

NAME

ACTIVCELL – aktiviert eine geladene Zelldefinition oder einen Generator

SYNOPSIS

ACTIVCELL name [-DIALOG] [gargs]

BESCHREIBUNG

Wählt eine geladene Zelldefinition bzw. einen Instanz-Generator aus, die mit dem nächsten ADDCELL-Befehl addiert werden kann; die Zelle bleibt bis zum nächsten ACTIVCELL-, EDIT-INPLACE-, DEFQUIT- oder DEFCLOSE-Kommando aktiviert. Das Kommando ist nicht erlaubt, wenn keine Zelldefinition editiert ist.

Es kann entweder ein Zellname oder der Generatorname eines Instanz- oder Zellgenerators angegeben werden. Wenn eine Zelle *name* nicht existiert oder kein Zellname eines I2L Gatters im Fall I2L Designtyp ist, sucht PARIS einen Instanz- oder Zell-Generator *name*. Wenn ein passender Generator existiert, wird der Generator mit den angegebenen Generatorparametern aufgerufen. Bei angegebener Option `-DIALOG` wird eine Dialogbox zur Eingabe der Generatorparameter angezeigt. In diesem Fall werden die evtl. zusätzlich angegebenen Generatorparameter *gargs* als Voreinstellungen in die Dialogbox übernommen. Nach der Parametereingabe wird dann die Instanz generiert. Danach können mit dem Kommandod ADDCELL neue Instanzen der ausgewählten oder generierten Zelle in die gerade offene Zelle eingefügt werden.

ARGUMENTE

<i>name</i> (cn)	Name der auszuwählenden Zelldefinition oder des Generators.
<i>gargs</i> (gargs)	Generatorparameter des gewählten Generators. Die Generatorparameter werden nur berücksichtigt, eratorparameter angezeigt. Diese Option wirkt nur, wenn <i>name</i> der Name eines Generators ist, sonst werden sie ignoriert.

OPTIONEN

<code>-DIALOG</code>	Wenn <code>-DIALOG</code> angegeben ist, wird eine Dialogbox zur Eingabe der Generatorparameter angezeigt. Diese Option wirkt nur, wenn <i>name</i> der Name eines Generators ist, sonst wird sie ignoriert.
----------------------	--

WEITERES

ADDCELL(PARIS),EXCHANGE(PARIS),REGENERATE(PARIS)

NAME

ADDCELL – setzt Addierstatus auf add cell

SYNOPSIS

ADDCELL [-PLNAME] [-SYMOP] [-SEL]

BESCHREIBUNG

startet das Addieren der aktivierten Zelle; die eigentliche Platzierung erfolgt mit den folgenden DS (Testplatzierung) und DF (endgültige Platzierung) Befehlen.

OPTIONEN

-PLNAME c9	Platzierungsname DEFAULT: fortlaufende Nummer
-SYMOP s	Orientierung der zu platzierenden Zelle DEFAULT: 0
-SEL	der Zellaufruf ist nach der Platzierung selektiert DEFAULT: nicht selektiert

WEITERES

DS(Paris), DP(Paris), DF(Paris), CANCELCOMP(Paris), ACTIVCELL(Paris)

NAME

ADDGEO – setzt Addierstatus auf add geo

SYNOPSIS

ADDGEO -RECT | -NRECT | <-PATH [-WIDTH] [-PATHTYPE]> | -POLY | <-CIRCLE
[-RADIUS]> [-LAYER] [-FUNC] [-SEL]

BESCHREIBUNG

Startet Addieren von Geoelementen; die Eingabe der Koordinaten erfolgt mit den folgenden DS Kommandos, der Abschluss mit einem DF-Kommando. Der aktuelle Stand der Addierparameter wird verwendet, falls nicht explizit beim Aufruf andere Werte spezifiziert sind. Sofern nicht anders angegeben, rasten die eingegebenen Punkte ins eingestellte Grid. Bei Designs vom Typ GA und TA rasten Pfade immer ins PATHGRID.

OPTIONEN

- RECT addiert Geotyp RECHTECK; Eingabe des ersten Diagonales mit DS, des zweiten mit DF.
- NRECT addiert Geotyp N_RECHTECK; dieser Komponententyp beschreibt eine Gruppe von Rechtecken, die alle die gleiche Knotennummer erhalten. Die Eingabe aller Punkte erfolgt mit DS, die Komponente wird abgeschlossen mit DF.
- PATH addiert Geotyp PFAD; die Breite des Pfades kann mit der Option WIDTH, der Pfadtyp (mit oder ohne Überhang) kann mit der Option PATHTYPE spezifiziert werden, andernfalls werden die DEFAULT Werte aus dem Parameterfile entnommen. Die Eingabe aller Punkte erfolgt mit dem DS-Kommando, der Abschluss mit einem DF.
- POLY addiert Geotyp POLYGON; die Eingabe aller Punkte erfolgt mit dem DS-Kommando, der Abschluss mit einem DF. Die Kante vom letzten zum ersten Punkt wird automatisch erzeugt, d.h. der erste und letzte Punkt ist nicht identisch.
- CIRCLE addiert einen Kreis; dieser wird als Komponententyp POLYGON abgelegt. Der Radius kann entweder mit der Option RADIUS spezifiziert werden oder durch Eingabe von zwei Punkten, nämlich dem Mittelpunkt und einem Punkt auf dem Kreisbogen. Die Auflösung bei der Berechnung des Polygons wird dem Parameter PCIRC entnommen. Ob der Kreis auf das Minimalgrid rasten soll oder nicht wird dem Parameter SNAPCIRCLE entnommen. Beide Parameter können mit SETPAR definiert werden.
- LAYER l zu belegende Ebene;
 DEFAULT: 'actlayer'
- FUNC gf [sw] [pc] Funktion des zu addierenden Geoelementes; im Fall von PIN folgen der Pinname und der Pintyp;
 DEFAULT: 'actfunc'
- WIDTH in Pfadweite
 DEFAULT: fuer die entsprechende Ebene voreingestellte Pfadweite
- PATHTYPE pt spezifiziert den Pfadtyp (OVERHANG, NOOVERHANG);
 DEFAULT: im Parameterfile voreingestellter Pfadtyp
- RADIUS in Radius
 DEFAULT: wird durch Eingabe von zwei Punkten spezifiziert

-SEL die addierte Komponente ist anschliessend selektiert
DEFAULT: unselektiert

VORSICHT

Die Varianten ADDGEO -CIRCLE | -RECT erfordern beim Abschluss des Addiervorgangs ein DF xy.

WEITERES

DS(Paris), DP(Paris), DF(Paris), ARC(Paris), SETPAR(Paris), CANCELCOMP(Paris),
APPEND(Paris), GRID(Paris)

NAME

ADDTEXT - setzt Addierstatus auf add text

SYNOPSIS

ADDTEXT string [-SEL]

BESCHREIBUNG

startet Addieren von freien Textkomponenten. Die neue Textkomponente erhält die Einstellungen des Layers und der Textgröße des aktuellen Eingabefensters.

ARGUMENTE

string (st) zu plzierender Textstring

OPTIONEN

-SEL Text ist nach dem Addieren selektiert
DEFAULT: ist nicht selektiert

WEITERES

DS(Paris), DP(Paris), DF(Paris), SETPAR(Paris), CANCELCOMP(Paris)

NAME

ALIGN – Ausrichten von selektierten Zellaufrufen

SYNOPSIS

ALIGN -TOP | -BOTTOM | -RIGHT | -LEFT | -HORIZONTAL | -VERTICAL [point]

BESCHREIBUNG

Mit dem Kommando **ALIGN** lassen sich selektierte Zellaufrufe auf eine Linie ausrichten. Als Bezugsmaß zum Ausrichten dient das umschliessende Rechteck der Komponenten. Die Zellaufrufe werden so verschoben, daß die durch die angegebene Option spezifizierte Kante eine Linie bildet. Die anderen Aufrufkoordinaten bleiben unverändert. Das Argument steht am Ende des Kommandos.

ARGUMENTE

point (p) Referenzpunkt für das Ausrichten der Zellen
DEFAULT: je nach Option wird an der entsprechenden Kante ausgerichtet

OPTIONEN

-TOP Der obere Rand der Komponenten wird ausgerichtet
-BOTTOM Der untere Rand der Komponenten wird ausgerichtet
-RIGHT Der rechte Rand der Komponenten wird ausgerichtet
-LEFT Der linke Rand der Komponenten wird ausgerichtet
-HORIZONTAL Die Mitte wird horizontal ausgerichtet. Die Mittelpunkte der selektierten Instanzen werden auf die gleich y Koordinate gebracht.
-VERTICAL Die Mitte wird vertikal ausgerichtet. Die Mittelpunkte der selektierten Instanzen werden auf die gleich x Koordinate gebracht.

FEHLER

ECOMNOTALLOWED
Es ist entweder keine Zell editiert oder der Designtyp ist TA oder GA.

WEITERES

SELECT(paris)

NAME

APPEND – Erweiterung eines vorhandenen Geoelements

SYNOPSIS

APPEND

BESCHREIBUNG

Erweitert ein GeoElement (Pfad oder Polygon) um zusätzliche Koordinaten. Ein GeoElement kann wie folgt erweitert werden:

- ein Pfad am Anfang oder Ende durch Selektion des Anfangs- bzw Endpunktes
- ein Pfad zwischen zwei aufeinanderfolgenden Koordinaten durch Selektion der entsprechenden Kante
- ein Polygon zwischen zwei aufeinanderfolgenden Koordinaten durch Selektion der entsprechenden Kante

Die Eingabe der Koordinaten erfolgt mit dem DS-Kommando, beginnend bei einem selektierten Eckpunkt. Der Abschluss eines APPEND erfolgt mit dem DF-Kommando. Die Komponenten rasten ins eingestellte Grid. Bei Designs vom Typ GA und TA rasten Pfade immer ins PATH-GRID.

Es ist sichergestellt, daß Die Orthogonalität des Geoelements nach dem Erweitern nicht schlechter wird als die des Originalelements. Wenn die beiden letzten eingefügten Kanten ein Element mit schlechterer Orthogonalität ergeben, werden diese korrigiert, so daß die vorhergehende Orthogonalität wieder hergestellt wird.

WEITERES

DS(Paris), DP(Paris), DF(Paris), ARC(Paris), CANCELCOMP(Paris)

NAME

ARC – Einfügen eines Bogens in offenes Geoelement

SYNOPSIS

ARC [-ANGLE] | [-C2PCCW] | [-C2PCW]

BESCHREIBUNG

Mit dem Kommando ARC wird ein Bogen in ein bereits offenes Geoelement eingefuegt. Vor dem Aufruf von ARC müssen mindestens 2 Punkte mittels DS eingegeben worden sein. Je nach angegebener Option wird die Verbindungslinie dieser beiden vorherigen Punkte als Grundlinie des Bogens oder als Verbindungslinie Bogenanfang - Bogenmittelpunkt interpretiert. Die nachfolgende Punkteingabe mit DS legt das Bogenstück fest. Der Bogen wird in einen Polygonzug umgewandelt, wobei die Auflösung aus dem Wert des Parameters PCIRCLE, der die Anzahl der Punkte eines Vollkreises angibt, bestimmt wird. Die Anzahl der Punkte des Bogens werden anteilig seines Winkels gegenüber dem Vollkreis berechnet.

Bei den interaktiven Varianten wird jeweils am Bildschirm ein entsprechender Prompt eingestellt, der die Lage des Bogens abhängig von der Cursor Position anzeigt.

OPTIONEN

- | | |
|----------|--|
| -ANGLE i | Erzeugt einen Bogen mit 'i' Grad, wobei die beiden zuletzt eingegebenen Punkte den Anfangs- und Mittelpunkt des Bogens bestimmen.
DEFAULT: Bogen, bei dem die Grundlinie durch die beiden zuletzt eingegeben Punkte festgelegt wird und mit der auf ARC folgenden Punkteingabe die Bogengrösse bestimmt wird. |
| -C2PCCW | Erzeugt einen Bogen im Gegenuhrzeigersinn, wobei die beiden zuletzt eingegeben Punkte den Anfangs- und Mittelpunkt des Bogens festlegen. Die auf ARC folgende Punkteingabe bestimmt die Bogengrösse.
DEFAULT: siehe oben |
| -C2PCW | Erzeugt einen Bogen im Uhrzeigersinn, wobei die beiden zuletzt eingegeben Punkte den Anfangs- und Mittelpunkt des Bogens festlegen. Die auf ARC folgende Punkteingabe bestimmt die Bogengrösse.
DEFAULT: siehe oben |

WEITERES

DS(Paris), DP(Paris), DF(Paris), ARC(Paris), SETPAR(Paris), CANCELCOMP(Paris), APPEND(Paris), ADDGEO(Paris)

NAME

ATTACH - Zuordnen von Eigenschaften an eine Zelle

SYNOPSIS

ATTACH [-CELL | -SEL] -SYMOP | -MACFUNC | -PROPERTY | - PLCHAR

BESCHREIBUNG

Ordnet einer Zelldefinition eine Macrofunktion, ein Property, die erlaubten Symmetrieoperationen bei der Platzierung zu oder das Platzierungsmerkmal zu.

OPTIONEN

- CELL cn [...] den spezifizierten Zelldefinitionen wird ein Attribut zugeordnet
DEFAULT: aktuelle Zelldefinition
- SEL den selektierten Zelldefinitionen wird ein Attribut zugeordnet
DEFAULT: aktuelle Zelldefinition
- MACFUNC mf den Zielkomponenten wird die Macrofunktion 'mf' angehängt.
- PROPERTY pn v den Zielkomponenten wird ein Property pn mit dem Wert v zugewiesen.
- SYMOP s die Zielkomponenten erhalten als erlaubte Symmetrieoperationen die spezifizierten Werte. S ist eine Oktalzahl, die als Bitmaske die erlaubten Symmetrieoperationen angibt. Die Bitposition in s steht dabei fuer die Kennzahl der Symmetrieoperationen. Position 1 bezeichnet die Symmetrieoperation 0, 2 die Symmetrieoperation 1, etc. Zum Beispiel s=1 erlaubt nur die Symmetrieoperation 0, s=577 erlaubt die Symmetrieoperationen 0-7.
- PLCHAR c die Zielkomponente erhält als Platzierungsmerkmal das Zeichen c. Das Platzierungsmerkmal ist für den Austausch von Zellen von Bedeutung. Es dürfen nur Zellen mit gleichem Platzierungsmerkmal oder Zellen mit undefiniertem Platzierungsmerkmal gegeneinander ausgetauscht werden. Das Platzierungsmerkmal muß ein Buschstabe oder eine Zahl sein. Es wird intern in Großbuchstaben umgewandelt. Als undefiniertes Platzierungsmerkmal zählt der Buschstabe X.

FEHLER

- EPARNOTLOADED**
Die Technology ist noch nicht geladen. Siehe LOADPAR.
- EILLPARATYP** Bei Angabe einer Symmetrieoperation oder des Platzierungsmerkmal ist eine falscher Wert angegeben worden.
- ECELLNOTINDB**
Die spezifizierte Zelle(n) existieren nicht.
- ECOMNOTALLOWED**
Bei Angabe des Schalters -SEL bzw. im default Modus ist keine Zelle bearbeitet.

WEITERES

PROPERTY(Paris),EXCHANGE(Paris)

NAME

BOOLOP – Boole'sche Operationen

SYNOPSIS

AND -DST [-SRC] [-NODELETE]
OR -DST [-SRC] [-NODELETE] [-BB]
NOT -DST [-SRC] [-NODELETE] [-WINDOW]
DIFF -DST -SRC [-NODELETE]

BESCHREIBUNG

Führt die boole'schen Operationen AND,OR,NOT sowie DIFF (Differenz) auf selektierte Komponenten aus. Es werden sowohl Geometrie als auch Instanzen betrachtet. Falls Instanzen selektiert sind, wird die Instanz rekursiv durchlaufen. Dabei werden alle Geometrieelemente, die durch die Angabe der Source layer (-SRC layerset) bestimmt sind, dupliziert und mit der ggf, akkumulierten Instanztransformation transformiert. Die so gewonnenen Originalelemente werden dann mit der angegebenen Operation verarbeitet. Bei der Operation OR werden alle Elemente verschmolzen. Bei AND entsteht der Durchschnitt aller Elemente, also nur diejenige Fläche, die von mehr als einem Element überdeckt ist. Die Inversion erzeugt diejenige Fläche bis zum umschließenden Rechteck der gerade bearbeiteten Zelle, die von keiner der selektierten Komponenten überdeckt ist. Die DIFF Funktion schneidet die Elemente aus layerset2 aus den Elementen aus layerset1 aus.

Anstelle der Originalelemente können auch deren umschließende Rechtecke verwendet werden (-BB).

Nach der Ausführung der boole'schen Operation werden die Originalelemente gelöscht und die neuen Elemente erscheinen selektiert. Dies gilt nur für Geometrieelemente. Instanzen werden grundsätzlich nicht gelöscht, sondern bleiben selektiert. Dieses Verhalten kann mit dem Schalter -NODELETE geändert werden.

OPTIONEN

-DST layer Die neuen Elemente können in den angegebenen Ziellayer.
-SRC layerset Es werden nur die Elemente im angegebenen Layer Set verwendet. Wenn diese Angabe fehlt, werden alle selektierten Komponenten verwendet. ! Dies gilt insbesondere auch für Instanzen, deren Inhalt vollständig verwendet wird, wenn die Angabe eines Source Layersets fehlt.
-SRC layerset1 layerset2 Nur bei der DIFF Funktion. Schneidet die Elemente in layerset2 aus den Elementen in layerset1 aus.
-NODEL Die Originalkomponenten werden nicht gelöscht, sondern werden deselektiert. Dies gilt auch für Instanzen.
-BB verwendet anstelle der spezifizierten Komponenten deren umschließende Rechtecke.
-WINDOW box Gibt das Bezugsfenster an für die Invertierung. Fehlt diese Angabe, wird die Bounding Box der aktuellen Zelle verwendet.

WEITERES

FETCH(Paris)

NAME

CALCC - Berechnet die Kapazität von selektierten Komponente(n)

SYNOPSIS

CALCC [-SELNODE | -ALL]

BESCHREIBUNG

Berechnet den Kapazitätswert einer selektierten Komponente (PATH, POLY oder RECTANGLE). Der Kapazitätswert pro Quadratmeter ist im Parameterfile für die relevanten Ebenen spezifiziert. Liegt keine Wertangabe fuer diese Ebene vor, so wird nur die Fläche der selektierten Komponente ausgegeben. Wenn die selektierten Komponenten in einer Zelle vom Typ CAPACITOR enthalten sind, wird der Zelle ein Property mit Namen C und dem Wert der berechneten Kapazität in Farad angehängt. Der Kapazitätsbelag für die betrachteten Layer muß im Technologiefile angegeben sein.

DEFAULT: Berechnet die Summe der Kapazität aller selektierten Geometrie.

OPTIONEN

-SELNODE Berechnet die Summe aller an das selektierte Netz angeschlossenen Kapazitäten.

-ALL Berechnet die Summe der Kapazität aller selektierten Geometrie.

WEITERES

SELECT(PARIS), FIND(PARIS), UNSELECT(PARIS), PRSPAR(5)

NAME

CALCR - Berechnet den Widerstandswert einer selektierten Komponente

SYNOPSIS

CALCR

BESCHREIBUNG

Berechnet den Widerstandswert einer selektierten Komponente. Diese muss vom Typ PATH sein. Der berechnete Wert wird der zugehörigen Macrodefinition als Property R (nur im Fall von Designtyp BCD,I2L) zugewiesen.

WIDERSTANDSBERECHNUNG

Die Widerstandsberechnung erfolgt mit Hilfe einer C Funktion, in dem die Berechnungsformel codiert ist. Das C Programm muß als shared Library in der Bibliothek der geladenen Technology vorliegen. Es wird bei Bedarf, also beim ersten Aufruf von CALCR von PARIS geladen. PARIS sucht diese Shared Library unter dem Pfadnamen \$AEGDIR/bib/\$TECH/\$PLATFORM/calcr.sl. Dabei ist \$AEGDIR der Wert der gleichnamigen Environmentvariablen, \$TECH der Name der geladenen Technology aus dem Parameterfile, \$PLATFORM der Name der Hardwareplattform, auf der das Programm läuft. Derzeit sind als PLATFORM die Werte hp700, hppa1.1-hp-hpux10.20, i486-linux, sparc-sun-sunos4.1, sparc-sun-solaris2.5.1 und alpha möglich.

Die Schnittstelle der C Funktion ist in der Headerdatei <paris/calcr.h> definiert:

```
/* error codes for resistor calculation */
enum {
    RETOMNYCOMPSEL      = 138, /* too many components selected */
    REWRGCOMPSEL        = 141, /* wrong components selected */
    RETOFEWCOMPSEL      = 162, /* too few components selected */
    RENOCOMPFOUND       = 164  /* no component found */
};

typedef struct _Rdesc {
    double width, length; /* Length, width of resistor path in Micron */
    int layer;           /* layer of resistor path */
} Rdesc;

typedef int (*CalcRProc) ARG_DECL((
    int    nc, /* number of elements in rd */
    Rdesc* rd, /* descriptor array of path elements sorted by layer */
    int    np, /* number of points of resistor path */
    double* r /* returned resistor value in OHM */
)); /* returns zero on success, positive value on error */
```

Von PARIS aus wird für jede selektierte Komponente ein Feldelement vom Typ Rdesc erzeugt. Rdesc enthält die Breite und die Länge sowie den Layer des selektierten Elementes. Wenn das selektierte Element kein Pfad ist, sind die Werte von width und length 0. PARIS ruft mit dem so erzeugten Array eine Funktion vom Typ CalcRProc mit dem Namen CalcR aus der Shared Library auf. Die Funktion CalcR führt die Widerstandsberechnung durch und gibt den berechneten Wert im vierten Argument *r zurück. Wenn die Berechnung erfolgreich war, ist der Returnwert von CalcR 0. Bei einem Fehler in der Widerstandsberechnung gibt CalcR einen positiven Wert zurück. Möglich Fehlercodes sind als enum in calcr.h vorgeschlagen.

Hier ein Beispiel eines Widerstandsberechnungsmoduls:

```
#include <paris/calcr.h>

#define SKP 200
#define SBP41 200
#define SBP43 1000
#define SBalu1 5e-02
#define SBalu2 4e-02
#define SBI67 15
#define SKI 15

int CalcR ARGS4(
    int,    nc,
    Rdesc*, rd,
    int,    np,
    double*, r
)
{
float ak;
if (nc > 1) return RETOMNYCOMPSEL;
switch (rd[0].layer) {
case 41: /* P Widerstand, niederohmig */
    ak = 12.0/rd[0].width;
    *r = SBP41*((rd[0].length - 10.0)/rd[0].width - 0.4*(np-2))
        + SKP*ak;
    break;
case 43: /* P Widerstand, hochohmig */
    ak = 12.0/rd[0].width;
    *r = SBP43*((rd[0].length - 10.0)/rd[0].width - 0.4*(np-2))
        + SKP*ak;
    break;
case 67: /* N Widerstand */
    ak = 12.0/rd[0].width;
    *r = SBI67*((rd[0].length - 10.0)/rd[0].width - 0.4*(np-2))
        + SKI*ak;
    break;
case 113: case 114: case 115: /* Metal 1 */
    *r = SBalu1*((rd[0].length)/rd[0].width - 0.4*(np-2));
    break;
case 177: case 178: case 179: /* Metal 2 */
    *r = SBalu2*((rd[0].length)/rd[0].width - 0.4*(np-2));
    break;
default:
    return REWRGCOMPSEL;
}
return 0;
}
```

ERZEUGEN DER SHARED LIBRARY

Das Erzeugen der shared Library ist unterschiedlich je nach der verwendeten Hardware Plattform. Die Library selbst muß aber immer unter dem Pfadnamen \$AEGDIR/bib/\$TECH/\$PLATFORM/calcr.sl stehen. Die Quelle zur Widerstandsberechnung ist für alle Plattformen gleich. Sie steht im Verzeichnis \$AEGDIR/bib/\$TECH/. Für die verschiedenen Plattformen wird jeweils ein Unterverzeichnis unter \$AEGDIR/bib/\$TECH/ eröffnet. Dort stehen dann die platformsspezifischen Dateien.

Abhängig von der Plattform sind folgende Kommandos erforderlich:

Plattform hp700:

Es muß mit der Option +z kompiliert, um position independent Code zu erzeugen. Danach wird mit der Option -b zum Erzeugen einer shared Library gelinkt.

```
cd $AEGDIR/bib/$TECH/$PLATFORM
cc -I$AEGINCLUDE +z -c ../calcr.c
ld -b -o calcr.sl calcr.o -lm
```

Plattform hppa1.1-hp-hpux10.20:

Der GNU Compiler benötigt den Schalter -fpic zum Erzeugen von position independent code. Das Linken erfolgt wie bei der Plattform hp700:

```
cd $AEGDIR/bib/$TECH/$PLATFORM
gcc -I$AEGINCLUDE -fpic -c ../calcr.c
ld -b -o calcr.sl calcr.o -lm
```

Plattformen i486-linux, sparc-sun-sunos4.1, sparc-sun-sunos2.5.1:

Für diese Plattformen wird der position independent Code ebenfalls mit -fpic erzeugt. Die Library wird mit Hilfe der Optionen -shared gelinkt:

```
cd $AEGDIR/bib/$TECH/$PLATFORM
gcc -I$AEGINCLUDE -fpic -c ../calcr.c
gcc -shared -o calcr.sl calcr.o
```

NAME

CANCELCOMP – Löschen einer angefangenen Komponente

SYNOPSIS

CANCELCOMP

BESCHREIBUNG

löscht die mit DS eingegebenen Punkte einer zuvor mit ADDGEO, ADDCELL, ADDTEXT spezifizierten (angefangenen) Komponente. Der Add- Status wird zurückgesetzt (CurrentAdd=noadd).

WEITERES

DS(Paris), DP(Paris), DF(Paris), ARC(Paris), ADDGEO(Paris), CANCELCOMP(Paris), APPEND(Paris), ADDCELL(Paris), ADDTEXT(Paris)

NAME

CDVIS - Verwalten der Stromdichtenvisualisierung

SYNOPSIS

CDVIS -ON | -OFF | -DELETE | -LAYER | -SCALE

BESCHREIBUNG

Die Stromdichte in Leiterbahnen kann mit Hilfe von Visualisierungsobjekten farblich dargestellt werden. Da die Anzahl von Visualisierungsobjekten sehr groß sein kann, gibt es dafür spezielle speicereffiziente Objekte, die sich von gewöhnlichen Geometrieobjekten unterscheiden. Ein Visualisierungsobjekte ist gekennzeichnet durch seine Grometrie, den Wert der Stromdichte in der Geometrie und den Verdrahtungslayer, zu dem das Objekt gehört. Die Darstellung der Visualisierungobjekte wird mit dem Kommando `cdvis` beeinflusst.

OPTIONEN

- OFF Die Darstellung der Stromdichtenvisualisierung wird ausgeschaltet. Die Visualisierungsobjekte bleiben erhalten.
- ON Die Darstellung der Stromdichtenvisualisierung wird eingeschaltet.
- DELETE Die Visualisierungsobjekte werden gelöscht und die ihre Darstellung ausgeschaltet.
- LAYER *gl* Es werden nur diejenigen Visualisierungsobjekte dargestellt, die zu den Verdrahtungsebenen in dem angegebenen Layerset gehören.
- SCALE *fl* Zur Darstellung werden alle Stromdichten um den Faktor *fl* skaliert. Diese Option beeinflusst nur die Darstellung, nicht aber den Wert des Stromdichtennattributs an den Visualisierungsobjekten. -SCALE kann benutzt werden, um den Einfluß von Stromänderungen auf die Stromverteilung in einer Struktur zu untersuchen. -SCALE wirkt immer auf alle Visualisierungsobjekte in einer Zelle. Es wird keine neue Berechnung der Stromdichten durchgeführt, sondern nur die Farben für die Darstellung neu zugewiesen.

WEITERES

`cds(1)`

NAME

CHANGECOMP – ändert die Parameter von selektierten Geo- und Textelementen

SYNOPSIS

CHANGECOMP -LAYER | -PATHWIDTH | -PATHTYPE | -TEXTSIZE | -TEXTDIR |
-TEXTSTRING | -ORTHO

BESCHREIBUNG

Selektierte Komponenten können einem anderen Layer zugeordnet werden. Parameter von selektierten Komponenten, wie Pfadbreite, Textgröße, Textrichtung und Orthogonalität können nachträglich verändert werden.

OPTIONEN

- LAYER l neue Ebenennummer von selektierten Geo- und Textelementen.
- PATHWIDTH in neue Pfadbreite der selektierten Pfade; selektierte Komponenten anderen Typs werden nicht berücksichtigt.
- PATHTYPE ändert den Pfadtyp der selektierten Pfade auf OVERHANG bzw. NOOVERHANG.
- TEXTSIZE in ändert die Textgröße der selektierten Textelemente; selektierte Komponenten anderen Typs werden nicht berücksichtigt.
- TEXTDIR td ändert die Darstellungsart (horizontal oder vertikal) der selektierten Textelemente; selektierte Komponenten anderen Typs werden nicht berücksichtigt.
- TEXTSTRING s ändert den Textstring der selektierten Textelemente. Selektierte Komponenten anderen Typs werden nicht berücksichtigt.
- ORTHO o ändert die Orthogonalität von selektierten Geoelementen; es kann nur in Richtung einer schwächeren Orthogonalität also von YES auf HALF oder NO geändert werden, nicht rückwärts.

NAME

CHANGEDEF – öffnet Zelle zum Editieren

SYNOPSIS

CHANGEDEF cellname [-GROUP]

BESCHREIBUNG

öffnet die Zelle 'cellname' zum Editieren; es wird eine Kopie der Zelldefinition angelegt. Falls 'cellname' nicht existiert wird eine neue Zelldefinition mit diesem Namen angelegt; in diesem Fall muss die Macrogruppe spezifiziert werden. Der Editiervorgang wird mit DEFCLOSE abgeschlossen oder mit DEFQUIT abgebrochen.

Das Erzeugen einer neuen Zelle ist im Fall von Designtyp TA nicht erlaubt.

ARGUMENTE

cellname (cn) Name der zu editierenden Zelle

OPTION

-GROUP mg Name der Macrogruppe, falls eine neue Zelle erstellt wird ('defstart'). Eine Macrogruppe muß angegeben werden, wenn eine neue Zelle erzeugt wird.

WEITERES

DEFCLOSE(PARIS), EDITINPLACE(PARIS)

NAME

CHANGEGEOFUNC – ändert die Funktion eines Geoelementes

SYNOPSIS

CHANGEGEOFUNC -PINWIRE | -WIREPIN | -DIVWIRE | -WIREDIV | -OBSTDIV |
-DIVOBST

BESCHREIBUNG

ändert die Funktion von selektierten Geoelementen; es sollten nur Komponenten mit der entsprechenden Funktion selektiert sein

OPTIONEN

-PINWIRE ändert die Funktion von PIN auf WIRE

-WIREPIN sw p [pc]

ändert die Funktion von WIRE auf PIN mit Pinnamen 'sw' an der Position 'p' und vom Typ 'pc' (DEFAULT: 'Undefined'); es darf nur ein Geoelement selektiert sein

-DIVWIRE ändert die Funktion von DIVERSE auf WIRE

-WIREDIV ändert die Funktion von WIRE auf DIVERSE;

ACHTUNG: Geoelemente mit der Funktion DIVERSE werden bei der Schaltungsrückgewinnung nicht berücksichtigt

-OBSTDIV ändert die Funktion von OBSTRUCTION auf DIVERSE

-DIVOBST ändert die Funktion von DIVERSE auf OBSTRUCTION

NAME

CHANGEGEOTYP – ändert den Typ eines Geoelementes

SYNOPSIS

CHANGEGEOTYP -PATHPOLY | -RECTPOLY | -POLYRECT | -POLYTRAPEZOID |
-PATHRECT | -PATHTRAPEZOID

BESCHREIBUNG

ändert den Komponententyp von selektierten Geoelementen; das Kommando wirkt nur auf die spezifizierten Komponenten. Andere selektierte Komponenten werden ignoriert.

OPTIONEN

- PATHPOLY wandelt Pfade in Polygone um; es wird nur die Umrisslinie gewandelt. Dabei wird eine Rasterisierung der Eckpunkte des Polygons durchgeführt. Dadurch können sich bei nicht orthogonalen Pfaden kleine Abweichungen von der Umrißlinie des Pfades ergeben.
- RECTPOLY wandelt Rechtecke in Polygone um; dieses ist notwendig, wenn zusätzliche Eckpunkte mit APPEND oder NOTCH eingefügt werden sollen
- POLYRECT zerlegt Polygone in Rechtecke; dieses ist nur möglich für orthogonale Strukturen
- POLYTRAP zerlegt Polygone in Rechtecke; dieses ist nur möglich für nicht orthogonale Strukturen
- PATHRECT zerlegt Pfade in Rechtecke; dieses ist nur möglich für orthogonale Strukturen
- PATHTRAP zerlegt Pfade in Rechtecke; dieses ist nur möglich für nicht orthogonale Strukturen

NAME

CHANGEPIN – ändert die Attribute eines PINS

SYNOPSIS

CHANGEPIN [-NAME] [-POSITION] [-CLASS]

BESCHREIBUNG

Ändert die Pin Attribute eines selektierten Geoelementes mit der Funktion PIN; im Fall von -NAME oder -POS darf nur eine Komponente selektiert sein. Es muss mindestens eine Option angegeben werden.

OPTIONEN

-NAME sw	neuer Pinname DEFAULT: alter Name wird beibehalten
-POSITION p	neue Position des Pinnamen DEFAULT: alte Position wird beibehalten
-CLASS pc	neuer Pintyp DEFAULT: alter Pintyp wird beibehalten

NAME

CHANGEPLNAME - ändert den Platzierungsnamen einer Instanz

SYNOPSIS

CHANGEPLNAME[-NAME] [-POSITION] | -EXCHANGE

BESCHREIBUNG

Das Kommando **CHANGEPLNAME** dient zum Ändern von Platzierungsnamen. Es hat zwei Arbeitsmodi, entweder kann der Platzierungsname bzw. die Position des Platzierungsnamens einer selektierten Instanz geändert werden, oder es werden die Platzierungsnamen von zwei selektierten Instanzen geändert. Im ersten Modus darf nur eine Komponente selektiert sein, im Austauschmodus müssen genau zwei Instanzen selektiert sein. Es muss mindestens eine Option angegeben werden.

OPTIONEN

- NAME c9 neuer Platzierungsname
 DEFAULT: alter Platzierungsname bleibt erhalten
- POSITION p neue Position des Platzierungsnamens
 DEFAULT: behält alte Position bei
- EXCHANGE vertauscht die Platzierungsnamen zweier selektierter Instanzen.

NAME

CHANGEPLNAMETODEFAULT - setzt Platzierungsnamen auf default Wert

SYNOPSIS

CHANGEPLNAMETODEFAULT [-ENUMERATE]

BESCHREIBUNG

Ändert die Platzierungsnamen der selektierten Instanzen auf den Defaultwert '??' oder weist den Instanznamen eine eindeutige Nummer zu.

OPTIONEN

-ENUMERATE

numeriert die Instanznamen der selektierten Instanzen, Die Zählung der Instanznamen beginnt bei der höchsten Nummer aller in der Zelle vorhandenen numerischen Instanznamen.

DEFAULT: es wird der Instanzname ?? zugewiesen.

WEITERES

SELECT(Paris), CHANGEPLNAME(Paris)

NAME

CHANGETOGROUP - ändern der Macrogruppen-Zugehörigkeit

SYNOPSIS

CHANGETOGROUP cellname -GROUP

BESCHREIBUNG

Ändert die Zugehörigkeit von Zelldefinitionen zur Macrogruppe. Die angegebene Macrogruppe muß existieren, sonst erfolgt eine Fehlermeldung. Eine Macrogruppe kann mit dem Kommando **CREATE -GROUP** erzeugt werden. Generierte Zellen aus der Macrogruppe `__GENERIC` können nicht in eine andere Gruppe geschoben werden. Diese werden als **READONLY** Zellen behandelt. Wenn eine generierte Zelle weiterbearbeitet werden soll, muß eine Kopie dieser Zelle mit **CPYCELDEF** oder **DEFCLOSE** erzeugt werden.

ARGUMENTE

cellname [...] (cn) Name(n) der Zelldefinition(en), die einer anderen Gruppe zugeordnet werden soll(en)

OPTIONEN

-GROUP mg Name der neuen Macrogruppe

FEHLER

cell not in database

cellname existiert nicht

macro group not found

Die angegebene Gruppe existiert nicht

group is generic group

Die angegebene Macrogruppe ist die Gruppe für generierte Zellen.

cell is a generic cell

Die angegebene Zelle ist eine generierte Zelle.

WEITERES

CPYCELDEF(paris), **RENCELDEF**(paris), **CREATE**(paris)

NAME

CHECK – führt formale Prüfungen in einer Zelle durch

SYNOPSIS

CHECK [cellname] [-RECURSIV] [-FORCE]

BESCHREIBUNG

führt formale Prüfungen fuer die spezifizierte(n) Zelldefinition(en) durch und erkennt dabei folgende Fehler:

- Überlappungen von Geoelementen des Typs OBSTRUCTION
- Kontaktierung von Gate Array Untergrundzellen, auf denen keine Verdrahtungszelle plaziert ist (nur Designtyp GA)

Im Fall von Gate Array Designs werden die angeschlossenen Tunnel erkannt und als Geoelemente in die aktuelle Zelldefinition kopiert. Falls keine Fehler auftreten, wird `check_flag = true` andernfalls `check_flag = false` gesetzt.

ACHTUNG: zu Anfang werden alle Fehlerflags in der definierten Fehlerebene gelöscht.

ARGUMENTE

cellname [...] (cn) Name(n) der zu überprüfenden Zelle(n) DEFAULT: aktuelle Zelldefinition

OPTIONEN

- RECURSIV falls in der zu prüfenden Zelldefinition Zellaufrufe vorhanden sind, deren Definitionen `check_flag=false` haben, werden zunächst diese Zelldefinitionen geprüft (rekursive Behandlung der kompletten Zelldefinition und deren Zellaufrufe, bis hin zu den Basiszellen)
- FORCE prüft in jedem Fall die angegebenen Zelldefinitionen, auch dann, wenn diese bereits `check_flag=true` haben.

WEITERES

NETEXT(Paris),ONLVERIFY(Paris)

NAME

CIFOUT – Ausgabe einer Zelle im CIF-Format

SYNOPSIS

CIFOUT [-LAYER] [-NOTECH] [-SUBTUN] [-WINDOW] [cellname [-OUTPUT file]] ...

BESCHREIBUNG

konvertiert die Design-Daten der spezifizierten Zelle(n) oder der gerade editierten Zelle ins CIF-Format (Caltech Intermediate Format) und erzeugt daraus eine ASCII-Datei. Default-Dateiname ist 'cellname'_lay.cif

ARGUMENTE

cellnamen [...] (cn)

Name(n) der zu konvertierenden Zelldefinition(en)
DEFAULT: aktuelle Zelldefinition;

OPTIONEN

- LAYER gl gibt nur die angegebenen Ebenen aus
DEFAULT: gibt alle Ebenen aus
- OUTPUT fn Die CIF Daten werden in der Datei mit dem Namen 'fn' abgelegt; Option ist nur bei Ausgabe einer Zelle erlaubt.
DEFAULT: cellname_lay.cif
- NOTECH Die Zellen mit der Macrofunction 'TECH' werden nicht ausgegeben (für DRC)
DEFAULT: Zellen mit Macrofunction 'TECH' werden ausgegeben
- SUBTUN Gate Array Untergrund wird mit ausgegeben
- WINDOW Es wird nur die Figuren ausgegeben, die im dargestellte Bildausschnitt sichtbar sind. Die Ausgabe erfolgt flach. Die Figuren werden nicht am Bildausschnitt abgeschnitten. Diese Option kann nur dann angegeben werden, wenn kein Zellname angegeben ist.

UNDEFINIERTER REFERENZEN

Eine Ausgabe erfolgt nur, wenn die Zelle vollständig gelinkt ist, also keine undefinierten Referenzen auf andere Zellen enthält. Dies gilt auch, wenn undefinierte Referenzen in tieferen Hierarchiestufen der Zelle enthalten sind. In diesem Fall wird ein `INSPECT -UNDEFINED` durchgeführt und die fehlenden Zellen angezeigt.

WEITERES

PLOT(PARIS), GDSOUT(PARIS), PRSOUT(PARIS)

NAME

CLEAR – Löschen aller Elemente einer Zelle

SYNOPSIS

CLEAR

BESCHREIBUNG

Löscht alle Komponente aus der aktuellen Zelldefinition. **CLEAR** wirkt nicht nur auf selektierte Komponenten, sondern leert immer die aktuell editierte Zelle, auch wenn nichts selektiert ist.

WEITERES

DELETE(PARIS)

NAME

CNVLAYER – ändert Ebenen mit einer Konvertierungstabelle

SYNOPSIS

CNVLAYER pathname [-CELL | -GROUP | -ALL] [-RECURSIV]

BESCHREIBUNG

ändert die Ebenenzuordnung aller Geometrie- und Textelemente der aktuellen Zelldefinition, der spezifizierten Zellen oder Gruppen bzw. des gesamten Layouts mit Hilfe einer Konvertierungstabelle. Bei allen gänderten Zellen wird das `net_flag` auf `false` gesetzt. Zellen in geschützten Macrogruppen, also diejenigen Gruppen, für die der Benutzer keine Schreibrechte hat, werden nicht konvertiert.

ARGUMENTE

pathname (fn) Die Datei 'pathname' enthält fuer alle zu ändernden Ebenen je eine Zeile mit der Zuordnung von altem und neuem Layer.

Format:

```
'alter Layer' 'neuer Layer'  
.  
.  
.
```

OPTIONEN

- CELL cn [...] die Ebenen der spezifizierten Zelldefinitionen werden konvertiert.
DEFAULT: aktuelle Zelldefinition
- GROUP mg [...] die Ebenen aller Zelldefinitionen der spezifizierten Macrogruppen werden konvertiert.
DEFAULT: aktuelle Zelldefinition
- ALL die Ebenen aller Zelldefinitionen des gesamten Layouts werden konvertiert.
DEFAULT: aktuelle Zelldefinition
- RECURSIV die Ebenen der Zelldefinitionen und deren Zellaufrufe werden (bis hin zu den Basiszellen) rekursiv konvertiert. Diese Option ist nur dann signifikant, wenn als Zielobjekt(e) ein oder mehrere Zellen angegeben sind.
DEFAULT: Ebenen von aufgerufenen Zellen werden nicht konvertiert

NAME

COPY – Kopieren von selektierten Komponenten

SYNOPSIS

COPY -POINT | -VECTOR | -DIST

BESCHREIBUNG

Kopiert die selektierten Komponenten und verschiebt die neuen Komponenten um den angegebenen Betrag.

OPTIONEN

-POINT p der Verschiebungsvektor setzt sich aus dem letzten MARK Punkt und 'p' zusammen

-VECTOR p1 p2 der Verschiebungsvektor setzt sich aus 'p1' und 'p2' zusammen

-DIST in1 in2 die Verschiebungsdistanz wird explizit als Zahlenpaar in internen Einheiten angegeben

WEITERES

MARK(Paris), STEP(Paris), MOVE(Paris)

NAME

CPYCELDEF – Kopieren einer Zelldefinition

SYNOPSIS

CPYCELDEF oldname newname [-GROUP]

BESCHREIBUNG

Kopiert eine Zelldefinition. Wenn die Zelle `newname` bereits existiert, wird diese nicht überschrieben, sondern es erfolgt eine Fehlermeldung. Wird eine Macrogruppe für die neue Zelle angegeben, muß diese schon existieren. Wenn die Gruppe nicht existiert, kann sie mit dem Kommando `CREATE -GROUP` erzeugt werden.

`CPYCELDEF` eignet sich auch zum Umwandeln generierter Zellen in gewöhnliche Zellen. Generierte Zellen sind intern mit dem Property `__GENERATOR__` markiert. Generierte Zellen werden in der von `PARIS` automatisch erzeugten Macrogruppe `__GENERIC` abgelegt. Diese Gruppe kann nicht abgespeichert werden, da die generierten Zellen im Generatorcache verwaltet werden. Es ist daher nicht erlaubt, generierte Zellen in eine andere Macrogruppe zu schieben oder zu ersetzen. Wenn es doch erforderlich wird, eine generierte Zelle zu bearbeiten, kann diese entweder mit `DEFCL0SE cn` oder `CPYCELDEF` in die normale Zell `cn` umgewandelt werden. Bei den Kopiervorgängen in `DEFCL0SE cn` oder `CPYCELDEF` wird das Property `__GENERATOR__` entfernt, und die Zelle ist dann unter ihrem neuen Namen als gewöhnliche Zelle handhabbar.

ARGUMENTE

oldname (cn) Name der ursprünglichen Zelldefinition
newname (cn) Name der neu erzeugten Zelldefinition

OPTIONEN

-GROUP mg Name der Macrogruppe, der die neue Zelle zugeordnet werden soll
DEFAULT: ursprüngliche Macrogruppe wird beibehalten

FEHLER

ECELLNOTINDB oldname existiert nicht
EDEFEXISTS newname existiert schon
EGROUPNOTEXISTS Die angegebene Gruppe existiert nicht

WEITERES

`CHANGETOGROUP(Paris)`, `RENCELDEF(Paris)`, `CREATE(Paris)`

NAME

CREATE - Erzeugen einer Zelle oder Macrogruppe

SYNOPSIS

CREATE -GROUP | -CELL [-GROUP] | -NET | -BAG

BESCHREIBUNG

Erzeugt leere Macrogruppe(n) oder Zelldefinition(en).

OPTIONEN

- GROUP mg [...] erzeugt die Macrogruppe(n) 'mg' bzw. erzeugt die neue(n) Zelldefinition(en) in der Macrogruppe 'mg'
- CELL cn [...] erzeugt die neue(n) leere(n) Zelldefinition(en) 'cn'
- NET n erzeugt ein Netz mit dem Namen 'n', das die selektierten Komponenten beinhaltet. Wenn bereits ein Netz mit angegebenen Namen existiert, wird es überschrieben.
- BAG n erzeugt eine Bag mit dem Namen 'n', die die selektierten Komponenten beinhaltet. Es kann mehrere Bags mit dem gleichen Namen geben.

WEITERES

INSPECT(Paris), EXIST(Paris)

NAME

CREATELIST- erzeugt Ausgabelisten der spezifizierten Zelldefinition

SYNOPSIS

CREATELIST [cellname] [-NET] [-ATR [fn]] [-BOX [fn]] [-TUN] [-ILO] [-NOPADS]
[-NOSUBCO] [-COND]

BESCHREIBUNG

Erzeugt die gewünschten Ausgabelisten der spezifizierten Zelldefinitionen. Es muss mindestens eine Option angegeben werden. Voraussetzung für die Erzeugung der Ausgabelisten ist ein erfolgreicher CHECK und NETTEXT der Zelle.

ARGUMENTE

cellname [...] (cn) Name(n) der zu bearbeitenden Zelldefinition(en)
DEFAULT: aktuelle Zelldefinition

OPTIONEN

- NET erstellt eine Netzliste im AEG-Format;
Dateiname: 'cellname'_lay.net
- ATR erstellt eine Liste mit Attributwerten. Es werden nur die Attribute von angeschlossenen Instanzen ausgegeben. Eine Instanz zählt dann als angeschlossen, wenn mindestens ein Pin an ein externes Netz angeschlossen ist und nicht alle Pins an das selbe Netz angeschlossen sind. Das Format der Attributausgabe wird über eine externe Datei gesteuert (Siehe Abschnitt **FORMATIERUNG DER ATTRIBUTLISTE**).
Dateiname: 'cellname'_lay.atr
- BOX erstellt eine Liste der angeschlossenen Widerstandsboxen und deren Inhalt;
Dateiname: 'cellname'_lay.box
- TUN erstellt eine Liste der benutzten Tunnel sortiert nach Knotennummern;
Dateiname: 'cellname'_lay.tun
- ILO erstellt einen I2L-Logfile (nur bei Designtyp I2L);
Dateiname: 'cellname'_lay.ilo
- NOPADS gilt nur fuer Netzlisten (-NET);
Zellen vom Typ IO_PAD erscheinen nicht in der Netzliste
- NOSUBCO gilt nur fuer Netzlisten (-NET);
Zellen vom Typ SUB_CONTACT erscheinen nicht in der Netzliste
- COND Die Ausgabelisten werden nur dann erzeugt, wenn die Zelle jünger ist als die Ausgabeliste. Dies gilt nur für nicht editierte Zellen. Die Ausgabelisten editierter Zellen werden immer erzeugt.

FILTERFUNKTIONEN BEI DER NETZLISTENERZEUGUNG

Die Netzliste (Schalter -NET) wird aus den Pins der Instanzen in einer Zelle erzeugt, wobei normalerweise für jeden Pin ein Eintrag in der Netzliste erscheint. In bestimmten Fällen werden dabei die Pins nicht eins zu eins übernommen, sondern es werden Modifikationen der Netzliste mit eingebauten Filterfunktionen vorgenommen, wobei bestimmte Pins herausgefiltert werden, die Namen von Pins oder Zellen ersetzt werden oder Elemente durch äquivalente Elemente ersetzt werden. In den folgenden Fällen wird eine Veränderung der Netzliste durchgeführt:

Nicht angeschlossene Bauelemente: Bauelemente die nicht angeschlossen sind, d.h. jeder Pin der Instanz ist an kein anderes Netz angeschlossen, erscheinen nicht in der Netzliste.

Kurzgeschlossene Bauelemente: vollständig kurzgeschlossene Bauelemente, d.h. Alle Pins der Instanz sind am selben Netz angeschlossen, erscheinen nicht in der Netzliste. Diese Filterung wird nur angewendet beim Designtyp TA für Linear Arrays für Zellen mit der Zellfunktion BJT, MOSFET, JFET, I2L_GATE, RESISTOR, DIODE.

Zellen mit bestimmten Zellfunktionen: Zellen mit der Zellfunktion VIA, CONTACT, MAC_UNDERPATH, C_BOX, R_BOX erscheinen nicht in der Netzliste.

Zellen mit CLASS Property: wenn eine aufgerufene Zelle ein Property mit dem Namen CLASS hat, erscheint nicht der Zellname sondern der Wert des CLASS Properties als Zellname in der Netzliste. Damit wird die Realisierung verschiedener Layoutvarianten einer Zelle ermöglicht, die alle durch eine Referenzzelle im Schematic beschreiben sind.

Substrat oder Wannenkontakte: wenn in einem Layout mehrere Zellen mit der Zellfunktion SUB_CONTACT an das selbe Netz angeschlossen sind, erscheint nur *eine* dieser Kontaktzellen in der Netzliste. Dabei wird diejenige Instanz der Substratkontaktzellen ausgewählt, die einen "sinnvollen" Instanznamen hat. Als "sinnvoller" Instanznamen wird ein Name betrachtet, der nicht ??, nicht leer und nicht numerisch ist. Gibt es keinen solchen Instanznamen, wird irgendein Instanzname der an dem Netz angeschlossen Substratkontakte ausgewählt.

Zusammengefaßte Widerstände/Kapazitäten: Netzwerke aus parallel- oder reihengeschalteten Widerständen/Kapazitäten werden durch ihr äquivalentes Bauelement ersetzt. Es erscheinen dann nicht alle Widerstände/Kapazitäten sondern jeweils nur ein Widerstand bzw. eine Kapazität. Wenn alle Bauelemente im zusammengefaßten Netzwerk den gleichen Zellnamen bzw. den gleichen Wert des CLASS Properties haben, erscheint in der Netzliste als Zellname der gemeinsame Zellname bzw. Wert des CLASS Properties. Andernfalls erscheint für Widerstände als Zellname R und für Kapazitäten C. Als Pinnamen der äquivalenten Elemente erscheinen P1 und P2 für Widerstände und C1 und C2 für Kapazitäten.

Pins mit gleichem Basisnamen: ein Pinname kann aus einem Basisnamen und einer nachgestellten Nummer bestehen. Basisname und Nummer sind durch einen . (Punkt) voneinander getrennt. Existiert eine derartige Benennung von Pins, wird davon ausgegangen, daß die so benannten Pins in ihrer Zelle nicht in einer Verdrahtungsebene miteinander verbunden sind, aber in der nächst höheren Hierarchiestufe verbunden werden sollen. Wenn alle Pins mit gleichen Basisnamen verbunden sind, erscheint dann auch nur ein Eintrag mit dem Basisnamen als Pinname in der Netzliste. Sind Pins mit gleichem Basisnamen nicht verbunden erscheinen sie einzeln in mehreren Einträgen in der Netzliste. Es ist ebenfalls möglich, zwar nicht alle Pins mit gleichem Basisnamen, wohl aber mehrere Gruppen verbundener Pins mit gleichem Basisnamen zu bilden. Auch dieser Fall wird von CREATELIST erkannt; es wird dann eine neue Enumerierung der Pins durchgeführt.

FORMATIERUNG DER ATTRIBUTLISTE

Die Formatierung der Attributliste wird über eine externe Datei gesteuert. Es lassen sich für jede Macrofunktion individuell das Format einer Zeile in der Attributdatei angeben. Fehlt eine Angabe für eine bestimmte Zellfunktion, erfolgt keine Ausgabe der Properties ihrer Instanzen. Die Formatsteuerdatei enthält für jede Zellfunktion eine Zeile mit den Formatierungsangaben in der folgenden Syntax:

```
ZellFunktion FormatAnweisung AttributName ...
```

Die Komponenten FormatAnweisung und AttributName müssen in Anführungszeichen stehen, wenn diese Leerzeichen enthalten. In der Formatanweisung wird die Spezifikation %p durch den Namen eines Attributs und die Spezifikation %v durch den Wert eines Attributs ersetzt. Alle anderen Zeichen erscheinen wie angegeben in der Attributliste. Die Attribute werden in der angegebenen Reihenfolge durchlaufen.

Hier die voreingestellte Formatspezifikation (\$AEGDIR/paris/v5/atr.fmt):

```
RESISTOR    "%p=%v"      "R"  
CAPACITOR  "%p=%v"      "C"  
INDUCTOR   "%p=%v"      "L"  
BJT        "%p=%v"      AREA  
DIODE      "%p=%v"      AREA  
MOSFET     "%p=%v %p=%v"  "W" "L"
```

Damit ergibt sich z.B. folgende Ausgabe:

```
R1: R=10k  
C1: C=10p  
Q1: AREA=1  
M1: W=10u L=100u
```

WEITERES

CHECK(PARIS),NETEXT(PARIS),ONLVERIFY(PARIS),RCEQUI(PARIS)

NAME

CUT – zerteilt selektierte Komponenten

SYNOPSIS

CUT [-LINE | -POINT | -MAXPTS [-NODEL]]

BESCHREIBUNG

schneidet die selektierten Komponenten entweder an einer Linie oder so lange, bis eine vorgegebene Punktzahl unterschritten wird. Die Linie ist durch zwei Punkte durch die Option LINE oder POINT mit dem vorhergehenden MARK Punkt definiert. Es entstehen neue Komponenten, die ursprüngliche bleibt selektiert oder wird gelöscht.

OPTIONEN

- LINE p1 p2 die Strecke von 'p1' nach 'p2' bestimmt die Schnittlinie
- POINT p die Strecke vom letzten MARK Punkt und 'p' bestimmt die Schnittlinie
- MAXPTS i die Polygone werden rekursiv so lange abwechselnd horizontal und vertikal in der Mitte ihrer Bounding Box geschnitten, bis keines der entstehenden Schnittpolygone die angegebene Punktzahl überschreitet. Pfade werden dabei in Polygone umgewandelt. Durch Einrasten auf das eingestellte Minimalraster kann es zu geringfügigen Änderungen gegenüber der vorherigen Geometrie kommen.
- NODEL die ursprüngliche Komponente wird nicht gelöscht. erhalten.
DEFAULT: die ursprüngliche Komponente wird nach dem Schnitt gelöscht

WEITERES

MARK(Paris), FIND(Paris)

NAME

DEFDCLOSE – beendet das Editieren einer Zelle mit Rückspeichern

SYNOPSIS

DEFDCLOSE [cellname] [-NOVERIFY] [-FORCE] [-EXCHANGE] [-NOCLOSE] [-GROUP]

BESCHREIBUNG

beendet das Editieren einer Zelldefinition, was durch **CHANGEDEF** oder **EDITINPLACE** begonnen wurde. Dabei wird zunächst **CHECK** und **NETEXT** ausgeführt (**-VERIFY**) und bei erfolgreichem Abschluß versucht, die alte Zelldefinition durch die Neue zu ersetzen. Falls die Zelle schon plaziert ist, werden auch die Zelldefinitionen ueberprüft, in denen die Zelle aufgerufen ist, bis hin zur obersten Hierarchiestufe. Dies kann je nach Komplexität der Zellen einige Zeit dauern.

ARGUMENTE

cellname (cn) bei Angabe eines neuen Zellnamens wird eine neue Zelldefinition erstellt (nicht erlaubt nach 'defstart')
DEFAULT: die alte Zelldefinition wird überschrieben

OPTIONEN

-NOVERIFY `check_flag` und `net_flag` werden auf `false` gesetzt. Falls die Zelle plaziert war und die aufrufenden Zellen beide Prüfflags auf `true` hatte, wird der Zellaufruf nicht ersetzt und die Definition wird nicht abgeschlossen. Soll in jedem Fall ersetzt werden, so muss die Option **-FORCE** angegeben werden.
DEFAULT: **CHECK** und **NETEXT** wird ausgeführt

-NOCLOSE Die Zelle wird in die Datenbasis zurückgeschrieben. Das Editfenster bleibt aber offen, so daß die Zelle weiter bearbeitet werden kann.

-FORCE falls die bearbeitete Zelle aufgerufen war, werden die Plazierungen in jedem Fall ersetzt (auch wenn dadurch Fehler auftreten). Im Fall eines Fehlers in einer aufrufenden Zelle werden deren **CHECK** -bzw **NETFLAGS** auf `false` gesetzt.

-EXCHANGE wenn die in Place editiert war, wird die vorherige Instanz gegen eine Instanz der geänderten Zelle ausgetauscht.
DEFAULT: die Instanz der vorherigen Zelle bleibterhalten.

-GROUP mg Macrogruppe, in der die neue Definition abgelegt wird (falls eine neue Definition erstellt wird)
DEFAULT: die neue Zelldefinition ist in der gleichen Macrogruppe wie die ursprüngliche.

WEITERES

CHANGEDEF(Paris), **DEFQUIT(Paris)**, **ATTACH(Paris)**

NAME

DEFFROMSELECT – erstellt neue Zelldefinition aus selektierten Komponenten

SYNOPSIS

DEFFROMSELECT cellname [-MACROFUNC] [-GROUP] [-KEEPORIGIN]
[-REPLACE] [-KEEPPIN]

BESCHREIBUNG

erstellt eine neue Zelldefinition aus den selektierten Komponenten und tauscht sie (optional) gegen die selektierten Komponenten aus

ARGUMENTE

cellname (cn) Name der neuen Zelldefinition

OPTIONEN

-MACROFUNC mf
spezifiziert die Macrofunktion der neuen Zelldefinition
DEFAULT: TYPUNDEF

-GROUP mg Name der Macrogruppe, zu der die neue Zelle gehören soll
DEFAULT: neue Zelldefinition wird der aktuellen Macrogruppe zugeordnet

-KEEPORIGIN Nullpunkt liegt an der gleichen Position wie in der aktuellen Zelldefinition.
DEFAULT: Als Nullpunkt der neuen Zelle wird ihre linke untere Ecke gesetzt.

-REPLACE [c9] ersetzt die selektierten Komponenten gegen die soeben generierte Zelle und vergibt (optional) den Plazierungsnamen 'c9' an der Position (0,0)
DEFAULT: es wird keine Ersetzung durchgeführt, sondern nur eine neue Zelldefinition erstellt

-KEEPPIN in der Selektliste vorhandene Pins bleiben in der neuen Zelldefinition erhalten
DEFAULT: Pins werden in WIRE umgewandelt

WEITERES

SELECT(PARIS)

NAME

DEFINE – definiert Menü oder Funktionstasten

SYNOPSIS

DEFINE -POPUP

DEFINE -PULLDOWN -PARENT [-TITLE]

DEFINE -MENU -ITEM [-MNEMONIC] -CASCADE menu | command

DEFINE -KEY kn command

BESCHREIBUNG

erstellt neues Menü, ändert oder erstellt Menü-Einträge in vorhandenen Menüs oder definiert Funktionstasten

ARGUMENTE

command (cs)	Kommando oder Kommandofolge, mit der das Menüfeld oder die Funktionstaste belegt werden soll ACHTUNG: der commandstring steht immer am Ende des DEFINE Kommandos
menu	der Name eines Menüs
kn	eine Eventbezeichnung wie in den X-Window Intrinsics.
item	ein Eintrag in einem Menü.
c	ein einzelnes Zeichen zur Mnemonicbezeichnung eines Menüeintrags. C muß in item vorkommen. Die Mnemonics müssen innerhalb eines Menüs eindeutig sein.

OPTIONEN

-POPUP	erstellt ein Popup-Menü; menu muß spezifiziert werden. Es kann nur ein Popup-Menü definiert werden.
-PULLDOWN	erstellt ein Pulldown-Menü; menu muß spezifiziert werden. Ein Pulldown-Menü muss immer ein Parent-Menü haben. Das Parent-Menü kann der Menübalken oder ein anderes Pulldown-Menü sein. Das Parent-Menü muss vorher definiert worden sein.
-PARENT menu	Spezifiziert das Parent-Menü, unter dem ein Pulldown-Menü aufgehängt wird. Das Menü paris_bar wird von PARIS immer erzeugt. Paris_bar ist der Menübalken des Arbeitsfensters.
-TITLE title	spezifiziert den Titel eines Menüs.
-MENU menu	gibt an, in welchem Menü ein Menüeintrag aufgehängt wird.
-ITEM item	bezeichnet den Namen eines Menüeintrags. Dieser Name erscheint auch als Text in dem Menü.
-MNEMONIC c	bezeichnet das Mnemonic Zeichen eines Menüeintrags.
-CASCADE menu	erzeugt einen Menüeintrag, mit dem ein Pulldown-Menü verknüpft ist.
-KEY	definiert eine Tastenbelegung.

NAME

DEFLAYER – ändert Parameter einer Ebene

SYNOPSIS

DEFLAYER layer [-ALL] [-NAME] [-WIDTH] [-COLOR] [-FILL] [-NEITHER | -VISIBLE
| -DETECTABLE] [-OUTLINE] [-CIFLAYER][[-STORE]

BESCHREIBUNG

Definiert die Parameter einer Ebene bzw. setzt bereits vorhandene Parameter um. Mindestens eine Option muß angegeben werden.

ARGUMENTE

layer (1)

Nummer der zu definierenden Ebene. Die Ebenennummer muß immer angegeben sein.

OPTIONEN

- STORE** Die Layereinstellungen des aktuellen Fensters wird der globalen Layereinstellung zugewiesen. Dieser Schalter wirkt nur, wenn eine Zelle zur Bearbeitung offen ist, andernfalls ist er ohne Funktion.
- ALL** Die Layereinstellungen werden an allen Fenstern und der Technologiebeschreibung durchgeführt.
DEFAULT: die Einstellung wird im aktuellen Fenster wirksam, wenn DEFLAYER aus einer editierten Zelle aufgerufen wird, bzw. in der Technologie, wenn keine Zelle editiert ist.
- NAME ln** Ebenenname (sollte spezifiziert werden)
DEFAULT: es wird kein neuer Name vergeben
- WIDTH in** voreingestellte Weite fuer zu addierende Pfade in dieser Ebene
DEFAULT: 0
- COLOR c** Ebenenfarbe
DEFAULT: Farbe 1 (weiss)
- FILL f** Nummer des Füllmusters DEFAULT: Muster 1 (ungefüllt)
- NEITHER** Ebene ist unsichtbar
DEFAULT: detectable
- VISIBLE** Ebene ist sichtbar, aber nicht editierbar
DEFAULT: detectable
- DETECTABLE** Ebene ist sichtbar und editierbar
- OUTLINE ol** Umrandungsmuster (z. Zt. nur 1 = durchgehende Linie möglich)
DEFAULT: 1
- CIFLAYER c4** Layerstring für die Ausgabe im CIF-Format
DEFAULT: 0

WEITERES

SETLAYER(Paris),LOADPAR(Paris),SAVEPAR(Paris),INSPECT(Paris)

NAME

DEFQUIT – Abbruch des Editierens einer Zelle

SYNOPSIS

DEFQUIT

BESCHREIBUNG

beendet das Editieren einer Zelle, ohne die Änderungen abzuspeichern (Abbruch).

WEITERES

CHANGEDEF(paris), DEFCLOSE(paris)

NAME

DELCELDEF – löscht Zelle(n) aus der Datenbasis

SYNOPSIS

DELCELDEF cellname | -UNUSED [-EXCEPT] [-ALL]

BESCHREIBUNG

löscht die spezifizierte(n) Zelldefinition(en) bzw. alle nicht aufgerufenen Zellen innerhalb einer Gruppe aus der Datenbasis. Es werden nur Zellen gelöscht, die nicht plaziert sind

ARGUMENTE

cellname [...] (cn) Name(n) der zu löschenden Zelldefinition(en)

OPTIONEN

- UNUSED mg löscht alle nicht plazierten Zelldefinitionen innerhalb der Macrogruppe 'mg'
VORSICHT: Zellen auf der obersten Hierarchiestufe (z.B. das Gesamtlayout) sind auch nicht plaziert
- EXCEPT cn [...] nur mit Option -UNUSED: die spezifizierte(n) Zelldefinitionen werden nicht gelöscht
- ALL nur mit Option -UNUSED: es werden alle unbenutzten Zellen gelöscht, also auch diejenigen, die nach dem Löschen einer unbenutzten Zelle unbenutzt sind. Im Prinzip wird also solange DELCELDEF -UNUSED durchlaufen, bis in der spezifizierten Macrogruppe keine unbenutzten Zellen übrigbleiben.

WEITERES

INSPECT(Paris), RENDELDEF(Paris), CHANGETOGROUP(Paris)

NAME

DELETE – löscht selektierte Komponenten

SYNOPSIS

DELETE [-TOP]

BESCHREIBUNG

löscht selektierte Komponenten oder (und) Punkte von Geoelementen aus der aktuellen Zelldefinition

OPTIONEN

-TOP

löscht nur das erste Element der Selektliste

DEFAULT: löscht alle selektierten Komponenten

WEITERES

CLEAR(PARIS)

NAME

DESCEND – ändert die Descend Attribute

SYNOPSIS

DESCEND -CELL cn | -GROUP gn -LAYER lset -DESCEND onloff [-PROPAGATE]

DESCEND -SEL -LAYER lset -DESCEND onloff

BESCHREIBUNG

Jeder Zelle und jeder Instanz sind als Descend Attribute ein Layerset und ein Descend Flag zugeordnet. Mit Hilfe der Descend Attribute wird die Darstellung von Instanzen beim Bildaufbau gesteuert. Wenn das Descend Flag gesetzt ist, steigt die Zeichenfunktion in die Hierarchie ab und zeichnet den Inhalt der Zelle. Dabei werden nur diejenigen Komponenten gezeichnet, deren Layer im Descend Layerset enthalten sind. Wenn das Descend Flag nicht gesetzt ist, steigt die Zeichenfunktion nicht weiter in die Hierarchie ab, sondern zeichnet die Instanz symbolisch, d.h. es wird das Interface der Instanz gezeichnet bestehend aus der Bounding Box und den Pins der Instanz.

Für den Bildaufbau relevant sind die Descend Attribute an Instanzen. Die Descend Attribute an Zellen dienen als Voreinstellung, wenn von der Zelle eine Instanz erzeugt wird. Eine Änderung der Descend Attribute an einer Zelle führt daher nicht direkt zu einer Änderung der Darstellung ihrer Instanzen. Mit Hilfe des -PROPAGATE Schalters können die Descend Attribute einer Zelle an ihre Instanzen weitergegeben werden. Dann erfolgt ein Neuzeichnen aller betroffenen Instanzen.

Wird die Descend Behandlung nicht gewünscht, kann diese global ausgeschaltet werden mit dem Kommando `SETPAR -DESCEND OFF`. Wenn der globale Descend Schalter aus ist, wird immer alles dargestellt. Mit `SETPAR -DESCEND ON` kann die Descend Behandlung wieder aktiviert werden.

ARGUMENTE

cellname (cn)	Name der bearbeiteten Zellen
mg	Name der bearbeiteten Macrogruppen
lset	Menge von Layern

OPTION

-CELL cn ...	Es werden die angegebenen Zellen bearbeitet
-GROUP mg ...	Es werden alle Zellen in den angegebenen Macrogruppen bearbeitet
-SEL	Es werden die selektierten Instanzen bearbeitet.
-LAYER lset	Das Descend Layerset wird angegeben.
-DESCEND onloff	Das Descend Flag wird ein- oder ausgeschaltet.
-PROPAGATE	nur bei Optionen -CELL oder -GROUP. Die Descend Attribute der spezifizierten Zellen werden an die Instanzen dieser Zellen weitergereicht.

WEITERES

SETPAR(PARIS)

NAME

DF – Abschluß einer angefangenen Komponente

SYNOPSIS

DF [-RELATIVE]

BESCHREIBUNG

Abschluß einer Komponenteneingabe die mit DS erfolgte; die genaue Bedeutung ist vom vorhergehenden ADD Befehl abhaengig;

- ADDCELL endgültige Plazierung (Bestätigung der letzten DS Eingabe)
- ADDTEXT endgültige Plazierung (Bestätigung der letzten DS Eingabe)
- ADDGEO Abschluß eines Geoelementes
- APPEND Abschluß des zu erweiternden GEOELEMENTS

Wenn der Typ der angefangenen Komponente Rechteck oder Kreis ist, muss dem DF Befehl eine Koordinate mitgegeben werden, der den zweiten Punkt des Rechtecks bzw. einen Punkt auf dem Kreis angibt.

OPTIONEN

–RELATIVE p Hierbei gibt die Koordinate 'p' den Abstand zum zuvor eingegebenen Punkt an.

WEITERES

DS(Paris), DP(Paris), ADDGEO(Paris), ADDCELL(Paris), ADDTEXT(Paris),
CANCELCOMP(Paris)

NAME

DIST – Abstand zwischen zwei Punkten

SYNOPSIS

DIST

BESCHREIBUNG

mißt den Abstand zwischen zwei Punkten. Nach Aufruf des Kommandos erscheint im Grafikfenster ein Fadenkreuz zur Eingabe des Startpunktes. Die Eingabe des Startpunkts erfolgt mit der Maus. Die Maustaste muss nach der Eingabe weiterhin gedrückt bleiben. Im Grafikfenster erscheint die Distanzlinie und im Messagefenster wird fortlaufend die Distanz in internen Einheiten und in Mikrometern angezeigt. Nach dem Loslassen der Maustaste ist das Kommando abgeschlossen.

NAME

DP – Löschen des letzten Punktes eines offenen Geoelements

SYNOPSIS

DP

BESCHREIBUNG

Löschen des zuletzt eingegebenen Punktes (nur bei ADDGEO und APPEND); nur möglich, wenn noch kein DF eingegeben wurde.

WEITERES

DS(Paris), DF(Paris), ADDGEO(Paris), APPEND(Paris)

NAME

DS – Eingabe eines Punktes fuer ADD

SYNOPSIS

DS [-RELATIVE]

BESCHREIBUNG

Punkteingabe, dessen Bedeutung vom vorhergehenden ADD Befehl abhaengt:

- ADDCELL Testplazierung
- ADDTEXT Testplazierung
- ADDGEO Eckpunkte, einschl. Endpunkt
- APPEND Eckpunkte, einschl. Endpunkt

OPTIONEN

- RELATIVE p Gibt an, dass der Punkt relativ zum vorhergehenden Punkt zu addieren ist.
Diese Option ist nur gültig, wenn schon eine Komponente angefangen ist.

WEITERES

ADDGEO(Paris), ADDCELL(Paris), ADDGCELL(Paris), ADDTEXT(Paris), APPEND(Paris),
CANCELCOMP(Paris), DP(Paris), DF(Paris)

NAME

EDITINPLACE – Editieren einer aufgerufenen Zelle in ihrer Umgebung

SYNOPSIS

EDITINPLACE [-IDENT | -UP | -DOWN]

BESCHREIBUNG

stellt die spezifizierte Zelle in ihrer Plazierungsumgebung zum Editieren bereit, d.h. es wird eine Kopie der Zelldefinition angelegt, auf der gearbeitet wird. Dieser Arbeitsschritt wird, ebenso wie 'CHANGEDEF', mit dem Kommando DEFCLOSE bzw. DEFQUIT abgeschlossen.

Es ist möglich in dem dabei aufgebauten Editierhierarchie auf- oder abzustiegen. Damit kann in verschiedenen Hiarchiestufen des Layouts gearbeitet werden. Die Editierhierarchie wird in einem Stapel (Edit Stack) für jedes Fenster verwaltet. Der Edit Stack kann mit Hilfe des Kommandos INSPECT -EDITSTACK angezeigt werden. Die aktuelle Position im Stack ist dabei mit einem * in der ersten Spalte gekennzeichnet. Die Kommandos DEFCLOSE bzw. DEFQUIT können nur für Zellen am Ende des Edit Stackes aufgerufen werden. Es ist also nicht möglich, eine Zelle zu schließen, die in der Mitte des Edit Stacks liegen. Der Edit Stack kann immer nur von unten nach oben geschlossen werden.

Wenn man sich in einer höheren Hierarchiestufe des Edit Stacks befindet, sind die gerade editierten Instanzen nicht selektierbar.

OPTIONEN

- IDENT die identifizierte Zelle (es darf nur genau eine Zelle identifiziert sein) wird editiert
DEFAULT: selektierte Zelle
- UP stiegt in der Editierhierarchie eine Stufe nach oben. Die vorhergehende Zelle bleibt dabei editiert.
- DOWN stiegt in der Editierhierarchie eine Stufe nach unten. Die vorhergehende Zelle bleibt dabei editiert.

WEITERES

DEFCLOSE(Paris), DEFQUIT(Paris), SELECT(Paris), IDENTIFY(Paris), INSPECT(PARIS)

NAME

ENTER – Anfordern einer Eingabe

SYNOPSIS

ENTER [-PROMPT] -POINT | -DIST | -LINE | -WINDOW | -STRING | -INT | -FLOAT |
-FILENAME | -GROUPNAME | -CELLNAME | -ACTIVECELL | -PINNAME |
-PLNAME | -ACTLAYER | -TEXTSIZE | -TEXTSTRING | -SIZEPINNAME |
-SIZEPLNAME | -SIZECELLNAME | -PCIRCLE | -PATHWIDTH | -GRIDFACTOR
[-SNAP]

BESCHREIBUNG

Fordert eine Eingabe vom Benutzer über die Maus bzw. die Tastatur an. Der eingegebene Wert wird anschliessend auf die Standardausgabe ausgegeben und kann z.B. über eine Kommandosubstitution als Eingabe fuer andere Kommandos dienen. Wird hauptsächlich für Macros, Menü- und Funktionstastenbelegungen verwendet.

OPTIONEN

-POINT	liest ein Koordinatenpaar über die Maus ein
-DIST	gibt die Distanz zwischen zwei Punkten aus (Cursor-Typ: Gummiband)
-LINE	gibt den Anfang- und Endpunkt einer Linie aus
-WINDOW	liest Fensterkoordinaten (Punktepaar) über die Maus ein (Cursor-Typ: Rechteck)
-STRING	liest einen String über die Tastatur ein
-INT	liest einen Wert vom Typ Integer über die Tastatur ein
-FLOAT	liest eine Gleitkommazahl über die Tastatur ein
-FILENAME	öffnet eine File-Selection-Box und fordert die Eingabe eines Dateinamens an.
-GROUPNAME	öffnet eine Liste aller Macrogruppen
-CELLNAME	fordert die Eingabe eines Zellnamens an. Dabei werden alle geladenen Zellnamen dargestellt.
-ACTIVECELL	fordert die Eingabe eines Zellnamens an. Dabei werden nur die Zellnamen dargestellt, die in der aktuellen Hierarchiestufe zur Instanziierung erlaubt sind, also diejenigen, durch deren Instanziierung keine rekursiven Instanzen auftreten.
-PINNAME	fordert die Eingabe eines Pinnamens an. Wenn ein Element vom Typ Pin selektiert ist, erscheint als Voreinstellung der Pinname des selektierten Elements.
-PLNAME	fordert die Eingabe eines Instanznamens an. Wenn ein Element vom Typ Instanz selektiert ist, erscheint als Voreinstellung der Instanzname des selektierten Elements.
-ACTLAYER	fordert die Eingabe des Layers für die nächste zu addierende Geometrie Komponente an.
-TEXTSTRING	fordert die Eingabe eines Strings an. Wenn ein Element vom Typ Text selektiert ist, erscheint als Voreinstellung der Text des selektierten Elements.
-TEXTSIZE	fordert die Eingabe der Textgröße an. Als Voreinstellung erscheint die Textgröße aus dem Parameterfile bzw. des aktuellen Fensters.

- SIZEPLNAME fordert die Eingabe der Textgröße fuer Instanznamen an. Als Voreinstellung erscheint die Textgröße aus dem Parameterfile bzw. des aktuellen Fensters.
- SIZEPINNAME fordert die Eingabe der Textgröße fuer Pinnamen an. Als Voreinstellung erscheint die Textgröße aus dem Parameterfile bzw. des aktuellen Fensters.
- SIZECELLNAME fordert die Eingabe der Textgröße fuer Zellnamen an. Als Voreinstellung erscheint die Textgröße aus dem Parameterfile bzw. des aktuellen Fensters.
- PCIRCLE fordert die Eingabe der Punktzahl fuer Kreise an. Als Voreinstellung erscheint die Kreisauflösung aus dem Parameterfile bzw. des aktuellen Fensters.
- PATHWIDTH fordert die Eingabe der Breite des nächsten zu addierenden Pfades an.
- GRIDFACTOR fordert die Eingabe der Gridfaktors an. Als Voreinstellung erscheint der Gridfaktor aus dem Parameterfile bzw. des aktuellen Fensters.
- PROMPT String, der nach Kommandoeingabe auf dem Bildschirm erscheint
- SNAP ON | OFF gibt an, ob die Koordinaten bei Point-, Dist-, Area- Eingabe eingerastet werden oder nicht.
DEFAULT: ON

NAME

EPSOUT - Ausgabe einer Zelle im EPS Format

SYNOPSIS

EPSOUT [-LAYER [-SCALE] [-OFFSET] [-NOTEXT] name [-OUTPUT] ...

BESCHREIBUNG

Mit dem `EPSOUT` Kommando werden einzelne Zellen in das EPS-Format (Extended Postscript) übersetzt und als Datei abgelegt. Die Zellen werden dabei rekursiv durchlaufen und die Geometrie in den angegebenen Layern ausgegeben. Im Gegensatz zum `PLOT` Kommando wird die Geometrie immer exakt skaliert. Es findet kein Clipping und keine Rücksichtname auf Papiergrößen statt. Die Koordinaten werden relativ zur linken unteren Ecke der Zelle ausgegeben, es findet also eine entsprechende Verschiebung statt, wenn der Nullpunkt der Zelle nicht mit ihrer linken unteren Ecke übereinstimmt. Diese Verschiebung kann mit der `-OFFSET` Option verändert werden. Die Ausgabe erfolgt mit den in `PARIS` definierten Farben und Füllmustern. Wenn die Ausgabe in anderen Farben/Füllmustern gewünscht wird, müssen diese vorher umgestellt werden.

Die `PARIS` Textkomponente wird als Textelement ins EPS-Format übertragen, Platzierungsnamen, Pinnamen werden nicht ausgegeben.

Die Reihenfolge der Argumente und Optionen muß unbedingt eingehalten werden.

ARGUMENTE

name [...] (cn) Name(n) der Zelle(n) , die ins EPS-Format übersetzt werden sollen. Die Ausgabe erfolgt auf den Dateinamen `<name>_lay.eps`. Wird mit `-OUTPUT filename` ein Dateiname angegeben, erfolgt die Ausgabe auf diesen Dateinamen.
DEFAULT: wird kein Zellname angegeben, wird die Zell im aktuellen Fenster ausgegeben.

OPTIONEN

`-LAYER gl`
Gibt die auszugebenden Ebenen an.

`-SCALE fl`
Die Zeichnung wird in der Ausgabe um den Faktor `fl` skaliert. Dabei wird die `PARIS` interne `RESOLUTION` zur Berechnung der Geometriegröße als Basis verwendet.
DEFAULT: Die Ausgabe erfolgt im Maßstab 1:1.

`-OFFSET dx dy`
Die Zeichnung wird um den Offset (`dx`, `dy`) gegenüber der linken unteren Ecke verschoben. `Dx` und `dy` sind Angaben in internen Einheiten. Der Offset wird bei der Ausgabe alle anderen Koordinaten skaliert, falls ein Skalierungsfaktor angegeben ist. Durch die Angabe eines Offsets kann ein Bildrand erzeugt werden, so daß das Bild auf einem Drucker gedruckt werden kann, der einen bestimmte Randbreite verlangt.

`-NOTEXT`
Es werden keine Texte in die EPS Datei ausgegeben. Dies betrifft Textelemente. Die Ausgabe von Texten erfolgt in Courier Schrift.
DEFAULT: Texte werden ausgegeben.

`-OUTPUT fn`
Gibt an, daß die Ausgabe der mit `name` spezifizierten Zelle oder Macrogruppe auf `fn` erfolgen soll
DEFAULT: es wird auf den Pfadnamen `name_lay.eps` ausgegeben

UNDEFINIERTER REFERENZEN

Wenn eine Zelle exportiert wird erfolgt nur dann eine Ausgabe, wenn die Zelle vollständig gelinkt ist, also keine undefinierten Referenzen auf andere Zellen enthält. Dies gilt auch, wenn undefinierte Referenzen in tieferen Hierarchiestufen der Zelle enthalten sind. In diesem Fall wird ein `INSPECT -UNDEFINED` durchgeführt und die fehlenden Zellen angezeigt.

WEITERES

`view(PARIS)`, `inspect(PARIS)`, `setlayer(PARIS)`, `defflayer(PARIS)`

NAME

EQUIDIST – Anordnen von Komponenten auf gleichen Abstand

SYNOPSIS

EQUIDIST distance -HORIZONTAL | -VERTICAL

BESCHREIBUNG

Mit dem Kommando EQUIDIST werden voll selektierte Komponenten so angeordnet, daß zwischen den Komponenten der angegebene Abstand eingehalten wird. Bezugsmaß für die Abstandsberechnung ist das umschließende Rechteck der einzelnen Komponenten.

ARGUMENTE

distance (in) Gibt den Abstand zwischen den anzuordnenden Komponenten an. Die Angabe eines negativen Abstandswertes ergibt eine überlappende Anordnung.

OPTIONEN

-HORIZONTAL Der Abstand in horizontaler Richtung wird eingestellt

-VERTICAL Der Abstand in vertikaler Richtung wird eingestellt

WEITERES

SELECT(Paris), ALIGN(Paris)

NAME

EXCHANGE – austauschen von Zellen in einem Zellaufruf

SYNOPSIS

EXCHANGE [-CELL cn|-GCELL gn] -NEWCELL cn

EXCHANGE [-CELL cn|-GCELL gn] -NEWGCELL gn [-DIALOG] [gargs]

BESCHREIBUNG

Tauscht eine oder mehrere Zellaufrufe gegen neue Zellaufrufe aus; dabei bleibt der Platzierungsname und die Instanztransformation erhalten. Im Fall von GA und TA werden die Platzierungsmerkmale der alten und neuen Zellen verglichen: ein Austausch ist nur bei gleichen Platzierungsmerkmalen zugelassen. Wenn die neue Zelle nicht mit der Transformation einer vorhandenen Instanz plaziert werden darf, wird ebenfalls der Austausch verweigert.

EXCHANGE erlaubt jede beliebige Kombination des Austauschs von Instanzen, also normale Instanzen gegen normale- oder generische Instanzen oder generische Instanzen gegen normale- oder generische Instanzen.

Wenn die neuen Instanzen generische Instanzen sind, Option -NEWGCELL, können neue Generatorparameter angegeben werden. Die neu angegebenen überschreiben bereits vorhandene, wenn die auszutauschenden Instanzen ebenfalls generische Instanzen sind. Es werden jedoch nur diejenigen Generatorparameter überschrieben, die tatsächlich angegeben werden. Nicht angegebene Parameter werden von den bereits bestehenden generischen Instanzen übernommen. Damit lassen sich alle Instanzen, die von einem bestimmten Generator erzeugt wurden, unter Beibehaltung ihrer Generatorparameter durch Instanzen von einem anderen Generator ersetzen.

Falls der neue Generator andere Parameter hat, als der alte, werden die zum alten Generator gehörenden Parameter entfernt. Die neuen Instanzen erscheinen dann konsistent als Instanzen des neuen Generators. Eine Übernahme der Generatorparameter der alten Instanzen in die Dialogbox zur Parametereingabe als Voreinstellung findet nicht statt. Der Grund ist, daß EXCHANGE nicht nur eine Instanz austauschen kann, sondern beliebig viele, so daß dann eine eindeutige Zuordnung der Generatorparameter nicht möglich ist. Ein Austausch mit Voreinstellung der vorhandenen Generatorparameter kann mit dem Kommando REGENERATE(PARIS) vorgenommen werden. REGENERATE wirkt aber nur auf eine Instanz und kann nicht den Generator wechseln.

ARGUMENTE

gargs In der zweiten Form zum Erzeugen generischer Instanzen können die Generatorparameter im Argument gargs angegeben werden. Neue Generatorargumente überschreiben vorhandene. Die Angabe von gargs ist optional.

OPTIONEN

-CELL cn tauscht alle Zellaufrufe vom Typ 'cn' gegen die Zelle 'newcell' aus
DEFAULT: alle selektierten Instanzen

-GCELL gn tauscht alle generischen Instanzen aus, die mit dem Generator gn erzeugt wurden.
DEFAULT: alle selektierten Instanzen

-NEWCELL cn Name der neu zu plazierenden Zelle

-NEWGCELL gn Name des Generators für die neu zu erzeugenden Instanzen.

-DIALOG Wenn -DIALOG angegeben wird, wird vor dem Austauschen eine Dialogbox zur Eingabe der Generatorparameter angezeigt. Diese Option ist nur in der zweiten Form, die generische Instanzen erzeugt, von EXCHANGE erlaubt. Zusätzlich zum Dialog können auch Generatorparameter angegeben werden. Diese werden dann als Voreinstellung im Dialog übernommen. Andernfalls werden in der Dialogbox die Defaultwerte der

Generatorparameter als Voreinstellung angeboten.

WARNUNGEN

`inconsistent pin list`

Diese Meldung erscheint dann, wenn die neue Instanz andere Pins hat als die alte ausgetauschte. EXCHANGE versucht immer, die Pinliste der Instanz zu erhalten. Dies ist jedoch nur dann möglich, wenn die Portlisten der alten und neuen Zelle übereinstimmen. Andernfalls wird die Pinliste der neuen Instanz an die Portliste der neuen Zelle angepaßt. Dabei gibt PARIS dann die Warnung `inconsistent pin list` aus. Diese Meldung ist eine Warnung und dient zur Information des Benutzers. Der Austausch wird trotzdem durchgeführt.

`invalid generator parameter`

Diese Meldung erscheint dann, wenn der alte Generator andere Parameter hat als der neue Generator. Der alte Generatorparameter wird entfernt.

`exchange of same cell not performed`

Die neue Zelle ist identisch mit der Alten. Ein Austausch wird in diesem Fall nicht vorgenommen.

FEHLER

`can't exchange unresolved instance`

Es wird versucht, eine Instanz auszutauschen, deren Zelle nicht existiert.

`component is protected for selected operation`

Es wird versucht, eine Instanz auszutauschen, die gegen Austausch geschützt ist.

`generate error`

Bei der Generierung der neuen Instanz ist ein Fehler aufgetreten.

`different place character of old and new cell`

Die neue Zelle hat ein anderes Platzierungsmerkmal als die alte und darf daher nicht ausgetauscht werden.

`invalid symmetry key during base cell load`

Die neue Zelle darf nicht mit der Transformation der alten Instanz plaziert werden.

WEITERES

REGENERATE(PARIS), ADDCELL(PARIS), ACTIVCELL(PARIS), FIND(PARIS),
SELECT(PARIS)

NAME

EXIST – prüfen, ob ein Objekt existiert

SYNOPSIS

EXIST -CELL | -GROUP | -EDITCELL

BESCHREIBUNG

Das Kommando **EXIST** liefert den Status 0 (**TRUE**), wenn das angegebene Objekt existiert, sonst den Status 1 (**FALSE**). Der so gelieferte Status kann in der A-Shell weiterverwendet werden.

OPTIONEN

-CELL cn Prüft, ob die Zelle *cn* existiert.
-GROUP mg Prüft, ob die Gruppe *mg* existiert.
-EDITCELL cn Prüft, ob die Zelle *cn* editiert ist.

WEITERES

INSPECT(*paris*)

NAME

FETCH – holen von Komponenten aus selektierten Instanzen

SYNOPSIS

FETCH [-DST] <-LAYER | -PIN> [-BB] [-KEEPPIN] [-RECURSIVE]

BESCHREIBUNG

FETCH holt die spezifizierten Komponenten aus selektierten Instanzen auf die aktuelle Hierarchiestufe. Die Komponenten werden dabei dupliziert und mit der Instanztransformation transformiert. Sie erscheinen dann als Geometrieelemente selektiert in der aktuellen Hierarchiestufe. Es können entweder die Komponenten selbst oder deren umschließende Rechtecke geholt werden. Für die neu erzeugten Elemente kann ein Ziellayer angegeben werden. Fehlt diese Angabe, erscheinen sie in ihrem alten Layer. Das Kommando arbeitet nicht rekursiv, d.h. es wird nur eine Hierarchiestufe der selektierten Instanzen bearbeitet.

OPTIONEN

- DST layer Die neuen Elemente können in den angegebenen Ziellayer.
- LAYER layerset holt alle Elemente in dem angegebenen Layer Set.
- PIN [pn] holt alle PINS, bzw. nur die PINS mit dem angegebenen Pinnamen. Wird zusätzlich noch -KEEPPIN angegeben, bleiben die Pinnamen wenn möglich erhalten. Ein Erhalten der Pinnamen ist nur dann möglich, wenn keine Konflikte mit schon vorhandenen Pinnamen auftreten. Ohne -KEEPPIN oder bei Pinnamenskonflikten erscheinen die neuen Elemente mit der Funktion WIRE.
- KEEPPIN Falls Pinelemente geholt werden, bleiben deren Pinnamen soweit möglich erhalten. (siehe -PIN)
- BB holt anstelle der spezifizierten Komponenten deren umschließende Rechtecke.
- RECURSIVE holt die Elemente im spezifizierten Layer rekursiv aus allen Hierarchiestufen in und unter den selektierten Instanzen. Ohne diese Option werden nur die Elemente direkt aus den selektierten Instanzen geholt.

WEITERES

BOOLOP(paris)

NAME

FIND - suchen von Komponenten

SYNOPSIS

FIND -NODE | -PLNAME | -CELLNAME | -GENERATOR | -CELLFUNC | -PLCHAR |
-CELLPROPERTY | -GEOTYP | -GEOFUNC | -TEXT | -LAYER | --BAG | NET |
-PROPERTY | -INVALID_GEO | -OFFGRID -ANGLE_ERROR | -POINTLIM | -PIN |
INCOMPLETE | -ISOLATED

BESCHREIBUNG

sucht die spezifizierten Komponenten und stellt sie selektiert dar. Bei allen Operationen, bei denen ein Name angegeben wird, kann auch ein Suchmuster wie für Dateinamen in der Unix Shell angegeben werden, z.B. R2* findet alle Namen, die mit R2 anfangen, dem dann beliebige andere Zeichen folgen.

OPTIONEN

- NODE *i* selektiert die Geoelemente des Knotens *i*, wobei für *i* nur der Betrag angegeben wird; anschliessend können mit dem Kommando INSPECT -SELNODE alle Komponenten, einschließlich angeschlossener Zellen und betroffener Pins, aufgelistet werden
- PLNAME *c9* selektiert alle Zellaufrufe mit dem Platzierungsnamen *c9*; hier kann auch der Vorname von zusammenzufassenden Zellen angegeben werden, z.B. R27. Es werden dann alle Zellen selektiert, die mit R27. beginnen.
- CELLNAME *cn* selektiert alle Zellaufrufe mit dem Zellnamen *cn*
- GENERATOR *cn* selektiert alle generischen Instanzen, die mit dem Generator *cn* erzeugt wurden.
- CELLFUNC *mf* selektiert alle Zellaufrufe mit der Macrofunktion *mf*
- PLCHAR *c* selektiert alle Zellaufrufe mit dem Platzierungsmerkmal *c*.
- CELLPROPERTY *propname* [*propvalue*] selektiert alle Zellaufrufe, deren Zellen ein Property *propname* mit dem Wert *propvalue* haben.
- PROPERTY *propname* [*propvalue*] selektiert alle Komponenten, die ein Property *propname* mit dem Wert *propvalue* haben.
- GEOTYP *gt* [*l*] selektiert alle Geoelemente in der Ebene *l* vom Typ *gt*
DEFAULT: es werden alle Geoelemente vom Typ *g'* in allen Ebenen selektiert
- GEOFUNC *gf* [*sw*] [*l*] selektiert alle Geoelemente in der Ebene '*l*' mit der Funktion '*gf*'.
Im Fall von Funktion PIN wird mit *sw* der Pinname spezifiziert (DEFAULT: alle Pins).
DEFAULT: es werden alle Geoelemente mit der Funktion *g'* in allen Ebenen selektiert
- LAYER *l* selektiert Geoelemente und Texte in den Ebenen *l*
- TEXT *st* selektiert den Text *st*
- BAG *st* selektiert alle Bag mit dem Namen *st*
- NET *st* selektiert das Schematic-Netz mit dem Namen *st*

- INCOMPLETE selektiert alle Netze, die noch nicht vollständig verdrahtet sind.
- ISOLATED selektiert alle Geoelemente und Pins, die nicht an ein Netz angeschlossen sind.
- POINTLIM i selektiert alle Pfade und Polygone, die mehr als i Eckpunkte haben.
- ANGLE_ERROR selektiert alle Pfade und Polygone mit spitzen Winkeln.
- INVALID_GEO selektiert alle selbstüberlappenden Elemente.
- OFFGRID selektiert alle Elemente, deren Koordinaten nicht im Normgrid des aktuellen Fensters liegen.
- PIN pn ... selektiert die Pins mit den angegebenen Namen aller selektierten Instanzen. Die Instanzen sind danach unselektiert.

WEITERES

SELECT(Paris), UNSELECT(Paris)

NAME

GDSIN - Laden einer Macrogruppe im GDS2 Format

SYNOPSIS

GDSIN -CNV [-NOTEXT] [-RESOLUTION in] [-APPEND gn] fn ...

BESCHREIBUNG

Mit dem GDSIN Kommando GDSII Dateien importiert. Das Kommando verhält sich wie das LOAD Kommando für PARIS Dateien. Zwischen dem PARIS-Format und dem GDS2-Format erfolgt eine Layer Konversion, da das GDS2-Format nur Layer von 0-63 erlaubt. Jeder GDS2-Layer kann zusätzlich in 0-63 Datatypes unterteilt sein. (Segmentierung) Der Import von Properties aus der GDSII Datei findet ebenfalls unter Berücksichtigung des im Konvertierfile definierten Property Mappings statt.

ARGUMENTE

name [...] (fn)

Name(n) der zu ladenden GDSII Dateien. Für jede geladene Datei wird eine neue Macrogruppe mit dem Namen der Datei erzeugt.

OPTIONEN

-CNV filename (fn)

Die Datei `filename` enthält die Zuordnung PARIS-Layer - GDS-Layer/Datatype, PARIS-Property, GDS-Property, die PARIS Textgröße und die PARIS Resolution. Die Syntax ist in der Manualseite von GDSOUT beschrieben.

ACHTUNG: eine Konversions-Tabelle muß immer angegeben werden, auch bei einer 1:1 Übersetzung.

-NOTEXT

Es werden keine Texte aus der GDSII Datei übernommen.

DEFAULT: Texte werden geladen.

-RESOLUTION in

Diese Option spezifiziert die Auflösung der Koordinaten in der GDSII Datei. Falls erforderlich, erfolgt eine Skalierung der Koordinaten auf die PARIS interne Auflösung. Die Database Units der GDSII Datei werden bei Angabe dieser Option ignoriert.

DEFAULT: Die sKalierung der Koordinaten erfolgt entsprechend den Angaben der Database Units in der GDSII Datei.

-APPEND gn

erzeugt keine neue Macrogruppe, sondern fügt die zu ladenden Zelldefinitionen zur Gruppe `mg` hinzu.

NAMENSKONVERTIERUNG

In PARIS werden Zellnamen immer in Großbuchstaben abgelegt. Daher werden Zellnamen, die in GDSII Kleinbuchstaben enthalten, in Großbuchstaben umgewandelt. Wenn dadurch Namenskollisionen mit Zellen aus der gerade geladenen GDSII Datei entstehen, werden neue Zellnamen in der Form `$$i` erzeugt, wobei `i` dabei fortlaufend durchnummeriert wird.

Bei Namenskollision von Zellen wird eine bereits vorhandene Zelle nicht überschrieben, sondern die neue Zelle aus der GDSII Datei ignoriert.

WEITERES

`gdstoprs(1)`, `gdsout(PARIS)`, `load(PARIS)`

NAME

GDSOUT - Ausgabe einer Zelle/Macrogruppe im GDS2 Format

SYNOPSIS

```
GDSOUT -CNV [-GROUP] [-LAYER] [-RESOLUTION] [-BLOCKSIZE] [-NOTEXT]
[-NOTECH] name [-OUT] ...
```

BESCHREIBUNG

Mit dem GDSOUT Kommando werden einzelne Zellen oder komplette Macrogruppen in das GDS2-Format übersetzt und als Datei abgelegt. Zwischen dem PARIS-Format und dem GDS2-Format erfolgt eine Layer Konversion, da das GDS2-Format nur Layer von 0-63 erlaubt. Jeder GDS2-Layer kann zusätzlich in 0-63 Datatypes unterteilt sein. (Segmentierung)

Die PARIS Textkomponente wird als Textelement ins GDS2-Format übertragen, Platzierungsnamen, Pinnamen werden als Textelemente ausgegeben. Pinnamen erscheinen dabei im GDS2-Layer 60, Platzierungsnamen im Layer 61 und Zellnamen im GDS2-Layer 63. Alle diese Texte erhalten den Texttype 63. Die Ausgabe dieser Attribute erfolgt nur, wenn diese sichtbar geschaltet sind (siehe VIEW).

Die Reihenfolge der Argumente und Optionen muß unbedingt eingehalten werden.

ARGUMENTE

name [...] (cn) | (mg)

Name(n) der Zelle(n) bzw. bei Anwendung von -GROUP der Macrogruppe(n), die ins GDS2-Format übersetzt werden sollen. Die Ausgabe erfolgt auf den Dateinamen <name>_lay.gds. Wird mit -OUTPUT filename ein Dateiname angegeben, erfolgt die Ausgabe auf diesen Dateinamen.

OPTIONEN

-CNV filename (fn)

Die Datei filename enthält die Zuordnung PARIS-Layer - GDS-Layer/Datatype, PARIS-Property, GDS-Property, die PARIS Textgröße und die PARIS Resolution. Die Syntax des Konvertierungsfiles ist wie folgt:

```
gds_conversion: conversion_specs

conversion_specs: conversion_spec
                  | conversion_specs conversion_spec

conversion_spec: resolution
                 | text_size
                 | layer_conversion_table
                 | property_conversion_table

resolution: "RESOLUTION" INT

text_size: "TEXT_SIZE" INT

layer_conversion_table: "LAYER" { layer_conversions }

layer_conversions: layer_conversion
                  | layer_conversions layer_conversion

layer_conversion: INT INT INT
```

```

property_conversion_table: "PROPERTIES" { property_conversions }

property_conversions: property_conversion
                    | property_conversions property_conversion

property_conversion: INT STRING

```

Kommentare können an beliebiger Stelle stehen. Als Kommentare wird alles von ! bzw. # bis zum Zeilenende interpretiert. Beispiel:

```

RESOLUTION 100
TEXT_SIZE 500
LAYER
{
#  PARISlayer GDSlayer GDSdatatype

    1  1  0    !BURIED LAYER
    2  1  0    !BURIED LAYER
    3  1  0    !BURIED LAYER
   17  2  0    !SEPARATION
   18  2  0    !SEPARATION
   33  3  0
   34  3  0
}

PROPERTIES
{
#  PARISpropertyName, GDSpropertyName
  INST_NAME      3
  PIN_NAME       2
}

```

ACHTUNG: eine Konversions-Tabelle muss auch bei einer 1:1 Übersetzung definiert werden

-GROUP

Gibt an, dass es sich beim Argument `name` um eine Macrogruppe handelt.
 DEFAULT: es werden die mit `name` bezeichneten Zellen übersetzt

-LAYER gl

Gibt die auszugebenden Ebenen an. Wird mit '1 eine Gruppe von Ebenen angegeben, so ist darauf zu achten, dass jede Ebene dieser Gruppe in der Konversions-Tabelle angegeben werden muß.
 DEFAULT: alle im Paramter-File definierten Ebenen

-RESOLUTION i

Die Auflösung im GDS2 File ist ohne die Angabe einer Resolution die gleiche wie in der PARIS internen Datenbasis. Wird eine bestimmte Auflösung gewünscht, kann diese mit der Option `-RESOLUTION` angegeben werden. Sollen z.B die Daten im GDS2 File in 1/1000u Auflösung erscheinen, muß `-RESOLUTION 1000` angegeben werden.

-BLOCKSIZE i

Die GDSII Datei wird immer bis zu einem ganzzahligen Vielfachen der angegebenen Blockgröße mit Nullbytes aufgefüllt, wenn die tatsächliche Größe kleiner ist als ein Vielfaches der Blockgröße.

DEFAULT: 2048

-NOTEXT

Es werden keine Texte in die GDSII Datei ausgegeben. Dies betrifft Textelemente, Pinnamen, Instanznamen und Zellnamen. -NOTEXT hat Vorrang vor dem Schalter zur Darstellung von Pin-,Instanz- und Zellnamen. Auch wenn die Darstellung von Pin-,Instanz- und Zellnamen eingeschaltet ist, werden diese nicht in die GDSII Datei geschrieben.

DEFAULT: Texte werden ausgegeben.

-NOTECH

Zellen und Instanzen von Zellen, die die Zellfunktion TECH haben, werden nicht in die GDSII Datei ausgegeben. Diese Option dient zum Ausblenden bestimmter Zellen, für die kein Design Rule Check gewünscht wird.

DEFAULT: Alle Zellen werden ausgegeben.

-OUT fn

Gibt an, daß die Ausgabe der mit name spezifizierten Zelle oder Macrogruppe auf fn erfolgen soll

DEFAULT: es wird auf den Pfadnamen name_lay.gds ausgegeben

NAMENSKONVERTIERUNG

Die maximale Länge von Zellnamen in GDS2 ist 32 Zeichen. In PARIS beträgt diese Länge 256 Zeichen. Zellnamen mit mehr als 32 Zeichen müssen daher in kürzere umgewandelt werden. Dies erfolgt durch Vergabe eines Namens der Form __xxxx__. xxxx steht dabei für eine Nummer, die vom System vergeben wird. Die Vergabe eines solchen neuen Namens wird im Protokoll angezeigt.

UNDEFINIERTER REFERENZEN

Wenn eine Zelle exportiert wird, also der -GROUP nicht angegeben ist, erfolgt nur dann eine Ausgabe, wenn die Zelle vollständig gelinkt ist, also keine undefinierten Referenzen auf andere Zellen enthält. Dies gilt auch, wenn undefinierte Referenzen in tieferen Hierarchiestufen der Zelle enthalten sind. In diesem Fall wird ein INSPECT -UNDEFINED durchgeführt und die fehlenden Zellen angezeigt.

WEITERES

gdstoprs(1), view(PARIS), inspect(PARIS)

NAME

GETDRCERR - einlesen von DRC Fehlerflags

SYNOPSIS

GETDRCERR [-ERRFILE] [-ERRLAYER] [-RULE]

BESCHREIBUNG

Liest die Fehlerflags aus einer von WATSON, ICVERIFY oder CHECKMATE erzeugten Fehlerflag-Datei.

Bei WATSON Fehlerflag-Dateien können über die -ERRLAYER Option alle Fehlerebenen getrennt angesprochen werden.

Bei ICVERIFY- bzw. CHECKMATE-Dateien können über die -RULE Option Fehler zu einer bestimmten Regel ausgewählt werden. Fehlerkomponenten aus ICVERIFY- bzw. CHECKMATE-Dateien erhalten ein Property mit dem Namen der verletzten Regel.

Die eingelesenen Komponenten werden in die ERROR-Ebene aufgenommen und selektiert dargestellt. Ein Protokoll der DRC Fehler wird mit Angabe des Datums und des Benutzers, der/die den DRC durchgeführt hat, an die Datei <cellname>_lay.drc angehängt.

OPTIONEN

-ERRLAYER l,[...]

Fehlerebenen, die eingelesen werden sollen. Sofern vorhanden, wird der zugehörige FLAGTEXT auch ausgegeben
(nur bei WATSON-Dateien)
DEFAULT: alle Ebenen

-RULE st

Name der Regel, zu der die Fehlerkomponenten eingelesen werden sollen.
(nur bei ICVERIFY- und CHECKMATE-Dateien)
DEFAULT: alle Regeln

-ERRFILE fn

Name der Fehlerdatei
DEFAULT: 'cellname'_lay.err

NAME

GRID – Ein-/Aus-/Umschalten des Rasters

SYNOPSIS

GRID -NORM | -PATH | -OFF

BESCHREIBUNG

Kontrolliert die Darstellung des Rasters und damit auch die zulässige Koordinateneingabe und das Einrastgitter. Bei folgenden Kommandos bezieht sich die Koordinateneingabe auf das eingestellte Raster: ADD, APPEND, ADDCELL, ADDTEXT, COPY, MOVE, DFFROMSELECT(Nullpunkt), DIST, SYMOP, ORIGIN.

AUSNAHMEN: Bei Designs vom Typ GA und TA rasten Pfade bei ADD und APPEND grundsätzlich ins PATHGRID. Bei MOVE -DIST rasten die Komponenten ins Minimalgrid.

OPTIONEN

-NORM	Einschalten des Norm-Grids Grundraster * Gridfaktor)
-PATH	Einschalten des Pfadrasters
-OFF	Grid ausschalten, d.h. alle Koordinaten schnappen ins Grundraster

WEITERES

SETPAR(Paris)

NAME

HARDCOPY – Erstellen einer Hardcopy

SYNOPSIS

HARDCOPY [-COLOR]

BESCHREIBUNG

Erstellt eine Hardcopy des aktuellen Bildausschnitts und gibt sie auf dem Laserdrucker aus. Es wird eine X-Window Bitmap/Pixmap des Grafikfensters erstellt. Die Bitmap wird über eine Pipeline an das Kommando in der Shellvariablen `HARDCOPY` weitergeleitet. Diese kann dann mit den Standard X-Window Kommandos `xpr`, . . . weiterverarbeitet werden.

OPTIONEN

`-COLOR` Es wird anstelle einer Bitmap eine Pixmap des Grafikfensters erstellt. Die Pixmap enthält die volle Farb-information. Damit kann auch eine Farbhardcopy erzeugt werden.

`DEFAULT:` Es wird eine schwarz-weiß Hardcopy erzeugt.

WEITERES

`PLOT(Paris)`, `xpr(1)`, `xwd(1)`, `lp(1)`, `lpr(1)`, `popen(3)`

NAME

IDENTIFY – Identifizieren eines Zellaufrufs

SYNOPSIS

IDENTIFY [-POINT | -RECT

BESCHREIBUNG

Markiert den Zellaufruf im spezifizierten Fenster als *identified*; identifizierte Zellen koennen beim EDITINPLACE angesprochen werden

Optionen

-POINT p das Fenster wird durch den letzten MARK Punkt und 'p' definiert

-RECT p1 p2 das Fenster wird durch 'p1' und 'p2' definiert

WEITERES

SELECT(paris), INSPECT(paris)

NAME

INCLUDE – einfügen von Komponenten aus Datei oder Puffer

SYNOPSIS

INCLUDE [-INTO] [-RESOLUTION] [-NOTEXT] [-SEL] [buffer | -FILE | -]

BESCHREIBUNG

Liest die Komponenten aus Pastebuffer, Datei oder Standardeingabe und addiert die Komponenten in die entsprechende Ebene der spezifizierten Zelldefinition. Mit Hilfe von **WRITE/INCLUDE** können z.B. Komponenten aus einer anderen Hierarchiestufe in eine bestimmte Zelldefinition kopiert oder transferiert werden.

ARGUMENTE

buffer (c256) Puffer, aus der die Komponenten zu lesen sind. Es können beliebig viele Puffernamen existieren und angegeben werden. Es gibt aber nur einen Standardpuffer. Die Puffer sind so lange aktiv wie das Programm **PARIS** aktiv ist. Ein Puffer kann mit Hilfe von **WRITE** erzeugt werden.
DEFAULT: namenloser Standardpuffer

OPTIONEN

-INTO cn Zelldefinition, in die die Komponenten addiert werden sollen
DEFAULT: aktuelle Zelldefinition

-RESOLUTION i Auflösung der Daten in der einzulesenden Datei
DEFAULT: 100

-NOTEXT evtl. vorhandene Textkomponenten werden nicht eingelesen
DEFAULT: Text wird eingelesen

-SEL die eingelesenen Komponenten sind nach dem Addieren selektiert
DEFAULT: Komponenten sind nach dem Addieren unselektiert

-FILE fn liest von den angegebenen Dateinamen ein.
DEFAULT: namenloser Standardpuffer

- liest von der Standardeingabe ein.
DEFAULT: namenloser Standardpuffer

WEITERES

WRITE(paris)

NAME

INSPECT – Auflisten verschiedener Parameter

SYNOPSIS

INSPECT -SELECT | -SELNODE | -SELECTION | -INTNET | -NET | -REQUI | -CEQUI |
-CELLDEF | -GEOCOUNT | -HIERARCHY | -LOADDEF | -PARAMETER |
-UNDEFINED | -LAYER | -VISLAYER | -DETLAYER | -GENERATORS | EDIT-
STACK [-FULL] [-OUTPUT]

BESCHREIBUNG

Gibt Informationen über die aktuelle Zelldefinition, den aktuellen Parametersatz oder selektierte Komponenten aus. Die Ausgabe erscheint entweder auf dem Bildschirm oder auch zusätzlich in einer Datei.

OPTIONEN

- SELECT gibt Information über die selektierten Komponenten
- SELNODE listet alle Elemente des aktuell selektierten Knotens
- SELECTION listet die aktuelle Netzliste der selektierten Zellaufrufe
- INTNET [cn] listet die interne Netzliste der spezifizierten Zelle
 DEFAULT: aktuelle Zelldefinition
- NET [cn] listet von der spezifizierten Zelldefinition die Schematic-Netze
 DEFAULT: aktuelle Zelldefinition
- REQUI [cn] listet von der spezifizierten Zelldefinition die zusammengefaßten Wider-
 staende und deren Werte
 DEFAULT: aktuelle Zelldefinition
- CEQUI [cn] listet von der spezifizierten Zelldefinition die zusammengefaßten Kapazi-
 taeten und deren Werte
 DEFAULT: aktuelle Zelldefinition
- CELLDEF [cn] gibt Informationen über die spezifizierte Zelldefinition aus
 DEFAULT: aktuelle Zelldefinition
- GEOCOUNT [cn] listet die Anzahl der Geoelemente pro Ebene
 DEFAULT: aktuelle Zelldefinition
- HIERARCHY [cn] listet alle plazierten Zellen und deren Schachtelungstiefe in der spezifizierten
 Zelldefinition
 DEFAULT: aktuelle Zelldefinition
- LOADDEF listet alle geladenen Zelldefinitionen
- PARAMETER listet die aktuell gesetzten PARIS Parameter
- UNDEFINED listet alle unreferenzierten Instanzen
- LAYER listet alle Ebenen und deren aktuell gesetzten Parameterwerte
- VISLAYER listet alle sichtbaren Ebenen und deren aktuell gesetzten Parameterwerte
- DETLAYER listet alle editierbaren Ebenen und deren aktuell gesetzten Parameterwerte
- GENERATORS listet die Namen und die Kurzbeschreibung aller registrierten Generatoren.
- EDITSTACK zeigt den Edit Stack des aktuellen Fensters an.
- FULL erstellt ein ausführliches Listing
 (nur mit -SELECT und -SELNODE erlaubt)

DEFAULT: kurzes Listing
-OUTPUT fn schreibt die erwünschte Information in die Datei mit dem Namen fn.
DEFAULT: Informationen werden am Bildschirm ausgegeben

WEITERES

SELECT(Paris), LIST(Paris), REGISTER(Paris)

NAME

LIST - Auflisten von Zellnamen

SYNOPSIS

LIST -CELLS [gn ...]|-USED [cn ...]|-INSTANCES [cn ...]|-USER [cn ...]

BESCHREIBUNG

Schreibt die Namen der Zellen in den angegebenen Gruppen, (-CELLS) Modus, die Namen der benutzten Zellen, (-USED) Modus, eine Tabelle der Instances bestehend aus Instance Name und Zellname (-INSTANCES) Modus, oder die Namen der Nutzer, (-USER) Modus, in den angegebenen Gruppen bzw. Zellen in die Standardausgabe. Die ausgegebenen Namen können dann mittels einer Kommandosubstitution als Argumente für andere Kommandos verwendet werden. Beispiel: Netext für alle Zellen in der Gruppe `my_group`:

```
netext ('list my_group') -r
```

oder Anwendung einer Kommandoprozedur für alle Zellen:

```
for c in ('list')
do
    my_procedure $c
done
```

ARGUMENTE

`gn` Name einer Macrogruppe.
`cn` Name einer Zelle

WEITERES

INSPECT(Paris)

NAME

LOAD - Laden von Macrogruppen

SYNOPSIS

LOAD [-APPEND] [-REPLACE][[-MERGE] [pathname ...]

BESCHREIBUNG

Lädt die in den Datei(en) *pathname* (default extension ist Macrogruppe *pathname*).

ARGUMENTE

[*pathname ...*] (fn) Datei(en), aus der die Zelldefinitionen zu laden sind; die erzeugte Macrogruppe erhält den gleichen Namen

OPTIONEN

-APPEND *mg* erzeugt keine neue Macrogruppe, sondern fügt die zu ladenden Zelldefinitionen zur Gruppe *mg* hinzu

-REPLACE überschreibt vorhandene Zelldefinitionen mit gleichem Namen.

-MERGE überschreibt vorhandene Zelldefinitionen mit gleichem Namen nur, wenn die Zelle in der zu ladenden Datei jünger ist als bereits geladene Zelle. Alle anderen Zellen bleiben unverändert.

DEFAULT: bei Zelldefinitionen mit gleichem Namen bleibt die vorhandene Version erhalten; die neue Version aus der zu ladenden Datei wird ignoriert

Die Optionen -MERGE und -REPLACE können auch ohne Pfadname angegeben werden. In diesem Fall werden alle geladenen Macrogruppen bearbeitet. Wenn weder -MERGE noch -REPLACE angegeben ist, muß mindestens ein Pfadname angegeben sein. Editierte Zellen werden grundsätzlich nicht überschrieben.

WEITERES

SAVE(*paris*)

NAME

LOADARRAY – Einlesen eines Gate Array Untergrundes

SYNOPSIS

LOADARRAY pathname

BESCHREIBUNG

Laedt den in 'pathname' enthaltenen Gate Array Untergrund; dieses ist nur moeglich, wenn im vorher geladenen Parameter-File als Designtyp GA spezifiziert wurde. Das **LOADARRAY** Kommando muss vor dem ersten **LOAD** Kommando kommen

ARGUMENTE

pathname (fn) Datei, die den Gate Array Untergrund enthaelt

WEITERES

LOAD(paris)

NAME

LOADPAR – Laden der Design- und Technologie Parameter

SYNOPSIS

LOADPAR pathname

BESCHREIBUNG

Mit **LOADPAR** werden die Design- und Technologie Parameter geladen. Die Design-Parameter müssen geladen sein, bevor ein Editor Befehl aufgerufen werden kann. **LOADPAR** kann nur einmal während einer Session aufgerufen werden. Normalerweise ist **LOADPAR** die erste Anweisung einer **PARIS** Session.

ARGUMENTE

pathname (fn) Datei welche die Design-Parameter enhaelt

WEITERES

SAVEPAR(paris)

NAME

MAGNIFY – Ändern der Skalierung der Instanztransformation

SYNOPSIS

MAGNIFY [-X fl] | [-Y fl] | fl fl

BESCHREIBUNG

Ändern die Skalierung der Instanztransformation für alle selektierten Instanzen. Die Skalierung kann für x- und y-Richtung getrennt angegeben werden. Es ist damit möglich eine beliebig verzerrte Skalierung einer Instanz zu erzeugen. Ein negativer Skalierungsfaktor erzeugt eine Spiegelung an der jeweiligen Achse. **MAGNIFY** ist nur anwendbar, wenn für die aufgerufene Zelle die Symmetrieoperation 8 (beliebige Transformation) zugelassen ist. Wenn nicht, wird die Änderung verweigert.

ARGUMENT

fl fl Faktoren, mit denen die Instanztransformation in x- bzw. y-Richtung skaliert wird.

OPTIONEN

-X fl Skalierung in x-Richtung
-Y fl Skalierung in y-Richtung

WEITERES

MIRROR(PARIS), ROTATE(PARIS), SYMOP(PARIS), SCALE(PARIS)

NAME

MARK – Markieren eines Punktes

SYNOPSIS

MARK point [-POSITION | -LINE | -RECT] | [-CLEAR]

BESCHREIBUNG

Markiert genau einen Punkt, der vom nachfolgenden Kommando verwendet werden kann. Der Eingabecursor wird je nach Option auf eine Eingabelinie (Gummiband) oder ein Eingaberechteck mit dem angegebenen Punkt als Bezugspunkt umgestellt. Die MARK Punkte werden intern in einem Stack verwaltet. Es ist also möglich, geschachtelte Marks zu verwenden. Von den Kommandos SELECT, UNSELECT, COPY, STEP, MOVE, CUT, ZOOM wird dann jeweils der oberste Punkt vom Mark Stack geholt und als erster Punkt eines Vektors oder Fensters verwendet.

ARGUMENTE

point (p) Markierungspunkt

OPTIONEN

-POSITION der Cursor ist nach der Eingabe des Punktes frei.
-LINE der Cursor ist nach der Eingabe des Punktes durch eine Linie (Gummiband-Cursor) mit dem markierten Punkt verbunden
-RECT der Cursor ist nach der Eingabe des Punktes durch ein Rechteck mit dem markierten Punkt verbunden
-CLEAR Der gesamte Mark Stack wird gelöscht.

WEITERES

SELECT(paris), UNSELECT(paris), COPY(paris), MOVE(paris), CUT(paris), DIST(paris)

NAME

MIRROR – Spiegeln von Instanzen

SYNOPSIS

MIRROR -FLIPX | -FLIPY

BESCHREIBUNG

führt die spezifizierte Spiegelung für alle selektierten Instanzen durch. Die Instanz verändert dabei Ihre Position nicht.

OPTIONEN

-FLIPX Spiegelung um die x-Achse

-FLIPY Spiegelung um die y-Achse

WEITERES

SYMOP(PARIS), ROTATE(PARIS)

NAME

MKGEO – erzeugt ein GeoElement

SYNOPSIS

MKGEO -BBI-ABB -LAYER [-FUNCTION] [-DELETE] | -EPIN [-LAYER]

BESCHREIBUNG

Erzeugt ein oder mehrere GeoElement(e) anhand der selektierten Komponenten. Das neu erzeugte Element erhält den Layer l und die Geofunktion gf. Falls keine Geofunktion angegeben ist, erhält das Element die Funktion WIRE. Die selektierten Komponenten sind nach der Ausführung des Kommandos deselektiert, das neue Element erscheint selektiert.

OPTIONEN

- ABB erzeugt die einzelnen umschließenden Rechtecke der selektierten Komponenten.
- BB erzeugt das gesamte umschließende Rechteck aller selektierten Komponenten.
- EPIN pn erzeugt ein Polygon mit einer Breite von 1 Einheit, das mit einer Kante auf der selektierten Kante liegt. Für diese Funktion darf nur genau eine Kante eines Polygons selektiert sein. Das neue Poylgon erhält die Geofunktion PIN und den Pinnamen pn.
- LAYER l Die neuen Elemente erhalten den Layer l.
- FUNCTION gf Die neuen Elemente erhalten die Geofunktion gf. Diese Angabe ist optional. Die default Geofunktion ist WIRE.
- DELETE die selektierten Elemente werden gelöscht.
DEFAULT: die selektierten Elemente werden nicht gelöscht

WEITERES

SELECT(paris)

NAME

MOVE - Verschieben von selektierten Komponenten

SYNOPSIS

MOVE -POINT | -VECTOR | -DIST | -PATHGRID | -NORMGRID

BESCHREIBUNG

verschiebt die selektierten Komponenten um den angegebenen Betrag

Optionen

-POINT p der Verschiebungsvektor setzt sich aus dem letzten MARK Punkt und 'p' zusammen

-VECTOR p1 p2 der Verschiebungsvektor setzt sich aus 'p1' und 'p2' zusammen

-DIST in1 in2 verschiebt um in1 Grundeinheiten in x-Richtung und in2 Grundeinheiten in y-Richtung. Rastet ins Minimalgrid.

-PATHGRID i1 i2 verschiebt um (i1*pathgrid_x) Grundeinheiten in x-Richtung und (i2*pathgrid_y) Grundeinheiten in y-Richtung

-NORMGRID i1 i2 verschiebt um (i1*normgrid_x) Grundeinheiten in x-Richtung und (i2*normgrid_y) Grundeinheiten in y-Richtung

WEITERES

MARK(Paris), COPY(Paris), STEP(Paris)

NAME

MSG - Ausgabe von Argumenten

SYNOPSIS

MSG argument

BESCHREIBUNG

Gibt Argumente als Meldung auf dem Bildschirm aus. (Entspricht dem 'echo'-Kommando der Bourne-Shell)

ARGUMENTE

argument (a) 'a' kann sowohl Text als auch der Wert einer Variablen innerhalb der A-Shell sein.

BEISPIEL:

msg abc --> abc
msg \$PLTLAYER --> BL, SEP, P+

NAME

NETEXT - Durchführung der Schaltungsrückgewinnung

SYNOPSIS

NETEXT [cellname] [-RECURSIV] [-FORCE]

BESCHREIBUNG

führt die Schaltungsrückgewinnung durch und weist dabei geeignete Knotennummern zu.
Voraussetzung fuer NETEXT ist ein erfolgreicher CHECK.

ARGUMENTE

cellname [...] (cn) Name der zu bearbeitenden Zelle(n)
DEFAULT: aktuelle Zelldefinition

OPTIONEN

-RECURSIV falls in der zu prüfenden Zelldefinition Zellaufufe vorhanden sind, deren Definitionen `net_flag=false` haben, werden zunächst diese Zelldefinitionen geprüft (rekursive Behandlung der kompletten Zelldefinition und deren Zellaufufe, bis hin zu den Basiszellen)
DEFAULT: oberste Hierarchiestufe der zu bearbeitenden Zelldefinition

-FORCE prüft in jedem Fall die angegebenen Zelldefinitionen, auch dann, wenn diese bereits `net_flag=true` haben.

WEITERES

CHECK(Paris)

NAME

NETLIST – Installieren/Deinstallieren von Netzlisten in einem Layout

SYNOPSIS

NETLIST [-INSTALL [-REPLACE|-AUGMENT|-OVERRIDE] fn] [-UNINSTALL
[-RECURSIVE]]

BESCHREIBUNG

Überlagert oder entfernt die Netzinformation der aktuell editierten Layoutzelle.

ARGUMENTE

`fn` Die Netzliste wird von dem angegebenen Dateinamen `fn` gelesen. Als Netzlistenformat wird derzeit das TEMIC Format verarbeitet.

OPTIONEN

`-INSTALL` Die Netzliste wird eingelesen. Über den folgende Schalter wird eingestellt, ob eine evtl bereits vorhandene Netzinformation überschreiben oder erweitert wird. Durch das Installieren der Netzliste entstehen in der Zelle zusätzliche Netz-Objekte. Netze sind eine spezielle Form von Gruppierungen von Objekten, die in der Regel Pins und Geometrie-Elemente enthalten. Nach dem Installieren der Netzliste wird ein `CHECK` und `NETEXT` durchgeführt. Das `snet`-Flag der Zelle wird auf `true` gesetzt.

In einer Zelle mit gültiger Netzinformation, was an `snetFlag == true` erkannt wird, werden fehlende elektrische Verbindungen durch Gummibandlinien angezeigt. Die Gummibandlinien stellen immer die kürzeste Verbindung zweier noch nicht verdrahteter Teilnetze eines Netzes dar. Ein Netz ist über seine Pins, Geometrie oder Gummibandlinien selektierbar. Ein vollständig verdrahtetes Netz enthält keine Gummibandlinien. Wenn die Online Verifikation eingeschaltet ist, werden die Gummibandlinien mit dem Editiervorgang an den aktuelle Zustand angepaßt.

`-REPLACE` Die evtl. in der Zelle vorhandene Netzinformation wird vollständig durch die neu Eingelesene ersetzt. Alle vorher vorhandenen Netze werden gelöscht.

`-OVERRIDE` Die Netze aus der eingelesenen Netzliste werden mit den Netzen in der Layoutzelle verschmolzen. Nicht vorhandene Netze werden neu erzeugt. Netze aus der Netzliste überschreiben dabei evtl. vorhandene Netze im Layout. Netze, die im Layout vorhanden sind, aber nicht in der Netzliste, bleiben unverändert.

`-AUGMENT` Die Netze aus der eingelesenen Netzliste werden mit den Netzen in der Layoutzelle verschmolzen. Nicht vorhandene Netze werden neu erzeugt. Vorhandene Netze im Layout werden nicht durch Netze aus der Netzliste überschrieben. Netze, die im Layout vorhanden sind, aber nicht in der Netzliste, bleiben unverändert.

`-UNINSTALL` Die Netzinformation aus der Layoutzelle wird entfernt. Das `snet`-Flag der Zelle ist danach `false`.

`-RECURSIVE` Die Netzinformation wird rekursiv auch aus allen in der aktuellen Zelle aufgerufenen Zellen entfernt. Dieser Schalter wirkt nur in Verbindung mit `-UNINSTALL`.

WEITERES
CREATELIST(Paris)

NAME

NOTCH - Einfügen einer Ecke/Kante in ein Geoelement

SYNOPSIS

NOTCH [-SMOOTH]

BESCHREIBUNG

Erweitert ein GeoElement (Pfad oder Polygon) um eine zusätzliche Ecke bzw Kante. Ein GEOELEMENT kann wie folgt erweitert werden:

- ein Pfad an einer Ecke durch Selektion eines Pfadpunktes. Der selektierte Punkt darf nicht der Anfangs- oder End- punkt des Pfades sein.
- ein Polygon an einer Ecke. Hier ist jeder Punkt des Polygons erlaubt.

Nach Aufruf des NOTCH Kommandos wird die aktuelle Cursorposition als Eingabe des neuen Eckpunktes interpretiert. Das durch den neuen Punkt erweiterte Element wird dabei als Phantomelement angezeigt. Der Abschluß der Eingabe erfolgt mit dem Kommando `DF p`. Der Punkt `p` wird dabei als neuer Eckpunkt des selektierten Elements eingefügt. Bei Designs vom Typ `GA` und `TA` rasten Pfade immer ins Pfadgrid.

OPTIONEN

`-SMOOTH` Fügt anstelle eines neuen Punktes eine 45 Grad Kante in ein GeoElement ein. Das ist nur erlaubt, wenn die beiden Kanten vor und hinter dem selektierten Punkt orthogonal sind, also entweder horizontal oder vertikal verlaufen, und einen rechten Winkel einschließen. Diese Option ist nur erlaubt, wenn die aktuelle Orthogonalität nicht auf `YES` steht.

FEHLER

`ECOMNOTALLOWED` Es ist keine Zelle editiert

`ENOSELCOMP` Es ist keine Komponente selektiert

`ETOMNYCOMPSEL` Es ist mehr als eine Komponente selektiert

`EWRGCOMPSEL` Die selektierte Komponente ist kein Pfad oder kein Polygon

`EORTHOCHK` Die aktuelle Orthogonalität steht auf 90 Grad und läßt dadurch das Digitalisieren von mit 90 Grad Winkeln nicht zu.

WEITERES

`APPEND(Paris)`, `DF(Paris)`, `CANCELCOMP(Paris)`

NAME

ONLVERIFY - Ein-/Ausschalten der Online-Prüfung

SYNOPSIS

ONLVERIFY -ON | -OFF

BESCHREIBUNG

schaltet die Online-Überprüfung während der Editieroperationen an bzw. aus

OPTIONEN

-ON schaltet die Online-Schaltungsrückgewinnung ein; falls `check_flag` und/oder `net_flag=false`, wird vorher `NETEXT` bzw. `CHECK` und `NETEXT` ausgeführt

-OFF schaltet die Online-Schaltungsrückgewinnung aus und setzt `check_flag` und `net_flag=false`.

WEITERES

`CHECK(Paris)`, `NETEXT(Paris)`

NAME

ORIGIN – Nullpunkt der editierten Zelle setzen

SYNOPSIS

ORIGIN [-LL | -POINT

BESCHREIBUNG

setzt den Nullpunkt an die spezifizierte Position

OPTIONEN

-LL neue Nullpunktposition ist die linke untere Ecke des umschreibenden Rechtecks

-POINT p neue Nullpunktposition ist p

NAME

PLOT – Erstellen einer Plot Datei

SYNOPSIS

PLOT [cellname] -DEV -LAYER [-CNV] [-SCALE] [-FORMAT] [-ROTATE]
[-NODESCEND] [-NOTEXT] [-OUTPUT] [-SUBTUN]

BESCHREIBUNG

Erstellt eine Plotdatei von der spezifizierten oder aktuellen Zelldefinition. Folgende Plotter bzw. Plotformate werden unterstützt:

- CIF Format
- GDSII Format
- HPGL Format, z.B. fuer Einbindung in Textsysteme wie WORD oder WORD PERFEKT. Dieses Format ist für die Anwendung in Stiftplottern gedacht. Es werden keine gefüllten Polygone erzeugt.
- HPGL2 Format zur Ausgabe auf HP Design Jet Plottern. Dieser Treiber erzeugt gefüllte Polygone mit den selben Mustern und Farben wie am Bildschirm.
- CALC925-Format für CalComp Plotter
- PostScript-Format(PS), z.B. zur Ausgabe auf einem PostScript-Drucker oder zur Einbindung in ein Textsystem. Der PS Treiber gibt die volle Farbinformation aus. Es wird eine EPS Datei erzeugt. Der Postscript Treiber erzeugt einen Rand von ca. 25mm.
- PS-MONO: monochrome Postscript Ausgabe. Alle Farben werden auf schwarz abgebildet. Zur Ausgabe auf einem schwarzweiß Drucker.

Zum Erzeugen der Plotdateien wird intern die gleiche Prozedur durchlaufen wie für den Bildaufbau am Bildschirm. Das hat zur Folge, daß die Plotdateien für CIF und GDSII nicht hierarchisch sondern immer flach sind, obwohl diese Formate an sich eine hierarchische Darstellung unterstützen. Die Auflösung der Hierarchie ist aber erforderlich, da PLOT nicht nur die gesamte Zelle sondern auch den gerade aktuellen Bildausschnitt ausgeben kann. Dabei müssen dann die Elemente, die über den Bildausschnitt hinausragen abgeschnitten werden. Es können dann neue Elemente auch in den unteren Hierarchiestufen entstehen, die nicht in der Originalzelle enthalten sind.

Die voreingestellte Orientierung der Ausgabe ist Querformat (Landscape). Für Postscript Drucker, die standardmäßig im Hochformat (Portrait) drucken, wird das Bild um 90 Grad im Uhrzeigersinn gedreht.

ARGUMENTE

cellname (cn) Name der zu plottenden Zelldefinition
 DEFAULT: der dargestellte Ausschnitt der
 aktuellen Zelldefinition wird geplottet

OPTIONEN

-LAYER gl zu plottende Ebenen
-CNV fn nur fuer GDSII. Enthält der Layer Umsetzung von PARIS Layern auf GDSII Layer/Datatype.

- SCALE i Massstabsangabe (nur fuer HPGL, HPGL2, CAL925 und PS -Format) z.B. 200 bedeutet 200:1. Ist zusätzlich die Option -FORMAT spezifiziert, wird im gewählten Massstab geplottet und beim Überschreiten des eingestellten Formats entsprechend abgeschnitten.
DEFAULT: SCALE 0
- FORMAT din Angabe eines A-Formates (A0...A5, nur fuer HPGL, HPGL2 CALC925, PS und PS-MONO); der Maßstab wird so berechnet, dass die Daten genau das spezifizierte A-Format ausfüllen. Ist zusätzlich die Option -SCALE spezifiziert, wird im gewählten Maßstab geplottet und beim Überschreiten des eingestellten Formats entsprechend abgeschnitten.
DEFAULT: A0
- ROTATE nur fuer HPGL, HPGL2, CALC925, PS, PS-MONO möglich. Falls die y-Richtung wesentlich größer ist als die x-Richtung, also das Bild im Hochformat vorliegt, kann der Plot gedreht werden, um das Papier besser auszunützen
DEFAULT: Plot wird nicht gedreht
- DEVICE dt Angabe des Plottertyps bzw. Plot Formats (CIF, GDS, HPGL, HPGL2, CALC925, PS, PS-MONO)
- NODESCEND Bei Angabe dieser Option werden die Descend Attribute an Instanzen nicht ausgewertet. Es erscheint dann immer die gesamte Geometrie im Plot. Fehlt diese Option wird die Descend Information ausgewertet. Es wird dann je nach Einstellung von bestimmten Instanzen nur ein reduziertes Layerset oder eine symbolische Darstellung ausgegeben.
- NOTEXT Bei Angabe dieser Option werden keine Texte ausgegeben. Diese Option wird für alle Ausgabetypen unterstützt.
- OUTPUT Name der Plot-Datei. Wenn der Name mit einem Pipe Zeichen (|) anfängt, wird nicht eine Datei geschrieben, sondern die Daten an das Kommando nach | weitergegeben. Damit können die Daten direkt an einen System Drucker geschickt werden, ohne daß eine Zwischendatei geschrieben wird. Wird z.B. als Ausgabedatei lp angegeben, wird die Ausgabedatei des PLOT Kommandos direkt in die Eingabe des Kommandos lp weitergegeben und dann automatisch gedruckt. Dieses Verhalten geht nur für die Plottertreiber HPGL, HPGL2, CALC925,PS und PS-MONO.
DEFAULT: abhängig vom gewählten Ausgabeformat
 'cellname'_lay.cif
 'cellname'_lay.gds
 'cellname'_lay.hpgl
 'cellname'_lay.hpgl2
 'cellname'_lay.calc925
 'cellname'_lay.ps
- SUBTUN der Untergrund bei Gate-Arrays wird mit ausgegeben

WEITERES

GDSOUT(PARIS), CIFOUT(PARIS), PRSOUT(PARIS), popen(3)

NAME

PROPERTY – Zuordnen einer Property an Komponenten

SYNOPSIS

PROPERTY [-CELL | -SEL] -ADD | -CHANGE | -REMOVE | -ECHO | -EXISTS | -LIST
[-PROTECTION] [-VISIBILITY] [-POSITION]

BESCHREIBUNG

ordnet den selektierten, den spezifizierten Komponenten bzw. der aktuellen Zelldefinition eine Property - mit einem Namen, einem Wert und Attributen für die Protection, die Sichtbarkeit und die Position - zu und erlaubt eine Änderung oder Löschung einer Property. Alle Properties der ausgewählten Komponenten können angezeigt werden.

OPTIONEN

- CELL cn [...] den spezifizierten Zelldefinitionen wird eine Property zugeordnet.
DEFAULT: aktuelle Zelldefinition
- SEL den selektierten Komponenten wird ein Property zugeordnet.
DEFAULT: aktuelle Zelldefinition
- ADD name v die Zielkomponenten erhalten eine Property 'name' mit dem Wert v und Attributen zugewiesen.
- CHANGE name [v] bei der Property 'name' der Zielkomponenten werden der Wert v und/oder die Attribute geändert.
- REMOVE name die Property 'name' der Zielkomponenten wird gelöscht.
- ECHO name gibt den Wert des Properties name analog des echo Kommandos im Betriebssystem auf die Standardausgabe aus. Diese Option kann benutzt werden, wenn der Wert eines Properties in einem Shellskript verwendet werden soll.
- EXISTS name prüft, ob das Property mit dem Namen name existiert. Wenn ja, ist das Ergebnis des Kommandos true, sonst false.
- LIST die Properties der Zielkomponenten werden aufgelistet.
- PROTECTION REMOVABLE | CHANGEABLE | FIXED
ändert die Protection der Property. Die Protection RESERVED wird nur intern vergeben.
DEFAULT: REMOVABLE bzw. bisheriger Wert
- VISIBILITY OFF | ON
ändert die Sichtbarkeit der Property.
DEFAULT: OFF bzw. bisheriger Wert
- POSITION p ändert die Position der Property.
DEFAULT: 0,0 bzw. bisheriger Wert

WEITERES

ATTACH(PARIS)

NAME

PROTECTION – Schutz von Komponenten gegen Veränderung

SYNOPSIS

PROTECTION [-ALL | -NONE | -ADD | -REMOVE]

BESCHREIBUNG

Jeder Makrokomponente ist eine Schutzmaske zugeordnet. Die Komponente läßt sich dadurch gegen bestimmte Editierfunktionen schützen. Ein Schutz gegen folgende Editierfunktionen ist möglich:

EXCHANGE	Austausch einer Instanz gegen andere Zelle
MOVE	Verschieben von Komponenten
APPLYSYM	Spiegeln/Drehen/Verschieben
DUP	Kopieren
EDIT	Editinplace von Instanzen
DELETE	Löschen
SIZE	Verzerren

Falls eine Komponente für eine der angegebenen Operationen geschützt ist, wird die Ausführung der Operation verweigert. Mit Hilfe des **PROTECTION** Kommandos kann die Schutzmaske von selektierten Elementen modifiziert werden.

OPTIONEN

-ALL	Die Komponenten werden vor allen Editierfunktionen geschützt.
-NONE	Der Schutz wird aufgehoben.
-ADD prot	Der angegebene Schutz wird in die Schutzmaske aufgenommen.
-REMOVE prot	Der angegebene Schutz wird aus der Schutzmaske entfernt.

WEITERES

SYMOP(PARIS), **MIRROR(PARIS)**, **COPY(PARIS)**, **MOVE(PARIS)**, **STEP(PARIS)**,
DELETE(PARIS), **EDITINPLACE(PARIS)**, **SIZE(PARIS)**, **EXCHANGE(PARIS)**

NAME

PRSOUT - Ausgabe von Zellen im PARIS-Format

SYNOPSIS

PRSOUT [-NOGENERICS] [cellname [-OUTPUT]] [...]

BESCHREIBUNG

schreibt für jede spezifizierte Zelle bzw. für die gerade editierte Zelle die Design-Daten einschließlich der kompletten Zellhierarchie im PARIS-Format in eine Datei.

ARGUMENTE

cellname (cn) Name der auszugebenden Zelldefinition
DEFAULT: aktuelle Zelldefinition; In diesem Fall werden nur die Daten der Zelldefinition des aktuellen Fensters ausgegeben.

OPTIONEN

-NOGENERICS Generierte Zellen werden nicht ausgegeben.
DEFAULT: generierte Zellen werden mit ausgegeben.

-OUTPUT fn Die PARIS-Daten der Zelle werden in einer Datei mit dem Namen *fn* bzw. *'fn'.prs* abgelegt.
DEFAULT: *cellname_lay.prs*

WEITERES

SAVE(Paris), LOAD(Paris), GDSOUT(Paris), CIFOUT(Paris)

NAME

QUIT – Abbruch des PARIS Editors

SYNOPSIS

QUIT [-FORCE]

BESCHREIBUNG

Bricht den PARIS Editor ab. Bei QUIT ohne Option wird überprüft ob seit dem letzten Abspeichern noch Änderungen vorgenommen wurden. Ist dies der Fall wird QUIT nicht ausgeführt.

OPTIONEN

-FORCE bricht dem PARIS Editor ohne Vorwarnung ab; kann zu jeder Zeit aufgerufen werden.
DEFAULT: bricht nur ab, wenn vorher abgespeichert wurde

NAME

RCEQUI – Berechnung des resultierenden R/C Wertes

SYNOPSIS

RCEQUI [cellname] [-ALL]

BESCHREIBUNG

Berechnet den resultierenden Widerstands- bzw. Kapazitätswert von Widerstands- bzw. Kapazitätsnetzwerken, d.h. seriell, parallel oder gemischt seriell-parallel verknüpfte Widerstände bzw. Kapazitäten werden zusammengefaßt und deren Gesamtwert berechnet. Die errechneten Werte werden der Zelldefinition angehängt.

Die Netzwerke werden an einem Punkt im Platzierungsnamen erkannt, z.B. R50.1, R50.2, ... R50.n oder C10.1, C10.2 ... C10.n. Der Teil des Names vor dem Punkt erscheint dann als der Instanzname des zusammengefaßten Elements in der Netzliste.

ARGUMENTE

cellname [...] (cn) Namen der zu bearbeitenden Zelldefinitionen; in diesem Fall werden alle Widerstands- bzw. Kapazitätsnetzwerke der spezifizierten Zelldefinitionen berechnet. Ist nur in Verbindung mit der Option ALL erlaubt.
DEFAULT: aktuelle Zelldefinition

OPTIONEN

-ALL behandelt alle Widerstands- bzw. Kapazitätsnetzwerke der aktuellen Zelldefinition
DEFAULT: das durch eine selektierte Widerstands- bzw. Kapazitätzelle spezifizierte Widerstands- bzw. Kapazitätsnetzwerk

WEITERES

CALCR(PARIS), CALCC(PARIS)

NAME

REGENERATE – nachgenerieren einer generischen Instanz mit neuen Parametern

SYNOPSIS

REGENERATE [-DIALOG] [gargs]

BESCHREIBUNG

REGENERATE führt die Neugenerierung einer generischen Instanz mit neuen Generatorparametern durch. Es wird der selbe Generator wie bei der Erzeugung der Instanz verwendet. **REGENERATE** erzeugt keine neue generische Instanz, sondern generiert eine neue Zelle mit den angegebenen Generatorparametern und linkt die bestehende Instanz an die neue Zelle. Die neuen Generatorparameter können in der Generatordialogbox oder in der Kommandozeile angegeben werden. Die bestehenden Generatorparameter der Instanz werden als Voreinstellung verwendet. Wenn zusätzlich neue Parameter in der Kommandozeile angegeben sind, überschreiben diese die vorhandenen Generatorparameter der Instanz.

ARGUMENTE

gargs Die Generatorparameter werden im Argument **gargs** angegeben. Neue Generatorargumente überschreiben vorhandene. Die Angabe von **gargs** ist optional.

OPTIONEN

-DIALOG Wenn **-DIALOG** angegeben wird, wird vor dem Neugenerieren eine Dialogbox zur Eingabe der Generatorparameter angezeigt. Zusätzlich zum Dialog können auch Generatorparameter angegeben werden. Diese werden dann als Voreinstellung im Dialog übernommen. Andernfalls werden in der Dialogbox die vorhandenen Generatorparameter als Voreinstellung angeboten.

WARNUNGEN

`inconsistent pin list`

Diese Meldung erscheint dann, wenn die neue Instanz andere Pins hat als die alte ausgetauschte. **REGENERATE** versucht immer, die Pinliste der Instanz zu erhalten. Dies ist jedoch nur dann möglich, wenn die Portlisten der alten und neuen Zelle übereinstimmen. Andernfalls wird die Pinliste der neuen Instanz an die Portliste der neuen Zelle angepaßt. Dabei gibt **PARIS** dann die Warnung `inconsistent pin list` aus. Diese Meldung ist eine Warnung und dient zur Information des Benutzers. Der Austausch wird trotzdem durchgeführt.

FEHLER

`command not allowed`

Es ist keine Zelle zum editieren offen.

`no selected component`

Es ist nichts selektiert.

`too many components selected`

Es darf nur eine Komponente selektiert sein, da **REGENERATE** nur auf genau eine generische Instanz wirkt.

`wrong components selected`

Die selektierte Komponente ist keine generische Instanz.

`no such generator`

Der Generator für die generische Instanz existiert nicht.

`invalid generator parameter`

Diese Meldung erscheint dann, wenn unbekannte Generatorparameter angegeben wurden oder unzulässige Werte für Generatorparameter.

generate error

Bei der Generierung der neuen Instanz ist ein Fehler aufgetreten.

WEITERES

EXCHANGE(PARIS), ADDCELL(PARIS), ACTIVCELL(PARIS), FIND(PARIS),
SELECT(PARIS)

NAME

REGISTER – registrieren von Generatoren oder des Generatorcaches

SYNOPSIS

REGISTER -GENERATOR pn

REGISTER -GCACHE pn

BESCHREIBUNG

Mit Hilfe des Kommandos **REGISTER** wird eine Menge von Generatoren oder der Pfadname des Generatorcaches in **PARIS** bekanntgemacht.

Bevor Generatoren in **PARIS** benutzt werden können, müssen diese bekannt sein. Dies erfolgt mit der ersten Version des **REGISTER** Kommandos. Alle bekannten Generatoren werden intern im Generatorverzeichnis verwaltet. Zum Eintragen von Generatoren in das interne Generatorverzeichnis ruft **PARIS** den Generator in der Form pn -P auf. Der Generator ist in der Regel ein Verbundgenerator, in dem mehrere Einzelgeneratoren zusammengebunden sein können. Der Generator gibt dann die Deklarationen aller Einzelgeneratoren aus, die in ihm zusammengebunden sind. Eine Generatordeklaration besteht aus dem Typ des erzeugten Elements, dem Namen des Generators und den Generatorparametern. Mit Hilfe dieser Angaben kann **PARIS** dann den Generator aufrufen und die Dialogboxen zur Eingabe der Generatorparameter erzeugen.

Der Generatorcache dient zum Abspeichern von bereits generierten Zellen. Wenn eine neue Zelle generiert wird, schaut **PARIS** erst einmal im Generatorcache nach, ob dort schon das angeforderte Element vorhanden ist. Wenn ja, wird es verwendet, ohne daß der eigentliche Generator aufgerufen werden muß. In einer **PARIS** Sitzung gibt es immer einen internen Generatorcache. Der interne Cache existiert so lange, wie das Programm aktiv ist. Der Generatorcache kann auch extern auf der Platte abgelegt werden. Dort werden dann die bereits generierten Zellen abgespeichert. Wenn dann eine generierte Zelle angefordert wird, schaut **PARIS** erst im internen, dann im externen Cache nach und zuletzt wird der Generator aufgerufen. Der Pfadname des externen Caches wird mit dem Kommando **REGISTER -GCACHE** bekanntgemacht.

Das Kommando **REGISTER** sollte laufen, bevor Generatoren aufgerufen werden oder der Generatorcache benutzt wird. Dies ist in der Regel der Fall, wenn Daten geladen werden, in denen generische Objekte enthalten sind. **REGISTER** sollte also vor dem ersten **LOAD** Kommando laufen.

Detaillierte Information über die Arbeitsweise des Generatorcaches und des Generatorverzeichnisses finden Sie im *MGEN Programmer's Guide*.

OPTIONEN

-GENERATOR pn

es wird der Generator gen registriert. pn ist der Pfadname eines ausführbaren Programms.

-GCACHE pn

es wird der Pfadname des Generatorcaches registriert. pn ist der Pfadname eines Directories. Es kann immer nur eine externer Generatorcache benutzt werden.

WEITERES

INSPECT(PARIS),LOAD(PARIS),MgenGeneratorCache(3),MgenGeneratorDirectory(3),
The MGEN Programmer's Guide

NAME

RENCELDEF – Umbenennen einer Zelldefinition

SYNOPSIS

RENCELDEF oldname newname [-GROUP]

BESCHREIBUNG

Ändert den Namen einer Zelldefinition. Wenn die Zelle `newname` bereits existiert, wird diese nicht gelöscht, sondern es erfolgt eine Fehlermeldung. Wird eine Macrogruppe für die neue Zelle angegeben, wird die Zelle zusätzlich unter dem neuen Namen in der angegebenen Macrogruppe eingehängt. Die angegebene Macrogruppe muß diese schon existieren. Wenn die Gruppe nicht existiert, kann sie mit dem Kommando `CREATE -GROUP` erzeugt werden.

ARGUMENTE

oldname (cn) Alter Name der Zelldefinition

newname (cn) Neuer Name der Zelldefinition

OPTIONEN

-GROUP mg Name der Macrogruppe, der die neue Zelle zugeordnet werden soll
DEFAULT: ursprüngliche Macrogruppe wird beibehalten

FEHLER

ECELLNOTINDB oldname existiert nicht

EDEFEXISTS newname existiert schon

EGROUPNOTEXISTS Die angegebene Gruppe existiert nicht

WEITERES

CHANGETOGROUP(paris), CREATE(paris)

NAME

ROTATE – Rotieren von Instanzen

SYNOPSIS

ROTATE -ROT90 | -ROT180 | -ROT270

BESCHREIBUNG

führt die spezifizierte Rotation für alle selektierten Instanzen durch. Die Instanzen werden um ihren Aufrufpunkt rotiert, verändern dabei aber ihre Translation nicht.

OPTIONEN

-R90	Rotation um 90 Grad.
-R180	Rotation um 180 Grad.
-R270	Rotation um 270 Grad.

WEITERES

SYMOP(PARIS), MIRROR(PARIS)

NAME

RPC - RPC Service (de)aktivieren

SYNOPSIS

RPC -ON | -OFF

BESCHREIBUNG

PARIS bietet über eine RPC (**R**emote **P**rocedure **C**all) Schnittstelle die Möglichkeit, auf die PARIS internen Daten zuzugreifen oder PARIS Kommandos von einem anderen Prozeß aus aufzurufen. Die RPC Schnittstelle ist auf dem ONC RPC System aufgebaut. Die Kommunikation mit einem anderen Prozeß erfolgt mit Hilfe des TCP/IP Protokolls. PARIS dient dabei als Server, die über RPC integrierte Applikation als Client. Zum Programmieren des Clients existiert ein Client API, in dem die Funktionalität der RPC Schnittstelle gekapselt ist.

Mit Hilfe des Kommandos RPC wird die RPC Schnittstelle von PARIS aktiviert bzw. deaktiviert. Wenn die RPC Schnittstelle aktiviert ist, erzeugt PARIS eine Shell Variable `RPCPORT`, die die Portnummer enthält, unter der der PARIS Prozeß adressiert werden kann. Bei inaktiver RPC Schnittstelle existiert die Shell Variable `RPCPORT` entweder nicht, oder hat den Wert 0. Mit Hilfe der Variablen `RPCPORT` ist es möglich, Client Prozesse innerhalb von PARIS als normale Kommandos zu starten. Dabei müssen Clients aber immer als Hintergrundprozesse laufen.

RPC liefert `true` als return Wert, wenn alles erfolgreich verlaufen ist, andernfalls `false`. Damit lassen sich Clients durch folgende Prozedur starten:

```
if [ rpc -on ]
then
    client -host :$RPCPORT &
fi
```

Es ist ebenfalls möglich, Clients auf einem anderen Rechner abelaufen zu lassen. Dabei muß allerdings der Benutzer dafür sorgen, daß der richtige Server Prozeß adressiert wird.

OPTIONEN

- ON Die RPC Schnittstelle wird aktiviert. Wenn die Schnittstelle schon aktiv ist, bleibt sie weiterhin aktiv und es geschieht nichts. Ein mehrfacher Aufruf von `RPC -ON` ist möglich. Beim ersten Aktivieren wird die Shell Variable `RPCPORT` erzeugt.
- OFF Deaktiviert die RPC Schnittstelle. Der Wert der Shell Variablen `RPCPORT` wird auf 0 zurückgesetzt.

VORSICHT

Wenn ein Client Prozeß innerhalb von PARIS gestartet wird, muß dieser unbedingt als Hintergrundprozeß laufen. Andernfalls blockieren sich PARIS und der Client gegenseitig.

WEITERES

PRSRPC(3), ash(1)

NAME

SAVE - Abspeichern von Macrogruppen

SYNOPSIS

SAVE [-V4] [mg [-PATHNAME]] ...

BESCHREIBUNG

Speichert die spezifizierte(n) Macrogruppe mg bzw. alle geänderten Macrogruppen als Datei(en) auf der Platte ab, falls die Gruppen nicht schreibgeschützt sind. Wenn keine Macrogruppen angegeben sind, werden alle geänderten Gruppen abgespeichert.

ARGUMENTE

mg Name der zu speichernden Macrogruppe.

OPTIONEN

-PATHNAME fn zu jeder Angabe eines Gruppennames kann optional mit dem Schalter -PATHNAME ein Pfadname angegeben werden, in dem die Gruppe auf der Platte abgespeichert wird. Fehlt die Angabe eines Pfadnames, wird die Gruppe unter ihrem Gruppennamen abgespeichert.

-V4 Bei Angabe dieses Schalters wird die Macrogruppe in dem von PARIS Version 4 lesbaren Format geschrieben. Die Information, die in Version 5 gegenüber Version 4 neu ist, i.e. Properties, nicht 90 Grad Winkel bei Instanzen, geht dabei verloren.

AKTUALISIEREN VON GENERIERTEN ZELLEN

Manche Generatoren benutzen von Hand erzeugte Zellen, um damit größere Zellen aufzubauen. Bei einer Änderung der handgemachten Zellen gibt es dann Unstimmigkeiten zwischen den generierten und den handgemachten Zellen. In derartigen Fällen ist ein automatisches Aktualisieren der generierten Zellen erforderlich. Dies erfolgt nach dem Abspeichern, da zu diesem Zeitpunkt der Generator den Zugriff auf die geänderten Zellen hat, die in der abgespeicherten Macrogruppe stehen.

Das automatische Aktualisieren generierter Zellen erfolgt nur für solche Zellen, die von handgemachten Zellen abhängig sind. Diese Abhängigkeit wird beim Generieren der Zellen bzw. beim Laden der generierten Zellen vom Generatorcache festgestellt. Zusätzliche Benutzereingriffe sind dafür nicht erforderlich.

Das Aktualisieren generierter Zellen nach dem Abspeichern einer Macrogruppe erfolgt im Prinzip wie das Neugenerieren dieser Zellen. Dabei wird an Hand des Modifikationsdatums der generierten Zelle und der Macrogruppe ermittelt, ob eine Aktualisierung erforderlich ist. Wenn ja, werden die generierten Zellen im internen und evtl. externen Generatorcache ersetzt. Durch das Neugenerieren wird immer eine Netzlistenextraktion der neu generierten Zelle angestoßen, wodurch evtl. bei größeren generierten Zellen mit einer entsprechenden Laufzeit gerechnet werden muß. Die Check-/Netflags der Zellen in der Hierarchie oberhalb der Instanz der neugenerierten Zellen werden zurückgesetzt.

In der derzeitigen Implementierung wird der Updatevorgang nur dann richtig ausgeführt, wenn die von Hand veränderte Macrogruppe und die davon abhängige(n) generierten Zellen von ein und demselben Prozeß kontrolliert werden. Wird die originäre Macrogruppe von einem anderen Prozeß bearbeitet, werden Veränderungen möglicherweise nicht richtig erkannt, so daß es dann zu Unstimmigkeiten zwischen den Originalzellen und den generierten Zellen kommen kann.

FEHLER

EGROUPNOTEXISTS

Die spezifizierte Gruppe existiert nicht.

EGRNOTCHANGED

Die spezifizierte Gruppe wurde nicht verändert.

WEITERES

LOAD(Paris)

NAME

SAVEPAR – Abspeichern der Designparameter

SYNOPSIS

SAVEPAR [name]

BESCHREIBUNG

speichert die aktuellen Layoutparameter ab

ARGUMENTE

name (fn) speichert die Parameter unter dem neuen Namen 'fn' ab.
DEFAULT: alter Name wird beibehalten

WEITERES

LOADPAR(paris)

NAME

SCALE – skaliert Komponenten

SYNOPSIS

SCALE *i1 i2* [-CELL | -GROUP | -SELECT | -ALL]

BESCHREIBUNG

skaliert die Komponenten der ausgewählten Zellen, bzw. der aktuellen Zelldefinition, oder die selektierten Komponenten. Skalierung von Komponenten bedeutet die Multiplikation aller Koordinaten mit $i1/i2$. Nach der Skalierung werden die Koordinaten auf das aktuelle Minimalgrid gerastet. Die Skalierung von Instanzen betrifft nur den Translationsanteil der Instanztransformation. Der Transformationsanteil bleibt unverändert. Für alle geänderten Zellen werden `net_flag` und `check_flag` auf `false` gesetzt.

ARGUMENTE

i1 i2 (i) Die durch die Optionen ausgewählten Komponenten bzw. die aktuelle Zelldefinition werden mit dem Faktor $i1/i2$ skaliert.

OPTIONEN

-CELL *cn* [...] die Komponenten der spezifizierten Zelldefinitionen werden skaliert.
-GROUP *mg* [...] die Komponenten aller Zelldefinitionen der spezifizierten Macrogruppen werden skaliert.
-SELECT die selektierten Komponenten werden skaliert.
-ALL die Komponenten aller Zelldefinitionen des gesamten Layouts werden skaliert.
DEFAULT: aktuelle Zelldefinition

NAME

SELECT – Selektieren von Komponenten

SYNOPSIS

SELECT -POINT | -RECT | -ALL -VERTEX | -EDGE | -GEO | -TEXT | -CELL | -COMP |
-NODE | -BAG | -NET | -PIN | -CONNECTIONS | -INPOLY

BESCHREIBUNG

Selektiert die Komponenten des spezifizierten Typs, die ganz oder teilweise innerhalb des angegebenen Fensters liegen.

OPTIONEN

-POINT p das 'Select Window' wird durch den letzten MARK Punkt und p definiert
-RECT p1 p2 das 'Select Window' wird durch p1 und p2 definiert
-ALL selektiert all Komponenten
-VERTEX selektiert alle Eckpunkte im 'Select Window'
-EDGE selektiert die Kanten bzw. deren Eckpunkte, die ganz oder teilweise im 'Select Window' liegen
-GEO selektiert alle Geoelemente, die ganz oder teilweise im 'Select Window' liegen
-TEXT selektiert alle Textelemente, deren Referenzpunkte im 'Select Window' liegen
-CELL selektiert alle Zellaufrufe, die ganz oder teilweise im 'Select Window' liegen
-COMP selektiert alle Geoelemente, Textelemente und Zellaufrufe, die ganz oder teilweise im 'Select Window' liegen
-NODE selektiert alle Geoelemente, die zum Knoten des im 'Select Window' liegenden Geoelementes gehören; die angeschlossenen Zellaufrufe können mit dem Kommando `INSPECT -SELNODE` gelistet werden. Es darf nur eine Komponente selektiert werden.
-BAG selektiert alle Bags, die ganz oder teilweise im 'Select Window' liegen
-NET selektiert alle Netze, die ganz oder teilweise im 'Select Window' liegen
-PIN selektiert alle Pins, die ganz oder teilweise im 'Select Window' liegen
-CONNECTIONS selektiert alle Netze, die an die Instanz angeschlossen sind, die im 'Select Window' liegt.
-INPOLY selektiert alle Eckpunkte und alle Textelemente, die in den partiell oder total selektierten Geometrieelementen liegen. Die schon vorher selektierte Geometrie bleibt selektiert.

WEITERES

`UNSELECT(Paris),IDENTIFY(Paris),FIND(Paris)`

NAME

SELECTSTACK – Verwaltung einer Menge von Selektlisten

SYNOPSIS

SELECTSTACK -CLEAR | -EXCHANGE | -MERGE | -POP | -PUSH | -REMOVE

BESCHREIBUNG

In einer editierten Zelle wird eine Menge von Selektlisten angelegt. Die jeweils aktive Selektliste kann auf den Selektstack geschoben werden. Die Elemente bleiben dabei selektiert, sind aber nicht bearbeitbar (passive Selektion). Zwischen der aktiven Selektion und dem passiven Selektstack können die jeweiligen Selektlisten hin und her fließen. Die Tiefe des Selektstacks ist beliebig.

OPTIONEN

- CLEAR deselektiert alle Komponenten im Selektstack und leert den gesamten Selektstack.
- EXCHANGE [i] tauscht die aktive Selektliste mit dem i-ten (default obersten) Element im Selektstack. Falls das spezifizierte Element im Selektstack nicht existiert, erfolgt **SELECTSTACK -PUSH**.
- MERGE [i] fügt i-tes/oberstes Element vom Selektstack in die aktive Selektliste ein und entfernt Element vom Selektstack. Die schon aktive selektierten Elemente bleiben selektiert.
- POP [i] deselektiert die aktive Selektliste und holt i-tes/oberstes Element vom Selektstack.
- PUSH schiebt die aktive Selektliste auf den Selektstack.
- REMOVE [i] entfernt i-tes/oberstes Element vom Selektstack und deselektiert alle Komponenten in diesem Element.

WEITERES

SELECT(Paris), UNSELECT(Paris), FIND(Paris)

NAME

SETLAYER – Ändern der Sichtbarkeit/Editierbarkeit von Ebenen

SYNOPSIS

SETLAYER [-ALL] -NEITHER | -VISIBLE | -DETECTABLE

BESCHREIBUNG

setzt die Darstellungsart (unsichtbar, sichtbar, editierbar) der spezifizierten Ebenengruppen; es können mehrere Optionen hintereinander folgen. Die Optionen werden in der Reihenfolge ihres Auftretens abgearbeitet. Die letzte Einstellung für eine Ebene in einer Kommandozeile ist daher effektiv wirksam.

BEISPIEL

```
setlayer -n all -v 33,80 -d alu1
```

OPTIONEN

- ALL Die Layereinstellungen werden an allen Fenstern und der Technologiebeschreibung durchgeführt.
DEFAULT: die Einstellung wird im aktuellen Fenster durchgeführt, wenn SETLAYER aus einer editierten Zelle aufgerufen wird, bzw. in der Technologie, wenn keine Zelle editiert ist.
- NEITHER gl spezifizierte Ebenen werden auf unsichtbar gesetzt
- VISIBLE gl spezifizierte Ebenen werden auf sichtbar gesetzt
- DETECTABLE gl spezifizierte Ebenen werden auf editierbar gesetzt

WEITERES

DEFLAYER(paris)

NAME

SETPAR - Setzen der Designparameter

SYNOPSIS

SETPAR -TEXTSIZE | -ACTORTHO | -ACTLAYER | -TOLCELL | -SIZEPINNAME |
-SIZEPLNAME | -SIZECELLNAME | -LAYERPINNAME | -LAYERPLNAME |
-LAYERCELLNAME | -LAYERSYMBOLIC -AUTOPNAME |
-AUTOLOCPNAME | -LOCPNAME | -GRID | -GRIDOFFSET | -GRIDFAKTOR |
-PATHGRID | -PATHGRIDOFFSET | -PCIRCLE | -SNAPCIRCLE | -PATHTYPE |
-PATHWIDTH | -VECORTHO | -DESCEND | -DRAWCROSS | -AMSUBST

BESCHREIBUNG

setzt verschiedene Designparameter. Sie können durch ein SAVEPAR in eine Datei gespeichert werden

OPTIONEN

- TEXTSIZE in voreingestellte Größe für neu zu addierende Textelemente
- ACTORTHO o voreingestellte Orthogonalität für neu zu addierende Komponenten
- ACTLAYER i voreingestellte Ebene für neu zu addierende Komponenten
- TOLCELL i Toleranz (in %) für das Einrasten von Zellaufrufen
- SIZEPINNAME in
 Textgröße der Pinnamen
- SIZEPLNAME in
 Textgröße der Plazierungsnamen. Wenn die angegebene Größe ein positiver Wert ist, werden die Plazierungsnamen in dieser Größe gezeichnet. Bei einem negativen Wert werden die Plazierungsnamen in 90% der Größe des umschließenden Rechtecks der Zellen gezeichnet in dem umschließenden Rechteck zentriert.
- SIZECELLNAME in
 Textgröße der Zellnamen. Wenn die angegebene Größe ein positiver Wert ist, werden die Zellnamen in dieser Größe gezeichnet. Bei einem negativen Wert werden die Zellnamen in 90% der Größe des umschließenden Rechtecks der Zellen gezeichnet in dem umschließenden Rechteck zentriert.
- LAYERPINNAME in
 setzt den Layer der Pinnamen. Der Layer für die Pin-, Plazierungs- und Zellnamen ist relevant für die Darstellung der Namen am Bildschirm und die GDSII Ausgabe dieser Texte. Für die Namensdarstellung wird die Farbe des gewählten Layers verwendet. Für die GDSII Ausgabe die Konvertierung des eingestellten Layers.
- LAYERPLNAME in
 setzt den Layer der Plazierungsnamen. Die Plazierungsnamen werden in der Farbe des angegebenen Layers gezeichnet.
- LAYERCELLNAME in
 setzt den Layer der Zellnamen. Die Zellnamen werden in der Farbe des angegebenen Layers gezeichnet.
- LAYERSYMBOLIC in
 setzt den Layer für die Darstellung von symbolisch gezeichneten Instanzen. Die Boxen von symbolisch gezeichneten Instanzen werden mit der Farbe und dem Füllmuster des angegebenen Layers gezeichnet. In der Box zentriert erscheint der Zellname.

- AUTOPNAME ON | OFF
automatische Vergabe der Platzierungsnamen (Durchnummerierung)
- AUTOLOCPNAME ON | OFF
automatische Positionierung der Platzierungsnamen an der mit LOCPNAME spezifizierten Position
- LOCPNAME in1 in2
voreingestellte Position für die automatische Vergabe der Position von Platzierungsnamen
- PATHGRID in1 in2
Setzen des 'pathgrid' auf die spezifizierten Werte 'in1 in2'
- PATHGRIDOFFSET in1 in2
Setzen des 'pathgrid offset' auf die spezifizierten Werte 'in1 in2'
- GRID in1 in2 Setzen des Grundrasters (Minimalgrid) auf die spezifizierten Werte 'in1 in2'
- GRIDOFFSET in1 in2
Setzen des 'grid offset' auf die spezifizierten Werte 'in1 in2'
- GRIDFAKTOR i Setzen des Gridfaktors. Grundraster `grid` * Gridfaktor ergeben das Normgrid
- PCIRCLE i Setzt die Auflösung, mit der ein Kreis oder Bogen in ein Polygon zerlegt wird bei der Eingabe eines Kreises oder Bogens. Die Zahl `i` gibt dabei die Anzahl der Eckpunkte eines Vollkreises an. Bei einem Bogen wird die Anzahl der Eckpunkte aus dem Winkel, den der Bogen überspannt und des Parameters `i` anteilig bestimmt.
DEFAULT: PCIRCLE=32
- SNAPCIRCLE ON | OFF
gibt an, ob die Punkte von Kreisen oder Bögen auf das Minimalgrid gerastet werden oder nicht.
DEFAULT: ON
- PATHWIDTH voreingestellte Weite für neu zu addierende Pfade
- PATHTYPE pt gibt an, ob Pfade mit oder ohne Überhang addiert werden
DEFAULT: mit Überhang
- VECORTHO gibt an, auf welche Orthogonalität die Verschiebungsvektoren bei MOVE und COPY zu rasten sind.
- DESCEND i steuert die die DESCEND Behandlung beim Bildaufbau. Der DESCEND Level `i` kann die Werte 0, 1 oder 2 annehmen. Wenn `i` 0 ist, wird die symbolische Darstellung von Instanzen ganz ausgeschaltet. Es werden dann immer alle Ebenen gezeichnet. Wenn `i` 1 ist, wird die objektbezogene symbolische Darstellung durchgeführt. Für die Darstellung relevant sind dann die DESCEND Attribute an den einzelnen Instanzen. Dies ist die Voreinstellung. Wenn `i` den Wert 2 hat, werden alle Instanzen im aktuellen Toplevel symbolisch gezeichnet. Es erscheinen dann nur die Bounding Box und die Pins der Instanzen. Damit wird am wenigsten gezeichnet, so daß der Bildaufbau optimal schnell geht.
- DRAWCROSS on|off
gibt an, ob beim Zeichnen des Rasters auf jeden fünften Rasterpunkt ein Kreuz gezeichnet wird oder nicht.

NAME

SIZE – Verzerren von selektierten Komponenten

SYNOPSIS

SIZE -DELTA

BESCHREIBUNG

verzerrt die selektierten Geoelemente um den Betrag `delta` (positiv oder negativ)

OPTIONEN

-DELTA in Verzerrwert

NAME

SMASH - Auflösung der Zellhierarchie

SYNOPSIS

SMASH [-SEL | -DEF [-KEEP] [-ADDSEL]

BESCHREIBUNG

Expandiert Zellhierarchie bis zur angegebenen Schachtelungstiefe.

OPTIONEN

- SEL expandiert die selektierten Zellaufrufe und ersetzt sie durch den Inhalt der Zellen;
- DEF cn1 cn2 mg expandiert die Zelle 'cn1' und erstellt eine neue Zelldefinition mit dem Namen 'cn2' in der Macrogruppe 'mg'; die Zelldefinition 'cn1' bleibt erhalten
- KEEP i erhält alle Zelldefinitionen, deren Nestingtiefe nicht größer als i ist
DEFAULT: i=0, d.h. es wird bis zur untersten Hierarchiestufe expandiert
- ADDSEL nur mit Option -SEL: addiert die gesmashten Komponenten selektiert
DEFAULT: die gesmashten Komponenten sind nicht selektiert

WEITERES

SMASHSIZE(PARIS)

NAME

SMASHSIZE – Auflösen der Zellhierarchie mit Verzerren der Komponenten

SYNOPSIS

SMASHSIZE cellname -NEWNAME -GROUP [-SIZEFILE] [-LAYER]

BESCHREIBUNG

Expandiert eine Zelldefinition bis zur untersten Hierarchiestufe und verzerrt anschließend die Grundkomponenten nach der mit SIZEFILE spezifizierten Verzerrtabelle. Es wird eine neue Zelldefinition erstellt.

ARGUMENTE

cellname (cn) Name der zu bearbeitenden Zelldefinition

OPTIONEN

-SIZEFILE fn Datei, in der die Verzerrmaße der einzelnen Ebenen aufgeführt sind
-NEWNAME cn Name der neuen Zelldefinition, die die gesmashten und verzerrten Komponenten der Definition 'cn' enthält
-GROUP mg Name der Macrogruppe, der die neue Zelldefinition zugeordnet werden soll
-LAYER lset Es werden nur Elemente im Layerset lset berücksichtigt.

WEITERES

SMASH(PARIS)

NAME

SNAP – Einrasten von Komponenten

SYNOPSIS

SNAP [-CELL | -GROUP | -SELECT | -ALL]

BESCHREIBUNG

rastet die Komponenten der ausgewählten Zellen, bzw. der aktuellen Zelldefinition, oder die selektierten Komponenten. Bei Zellaufrufen wird die Translation gerastet. Für alle gänderten Zellen werden `net_flag` und `CHECK_FLAG` auf `false` gesetzt.

OPTIONEN

- CELL cn [...] die Komponenten der spezifizierten Zelldefinitionen werden gerastet.
- GROUP mg [...] die Komponenten aller Zelldefinitionen der angegebenen Macrogruppen werden gerastet.
- SELECT die selektierten Komponenten werden gerastet.
- ALL die Komponenten aller Zelldefinitionen des gesamten Layouts werden gerastet.
- DEFAULT: aktuelle Zelldefinition

NAME

STEP – kopiert Komponenten und ordnet sie als Array an

SYNOPSIS

STEP -POINT | -VECTOR | -DIST | -BBG -DIM

BESCHREIBUNG

kopiert die spezifizierten Komponenten und ordnet sie als Array an

OPTIONEN

- POINT p der Verschiebungsvektor setzt sich aus dem letzten MARK und 'p' zusammen
- VECTOR p1 p2 spezifiziert den Abstand zwischen zwei Array Elementen durch Eingabe der Punkte 'p1' und 'p2'
- DIST in1 in2 spezifiziert den Abstand zwischen zwei Array Elementen durch Eingabe der Distanz in x- und y-Richtung
- BBG als Abstand zwischen zwei Array Elementen wird das umschließende Rechteck der Geometrie aller selektierten Elemente verwendet.
- DIM i1 i2 Anzahl der Array Elemente in x- und y-Richtung

WEITERES

COPY(PARIS), MOVE(PARIS)

NAME

SUBEX – Extraktion von Substrat Parasiten

SYNOPSIS

SUBEX [-SUSI] [-SELECTED | -WINDOW] [-SUBAREA] -MODEL
[-OUTPUT] [-HEADER] [-HMIN] [-WS] [-SET] [-FMIN] [-FMAX] [-FSTEP]
[-MAXDELTAU] [-LINEAR] [-METALLIZED]

BESCHREIBUNG

extrahiert die ausgewählten Komponenten unter Berücksichtigung der Daten der Modelldatei und gibt sie in eine Datei aus, die als Eingangsdatei fuer eine Parasitenberechnung mit CAPCAL bzw. SUSI dient. Die ausgegebenen Komponenten werden zusätzlich in eine neu erzeugte Zelldefinition in der (ggf. neu erzeugten) Macrogruppe SUBEXDATA aufgenommen. Die neue Zelldefinition erhält den Namen der Ausgangsdatei (ohne Extension).

OPTIONEN

-SUSI Parasitenberechnung mit SUSI
DEFAULT: Parasitenberechnung mit CAPCAL

-SELECTED die selektierten Komponenten werden bearbeitet

-WINDOW die Komponenten des aktuellen Bildschirm- ausschnittes werden bearbeitet
DEFAULT: die Komponenten der aktuellen Zelldefinition werden bearbeitet

-MODEL fn Pfadname der Modelldatei

-OUTPUT fn Pfadname der Ausgabedatei (CAPCAL/SUSI-Inputfile)
DEFAULT: data.inp

-HEADER st String für den Header im CAPCAL/SUSI-Inputfile

-METALLIZED Rückseite des Substrats ist metallisiert
DEFAULT: Rückseite des Substrats ist nicht metallisiert

Optionen fuer CAPCAL:

-SUBAREA ein Rechteck im Layer SUBSTRAT bestimmt das Substratgebiet
DEFAULT: die Boundingbox der Zelldefinition bestimmt das Substratgebiet

-HMIN v Wert des minimalen Grids (in um)
DEFAULT: 5 um

-WS v Wert der Substratdicke (in um)
DEFAULT: 300 um

Optionen fuer SUSI:

-SET st Name des Parametersets

-FMIN v Minimale Frequenz in Hz
DEFAULT: 1.E6 Hz

-FMAX v Maximale Frequenz in Hz
DEFAULT: 1.E10 Hz

-FSTEP v Schrittweite der Frequenz in Hz
DEFAULT: 10 Hz

-LINEAR Lineare Frequenzteilung
DEFAULT: Logarithmische Frequenzteilung

-MAXDELTAU v Maximale Spannungsdifferenz in V
DEFAULT: 0 V

NAME

SUBTUN - Ein-/Ausschalten des Gate Array Untergrundes

SYNOPSIS

SUBTUN -ON | -OFF

BESCHREIBUNG

kontrolliert die Darstellung des Untergrunds (Gate Array Master) bei Designs vom Typ GA

OPTIONEN

-ON schaltet Untergrund Darstellung ein
-OFF schaltet Untergrund Darstellung aus

WEITERES

PLOT(paris)

NAME

SYMOP – Spiegeln/Drehen von selektierten Komponenten

SYNOPSIS

SYMOP [point] -FLIPX | -FLIPY | -ROT90 | -ROT90FLIPX | -ROT90FLIPY | -ROT180 |
-ROT270 | -ANGLE

BESCHREIBUNG

Führt die spezifizierte Symmetrieoperation für alle selektierten Komponenten durch. Die durchgeführte Aktion ist abhängig davon, ob ein Bezugspunkt angegeben wird oder nicht. Wenn ein Bezugspunkt angegeben ist, werden alle selektierten Elemente um diesen Bezugspunkt gedreht oder gespiegelt. Wenn kein Bezugspunkt angegeben ist, werden nur die selektierten Instanzen um ihren Aufrufpunkt gedreht oder gespiegelt.

ARGUMENT

point (p) Punkt, an dem die entsprechende Symmetrieoperation durchgeführt werden soll

OPTIONEN

-FLIPX Spiegelung um die x-Achse
-FLIPY Spiegelung um die y-Achse
-ROT90 Drehung um 90 Grad (mathem. positiv)
-ROT180 Drehung um 180 Grad
-ROT270 Drehung um 270 Grad
-ROT90FLIPY Drehung um 90 Grad und Spiegelung um die y-Achse.
-ROT90FLIPX Drehung um 90 Grad und Spiegelung um die x-Achse.
-ANGLE a Drehung um den Winkel a. A wird in Grad angegeben.

WEITERES

MIRROR(PARIS)

NAME

TRAVERSEDEFHIERARCHY - traversiert die Zellhierarchie

SYNOPSIS

TRAVERSEDEFHIERARCHY [cellname] [-TOPDOWN | -BOTTOMUP] [-NOLEAF] [NO-GENERIC] [-EXCLUDE cn ...] -COMMAND com [-ARGS arg ...]

BESCHREIBUNG

Traversiert die Zellhierarchie der angegebenen Zelle entweder in bottom up oder in top down Reihenfolge. Für jede Zelle in der Hierarchie wird das angegebene Kommando `com` mit den Argumenten `arg ...` in folgender Form aufgerufen:

```
eval com cellname arg ...
```

Der Zellname erscheint dabei immer in Kleinbuchstaben. Jede Zelle wird nur einmal besucht, auch wenn sie mehrfach in der Hierarchie aufgerufen wird. Es besteht die Möglichkeit bestimmte Zellen nicht zu bearbeiten, entweder über den Zellnamen, oder grundsätzlich alle Blattzellen.

ARGUMENTE

cellname [...] (cn) Name(n) der zu bearbeitenden Zelldefinition(en).
DEFAULT: aktuelle Zelldefinition

OPTIONEN

- TOPDOWN Die Traversierung erfolgt in top down Reihenfolge, die höher in der Hierarchie liegenden Zellen kommen zuerst.
- BOTTOMUP Die Traversierung erfolgt in bottom up Reihenfolge, die tiefer in der Hierarchie liegenden Zellen kommen zuerst. Dies ist die default Einstellung. Die Optionen `-TOPDOWN` und `-BOTTOMUP` sind alternativ. Sie dürfen nicht beide angegeben werden.
- NOLEAF Blattzellen, also solche Zellen, in denen keine Instanzen anderer Zellen enthalten sind, werden nicht besucht.
- NOGENERIC generierte Zellen werden nicht besucht.
- EXCLUDE cn ... Die angegebenen Zellen werden nicht besucht. Die Ausschlußoptionen arbeitet hierarchisch. Wenn eine Zelle von der Bearbeitung ausgeschlossen wird, wird der gesamte Hierarchiebaum unterhalb dieser Zelle nicht betrachtet. Für den Zellnamen können auch Suchmuster analog den Dateinamensmustern in der UNIX Shell angegeben werden, z.B. die Angabe `R*` schließt alle Zellen aus, deren namen mit `R` anfängt.
- COMMAND com Das Kommando `com` wird für jede der besuchten Zellen aufgerufen. Ein Kommando muß immer angegeben werden.
- ARGS arg ... Die angegebenen Argumente werden an das Kommando weitergegeben. Das erste Argument des Kommandos ist immer der Zellname der besuchten Zelle. Die angegebenen Argumente erscheinen hinter dem Zellnamen.

BEISPIEL

Alle Zellen, die in einer bestimmten Zelle benutzt werden, können mit dem folgenden Kommando angezeigt werden:

```
traversedefhierarchy -command echo
```

Dieses Kommando erzeugt die Liste aller benutzten Zellen in der gerade editierten Zelldefinition. Jeder Zellname steht dabei in einer eigenen Zeile. Die Ausgabe erfolgt in bottom up Reihenfolge. Es sind keine Zellen von der Bearbeitung ausgeschlossen.

NAME

UNDO- Rückgängigmachen der letzten Editiervorgänge

SYNOPSIS

UNDO

BESCHREIBUNG

UNDO macht den letzte Editiervorgang rückgängig. Es werden die durch den Editiervorgang erzeugten neuen Elemente wieder gelöscht und die vorher vorhandenen Elemente im selben Selektierstatus wie vor dem Editiervorgang wieder eingefügt.

Innerhalb einer editierten Zelle gibt es einen Undo Stack mit 16 Elementen. Es können also die letzten 16 Änderungen zurückgeholt werden.

Es lassen sich die Änderungen durch folgende Kommandos rückgängig machen:

DELETE, MOVE, COPY STEP, CUT, ALIGN, EQUIDIST, CLEAR, SYMOP,
MIRROR, ROTATE, EXCHANGE, REGENERATE, APPEND, NOTCH, SIZE,
SCALE, WRITE, INCLUDE.

WEITERES

DELETE(PARIS)

NAME

UNSELECT – Deselektieren von Komponenten

SYNOPSIS

UNSELECT -ALL

UNSELECT -TOP

**UNSELECT -VERTEXI-EDGEI-GEOI-TEXTI-CELLI-COMPI-NETI-PINI-CONNECTIONS
-POINTI-RECT**

UNSELECT -VERTEXI-EDGEI-GEOI-TEXTI-CELLI-COMP -ALL

BESCHREIBUNG

setzt den Status der spezifizierten Komponenten zurück auf 'unselect'

OPTIONEN

- POINT p das 'Unselect Window' wird durch den letzten MARK-Befehl und 'p' definiert
- RECT p1 p2 das 'Unselect Window' wird durch 'p1' und 'p2' definiert
- ALL deselektiert alle Komponenten. Wird -ALL in Zusammenhang mit einer Komponentenspezifikation angegeben, werden alle Komponenten vom spezifizierten Typ deselektiert,
- TOP deselektiert die erste Komponente in der Selektliste.
- VERTEX setzt den Status aller Eckpunkte im 'Unselect Window' auf 'unselect'
- EDGE setzt den Status aller Kanten, die ganz oder teilweise im 'Unselect Window' liegen auf 'unselect'
- GEO setzt den Status aller Geoelemente, die ganz oder teilweise im 'Unselect Window' liegen auf 'unselect'
- TEXT setzt den Status aller Textelemente, deren Referenzpunkte im 'Unselect Window' liegen auf 'unselect'
- CELL setzt den Status aller Zellaufrufe, die ganz oder teilweise im 'Unselect Window' liegen auf 'unselect'
- COMP setzt den Status aller Geoelemente, Textelemente und Zellaufrufe, die ganz oder teilweise im 'Unselect Window' liegen auf 'unselect'
- NET deselektiert alle Netze, die ganz oder teilweise im 'Select Window' liegen
- PIN deselektiert alle Pins, die ganz oder teilweise im 'Select Window' liegen
-CONNECTIONS deselektiert alle Netze, die an die Instanz angeschlossen sind, die im 'Select Window' liegt.

WEITERES

SELECT(Paris),FIND(Paris)

NAME

VIEW – Darstellung von Pin-,Plazierung-,Zellnamen

SYNOPSIS

VIEW [-PINNAME | -PLNAME | -CELLNAME | -AXES | NETLINES] | fn

BESCHREIBUNG

Steuert die Darstellung von Pin-, Plazierungs-, Zellnamen, Netzlinien oder öffnet ein Fenster auf die Datei `fn`. Es muss mindestens eine Option angegeben sein.

OPTIONEN

-PINNAME 0 | 1 | 2 die Darstellung der Pinnamen wird ein bzw. ausgeschaltet. Level 0 schaltet die Pinnamesdarstellung ganz ab, im Level 1 werden nur die Pinnamen von hierarchischen Zellen angezeigt, im Level 2 alle Pinnamen, auch die von Basiszellen.

-PLNAME ON | OFF
die Darstellung der Plazierungsnamen wird ein bzw. ausgeschaltet

-CELLNAME ON | OFF
die Darstellung der Zellnamen wird ein bzw. ausgeschaltet

-AXES ON | OFF die Darstellung der x/y Achsen wird ein bzw. ausgeschaltet.

-NETLINES ON | OFF
schaltet die Darstellung von Netzlinien an oder aus. Dieser Schalter setzt das `snetFlag` der aktuellen Zelle auf 2, wenn ON angegeben ist, auf 1, wenn OFF angegeben ist. Die Netzinformation der Zelle bleibt unverändert.

WEITERES

NETLIST(PARIS)

NAME

WINDOW – Aendern des Bildausschnitts

SYNOPSIS

WINDOW -NEW | -MAX | -INC | -DEC | -FIT | -FITSEL | -FITOBJ | -RETURN | -ZOOMABS
| -ZOOMREL | -CENTER | -CENTEROBJ | -MOVEABS | -MOVEREL | -MOVE-
HALF | -WIDTH | -STORE | -RESTORE | -DEFABS | -DEFREL

BESCHREIBUNG

setzt bzw. veraendert den Bildausschnitt in der aktuellen Zelldefinition

OPTIONEN

- NEW neuer Bildaufbau
- MAX maximales Fenster
- INC i vergrößert den Bildausschnitt um den Faktor i
- DEC i verkleinert den Bildausschnitt um den Faktor i
- FIT paßt den Bildausschnitt der Groesse der aktuellen Zelldefinition an
- FITSEL paßt den Bildausschnitt dem umschreibenden Rechteck der selektierten
Komponenten an
- FITOBJ oid paßt den Bildausschnitt dem umschreibenden Rechteck des Objekts mit der
Id oid an
- RETURN stellt das vorletzte Fenster noch einmal dar
- ZOOMABS p1 p2 Ausschnittsvergrößerung entsprechend dem mit 'p1' und 'p2' angegebenen
Punktepaar
- ZOOMREL p Ausschnittsvergrößerung entsprechend dem mit MARK und 'p' angegebene-
nen Punktepaar
- CENTER p behält die Bildgröße bei und setzt den Bildmittelpunkt nach 'p'
- CENTEROBJ oid behält die Bildgröße bei und setzt den Bildmittelpunkt auf den Mittelpunkt
des umschließenden Rechtecks des Objekts mit der Id oid.
- MOVEABS in1 in2 verschiebt das Fenster um den mit 'in1' (x-Richtung) und 'in2' (y-Richtung)
spezifizierten Betrag und behält die Fenstergröße bei
- MOVEREL p verschiebt das Fenster um den mit MARK und 'p' spezifizierten Betrag und
behält die Fenstergröße bei
- MOVEHALF i1 i2 verschiebt das Fenster um 'i1' bzw. 'i2' mal die halbe Fensterweite in x bzw.
y-Richtung und behält die Fenstergröße bei
- WIDTH in setzt die Fenstergröße auf die angegebene Weite
- STORE i speichert das aktuelle Fenster unter der Nummer 'i' (1-16) ab
- RESTORE i stellt das unter 'i' (1-16) abgelegte Fenster dar
- DEFABS p1 p2 i definiert den mit 'p1' und 'p2' spezifizierten Bildausschnitt als Fenster Num-
mer 'i' (1-16)
- DEFREL p i definiert den mit MARK und 'p' spezifizierten Bildausschnitt als Fenster
Nummer 'i' (1-16)

BILDAUFBAU

Kleine Elemente

Wenn bei Einstellung eines großen Bildausschnitts Elemente kleiner als 3 Bildpunkte auf dem Bildschirm abgebildet werden, werden diese nicht mehr gezeichnet. Dadurch kann die Bildaufbaugeschwindigkeit deutlich gesteigert werden. Es ist daher kein Fehler, wenn von einem Layout, in dem überwiegend kleine Elemente vorkommen, bei großem Bildausschnitt nichts auf dem Bildschirm zu sehen ist. Durch Verkleinern des Bildausschnitts werden die kleinen Elemente dann wieder sichtbar.

Die Selektierbarkeit von Elementen wird durch das Weglassen der kleinen Elemente nicht beeinflusst. Diese sind in jedem Fall selektierbar, auch wenn sie nicht sichtbar sind.

Abbruch des Bildaufbaus

Der Bildaufbau kann jederzeit durch Drücken einer beliebigen Taste innerhalb des Grafikfensters abgebrochen werden. PARIS prüft im Abstand von ca. 1s, ob eine Tastendruck im Bildfenster erfolgt ist. Wenn ja, wird der Bildaufbau abgebrochen. Damit läßt sich die Wartezeit beim Zeichnen von sehr großen Layouts vermindern.

Der Abbruch des Bildaufbaus geht analog, wenn ein Element selektiert bzw. deselektiert wird; dann jedoch für jedes Element einzeln. Werden also mehrere Elemente selektiert, muß evtl. mehrfach eine Taste gedrückt werden, um die Zeit zum Neuzeichnen der selektierten/deselektierten Elemente abzukürzen.

NAME

WRITE – Schreibt selektierte Komponenten in Puffer oder File.

SYNOPSIS

WRITE [-GEO] [-CAL] [-TEX] [-DELETE] [buffer | -FILE | -]

BESCHREIBUNG

Mit dem **WRITE** Befehl lassen sich selektierte Komponenten in einen Puffer, Datei oder die Standardausgabe schreiben. Diese können mit dem **'INCLUDE'**-Kommando wieder eingelesen werden.

ARGUMENTE

buffer (fn) Puffer, in den die Komponenten geschrieben werden. Der Puffer kann ein benannter Puffer oder der namenlose Standardpuffer sein. Es kann beliebig viele Puffer mit Namen geben, jedoch nur einen Namenlosen. Die Puffer sind so lange aktiv, wie das Programm aktiv ist.
DEFAULT: namenloser Standardpuffer

OPTIONEN

-GEO Gibt nur Geoelemente aus
DEFAULT: alle selektierten Komponenten

-CAL Gibt nur Zellaufrufe aus
DEFAULT: alle selektierten Komponenten

-TEX Gibt nur Textelemente aus
DEFAULT: alle selektierten Komponenten

-DELETE Löscht Komponenten nach der Ausgabe

-FILE fn Gibt auf 'fn' aus
DEFAULT: namenloser Standardpuffer

- Gibt auf Standardausgabe aus.
DEFAULT: namenloser Standardpuffer

WEITERES

INCLUDE(paris)

NAME

PARIS FORMAT - Beschreibung der PARIS Ein-/Ausgabe

SYNOPSIS

```
#ifndef _PRSIO_H
#define _PRSIO_H

#include <sys/types.h>
#include <stdio.h>
#include <base.h>
#include <gu.h>
#include <mgen2/globals.h>

/* revisions of data format:
 * < 2.00 PARIS version < 5
 * 2.00 PARIS version 5, add file header, properties, place flag 8
 * 2.01 add schematic nets
 * 2.02 longer pin names, 32 characters
 * 2.03 add protections
 * 2.04 add bags and object ids, shaped instances
 * 2.05 add port, treat net as special bags, treat actnet as pin objects
 * 2.06 add selected shape for sinst, generic instances
 * 2.07 add protection for pins
 * 2.08 add text layer
 * 2.09 add visualization polygons
 */

#define MAXPNAMLEN 9
#define MAXNAMELEN 256
#define MAXPINLEN 32

/* open mode for prs_fopen */
enum {prs_READ, prs_WRITE};
/* types of PRStokenData_t */
enum {PRSDEF, PRSGEO, PRSCAL, PRSTEX, PRSNET, PRSBAG, PRSSCL, PRSPRT,
      PRSCPN, PRSGCL, PRSNTL, PRSVPL,
      PRSNODNAM, PRSEQUIR, PRSEQUIC, PRSREV, PRSPROP, PRSNONE};

typedef char PRSname_t[MAXNAMELEN];
typedef gu_point_t PRSxy_t;
typedef gu_rect_t PRSbox_t;

typedef struct { /* one Element of a multiple rectangle */
    PRSxy_t ll, ur; /* lower left corner, upper right corner */
    int layerinc; /* layer increment */
} PRSnr_t;

typedef struct { /* snap point, only for compatibility mode with older PARIS versio
    PRSxy_t xy;
    char p0;
} PRSsnap_t;

typedef struct { /* this information contains a layer set and a boolean flag */
```

```

    int nDescendLayer; /* the layer that are drawn during the drawing procedure of
    int descendLayer[MAXLAYER]; /* the layout editor, when it "descends" one step dow
    Boolean_t descend; /* in hierarchy, this information is internal to the PARIS 1
} PRSdescend_t;

typedef struct { /* descriptor of geometric elements */
    GeoFunc func; /* geo function, one of OBSTRUCTION, PIN, WIRE, DIVERSE, SUPPL
    char pinname[MAXPINLEN]; /* if func == PIN, pinname */
    PinType pintype; /* type of pin, one of INP, OUTP, UNDEF, E, A */
    char textDir; /* how the pinname is drawn by the layout editor, [V]ERTICAL o
    PRSxy_t ppos; /* where the pin name is to be drawn */
} PRSgeoFuncDesc_t;

typedef struct { /* buffer for polygon */
    PRSxy_t *xy; /* points */
    int nxy; /* number of points */
    gu_Ortho_t ortho; /* orthogonality, one of gu_OYES, gu_HALF, gu_ONO, gu_OTEST
} PRSpolygon_t;

typedef struct { /* path */
    PRSxy_t *xy; /* points */
    long width; /* path width */
    int nxy; /* number of points */
    gu_PathType_t patype; /* path type, gu_OVERHANGPATH or gu_NOOVERHANGPATH */
    gu_Ortho_t ortho; /* orthogonality, one of gu_OYES, gu_HALF, gu_ONO, gu_OTEST
} PRSpath_t;

typedef struct { /* rectangle */
    PRSxy_t xy[4]; /* points */
    gu_Ortho_t ortho; /* orthogonality, one of gu_OYES, gu_HALF, gu_ONO, gu_OTEST
} PRSrect_t;

typedef struct { /* multiple rectangle */
    PRSnr_t *nrs; /* rectangles */
    int nnrs; /* number of rectangles */
} PRSnrect_t;

typedef struct _PRSgeo_t { /* geometric element */
    ProtectionType protection;
    int layer; /* layer of element */
    long node; /* net number of element in layout */
    long snode; /* net number of element in schematic */
    PRSgeoFuncDesc_t gf; /* function descriptor */
    int type; /* type flag, PATH, POLYGON, RECTANGLE, N_RECTANGLE */
    union {
        PRSpath_t pa;
        PRSpolygon_t po;
        PRSrect_t r;
        PRSnrect_t nr;
    } body;
} PRSgeo_t;

```

```

typedef struct _PRSactnet_t {          /* connectivity of an instance */
    char pinname[MAXPINLEN];          /* name of connected pin of instanced cell */
    long enode;                        /* net number of connection in layout */
    long senode;                       /* net number of connection in schematic */
} PRSactnet_t;

typedef struct {                      /* a cell's port */
    PRSname_t name;                   /* name of port */
    long inode;                       /* net number of connection in layout */
    PinType type;                     /* port type */
} PRSport_t;

typedef struct {                      /* connectivity of an instance */
    PRSname_t name;                   /* name of connected pin of instanced cell */
    long enode;                       /* net number of connection in layout in the instancin
    ProtectionType protection;        /* normally only contains the sticky bit */
} PRSpin_t;

typedef struct _PRSbag_t {
    PRSname_t name;
    PRSoid_t *oids;
    int noids;
} PRSbag_t;

typedef struct _PRSinst_t {          /* instance */
    PRSbag_t bag;                     /* bag part */
    PRSname_t mname;                  /* instance name, cell name */
    ProtectionType protection;
    PRSxy_t translation, mtr, ptr;    /* instance translation and the drawing position
    PRSactnet_t *an;                  /* the connectivity (obsolete since version 2.05) */
    int nan;                          /* number of elements in an */
    symop_t symop;                   /* transformation NOSYM=0, FY=1, R90=2, R90FX=3, R180=4, FX=5,
    char mdir, pdir;                  /* the drawing direction of cell name, instance nam
    double transform[2][2];           /* tranformation matrix, if symop == 8 */
    PRSdescend_t descend;             /* how to draw the instance */
} PRSinst_t;

typedef struct _PRStext_t {          /* graphic text */
    ProtectionType protection;
    PRSname_t txt;                    /* text string */
    long x, y, size;                  /* position and size of text */
    int layer;                        /* layer of text, in PARIS unused */
    char td;                          /* drawing direction */
} PRStext_t;

typedef struct _PRSnet_t {          /* net to hold net information from schematic */
    PRSbag_t b;                       /* bag part */
    long nn;                          /* net number, works as a pointer to elements identified bys
} PRSnet_t;

typedef struct _PRSsinst_t {        /* instance with shapes buffer, type SCL */
    PRSinst_t i;                      /* instance part */

```

```

    Const char* shapes;          /* containing shape information */
    int          selectedShape;   /* selected shape, -1 for no selection */
} PRSsinst_t;

typedef struct _PRSGinst_t { /* generic instance type GCL */
    PRSsinst_t s;      /* shaped instance part */
    PRSname_t  g;      /* generator name */
    Const char* args;  /* generator args */
} PRSGinst_t;

typedef struct _PRSVpl_t { /* polygon for current density viualization */
    double idensity;     /* current density */
    PRSxy_t *xy;        /* points */
    int     nxy;         /* number of points */
    int     layer;      /* layer of element */
} PRSVpl_t;

typedef struct _PRScomp_t { /* buffer for components */
    PRSoid_t oid;        /* object identifier */
    CompType type;      /* GEO, CAL, TEX, NET, BAG, SCL, PRT, CPN , VPL */
    union {
        PRSgeo_t  g;
        PRSinst_t i;
        PRStext_t t;
        PRSnet_t  n;
        PRSbag_t  b;
        PRSsinst_t s;
        PRSGinst_t gi;
        PRSpin_t  pin;
        PRSport_t port;
        PRSVpl_t  v;
    } body;
} PRScomp_t;

typedef struct _PRSprp_t { /* property */
    PRSname_t name;      /* name of property */
    PRSxy_t pos;        /* where to draw the property value */
    int prot, vis;      /* protection and visibility */
    PropTyp type;      /* property type, SPROP ->sval, IPROP ->ival, RPROP ->rval */
    union {
        Const char *sval;
        int ival;
        double rval;
    } u;
} PRSprp_t;

typedef struct { /* equivalent element mapped for parallel and/or serial conn
    PRSname_t ename, etype;
    float value;
    long n1, n2, nb;
} PRSequiEl_t;

```

```

typedef struct {
    int n;
    PRSequiEl_t *eqi;
} PRSequi_t;

typedef struct { /* compatibility to older PARIS Versions */
    gu_point_t p;
    char d;
} PRStextPos_t;

typedef struct { /* compatibility to older PARIS Versions */
    PRSname_t name;
    long node;
    int n;
    PRStextPos_t *pts;
} PRSnNo_t;

typedef struct { /* compatibility to older PARIS Versions */
    int n;
    PRSnNo_t *nn;
} PRSnn_t;

typedef struct _PRSdef_t { /* a cell definition */
    PRSname_t name; /* name of cell */
    PRSbox_t area; /* bounding box */
    long ns; /* number of snappoints, only for gate arrays */
    long nc; /* number of components */
    time_t dtc, dtm; /* creation/modification date */
    CellFunc func; /* cell function */
    PRSsnap_t *snp; /* array of snappoints */
    char plc; /* place sign */
    unsigned short plf; /* bit set of place flags, determines which instance t
    Boolean_t checkFlag, netFlag, subnetFlag, snetFlag; /* validity of net infor
    Boolean_t visFlag; /* visibility of current density vis objects */
    PRSequi_t equir, equic; /* list of equivalent elements */
    PRSnn_t namedNodes; /* compatibility to older PARIS Versions */
    PRSdescend_t descend; /* descend Information, serves as template, when an in
} PRSdef_t;

typedef struct _PRSheader_t {
    int rev; /* revision number of PARIS file */
    long resolution;
    PRSname_t technology;
} PRSheader_t;

typedef struct _PRSmpa_t { /* compatibility to older PARIS Versions */
    Const char *name; /* this information is handled now by properties in a mor
    CellFunc type;
    union {
        float value;
    } body;
} PRSmpa_t;

```

```

typedef struct _PRSdata_t *PRSdata_t;

typedef void (*PrsCompActionT)ARG_DECL3(PRScomp_t*, c, void*, action_data, PRSdef_t*);
typedef void (*PrsPropActionT)ARG_DECL2(PRSprop_t*, c, void*, action_data);
typedef void (*PrsDefActionT)ARG_DECL2(PRSdef_t*, d, void*, action_data);
typedef void (*PrsLibActionT)ARG_DECL2(PRSheader_t*, d, void*, action_data);
typedef void (*PrsMpaActionT)ARG_DECL2(PRSmpa_t*, d, void*, action_data);

#ifdef __cplusplus
extern "C" {
#endif

/***** CREATE/DELETE *****/

/* Create a PRS interface instance and set the verbose level to the
   specified value. The verbose level means:
       0: issue only error messages
       1: issue protocol and warning messages
   The pointer to the instance data record is returned.
*/
PRSdata_t prs_create_interface ARG_DECL1(
    int,    verbose    /* verbose level */
);

/* deletes the specified PRS interface instance */
void prs_delete ARG_DECL1(
    PRSdata_t, d    /* pointer to PRS interface instance data record */
);

/***** INPUT *****/

/* Open a PRS file for reading or writing. The FILE pointer of
   the open file is returned. The PRS interface instance data record
   must have been created previously by prs_create_interface or
   prs_create_interface_conv. The functions returns NULL on
   open failure. If pathname is null or the null string, the
   function return stdin or stdout dependent on open mode.
*/
FILE *prs_fopen ARG_DECL6(
    PRSdata_t,    d,    /* pointer to PRS interface instance data record */
    Const char*, pathname, /* base of pathname of file to be opened */
    int,    lp,    /* length of pathname, must be specified, a termina
    Const char*, extension, /* extension of pathname */
    int,    le,    /* length of extension */
    short, open_mode    /* one of prs_READ, prs_WRITE */
);

/* Open a PRS stream on open file descriptor. The FILE pointer of
   the open file is returned. The PRS interface instance data record
   must have been created previously by prs_create_interface or

```

```

    prs_create_interface_conv. The functions returns NULL on
    open failure.
*/
FILE *prs_fdopen ARG_DECL3(
    PRSdata_t, ths,
    int, fd,
    short, mode
);

/* Assign the file pointer f to d to be used in subsequent
 * input/output operations. Return true on success, or false if a file
 * was already open for d. */
Boolean_t prs_setfile ARG_DECL((PRSdata_t d, FILE* f));

/* Close a PRS file previously opened by prs_fopen */
void prs_fclose ARG_DECL1(
    PRSdata_t, d /* pointer to PRS interface instance data record */
);

/* Read a compList. The file to read from must be opened previously
 * either by calling prs_fopen or assigned by prs_setfile. The action
 * function pointer may be NULL. In this case no action is performed.
 * Returns true on success or false on error (open error for file or
 * syntax error in compList file. */
Boolean_t prs_read_complist ARG_DECL((
    PRSdata_t d, /* pointer to PRS interface instance data re
    PrsCompActionT compAction, /* function invoked for each recognized comp
    PrsPropActionT propAction, /* function invoked for each recognized prop
    PrsLibActionT libAction, /* function invoked for file header */
    void* actionData /* passed to *Action as 2nd argument */
));

/* Read a compList from file pathname and invoke user definable
 * actions for comps. The action function pointer may be NULL. In
 * this case no action is performed. Returns true on success or
 * false on error (open error for file or syntax error in compList
 * file. */
Boolean_t prs_pread_complist ARG_DECL7(
    PRSdata_t, d, /* pointer to PRS interface instance data re
    char*, pathname, /* pathname of file to be opened */
    int, lp, /* length of pathname, must be specified, a
    PrsCompActionT, compAction, /* function invoked for each recognized comp
    PrsPropActionT, propAction, /* function invoked for each recognized prop
    PrsLibActionT, libAction, /* function invoked for file header */
    void*, actionData /* passed to *Action as 2nd argument */
);

/* Read a compList from stream fp and invoke user definable
 * actions for comps. The action function
 * pointer may be NULL. In this case no action is performed.

```

```

Returns true on success or false on error (open error for file or
syntax error in compList file.
*/
Boolean_t prs_fread_complist ARG_DECL6(
    PRSdata_t,                d, /* pointer to PRS interface instance data re
    FILE*,                    fp, /* pointer to stream */
    PrsCompActionT,          compAction, /* function invoked for each recognized comp
    PrsPropActionT,          propAction, /* function invoked for each recognized prop
    PrsLibActionT,           libAction, /* function invoked for file header */
    void*,                   actionData /* passed to *Action as 2nd argument */
);

/* Read a PRS file from file pathname and invoke user definable
actions for comps and defs. The action function
pointer may be NULL. In this case no action is performed.
Returns true on success or false on error (open error for file or
syntax error in PRS file.
*/
Boolean_t prs_read_file ARG_DECL8(
    PRSdata_t,                d, /* pointer to PRS interface instance data re
    Const char*,              pathname, /* pathname of file to be opened */
    int,                      lp, /* length of pathname, must be specified, a
    PrsCompActionT,          compAction, /* function invoked for each recognized comp
    PrsDefActionT,           defAction, /* function invoked for each recognized def
    PrsPropActionT,          propAction, /* function invoked for each recognized prop
    PrsLibActionT,           libAction, /* function invoked for file header */
    void*,                   actionData /* passed to *Action as 2nd argument */
);

/* Read a PRS file from stream fp and invoke user definable actions
for comps and defs. The action function pointer
may be NULL. In this case no action is performed. Returns true on
success or false on error (open error for file or syntax error in
PRS file.
*/
Boolean_t prs_read_stream ARG_DECL7(
    PRSdata_t,                d, /* pointer to PRS interface instance data re
    FILE*,                    fp, /* pointer to stream */
    PrsCompActionT,          compAction, /* function invoked for each recognized comp
    PrsDefActionT,           defAction, /* function invoked for each recognized def
    PrsPropActionT,          propAction, /* function invoked for each recognized prop
    PrsLibActionT,           libAction, /* function invoked for file header */
    void*,                   actionData /* passed to *Action as 2nd argument */
);

/* Read a PRS file from the internal stream of df and invoke user
definable actions for comps and defs. The action function pointer
may be NULL. In this case no action is performed. Returns true on
success or false on error (open error for file or syntax error in
PRS file. */
Boolean_t prs_read_data ARG_DECL6(
    PRSdata_t,                d, /* pointer to PRS interface instance data re

```

```

    PrsCompActionT,          compAction, /* function invoked for each recognized comp
    PrsDefActionT,          defAction, /* function invoked for each recognized def
    PrsPropActionT,        propAction, /* function invoked for each recognized prop
    PrsLibActionT,         libAction, /* function invoked for file header */
    void*,                  actionData /* passed to *Action as 2nd argument */
);

/* Read an MPA file from file pathname and invoke user definable
actions for defs. The action function
pointer may be NULL. In this case no action is performed.
Returns true on success or false on error (open error for file or
syntax error in MPA file.
*/
Boolean_t prs_read_mpa ARG_DECL5(
    PRSdata_t,              d, /* pointer to PRS interface instance data re
    Const char*,            pathname, /* pathname of file to be opened */
    int,                    lp, /* length of pathname, must be specified, a
    PrsMpaActionT,          defAction, /* function invoked for each recognized def
    void*,                  defActionData /* passed to defAction as 2nd argument */
);

/* return the full pathname got from a prs_fopen call */

char* prs_pathname ARG_DECL1(
    PRSdata_t, ths
);

/* return write protection from an open prs file */

int prs_protection ARG_DECL1(
    PRSdata_t, ths
);

/* return revision from an open prs file */

int prs_revision ARG_DECL1(
    PRSdata_t, ths
);

/***** OUTPUT *****/

/* write revision nu,ber to prs file */
void prs_write_rev ARG_DECL((
    PRSdata_t ths, /* pointer to PRS interface instance data record */
    Const int rev /* revision number */
));

/* write file header to prs file */
void prs_write_header ARG_DECL((
    PRSdata_t ths, /* pointer to PRS interface instance data record */
    PRSheader_t* h /* pointer to header data record */
));

```

```

/* Write a macro definition(cell) header to PRS file. */
void prs_write_def_header ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSdef_t*, def /* pointer to macro definition data record */
);

/* Write a GeoElement to PRS file. */
void prs_write_geo ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSgeo_t*, g /* pointer to geo data record */
);

/* Write an instance to PRS file. */
void prs_write_inst ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSinst_t*, i /* pointer to inst data record */
);

/* Write a shaped instance to PRS file. */
void prs_write_sinst ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSsinst_t*, i /* pointer to sinst data record */
);

/* Write a generic instance to PRS file. */
void prs_write_ginst ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSginst_t*, i /* pointer to ginst data record */
);

/* Write an actnet to PRS file. */
void prs_write_actnet ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSactnet_t*, i /* pointer to actnet data record */
);

/* Write an port to PRS file. */
void prs_write_port ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSport_t*, i /* pointer to port data record */
);

/* Write a pin to PRS file. */
void prs_write_pin ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSpin_t*, i /* pointer to port data record */
);

/* Write a text element to PRS file. */
void prs_write_text ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRStext_t*, t /* pointer to text data record */
);

```

```

);

/* Write a net element to PRS file. */
void prs_write_net ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSnet_t*, n /* pointer to net data record */
);

/* Write a bag element to PRS file. */
void prs_write_bag ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSbag_t*, b /* pointer to bag data record */
);

/* Write a bvpl element to PRS file. */
void prs_write_vpl ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSvpl_t*, v /* pointer to vpl data record */
);

/* Write an element to PRS file. */
void prs_write_comp ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRScomp_t*, c /* pointer to comp data record */
);

/* Write property to prs file */
void prs_write_prop ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSprop_t*, p /* pointer to prop */
);

/* Write line to mpa file */
void prs_write_mpa ARG_DECL2(
    PRSdata_t, ths, /* pointer to PRS interface instance data record */
    PRSmpa_t*, t /* pointer to parameter data record */
);

/*****/

CompType prs_get_ctyp ARG_DECL1(char*, ostr);
gu_Ortho_t prs_get_ortho ARG_DECL1(char*, ostr);
GeoType prs_get_gtyp ARG_DECL1(char*, ostr);
GeoFunc prs_get_gfunc ARG_DECL1(char*, ostr);
PinType prs_get_pintyp ARG_DECL1(char*, ostr);
gu_PathType_t prs_get_patyp ARG_DECL1(char*, ostr);
int prs_get_designtyp ARG_DECL1(char*, ostr);
CellFunc prs_get_cellfunc ARG_DECL1(char*, ostr);
int prs_get_propertyprot ARG_DECL1(char*, ostr);
int prs_get_protection ARG_DECL1(char*, ostr);

char* prs_ctyp_str ARG_DECL1(CompType, i);

```

```

char* prs_ortho_str ARG_DECL1(gu_Ortho_t, i);
char* prs_gtyp_str ARG_DECL1(GeoType, i);
char* prs_gfunc_str ARG_DECL1(GeoFunc, i);
char* prs_pintyp_str ARG_DECL1(PinType, i);
char* prs_patyp_str ARG_DECL1(gu_PathType_t, i);
char* prs_designtyp_str ARG_DECL1(int, i);
char* prs_cellfunc_str ARG_DECL1(CellFunc, i);
char* prs_protection_str ARG_DECL1(int, i);

/***** SERVICE ROUTINES *****/

/* free allocated memory */
void prs_free ARG_DECL1(
    void*, p          /* pointer to memory */
);

/*****
function: prs_init_xyz
purpose:  Provides initialization for different buffers.
params:  PRSxyz_t *p -- pointer to the buffer to be initialized
created:  17.08.95 VMzB
*****/

#define prs_init_unused(x)
#define prs_init_layer(p) (*p) = 0
#define prs_init_name(p) (*p) = ' '
#define prs_init_xy(p) ((p)->x = (p)->y = 0)
#define prs_init_box(p) ((p)->x1 = (p)->y1 = (p)->x2 = (p)->y2 = 0)
void prs_init_nr ARG_DECL1(PRSnr_t *, p);
void prs_init_snap ARG_DECL1(PRSsnap_t *, p);
void prs_init_descend ARG_DECL1(PRSdescend_t *, p);
void prs_init_geoFuncDesc ARG_DECL1(PRSgeoFuncDesc_t *, p);
void prs_init_polygon ARG_DECL1(PRSpolygon_t *, p);
void prs_init_path ARG_DECL1(PRSpath_t *, p);
void prs_init_rect ARG_DECL1(PRSrect_t *, p);
void prs_init_nrect ARG_DECL1(PRSnrect_t *, p);
void prs_init_geo ARG_DECL1(PRSgeo_t *, p);
void prs_init_actnet ARG_DECL1(PRSactnet_t *, p);
void prs_init_inst ARG_DECL1(PRSinst_t *, p);
void prs_init_text ARG_DECL1(PRSstext_t *, p);
void prs_init_net ARG_DECL1(PRSnet_t *, p);
void prs_init_port ARG_DECL1(PRSport_t *, p);
void prs_init_pin ARG_DECL1(PRSpin_t *, p);
void prs_init_bag ARG_DECL1(PRSbag_t *, p);
void prs_init_sinst ARG_DECL1(PRSsinst_t *, p);
void prs_init_comp ARG_DECL1(PRScomp_t *, p);
void prs_init_prop ARG_DECL1(PRSprop_t *, p);
void prs_init_vpl ARG_DECL1((PRSVpl_t*));
void prs_init_def ARG_DECL1(PRSdef_t *, p);
void prs_init_header ARG_DECL1(PRSheader_t*, h);

```

```

#ifdef __cplusplus
}
#endif

#endif
#endif

```

PARIS FORMAT (REV 2.06)

```

paris_file: REV major_rev minor_rev resolution technology
optional_def_list

```

```

optional_def_list:
| def_list

```

```

def_list: def
| def_list def

```

```

def: def_header
optional_equi_list optional_equi_list
NODNAM '0'
optional_property_list optional_comp_list

```

```

optional_equi_list:
| equi_list

```

```

equi_list: equi
| equi_list equi

```

```

equi: STRING STRING REAL INT INT optional_int

```

```

optional_int:
| INT

```

```

optional_property_list:
} property_list

```

```

property_list: property
| property_list property

```

```

optional_component_list:
| component_list

```

```

component_list: component
| component_list component

```

```

property: PRP NAME point INT INT prop_value

```

```

prop_value: 'I' INT
| 'R' REAL
| 'S' STRING

```

point: INT INT

component: geo

- | inst
- | sinst
- | ginst
- | text
- | net
- | bag
- | port
- | pin
- | visobj

def_header: DEF STRING

BOOLEAN BOOLEAN BOOLEAN BOOLEAN BOOLEAN BOOLEAN BOOLEAN BOOLEAN BOOLEAN
CHAR box INT INT BOOLEAN BOOLEAN BOOLEAN BOOLEAN TIME TIME CELL_FUNC
descend

geo: GEO oid

- layer node
- geo_body gfunc
- protection

gfunc: GFUNC optional_pin_desc

optional_pin_desc:

- | STRING PIN_TYPE TEXT_DIRECTION point

geo_body: path

- | polygon
- | rectangle
- | n_rectangle

path: PATH INT point_list ORTHO width PATHTYPE

polygon: POLYGON INT point_list ORTHO

rectangle: RECTANGLE point point point point ORTHO

n_rectangle: NRECTANGLE INT rectangle_list

rectangle_list: rectangle

- | rectangle_list rectangle

rectangle: box INT

inst: CAL oid

- inst_body

inst_body: mname TEXT_DIRECTION moffset

- iname TEXT_DIRECTION ioffset

```

translation transformation
descend
protection

sinst: SCL oid
inst_body optional_shape_list

optional_shape_list:
| SHAPES { shape_list } selected_shape

ginst: GCL oid
inst_body
optional_shape_list
generator_name generator_args

text: TEX oid
point TEXT_DIRECTION text_size layer
text_string
protection

bag: BAG oid
bag_name { bag_contents }

net: NET oid
net_name { bag_contents }

port: PRT oid
port_name node PIN_TYPE

pin: CPN oid
pin_name node protection

visobj: VPL oid
layer idensity point_list

bag_contents:
| oid_list

oid_list: INT
| oid_list INT

shape_list
: shape
| shape_list shape

shape: quality width height routing_space
optional_parameter_list optional_pin_list

routing_space
: north west south east

optional_parameter_list

```

```

:
| parameter_list

parameter_list
: parameter
| parameter_list parameter

optional_pin_list
:
| { pin_list }

pin_list: pin_def
| pin_list pin_def

pin_def : pin_name pin_layer pin_box

pin_box : xlower ylower xupper yupper

north : INT
west : INT
sourth : INT
east : INT
quality : INT
parameter: DOUBLE
pin_name : STRING
pin_layer: INT
xlower : INT
ylower : INT
xupper : INT
yupper : INT
layer : INT
identity : DOUBLE

point: INT INT

box: point point

descend: BOOLEAN layer_set

layer_set: INT optional_int_list

optional_int_list:
| int_list

int_list: INT
| int_list INT

transformation: [0-7]
| 8 transformation_matrix

transformation_matrix: DOUBLE DOUBLE DOUBLE DOUBLE

```

selected_shape: INT

generator_name: STRING

generator_args:

| generator_args arg_name = value

KOMPILIEREN und LINKEN

#include <prsio.h>

\$(CC) -I\$(AEGINCLUDE)

\$(CC) -o myprog mymodules mylibraries -L\$(AEGLIB)/\$(PLATFORM) -lparis -lbase -lm

WEITERES

SAVE(paris), LOAD(paris)

NAME

PrsRpcClnt - RPC client basic functionality

SYNOPSIS

```
class PrsRpcClnt {
public:
    CLIENT *clnt;
    SockAddrIn sin;
    TimeVal timeout;
    int sock;

    operator void*() { return (void*)clnt; }
    int operator!() { return !clnt; }
    int operator()(); // NULL_PROC to check the connection
    int Call(u_long procnum, XdrProc_t inproc, void* in,
            XdrProc_t outproc, void* out);

    void Create();
    void Destroy();
    PrsRpcClnt(const char* hostname);
    ~PrsRpcClnt();
};

class PrsRpcObj {
protected:
    PrsRpcClnt *clnt; // PrsRpc client handle
    PrsRpcId id;      // PrsRpc id of object, assigned by server
public:
    PrsRpcId Id() const;
    int operator()(); // NULL_PROC to check the connection
    PrsRpcObj(PrsRpcClnt* cl, PrsRpcId i=0);
    PrsRpcObj(const PrsRpcObj* s, PrsRpcId i=0);
    PrsRpcObj(const PrsRpcObj& s);
    ~PrsRpcObj(){}
};
```

DESCRIPTION

The classes `PrsRpcClnt` and `PrsRpcObj` provide the basic functionality needed to run the RPC interface of PARIS. `PrsRpcClnt` provides the functionality that is common to all other RPC functions. It maintains the RPC client connection to the server and the invocation of remote function calls.

`PrsRpcObj` is inherited by all objects, that are addressed as remote objects, either remote cell definitions or remote layout components. `PrsRpcObj` maintains the low level RPC client handle and an identifier, which is used as a unique identifier of the object common to the server and the client. Thus the rpc object identifier functions as a kind of object pointer that is known by the server and the client. The rpc object id is assigned to the object by the server, when the object is addressed by the client. It must not be changed by the client.

Most of the functionality of both classes is used internally in the RPC interface. The functions and members are public or protected, because they are addressed by other classes of the RPC interface. There is no need for a client to use the internal functionality directly. There is only one exception to this rule concerning the creation of the RPC connection, which is performed in the

constructor of `PrsRpcClnt`. An instance of `PrsRpcClnt` must be created before any other parts of the RPC interface can be used. The constructor of `PrsRpcClnt` that takes the hostname argument, establishes the connection to the PARIS server. The pointer of this instance must be passed to all other constructors creating an instance of a class that represents a remote PARIS object. The `PrsRpcClnt` object can be checked, whether the connection could be successfully created by `operator void*()` or `operator!()`.

In the following assume that

- `clnt` is a `PrsRpcClnt`
- `connection` is a character string
- `obj` is a `PrsRpcObj` object

Construction

```
PrsRpcClnt clnt(connection)
```

The constructor tries to establish a connection to the PARIS server listening on `hostname`. The `connection` argument specifies the hostname and the port number, the server is listening on. The syntax of `connection` is: `[hostname]:port`. The hostname part is optional. If no host name is specified, the default host name `localhost` is assumed. The port number must always be specified, because a PARIS server does not register itself with the portmapper. Thus there is no way to find a PARIS server automatically. This behaviour is intended in order to allow for multiple servers running on the same host. The hostname part can be any valid internet host name, either as a name or as an internet address in dot notation.

Operators

```
clnt(), obj()
```

The function call operator of the client handle or a remote object calls the `NULL` procedure of the RPC interface. The `NULL` procedure does nothing except to check, whether the server is still alive. The function returns zero on success and nonzero on failure

```
(void*)clnt, if (clnt) ...
```

If the client creation was successful, `operator void*` returns the RPC client handle, otherwise the `NULL` pointer.

```
!clnt, if (!clnt) ...
```

If the client creation was not successful, `operator!` returns true, otherwise false.

The operators methods `operator void*` and `operator!` can be used to check the success of the client creation. Using the operators, the client appears like a boolean variable and can thus be used directly in conditional expressions.

NAME

PrsRpcMacroDef - remote cell definition

SYNOPSIS

```
class PrsRpcMacroDef : public MgenDef_t, public PrsRpcObj {
public:
    void Append(MgenObj_t*);
    void Remove(MgenObj_t*);

    void Clear();
    void Get();
    void GetRec();
    void Put();
    void PutRec();

    PrsRpcMacroDef(const char*, PrsRpcClnt* clnt);
    PrsRpcMacroDef(const MgenDef_t&, PrsRpcClnt* clnt);
    ~PrsRpcMacroDef();
};
```

DESCRIPTION

The class `PrsRpcMacroDef` handles a cell definition that comes from a remote PARIS application. `PrsRpcMacroDef` is used in the client part, whereas the remote PARIS programs is the server part. `PrsRpcMacroDef` is derived from `MgenDef_t`, so a remote cell definition can be treated like an MGEN cell. The additional functionality of a `PrsRpcMacroDef` addresses the functionality for data transfer between the MGEN cell on the client side and the PARIS cell on the server side. The `Append` and `Remove` methods of `PrsRpcMacroDef` do not only add/remove an object to/from the MGEN cell of the client, but also perform the corresponding operation on the server side.

In the following assume that

- d is a `PrsRpcMacroDef`
- def is an `MgenDef`
- dname is a character string
- rpc_clnt_handle is a `PrsRpcClnt` handle
- c is an

Construction

`PrsRpcMacroDef d(dname, rpc_clnt_handle)`

Creates an MGEN cell `d` bound to the remote PARIS cell named `dname`. If the remote cell does not exist, it is created and added to the last macro group that is loaded within PARIS. The local cell is initially empty. The client must call `Get` or `GetRec` to initiate the data transfer from the server to the client. The argument `rpc_clnt_handle` is the client handle to identify the the RPC connection to the PARIS server process. The client handle is created by the class `PrsRpcClnt`.

`PrsRpcMacroDef d(def, rpc_clnt_handle)`

Creates an MGEN cell `d` bound to the remote PARIS cell of the same name as the local MGEN cell `def`. If the remote cell does not exist, it is created and added to the last macro group that is loaded within PARIS. The contents of the local cell is copied from `def`. The client must call `Get` or `GetRec` to initiate the data transfer from the server to the client. The argument `rpc_clnt_handle` is the client handle to identify the the RPC connection to the PARIS server process. The client handle is created by the class

PrsRpcClnt.

Caution: the new cell is added to the global `def_list` in the base class constructor of `MgenDef_t`.

Modification

`d.Append(c)`

Appends the object `c` to the local MGEN cell and to the remote PARIS cell.

`d.Remove(c)`

Removes the object `c` from the local MGEN cell and to the remote PARIS cell. The key to identify the object is its object id. If there is no object with the same object id as `c` in the remote cell, no remote object is removed.

`d.Clear()`

Clears the local and the remote cell.

`d.Get()`

Transfers the contents of the remote cell to the local cell. The original contents of the local cell is deleted.

`d.GetRec()`

Transfers the contents of the remote cell to the local cell recursively. The complete hierarchy tree of the remote cell is transferred including all the cells that are instantiated in the remote cell. `GetRec` performs a merge depending on the cell's modification time of the transferred data. A cell got from PARIS that is younger than its local pendant, replaces the local one. If the remote cell has the same modification time or is younger than the local one, the local cell remains unchanged. Not existing local cells are created. All cells are appended to the global `def_list`.

`d.Put()`

Transfers the contents of the local cell to the remote cell. The original contents of the remote cell is deleted.

`d.PutRec()`

Transfers the contents of the local cell to the remote cell recursively including the complete hierarchy tree of `d`. The original contents of the remote top level cell is deleted. Other remote cells involved are not replaced. Not existing remote cells are created.

SEE ALSO

`PrsRpcEditDef(3p)`, `PrsRpcClnt(3p)`, `MgenPrsRpcDB(3)`

NAME

PrsRpcEditDef - remote edited cell definition

SYNOPSIS

```
struct PrsRpcCmdResult {
    String_t str;
    int status;
    PrsRpcCmdResult(const String_t& s, int st);
    ~PrsRpcCmdResult();
};

class PrsRpcEditDef : public PrsRpcMacroDef {
public:
    void SelectInst(const char* iname);
    void SelectNet(const char* netname);
    void SelectPin(const char* iname, const char* pinname);
    void UnselectInst(const char* iname);
    void UnselectNet(const char* netname);
    void UnselectPin(const char* iname, const char* pinname);
    void Unselect();
    void GetSelected(PrsRpcCompList&);

    int Execute(const char*);
    PrsRpcCmdResult ExecuteTcl(const char* command);

    void Open();
    void Close();
    MgenBox Window() const;

    PrsRpcEditDef(const char*, PrsRpcClnt* clnt);
    PrsRpcEditDef(const MgenDef_t&, PrsRpcClnt* clnt);
    ~PrsRpcEditDef();
};
```

DESCRIPTION

The class `PrsRpcEditDef` handles an edited cell definition that comes from a remote PARIS application. `PrsRpcEditDef` is used in the client part, whereas the remote PARIS programs is the server part. `PrsRpcEdit` is derived from `PrsRpcMacroDef`, so a remote edited cell definition behaves like a remote cell and an MGEN cell. The additional functionality of a `PrsRpcEditDef` compared to `PrsRpcMacroDef` addresses the functionality of object selection, obtaining selections, opening and closing edit windows and remote command execution.

The virtual functions `Append` and `Remove` inherited from `PrsRpcMacroDef` also have their virtual pendants within PARIS. So an object `Append` or `Remove` to or from a remote edited cell is immediately visible in the PARIS edit window.

In the following assume that

```
- d          is a PrsRpcEditDef
- def       is an MgenDef
- dname     is a character string
- inst_name, net_name, pin_name, command are character strings
```

- rpc_clnt_handle is a PrsRpcClnt handle
- boxR is an MGenBox PrsRpcCmdResult
- res

Construction, Destruction

PrsRpcEditDef d(dname, rpc_clnt_handle)

The constructor creates a connection to a remote edited cell in PARIS. If the cell dname is not edited in PARIS, dname is opened for editing. If dname is already edited, only the rpc connection bound to the remote cell dname is obtained and no additional edit window on dname is opened. The local cell dname is initially empty. The client is responsible for data transfer. For an explanation of the rpc_clnt_handle argument, see PrsRpcMacroDef. This function is equivalent to the PARIS command `changedef dname` with the exception, that it does not open an additional edit window, if there is already an open edit window for the cell.

PrsRpcEditDef d(def, rpc_clnt_handle)

The constructor creates a connection to a remote edited cell in PARIS. If the cell with the same name asdef is not edited in PARIS, an edit window for def is opened. If def is already edited, only the rpc connection bound to the remote cell def is obtained and no additional edit window on def is opened. The contents of the local cell d is copied from the MGEN cell def. The client is responsible for data transfer to the server. At construction no data is transferred to the server. For an explanation of the rpc_clnt_handle argument, see PrsRpcMacroDef. This function is equivalent to the PARIS command `changedef dname` with the exception, that it does not open an additional edit window, if there is already an open edit window for the cell.

Caution: the new cell is added to the global `def_list` in the base class constructor of `MgenDef_t`.

d.Close()

Close closes the edit window of d. The action is the same as the PARIS command `defclose`. The edited cell is closed in any case despite of the occurrence of check or netext errors during close. First Close tries to do a `defclose` including check and netext. If the verification was not successful, Close performs the equivalent action to `defclose -noverify`.

d.Open()

Open opens an edit window for d in PARIS. Open is called from the constructors if `PrsRpcEditDef`. If an edit window is already open, Open does nothing.

~d() The destructor deletes the local cell. It does not imply a Close of the remote cell.

Remote Object Selection

d.SelectInst(inst_name)

Selects the instance `inst_name` in the remote cell. This action is equivalent to the PARIS command `find -pname inst_name`.

d.SelectNet(net_name)

Selects the net `net_name` in the remote cell. This action is equivalent to the PARIS command `find -net net_name`.

d.SelectPin(inst_name, pin_name)

Selects the pin `pin_name` of the instance `inst_name` in the remote cell.

- d.UnselectInst(inst_name)
Unselects the instance inst_name in the remote cell.
- d.UnselectNet(net_name)
Unselects the net net_name in the remote cell.
- d.UnselectPin(inst_name, pin_name)
Unselects the pin pin_name of the instance inst_name in the remote cell.
- d.Unselect()
Unselects all selected elements in the remote cell. This is equivalent to the command unselect -all.
- d.GetSelected(rpc_comp_list)
Obtains the object ids of the selected elements of the remote cells. The result is a list of PrsRpcComps. A PrsRpcComp is a handle for both the remote and the local object. It can be used to modify the object's properties. GetSelected first searches the cell d for objects of the object ids retrieved. If the objects are found, their pointers are associated with the RPC objects in the object list. If the objects can't be found in d, the objects are retrieved from the remote PARIS server.

Inquiry

- box = d.Window()
Window returns the current view window of the edited cell d. If d does not have a corresponding edited cell in PARIS, an undefined box is returned.

Remote Command Execution

- d.Execute(command)
Execute the PARIS command command in the PARIS window open to edit the remote cell d. Remote command execution can only be performed in conjunction with an edited cell, because the effect of a command depends on the environment the command was executed in. Within PARIS, a command's environment is determined by the window, in which the event occurred, that triggered the command. Unlike a local PARIS command, an association of a window to a remote command is not possible. So the client must pass the target object for the remote command. Execute returns the return value of the last command executed in PARIS.
- res = d.ExecuteTcl(command)
Executes a PARIS command like Execute. Unlike Execute ExecuteTcl does not only return the command's return status, but also returns the command's standard output in the result string. Thus, ExecuteTcl behaves like a usual Tcl command.

SEE ALSO

PrsRpcMacroDef(3p), PrsRpcComp(3p)

NAME

PrsRpcComp - remote layout component

SYNOPSIS

```
class PrsRpcComp : public PrsRpcObj {
public:
    MgenObj_t* Get();
    operator MgenObj_t*();
    MgenOid Oid() const;
    MgenProp PropGet(const char* prop_name);
    void PropAdd(MgenProp);
    void PropRem(const char*);
    void PropModify(const char*, char*);
    void PropModify(const char*, int);
    void PropModify(const char*, double);

    PrsRpcComp(PrsRpcClnt* cl, PrsRpcId oid, PrsRpcId did, MgenObj_t* obj);
    PrsRpcComp(PrsRpcMacroDef*, MgenObj_t* obj);
};

class gllink_t(PrsRpcComp) : public olink_t, public PrsRpcComp {
public:
    gllink_t(PrsRpcComp)(PrsRpcClnt* cl, PrsRpcId oid,
                        PrsRpcId did, MgenObj_t* obj);
    gllink_t(PrsRpcComp)(PrsRpcMacroDef*, MgenObj_t* obj);
};

declare(gllist_t, PrsRpcComp)
typedef gllist_t(PrsRpcComp) PrsRpcCompList;
```

DESCRIPTION

The class `PrsRpcCom` is a handle class for layout objects. It can address both, the local object of type `MgenObj_t` and its remote pendant in the PARIS server. An object of type `PrsRpcCom` holds internally the information to address the object, but not the object itself. The local object is addressed through its `MgenObj_t` pointer, the remote object is addressed by its object id and the rpc id of the cell it is contained in.

Currently `PrsRpcCom` provides the functionality to inquire and change properties of the local and remote object. Property changes appear in the client's and in the server's data base immediately. Other (gemetric) modifications of the object can be done by changing the local object directly and a subsequent explicit data transfer of the client data to the server.

Remote component objects are created in the method `PrsRpcEditDef::GetSelected`, which return the handles of the selected objects of an edited cell in the server. This is the usual way to obtain remote object handles. In normal operation remote components are not created directly by a client.

In the following assume that

```
- clnt      is a PrsRpcClnt
- oid       is an object id
- defid     is an rpc id
```

- def is a pointer to an PrsRpcMacroDef
- rc is a PrsRpcComp
- prop_name, str are character strings
- prop is an MgenProp
- c is an MgenObj pointer
- i is an integer
- d is a floating point value

Construction

PrsRpcComp rc(clnt, oid, defid, c)

Creates a remote component handle for a layout component. The component's object id is `oid`, it is contained in the cell with the rpc id `defid` and its local pointer is `c`. The remote connection is maintained by the client handle `rpc_clnt_handle`. This constructor is called internally in the method `PrsRpcEditDef::GetSelected`.

PrsRpcComp rc(def, c)

Creates a remote component handle for a layout component `c`. The component is assumed to be contained in the remote cell `def`. The client handle and the remote cell id is obtained from `def`. The object's rpc id is its object id obtained from `c`.

Property Modification

rc.PropAdd(prop)

Add the property `prop` to the local and the remote object identified by `rc`.

rc.PropRem(prop_name)

Remove the property named `prop_name` from the local and the remote object identified by `rc`.

rc.PropModify(prop_name, str)

Change the property value of the property named `prop_name` to the string `str`. This command affects the local and the remote object.

rc.PropModify(prop_name, i)

Change the property value of the property named `prop_name` to the integer value `i`. This command affects the local and the remote object.

rc.PropModify(prop_name, d)

Change the property value of the property named `prop_name` to the floating point value `d`. This command affects the local and the remote object.

Inquiry

oid = rc.Oid() F

Return the object id of `rc`.

c = (MgenObj_t*)rc

The cast operator `operator MgenObj_t*()` returns the pointer of the MGEN object associated with the RPC object.

prop = rc.PropGet(prop_name)

Reads the property `prop_name` from the remote objects. The property is returned. If no property of the requested name exists, an undefined property is returned. This function always takes the property from the remote object, it does not consider the local object, if any.

Data Transfer

```
c = rc.Get()
```

The `Get` method reads the component with the same object id as the local RPC component from the remote PARIS process. After `Get` has been called, the RPC component has a valid `MgenObj` pointer.

The Remote Component List

Remote component handles are entered on a linked list of type `PrsRpcCompList` by the method `PrsRpcEditDef::GetSelected`. A `PrsRpcCompList` can be used as any other list of type `gllist`. See the manual page of `gllist(3)` for details how to use the `gllist`.

SEE ALSO

`gllist(3)`, `MgenObj(3)`, `MgenProp(3)`, `MgenPropObj(3)`, `PrsRpcClnt(3p)`

NAME

parameter – Technologie und Editor Parameter

SYNTAX

Es gibt im Parameterfile folgend Arten von Information:

Key	ein Schlüsselwort, das immer genau so stehen muß
String	ist eine Zeichenkette
Boolean	ist ein boole'scher Wert der die Zustände F für false oder T für true annehmen kann.
Coord	Ist eine Koordinate in internen Einheiten, also eine Integerzahl.
Point	Ist ein Punkt bestehend aus 2 Coords.
Int	Ist eine Integerzahl.
Float	Ist eine Gleitkommazahl.
LayerSet	ist die Bezeichnung einer Menge von Layern bestehend aus einer Folge von Integer Nummern, Bereichen in der Form a-b oder Layernamen. Die einzelnen Elemente sind durch ein Komma getrennt. Es darf keine Leerzeichen in dem String geben.
LayerList	ist eine andere Bezeichnung einer Menge von Layern. Eine Layerliste beginnt mit einer Zahl n, die die Anzahl der Layer bezeichnet gefolgt von n weiteren Zahlen die die in der Layerliste enthaltenen Layernummern bezeichnen.

In folgenden ist die Syntax des Parameterfiles beschrieben. Schlüsselwörter sind in **Courier-Bold** Schrift, Angaben von Zahlen oder Strings sind in *Courier* Schrift. Es wird jeweils der Name des Elements gefolgt von einem Doppelpunkt und dem Typ des Element angegeben. Kommentare sind in Roman Schrift. Wenn wegen der Länge der in der Syntaxbeschreibung verwendeten Namen nicht alles in eine Zeile paßt, ist an geeigneter Stelle ein \ eingesetzt. Wenn ein \ am Zeilenende steht, bedeutet das kein Zeilenende im tatsächlichen Parameterfile.

PAR:Key

```
design-typ:String resolution:Coord TECH:Keyprocess-name:String
layer-name:String layer-number:Int visibility:Boolean\
  detectability:Boolean path-width:Coord color:Int\
  fill-pattern:Int outline-pattern:Int
layer-name:String 255 visibility:Boolean\
  detectability:Boolean path-width:Coord color:Int\
  fill-pattern:Int outline-pattern:Int

viewPinName:Boolean ViewInstName:Boolean viewCellName:Boolean\
  viewObst:Boolean viewGrid:Boolean viewHierarchy:Boolean
sizeNetName:Coord sizePinName:Coord sizeCellName:Coord
sizeInstName:Coord locInstName:Point
grid:Point gridOffset:Point gridFactor:Int
pathGrid: Point pathGrid:Point
textSize:Coord textDirection:Char inputLayer:Int
inputGeoFunc:GeoFunc inputOrtho:Ortho tolpath:Int tolcell:Int

internalLayer:LayerList
obstructionLayer:LayerList
errorLayer:LayerList
```

Widerstands- und Kapazitätsbelag für die nachfolgenden Masken
noOfMask: Int
maskName: String layerList: LayerList cSqr: Float rSqr: Float
die vorhergehende Zeile noOfMask mal

Routing Layer
noOfRoutingLayer: Int
routingLayer: LayerSet obstructionLayer: LayerSet
die vorhergehende Zeile noOfRoutingLayer mal

Contact Layer
noOfContactLayer: Int
contactLayer: LayerSet routingLayer1: LayerSet routingLayer2: LayerSet
die vorhergehende Zeile noOfContactLayer mal

I2L Generator Daten
i2lGeneratorSwitch: Int
ktboxOffset: Point tboxOffset: Point step: Point
die vorhergehende Zeile nur, wenn i2lGeneratorSwitch nicht 0 ist

BESCHREIBUNG

Der Parameterfile enthält Information zur Beschreibung der Technologie wie z.B. die Definition der Designlayer, das Prozeßraster, sowie Voreinstellungen, die zum Betrieb von PARIS erforderlich sind. Am Anfang steht das Schlüsselwort **PAR**.

Am Anfang des Parameterfiles steht der header beginnend mit dem Schlüsselwort **PAR**. In der zweiten Zeile stehen der Designtyp, die Auflösung und der Name des Prozesses. Als Designtyp gibt es folgende Möglichkeiten:

- GA** Gate Array Design (nur für Version 4, in Version 5 noch nicht unterstützt)
- TA** Linear Array Design, Pfade rasten immer auf das Pfadraster
- BCD** Full Custom Design, der eingebaute I2L Generator ist nicht aktiviert.
- I2L** Bipolar Full Custom Design, der eingebaute I2L Generator ist aktiviert.

Die Prozeßauflösung, *resolution*, bestimmt die Interpretation der internen Einheiten. Eine Koordinate in internen Einheiten *x* wird nach folgender Gleichung in eine Koordinate *y* in μ umgerechnet:

$$y = x/\text{resolution}$$
$$x = y*\text{resolution}$$

Vor dem Prozeßnamen steht immer das Schlüsselwort **TECH**. Zwischen **TECH** und dem Prozeßnamen darf kein Leerzeichen stehen. Der Header eines Parameterfiles sieht dann folgendermaßen aus:

PAR
design-typ resolution **TECH**process-name

Im Anschluß an den Header folgt die Layertabelle. In der Layertabelle sind die Designlayer und deren Attribute definiert. Es gibt maximal 256 Layer in PARIS beginnend mit der Layernummer 0 und endend mit der Layernummer 255. Die Definition des Layers 255 schließt die Layertabelle ab und muß daher immer mit der Definition des Layers 255 enden.

Layer in PARIS sind durch ihre Nummer eindeutig identifiziert. Ein Layer hat einen Namen. Der Layername bezeichnet die Maske, zu der der Layer gehört. Es kann mehrere Layer mit dem selben Namen geben. In einem solchen Fall liegt eine Segmentierung einer Maske des Prozesses in verschiedene Designlayer vor. Der Layername wird immer dann in PARIS verwendet, wenn ein Layerset als Argument eines Kommandos angegeben wird. Steht in einem Layerset ein Layername, werden damit alle Layer mit diesem Namen erfaßt.

Neben dem Namen hat jeder Layer ein Sichtbarkeit- und Editierbarkeits Attribut. Wenn ein Layer sichtbar (`VISIBLE`) ist, werden alle Elemente in diesem Layer gezeichnet, sind aber nicht selektierbar. Wenn ein Layer editierbar (`DETECTABLE`) ist, sind die Elemente in diesem Layer sichtbar und selektierbar. Die Attribute zur Steuerung der Sichtbarkeit und Editierbarkeit sind als boole'sche Werte eingetragen. Ein Wert von `T` steht dabei für `true`, ein Wert von `F` steht für `false`.

Im einzelnen haben die in der Syntaxbeschreibung verwendeten Variablen folgende Bedeutung:

<code>design-typ</code>	der Designtyp, siehe oben.
<code>resolution</code>	die Prozeßauflösung, siehe oben.
<code>process-name</code>	der Name der Technologie. Dieser Name wird verwendet zum Auffinden verschiedener Dateien. Der Dateiname wird dabei konstruiert als <code>\$/AEGDIR/bib/process-name</code> . <code>process-name</code> ist also der Teil des Pfadnamens einer Technologie unterhalb von <code>\$/AEGDIR/bib</code> .
<code>layer-name</code>	der Name eines Layers. Die Semantik von Layernamen wurde oben erläutert.
<code>layer-number</code>	die Nummer eines Layers.
<code>visibility, detectability</code>	die Sichtbarkeit bzw. Editierbarkeit eines Layers.
<code>color, fill-pattern, outline</code>	Farbe, Füllmuster und Linienmuster eines Layers.
<code>viewPinName</code>	gibt an, ob Pinnamen beim Bildaufbau dargestellt werden sollen.
<code>viewInstName</code>	gibt an, ob Instanznamen beim Bildaufbau dargestellt werden sollen.
<code>viewCellName</code>	gibt an, ob Zellanznamen beim Bildaufbau dargestellt werden sollen.
<code>viewObst</code>	dieser Parameter ist obsolet. Er ist nur wegen der Kompatibilität mit Version 4 vorhanden.
<code>viewGrid</code>	gibt an ob das eingestellte Raster gezeichnet werden soll.
<code>viewObst</code>	dieser Parameter ist obsolet. Er ist nur wegen der Kompatibilität mit Version 4 vorhanden. Die descend Attribute an Zellen und Instanzen sind an seine Stelle getreten.
<code>sizeNetName</code>	gibt die Texthöhe an, mit der Netznamen gezeichnet werden.
<code>sizePinName</code>	gibt die Texthöhe an, mit der Pinnamen gezeichnet werden.
<code>sizeCellName</code>	gibt die Texthöhe an, mit der Zellnamen gezeichnet werden.
<code>sizeInstName</code>	gibt die Texthöhe an, mit der Instanznamen gezeichnet werden.
<code>locInstName</code>	gibt die Offset an, wo Instanznamen bei neu erzeugten Instanzen positioniert werden sollen.
<code>grid</code>	ist das Grundraster des Prozesses

<code>gridOffset</code>	ist der Offset des Grundrasters
<code>gridFactor</code>	das Normgrid in PARIS wird berechnet aus <code>grid*gridFactor</code> .
<code>pathGrid</code>	ist das Pfadraster des Prozesses. Bei bestimmten Designtypen (GA, TA) rasten Pfade grunds{tzlich ins Pfadraster. Bei Full Custom Design kann das Pfadgrid als zusätzliches Raster verwendet werden.
<code>pathGridOffset</code>	ist der Offset des Pfadrasters
<code>textSize</code>	die Größe von neu erzeugten Textelementen.
<code>textDirection</code>	die Richtung von neu erzeugten Textelementen.
<code>inputLayer</code>	der Layer von neu erzeugten GeoElementen.
<code>inputGeoFuncLayer</code>	der Geofunktion von neu erzeugten GeoElementen.
<code>inputOrtho</code>	der Orthogonalität von neu erzeugten GeoElementen.
<code>tolPath, tolCell</code>	diese Parameter sind obsolet. Sie sind nur wegen der Kompatibilität mit Version 4 vorhanden.
<code>internalLayer, obstructionLayer</code>	diese Parameter sind obsolet. Sie sind nur wegen der Kompatibilität mit Version 4 vorhanden.
<code>errorLayer</code>	die Menge von Ebenen zur Darstellung von Fehlerflags. Die erste dieser Ebenen wird benutzt, um Fehlerflags in eine Zelle einzufügen.
<code>noOfMaskDefinitions</code>	Die Anzahl der nachfolgenden Maskendefinitionen.
<code>maskName</code>	Name einer Maske. Dieser Name ist frei wählbar. Er dient nur zur Identifizierung eines LayerSets, für das ein Kapazitätsbelag definiert wird.
<code>layerList</code>	Alle Layer, für die der angegebene Kapazitätsbelag gilt.
<code>cSqr</code>	der Kapazitätsbelag der vorher angegebenen Layer.
<code>rSqr</code>	der Widerstandsbelag der vorher angegebenen Layer. Dieser Parameter ist obsolet. Er diente in Version 4 zur Widerstandsberechnung. In Version 4 war die Widerstandsberechnung direkt in PARIS hart kodiert. Diese Vorgehensweise hat sich zu unflexibel erwiesen. Statt dessen wird die Widerstandsberechnung mit Hilfe eines dynamisch gelinketen Moduls durchgeführt, siehe CALCR(PARIS).
<code>noOfRoutingLayer</code>	die Anzahl der nachfolgenden Verdrahtungsebenen für die Schaltungsextraktion. Die maximal erlaubte Anzahl von Verdrahtungsebenen ist 32. Wenn mehr als 32 Ebenen angegeben sind, werden die überzähligen ignoriert.
<code>routingLayer</code>	ein LayerSet, das alle Ebenen einer Verdrahtungsebene angibt.
<code>obstructionLayer</code>	die Menge aller Layer, die für die vorher angegebene Verdrahtungsebene als Obstruktionsebene wirken.

noOfContactLayer die Anzahl der nachfolgenden Kontaktebenen für die Schaltungsextraktion. Die maximal erlaubte Anzahl von Kontaktebenen ist 32. Wenn mehr als 32 Ebenen angegeben sind, werden die überzähligen ignoriert.

contactLayer ein LayerSet, das alle Ebenen einer Kontaktebene angibt.

routingLayer1, routingLayer2 die beiden Verdrahtungsebenen, die von contactLayer verbunden werden. Beide Ebene müssen vorher als Verdrahtungsebene definiert sein.

i2lGeneratorSwitch wenn hier eine Zahl nicht 0 steht, kommen danach die Generatorparameter für den eingebauten I2l Generator.

ktboxOffset der Offset, wo der erste Kontakt eines I2l Gatters in einer KTBOX positioniert werden soll. Eine KTBOX ist eine I2l Gatterstruktur mit einem Injektorkontakt.

tboxOffset der Offset, wo der erste Kontakt eines I2l Gatters in einer TBOX positioniert werden soll. Eine TBOX ist eine I2l Gatterstruktur ohne Injektorkontakt.

step Der Abstand zwischen zwei Kontakten eines I2l Gatters.

BEISPIEL

Das folgende Beispiel zeigt die Datei \$AEGDIR/paris/v5/template.par. Diese Musterdatei ist ein vollständig lauffähiger Parameterfile. Er kann zum Erstellen eigener Parameterfiles verwendet werden.

```

PAR
I2L 100 TECHmy_technology
ALU1      0 T T      0      15      1      1
VIA       1 T T      0      15      1      1
ALU2      2 T T      0      15      1      1
OBSTR     254 T T     0      15      1      1
ERROR     255 T T     0      15      1      1

T T F F T T
600 600 500
600 0 0
50 50 0 0 2
200 200 0 0
1000 H 60
WIRE HALF 49 1000
1 60
1 250
1 255

1
ALU1      2 60 61 0 0.05

2
ALU1 OBSTR
ALU2 OBSTR

1

```

VIA ALU1 ALU2

1

1300 3700 1300 1300 0 1400 # I2L generator data

WEITERES

LOADPAR(PARIS), SAVEPAR(PARIS)