

Manual de Uso do NET 3.0

por

**Roberto da Silva Bigonha
Mariza Andrade da Silva Bigonha**

Relatório Técnico RT 011/89

**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
INSTITUTO DE CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DE MINAS GERAIS**

1989

O desenvolvimento do Sistema NET foi financiado pelo Convênio
UFMG/Telebrás (CPqD) .

SUMÁRIO

Sistema NET extrai a NETLIST a partir de especificações estruturais de circuitos codificadas na linguagem HILO-2. O sistema lê e processa descrições HILO-2, produzindo a NETLIST, referências de pinos e de sinais e identificando "clusters" de células SOFT através da expansão dos macro-circuitos, os quais definem circuitos em termos de células-padrão e outros circuitos.

ABSTRACT

The NET SYSTEM extracts the NETLIST configuration from structural circuit specifications encoded in the HILO-2 circuit descriptor language. The system inputs and processes HILO-2 descriptions, producing the circuit NETLIST, cross-references of pins and signals, and identifying all SOFT cell clusters by expanding macrocircuits, which are defined in terms of standard cells or other macrocircuits.

SUMÁRIO

1	Introdução	1
2	Disquetes de Distribuição	2
2.1	Disquete do Fonte	2
2.2	Disquete do Objeto	3
3	Instalação e Ativação do Sistema	3
4	Bibliotecas	4
5	Linguagem Intermediária	6
6	Operação do Sistema NET	10
6.1	O Menu	10
6.2	Funções do Sistema NET	12
6.2.1	Compilar Macro-Circuitos	12
6.2.2	Compilar Soft Cells	13
6.2.3	Compilar Standard Cells	13
6.2.4	Extrair NETLIST	13
6.2.5	Exiba Erros Encontrados	14
6.2.6	Listar NETLIST	14
6.2.7	Listar Referências Cruzadas	15
6.2.8	Listar Clusters	16
6.2.9	Listar Diretório de Trabalho	16
6.2.10	Listar Outro Diretório	17
6.2.11	Remover Circuitos	17
6.2.12	Trocar Nome de Circuito	18
6.2.13	Listar Circuito-objeto	18
6.2.14	Criar Biblioteca	19
6.2.15	Copiar Bibliotecas	19
6.2.16	Abrir Biblioteca	20
6.2.17	Fechar Biblioteca	20
6.2.18	Apagar Biblioteca	20
6.2.19	Limpar Ambiente	20
6.2.20	Definir Ambiente	20
6.2.21	Apagar Batch File	21

6.2.22 Definir Batch File	22
6.3 Execução no Modo Batch	22
7 Mensagens de Erro	22
8 Arquivos de Saída	24
8.1 Arquivo NETLIST	24
8.2 Arquivo de Referências Cruzadas	25
8.3 Arquivo de Clusters	26
9 Exemplos de Circuitos, Netlist e Referências	26
10 Referências	32
Apêndice A: Sintaxe da linguagem HILO-2	33

1 INTRODUÇÃO

O Sistema NET é um gerador de "netlist" através do processamento de descrições estruturais de circuitos. O sistema aceita como entrada descrições de circuitos na linguagem HILO-2 [GenRad, 1984], analisa sintaticamente estas descrições, compilando-as para uma linguagem intermediária e armazenando as descrições compiladas, também chamadas circuitos-objeto, em bibliotecas.

Posteriormente, mediante ativação de funções específicas do sistema, essas descrições compiladas podem ser "executadas" no sentido de gerar a "netlist" e referências cruzadas das conexões dos circuitos e os "clusters" de instâncias de componentes de células SOFT.

A "netlist" de um circuito é a descrição de sua conectividade em termos de fios, sinais, pinos e componentes. A "netlist" mostra os fios que estão conectados aos pinos de cada instância das células-padrão presentes no circuito.

Referências cruzadas são formas redundantes de NETLIST classificadas em função de sinais, pinos e componentes.

Células SOFT são células que não possuem leiaute único, sendo formada por células-padrão.

O Sistema NET foi implementado em equipamento de família IBM-PC utilizando-se o SIC [Bigonha, 1986], que é um sistema de geração semi-automática de compiladores baseado na linguagem Turbo Pascal versão 3.0 [Borland 1985]. Parte do sistema foi escrito em SIC e o restante em Pascal TURBO 3.0.

Para se obter o sistema NET em forma executável parte do conjunto de programas que o compõem deve ser submetido ao SIC, o qual produz como resultado um conjunto de programas em TURBO PASCAL 3.0, podendo então ser compilado juntamente com os demais.

A dependência do sistema ao TURBO PASCAL 3.0 está localizada nos

seguintes pontos:

- a) entrada de dados via teclado;
- b) manipulação de tela, especificamente com relação ao uso de funções GOTOXY, CLRSCR, etc;
- c) estrutura dos nomes de arquivo, procedimentos ASSIGN e RESET;
- d) arquivo de acesso direto: uso da função SEEK;
- e) recursos da linguagem Turbo Pascal para determinar se arquivos existem ou não;
- f) estrutura e facilidades para OVERLAY;
- g) diretiva de compilação INCLUDE.

No caso de mudança de compilador PASCAL, estes pontos devem ser analisados cuidadosamente. O restante do "software" segue o PASCAL descrito em [Jensen, 1974].

O conjunto de programas contém aproximadamente 5.000 linhas de código SIC, que se transformam em cerca de 7.500 linhas de PASCAL. O Sistema está configurado para ser executado sob o MS-DOS em microcomputadores da família IBM-PC com um mínimo de 256K de memória RAM e pelo menos um acionador para disquete 5 1/4".

2 DISQUETES DE DISTRIBUIÇÃO

2.1 Disquete do Fonte

Os arquivos contendo os programas que compõem o fonte do sistema NET são:

a) NET.PAS

Programa principal do sistema em TURBO PASCAL 3.0.

b) GLOBAL.PAS

Declarações das variáveis, tipos e constantes globais do sistema.

c) NETBAS1.PAS, NETBAS2.NET e NETBAS3.NET

Conjunto de procedimentos auxiliares em TURBO PASCAL 3.0 que tratam principalmente da entrada e saída do sistema.

d) NETOVR1.PAS e NETOVR2.PAS

Conjunto de procedimentos em TURBO PASCAL 3.0 que implementam as funções desempenhadas pelo Sistema NET. Todos os procedimentos declarados neste arquivo participam do esquema de OVERLAY utilizado pelo NET.

e) NETGEN.PAS

Conjunto de procedimentos auxiliares, em TURBO PASCAL 3.0, utilizado pelo compilador de descrições HILO-2 para a linguagem intermediária definida na Seção 5.

f) NETCOMP.SIC

Núcleo do compilador de HILO-2 para a linguagem intermediária. Esta parte do Sistema NET está escrita em SIC; podendo ser automaticamente convertida para TURBO PASCAL 3.0 através do sistema de implementação de compiladores SIC .

g) NETCOMP.DCL, NETCOMP.SEM, NETCOMP.PRO e NETCOMP.PRS

Compilador HILO-2 gerado pelo SIC.

2.2 Disquete do Objeto

Os arquivos que compõem o programa objeto do Sistema NET são:

a) NET.COM

Módulo principal do sistema.

b) NET.000, NET.001,, NET.010

Arquivos auxiliares contendo os códigos dos procedimentos que participam do esquema de OVERLAY.

3. ATIVAÇÃO DO SISTEMA

O NET pode ser executado interativamente ou em "batch". Para trabalhar interativamente o usuário deve chamar NET sem especificar parâmetros na linha de comandos. Imediatamente após ativado, um menu das funções disponíveis é exibido. Por outro lado, a execução em modo "batch" requer a especificação do nome de um arquivo, que chamamos de batch file, contendo os parâmetros e as funções a serem efetuadas sem intervenção do operador. O "batch file" tem sempre a extensão .BTC. O usuário

deve usar primeiramente usar o NET na forma interativa para preparar o batch file.

4. BIBLIOTECAS

As células-padrão, células SOFT e macrocircuitos definidos em HILO-2, e compiladas pelo NET, são armazenados em uma forma interna em bibliotecas, possibilitando o seu uso em outras descrições de circuitos e também facilitando o manuseio de grande número de descrições HILO-2.

Uma biblioteca é formada por dois arquivos de mesmo nome principal e extensões .DIR e .BBL, que denotam, respectivamente, o diretório e o corpo da biblioteca.

Arquivos com extensão .DIR possuem seus registros organizados conforme uma árvore binária de ordenação. Cada registro, que é um nodo da árvore, está associado a um circuito e contém as seguintes informações:

a) NOME

Nome do circuito.

b) TIPO

Tipo do circuito obtido da descrição HILO-2, i.e., ICB, PCB ou CCT.

c) STATUS

Identifica o "status" do circuito. Pode ser:

.MACRO

Trata-se de um macrocircuito que será usado por algum outro.

.PRINCIPAL

Trata-se de um macrocircuito que não é chamado por nenhum outro e a partir do qual serão feitas as expansões de macros para fins de obtenção da NETLIST.

.PADRÃO

Trata-se de uma célula-padrão.

. SOFTCELL

Trata-se de uma célula SOFT.

d) NPINOS

Número de pinos presentes na interface do macrocircuito ou célula-padrão.

e) ENDEREÇO

Endereço do corpo da macro no arquivo de extensão .BBL correspondente.

f) TAMANHO

Tamanho do corpo da macro.

g) ESQUERDO, DIREITO

Elos da estrutura da árvore.

Os nodos da árvore são armazenados de forma que, para um nodo de endereço p, tenha-se:

```
se ESQUERDO[p] <> NULO, NOME[ESQUERDO[p]] < NOME[p]
se DIREITO[p] <> NULO, NOME[DIREITO[p]] >= NOME[p]
```

Em um mesmo drive podem coexistir várias bibliotecas. As operações de geração de NETLISTs são executadas na presença de uma ou mais bibliotecas. Ao conjunto de bibliotecas simultaneamente acessíveis denomina-se AMBIENTE. O ambiente que pode ser definido, sobrevive às execuções do NET, sendo gravado no arquivo NET.AMB no drive default do sistema e recuperado automaticamente sempre que o NET for ativado. O Sistema NET possui funções para criar, copiar, abrir e apagar bibliotecas e para definir e administrar ambientes.

Em geral, as bibliotecas deverão conter as células-padrão usadas nas descrições de circuitos e as células SOFT e os macrocircuitos de uso mais frequentes.

Durante o processamento de certas funções do Sistema NET, a biblioteca de trabalho deve estar aberta, i.e., deve ter uma cópia de seu diretório na memória. Caso não haja biblioteca aberta ao se iniciar uma dessas funções, o NET solicita ao usuário o nome da biblioteca a ser usada, abrindo-a convenientemente. As demais bibliotecas do ambiente são automaticamente abertas para leitura.

Sempre antes de se iniciar uma função que pode causar alguma

modificação na cópia do diretório da memória, o Sistema NET dá ao usuário a oportunidade de atualizar a cópia do disquete, perguntando se a cópia da memória deve ou não ser salva. Especificamente, a pergunta é:

Biblioteca XXXXX não foi salva ? Salva (S/N) ? █

Sendo XXXXX o nome de biblioteca. O usuário pode responder S, s, N, ou n conforme a alternativa desejada. O símbolo █ denota a posição do cursor.

5. LINGUAGEM INTERMEDIÁRIA

Descrições HILO-2 de circuitos são traduzidas para a linguagem de uma máquina virtual que facilita a expansão dos circuitos para produzir a NETLIST. Esta máquina possui as seguintes facilidades:

- a) Pilha ASTACK de registros de ativação de macros. Cada registro de ativação possui área para os nomes dos parâmetros passados à macro e das conexões geradas internamente. Os registros de ativação são alocados no momento de chamada do circuito e liberados no seu retorno.
- b) Registrador at que aponta para o topo da pilha ASTACK.
- c) Registrador b que aponta para base do registro de ativação mais ao topo da pilha ASTACK. O nome do i-ésimo parâmetro passado à macro está localizado em:

ASTACK[b + i - n]

onde n é o número de pinos do circuito. Nomes gerados internamente são armazenados nas posições b + 1, b + 2 etc.

- d) Pilha RSTACK de controle de ativação de macros. Para cada macro chamada empilham-se o valor corrente do registrador b antes da chamada, o endereço de retorno (e.r.) da chamada e o nome da instância do circuito chamado.
- e) Registrador rt que aponta para o topo de RSTACK.

Na descrição das instruções da máquina virtual apresentadas a seguir são usadas as seguintes convenções:

<n1>

Campo de um "byte" denotando um valor inteiro no intervalo [-128, 127].

<n2>

Campo de dois "bytes" denotando um valor inteiro no intervalo [-32768, 32767].

<m2>

Idem <n2>.

<nome>

Campo de 15 "bytes" representando o nome de fio, pino ou circuito.

<instancia>

Campo de 15 "bytes" representando o nome de uma instância de circuito.

Toda instrução possui um campo de operação de 1 "byte" de comprimento possivelmente seguido de 1 ou mais "bytes" de operandos. As instruções são:

NOP <n2>

Nada faz; serve para armazenar o número da linha (<n2>) com o objetivo de facilitar a produção de mensagens que fazem referências ao texto fonte.

CSCALAR1 <n1>

Empilha em ASTACK o nome que estiver armazenado em ASTACK [b + <n1>]. O valor <n1> é chamado de "offset" do valor a ser empilhado.

CSCALAR2 <n2>

Mesma função que CSCALAR1, porém o "offset" <n2> é de 2 "bytes".

CNOVO <n2> <nome>, <nome>, ..., <nome>

Gera <n2> novos nomes de sinal, empilhando-os em ASTACK. A instrução é usada para criar nomes únicos para sinais locais a descrições de circuito. Os <n2> nomes destes sinais locais são os operandos <nome> da instrução.

CHAME <n2>, <nome>, <instância>

Ativa a execução, para produção de NETLIST, do circuito de <n2> pinos, de nome dado por <nome> e correspondendo à instância <instância>. A execução da instrução CHAME cria um novo registro de ativação no topo de ASTACK, empilhando informação de controle em RSTACK.

A sequência de chamada de uma macro inicia-se com o empilhamento

(passagem) dos parâmetros, na ordem em que ocorrem na chamada, através das instruções CSCALAR1, CSCALAR2, CBRANCO, CVECTOR etc. A sequência de chamada termina com a instrução CHAME. Por exemplo, supondo que os pinos Q0, Q0N, D0, CL e CLEAR tenham "offsets" 300, -2, 0, 3 e -1 respectivamente. A chamada

DF115 FFO(Q0,Q0N,D0,CL,CLEAR)

é compilada na seguinte sequência de instruções:

CSCALAR2	300	(empilha Q0)
CSCALAR1	-2	(empilha Q0N)
CSCALAR1	0	(empilha D0)
CSCALAR1	3	(empilha CL)
CSCALAR1	-1	(empilha CLEAR)
CHAME	5, DF115, FFO	

RET <n2>

Promove o retorno de uma chamada a uma macro. Esta instrução é responsável pela liberação das áreas ocupadas por valores empilhados pela sequência de chamada correspondente e pela própria macro. O operando <n2> fornece o número de parâmetros empilhados na chamada ao macrocircuito.

RETP <n2>

Mesmo que RET, exceto que promove o retorno de células-padrão.

PADRÃO <nome>

Instrução que identifica o início de uma definição de célula-padrão de nome <nome>. Toda célula-padrão deve iniciar com esta instrução e terminar com RETP.

PINO <n2>, <nome>

Instrução que estabelece a associação de um parâmetro, através de seu "offset" <n2>, passado à célula-padrão e nome do pino correspondente. O "offset" do i-ésimo parâmetro é sempre i-n, sendo n o número total de parâmetros.

Por exemplo:

PADRÃO	AA025
PINO	-2, Q
PINO	-1, A
PINO	0, C
RETP	3

descreve as associações de pinos a serem produzidas na NETLIST para a célula-padrão AA025, i.e., o pino Q será associado ao nome em ASTACK [b - 2], o pino A com ASTACK [b - 1] e C com ASTACK[b]. Do ponto de vista do Sistema NET, células-padrão são atômicas, i.e., não são estruturadas em função de outras macros.

CBRANCO

Empilha um "string" de brancos na pilha ASTACK. "String" de branco é sempre o nome de parâmetro omitido em chamada de qualquer macro.

CZ

Empilha o nome interno gerado pelo NET para a constante Z.

C0

Empilha o nome interno gerado pelo NET para a constante 0.

C1

Empilha o nome interno gerado pelo NET para constante 1.

CVECTOR <m2>, <n2>

Empilha em ASTACK os | <n2> | nomes armazenados na área de ASTACK compreendida entre

b + <m2> até b + <m2> + <n2>

começando-se em b + <m2>. O valor de <n2> pode ser negativo.

CNOMES <nome>

Empilha em ASTACK o nome dado pelo operando <nome>.

CNOMEV <nome>, <m2>, <n2>

Constrói | <n2> | novos nomes de sinal concatenando o nome <nome> com os inteiros que vão de <m2> a <m2> + <n2>, excluindo o último valor, e colocando cada inteiro dentro de um par de colchetes. Por exemplo, a instrução

CNOMEV A, 10, 3

empilha A[10], A[11], A[12] na pilha ASTACK; enquanto que a instrução

CNOMEV A, 10, -3

empilha A[10], A[9], A[8].

6 OPERAÇÕES DO SISTEMA NET

6.1 Execução Interativa

A ativação do programa NET na forma interativa inicia-se com a exibição, na tela do microcomputador, do seguinte menu de funções:

SISTEMA NET versao 3.0 -- CPqD/UFMG

```

=====
| A = Compilar MACROCIRCUITS      L = Trocar nome de circuito |
| B = Compilar SOFT CELLS          M = Listar circuito-objeto |
| C = Compilar STANDARD CELLS      N = Criar biblioteca        |
| D = Extrair netlist              O = Abrir biblioteca        |
| E = Exiba erros encontrados      P = Copiar biblioteca     |
| F = Listar netlist               Q = Fechar biblioteca     |
| G = Listar cross-ref              R = Apagar biblioteca     |
| H = Listar clusters              S = Limpar ambiente       |
| I = Listar diretorio trabalho    T = Definir ambiente     |
| J = Listar outro diretorio       U = Apagar batch file    |
| K = Remover circuitos            V = Definir batch file    |
|                                   [ESC = Finalizar sessão]    |
=====
>■                                   Biblioteca fechada

```

O símbolo ■ denota a posição do cursor na tela.

À direita do cursor, na mesma linha, tem-se inicialmente a mensagem "Biblioteca Fechada". Após execução da função de Abrir biblioteca, o NET passa a exibir nesta posição o nome da biblioteca em uso.

Para selecionar cada uma das 22 funções relacionadas no menu, deve-se teclar a letra (maiúscula ou minúscula) em frente da opção desejada, o que fará o menu desaparecer e dará início a execução da função escolhida. Após o término de toda função, o menu é sempre re-exibido. As mensagens produzidas durante a execução das funções do NET são encaminhadas ao arquivo NET.ERR, que pode ser inspecionado via função Exiba Erros Encontrados.

Toda função ao ser ativada interage com o usuário com o intuito de obter dados, como nome de arquivos e "drive" de saída de listagens, necessários ao seu processamento.

Há interações que requerem do usuário apenas a digitação de um caractere. Nestes casos, o NET o processa imediatamente o valor digitado e avança na execução de função. No caso de o usuário ter que digitar nomes de arquivos ou de macrocircuitos, o processamento só continua após a entrada de <CR> (carriage return).

Erros de digitação podem ser corrigidos via a tecla BACKSPACE (CTL-H), que apaga o nome digitado, letra por letra, da posição do cursor para a esquerda, ou através do código CTL-U, que apaga todo o nome.

Em funções do NET que solicitam nomes, sejam de arquivos, macrocircuitos ou bibliotecas, a digitação de apenas <CR> é tratada como cancelamento da função ou que a opção associada não deverá ser usada.

A tecla ESC ("escape") deve ser utilizada sempre que se desejar interromper uma função em execução, retornar ao menu exibido por último ou devolver o controle da execução ao sistema operacional do microcomputador.

Sempre que a tecla ESC é pressionada durante a execução de uma função, a mensagem

Aborta tarefa ? (S/N) : ■

é exibida para dar ao operador a opção de interromper a tarefa ou

retomá-la.

6.2 Funções do Sistema NET

6.2.1 Compilar MACROCIRCUITS

Esta função compila descrições de circuitos em HILO-2 para circuitos-objeto na linguagem da máquina virtual apresentada na Seção 5. O processo de compilação inclui análise sintática da entrada, com emissão de mensagens de erro, e listagens dos textos compilados. Os circuitos-objeto produzidos são incluídos na biblioteca de trabalho. No início da compilação, o NET pede ao usuário:

Arquivo de macroCIRCUITOS a serem COMPILADOS : ■

Em resposta, deve-se digitar o nome do arquivo que possua extensão .HIL contendo as descrições HILO-2 a serem compiladas. A extensão pode ser omitida; NET automaticamente a inclui. Uma resposta com apenas <CR> cancela a função.

Em seguida, NET solicita:

Macro principal : ■

Em resposta, deve-se informar o nome da descrição de circuito no arquivo fonte que é o macrocircuito principal, a partir do qual a expansão para extração de NETLIST será feita. Caso o arquivo não contenha macrocircuito principal, o usuário deve responder com <CR> apenas.

A seguir deve-se informar o destino das listagens em resposta a:

Destino da listagem (A,B,C,D-disco,T=tela,I=impressora,
N=sem listagem) : ■

Os valores indicados tem o seguinte significado:

A - destino da listagem do fonte é o arquivo A:xxx.PRN, onde xxx é o nome principal do arquivo fonte informado acima.

B - idem para "drive" B.

C - idem para "drive" C.

D - idem para "drive" D.

T - destino da listagem do fonte é a tela do monitor de vídeo.

I - destino da listagem do fonte é a impressora do sistema.

N - a listagem deve ser omitida.

6.2.2 Compilar SOFT CELLS

Esta função é semelhante a compilar macrocircuitos, exceto que neste caso as descrições de circuitos compiladas são marcadas na biblioteca como sendo do tipo SOFTCELLS.

6.2.3 Compilar STANDARD CELLS

Esta função é semelhante a compilar macrocircuitos, exceto que todos os macrocircuitos encontrados na entrada são tratados como células-padrão, sendo os seus corpos substituídos por sequência de instruções PINO que relacionam pinos a parâmetros (vide Seção 5). O diálogo com o usuário é análogo ao descrito em 6.2.1.

6.2.4 Extrair NETLIST

Esta função expande o macrocircuito principal, realizando chamadas ou expansões de macros, produzindo a NETLIST, as referências cruzadas de instâncias de circuitos e de sinais e o arquivo de cluster. Os resultados desta função são armazenados em arquivos TEXT que tem como nome principal o da macro principal especificada e as extensões .NET, .RFC, .RFS e .CLT para os arquivos de NETLIST, de referências cruzadas de instâncias e de sinais e dos "clusters" de instâncias de componentes de células SOFT, respectivamente.

Inicialmente, para execução da função, o NET solicita:

Circuito principal para extracao de NETLIST : ■

Em resposta, deve-se digitar o nome do circuito principal, o

qual deve necessariamente ser um circuito catalogado na biblioteca de trabalho com STATUS = PRINCIPAL.

6.2.5 Exiba Erros Encontrados

Esta função exiba na tela ou impressora os erros encontrados na última função executada.

6.2.6 Listar Netlist

Esta função lista na tela ou na impressora a NETLIST gravada em arquivo com extensão .NET. Inicialmente, NET solicita:

Macro principal da NETLIST : ■

Em resposta, o usuário deve informar o nome da macro principal especificada quando da ativação da função para extrair netlist. O arquivo de onde a NETLIST será obtida tem nome XXX.NET onde XXX é o nome da macro principal especificada.

Exemplo de listagem de NETLIST:

NETLIST EM C:MAIN.NET

...

4	VETOR2
X[1]	U[44]
X[2]	U[45]
X[3]	U[46]
X[4]	U[47]
X[5]	R[2]
X[6]	R[3]
X[7]	R[4]
X[8]	R[5]
X[9]	R[6]
X[10]	K
#	
5	DF115
Q	1
QN	6
D	7

C	8
RN	9
#	
6	DF115
Q	2
QN	4
D	7
C	8
RN	9
#	

...

6.2.7 Listar Referências Cruzadas

Esta função lista, na tela ou na impressora, referências cruzadas de instâncias e de sinais coletados pelo NET para um dado circuito principal. Estas referências são sempre armazenadas em arquivos com nome XXX.RFC e XXX.RFS, onde XXX é o nome de um circuito principal. Inicialmente, NET solicita:

Macro principal das referências : ■

Se for digitado um nome diferente de branco, a função é executada.

Exemplo de listagem de referência cruzadas:

REFERÊNCIAS CRUZADAS DE INSTÂNCIAS EM C:MAIN.RFC

1	MAIN.JKL.A1
2	MAIN.JKL.A2
3	MAIN.JKL.A3
4	MAIN.JKL.A4
5	MAIN.JKL.A5
6	MAIN.JKL.A6[1]
7	MAIN.JKL.A6[0]
8	MAIN.JKL.A9[2]
9	MAIN.JKL.A9[3]

10	MAIN.JKL.A9[4]
11	MAIN.JKL.A9[5]
12	MAIN.JKL.A9[6]
13	MAIN.JKL.A11[0]
14	MAIN.JKL.A11[1]
15	MAIN.JKL.A11[2]
16	MAIN.JKL.A11[3]
17	MAIN.JKL.A11[4]

REFERÊNCIAS CRUZADAS DE SINAIS EM C:MAIN.RFS

ZZZ	MAIN.ZZZ
T[0]	MAIN.T[0]
T[1]	MAIN.T[1]
T[2]	MAIN.T[2]
T[3]	MAIN.T[3]
T[4]	MAIN.T[4]
T[5]	MAIN.T[5]
T[6]	MAIN.T[6]
T[7]	MAIN.T[7]
T[8]	MAIN.T[8]
T[9]	MAIN.T[9]
K	MAIN.K
R[9]	MAIN.R[9]
R[8]	MAIN.R[8]
R[7]	MAIN.R[7]
R[6]	MAIN.R[6]
R[5]	MAIN.R[5]
R[4]	MAIN.R[4]
R[3]	MAIN.R[3]
R[2]	MAIN.R[2]
R[1]	MAIN.R[1]
R[0]	MAIN.R[0]
L	MAIN.L

...

6.2.8 Listar Clusters

Lista os "clusters" de instâncias de células SOFT e seus grupos identificados durante a geração da netlist do circuito.

6.2.9 Listar Diretório de trabalho

Esta função lista na tela o diretório da biblioteca de trabalho usando a cópia da memória. O diretório listado tem o formato do seguinte exemplo:

DIRETÓRIO : xxx.DIR				
MACRO	NUMPINO	TAMANHO	STATUS	TIPO
AA025	3	500	MACRO	CCT
AA065	3	400	PADRÃO	CCT
FF115	2	100	SOFT	CCT

As informações listadas por diretório são:

MACRO

Nome da macro.

NUMPINO

Número de pinos na interface.

TAMANHO

Número de "bytes" ocupados pelo circuito-objeto na biblioteca.

STATUS

PADRÃO, PRINCIPAL, MACRO ou SOFT.

TIPO

IC, PCB, ou CCT.

6.2.10 Listar Outro Diretório

Esta função lista na tela o diretório de uma qualquer biblioteca armazenada no disquete do "drive" corrente. O formato da saída é o mesmo da Seção 6.2.9. Interage com o usuário, pedindo o nome da biblioteca a ser listada, da seguinte forma:

Nome da Biblioteca : ■

6.2.11 Remover Circuitos

Esta função interage com o usuário pedindo nomes de macrocircuitos ou células-padrão a serem removidas da cópia do diretório corrente na memória. A cópia em disco poderá ser atualizada uma única vez quando outras operações que alteram o diretório forem ativadas, ou quando a sessão for finalizada.

O NET pede nomes de circuitos a serem removidos da seguinte forma:

Circuito a ser removido : █

até que o usuário responda com apenas <CR>, quando a função termina.

6.2.12 Trocar Nome de Circuito

Esta função serve para trocar nomes de circuitos da biblioteca corrente.

O Sistema NET interage com o usuário pedindo o nome velho e o nome novo dos circuitos desejados:

Nome velho do circuito : █

e após resposta do usuário com um valor diferente de <CR>:

Nome novo do circuito : █

Esta interação ocorre até que um nome velho igual a branco ou apenas <CR> seja digitado.

Caso o novo nome já exista no diretório, o sistema pergunta ao usuário como proceder:

Novo nome já existe. Apaga ? (S/N) : █

6.2.13 Listar Circuito-objeto

Esta função solicita o nome de um circuito catalogado na biblioteca de trabalho e lista o seu corpo na forma compilada para a

linguagem de máquina virtual da Seção 5. Opcionalmente a listagem pode ser direcionada à tela ou à impressora. Inicialmente, o NET solicita:

Circuito a ser LISTADO : ■

Após o circuito ser encontrado na biblioteca, tem-se a solicitação:

Saída da listagem ? (T = Tela, I = Impressora) : ■

EXEMPLO:

CCT	MAIN			
NOP	5			
CNOMES	ZZZ			
CNAMEV	T	0	10	
CNOMES	K			
CNAMEV	R	9	-10	
CNOMES	L			
CNAMEV	U	40	11	
CVECTOR	2	10		
CSCALAR1	12			
CVECTOR	13	10		
CSCALAR1	23			
CVECTOR	24	11		
CHAME	33	VETOR		JKL
RET	1			

6.2.14 Criar Biblioteca

Esta função cria uma nova biblioteca com o nome fornecido pelo usuário em resposta à solicitação:

Biblioteca a ser CRIADA : ■

O nome de biblioteca não deve possuir extensão. Caso a biblioteca já exista, esta função consulta ao usuário como proceder:

Biblioteca já existe. Apaga ? (S/N) : ■

6.2.15 Copiar Bibliotecas

Esta função solicita ao usuário os nomes das bibliotecas a copiar e a receber a cópia. A biblioteca que receberá a cópia será criada automaticamente. Inicialmente, NET pede:

Biblioteca a COPIAR : ■

Se a biblioteca especificada não existir, o NET produz uma mensagem neste sentido e finaliza a função. Do contrário, o NET solicita:

Biblioteca a ser CRIADA : ■

6.2.16 Abrir Biblioteca

Esta função abre a biblioteca, criando uma cópia, na memória, do diretório da biblioteca especificada pelo usuário em resposta a:

Biblioteca a ser ABERTA : ■

Caso a biblioteca especificada não exista, uma mensagem neste sentido será produzida.

6.2.17 Fechar Biblioteca

Esta função fecha a biblioteca de trabalho.

6.2.18 Apagar Biblioteca

Esta função remove os arquivos .DIR e .BBL da biblioteca especificada em resposta a:

Biblioteca a ser REMOVIDA : ■

A remoção é definitiva e executada imediatamente sempre que solicitada.

6.2.19 Limpar Ambiente

Esta função serve para remover do ambiente de execução do NET todos os nomes de biblioteca previamente definidos. Deve ser utilizado quando deseja-se criar um ambiente inteiramente novo.

6.2.20 Definir Ambiente

Esta função ativa o administrador de ambientes. Por exemplo, ao se ativar esta função num ambiente, definido pelo arquivo NET.AMB, em que TB15CELL, SC2U e SCORE15 são as bibliotecas acessíveis e TB15 a biblioteca de trabalho, o seguinte menu de funções será exibido:

Sistema NET versão 3.0 - CPqD/UFMG

BIBLIOTECAS CORRENTEMENTE ACESSIVEIS	
TB15	<== Biblioteca de trabalho
TB15CELL	
SC2U	
SCORE15	
Funcao desejada :	
1-	Excluir uma biblioteca do ambiente
2-	Excluir todas bibliotecas do ambiente
3-	Abrir a biblioteca de trabalho
4-	Incluir uma biblioteca na ambiente
ESC-	Retorne ao menu principal

> █

Durante a operação de geração de NETLISTs, a busca por um circuito qualquer no ambiente é feita pesquisando-se em primeiro lugar na biblioteca de trabalho; e caso não seja encontrado nesta, procura-se nas demais bibliotecas do ambiente pela ordem em que são listadas pela presente função. Desta forma, se um circuito ocorre em mais de uma biblioteca, somente a primeira ocorrência, de acordo com o método de pesquisa acima, será considerada.

Somente bibliotecas existentes podem ser incluídas num ambiente. Excluir uma biblioteca do ambiente apenas significa retirar seu nome da lista de bibliotecas acessíveis. Para efetivamente apagar uma biblioteca deve-se usar a função própria do menu principal.

A função 3 listada no menu acima produz o mesmo efeito que a função correspondente no menu principal.

6.2.21 Apagar Batch File

Esta função limpa o arquivo que contém os parâmetros para execução do NET no modo "batch". Os arquivos "batch" têm sempre o nome do arquivo de entrada para a compilação do NET e extensão

6.2.22 Definir Batch File

Esta função permite a especificação dos parâmetros para execução do NET na modalidade "batch". Supondo que o arquivo "batch" tenha sido previamente definido, o seguinte menu é exibido pela ativação desta função :

Sistema NET 3.0 - CPqD/UFGM

```

=====
| Parametros para execucao em BATCH
|=====
1 - Arquivo de entrada para compilacao ..... : TB15
2 - Tipo das macros no arquivo de entrada ... : MACROCIRCUITOS
3 - Nome do macrocircuito principal ..... | main
4 - Biblioteca de trabalho ..... : bbl
5 - Destino das listagens ..... : Tela
6 - Apagar arquivo Batch TB15.BTC
=====
Item a ser alterado/ESC :
```

O usuário pode então alterar os parâmetros acima, sendo possível criar vários "batch files" definindo novos arquivos para entrada de compilação. Para cada uma destas entradas é criado um arquivo batch.

Os parâmetros especificados são gravados no arquivo Batch que sempre tem o nome do arquivo de entrada para compilação com extensão .BTC.

6.4 Execução na forma BATCH

Para iniciar uma execução do NET na modalidade "batch" deve-se especificar o nome do batch file na linha do comando, como por exemplo,

NET TB15

para que os parâmetros armazenados no arquivo TB15.BTC sejam utilizados.

Na execução em "batch", o NET automaticamente abre as bibliotecas necessárias, e que foram especificadas no ambiente, compila os macrocircuitos do arquivo de entrada especificado e, se o circuito principal tiver sido especificado no batch file, gera também a netlist correspondente.

7 MENSAGENS DE ERRO

As mensagens de erro que podem ser produzidas pelo NET são:

1. "Transbordamento da tabela de símbolos."
2. "Esgotou espaço disponível de células."
3. "Esgotou espaço disponível de instabela de símbolos." 2. "Esgo
14. "Número de parâmetros errado no circuito:"
15. "Instrução inválida."
16. "Transbordamento da pilha de parâmetros."
17. "Transbordamento da pilha de controle."
18. "Diretório não foi encontrado:"
19. "Biblioteca não foi encontrada:"
20. "Arquivo não encontrado:"
21. "Memória para compilação de circuitos esgotada."
22. "Biblioteca não está aberta:"
23. "Circuito não existe:"
24. "Erro de sintaxe."
25. "Indexação inválida de elemento escalar:"
26. "Circuito dado deve ser principal"
27. "Circuito principal ativado como macro"

8 ARQUIVOS DE SAÍDA

A função "Extrair netlist" gera três arquivos de nomes XXX.NET, XXX.RFC, XXX.RFS e XXX.CLT, onde XXX é o nome do circuito principal associado. Estes arquivos são do tipo TEXT, no sentido PASCAL, isto é, são estruturados como uma sequência de linhas separadas pelo par <CR><LF> (carriage return, line feed).

8.1 Arquivo de Netlist

Arquivos de NETLIST são formados por grupo de linhas terminadas por uma contendo apenas "#". Cada linha de um grupo contém um par de nomes, armazenados em campos de 15 caracteres. A primeira linha de cada grupo armazena o nome de uma célula-padrão precedido pelo nome de uma de suas instâncias. As demais linhas, relacionam os nomes dos pinos da célula-padrão com os fios que lhe estão conectados nesta instância.

O último grupo termina excepcionalmente, por duas linhas, cada uma contendo apenas o símbolo "#". Exemplo:

```

...
I#9          DF115
Q            11
QN           16
D            7
C            8
RN           9

```

```

...
#
I#17         DF115
Q            5
QN           21
D            7
C            8
RN           9
#

```

#

8.2 Arquivos de Referências Cruzadas

Os arquivos com extensão .RFC são constituídos de linhas contendo o nome interno de uma instância (com 15 caracteres) seguido de uma sequência de nomes, separados por ".", das instâncias de circuitos que foram ativados até o ponto considerado.

Para cada instância de circuito ativado, um novo nome interno é gerado e associado com a ativação.

Exemplos:

```
1          MAIN.JKL.A1
2          MAIN.JKL.A2
...
17         MAIN.JKL.A11[4]
```

Os arquivos com extensão .RFS têm a mesma estrutura dos .RFC. Neste caso, cada linha associa o nome de um sinal que pode ser encontrado no arquivo NETLIST com o seu correspondente na descrição HILO-2.

EXEMPLO:

```
...
K          MAIN.K
U[40]      MAIN.U[40]
10         MAIN.JKL.NF[6]
...
```

8.3 Arquivo de Clusters

O arquivo de "clusters" de componentes de células SOFT relaciona os nomes qualificados das instâncias destes componentes com o seu grupo de ativação da célula SOFT. Os grupos são identificados pelo valor de um contador que é incrementado toda vez que se faz a expansão de uma célula SOFT.

Exemplo :

PRINCIPAL.FLIP.GMOX18	1
PRINCIPAL.FLIP.GMOX19	1
PRINCIPAL.FLIP.GMOX20	1
PRINCIPAL.FLIP.MOX24	1
PRINCIPAL.FLIP.GMOX26	1
PRINCIPAL.FLIP.GMOX27	1
PRINCIPAL.MUX.MOX13	2
PRINCIPAL.MUX.MOX8	2
PRINCIPAL.MUX.MOX9	2

9 EXEMPLOS DE CIRCUITOS, NETLIST E REFERÊNCIAS

```

1 *****
2 **                STANDARD CELLS                *
3 *****
4
5 CCT CONT512(Q[0:8],QN[0:8],CLS,RN)
6 DF115 XXXXXX(A,B,C,D,E);
7 INPUT Q[0:8] QN[0:8] CLS RN.
8
9 CCT AA045(Q,A,B,C,D)
10 AND (8,8,8700,8700,150) G1(Q,A,B,C,D);
11 CAPACITOR (10) CA(A) CD(D);
12 UNID Q;
13 INPUT A B C D .
14
15 CCT AO065(Q,A,B,C,D,E,F)
16 AND(0,0,0,0,120) G1(ABC,A,B,C);
17 AND(0,0,0,0,110) G2(DE,D,E);
18 OR(6,6,7420,7420,100) G3(Q,ABC,DE,F);
19 CAPACITOR(10) CC(C);
20 UNID ABC DE Q;
21 INPUT A B CD E F.
22
23 CCT DF115(Q,QN,D,C,RN)
24 BUFIF0(0,0) G1(A,D,C);
25 BUFIF0(7,7) G2(F,QNI,C);
26 BUFIF1(0,0) G3(A,E,C);
27 BUFIF1(15,15) G4(F,B,C);
28 NAND(0,0) G5(B,RN,A);
29 NAND(25,25) G6(E,B);
30 NAND(6,6) G7(QI,F);
31 NAND(8,8) G8(QNI,RN,QI);
32 NAND(3,3,8960,8960,0) G9(QN,QI);
33 NAND(3,3,7940,7940,0) G10(Q,QNI);
34 CAPACITOR(110) CC(C);
35 CAPACITOR(220) CD(D) CRN(RN);
36 UNID B E QI QNI Q QN;
37 TRI A F;
38 INPUT D C RN.
39
40 CCT IN015(Q,A)
41 NAND (2,2,2450,8450,180) G1(Q,A);
42 UNID Q;
43 INPUT A.
44
45 CCT MU215(Q,A,B,SEL)
46 AND(2,2,0,0,110) G1(AI,A);

```

```

47 AND(2,2,0,0,100) G2(BI,B);
48 BUFIF0(0,0) G3(OI,AI,SEL);
49 BUFIF1(0,0) G4(OI,BI,SEL);
50 AND(9,9,8700,8700,0) G5(Q,QI);

51 CAPACITOR(180) CSEL(SEL);
52 UNID AI BI Q;
53 TRI OI;
54 INPUT A B SEL.
55
56 CCT NA025(Q,A,B)
57 NAND(5,5,7170,7170,220) G1(Q,A,B);
58 UNID Q;
59 INPUT A B.
60
61 CCT TT025(Q,A,EN)
62 INPUT A,Q,EN.
63
64 CCT VETOR1(A,B,C)
65 INPUT A B C.
66
67 CCT VETOR2(X[1:10])
68 UNID X[1:10]
69 .

```

```

1 *****
2 **                                **
3                                CIRCUITOS
4 *****
5
6 CCT MAIN (ZZZ)
7
8 VETOR JKL(T[0:9],K,R[9:0],L,U[40:50]);
9 INPUT ZZZ
10 .
11 CCT VETOR (A[0:9],B,C[9:0],D,E[10:20])
12 VETOR2 A1(A[1:9],B)
13         A2(A[9:1],B)
14         A3(A[6:3],A[0:2],A[9:8],B)
15         A4(E[14:17],C[2:6],B);
16
17 DF115 A5 (F[2],G[2],h,i,j)
18        A6[1:0] (F[1:0],G[0:1],h,i,j)
19        A9[2:6] (nF[6:2],nG[6:2],h,i,j)
20        A11[0:4] ((F,h,g[1],nG[4:0],h,i,j));
21 INPUT A[0:9] B C[9:0] D E[10:20]
22 .

```

NETLIST EM C:MAIN.NET

```

1 VETOR2
X[1] T[1]
X[2] T[2]
X[3] T[3]
X[4] T[4]
X[5] T[5]
X[6] T[6]
X[7] T[7]
X[8] T[8]
X[9] T[9]
X[10] T[K]
#
2 VETOR2
X[1] T[9]
X[2] T[8]
X[3] T[7]

```

X[4]
X[5]
X[6]
X[7]
X[8]
X[9]
X[10]

#3
X[1]
X[2]
X[3]
X[4]
X[5]
X[6]
X[7]
X[8]
X[9]
X[10]

#4
X[1]
X[2]
X[3]
X[4]
X[5]
X[6]
X[7]
X[8]
X[9]
X[10]

#5
OON
D
C
RN

#6
OON
D
C
RN

#7
OON
D
C
RN

#8
OON
D
C
RN

#9
OON
D
C
RN

#10
OON
D
C
RN

#11
OON

T[6]
T[5]
T[4]
T[3]
T[2]
T[1]
K

VETOR2

T[6]
T[5]
T[4]
T[3]
T[0]
T[1]
T[2]
T[9]
T[8]
K

VETOR2

U[44]
U[45]
U[46]
U[47]
R[2]
R[3]
R[4]
R[5]
R[6]
K

DF115

1
6
7
8
9

DF115

2
4
7
8
9

DF115

3
5
7
8
9

DF115

10
15
7
8
9

DF115

11
16
7
8
9

DF115

12
17
7
8
9

DF115

13
18

D	7
C	8
RN	9
#	
12	DF115
O	14
ON	19
D	7
C	8
RN	9
#	
13	DF115
O	1
ON	17
D	7
C	8
RN	9
#	
14	DF115
O	2
ON	18
D	7
C	8
RN	9
#	
15	DF115
O	3
ON	19
D	7
C	8
RN	9
#	
16	DF115
O	7
ON	20
D	7
C	8
RN	9
#	
17	DF115
O	5
ON	21
D	7
C	8
RN	9
#	
#	

REFERÊNCIAS CRUZADAS DE INSTÂNCIAS EM C:MAIN.RRC

INSTÂNCIAS:

1	MAIN.JKL.A1
2	MAIN.JKL.A2
3	MAIN.JKL.A3
4	MAIN.JKL.A4
5	MAIN.JKL.A5
6	MAIN.JKL.A6 [1]
7	MAIN.JKL.A6 [0]
8	MAIN.JKL.A9 [2]
9	MAIN.JKL.A9 [3]
10	MAIN.JKL.A9 [4]
11	MAIN.JKL.A9 [5]
12	MAIN.JKL.A9 [6]
13	MAIN.JKL.A11 [0]
14	MAIN.JKL.A11 [1]
15	MAIN.JKL.A11 [2]
16	MAIN.JKL.A11 [3]
17	MAIN.JKL.A11 [4]

REFERÊNCIAS CRUZADAS DE SINAIS EM C:MAIN.RFS

ZZZ	MAIN.ZZZ
T[0]	MAIN.T[0]
T[1]	MAIN.T[1]
T[2]	MAIN.T[2]
T[3]	MAIN.T[3]
T[4]	MAIN.T[4]
T[5]	MAIN.T[5]
T[6]	MAIN.T[6]
T[7]	MAIN.T[7]
T[8]	MAIN.T[8]
T[9]	MAIN.T[9]
K	MAIN.K
R[9]	MAIN.R[9]
R[8]	MAIN.R[8]
R[7]	MAIN.R[7]
R[6]	MAIN.R[6]
R[5]	MAIN.R[5]
R[4]	MAIN.R[4]
R[3]	MAIN.R[3]
R[2]	MAIN.R[2]
R[1]	MAIN.R[1]
R[0]	MAIN.R[0]
L	MAIN.L
U[40]	MAIN.U[40]
U[41]	MAIN.U[41]
U[42]	MAIN.U[42]
U[43]	MAIN.U[43]
U[44]	MAIN.U[44]
U[45]	MAIN.U[45]
U[46]	MAIN.U[46]
U[47]	MAIN.U[47]
U[48]	MAIN.U[48]
U[49]	MAIN.U[49]
U[50]	MAIN.U[50]
1	MAIN.JKL.F[2]
2	MAIN.JKL.F[1]
3	MAIN.JKL.F[0]
4	MAIN.JKL.G[0]
5	MAIN.JKL.G[1]
6	MAIN.JKL.G[2]
7	MAIN.JKL.H
8	MAIN.JKL.I
9	MAIN.JKL.J
10	MAIN.JKL.NF[6]
11	MAIN.JKL.NF[5]
12	MAIN.JKL.NF[4]
13	MAIN.JKL.NF[3]
14	MAIN.JKL.NF[2]
15	MAIN.JKL.NG[6]
16	MAIN.JKL.NG[5]
17	MAIN.JKL.NG[4]
18	MAIN.JKL.NG[3]
19	MAIN.JKL.NG[2]
20	MAIN.JKL.NG[1]
21	MAIN.JKL.NG[0]

10 REFERÊNCIAS BIBLIOGRÁFICAS

1. GenRAD, HILO-2 INTERFACE REFERENCE MANUAL, Document # 2521-0103/1, December 1984.
2. BIGONHA, M. A. S., BIGONHA, R. S., SIC: SISTEMA DE IMPLEMENTAÇÃO DE COMPILADORES, Série de Monografia do DCC, número T02/86, DCC - ICEX - UFMG.
3. BIGONHA, M. A. S., BIGONHA, R. S., SISTEMA DE SUPORTE A IMPLEMEN TAÇÃO DE COMPILADORES, Anais do XIII Seminário Integrado de "Software" e "Hardware", UFPE - RECIFE, 1986.
4. BORLAND INTERNATIONAL, Turbo PASCAL version 3.0 Reference Manual, 1985.
5. JENSEN, K. & WIRTH, N., PASCAL User Manual and Report, Springer-Verlag 1974.

APÊNDICE A : SINTAXE DO HILO-2

```

1  S'                = hilo_2 "eof"
2  hilo_2            = L_circuit_des
3  L_circuit_des     = L_circuit_des circuit_des
4                    | empty
5  empty             =
6  circuit_des       = circuit_header O_semicolon
                    elem_decls circuit_tail "<cr>"
7                    | circuit_header O_semicolon
                    lookahead "<cr>"
8                    | other_feature
9                    | other_feature "<cr>"
10 circuit_tail      = L_other_feature R_wire_decls
                    O_semicolon O_other_feature
11                  | L_other_feature
12 lookahead         = "FUNCTION"
13                  | empty
14                  | "WHEN"
15                  | "REGISTER"
16                  | "RAM"
17                  | "EVENT"
18                  | "IF"
19                  | "IFNOT"
20                  | "NEXT"
21                  | "TEXT"
22                  | "identifier"
23                  | "AND"
24                  | "OR"
25                  | "NAND"
26                  | "NOR"
27                  | "BUF"
28                  | "NOT"
29                  | "BUFIFO"
30                  | "BUFIF1"
31                  | "NOTIFO"
32                  | "NOTIF1"
33                  | "TRANIFO"
34                  | "TRANIF1"
35                  | "CLOCK0"
36                  | "CLOCK1"
37                  | "CAPACITOR"
38                  | "BALR"
39                  | "UNID"
40                  | "WAND"
41                  | "WOR"
42                  | "TRI"
43                  | "TRI1"
44                  | "TRIO"
45                  | "INPUT"
46                  | "INPUT1"
47                  | "INPUT0"
48                  | "SUPPLY0"
49                  | "SUPPLY1"
50 elem_decls        = L_elem_decl
51 L_elem_decl       = L_elem_decl elem_decl
52                  | elem_decl
53 circuit_header    = circuit_spec "(" ext_connection ")"
54                  | circuit_spec

```

```

55  IC_PCB_CCT      = "IC"
56                    | "PCB"
57                    | "CCT"

58  circuit_name    = "identifier"

59  circuit_spec     = IC_PCB_CCT O_delay par lst
                    O_delay_scaling circuit_name

60  ext_connection   = ext_connection "," wire
61                    | wire

62  wire             = wire_name
63                    | wire_vector

64  elem_decl        = element_type O_delay_values
                    L_elem_instance ";"

65  wire_name        = "identifier"

66  wire_vector      = "identifier" "[" indice "]"

67  L_elem_instance  = L_elem_instance elem_instance
68                    | empty

69  elem_instance    = instance_call O_connecton_lst

70  instance_name    = "identifier"
71                    | "identifier" "[" indice "]"

72  instance_call    = instance_name

73  O_connecton_lst  = "(" connection_list ")"
74                    | empty

75  element_type     = primitive_elem
76                    | "identifier"

77  primitive_elem   = OK_prim_elem
78                    | NOK_prim_elem

79  OK_prim_elem     = "CAPACITOR"

80  NOK_prim_elem    = "AND"
81                    | "OR"
82                    | "NAND"
83                    | "NOR"
84                    | "BUF"
85                    | "NOT"
86                    | "BUFIF0"
87                    | "BUFIF1"
88                    | "NOTIF0"
89                    | "NOTIF1"
90                    | "TRANIF0"
91                    | "TRANIF1"
92                    | "CLOCK0"
93                    | "CLOCK1"
94                    | "BALR"

95  connection_list = connection_list "," O_connection
96                    | O_connection

97  O_connection     = connection
98                    | empty

99  connection       = "identifier"
100                   | "integer_cte"
101                   | "Z"
102                   | vector_instance
103                   | "{" R_elem "}"

104  R_elem           = R_elem "," elem
105                   | elem

106  elem             = "identifier"
107                   | vector_instance

```

```

108 vector_instance = indices "]"
109 indices          = "identifier" "[" indice
110                 | indices "," indice
111 indice           = "integer_cte" O_integer_cte
112 O_integer_cte    = ":" "integer_cte"
113                 | empty
114 R_wire_decls     = R_wire_decls ";" wire_decls
115                 | wire_decls
116 wire_decls       = wire_definition O_delay_values
117                 | L_wire_instance
118 L_wire_instance  = L_wire_instance wire_instance
119                 | O_terminal_list O_comma
120                 | empty
121 wire_instance    = "identifier"
122                 | "identifier" "[" indice "]"
123 wire_definition  = "UNID"
124                 | "WAND"
125                 | "WOR"
126                 | "TRI"
127                 | "TRI1"
128                 | "TRIO"
129                 | "INPUT"
130                 | "INPUT1"
131                 | "INPUT0"
132                 | "SUPPLY0"
133                 | "SUPPLY1"
134 O_terminal_list  = "(" terminal_list ")"
135                 | empty
136 terminal_list    = terminal_list "," O_terminal
137                 | terminal
138                 | empty
139 terminal         = instance_id "." terminal_number
140                 | "identifier"
141 instance_id      = "identifier"
142                 | "identifier" "[" indice "]"
143 terminal_number  = "integer_cte"
144 O_delay_values   = "(" delay_values ")"
145                 | empty
146 O_other_feature  = R_other_feature
147                 | empty
148 L_other_feature  = L_other_feature other_feature ";"
149                 | empty
150 R_other_feature  = R_other_feature ";" other_feature
151                 | other_feature
152 O_delay_par_list = "(" delay_par_list ")"
153                 | empty
154 O_delay_scaling  = NS_PS
155                 | NS_PS "integer_cte"
156                 | empty
157 NS_PS           = "NS"
158                 | "PS"
159 delay_par_list   = delay_par_list "," delay_par
160                 | delay_par

```



```

160 delay_par      = "identifier"
161 delay_values   = delay_values "," range
162               | range
163 range          = minimum
164               | minimum ":" maximum
165               | minimum ":" typical ":" maximum
166 minimum        = "integer_cte"
167               | "integer_cte" MK
168 typical        = "integer_cte"
169               | "integer_cte" MK
170 maximum        = "integer_cte"
171               | "integer_cte" MK
172 MK             = "identifier"
173 other_feature  = "FUNCTION" anything
174               | "WHEN" anything
175               | "REGISTER" anything
176               | "RAM" anything
177               | "EVENT" anything
178               | "IF" anything
179               | "IFNOT" anything
180               | "NEXT" anything
181               | "TEXT" anything
182 O_semicolon    = ";"
183               |
184 O_comma         = ","
185               |
186 anything       = empty

```

TABELA DE REFERENCIAS CRUZADAS DE NAO-TERMINAIS

anything									
Definicao	: 186								
Ocorrencias	: 173	174	175	176	177	178	179	180	181
circuit des									
Definicoes	: 6	7	8	9					
Ocorrencias	: 3								
circuit header									
Definicoes	: 53	54							
Ocorrencias	: 6	7							
circuit name									
Definicao	: 58								
Ocorrencias	: 59								
circuit spec									
Definicao	: 59								
Ocorrencias	: 53	54							
circuit tail									
Definicoes	: 10	11							
Ocorrencias	: 6								
connection									
Definicoes	: 99	100	101	102	103				
Ocorrencias	: 97								
connection list									
Definicoes	: 95	96							
Ocorrencias	: 73	95							
delay par									
Definicao	: 160								
Ocorrencias	: 158	159							
delay par list									
Definicoes	: 158	159							
Ocorrencias	: 151	158							
delay values									
Definicoes	: 161	162							
Ocorrencias	: 143	161							
elem									
Definicoes	: 106	107							
Ocorrencias	: 104	105							
elem decl									
Definicao	: 64								
Ocorrencias	: 51	52							
elem decls									
Definicao	: 50								
Ocorrencias	: 6								
elem instance									
Definicao	: 69								
Ocorrencias	: 67								
element type									
Definicoes	: 75	76							
Ocorrencias	: 64								
empty									
Definicao	: 5								
Ocorrencias	: 4	13	68	74	98	113	118	133	137
		144	146	148	152	155	186		
ext connection									
Definicoes	: 60	61							
Ocorrencias	: 53	60							
hilo 2									
Definicao	: 2								

Ocorrencia	:	1						
IC PCB CCT								
-Definicoes	:	55	56	57				
Ocorrencia	:	59						
indice								
Definicao	:	111						
Ocorrencias	:	66	71	109	110	120	141	
indices								
Definicoes	:	109	110					
Ocorrencias	:	108	110					
instance call								
Definicao	:	72						
Ocorrencia	:	69						
instance id								
Definicoes	:	140	141					
Ocorrencia	:	138						
instance name								
Definicoes	:	70	71					
Ocorrencia	:	72						
L_circuit des								
-Definicoes	:	3	4					
Ocorrencias	:	2	3					
L_elem decl								
-Definicoes	:	51	52					
Ocorrencias	:	50	51					
L_elem instance								
-Definicoes	:	67	68					
Ocorrencias	:	64	67					
L_other feature								
-Definicoes	:	147	148					
Ocorrencias	:	10	11	147				
L_wire instance								
-Definicoes	:	117	118					
Ocorrencias	:	116	117					
lookahead								
Definicoes	:	12	13	14	15	16	17	18
		20	21	22	23	24	25	26
		28	29	30	31	32	33	34
		36	37	38	39	40	41	42
		44	45	46	47	48	49	43
Ocorrencia	:	7						
maximum								
Definicoes	:	170	171					
Ocorrencias	:	164	165					
minimum								
Definicoes	:	166	167					
Ocorrencias	:	163	164	165				
MK								
Definicao	:	172						
Ocorrencias	:	167	169	171				
NOK prim elem								
Definicoes	:	80	81	82	83	84	85	86
		88	89	90	91	92	93	94
Ocorrencia	:	78						87
NS PS								
-Definicoes	:	156	157					
Ocorrencias	:	153	154					
O_comma								
-Definicoes	:	184	185					

[illegible]

```

terminal number
  Definicao : 142
  Ocorrencia : 138

typical
  Definicoes : 168 169
  Ocorrencia : 165

vector instance
  Definicao : 108
  Ocorrencias : 102 107

wire
  Definicoes : 62 63
  Ocorrencias : 60 61

wire decls
  Definicao : 116
  Ocorrencias : 114 115

wire definition
  Definicoes : 121 122 123 124 125 126 127 128 129
               130 131
  Ocorrencia : 116

wire instance
  Definicoes : 119 120
  Ocorrencia : 117

wire name
  Definicao : 65
  Ocorrencia : 62

wire vector
  Definicao : 66
  Ocorrencia : 63

"("
  Ocorrencias : 53 73 132 143 151

")"
  Ocorrencias : 53 73 132 143 151

"*"
  Ocorrencia : Simbolo nao usado na gramatica.

","
  Ocorrencias : 60 95 104 110 134 158 161 184

"."
  Ocorrencia : 138

".<cr>"
  Ocorrencias : 6 7 9

":"
  Ocorrencias : 112 164 165 165

";"
  Ocorrencias : 64 114 147 149 182

"["
  Ocorrencias : 66 71 109 120 141

"]"
  Ocorrencias : 66 71 108 120 141

" "
  - Ocorrencia : Simbolo nao usado na gramatica.

"AND"
  Ocorrencias : 23 80

"BALR"
  Ocorrencias : 38 94

```

[illegible]

"NS"	Ocorrencia	:	156	
"OR"	Ocorrencias	:	24	81
"PCB"	Ocorrencia	:	56	
"PS"	Ocorrencia	:	157	
"RAM"	Ocorrencias	:	16	176
"REGISTER"	Ocorrencias	:	15	175
"SUPPLY0"	Ocorrencias	:	48	130
"SUPPLY1"	Ocorrencias	:	49	131
"TEXT"	Ocorrencias	:	21	181
"TRANIF0"	Ocorrencias	:	33	90
"TRANIF1"	Ocorrencias	:	34	91
"TRI"	Ocorrencias	:	42	124
"TRIO"	Ocorrencias	:	44	126
"TRI1"	Ocorrencias	:	43	125
"UNID"	Ocorrencias	:	39	121
"WAND"	Ocorrencias	:	40	122
"WHEN"	Ocorrencias	:	14	174
"WOR"	Ocorrencias	:	41	123
"Z"	Ocorrencia	:	101	
"{"	Ocorrencia	:	103	
"}"	Ocorrencia	:	103	

Esta gramatica é reduzida.