>pause
Pressione qualquer tecla para continuar . . .

```

FIGURA 6.6 - Sistema TRAGO - compilação do circuito *c:\tentosus\adder_tg.sim*

c) Particionamento do circuito

Nesta etapa, vide figura 6.7, do Sistema TRAGO é feita a divisão do circuito em bandas, atribuindo-se células para cada uma.

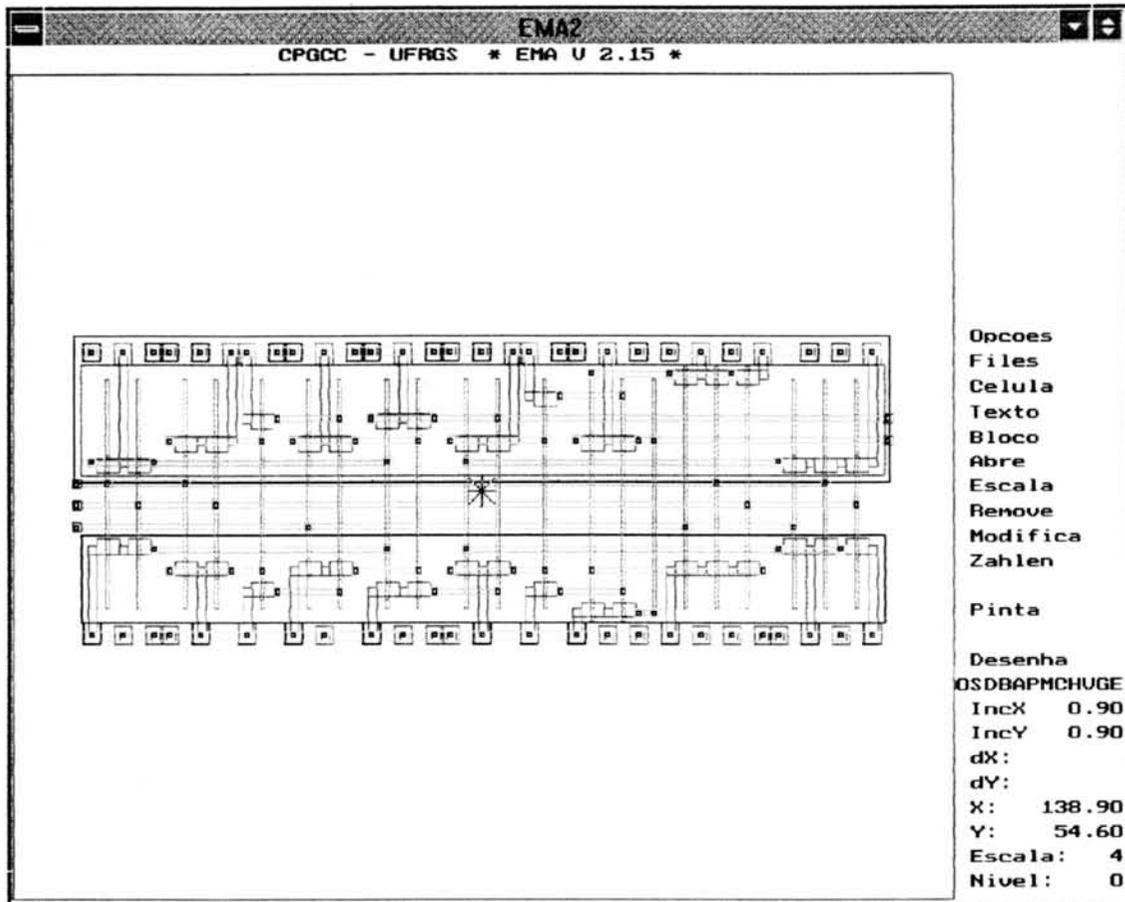


FIGURA 6.9 - Leiaute do circuito *c:\tentosus\adder_tg.sim* visualizado pelo EMA

b) Através do Editor de Leiautes - SELA

Ir no menu **Ferramentas - Conversor SEL↔RS**

A execução do conversor de formatos é necessária para a visualização através do Editor de Leiautes - SELA, pois o Sistema TRAGO gera um leiaute do circuito em formato Leiaute RS e o SELA visualiza circuitos descritos através de um formato próprio, o formato de SEL.

A figura 6.10 apresenta a interface do conversor de formatos.

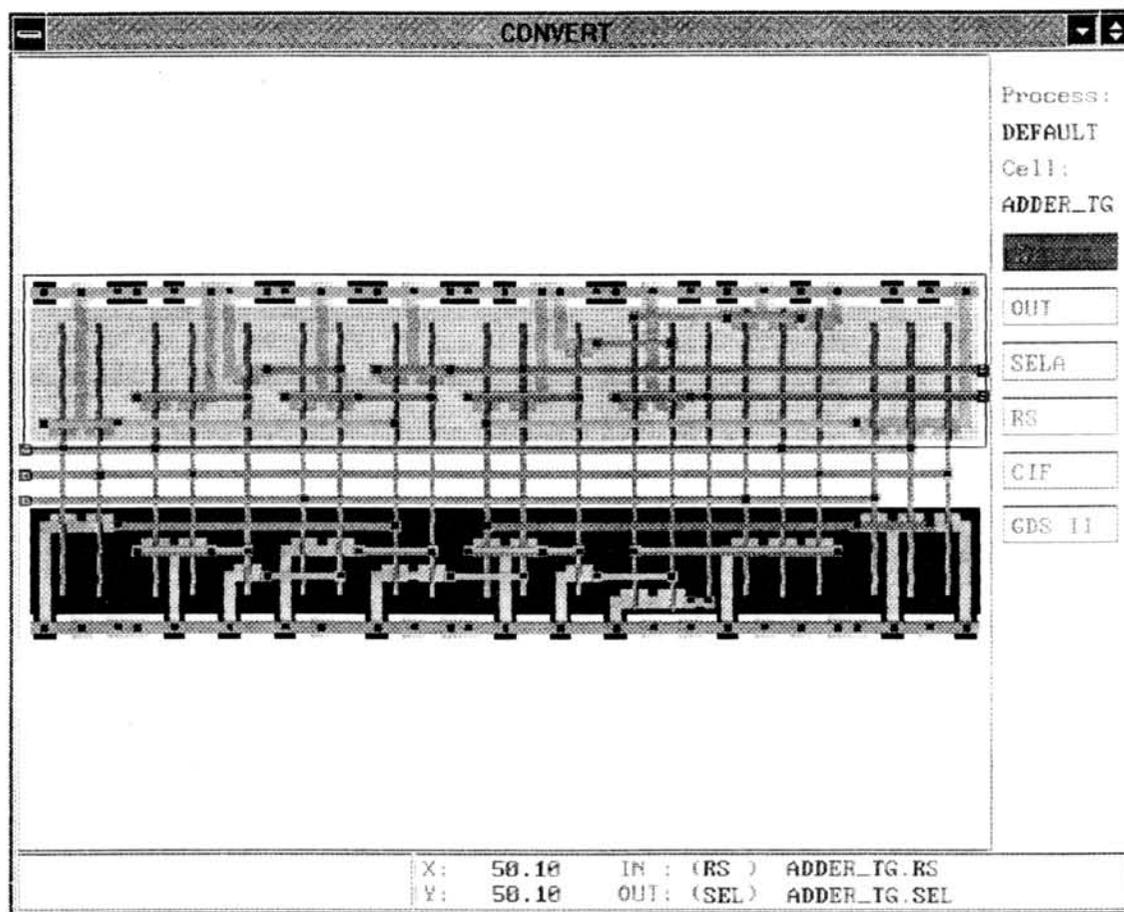


FIGURA 6.10 - Interface do conversor de formatos SEL \leftrightarrow RS

Ir no menu **Descrições - Layout Edition-SELA**

A figura 6.11 apresenta o leiaute gerado pelo Sistema TRAGO visualizado pelo editor de leiautes SELA, que encontra-se integrado ao Sistema TENTOS.

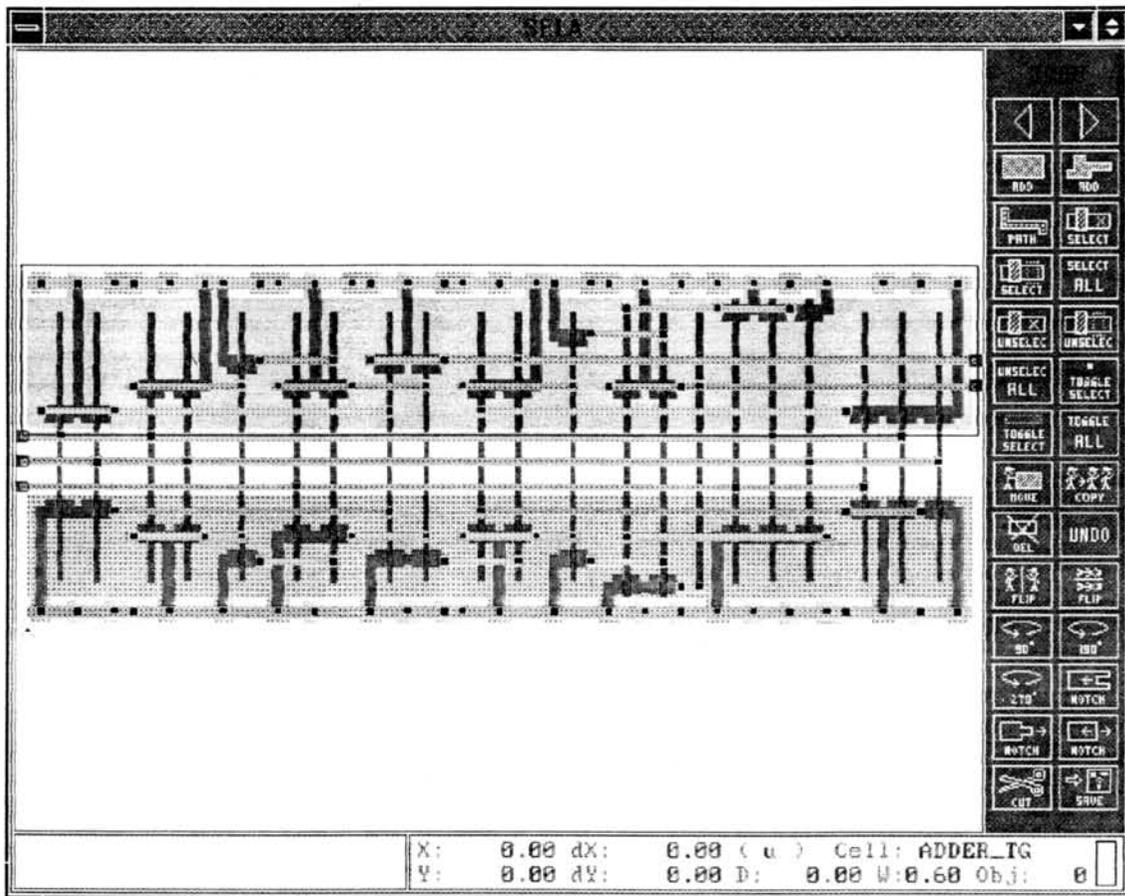


FIGURA 6.11 - Leiaute do circuito *c:\tentosus\adder_tg.sim* visualizado pelo SELA

Pode-se questionar a validade de utilizar o Editor de Leiautes SELA, pois para a sua utilização é necessário que se faça a conversão dos arquivos, executar o conversor de formatos de circuitos para transformar o arquivo descrito em Leiaute RS para Leiaute SEL. Caso o projetista apenas deseje visualizar o resultado obtido pelo TRAGO, o Editor de Leiautes EMA é mais do que suficiente, agora se o objetivo for alterar alguma especificação do circuito, a nível de leiaute, o Editor de Leiautes SELA oferece vários recursos extras que não são oferecidos pelo EMA.

6. Planificação

Ir no menu **Ferramentas - Flatenner**

A planificação do circuito é necessária para as etapas de verificação de regras de projeto. A figura 6.12 apresenta o resultado obtido após a execução do planificador.

```

EXPANSOR
UFRGS - PGCC Expansor de Layouts

Nome do arquivo a ser expandido : c:\tentosus\adder_tg.RS

Instanciando retangulo no. 643

Expansao completa.

C:\TENTOS\BIN>pause
Pressione qualquer tecla para continuar . . .

```

FIGURA 6.12 - Planificação do circuito *c:\tentosus\adder_tg.sim* gerado pelo TRAGO

7. Extração do *Netlist*

Ir no menu **Ferramentas - Extrator Elétrico**

A extração do *netlist* do circuito é necessária à etapa de comparação dos *netlists*. A figura 6.13 apresenta o resultado obtido após a execução do extrator de *netlists*.

```

EXT
EXTRATOR: EXTRATOR HIERARQUICO DE CIRCUITOS VERSAO 3.2
CPGCC - UFRGS - GRUPO DE MICROELETRONICA - MAS

Celula: 11:
  643 Retangulos

LIG A C
LIG P C
LIG M C
LIG H U
LIG H U
LIG P &

Tempo transcorrido: 0 segundos
Transistores tipo NMOS: 20
Transistores tipo PMOS: 20

Memoria livre=486696

* terminou *

C:\TENTOS\BIN>pause
Pressione qualquer tecla para continuar . . .

Lendo: c:\tentosus\adder_tg.RS

```

FIGURA 6.13 - Extração da *Netlist* do circuito *c:\tentosus\adder_tg.rs* gerado pelo TRAGO

8. Extração das portas lógicas

Ir no menu **Ferramentas - Extrator Lógico**

A figura 6.14 apresenta o resultado obtido após a execução do extrator lógico.

```

EXTRAL0
C:\TENTOSUS>C:\TENTOS\BIN\extra.exe c:\tentosus\adder_tg.sim c:\tentosus\adder_tg.HIE
-----
EXTRAL0 - logic extraction - 25/01/96
INPUT: c:\tentosus\adder_tg.sim   OUTPUT: c:\tentosus\adder_tg.HIE
-----
PRINTING OUTPUT FILE
*Number of transistors      40
*Number of cells           10
C:\TENTOSUS>pause
Pressione qualquer tecla para continuar . . .

```

FIGURA 6.14 - Extração lógica do circuito *c:\tentosus\adder_tg.sim* gerado pelo TRAGO

9. Verificação das regras de projeto

Ir no menu **Ferramentas - Layout Ruler Checker**

A figura 6.15 apresenta o resultado obtido após a execução do verificador de regras de projeto.

```

DARC3
DARC - VERIFICADOR DE REGRAS DE DESENHO
CPGCC/UFRRGS (VERSÃO 3.1)

Arquivo Layout RS: C:\TENTOSUS\adder_tg.RS
Arquivo de Regras: C:\tentos\tec\cnos12.drc

Numero de Retangulos: 643
Numero de Regras: 65

Retangulos Gerados: 583
Tempo decorrido: 1s
Erros encontrados: 10

Gravando arquivo C:\TENTOSUS\adder_tg.rel (relatorio) ...
Gravando arquivo C:\TENTOSUS\adder_tg.err (layout com erros) ...

C:\TENTOS\BIN>pause
Pressione qualquer tecla para continuar . . .

```

FIGURA 6.15 - Verificação das regras de projeto do circuito *c:\tentosus\adder_tg.sim*

10. Comparação das *netlists*

Ir no menu **Ferramentas - Comparador de Netlists**

A figura 6.16 apresenta a janela de entrada dos dados para o COTONET (comparador de *netlists*). Neste item será realizada a comparação entre o *netlist* inicial e o *netlist* gerado pelo extrator elétrico.

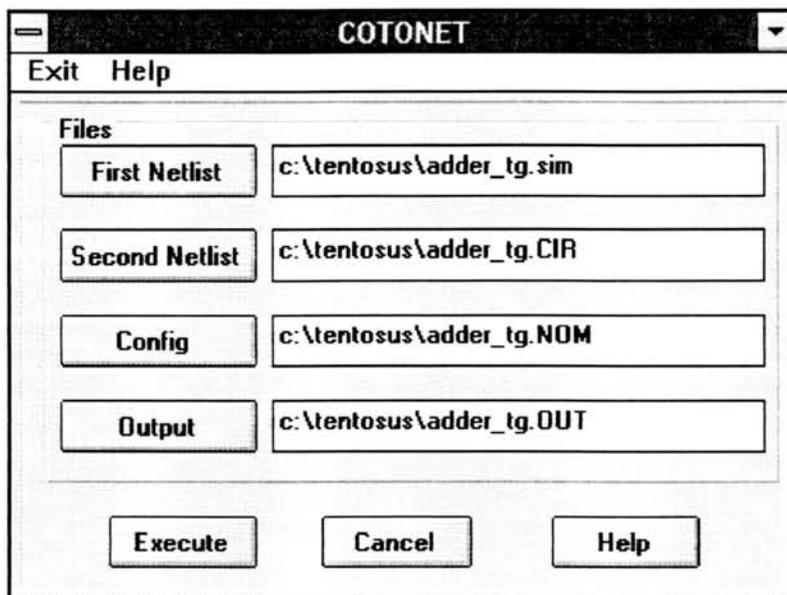


FIGURA 6.16 - Comparação dos *netlists* referentes ao circuito *c:\tentosus\adder_tg.sim*

A partir do momento em que o usuário pressiona o botão “Execute” é executada a comparação dos dois arquivos de *netlist* tendo como referência um arquivo de *dicas* (3º campo).

6.2 Projeto de um Circuito Utilizando o Sistema TROPIC

Neste exemplo serão apresentados os passos de síntese de um circuito através da ferramenta TROPIC. Os passos de verificação, extração e comparação que são executados após a geração do leiaute serão suprimidos nesta seção haja visto que são os mesmos passos apresentados na seção onde se apresentou o tutorial com a ferramenta TRAGO.

Será utilizado como circuito exemplo um somador cujas características são as mesmas das apresentadas na figura 6.2, tendo como única exceção a forma como foi implementada. Neste circuito foram utilizadas portas complexas.

A seguir serão apresentados os passos necessários para a realização deste processo.

1. Escolha do arquivo de configuração;

Ir no menu **Arquivo - Configuração - Lê “.CFG”**

Selecionar o arquivo C:\TENTOSUS\TENTOSUS.CFG

2. Seleção do Projeto que será manipulado

Ir no menu **Arquivo - Projeto** ou na barra de ferramentas e clicar sobre o ícone que representa **Projeto**

Selecionar o arquivo de *c:\tentosus\adder.prj*

Este arquivo já existe, e o seu conteúdo é o seguinte:

Circuito: <i>c:\tentosus\adder.sim</i> Tecnologia: CMOS12

No sistema TROPIC a tecnologia é configurada internamente à ferramenta, portanto o arquivo de tecnologia especificado no projeto será desconsiderado.

A seguir encontra-se a listagem do arquivo *c:\tentosus\adder.sim* em formato *spice*.

```
.subckt adder a b c carry sum vcc
M1 carry _16 vcc vcc PMOS l=1.0U w=8.0U
M2 _15 a vcc vcc PMOS l=1.0U w=8.0U
M3 vcc b _15 vcc PMOS l=1.0U w=8.0U
M4 _15 c _16 vcc PMOS l=1.0U w=8.0U
M5 _16 b _9 vcc PMOS l=1.0U w=8.0U
M6 _9 a _15 vcc PMOS l=1.0U w=8.0U
M7 _7 a _5 vcc PMOS l=1.0U w=8.0U
M8 _5 b _3 vcc PMOS l=1.0U w=8.0U
M9 _3 c _8 vcc PMOS l=1.0U w=8.0U
M10 _8 _16 _7 vcc PMOS l=1.0U w=8.0U
M11 _7 c vcc vcc PMOS l=1.0U w=8.0U
M12 vcc b _7 vcc PMOS l=1.0U w=8.0U
M13 _7 a vcc vcc PMOS l=1.0U w=8.0U
M14 vcc _8 sum vcc PMOS l=1.0U w=8.0U
M15 carry _16 gnd gnd NMOS l=1.0U w=8.0U
M16 gnd a _13 gnd NMOS l=1.0U w=8.0U
M17 _13 b _16 gnd NMOS l=1.0U w=8.0U
M18 _16 c _10 gnd NMOS l=1.0U w=8.0U
M19 _10 b gnd gnd NMOS l=1.0U w=8.0U
M20 gnd a _10 gnd NMOS l=1.0U w=8.0U
M21 gnd a _6 gnd NMOS l=1.0U w=8.0U
M22 _6 b _4 gnd NMOS l=1.0U w=8.0U
M23 _4 c _8 gnd NMOS l=1.0U w=8.0U
M24 _8 _16 _2 gnd NMOS l=1.0U w=8.0U
M25 _2 c gnd gnd NMOS l=1.0U w=8.0U
M26 gnd b _2 gnd NMOS l=1.0U w=8.0U
M27 _2 a gnd gnd NMOS l=1.0U w=8.0U
M28 gnd _8 sum gnd NMOS l=1.0U w=8.0U
.ends adder
X1 a b c carry sum vcc adder

*interface: a orient o
*interface: b orient o
*interface: c orient o
*interface: carry orient s
*interface: sum orient s
*width_supply 4.00
*width_metal2 1.20
```

Caso o arquivo de projeto não existisse, o usuário após especificar o nome do projeto deveria especificar o circuito e o arquivo de tecnologia que serão manipulados.

3. Manipulação do arquivo com a descrição *spice* do circuito

Ir no menu **Descrições - Textual Edition**

Para este exemplo será apresentada a interface de edição textual do circuito ilustrada pela figura 6.17.

```

Bloco de Notas - ADDER.SIM
Arquivo  Editar  Localizar  ?
.subckt adder a b c carry sum vcc
M1 carry _16 vcc vcc PMOS L=1.0U W=8.0U
M2 _15 a vcc vcc PMOS L=1.0U W=8.0U
M3 vcc b _15 vcc PMOS L=1.0U W=8.0U
M4 _15 c _16 vcc PMOS L=1.0U W=8.0U
M5 _16 b _9 vcc PMOS L=1.0U W=8.0U
M6 _9 a _15 vcc PMOS L=1.0U W=8.0U
M7 _7 a _5 vcc PMOS L=1.0U W=8.0U
M8 _5 b _3 vcc PMOS L=1.0U W=8.0U
M9 _3 c _8 vcc PMOS L=1.0U W=8.0U
M10 _8 _16 _7 vcc PMOS L=1.0U W=8.0U
M11 _7 c vcc vcc PMOS L=1.0U W=8.0U
M12 vcc b _7 vcc PMOS L=1.0U W=8.0U
M13 _7 a vcc vcc PMOS L=1.0U W=8.0U
M14 vcc _8 sum vcc PMOS L=1.0U W=8.0U
M15 carry _16 gnd gnd NMOS L=1.0U W=8.0U
M16 gnd a _13 gnd NMOS L=1.0U W=8.0U
M17 _13 b _16 gnd NMOS L=1.0U W=8.0U
M18 _16 c _10 gnd NMOS L=1.0U W=8.0U
M19 _10 b gnd gnd NMOS L=1.0U W=8.0U
M20 gnd a _10 gnd NMOS L=1.0U W=8.0U
M21 gnd a _6 gnd NMOS L=1.0U W=8.0U
M22 _6 b _4 gnd NMOS L=1.0U W=8.0U
M23 _4 c _8 gnd NMOS L=1.0U W=8.0U
M24 _8 _16 _2 gnd NMOS L=1.0U W=8.0U
M25 _2 c gnd gnd NMOS L=1.0U W=8.0U
M26 gnd b _2 gnd NMOS L=1.0U W=8.0U
M27 _2 a gnd gnd NMOS L=1.0U W=8.0U
M28 gnd _8 sum gnd NMOS L=1.0U W=8.0U
.ends adder

```

FIGURA 6.17 - Janela de Edição Textual com o arquivo *c:\tentosus\adder.sim*

Se o circuito já estiver correto está é uma fase que pode ser suprimida, agora caso faltem informações no circuito, ou o circuito ainda não exista, esta é a etapa onde tais informações deverão ser acrescentadas à descrição *SPICE*.

4. Geração do processo de síntese utilizando a ferramenta TROPIC

Ir no menu **Sintetizar - TROPIC**

A figura 6.18 apresenta a interface do Sistema TROPIC com os parâmetros apropriados.

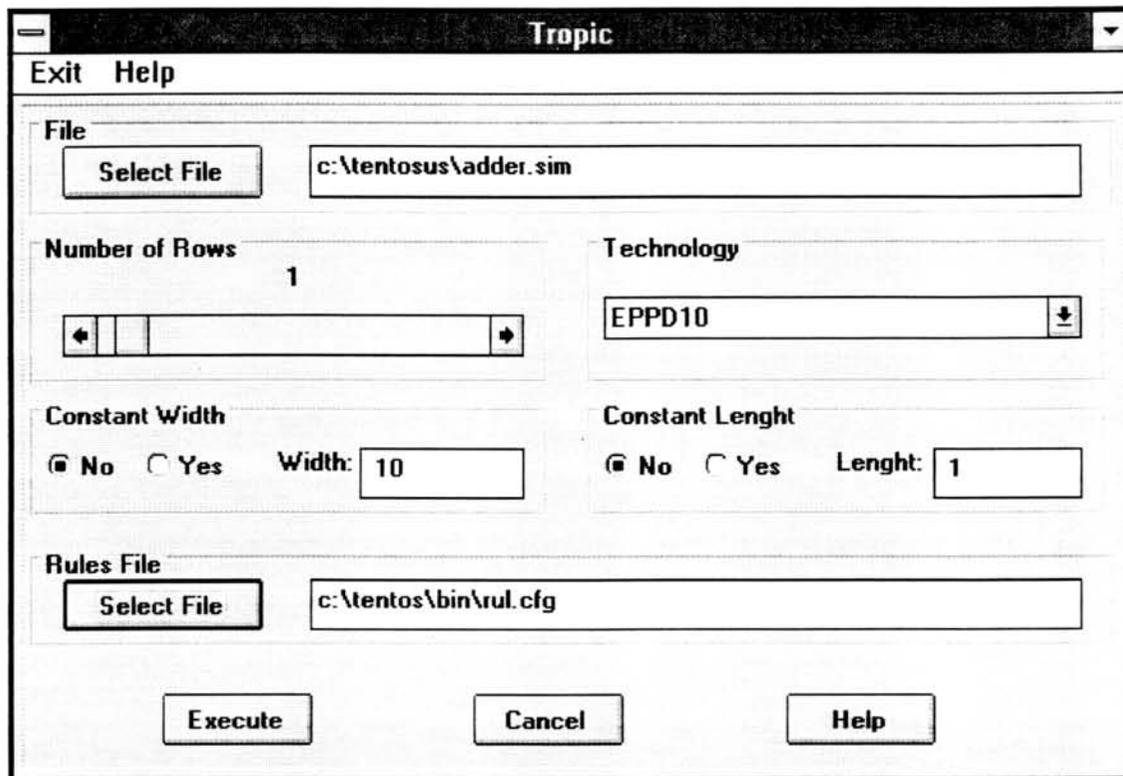


FIGURA 6.18 - Sistema TROPIC com os parâmetros certos para a geração do processo de síntese do arquivo *c:\tentosus\adde.sim*

Para a execução do sistema TROPIC todos os dados podem ser configurados internamente à ferramenta e são eles:

- Nome do circuito;
- Número de bandas;
- Tecnologia;
- Largura dos transistores;
- Comprimento dos transistores;
- Nome do arquivo de configuração.

No momento em que o projetista selecionar o botão *Execute* do TROPIC executará os seguintes passos:

a) Extração lógica circuito

Nesta etapa, vide figura 6.19, do Sistema TRAGO é feita a extração das portas lógicas do circuito, utilizando para isso o extrator lógico EXTRALO. Este processo é interno ao TROPIC.

```

EXTRALO
-----
EXTRALO - logic extraction - 25/01/96
INPUT: adder.sim   OUTPUT: adder.hie
-----
PRINTING OUTPUT FILE
*Number of transistors      28
*Number of cells           4
Pressione qualquer tecla para continuar . . .

```

FIGURA 6.19 - Sistema TROPIC - extração das portas lógicas do circuito *c:\tentosus\adder.sim*

b) Posicionamento do circuito

Nesta etapa, vide figura 6.20, do Sistema TROPIC é feito o posicionamento do circuito.

```

POSIC
-----
PLACEMENT - 24/01/95
-----
FILE adder.hie - LIB bib - cells#: 4 signals#: 9
Partitions : # VERT 1 # HORIZ 2
Rows:      1
Lengths:   28
Total length: 28 IDEAL 28 MIN 28 MAX 28 DELTA 0
CPU time:  0.0 sec
Pressione qualquer tecla para continuar . . .

```

FIGURA 6.20 - Sistema TROPIC - posicionamento do circuito *c:\tentosus\adder.sim*

c) Síntese do circuito

Nesta etapa, vide figura 6.21 e 6.22, do Sistema TROPIC é feita a síntese do circuito, propriamente dita, onde é gerado um arquivo em formato SYL do circuito com o nome de *c:\tentosus\adder.syl*.

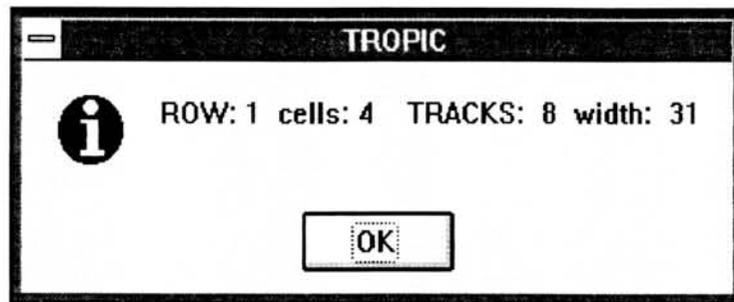


FIGURA 6.21 - Sistema TROPIC - mensagens apresentadas no momento da síntese

Na figura 6.21 são apresentados o número de bandas, número de células do circuito, quantas trilhas e a largura dos transistores em uma etapa intermediária do processo de síntese. Já na figura 6.22 é apresentado o resultado final do número de trilhas do circuito.

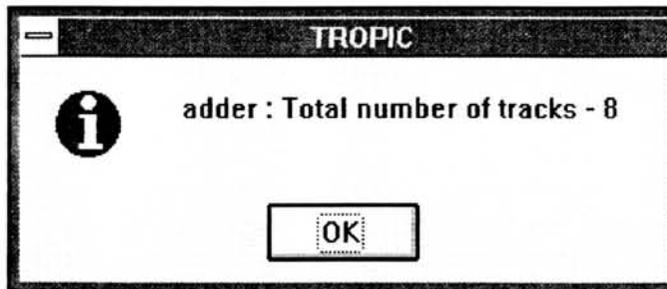


FIGURA 6.22 - Sistema TROPIC - mensagem apresentada no momento da síntese

Concluído o processo de síntese do circuito, o passo seguinte é a conversão de formatos a fim de que se possa compactar o circuito através da ferramenta SYLC.

5. Conversão do formato SYL para LDS

Ir no menu **Ferramentas - Conversor SYL <--> LDS**

Esta etapa tem a finalidade de converter o arquivo gerado pelo TROPIC, em formato .SYL, para o formato .LDS. Este formato serve como entrada tanto para o editor simbólico SIMBY como para o compactador SYLC.

6. Visualização do arquivo através do editor simbólico SIMBY

Ir no menu **Descrições - Symbolic Editor**

A figura 6.23 apresenta o editor simbólico SIMBY com o leiaute simbólico gerado pelo TROPIC, arquivo **c:\tentosus\adder.lds**.

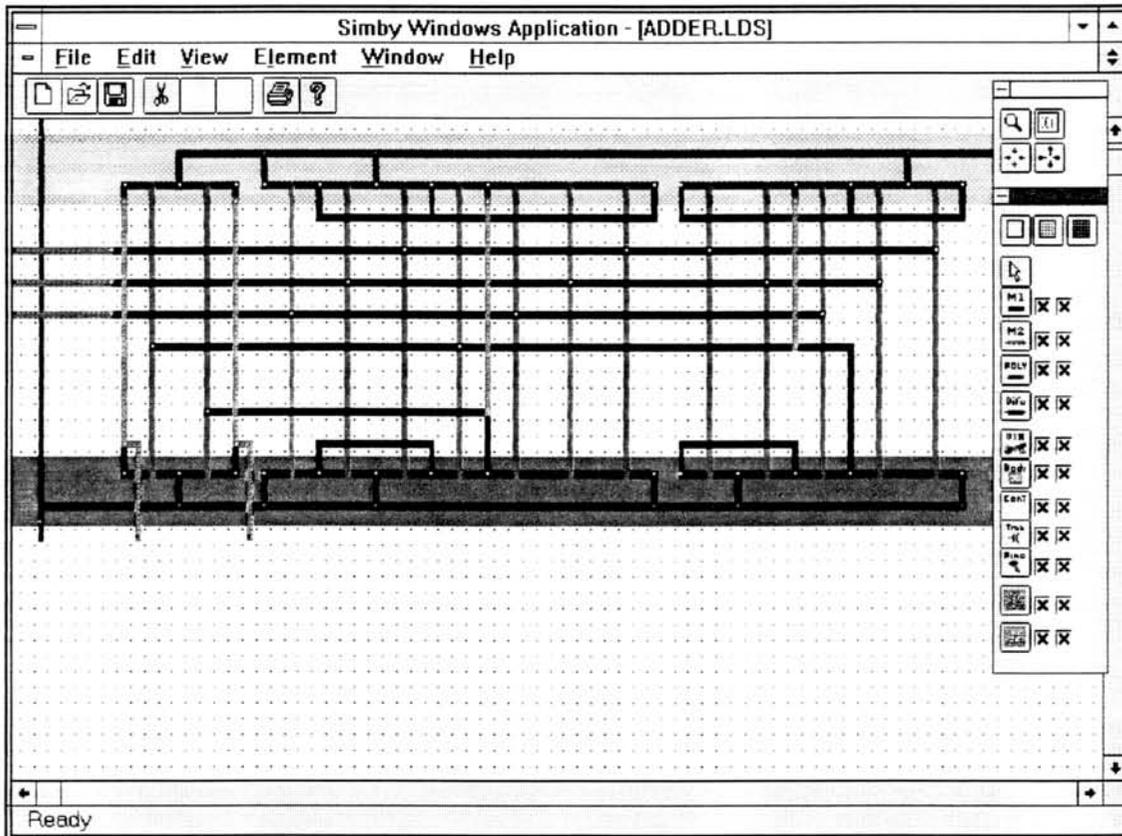


FIGURA 6.23 - Edição simbólica do arquivo gerado

7. Compactação do leiaute descrito em formato LDS

Ir no menu **Ferramentas - Compactador**

A figura 6.24 ilustra o resultado obtido através da compactação do circuito. Como resultado de saída do compactador é gerado um arquivo em formato Leiaute RS.

```

SYLC
C:\TENTOS\BIN>SYLC.EXE c:\tentosus\adder.LDS -tC:\tentos\tec\cmos12.TCN
Compacting cell adder.lds
Using technology file C:\tentos\tec\cmos12.TCN.
Caractere invalido '\'-92

Partial time: 10 s.
X: 10487
Y: 5782

Total time: 10 s.
C:\TENTOS\BIN>PAUSE
Pressione qualquer tecla para continuar . . .

```

FIGURA 6.24 - Compactação do circuito *c:\tentosus\adder.lds*

Concluída a geração do leiaute em formato RS, os passos seguintes são idênticos aos encontrados no tutorial do sistema TRAGO.

6.3 Tutorial de edição manual do leiaute

A seguir será apresentado um exemplo de geração manual do leiaute:

1. Edição do leiaute, utilizando as células da biblioteca ou gerando novas células;
2. Verificar as regras de projeto (DARC), e caso haja erros retornar ao editor de leiaute para corrigi-los. Se a descrição do leiaute for hierárquica, o circuito deve ser previamente planejado;
3. Realizar a extração elétrica, obtendo arquivos em formato *Spice*. Inserir no arquivo extraído os vetores de teste, os pontos a serem observados e as instruções do simulador. Caso o desempenho do circuito não satisfaça, remodela-se os transistores até obter-se o desempenho desejado;
4. Extração lógica: permite a obtenção de um arquivo com a lista de todas as células básicas (inversores, nands, nors, super-portas) do circuito. Auxilia muito quando temos um longo *netlist Spice*, pois fica impossível descobrir pela lista de transistores quais formam uma determinada célula;
5. Conversão para CIF: converte o formato RS para o formato CIF, normalmente aceito pelas *foundrys*.

6.4 Usando o Sistema TENTOS no Futuro

A partir das implementações sugeridas no capítulo 5 pode-se fazer uma previsão de como será o uso do sistema no futuro.

Através da alteração proposta para o arquivo de projeto, o sistema TENTOS mudaria completamente. Ele passaria a ser um editor, como o MS-WORD™ por exemplo, onde o documento iria conter as etapas do projeto e, a partir de um *duplo click* do mouse sobre um objeto do documento, o sistema iria executar a ferramenta que está vinculada àquele objeto via protocolo OLE.

Este arquivo de projeto, caso não exista, será inicializado através de um *script* básico. As informações, que farão parte desta estrutura básica, especificarão os passos de projeto para a execução da concepção do circuito. Todas estas informações serão configuradas no Gerenciador. Deste modo o documento será um conjunto de objetos onde cada um, quando selecionado, executará uma ferramenta.

Utilizando-se DLLs, os programas que não requeiram interface gráfica (por exemplo: Verificadores de Regras de Projeto, Extrator Lógico, Conversores de Formato) serão convertidos em funções que poderão ser executadas por qualquer programa, a qualquer momento, sem necessidade de interação com o Gerenciador. A realização da síntese automática do circuito poderá ser feita através de um programa que executa estas diversas funções disponíveis nas DLLs geradas.

A utilização de comunicação de dados via DDE poderá ser utilizada na realização de simulações, por exemplo. O projetista estando no Editor de Esquemáticos poderá disparar uma simulação de apenas uma parte do circuito. As variações de sinais na simulação poderão representar trocas de cores no esquemático.

Sendo assim, o Sistema TENTOS apresentará as seguintes etapas de projeto:

1. O projetista seleciona o arquivo de projeto com o circuito que será manipulado;
2. Executando a edição esquemática do circuito poder-se-á realizar simulações totais ou parciais do circuito;

3. No momento em que o projetista finalizar a edição do esquemático, o sistema deverá realizar automaticamente a conversão de formatos, de *ESQuemático* para *Spice*. Esta conversão de formatos terá como resultado a atualização do objeto que apresenta a descrição *Spice* do circuito. A conversão automática de formatos será prevista no documento. A configuração disto será feita no momento em que o usuário estiver instalando a ferramenta no sistema;
4. Concluída a descrição do circuito, o passo seguinte será o da realização da síntese automática do circuito. A realização da síntese automática através de um programa que executa funções que constam de uma DLL traz para a ferramenta benefícios, tais como:
 - i. maior agilidade quando da implementação de novas técnicas;
 - ii. verificação a partir do próprio sistema de síntese, uma vez que as ferramentas de validação do circuito também estarão em DLLs;
 - iii. independência da interface do programa de síntese da sua execução.
5. Visualização do leiaute obtido e/ou simulações a partir do leiaute.
6. Impressão do relatório de projeto, a partir do próprio arquivo de documento, acrescido de outras informações, tais como:
 - i. parâmetros do gerenciador;
 - ii. descrições textuais das visualizações esquemáticas e geométricas do circuito;
 - iii. tempo de processamento em cada ferramenta.

6.5 Conclusão

Apesar de possuir mais etapas de projeto, a utilização de ferramentas de síntese para o projeto de CIs consome muito menos tempo de projeto que a de edição manual do leiaute. Outro fato a ser citado é que as ferramentas de síntese são cada vez mais usadas em detrimento da edição manual do leiaute. Convém salientar que as ferramentas de edição de leiaute continuam a ser usadas, ou para a edição de células que irão compor uma biblioteca de células, ou para o projeto de algum circuito que possua características muito especiais que impossibilitem o uso de ferramentas de síntese.

A utilização do sistema TENTOS no futuro, tendo por base o exemplo apresentado na seção 6.4, será mais simplificada e com menos margem de erro, haja visto que o projetista seguirá um roteiro básico de etapas a cumprir especificado no corpo do arquivo do projeto.

7 Conclusões

As empresas do mercado mundial de circuitos integrados possui como ambiente de projeto, quase que na sua totalidade, os sistemas comerciais que aqui foram citados, pois tratam-se de ferramentas que integram programas que abordam todas as etapas do projeto, desde a descrição comportamental até a geração do leiaute. Por se tratarem de sistemas de grande complexidade, estes frameworks encontram-se desenvolvidos para estações de trabalho padrão UNIX, porque são sistemas que requerem muita memória, processadores extremamente rápidos e um sistema operacional adequado aos diversos tipos de operações que devem ser executados, tais como, processamento paralelo de dois ou mais programas. Estas características limitam o uso de computadores tipo IBM-PC. Já as ferramentas acadêmicas, por terem como um dos seus objetivos, atender a trabalhos não tão complexos, possuem, em alguns casos, versões para computadores tipo IBM-PC haja visto que muito dos alunos possuem em suas residências computadores deste tipo.

Buscando suprir a demanda do mercado acadêmico de projetos de circuitos integrados, o sistema TENTOS vem sofrendo evoluções a fim de que possam ser suprimidas determinadas carências do sistema, ora em relação ao ambiente de trabalho, ora com a inclusão de novas características (por exemplo: permitir que novas ferramentas sejam incorporadas ao sistema, sem a necessidade de alterar informações no código fonte do gerenciador). Através destas constantes evoluções do sistema, acreditamos estar fornecendo a empresas e instituições de ensino e pesquisa uma ferramenta de baixo custo para o projeto de circuitos integrados.

Dentre as características que orientaram este trabalho, merece uma atenção especial o fato desta evolução tornar a ferramenta mais ágil, que não seja tão sensível às mudanças tecnológicas que estão sempre ocorrendo. A partir do momento em que o TENTOS é um gerenciador aberto, em versão para MS-WINDOWS™, está-se dotando o sistema de recursos que permitirão a incorporação de novas ferramentas, novos arquivos de tecnologia, sem a necessidade de alteração do código fonte do TENTOS, para cada nova ferramenta que venha a ser incorporada ao sistema. Outro aspecto importante no trabalho realizado foi a inclusão de uma nova ferramenta de síntese, com novos conceitos de geração de leiaute, o TROPIC, na configuração *standard* do sistema.

Através das implementações sugeridas no capítulo 5, o sistema TENTOS poderá tornar-se um pacote homogêneo, pois todas as interações realizadas pelo projetista seriam feitas em telas *for Windows*. Deste modo, as *piscadas* que ocorrem no momento de troca do ambiente de trabalho MS-WINDOWS™→MS-DOS™→MS-WINDOWS™, não mais ocorreriam. A implementação destas sugestões contribuirá para que o projetista reduza o tempo de desenvolvimento do circuito. Já a utilização de recursos próprios do MS-WINDOWS™ (protocolo OLE, DDE, Clipboard) possibilitará a utilização de ferramentas em paralelo sendo que o resultado de uma irá gerar alterações no processo da outra e vice-versa.

Anexo 1 Listagem do Arquivo *TENTOS.DAT*

O Arquivo listado a seguir encontra-se no diretório *C:\TENTOS\BIN*.

[IniTentos]

FileIniName=C:\tentosus\tentos.ini

IniNameExt=tentos.ini

IniName=tentos

[EditorTentos]

Editor=notepad.exe

[PosicJanTentos]

AboutX= 2445

AboutY= 1275

AcessArqX= 2438

AcessArqY= 2588

AddCutTechX= 0

AddCutTechY= 0

AddCutExtX= 1365

AddCutExtY= 1155

AddCutX= 345

AddCutY= 180

DiretoriX= 645

DiretoriY= 855

JanParamX= 2220

JanParamY= 1245

JanEditorX=0

JanEditorY=0

SalvaArqX= 0

SalvaArqY= 0

TentosX= 1095

TentosY= 1920

TentosH= 4995

TentosL= 7620

SelecNameX= 3045

SelecNameY= 3300

SelecProjX= 0

SelecProjY= 0

[Lingua]

Lingua= 1

Anexo 2 Listagem do Arquivo *TENTOS.INI*

O Arquivo listado a seguir encontra-se no diretório *C:\TENTOSUS*.

[Tech_Tentos]

FileTech.NumTech.= 3

FileTech.File. 1=C:\tentos\tec\cmos12.tec

FileTech.TextoMen. 1=CMOS12

FileTech.File. 2=C:\tentos\tec\cmos15.tec

FileTech.TextoMen. 2=CMOS15

FileTech.File. 3=C:\tentos\tec\cmos20.tec

FileTech.TextoMen. 3=CMOS20

[MenuTentos]

MenFile.NumItens. 1= 0

MenFileConfigOptions.NumItens. 2= 0

MenDesktop.NumItens. 3= 6

MenDesktop.Comando. 3. 1=C:\tentos\BIN\EMA2.EXE

MenDesktop.TextoMen. 3. 1=Layout Edition - &EMA

MenDesktop.Argumentos. 3. 1=\$CIRC_D.rs \$TECH_DE

MenDesktop.InitialDir. 3. 1=C:\tentos\BIN

MenDesktop.HelpTool. 3. 1=

MenDesktop.Comando. 3. 2=C:\tentos\SELA\SELA.EXE

MenDesktop.TextoMen. 3. 2=Layout Edition - &SELA

MenDesktop.Argumentos. 3. 2=\$DIR_US\SELA.CFG

MenDesktop.InitialDir. 3. 2=

MenDesktop.HelpTool. 3. 2=

MenDesktop.Comando. 3. 3=NOTEPAD.EXE

MenDesktop.TextoMen. 3. 3=&Textual Edition

MenDesktop.Argumentos. 3. 3=\$CIRC_DE

MenDesktop.InitialDir. 3. 3=

MenDesktop.HelpTool. 3. 3=

MenDesktop.Comando. 3. 4=C:\tentos\esquelet\edesq.exe

MenDesktop.TextoMen. 3. 4=Schematic Edition - &ESQUELETO

MenDesktop.Argumentos. 3. 4=

MenDesktop.InitialDir. 3. 4=

MenDesktop.HelpTool. 3. 4=

MenDesktop.Comando. 3. 5=C:\tentos\esquelet\edsmb.exe

MenDesktop.TextoMen. 3. 5=Symbolic Editor - ESQUELETO

MenDesktop.Argumentos. 3. 5=

MenDesktop.InitialDir. 3. 5=

MenDesktop.HelpTool. 3. 5=

MenDesktop.Comando. 3. 6=C:\tentos\BIN\simby.exe

MenDesktop.TextoMen. 3. 6=Symbolic Editor

MenDesktop.Argumentos. 3. 6=

MenDesktop.InitialDir. 3. 6=

MenDesktop.HelpTool. 3. 6=

MenSynthesis.NumItens. 4= 2
 MenSynthesis.Comando. 4. 1=C:\tentos\BIN\TROPIC.EXE
 MenSynthesis.TextoMen. 4. 1=&TROPIC
 MenSynthesis.Argumentos. 4. 1=%CIRC_D.RS 1 10 10 0 \$DIR_US\TENTOSUS.CFG
 MenSynthesis.InitialDir. 4. 1=
 MenSynthesis.HelpTool. 4. 1=
 MenSynthesis.Comando. 4. 2=C:\tentos\BIN\TRAGOWIN.EXE
 MenSynthesis.TextoMen. 4. 2=T&RAGO
 MenSynthesis.Argumentos. 4. 2=%CIRC_DE \$CONF_DE \$DIR_US\PART.PRM
 MenSynthesis.InitialDir. 4. 2=
 MenSynthesis.HelpTool. 4. 2=

MenTools.NumItens. 5= 9
 MenTools.Comando. 5. 1=C:\tentos\sela\convert.exe
 MenTools.TextoMen. 5. 1=Conversor SEL <--> RS
 MenTools.Argumentos. 5. 1=\$DIR_US\sela.cfg
 MenTools.InitialDir. 5. 1=
 MenTools.HelpTool. 5. 1=
 MenTools.Comando. 5. 2=C:\tentos\bin\ednet.bat
 MenTools.TextoMen. 5. 2=Conversor ESQ <--> CIR
 MenTools.Argumentos. 5. 2=%CIRC.ESQ \$DIR_US\CIRC.CIR
 MenTools.InitialDir. 5. 2=
 MenTools.HelpTool. 5. 2=
 MenTools.Comando. 5. 3=C:\tentos\BIN\DARC3.bat
 MenTools.TextoMen. 5. 3=&Layout Ruler Checker
 MenTools.Argumentos. 5. 3=\$DIR_US\CIRC.RS \$TECH_D.drc
 MenTools.InitialDir. 5. 3=
 MenTools.HelpTool. 5. 3=
 MenTools.Comando. 5. 4=C:\tentos\BIN\rs2cif.bat
 MenTools.TextoMen. 5. 4=Conversor RS <--> CIF
 MenTools.Argumentos. 5. 4=\$DIR_US\CIRC.RS \$TECH_DE
 MenTools.InitialDir. 5. 4=
 MenTools.HelpTool. 5. 4=
 MenTools.Comando. 5. 5=C:\tentos\BIN\ex7.bat
 MenTools.TextoMen. 5. 5=&Extrator Eléctrico
 MenTools.Argumentos. 5. 5=%CIRC_D.RS \$TECH_DE -S\$DIR_US\OUT
 MenTools.InitialDir. 5. 5=
 MenTools.HelpTool. 5. 5=
 MenTools.Comando. 5. 6=C:\tentos\BIN\expansor.bat
 MenTools.TextoMen. 5. 6=&Flatenner
 MenTools.Argumentos. 5. 6=%CIRC_D.RS
 MenTools.InitialDir. 5. 6=
 MenTools.HelpTool. 5. 6=
 MenTools.Comando. 5. 7=C:\tentos\BIN\sylc.BAT
 MenTools.TextoMen. 5. 7=&Compactador
 MenTools.Argumentos. 5. 7=%CIRC_DE -t\$TECH_D.TCN
 MenTools.InitialDir. 5. 7=
 MenTools.HelpTool. 5. 7=
 MenTools.Comando. 5. 8=C:\tentos\BIN\EXTRALO.EXE

MenTools.TextoMen. 5. 8=Extrator &Lógico
 MenTools.Argumentos. 5. 8=\$CIRC_DE \$CIRC_D.HIE
 MenTools.InitialDir. 5. 8=
 MenTools.HelpTool. 5. 8=
 MenTools.Comando. 5. 9=C:\tentos\BIN\CNET.EXE
 MenTools.TextoMen. 5. 9=Comparador de &NetLists
 MenTools.Argumentos. 5. 9=\$CIRC_DE \$CIRC_D.CIR \$CIRC_D.NOM
 \$CIRC_D.OUT
 MenTools.InitialDir. 5. 9=
 MenTools.HelpTool. 5. 9=

MenToolsVerif.NumItens. 6= 0

MenSimulator.NumItens. 7= 1
 MenSimulator.Comando. 7. 1=C:\tentos\bin\spice.bat
 MenSimulator.TextoMen. 7. 1=Simulador &Spice
 MenSimulator.Argumentos. 7. 1=\$DIR_US\CIRC.CIR \$DIR_US\CIRC.REL
 MenSimulator.InitialDir. 7. 1=
 MenSimulator.HelpTool. 7. 1=

[Ext_Tentos]

Extension.NumExt= 6
 Extension.TextoMen. 1=All Files (*.*)
 Extension.TextoMen. 2=Spice (*.CIR)
 Extension.TextoMen. 3=Layout RS (*.RS)
 Extension.TextoMen. 4=Schematic (*.ESQ)
 Extension.TextoMen. 5=Simbolico (*.LDS)
 Extension.TextoMen. 6=Spice (*.SIM)

[DiretTentos]

DirUsu=C:\TENTOSUS
 DirEXE=C:\tentos\BIN
 DirTec=C:\tentos\TEC
 DirIn=C:\TENTOSUS
 DirOut=C:\TENTOSUS
 DirLib=C:\tentos\LIB

[ProjTentos]

FileProjName=c:\tentosus\adder_tg.prj
 ProjNameExt=adder_tg.prj

[CircTentos]

FileCircName=c:\tentosus\adder_tg.sim
 CircNameExt=adder_tg.sim
 CircName=adder_tg
 CircNameDir=c:\tentosus\adder_tg

[FileTechTentos]

FileTechName=C:\tentos\tec\cmos12.tec

TechName=cmos12
TechNameDir=C:\tentos\tec\cmos12

[ShowVarTentos]
ShowJanParam= 1
ShowToolBar= 1
ShowStatusBar= 1

[CFGTentos]
FileConfigName=C:\tentosus\tentosus.cfg
ConfigNameExt=tentosus.cfg
ConfigName=tentosus

[DiretoriosTentos]
DiretCirc=c:\tentosus
DiretCfg=C:\tentosus
DiretInit=C:\tentosus
DiretProj=c:\tentosus

Anexo 3 Listagem do Arquivo *TENTOSUS.CFG*

O Arquivo listado a seguir encontra-se no diretório *C:\TENTOSUS*.

**** ARQUIVO DE CONFIGURACAO PARA O TRANCA / TRAMO / TRAGO

**** OBSERVACAO: Apenas as palavras que comecam por \$ sao palavras reservadas, desta forma e' possivel termos tantas regras de desenho para o trago e tramo quanto quisermos, bastando ativar as regras com a palavra RESERVADA `Sregras`.

DIRETORIOS DO USUARIO

\$dir_entrada \TENTOSUS
\$dir_saida \TENTOSUS
\$dir_tmp \TENTOSUS

DIRETORIOS QUE DEVEM SER FIXOS:

\$dir_lib C:\tentos\LIB
\$dir_tec C:\tentos\TEC
\$dir_aramos C:\tentos\ARAMOS
\$dir_pac C:\tentos\BIN
\$dir_esqueleto C:\tentos\ESQUELET
\$dir_tranca C:\tentos\BIN
\$dir_spice C:\tentos\SPICE
\$dir_exemplos C:\tentos\SAMPLES
\$dir_texto C:\tentos\BIN

NOME DO ARQ. DE REGRAS QUE DESCRIVEM A TECNOLOGIA - no diretorio
dir_tec

\$arquivo_regras CMOS20

NOME DO ARQUIVO QUE DESCRIVE A BIBLIOTECA - no diretorio dir_lib

\$gerente_celula cel

EXTENSAO DAS CELULAS DA BIBLIOTECA

\$sufixo_celula CEL

EXTENSAO DOS ARQUIVOS SPICE

\$sufixo_spice SIM

FORMATO DA SAIDA DO EXTRIBO (EX7) valido: spice (default) ou aramos

\$out_extribo aramos

INDICA A NECES. OU NAO DE APERTAR UMA TECLA NO FIM DOS PROGRAMAS

\$wait 0

***** REGRAS DE DESENHO PARA O TRAGO - lambda

regras

lambda 1.0 o lambda deve vir antes das regras

LA 4#

DAA 4#

DAI 2#

DIW 3#

DDB 4.0

DSG 17#

LP 2#

DPP 3#

DAP 2#

STP 2#

LC 2#

LM 4#

DMC 2#

DMM 3#

LV 3#

LH 4#

DHV 2#

DHH 3#

\$endregras

***** REGRAS DE DESENHO PARA O TRAGO - 2.0u

regras

lambda 0.5 o lambda deve vir antes das regras

LA 3#

DAA 3#

DAI 2#

DIW 3#

DDB 4.0

DSG 17#

LP 2#

DPP 2.5

DAP 1#

STP 2#

LC 2#

LM 3.5

DMC 1.5

DMM 2.5

LV 2.5

LH 3.5

DHV 1#

DHH 3#

\$endregras

***** REGRAS DE DESENHO PARA O TRAGO - 1.5u

regras

lambda 0.2 o lambda deve vir antes das regras

LA 2.4

DAA 3.2

DAI 2#
 DIW 2.8
 DDB 4.0
 DSG 9.6
 LP 1.6
 DPP 2.4
 DAP 1#
 STP 1.6
 LC 2#
 LM 2.4
 DMC 1#
 DMM 2.4
 LV 2#
 LH 2.4
 DHV 1#
 DHH 2.4
 Sendregas

***** REGRAS DE DESENHO PARA O TRAGO - 1.2u

\$regras
 lambda 0.15 o lambda deve vir antes das regras
 LA 1.5
 DAA 2.4
 DAI 1.5
 DIW 2.1
 DDB 2.4
 DSG 7.2
 LP 1.2
 DPP 1.8
 DAP .6
 STP 1.2
 LC 1.5
 LM 1.8
 DMC .75
 DMM 1.8
 LV 1.5
 LH 1.8
 DHV .75
 DHH 1.8
 Sendregas

***** PARAMETROS PARA O TRAGO

\$TEC N tipo dos pocos: N, P, NP
 \$OUT rs saida: RS ou CIF
 \$SW_ALIMENTA 10.0 largura da alimentacao
 \$SPOSIC 4 grau para conectividade no placement
 \$OFFSET 0 distancia entre vcc,gnd e as difusoes
 \$REPETIR 8 numero de vezes para realizar o roteamento
 \$MIN_HORIZ 6 fator de justaposicao

SGND_ESQ 1 se 1 o terra e a esquerda

***** NOME DAS CAMADAS DE DESENHO, EM RS E CIF

\$camadas 12

poli LMP CPOL
 metal1 LMM CME1
 metal2 LMH CME2
 difn LMD CNPI
 difp LMB CPPI
 pocon LMS CNWI
 pocop LMG CPWI
 cont LMC CCON
 via LMV CVIA
 impn LMD CNPI
 impv LMB CPPI
 ativa LMA CTOX

***** DEFINICOES UTILIZADAS PELO ROTEADOR *****

====> File RO_LAY.C & RO_SAIDA.C:

\$regras_ro

YVCC 97 /* coordenada y da linha de VCC */
 LALIM 6 /* largura da trilha de VCC */
 LTRI 4 /* largura de trilha */
 YGND 3 /* coordenada y da linha de GND */
 LPOLI 2 /* largura de nodo */
 HPOLI 94 /* altura de nodo */
 YNOD 12 /* coordenada y de nodo */
 XENT 2 /* deslocamento com relacao a pos real do nodo */
 DH 12 /* delta de altura em caso de cel. int. EXT */
 DMM 3
 CC 6
 LCH 2
 LCV 2
 LMH 6
 LMV 6
 LPH 6
 LPV 6
 DYC 2
 LPCI 6
 DPFI 3
 HPOC 46
 OPB 60 /* coord. inferior do poco P */
 OPD 0 /* coord. inferior do poco N */
 LNOD 4 /* largura de nodo */
 HNOD 82 /* altura de nodo */
 OFFSET 1 /* deslocamento com relacao a pos real do nodo */
 FRONT 3 /* distancia do contato a fronteira de uma celula de interconexao */
 \$endregras

Bibliografia

- [BAG 89] BAGGIO, André. **Uma interface de gerenciamento para o Projeto Tranca**: Projeto de diplomação. Porto Alegre: CIC da UFRGS, 1989. 74 p.
- [CAA 91] CASACURTA, Alexandre et al. SELA: Sistema de Edição de LAYOUTS. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 7., 1991, Capão da Canoa, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1991. 156 p. p. 137-140.
- [CAS 91] CASOTO, Andrea. **Octtools 5.1: Part I - User guide**. Berkeley: University of California, 1991. 183 p.
- [CRA 94] CRAIG, John Clark. **Microsoft Visual Basic**: versão 3. São Paulo: Makron Books, 1994.
- [DOS 91] DOSSA, Marcos K.; CLETO, Laerte D. Avaliação das Ferramentas do TENTOS. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 7., 1991, Capão da Canoa, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1991. 156 p. p. 127 - 129.
- [FRE 92] FREITAS, Demétrio; REIS, Ricardo. COTONET: Um Comparador de Netlists Formato Spice. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 7., 1992, Rio de Janeiro, RJ. **Anais...** Rio de Janeiro: SBC, 1992. 284 p. p. 81 - 93.
- [GME 92] GRUPO DE MICROELETRÔNICA. **Sistema Esqueleto**. Porto Alegre: CPGCC da UFRGS, 1992. 76 p.
- [GOM 88] GOMES, Rogério F. DARC: Um Verificador de Regras de Projeto de CI's utilizando Programação em Lógica. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 3., 1988, Gramado, RS. **Anais...** Porto Alegre: SBC, 1988. 229 p. p. 85 - 94.
- [GRE 92] GREINER, Alain; PÊCHEUX, François. ALLIANCE: A complete Set of CAD Tools for teaching VLSI Design. In: EUROCHIP WORKSHOP ON VLSI DESIGN TRAINING, 3., 1992, Grenoble, França. **Proceedings...** Grenoble:EUROHIP, 1992. p. 230-237.
- [GRO 93] GROENEVELD, Patrick; STRAVERS, Paul. **Ocean**: the Sea-of-Gates Design System. [S.l.]: Delft University of Technology, 1993. 123 p.
- [HOL 94] HOLZNER, Steven. **Visual Basic for Windows**. Rio de Janeiro: Campus, 1994.
- [MÄH 94] MÄHLMANN, Luiz G. G. **O Sistema TENTOS para MS-Windows**. Porto Alegre: CPGCC da UFRGS, 1994. (TI - 437).
- [MÄH 95] MÄHLMANN, Luiz G. G.; REIS, Ricardo A. L. Analysis of the Current State of TENTOS Systems. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 10., 1995, Porto Alegre, RS. **Proceedings...** Porto Alegre: CPGCC da UFRGS, 1995. 134 p. p. 95 - 98.
- [MÄH 96] MÄHLMANN, Luiz G. G. et al. **O Sistema TENTOS para MS-Windows v. 1.00**. Porto Alegre: CPGCC da UFRGS, 1996. 77 p. (RP - 261).

- [MON 92] MONTALVO, Luis.; BERNAL, Iván.; LEMUS, Fredy. El Ecuador en la Microeletronica. In: JORNADAS EN ENGENIERIA ELECTRICA Y ELECTRONICA, 13., 1992, Quito. **Anais...** [S.l.: s.n.], 1992. p. 31 - 39.
- [MOR 89] MORAES, Fernando G.; REIS, Ricardo A. L. EXTRALO: Um Extrator Lógico. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 5., 1989, Tramandaí, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1989. 137 p. p. 99 - 102.
- [MOR 89a] MORAES, Fernando G.; REIS, Ricardo A. L. TRAGO: TRANCA Gate-Matrix Generator. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 5., 1989, Tramandaí, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1989. 137 p. p. 103 - 106.
- [MOR 90] MORAES, Fernando G. **TRAGO: Síntese Automática de Leiaute para Circuitos em Lógica Aleatória**. Porto Alegre: CPGCC da UFRGS, 1990. 199 p. Dissertação de Mestrado.
- [MOR 90a] MORAES, Fernando G.; REIS, Ricardo A. L. Avaliação da ferramenta TRAGO. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 5., 1990, Ouro Preto, MG. **Anais...** Ouro Preto: SBC, 1990. 334 p. p.197 - 206.
- [MOR 90b] MORAES, Fernando G.; REIS, Ricardo A. L. TRAGO: Resultados. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 6., 1990, Tramandaí, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1990. 152 p. p. 24 -28.
- [MOR 90c] MORAES, Fernando G.; REIS, Ricardo A. L. EXTRALO: Extrator Lógico. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 5., 1990, Ouro Preto, MG. **Anais...** Ouro Preto: SBC, 1990. 334 p. p.167 - 176.
- [MOR 91] MORAES, Fernando G.; REIS, Ricardo A. L. **TENTOS: Gerenciador de Software para Microeletrônica**. Porto Alegre: CPGCC da UFRGS, 1991. 36 p. (RP - 155).
- [MOR 91a] MORAES, Fernando G.; REIS, Ricardo A. L. TENTOS: Gerenciador de Software para Microeletrônica. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 6., 1991, Jaguariúna, SP. **Anais...** Campinas: SBC, 1991. 221 p. p.68 - 79.
- [MOR 94] MORAES, Fernando. **Synthese Topologique de Macro-cellules en Technologie CMOS**. Montpellier: Universite Montpellier II, 1994. Tese de Doutorado.
- [OLI 94] OLIVEIRA, C.; ANIDO, M. L. TEDMOS para Windows. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE MICROELETRÔNICA, 9., 1993, Campinas, Brasil. **Anais...** São Paulo: SBMICRO, 1994. p.188 -197.
- [PET 93] PETZOLD, Charles. **Programando para Windows 3.1**. São Paulo: Makron Books, 1993.

- [PIL 95] PILGER, Luiz Oscar R. **Geração do Layout do Chip-Set nas Ferramentas OCEAN e TROPIC**. Porto Alegre: CPGCC da UFRGS, 1995. 47 p. Relatório Técnico.
- [SAC 90] SACHET, Daniel et al. ESQUELETO: Editor de Esquemas Elétricos. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 6., 1990, Tramandaí, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1990. 152 p. p. 37 - 41.
- [SOT 93] SOTO, F.; HENTZ, A.; CARRO, L. et al. CHARRUA: A symbolic editor for analog circuits. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE MICROELETRÔNICA, 8., 1993, Campinas, Brasil. **Anais...** São Paulo: SBMICRO, 1993. p.12.20 - 12.22.
- [SOT 94] SOTO, F.; CARRO, L.; SUZIM, A. Compactação de Células-folha e Montagem Hierárquica com Minimização de Conexões. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE MICROELETRÔNICA, 9., 1994, Rio de Janeiro, Brasil. **Anais...** Rio de Janeiro: SBMICRO, 1994. p.58-64.
- [STE 89] STEMMER, Marcos A.; REIS, Ricardo A. L. **EXTRIBO: Um Extrator de Circuitos**. Porto Alegre: CPGCC da UFRGS, 1989. 27 p. (RP - 103).
- [STE 89a] STEMMER, Marcos A.; REIS, Ricardo A. L. EXTRIBO: Um Extrator de Circuitos. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS, 4., 1989, Rio de Janeiro, RJ. **Anais...** Rio de Janeiro: SBC, 1989. 220 p. p. 01 - 09.
- [STE 89b] STEMMER, Marcos A. Editor de Máscaras EMA2: Interface Gráfica do Extrator. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 5., 1989, Tramandaí, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1989. 137 p. p. 111 - 114.
- [STE 89c] STEMMER, Marcos A. **Editor de Máscaras EMA2: Interface Gráfica do Extrator**. Porto Alegre: CPGCC da UFRGS, 1989. 24 p. Trabalho Individual I.
- [STE 90] STEMMER, Marcos A.; REIS, Ricardo A. L. EXTRIBO: Uma Versão Corrigida e Melhorada do Extrator Hierárquico de Circuitos. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 6., 1990, Tramandaí, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1990. 152 p. p. 34 - 36.
- [VAS 90] VASCONCELLOS, Vinícius; VILLWOCK, Alexandre. EDSMB: Editor de Símbolos. In: SEMINÁRIO INTERNO DE MICROELETRÔNICA, 6., 1990, Tramandaí, RS. **Anais...** Porto Alegre: CPGCC da UFRGS, 1990. 152 p. p. 42 - 43.
- [VLA 81] VLADIMIRESCU, A. et al. **SPICE user's guide: version 2G**. Berkeley, CA: University of California, 1981. 55 p.
- [WAG 94] WAGNER, Flávio R. Ambientes de Projeto de Sistemas Eletrônicos. In: ESCOLA DE COMPUTAÇÃO, 9., 1994, Recife, PE. **Anais...** Recife: UFPE-DI, 1994. 156p.

- [WEB 96] WEBB, Jeff. **Usando o Visual Basic 4**: O guia de referência mais completo. Rio de Janeiro : Campus, 1996.
- [WIR 94] WIRTH, Gilson I. **SGC: Um Ambiente para a Automação de Procedimentos de Caracterização e Teste**. Porto Alegre: CPGCC da UFRGS, 1994. 134 p. Dissertação de Mestrado.

Informática



UFRGS

CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

O Sistema TENTOS for Windows-Um Gerenciador de Ferramentas para Microeletrônica.

por

Luiz Gustavo Galves Mählmann

Dissertação apresentada aos Senhores:

Prof. Dr. Manuel Lois Anido (UFRJ)

Prof. Dr. Flávio Rech Wagner

Dr. Fernando Gehm Moraes

Prof. Tiaraju Vasconcelos Wagner

Vista e permitida a impressão.
Porto Alegre, 18/10/96.

Prof. Dr. Ricardo Augusto da Luz Reis,
Orientador.

Prof. Flávio Rech Wagner
Coordenador do Curso de Pós-Graduação
em Ciência da Computação - CPGCC
Instituto de Informática - UFRGS