

Erweiterungen in PARIS

TEMIC
Semiconductor
GmbH

13. März 1998

Inhaltsverzeichnis

1	Hierarchischer Netzlistenvergleich	1
2	Überprüfung der Substrat-/Boxkontakierung im Layout	2
3	Behandlung von Netzen	3
4	RPC Schnittstelle	6
5	Steuerung von Hintergrundprozessen aus PARIS	6
6	Tcl/Tk Plugin	7
7	Layer für Texte	8

1 Hierarchischer Netzlistenvergleich

Um sicherzustellen, daß für alle Zellen eines Designs ein aktueller Netzlisten- und Attributvergleich durchgeführt wird, wurde ein Kommando zum hierarchischen Netzlistenvergleich implementiert. Der hierarchische Netzlistenvergleich ist im Verify Menü unter dem Knopf `CmpHierarchy` aufrufbar. Dieses Kommando durchläuft die Zellhierarchie der gerade editierten Zelle und führt einen Netzlisten- und Attributvergleich durch für die aktuelle Zelle und alle Zellen, die in der aktuellen Zelle aufgerufen sind. Dabei wird die Zellhierarchie in *bottom up* Reihenfolge, also von unten nach oben durchlaufen. Der Hierarchiedurchlauf erfolgt mit dem neuen Kommando `TRAVERSEDEFHIERARCHY`. Innerhalb von `TRAVERSEDEFHIERARCHY` wird jede Zelle in der Hierarchie einmal besucht und ein vom Benutzer angebbares Kommando für jede besuchte Zelle aufgerufen. Der Netzlistenvergleich für eine Zelle in der Hierarchie wird dabei in der Funktion `NetcmpCell` durchgeführt. `NetcmpCell` erledigt alle Aktionen, die zum Durchführen des Vergleichs notwendig sind, insbesondere auch den Aufruf des Mentor Netlisters `anetv8 -c` und den Aufruf des PARIS Netlisters `CREATELIST`. Auf einen expliziten Aufruf der Netlister kann bei Anwendung des hierarchischen Vergleichs also verzichtet werden. Es geschieht genau das, was gemacht werden muß, aber nicht mehr.

Basiszellen bleiben von der Bearbeitung ausgenommen. Für hierarchische Zellen wird der Vergleich nur für diejenigen Zellen in der Hierarchie durchgeführt, für die ein neuer Vergleich erforderlich ist. Ein neuer Vergleich ist dann notwendig, wenn die Eingangsdaten des Vergleichs jünger sind als die Ergebnisse. Die Eingangsdaten eines Vergleichs sind das Schematic und die Layoutzelle. Aus dem Schematic wird die Netzliste `schema/<schematic-cell>/lvs_vpt.kno` abgeleitet. Die Layoutzelle liefert die Netzliste `<layout-cell>_lay.kno`. Der Netzlistenvergleich ergibt dann das Ergebnis `<layout-cell>_lay.dif`. Analog entstehen die Listen für den Attributvergleich `schema/<schematic-cell>/lvs_vpt.atr` und `<layout-cell>_lay.atr`, deren Vergleich dann das Ergebnis `<layout-cell>_lay.adf` liefert.

Zum Erzeugen der Layoutnetzliste *nach Bedarf* wurde das Kommando `CREATELIST` um die Option `-COND` erweitert. `-COND` bewirkt, daß nur dann die Ausgabelisten geschrieben werden, wenn entweder die dazugehörige Datei nicht existiert oder älter ist als die Zelle, aus denen sie hervorgeht. Eine editierte Zelle liefert immer eine Netzliste unter der Annahme, daß eine editierte Zelle immer jünger ist als die Netz- bzw. Attributlisten. Alle anderen Aktionen basieren auf den Modifikationsdaten von Dateien. Diese Aktionen werden mit Hilfe des UNIX Programms `make` ausgeführt.

Als Name der einer Layoutzelle zugeordnete Schematiczelle wird der Wert des `CLASS` Properties der Layoutzelle oder bei nicht vorhandenem `CLASS` Property der Name der Layoutzelle angenommen. Dies entspricht der bisher bekannten Vorgehensweise zur Ermittlung der Zellnamen bei der Ausgabe der Layout Netzlisten.

Als Ausgabe neben den üblichen Differenzdateien wird beim hierarchischen Vergleich eine Zusammenfassung der durchgeführten Vergleiche an die Protokolldatei `netcmp.log` angehängt. Diese Datei enthält die Namen der bearbeiteten Zellen, das Datum der Bearbeitung und eine Zeile mit dem Ergebnis, entweder "Differences detected" oder "No differences detected". Damit läßt sich ein schneller Überblick gewinnen, was gemacht wurde. Die ausführlichen Vergleichsergebnisse stehen wie gewohnt in den `.dif` bzw. `.adf` Dateien. Die Protokolldatei `netcmp.log` wird nicht gelöscht oder überschrieben. Die Ergebnisse eines hierarchischen Vergleiches werden immer am Ende angehängt. `Netcmp.log` beinhaltet damit immer die gesamte Historie der durchgeführten Vergleiche.

2 Überprüfung der Substrat-/Boxkontaktierung im Layout

Die Prüfung der Substrat-/Boxkontaktierung betrifft alle nicht isolierten Bauelemente, also diffundierte Widerstände, MOS Transistoren, vertikale NPNS, laterale und Substrat PNPS, Kapazitäten, die in einer Wanne liegen müssen. Diese Elemente erhalten eine zusätzlichen Pin, der für die Substrat/Wannenkontaktierung verwendet wird. Ein Transistor hat dann 4 statt 3 Pins, ein Widerstand 3 statt 2 und der Substrat PNP erscheint als normaler Transistor mit 3 Anschlüssen. Die Schematic Symbole müssen dahingehend erweitert werden. In der Regel ist dazu eine neue Version der Bibliothek erforderlich.

Um die Substratkontaktierung zu prüfen, wird ein zusätzlicher Design Layer eingeführt, der als leitende Fläche das Substrat symbolisiert. Die Substrat Pins in den Bauelementen müssen zusätzlich in diesem Substrat Layer digitalisiert werden. Bei generierten Elementen erfolgt dies automatisch. Bei von Hand erzeugten Zellen muß dieser Layer zusätzlich digitalisiert werden. Es genügt in der Regel, die Bounding Box der Zelle zu digitalisieren. Dies kann ebenfalls per Knopfdruck automatisch erfolgen, so daß der zusätzliche Aufwand sehr gering ist. Der Substratlayer muß auch in hierarchischen Zellen vorhanden sein.

In der aus dem Layout extrahierten Netzliste erscheinen dann alle Zellen mit Substratlayer mit einem zusätzlichen Pin, so daß die vollständige Übereinstimmung mit der Schematicnetzliste erreicht wird. Irgendwelche Tricks mit NRECTANGLES zur Simulation der Substrat- verkopplung sind dann auch nicht mehr notwendig.

Der Pintyp des Substratpins ist PSUB, des Wannepins PBOX. Wenn diese Konvention eingehalten ist werden diese Pins bei der Zusammenfassung von Widerständen und Kapazitäten ignoriert, so daß alles wie bekannt funktioniert.

Die Routing Layer für die Schaltungsextraktion werden um den Substrat- layer und den Wannen- layer sowie geeignete Kontaktlayer, die das Substrat bzw. die Wanne kontaktieren, erweitert. Dadurch erhöht sich der Zeitaufwand für die Schaltungsextraktion. Eine konkretes Maß dafür kann noch nicht angegeben werden, eine entsprechende Messung ist geplant. Da die Anzahl der Elemente im Substrat- layer jedoch nicht sehr hoch ist und normalerweise nur Rechtecke im Substratlayer liegen, dürfte der Zusatzaufwand dafür in erträglichen Grenzen liegen.

3 Behandlung von Netzen

PARIS bietet die Möglichkeit, innerhalb einer Layoutzelle *Netze* zu behandeln. Ein Netz ist eine spezielle Form einer Gruppierung von Objekten. In einem Netz sind alle diejenigen Objekte eingebunden, die elektrisch miteinander verbunden sind. Ein Netz enthält daher normalerweise Pins oder Geometrie Elemente. Ein Pin ist ein Objekt, das den Anschluß an einen Port einer instantiierten Zelle beschreibt.

Die neu eingeführten Netzobjekte dienen dazu, den *Sollzustand* der elektrischen Verbindungen zu bezeichnen, also diejenigen elektrischen Verbindungen, die in einem korrekt erstellten Layout vorhanden sein müssen. Der Sollzustand der elektrischen Verbindungen ist üblicherweise in einem Schaltbild abgelegt. Aus dem Schaltbild wird eine Netzliste extrahiert. Die Netzliste beschreibt den Graphen der elektrischen Verbindungen ohne die Graphik des Schaltbildes zu beinhalten.

Netze sind, wie andere Objekte auch, selektierbar. Ein selektiertes Netz ist wie gewohnt durch das Zeichnen der selektierten Objekte in der Selektfarbe und mit fetten Linien erkennbar. Dabei sind die im Netz enthaltenen Objekte aber nicht direkt sondern nur indirekt selektiert. Direkt selektiert ist dabei nur das Netzobjekt selbst. Da ein Netz nur ein Containerobjekt ist, aber nicht ein konkretes geometrisches Objekt, können die indirekt über ein Netz selektierten Objekte nicht modifiziert werden. Sie erscheinen nur in Selektdarstellung, um die Bestandteile des Netzes hervorzuheben. Ein Netz wird selektiert entweder über den Menüknopf `PopupMenu/Select/SNet` oder mit Hilfe der Taste `n`. Diese Menüeinträge bzw. Tastenbelegung wurden neu eingefügt. Entsprechend können Netze über den Menüknopf `PopupMenu/Unselect/SNet` oder die Taste `N` deselektiert werden. `UNSELECT-ALL` wirkt ebenfalls auf Netze.

Die in einem Layout tatsächlich vorhandenen elektrischen Verbindungen werden in PARIS durch die *Knotennummer* identifiziert. Alle Objekte mit elektrisch relevanter Funktion, also Pins und Geometrie, haben ein solches Knotennummer Attribut. Die Knotennummern bezeichnen den *Istzustand* der elektrischen Verbindungen.

In einem Layout können im allgemeinen Unterschiede zwischen dem Soll- und dem Istzustand auftreten. Dabei kann es fehlende Verbindungen oder überzählige Verbindungen geben. Fehlende Verbindungen werden mit Hilfe von *Netzlinien* visualisiert. Überzählige Verbindungen werden daran erkannt, daß Elemente mit gleicher Knotennummer in unterschiedlichen Netzen enthalten sind. Überzählige

Verbindungen können nicht visualisiert werden, da der Layouteditor nicht entscheiden kann, welches Element in einem Knoten die falsche Verbindung der Netze verursacht. Statt dessen wird eine textuelle Fehlermeldung ausgegeben.

Ausgehend von einer Netzliste können in einer bestehenden Layoutzelle in PARIS Netzobjekte erzeugt werden. Dies erfolgt mit dem Kommando NETLIST. Das Kommando NETLIST dient dazu, entweder eine Netzliste in eine Layoutzelle zu überlagern, oder umgekehrt die Netzinformation aus einer Layoutzelle zu entfernen. Einzelheiten zum Kommando NETLIST stehen in der entsprechenden Manual Seite. Die Gültigkeit der Netzinformation einer Layoutzelle wird durch das Attribut `snetFlag` der Zelle gekennzeichnet. Das Installieren einer Netzliste in eine Zelle aktiviert das `snetFlag`, das Deinstallieren der Netzinformation deaktiviert das `snetFlag`. Gültige Netzinformation in einer Zelle kann ebenso in importierten Layouts vorhanden sein, z.B. wenn ein Layout durch eine automatische Platzierung erzeugt wurde.

Zum Installieren oder Deinstallieren einer Netzliste in einer Layoutzelle ist das Pulldown Menü `VerifyMenu/Netlist` definiert. Mit dem `Install` Knopf in diesem Menü wird die Netzliste unter dem Pfadnamen `schema/<schematic-name>/lvs.vpt.kno` in die Layoutzelle überlagert. Wenn das Schematic jünger als die Netzliste ist, wird vor dem `Import` mit `anetv8` automatisch eine aktuelle Netzliste erzeugt. Die Name der Schematiczelle wird aus der Layoutzelle gewonnen, entweder als Wert des `CLASS` Properties oder als Name der Layoutzelle.

In einem noch unvollständig verdrahteten Layout besteht ein Netz in der Regel aus einer Menge von Teilnetzen. Ein Teilnetz beinhaltet dabei eine Menge von Pins oder Geometrie mit gleicher Knotennummer. Die verschiedenen Teilnetze sind dadurch gekennzeichnet, daß sie zwar unterschiedliche Knotennummern haben, aber in das selbe Netz eingebunden sind. Die Einbindung in ein Netz kennzeichnet die noch getrennten Teilnetze als noch zu verbindende Teile eines Netzes. Ein vollständig verdrahtetes Netz beinhaltet nur noch ein Teilnetz, in einem vollständig unverdrahteten Netz bildet jedes Objekt des Netzes ein eigenes Teilnetz. Die Netzlinien in PARIS zur Visualisierung fehlender Verbindungen werden als die kürzesten Verbindungen aller noch verbliebenen Teilnetze eines Netzes erzeugt, wenn das Netz noch mindestens zwei Teilnetze enthält. Dagegen enthält ein vollständig verdrahtetes Netz keine Netzlinien mehr.

Netzlinien können nur in Zellen dargestellt werden, deren Netzinformation gültig ist. Netzlinien sind flüchtige Objekte, d.h. sie werden nicht permanent abgespeichert, sondern immer aktuell während einer PARIS Sitzung berechnet. Netzlinien existieren nur in editierten Zellen. Sie werden erstmalig beim Öffnen einer Zelle zum Editieren berechnet. Wenn die Online Verifikation eingeschaltet ist, erfolgt die Aktualisierung der Netzlinien nach jedem Editiervorgang. Dabei wird auch die Einbindung von Geometrie in Netze berücksichtigt. Es wird dann also immer der aktuelle Stand von noch fehlenden Verbindungen angezeigt. Bei abgeschalteter Online Verifikation, was auch immer ein ungültiges `netFlag` der editierten Zelle impliziert, wird die zuletzt durchgeführte Netzlinienberechnung beibehalten. Es erfolgt dann keine ständige Aktualisierung der Netzlinien. Bei ungültigem `netFlag` können Netzlinien nur zwischen den Pins eines Netzes erzeugt werden, Ein Einfügen von Geometrie in ein Netz findet dann wegen der fehlenden Layoutnetzinformation nicht statt. Eine Aktualisierung der Netzlinien wird immer nach einer vollständigen Konnektivitätsextraktion einer Zelle mit Hilfe des Kommandos NETEXT durchgeführt. Ein Aufruf von NETEXT erzeugt also auch einen aktuellen Stand der Netzlinien einschließlich der Einbindung von Geometrie in Netze.

Während der Konnektivitätsextraktion werden auch Geometrielemente, für die eine Überlappung mit einem bestehenden Netz festgestellt wird, in das Netz eingebunden. Wenn ein Geometrielement

dann nachträglich verschoben wird, so daß die geometrische Verbindung mit dem Netz zwar unterbrochen wird, aber keine Verbindung zu einem anderen Netz aufgebaut wird, bleibt das Geometrieelement in seinem vorherigen Netz eingebunden. Durch diese Technik können Vorverdrahtungen von Netzen sehr leicht vorgegeben werden. Die Zuordnung derartiger isolierten Geometrieelementen zu Netzen bleibt so lange erhalten, bis eine komplette Initialisierung der Netzinformation stattfindet. Die Netzinformation wird immer zu Beginn einer vollständigen Konnektivitätsextraktion neu initialisiert. Wenn also das Kommando NETEXT aufgerufen wird, geht die Netzzuordnung von isolierten Geometrieelementen verloren. Abbildung 1 zeigt einen Bildschirmausschnitt eines Layouts mit Netzlinien.

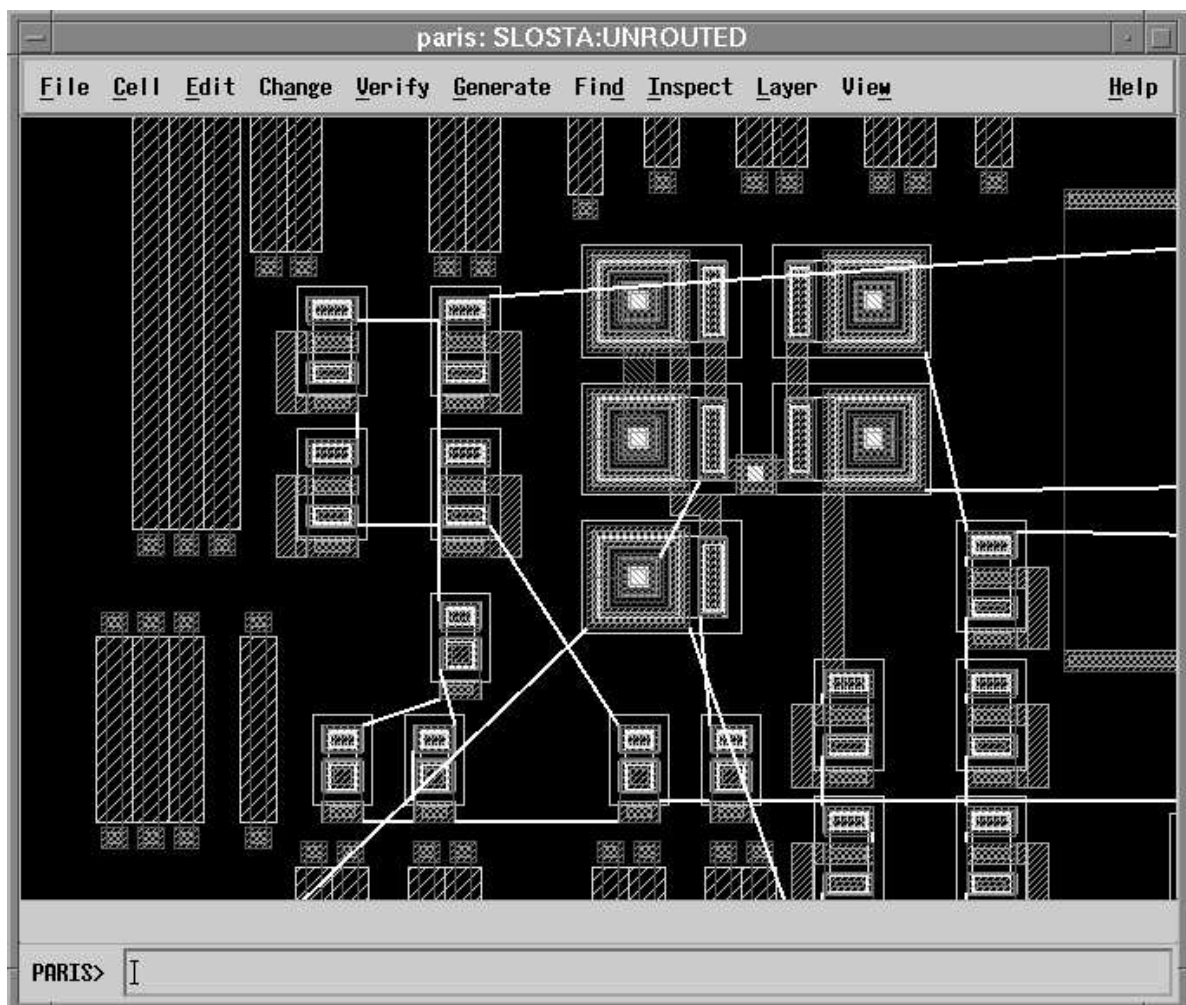


Abbildung 1: Layoutbeispiel mit Netzlinien

4 RPC Schnittstelle

Um anderen Programmen den Zugriff auf die PARIS internen Daten zu ermöglichen, kann PARIS als RPC Server laufen. Bisher war es nur beim Aufruf von PARIS möglich, die RPC Schnittstelle mit dem Schalter `-rpc` zu aktivieren. Die RPC Schnittstelle war dann während der gesamten PARIS Prozeßlaufzeit aktiv. Mit Hilfe des neuen Kommandos `RPC` kann jetzt zu jeder beliebigen Zeit während einer PARIS Sitzung die RPC Schnittstelle aktiviert oder auch deaktiviert werden.

Es konnte nur einen PARIS RPC Server auf einer Maschine geben, da der PARIS RPC Server auf dem lokalen Portmapper der Maschine registriert wurde. Dieses Verhalten führte zu Problemen, wenn mehrere Benutzer auf einer Maschine eine über RPC angebundene Applikation (ALSYN) verwenden wollten, die mit jeweils verschiedenen PARIS Prozessen auf der Maschine kommunizieren sollten. Das Kontaktieren des PARIS RPC Servers war bisher einfach durch Angabe des Hostnamens, auf dem der Server läuft, möglich. Die Portnummer wurde dabei automatisch mit Hilfe des Portmappers ermittelt.

Die neue Version der RPC Schnittstelle registriert den PARIS RPC Server nicht mehr beim Portmapper. Damit ist es möglich, mehrere PARIS RPC Server auf einer Maschine laufen zu lassen. Der Port, auf dem ein bestimmter PARIS RPC Server hört, kann dann allerdings nicht mehr beim Portmapper automatisch abgefragt werden, sondern ist nur innerhalb des PARIS Prozesses bekannt. Die Portnummer wird in die Shell Variable `RPCPORT` eingetragen. Sie kann dann zum Starten von Clients innerhalb von PARIS benutzt werden. Dies erfolgt in der Regel automatisch innerhalb von Shell Skripten, ohne daß der Benutzer sich um die Portnummer kümmern muß. Die Variable `RPCPORT` kann aber mit Hilfe der Kommandos `echo` oder `msg` abgefragt werden, falls ein Kontaktieren des PARIS Servers von Hand einmal notwendig werden sollte. Ein bestimmter PARIS RPC Server kann jetzt nur noch durch Angabe von Hostname und Portnummer in der Syntax `hostname:portnumber` kontaktiert werden. Bei fehlender Angabe des Hostnamens, z.B. in der Form `:portnumber` wird `localhost` als Hostname angenommen.

Ein RPC Client kann sowohl innerhalb als auch außerhalb von PARIS gestartet werden. Wird der Client innerhalb von PARIS gestartet, ist **unbedingt** darauf zu achten, daß der Client als **Hintergrundprozeß** mit dem `&` Zeichen am Ende der Kommandozeile gestartet wird. Wenn dies nicht geschieht, blockieren sich PARIS und der Client gegenseitig. Der Client Prozeß kann dann nur noch in einem anderen Terminal von Hand mit dem `kill` Kommando beendet werden.

5 Steuerung von Hintergrundprozessen aus PARIS

In manchen Anwendungen ist es notwendig, von PARIS aus Daten an einen Hintergrundprozeß weiterzugeben, um z.B. den Hintergrundprozeß mit Hilfe von Kommandos zu steuern. Die Steuerkommandos des Hintergrundprozesses können dabei innerhalb von PARIS durch die graphische Oberfläche erzeugt werden. Das im Hintergrund gestartete Kommando braucht dabei nicht über eine graphische Oberfläche zu verfügen. Dort ist lediglich eine textuelle Eingabe von Kommandos erforderlich.

Eine Anwendung für diese Systemarchitektur ist z.B. ein im interaktiven Modus laufender Verdrahter, der als Hintergrundprozeß gestartet wird. Der Verdrahter holt sich bestimmte Information von PARIS über die RPC Schnittstelle. PARIS läuft dabei also als Server für die Layoutdaten, der Router als Client. Zum interaktiven Betrieb des Routers muß aber der Router von PARIS aus mit Anweisungen versorgt werden. Unter diesem Gesichtspunkt wird das Server-Client Verhältnis umgedreht, der Router läuft als

Server und PARIS als Client.

Um die Übergabe von Kommandos von PARIS an andere Prozesse zu ermöglichen, können diese Prozesse in der `ash` als tty verbundene Jobs gestartet werden. Dabei wird die Standardeingabe des Hintergrundprozesses über ein Pseudoterminal (tty) mit einem Ausgabedatenstrom auf der PARIS Seite verbunden. Über diese Verbindung können dann Daten von PARIS aus zur Standardeingabe des Hintergrundprozesses fließen. Das Ganze verhält sich dann genauso, als ob der Prozeß in einem normalen Terminal gestartet wurde, nur mit dem Unterschied, daß seine Eingabedaten nicht mehr vom Terminal gelesen werden, sondern von PARIS erzeugt werden. Ein solcher über ein tty angebundener Prozeß wird mit dem Kürzel `&0` in der `ash` gestartet:

```
my-interactive-command ... &0
my_interactive_command_id=$!
```

Die Prozess-ID des Hintergrundprozesses wird dabei in die Shell Variable `$!` geschrieben. Der so gestartete Hintergrundprozeß unterliegt wie jedes andere gestartete Shell Kommando der Jobverwaltung der `ash`. Es können also mehrere Hintergrundprozesse gestartet werden. Mit Hilfe seiner Prozess-ID ist jeder einzelne Hintergrundprozeß adressierbar.

Die Datenübergabe an Hintergrundprozesse erfolgt mit dem Shell Kommando `pass`. `pass` kann entweder die Kommandoargumente an einen Hintergrundprozeß weitergeben oder den Inhalt von Dateien. Dabei gibt zwei Möglichkeiten:

```
pass process-id -s args ...
pass process-id -f file ...
```

In der ersten Variante, `-s` steht für Stringargumente, werden die restlichen Argumente der Kommandozeile durch Leerzeichen getrennt und mit einem Newline Zeichen am Ende an die Eingabe des Jobs mit der Prozeß-ID `process-id` weitergegeben. In der zweiten Variante, `-f` steht für Fileargumente, wird der Inhalt der angegebenen Dateien an den Prozeß weitergegeben.

6 Tcl/Tk Plugin

Mit Hilfe der Tcl/Tk Software lassen sich leicht komfortable Benutzeroberflächen implementieren. Um diese Möglichkeiten auch in PARIS nutzen zu können, wurde ein Paket zur Anbindung von Tcl/Tk Programmen an PARIS implementiert. Das Tcl/Tk Plugin besteht aus dem Tcl Package `pariscmd` und innerhalb von PARIS aus einer erweiterten RPC Schnittstelle. Eine Tcl/Tk Applikation wird über die RPC Schnittstelle an PARIS angebunden. Die Tcl/Tk Applikation läuft dabei als getrennter Client Prozeß und PARIS als Server. Die Tcl/Tk Applikation wird als RPC Client gestartet und kann dabei an einen bestimmten PARIS Prozeß angebunden werden (Abschnitt 4). Von der Tcl/Tk Applikation können dann Kommandos in PARIS ausgeführt und die Ausgaben von PARIS Kommandos gelesen werden.

Das Tcl Package `pariscmd` stellt in einer Tcl Shell die beiden Kommandos `paris-command` und `paris-command-res` bereit. Mit beiden Kommandos lassen sich im PARIS Server Kommandos ausführen. Das Verhalten von `paris-command` entspricht dem gewohnten Verhalten eines Kommandos in PARIS. `paris-command` liefert entweder `OK` oder `ERROR` als Returnstatus, aber keinen Ergebnistext. Im Gegenteil zu `paris-command` liefert `paris-command-res` neben dem Returnstatus

des Kommandos noch die Standardausgabe des Kommandos als Tcl Ergebnisstring. `paris-command-res` verhält sich also konform, zu anderen Tcl Kommandos, die ja auch neben dem Status einen Ergebnisstring liefern können.

Das Tcl Package `Pariscmd` wird als dynamisch ladbares Paket bereitgestellt. Die derzeitige Version 1.0 von `Pariscmd` erfordert die Tcl Version 7.6. In einem Tcl Skript sind zur Nutzung von `Pariscmd` folgende Kommandos einzusetzen:

```
lappend auto_path $env(AEGDIR)/sys/lib/$env(PLATFORM)
package require Pariscmd
```

Das Kommando `lappend auto_path` erweitert den Suchpfad für Tcl Packages um das Verzeichnis `$AEGDIR/sys/lib/$PLATFORM`, wo das Paket `Pariscmd` als shared Library `pariscmd.sl` bzw. `pariscmd.so` abgelegt ist. Mit dem Kommando `package require Pariscmd` wird dann dieses Paket in einer Tcl Shell dynamisch geladen. Danach sind die beiden Kommandos `paris-command` und `paris-command-res` aufrufbar. Den beiden `paris-command` Anweisungen muß immer die Hostspezifikation (Abschnitt 4) des PARIS RPC Servers als erstes Argument übergeben werden. Die weiteren Argumente werden dann zu dem PARIS Kommando zusammengesetzt. Die folgende Anweisung öffnet in dem PARIS Server, der auf dem lokalen Host unter der Portnummer 2222 hört, die Zelle X zum Editieren:

```
paris-command :2222 changedef x
```

`changedef` erzeugt keine Ausgaben, daher kann es mit `paris-command` aufgerufen werden. Das folgende Kommando fragt von PARIS alle Zellnamen ab und schreibt diese in die Tcl Liste `cells`:

```
set cells [list [paris-command-res :2222 list -cells]]
```

Hier muß `paris-command-res` verwendet werden, da `list -cells` alle Zellnamen in die Standardausgabe schreibt.

7 Layer für Texte

Zur Verbesserung der Bearbeitung von Texten werden Texte wie Geometrieelemente einem Layer zugeordnet. Zur Darstellung von Texten werden die Darstellungsattribute (Farben) verwendet, die für den jeweiligen Layer definiert sind. Damit ist jetzt eine farbige Darstellung von Texten möglich. Weiterhin wird eine bessere Kompatibilität mit anderen Designsystemen erreicht, die ebenfalls Layer für Texte zulassen. Programmintern ergibt sich eine Vereinfachung, da die gesonderte Behandlung von Texten entfernt werden konnte. Das führt in großen hierarchischen Designs zu einer deutlichen Verkürzung des Bildaufbaus.

Bestehende Layouts lassen sich ohne zusätzlichen Benutzereingriff migrieren. Texte ohne Layer erscheinen im Layer 255 in der für den Layer 255 definierten Farbe. Diese Zuweisung erfolgt automatisch beim Laden. Für die Bearbeitung von Textelementen haben die neuen Textlayer folgende Konsequenzen:

1. Es muß für neu mit `addtext` erzeugte Textelemente ein Layer zugewiesen werden. Hierfür wird derselbe aktuelle Layer wie für neu addierte `GeoElemente` in `addgeo` verwendet. Wenn Texte in einem anderen Layer addiert werden sollen, muß vorher `setpar -actlayer` aufgerufen werden.

2. Der Layer von bestehenden Textelementen kann mit dem Kommando `changecomp -layer` verändert werden. Dies trifft ebenfalls auf das Kommando `cnvlayer` zu, wo Textlayer genau wie Geometrielayer behandelt werden.

Bezüglich des Layers verhalten sich Texte als jetzt genauso wie GeoElemente.

Neben normalen Textobjekten werden die Texte von Zell-, Pin- und Platzierungsnamen ebenfalls mit den Zeichenfunktionen für Texte ausgegeben. Diese Namen können daher ebenfalls mit einer bestimmten Farbe gezeichnet werden. Natürlich nicht jeder Name einzeln, sondern nur gruppenweise, also eine Farbe jeweils für Zellnamen, Pinnamen und Platzierungsnamen. Standardmäßig werden die Namen im Layer 255 gezeichnet. Mit Hilfe des `setpar` Kommandos kann der Layer zur Darstellung der Namen umgeschaltet werden. `setpar` hat dazu die folgenden neuen Optionen:

```
setpar -layercellame <layer>  
setpar -layerpiname <layer>  
setpar -layerplname <layer>
```

Die jeweiligen Namen erscheinen dann so, als ob sie wie gewöhnliche Textobjekte in dem angegebenen Layer enthalten wären.

In GDSII werden die Texte *mit* ihrem Layer exportiert. Dabei unterliegen Texte der GDSII Layerkonvertierung. Dies gilt auch für Zell-, Pin- und Platzierungsnamen. Diese erscheinen in der GDSII Datei ebenfalls mit dem konvertierten Layer, in den sie mit `setpar` eingeordnet wurden. Werden spezielle Textlayer benutzt, muß natürlich eine GDSII Layerkonvertierung für die Textebenen vorhanden sein.