

## Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>1 Einführung</b>                           | <b>3</b>  |
| 1.1 offenes System . . . . .                  | 3         |
| 1.2 Bedienoberfläche . . . . .                | 4         |
| 1.3 Datenhaltung . . . . .                    | 5         |
| 1.4 Properties . . . . .                      | 7         |
| <b>2 Design Framework</b>                     | <b>7</b>  |
| 2.1 CADENCE starten . . . . .                 | 7         |
| 2.2 CADENCE beenden . . . . .                 | 8         |
| 2.3 Online-Dokumentation . . . . .            | 8         |
| 2.4 Library-Manager . . . . .                 | 10        |
| 2.5 Designs bearbeiten und erzeugen . . . . . | 11        |
| <b>3 Der Layout-Editor</b>                    | <b>12</b> |
| 3.1 Starten, Sichern, Verlassen . . . . .     | 12        |
| 3.2 Auswahl der Layer . . . . .               | 12        |
| 3.3 Benutzung der Maus . . . . .              | 14        |
| 3.4 Fensterkontrolle . . . . .                | 14        |
| 3.5 Eingabehilfen . . . . .                   | 15        |
| 3.6 Layout erzeugen . . . . .                 | 15        |
| 3.7 Layout verändern . . . . .                | 17        |
| 3.8 Hierarchie . . . . .                      | 18        |
| 3.9 Properties . . . . .                      | 19        |
| <b>4 Der Schematic-Editor</b>                 | <b>20</b> |
| 4.1 Starten, Verlassen . . . . .              | 20        |
| 4.2 Rule Check und Sichern . . . . .          | 20        |
| 4.3 Benutzung der Maus . . . . .              | 21        |
| 4.4 Fensterkontrolle . . . . .                | 21        |
| 4.5 Eingabehilfen . . . . .                   | 22        |
| 4.6 Schematic zeichnen . . . . .              | 23        |
| 4.7 Schematic verändern . . . . .             | 24        |
| 4.8 Hierarchie . . . . .                      | 25        |
| 4.8.1 Bottom-up Design . . . . .              | 26        |
| 4.8.2 Top-down Design . . . . .               | 26        |
| 4.9 Properties . . . . .                      | 27        |

## Legende

Für die Beschreibung der Aktionen bei der Benutzung unserer EDA-Programme wird die folgende Symbolik benutzt. Rechtsbündig steht jeweils der Titel des Fensters, in dem die Aktion durchzuführen ist, wie in den Beispielen:

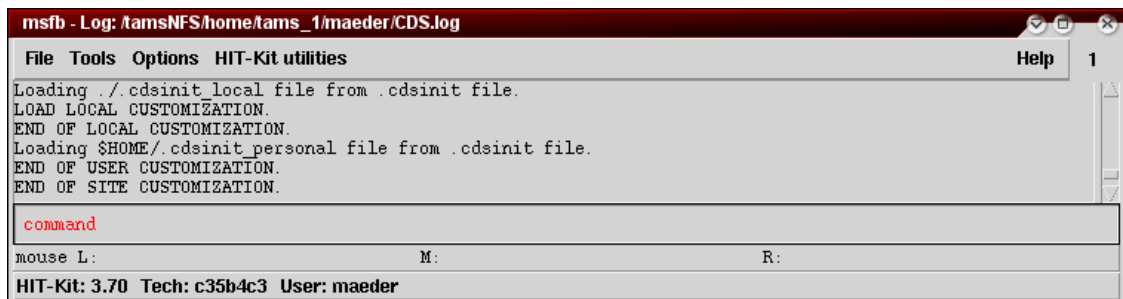
- > command [maeder on...]

Eingabe von command an die Unix-Shell:



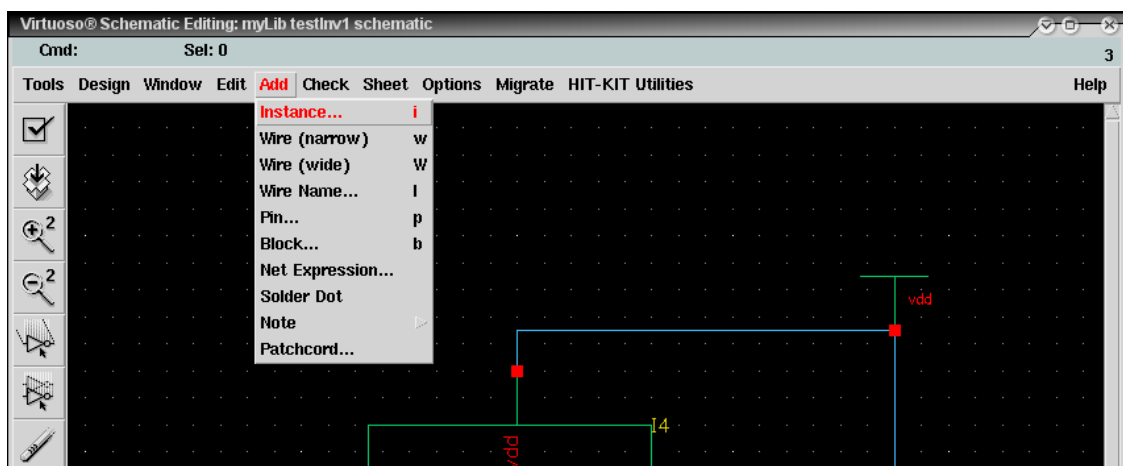
- ▷ command [msfb - Log]

Eingabe von command an das CADENCE-Kommandointerface:



- Add - Instance [Virtuoso Schematic...]

Auswahl aus einem pull-down Menü, bzw. Untermenü:

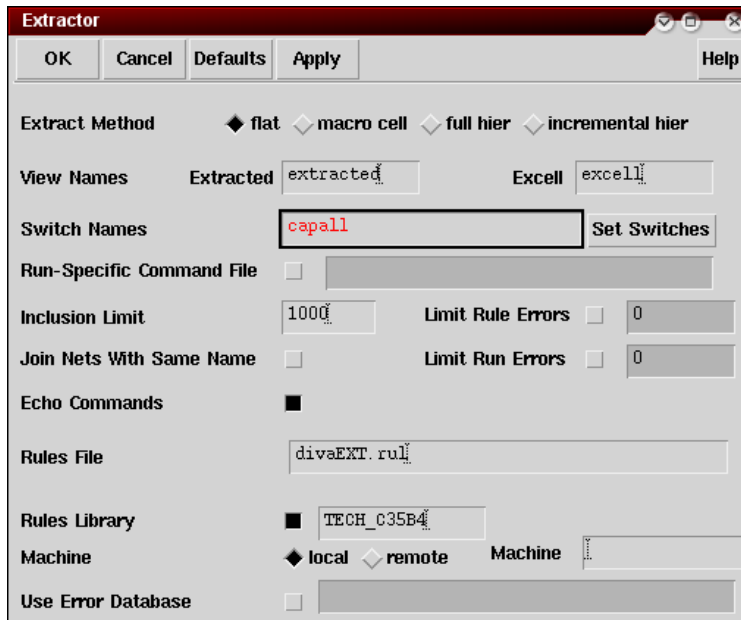


- ⊙ i [Virtuoso Schematic...]

alternative Eingabe eines „Bindkeys“ über die Tastatur.

≡ Switch Names = capall [Extractor]  
 <Schlüssel1> = <Wert1>  
 <Schlüssel2> = <Wert2>

Eingabe zum Ausfüllen von Formularfenstern:



Dabei wird für das Feld <Schlüssel> der <Wert> entweder über die Tastatur eingegeben, aus einer Liste vorgegebener Möglichkeiten ausgewählt, aktiviert etc. Sind alle Werte entsprechend den Vorgaben eingestellt, so wird die Eingabe anschließend durch OK, bzw. Apply ausgeführt. Bei den oft sehr umfangreichen Menüs sind nicht immer alle Optionen aufgelistet, sondern nur die jeweils wichtigen.

↑<sub>x</sub> object [Xwin]

Auswahl von <object> mit der Maus (Maustaste: x). Für Knöpfe in Menüs wird auch die Notation Text (s.o.) benutzt.

## 1 Einführung

Eine detailliertere Beschreibung der CADENCE Software würde den Rahmen dieser „Kurz“-Anleitung sprengen — dafür gibt es über 800 MB Online-Dokumentation. Hier werden nur einige Grundlagen angesprochen, die sich gezielt auf die in den Lehrveranstaltungen benutzten Teile des Systems beziehen. Sie sollen einen ersten Einstieg beim Umgang mit der Software ermöglichen.

### 1.1 offenes System

Das was hier immer als CADENCE IC bezeichnet wird ist eigentlich eine (sehr große) Menge von Programmen (Datenmanager, Editore, Simulatoren, Layoutprogramme...), die auf eine gemeinsame Designdatenbasis zugreifen. Die meisten Programmkomponenten sind über eine gemeinsame Oberfläche, das CADENCE DF II (Design Framework), erreichbar.

Dabei ist CADENCE ein offenes Design-System, in das eigene Komponenten eingebracht werden können. Eine eigene Programmierschnittstelle (SKILL) erlaubt es in das System einzugreifen und auf die Entwurfsdaten zuzugreifen.

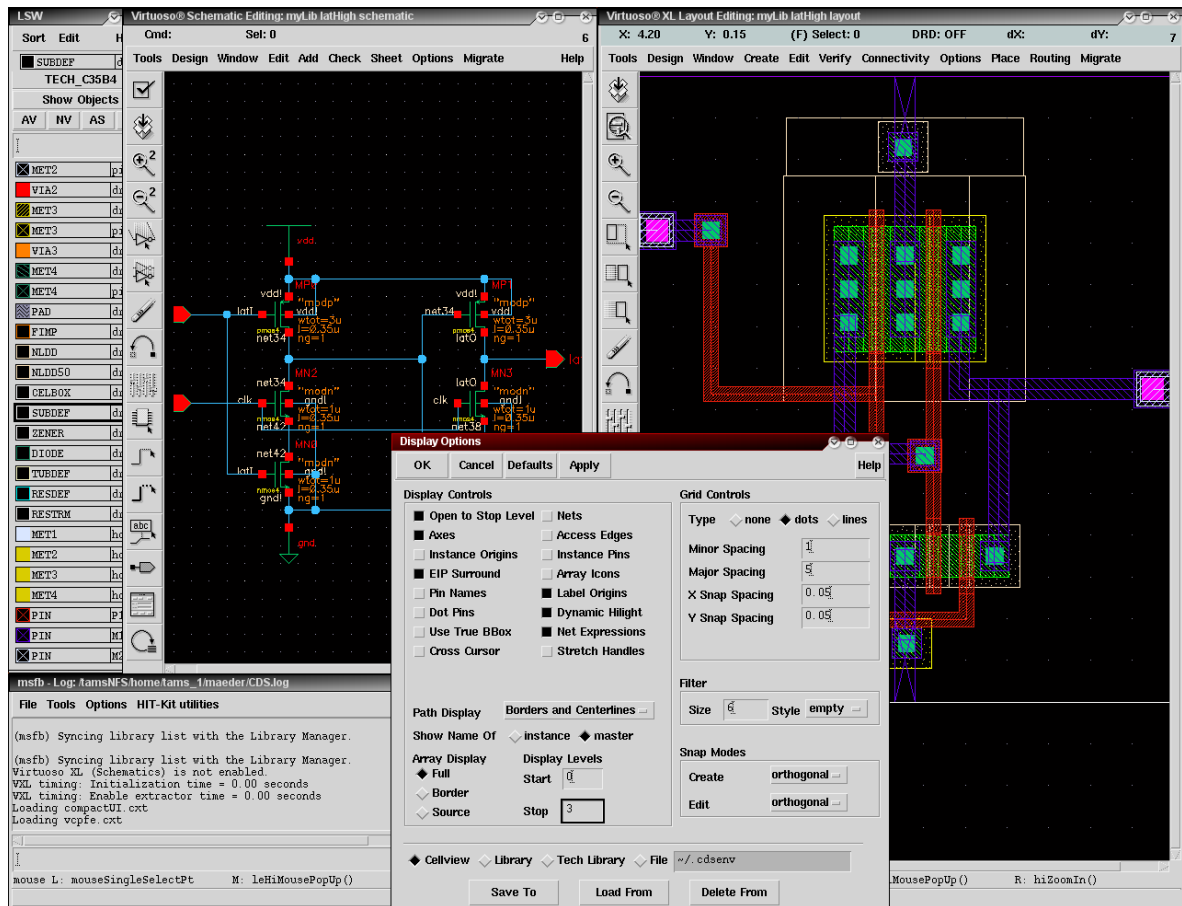


Abbildung 1: CADENCE Fenster

## 1.2 Bedienoberfläche

Typischerweise wird man bei der Benutzung von CADENCE mehrere Fenster offen haben, siehe Abbildung 1.

Hauptfenster der Entwurfsumgebung

- listet die Log-Datei mit den Ausgaben der CADENCE-Programme
- zeigt die Funktion der Maustasten an
- ist Eingabefenster für den SKILL-Kommandointerpreter
- erlaubt Einstellungen an dem Gesamtsystem
- startet über Menü die einzelnen Programme

Editor-Fenster

- hier wird ein Design als schematic, layout... bearbeitet
- weitere Entwurfsschritte, wie Simulationen dieses Designs, werden gestartet

Browser-Fenster stellen Hierarchien grafisch dar, wie den Aufbau von Designs oder die Struktur der CADENCE-Bibliotheken. Das letzte Beispiel, der Library Manager, wird später noch genauer beschrieben.

Ausgabe-Fenster listen (Text-) Ausgabedateien auf oder stellen Simulationsergebnisse grafisch dar und erlauben so deren Auswertung.

Auswahl-Fenster stellen als eigenes Fenster zusätzliche Menüs zu den Programmen bereit oder beeinflussen die Funktion von Befehlen (z.B. Layerselektion für Layout, Platzierung und Verdrahtung).

Formular-Fenster erlauben die Einstellung von Optionen und Parametern. Bei vielen Programmteilen von CADENCE wird die Arbeitsweise der einzelnen Werkzeuge über solche *Fill-Forms* gesteuert.

Die (hier vorgestellten) Programme werden über pull-down Menüs gesteuert. Die Menüleiste wird dabei, beispielsweise bei den Editor-Fenstern, abhängig von den Teilaufgaben dynamisch verändert. Häufig benutzte Punkte sind bei vielen Programmen als fixe Menüs am Fensterrand in Form von Symbolen aufgeführt.

### 1.3 Datenhaltung

Die Entwürfe sind hierarchisch, in drei Stufen, organisiert — siehe dazu Abschnitt 2.4.

**Library** : Bibliothek in der sich Entwürfe befinden. Dies sind einerseits vom Benutzer selbst erstellte Bibliotheken, in denen eigene Entwürfe gesammelt werden (z.B. *myLib*); andererseits werden die Zellbibliotheken für Standardzell- oder Gate-Array Entwürfe über Bibliotheken in das System eingebunden.

**Cell** : Name einer Zelle, dabei steht „Zelle“ nur für den Namen eines Elements einer beliebigen Abstraktionsebene beim Entwurf — vom komplett (hierarchisch) entworfenen Chip bis hin zum einzelnen Transistor.

Innerhalb der Bibliotheken können die Zellen noch zu *logischen Gruppen* zusammengefasst werden: den Zellkategorien.<sup>1</sup> Dieser Mechanismus wird beispielsweise bei den Standardzellbibliotheken benutzt, um die Zellen nach Funktionen (Adder, Buffer, Latches, FlipFlops...) zu gruppieren.

**View** : Sichtweise oder Art der Zelle. Sie beschreibt (wie) welche Werkzeuge auf dieses Element der Datenbasis zugreifen dürfen. Die Namen der View sind dabei von System fest vorgegeben:

|                   |  |
|-------------------|--|
| layout            | geometrische Darstellung                   |
| extracted         | schematische Netzliste                     |
| schematic         | schematische Netzliste                     |
| symbol            | schematisches Symbol                       |
| autoLayout        | flache Netzliste für physikalisches Layout |
| verilog, spice... | Simulationsmodell                          |
| abstract          | physikalisches (Flächen-) Modell           |

**Achtung:** Die Bibliotheken dürfen nur über den programminternen **Library Manager**, siehe Abschnitt 2.4, manipuliert werden. Das Löschen oder Kopieren mit Unix-Befehlen führt zu Inkonsistenzen.

<sup>1</sup>Zellkategorien können im **Library Manager** über den Knopf *Show Categories* aktiviert werden.

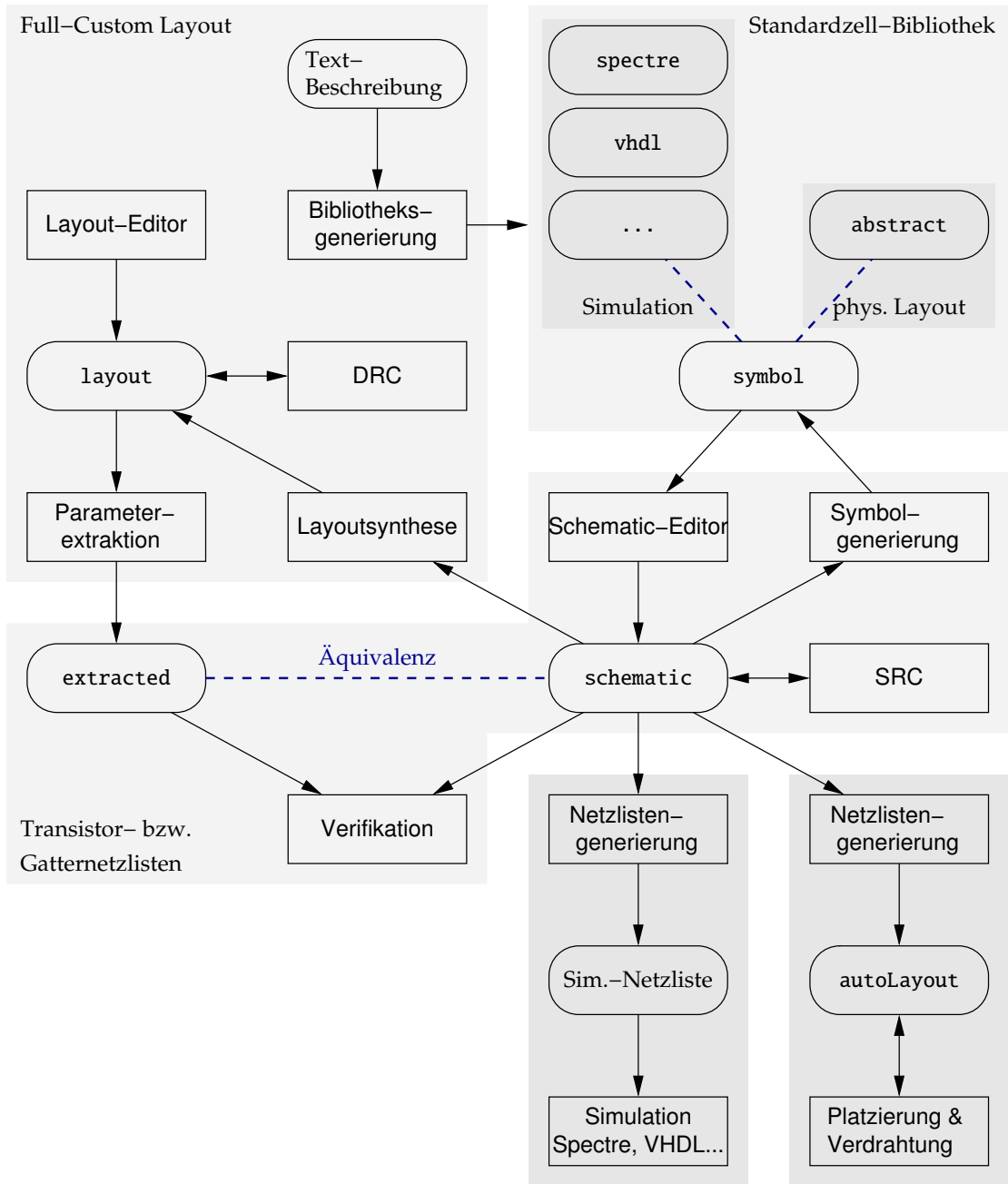


Abbildung 2: Abhängigkeiten zwischen Repräsentationen und den Werkzeugen

Da eine View nur mit Hilfe bestimmter Werkzeuge erzeugt und bearbeitet werden kann, wird beim Öffnen des Designs automatisch das entsprechende Programm gestartet (z.B. der Schematic-Editor). Die Zusammenhänge zwischen den verschiedenen Sichtweisen einer Zelle und den einzelnen Programmteilen in CADENCE sind in Abbildung 2 dargestellt.

## 1.4 Properties

Ein grundlegendes Konzept von CADENCE sind *Properties*. Jedes Objekt eines Layouts oder Schaltplans, aber auch die Zellen und Zellviews, besitzen bestimmte Eigenschaften und Werte, über die der Entwurfsablauf und die Funktion der Programme gesteuert werden. Entwürfe werden mit Properties parametrisiert, Bauteile der elektrischen Ebene, wie Widerstände, Kondensatoren und Transistoren, erhalten ihre Werte etc.

## 2 Design Framework

Neben der eigentlichen CADENCE Programmen stellen die Chip-Hersteller (AMS, Alcatel Mietec, UMC ...) so genannte Design-Kits zur Verfügung, die die Programme an die Entwurfskonzepte des Herstellers anpassen. Darin sind zum Teil erhebliche Softwareanteile enthalten. Außerdem werden in den Design-Kits über die oben angesprochene Programmierschnittstelle Funktionen und Verhalten der Programme umdefiniert. Da dabei auch viel in der Bedienung der Programme verändert wird (Befehle, Menüs...), kann es vorkommen, dass einige der nachfolgend beschriebenen Befehle anders heißen.<sup>2</sup>

### 2.1 CADENCE starten

Die Anpassung an die Design-Kits geschieht über Skripte, die Initialisierungsdateien in das Login-Verzeichnis des Benutzers, beziehungsweise in das aktuelle Verzeichnis kopieren und anschließend das entsprechende CADENCE Programm starten.

**Tipp:** Erhält man beim Start des Systems die „falsche“ Entwurfsumgebung oder werden Zellbibliotheken nicht mehr gefunden, so sollte man sich folgende Dateien ansehen — in den Design-Kits haben diese Dateien zum Teil andere (ähnliche) Namen:

```
~/ .cdsinit, ~/ .cdslocal   Startup -Benutzer
./ .cdsinit, ./ .cdslocal   Startup -lokal
~/ .simrc, ~/ .simlocal    Sim.-Konfiguration -Benutzer
./ .simrc, ./ .simlocal    Sim.-Konfiguration -lokal
./ cds.lib                 Bibliothekskonfiguration
```

**CADENCE standalone** Start der Programme ohne Herstellerbibliotheken

```
> msfb [xterm]
```

Nach dem Programmstart (was etwas dauern kann) erscheint das Eingabefenster der Entwurfsumgebung ([msfb - Log:...]), über das dann alle anderen Programme (Editore, Simulatoren...) aufgerufen werden können.

<sup>2</sup>Anmerkung: ich habe deshalb versucht, die Beschreibung so allgemein wie möglich zu halten. Da aber sowohl die CADENCE Software, als auch die Design-Kits 1-2 mal pro Jahr aktualisiert werden sind kleine Fehler unvermeidlich! -AJM-

**AMS – Cadence Design-Kit** für den Standardzellentwurf und Full-Custom Design

```
> ams_cds -mode msfb -tech c35b4 [xterm]
```

Nach der Initialisierung des Verzeichnisses wird die Entwurfsumgebung gestartet. Über die Parameter können unterschiedliche Betriebsmodi und Prozesse eingestellt werden. Hier ist ein „mixed-signal“ (Standardzell-) Entwurf mit dem 0,35  $\mu\text{m}$  Prozess (4 Metalllagen) begonnen.

Der Prozessparameter wird nur für den ersten Aufruf benötigt, später kann der Entwurf direkt mit `ams_cds -mode msfb` begonnen werden.

## 2.2 CADENCE beenden

```
□ File - Exit... [msfb - Log:...]
```

```
≡ - bestätigen [Exit msfb?]
```

Damit das Programm nicht verlassen werden kann, ohne dass vorher gemachte Änderungen gespeichert worden sind, können beim Beenden noch Meldungen über ungesicherte Entwürfe erscheinen.

```
≡ - entsprechend ausfüllen [Save Cellviews]
```

**Achtung:** für alle noch nicht gesicherten Designs wird gefragt, was mit ihnen geschehen soll und ggf. werden sie automatisch gesichert. Wurden die Daten schon vorher unter einem neuen Namen gesichert (z.B. mit `Design - Save As...`), kann man diese Meldung ignorieren und die Sicherung der Daten ausschalten.

```
≡ Cancel [Save Display Information]
```

Informationen zur Darstellung der Layer sollten nicht gesichert werden!

## 2.3 Online-Dokumentation

**Start** In allen Fenstern befindet sich ein Help-Button über den die Online-Dokumentation gestartet werden kann.

```
□ Help - Cadence Documentation [⟨DF II window⟩]
```

Ansonsten lässt sich die komplette CADENCE Online-Dokumentation auch von einer Unix-Shell aus aufrufen.

```
> ichelp [xterm]
```

**Benutzung** Die Dokumentation ist als HTML-Text organisiert, der über einen eigenen Serverprozess verteilt wird, siehe Abbildung 3. Über die Oberfläche können die Manuals einzeln ausgewählt werden; die Darstellung erfolgt in einem normalen Webbrowser. Mit Hilfe der Suchfunktion, die eine entsprechende Maske öffnet, kann auch nach Stichworten gesucht werden.



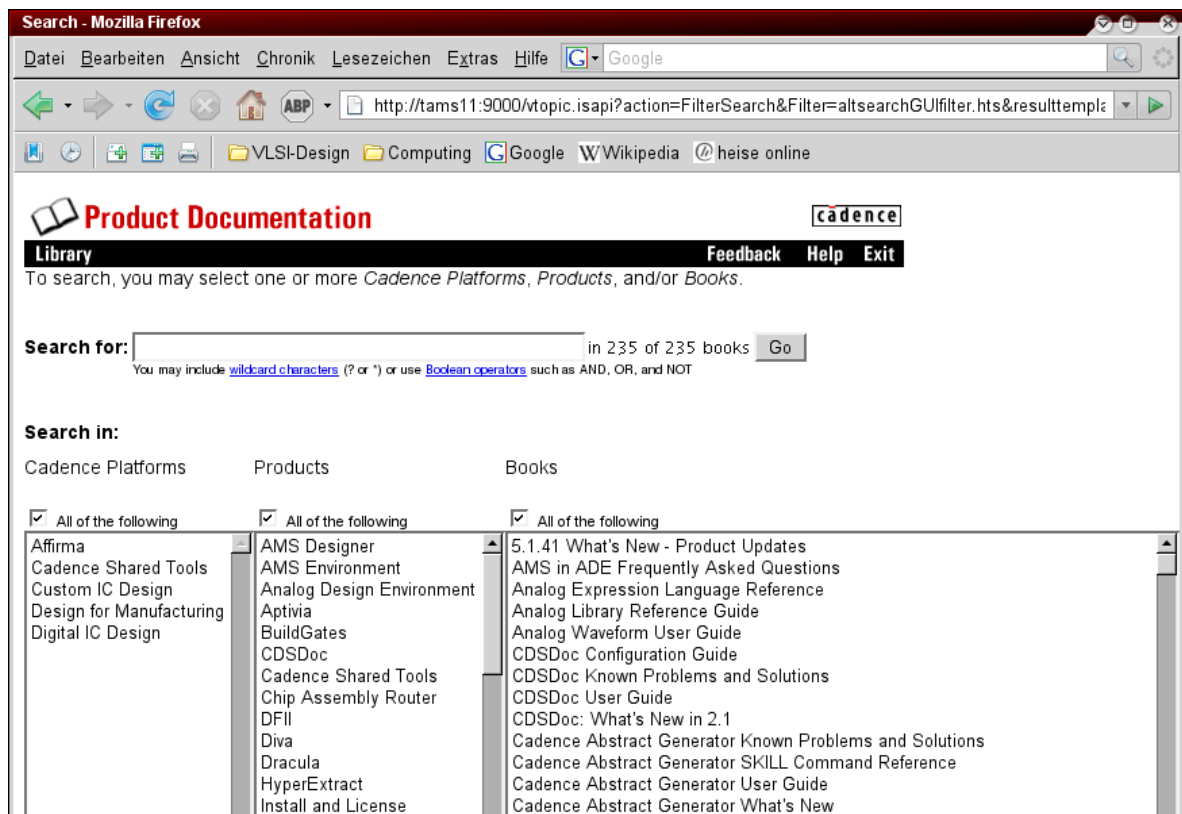
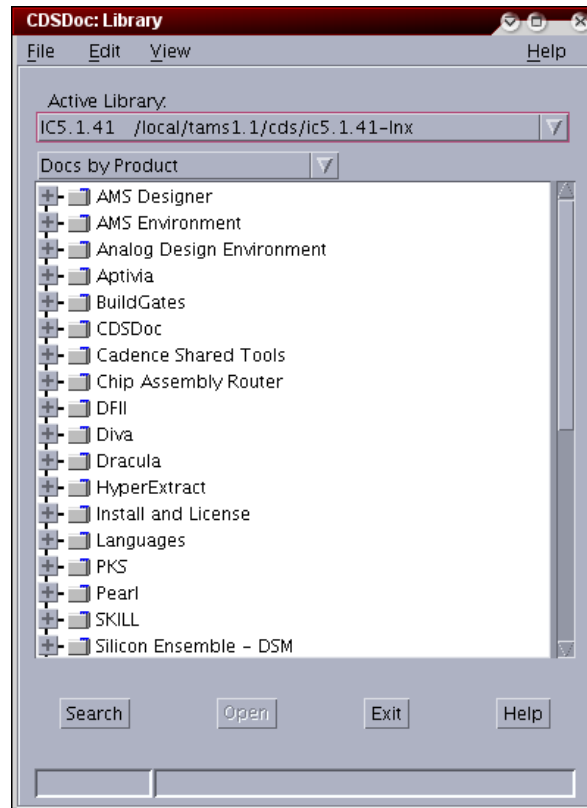


Abbildung 3: Online-Dokumentation

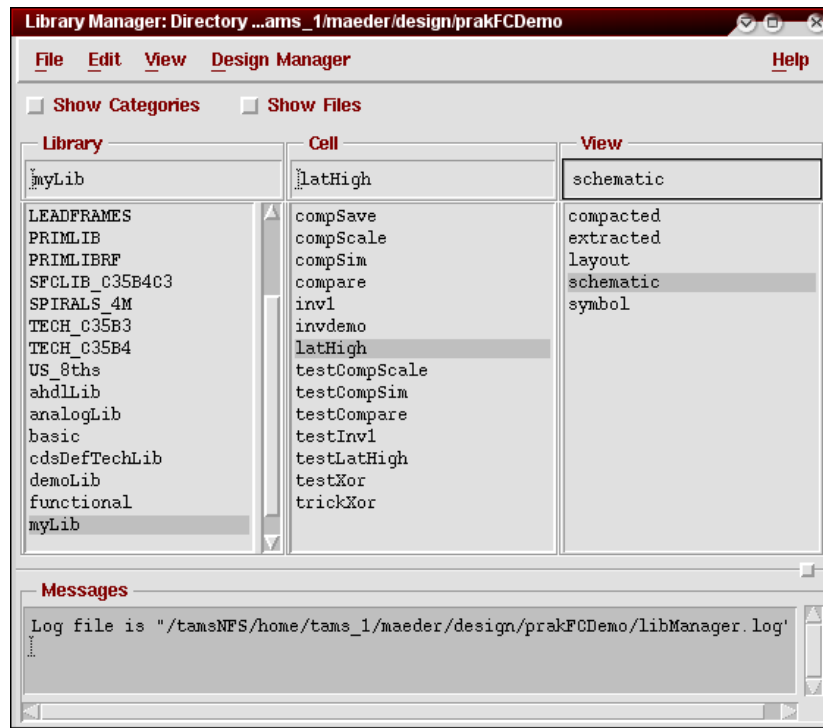


Abbildung 4: Der Library-Manager

## 2.4 Library-Manager

Dieses Werkzeug stellt die Bibliothekshierarchie grafisch dar und erlaubt es über Menüs die Designs zu bearbeiten (öffnen), zu kopieren, zu löschen...

Er ist außerdem die einfachste Möglichkeit um Bibliotheken, Zellen und Zellviews für den Eintrag in *Fill-Forms* auszuwählen. Bei vielen Formularen ist deshalb ein Button *Browse* vorhanden, der einen Library Browser – ähnlich dem Library Manager – startet und bei Auswahl von Elementen im Browser-Fenster den Eintrag in das entsprechende Feld des Formulars übernimmt.

- Tools - Library Manager... [msfb - Log:...]

Öffnet das Browser-Fenster, dabei werden alle Bibliotheken angezeigt, die sich im Suchpfad befinden. Um den Suchpfad zu verändern (was normalerweise nicht notwendig sein sollte) gibt es den Menüpunkt □ Edit - Library Path... [Library Manager].

**Benutzung** Die meisten Befehle des Library Managers werden kontextsensitiv direkt mit der Maus aufgerufen.

- ↑<sub>l</sub> *<item>* [Library Manager]

Zeigt die nächstniedrigere Ebene der Bibliothekshierarchie zu *<item>* an.

- ↑<sub>m</sub> *<item>* [Library Manager]

Erzeugt kontextsensitive Menüs (abhängig von der Ebene in der Bibliothek), die es erlauben den Inhalt der Bibliotheken zu verändern und Designs zu öffnen. Beispieler:

|   |   |
|---|---|
| $\uparrow_m\langle Cell \rangle$ - Copy...          | kopiert Zelle und Views (opt. hierarchisch)     |
| $\uparrow_m\langle Cell \rangle$ - Delete...        | löscht Zelle (und alle Views)                   |
| $\uparrow_m\langle View \rangle$ - Copy...          | kopiert einzelne View                           |
| $\uparrow_m\langle View \rangle$ - Delete...        | löscht einzelne View                            |
| $\uparrow_m\langle View \rangle$ - Open...          | öffnet Cellview, startet das „passenden“ Editor |
| $\uparrow_m\langle View \rangle$ - Open (Read-Only) | öffnet Cellview – read-only Zugriff             |

## 2.5 Designs bearbeiten und erzeugen

**existierende Entwürfe** Die einfachste Möglichkeit ein schon vorhandenes Design zu bearbeiten – „Design“ hier als Library Cell View –, ist die Auswahl mit dem Library Manager.

Alternativ dazu können auch die Menüs des des Library Managers oder der Entwurfsumgebung benutzt werden, um Entwürfe zu öffnen.

|   |                            |                  |
|---|----------------------------|------------------|
| <input type="checkbox"/> File - Open... |                            | [msfb - Log:...] |
| <input type="checkbox"/> Library Name   | = $\langle libId \rangle$  | [Open File]      |
| Cell Name                               | = $\langle cellId \rangle$ |                  |
| View Name                               | = $\langle viewId \rangle$ |                  |

Dabei stehen für Library und View entsprechende Auswahlfelder zur Verfügung.

**Bibliotheken erzeugen** Eigene Bibliotheken, die man braucht um seine Entwürfe durchzuführen, werden mit folgenden Befehlen eingerichtet. Sie werden dabei automatisch in den Suchpfad eingefügt.

|  |                             |                    |
|--|-----------------------------|--------------------|
| <input type="checkbox"/> File - New - Library... |                             | [msfb - Log:...]   |
| <input type="checkbox"/> Name                    | = $\langle libId \rangle$   | [New Library]      |
| Technology File                                  | = Attach...   Don't need... |                    |
| Design Manager                                   | = No DM                     |                    |
| <input type="checkbox"/> Attach To...            | = $\langle libId \rangle$   | [Attach Design...] |

**Achtung:** In der Regel muss eine prozessspezifische Technologiebibliothek des Chipherstellers angegeben werden Attach to an existing techfile:

|               |                    |                          |
|---------------|--------------------|--------------------------|
| TECH_C35B4    | AMS Hit-Kit        | CMOS 0,35 $\mu m$ 4 Met. |
| cdsDefTechLib | CADENCE-standalone |                          |

**Entwürfe erzeugen** In der eigenen Zellbibliothek können anschließend neue Entwürfe erzeugt werden.

|   |                             |                   |
|---|-----------------------------|-------------------|
| <input type="checkbox"/> File - New - Cellview... |                             | [msfb - Log:...]  |
| <input type="checkbox"/> Library Name             | = $\langle libId \rangle$   | [Create New File] |
| Cell Name   | = $\langle cellId \rangle$  |                   |
| Tool  | = $\langle toolSel \rangle$ |                   |

Das Feld View Name wird bei der Auswahl des Programms passend ausgefüllt:

|                           |                    |             |
|---------------------------|--------------------|-------------|
| $\langle toolSel \rangle$ | Virtuoso           | → layout    |
|                           | Composer-Schematic | → schematic |
|                           | Composer-Symbol    | → symbol    |

## 3 Der Layout-Editor

### 3.1 Starten, Sichern, Verlassen

**Starten** Das Öffnen einer layout-Zellview startet den Layout-Editor Virtuoso. Dazu gibt es drei, in Abschnitt 2.5 vorgestellte, Möglichkeiten:

1. über den Library Manager
2. als □ File - Open... [msfb - Log:...]
3. als □ File - New - Cellview... [msfb - Log:...]

**Entwurf Sichern** Sichert man seinen Entwurf unter einem anderen Namen, dann können später, beim Beenden der Entwurfsumgebung, entsprechende Meldungen über den ungesicherten Entwurf ignoriert werden.

- Design - Save/⊙ f2 [Virtuoso Layout...]
- Design - Save As... [Virtuoso Layout...]
- ≡ Library Name =  $\langle libId \rangle$  [Save As]
- Cell Name =  $\langle cellId \rangle$
- View Name = layout

#### Editor beenden

- Window - Close/⊙ ^w [Virtuoso Layout...]

### 3.2 Auswahl der Layer

Neben dem eigentlichen Layout-Fenster des Editors wird noch ein zweites Fenster erzeugt, auf dem die zur Verfügung stehenden Layer dargestellt sind: das LSW – Layer Selection Window. Die folgende Abbildung zeigt dieses Fenster und erklärt die Bedeutung der wichtigsten Layer des hier benutzten CMOS Prozesses.<sup>3</sup>

Außerdem besitzen die Layer auch noch eine Kennung, die die Art der Benutzung festlegt. Für die Eingabe von Geometrien werden hier nur Layer mit der Kennung drw (drawing) benutzt — Kennungen sind:

|     |    |          |
|-----|----|----------|
| drw | dg | drawing  |
| pin | pn | pin      |
|     | nt | net      |
|     | by | boundary |

Bevor Geometrien gezeichnet werden können, muss hier ein passendes Layer ausgewählt werden. Der nachfolgende Zeichenbefehl erzeugt die Geometrien auf diesem Layer.

↑<sub>l</sub>  $\langle layer \rangle$  [LSW]

Auswahl von  $\langle layer \rangle$ , dabei wird das ausgewählte Layer oben im LSW angezeigt.

↑<sub>m</sub>  $\langle layer \rangle$  [LSW]

Für  $\langle layer \rangle$  wird die Sichtbarkeit umgeschaltet.

↑<sub>r</sub>  $\langle layer \rangle$  [LSW]

Für  $\langle layer \rangle$  wird die Selektierbarkeit umgeschaltet.

<sup>3</sup>Die Funktionsweise und Herstellungstechnik von CMOS-Schaltungen werden hier als bekannt (aus den entsprechenden Vorlesungen) vorausgesetzt.

- Design - Tap/⊙ t [Virtuoso Layout...]  
Das aktive Layer wird durch Auswahl eines Layers im Layout bestimmt.

| LSW            |     |              |   |
|----------------|-----|--------------|---|
| Sort Edit Help |     |              |   |
| TECH_C35B4     |     |              |   |
| Show Objects   |     |              |   |
| AV NV AS NS    |     |              |   |
| Funktion       |     | für          |   |
| NTUB           | drw | N-Wanne      | P-Kanal Transistoren                        |
| DIFF           | drw | Dünnoxid     | alle Diffusionsgebiete (P- und N-Diff.)     |
| NPLUS          | drw | N-Diffusion  | N-Kanal Transistoren und Wannenkontakte     |
| PPLUS          | drw | P-Diffusion  | P-Kanal Transistoren und Substratkontakte   |
| MIDOX          | drw |              |   |
| HRES           | drw |              |   |
| POLY1          | drw | Polysilizium | Gates der Transistoren                      |
| POLY2          | drw | Polysilizium | Kondensatoren                               |
| CONT           | drw | Kontakt      | Metall 1 Anschlüsse auf Poly oder Diffusion |
| MET1           | drw | Metall       | Metall 1 Leitungen, Pins                    |
| MET1           | pin |              |   |
| VIA1           | drw | Kontakt      | Verbindungen zwischen Metall 1 und Metall 2 |
| MET2           | drw | Metall       | Metall 2 Leitungen, Pins                    |
| MET2           | pin |              |   |
| VIA2           | drw | Kontakt      | Verbindungen zwischen Metall 2 und Metall 3 |
| MET3           | drw | Metall       | Metall 3 Leitungen, Pins                    |
| MET3           | pin | ...          |   |
| VIA3           | drw |              |   |
| MET4           | drw |              |   |
| MET4           | pin |              |   |
| PAD            | drw |              |   |
| FIMP           | drw |              |   |
| NLDD           | drw |              |   |
| NLDD50         | drw |              |   |
| CELBOX         | drw |              |   |

Abbildung 5: Layer der 0,35 μm AMS-Prozesse

### 3.3 Benutzung der Maus

Die Belegung der Maustasten wird unten im Editor-Fenster angezeigt. Im allgemeinen gilt:

$\uparrow_l$   $\langle object \rangle$  [Virtuoso Layout...]

Auswahl (Selektion) von  $\langle object \rangle$  für nachfolgende Befehle wie das Löschen, Kopieren, Verschieben... Die Selektion arbeitet dabei folgendermaßen:

$\uparrow_l$ : ein einzelnes Element — wird der Cursor auf ein Design-Objekt bewegt, dann zeigt eine weiße Strichmarkierung an, was bei einer nachfolgenden Selektion ausgewählt wird. Dabei ist zu unterscheiden, ob vollständige Geometrien oder nur die Kanten von Objekten markiert sind!

$\uparrow_l$  festhalten: Selektionsfenster aufziehen

$\langle Shift \rangle + \uparrow_l$ : Selektion ergänzen

Die Anzahl der selektierten Objekte wird in der Statuszeile des Layout-Editors (oben im Fenster) angezeigt.

**Tipp:** selektierte Objekte werden hell umrahmt dargestellt. Sollten sich Objekte nicht, oder nur schwierig selektieren lassen, so kann man über die Selektierbarkeit einzelner Layer (siehe 3.2) eine Vorauswahl treffen.

$\uparrow_m$   $\langle object \rangle$  [Virtuoso Layout...]

Erzeugt ein (kontextsensitives) Menü, mit dem  $\langle object \rangle$  direkt manipuliert werden kann.

$\uparrow_r$   $\langle object \rangle$  [Virtuoso Layout...]

Wiederholt den letzten Befehl.

### 3.4 Fensterkontrolle

#### Scrolling

$\odot$  Pfeiltasten:  $\leftarrow$ ,  $\rightarrow$ ,  $\uparrow$ ,  $\downarrow$  [Virtuoso Layout...]

#### Vergrößern / Verkleinern

Window - Zoom In/ $\odot z$  [Virtuoso Layout...]

Window - Zoom In by 2/ $\odot \hat{z}$  [Virtuoso Layout...]

Window - Zoom To Grid/ $\odot \hat{g}$  [Virtuoso Layout...]

Window - Zoom To Sel Set/ $\odot \hat{t}$  [Virtuoso Layout...]

Window - Zoom Out by 2/ $\odot Z$  [Virtuoso Layout...]

#### weiteres

Window - Pan/ $\odot tab$  [Virtuoso Layout...]

Die Cursorposition wird der neue Fenstermittelpunkt.

Window - Fit All/ $\odot f$  [Virtuoso Layout...]

Das Design wird verkleinert/vergrößert, so dass es vollständig im Fenster sichtbar ist.

### 3.5 Eingabehilfen

**Undo / Redo** Die jeweils letzten 5 Befehle können wieder rückgängig gemacht werden.

- Edit - Undo/⊙ u [Virtuoso Layout...]
- Edit - Redo/⊙ U [Virtuoso Layout...]

**Maßstäbe** Um Abstände in dem Design zu messen, beispielsweise um die Einhaltung von Design Rules zu prüfen, können Maßstäbe erzeugt werden. Sie sind nur temporär vorhanden und werden nicht abgespeichert.

- Window - Create Ruler/⊙ k [Virtuoso Layout...]
- ≡ Keep Ruler = off | on [Create Ruler]
- Multi-segment Ruler = off | on
- Snap Mode = orthogonal | ...

Erzeugt einen oder mehrere Maßstäbe.

- Window - Clear All Rulers/⊙ K [Virtuoso Layout...]
- Löscht alle vorhandenen Maßstäbe im Design.

**Suchfunktion** Für die Suche nach Strukturen im Layout, hat man vielfältige Möglichkeiten Suchkriterien anzugeben, die gefundenen Elemente zu selektieren, zu verändern...

- Edit - Search.../⊙ S [Virtuoso Layout...]
  - ≡ - entsprechend ausfüllen [Search]
- |         |                                 |                                     |
|---------|---------------------------------|-------------------------------------|
| Search  | inst   contact   ...            | wonach wird gesucht                 |
|         | area   current   cellView   ... | wo wird gesucht                     |
|         | Add Criteria                    | Suchausdrücke werden gebildet       |
|         | Previous   Next                 | wechselt zwischen den Elementen     |
|         | Add Select   Select All         | Selektion gefundener Elemente       |
| Replace | none   cell name   ...          | zu verändernde Properties festlegen |
|         | Replace   Replace All           | Properties ändern                   |

### 3.6 Layout erzeugen

Fast alle Befehle des Layout-Editors sind so lange aktiv, bis sie explizit abgebrochen werden. Dazu muss entweder Esc eingegeben werden oder **Cancel** in dem entsprechenden Menü.

**Achtung:** die nachfolgenden Zeichenbefehle beziehen sich immer auf das gerade ausgewählte Layer (siehe 3.2), dementsprechend ist vorher eine geeignete Wahl zu treffen.

**Rechtecke** Wird mit der Maus aufgezogen.

- Create - Rectangle/⊙ r [Virtuoso Layout...]

**Polygone** Die Punkte werden der Reihe nach eingegeben, die zweimalige Eingabe des gleichen Punktes beendet den Befehl.

- Create - Polygon/⊙ P [Virtuoso Layout...]
- ≡ Snap Mode = orthogonal | L90.First [Create Polygon]

**Pfade** Die Punkte des Linienzuges werden der Reihe nach eingegeben, die zweimalige Eingabe des gleichen Punktes beendet den Befehl.

- Create - Path/⊙ p [Virtuoso Layout...]
- ≡ Width =  $\langle layer1Val \rangle$  [Create Path]
- Snap Mode = orthogonal | L90.First
- Change To =  $\langle layer1 \rangle$

Der Linienzug wird mit dem vorher eingestellten Layer (in [LSW]) begonnen. Anschließend werden die Punkte des Linienzuges für  $\langle layer1 \rangle$  eingegeben.

Während der Linienzug erzeugt wird, kann das Fenster mit den Optionen dazu benutzt werden, automatisch die Layer zu wechseln — die im Design notwendigen Kontakte werden automatisch erzeugt.

- ≡ Width =  $\langle layer2Val \rangle$  [Create Path]
- Snap Mode = orthogonal | L90.First
- Change To =  $\langle layer2 \rangle$
- Contact Justification =  $\langle alignment \rangle$

Es wird der Kontakt von  $\langle layer1 \rangle$  auf  $\langle layer2 \rangle$  erzeugt und  $\langle layer2 \rangle$  wird für die weitere Eingabe aktiv. Über die Angabe der Orientierung ( $\langle alignment \rangle$ ) kann der Kontakt relativ zum letzten Linienpunkt ausgerichtet werden.

### Kontakte

- Create - Contact.../⊙ o [Virtuoso]
- ≡ Auto Contact = off | on [Create Contact]
- Contact Type =  $\langle contact \rangle$
- Justification = centerCenter
- Width = 1
- Length = 1
- Rows =  $\langle nr \rangle$
- Columns =  $\langle nr \rangle$
- Rotate | Sideways | Upside Down

Sollen größere Kontakte realisiert werden, so werden diese als „Mehrfachkontakte“, über die Angabe von Rows oder Columns erzeugt. Ist Auto Contact aktiv, so können Kontakte automatisch an den Kreuzungspunkten von Pfaden generiert werden.

**Texte** Als Merkhilfe für den Designer.

- Create - Label.../⊙ l [Virtuoso Layout...]
- ≡ Label =  $\langle string \rangle$  [Create Label]
- Drafting = on
- Attach = off



### 3.7 Layout verändern

Nach der Eingabe der Befehle ist immer auszuwählen, welche Objekte bearbeitet werden sollen. Dies kann durch eine „normale“ Selektion geschehen, es ist aber auch möglich eine schon vorher selektierte Gruppe zu benutzen (Selektion 3.3).

Bei den Befehlen `move`, `copy`, `stretch` wird, vor dem endgültigen Absetzen, die Wirkung des Befehls durch eine helle Umrandung dargestellt.

#### Verschieben

- Edit - Move/⊙m [Virtuoso Layout...]
- ≡ Change Layer = off | on [Move]
- Snap Mode = anyAngle | diagonal | orthogonal | ...
- Rotate | Sideways | Upside Down

Ist Change Layer aktiv, so kann man in dem Layerauswahlfeld angeben, mit welchem Layer die Figur dargestellt wird.

#### Kopieren

- Edit - Copy/⊙c [Virtuoso Layout...]
- ≡ Change Layer = off | on [Copy]
- Snap Mode = anyAngle | diagonal | orthogonal | ...
- Rows = <nr>
- Columns = <nr>
- Rotate | Sideways | Upside Down

Ist Change Layer aktiv, so kann man in dem Layerauswahlfeld angeben, mit welchem Layer die kopierte Figur dargestellt wird.

**Verlängern / Verkürzen** Für Rechtecke, Polygone und die Endpunkte von Linienzügen können Kanten oder Eckpunkte verschoben werden. Es können aber auch ganze Bereiche modifiziert werden; dabei werden Kanten verlängert, die die Selektion schneiden, während Objekte, die sich vollständig in der Selektionsbox befinden, verschoben werden.

- Edit - Stretch/⊙s [Virtuoso Layout...]
- ≡ Lock Angles = on [Stretch]
- Snap Mode = anyAngle | diagonal | orthogonal | ...

**Form verändern** Zu bestehenden Rechtecken oder Polygone werden weitere Rechtecke hinzugefügt. Dazu wird an ein selektiertes Element ein schneidendes Rechteck angesetzt und mit dem entsprechenden Layer gefüllt.

- Edit - Reshape/⊙R [Virtuoso Layout...]
- ≡ Reshape Type = rectangle [Reshape]
- Snap Mode = anyAngle | diagonal | orthogonal | ...

#### Löschen

- Edit - Delete/⊙del [Virtuoso Layout...]

### 3.8 Hierarchie

**Erzeugen** Durch Instanziierung anderer Designs (layout-View) wird eine Hierarchie aufgebaut.

- Create - Instance.../⊙ i [Virtuoso Layout...]
- ≡ Library =  $\langle libId \rangle$  [Create Instance]
- Cell =  $\langle cellId \rangle$
- View = layout
- Names =  $\langle instIdLis \rangle$
- Rows =  $\langle nr \rangle$
- Delta Y =  $\langle nr \rangle$
- Columns =  $\langle nr \rangle$
- Delta X =  $\langle nr \rangle$
- Magnification = 1
- Rotate | Sideways | Upside Down

**Kennzeichnung der Anschlüsse** Die Anschlusspunkte der Schaltung spielen, im Gegensatz zum Schematic, keine direkte Rolle in der (Layout-) Hierarchie. Das liegt allerdings daran, dass Layouthierarchien nur die geometrische Information umfassen.

Die Anschlusspunkte werden erst beim Wechsel der *Abstraktionsebene* – durch Extraktion wird ein Schematic aus elektrischen Bauelementen erzeugt – benötigt, um Netze in einer Simulation ansprechen zu können. Diese *Pins* werden als Netznamen in der Extraktion behandelt und stellen später die elektrischen Anschlüsse der Schaltung dar.

Von den unterschiedlichen Möglichkeiten Pins zu erzeugen, wird hier die Methode durch Zeichnen eines Rechtecks beschrieben. Wie bei den anderen Zeichenbefehlen *muß vorher* ein entsprechendes Layer eingestellt worden sein, für die 0,35 µm AMS-Prozesse sind dies die Metalllayer MET1...MET4 mit der Kennung pin.

- Create - Pin.../⊙ ^p [Virtuoso Layout...]
- Startet das Menu Create Symbolic Pin, so ist dort shape pin auszuwählen.
- ≡ Terminal Names =  $\langle pinIdLis \rangle$  [Create Shape Pin]
- Mode = rectangle
- Display Pin Name = on
- I/O Type = input | output | inputOutput

| Konventionen: | Pin                 | I/O Type    | Terminal Names       |
|---------------|---------------------|-------------|----------------------|
|               | Eingänge            | input       | beliebiger Name      |
|               | Ausgänge            | output      | beliebiger Name      |
|               | Spannungsversorgung | inputOutput | festgelegt: vdd, gnd |

Die Pins müssen dabei über bestehendes Metall mit drw-Kennung gezeichnet werden.

**Traversieren** Ausgehend von dem ursprünglichen Layout kann die Hierarchie durchlaufen werden, dabei ist es auch möglich Teile zu editieren.

- Design - Hierarchy - Descend/⊙ X [Virtuoso Layout...]  
Abstieg innerhalb der Hierarchie, die selektierte Zelle wird in den Layout-Editor geladen.
- Design - Hierarchy - Edit In Place/⊙ x [Virtuoso Layout...]  
Die selektierte Zelle wird editierbar gemacht, bleibt aber in der Umgebung des derzeitigen Designs sichtbar.
- Design - Hierarchy - Return/⊙ B [Virtuoso Layout...]  
Rückkehr innerhalb der Hierarchie zur nächsthöheren Ebene.

### 3.9 Properties

Die Arbeitsweise vieler CADENCE-Programme wird durch Eigenschaften der Designs, bzw. deren Elemente, beeinflusst. Diese Properties können angesehen und modifiziert werden.

**Layoutelemente** So lassen sich beispielsweise die Layer gezeichneter Rechtecke oder Linienzüge über die Properties nachträglich ändern.

- Edit - Properties.../⊙ q [Virtuoso Layout...]
  - ≡ –ansehen oder ändern [Edit Properties]
- Sind mehrere Elemente selektiert, so kann in dem Fenster über **Next** und **Previous** direkt zwischen der selektierten Elementen umgeschaltet werden.

**Design** Properties des aktuellen Designs kann man sich mit folgendem Befehl ansehen:

- Design - Properties.../⊙ Q [Virtuoso Layout...]
- ≡ –ansehen oder ändern [Edit Cellview Properties]

## 4 Der Schematic-Editor

### 4.1 Starten, Verlassen

**Starten** Das Öffnen einer schematic-Zellview startet den Schematic-Editor Composer-Schematic. Dazu gibt es drei, in Abschnitt 2.5 vorgestellte, Möglichkeiten:

1. über den Library Manager
2. als □ File - Open... [msfb - Log:...]
3. als □ File - New - Cellview... [msfb - Log:...]

#### Editor beenden

- Window - Close [Virtuoso Schematic...]

### 4.2 Rule Check und Sichern

**Schematic Rule Check** Zur Überprüfung eines Schematic sollte ein SRC durchgeführt werden, um beispielsweise offene Eingänge, Netze ohne Treiber und ähnliche Fehlerquellen zu finden.

- Check - Current Cellview/⊙ x [Virtuoso Schematic...]

Ausgehend von dem aktuellen Design kann auch die komplette Hierarchie geprüft werden, dazu sind die folgenden Schritte durchzuführen:

- Check - Hierarchy... [Virtuoso Schematic...]
- ≡ - bestätigen [Check Hierarchy]

Der Rule Check wird bottom-up für die Designhierarchie durchgeführt, dabei werden die (Sub-) Designs nach dem Test gesichert.

#### Entwurf Sichern

- Design - Check and Save/⊙ X [Virtuoso Schematic...]

- Design - Save/⊙ S [Virtuoso Schematic...]

- Design - Save As.../⊙ ^s [Virtuoso Schematic...]

- ≡ Library Name =  $\langle libId \rangle$  [Save As]
- Cell Name =  $\langle cellId \rangle$
- View Name = schematic

### 4.3 Benutzung der Maus

Die Belegung der Maustasten wird unten im Editor-Fenster angezeigt. Im allgemeinen gilt:

$\uparrow_l$   $\langle object \rangle$  [Virtuoso Schematic...]

Auswahl (Selektion) von  $\langle object \rangle$  für nachfolgende Befehle wie das Löschen, Kopieren, Verschieben... Die Selektion arbeitet dabei folgendermaßen:

$\uparrow_l$ : ein einzelnes Element — wird der Cursor auf ein Design-Objekt bewegt, dann zeigt eine weisse Strichmarkierung an, was bei einer nachfolgenden Selektion ausgewählt wird.

$\uparrow_l$  festhalten: Selektionsfenster aufziehen

$\langle Shift \rangle + \uparrow_l$ : Selektion ergänzen

Selektierte Objekte werden hell umrahmt dargestellt. Durch Angabe eines Filters kann die Selektion auf bestimmte Objekte eingeschränkt werden, siehe 4.5. Die Anzahl der selektierten Objekte wird in der Statuszeile des Schematic-Editors (oben im Fenster) angezeigt.

$\uparrow_m$   $\langle object \rangle$  [Virtuoso Schematic...]

Erzeugt ein (kontextsensitives) Menü, mit dem  $\langle object \rangle$  direkt manipuliert werden kann.

$\uparrow_r$   $\langle object \rangle$  [Virtuoso Schematic...]

Wiederholt den letzten Befehl.

### 4.4 Fensterkontrolle

#### Scrolling

$\odot$  Pfeiltasten:  $\leftarrow$ ,  $\rightarrow$ ,  $\uparrow$ ,  $\downarrow$  [Virtuoso Schematic...]

#### Vergrößern / Verkleinern

$\square$  Window - Zoom - Zoom In/ $\odot$  z [Virtuoso Schematic...]

$\square$  Window - Zoom - Zoom In by 2/ $\odot$ ] [Virtuoso Schematic...]

$\square$  Window - Zoom - Zoom Out by 2/ $\odot$  [ [Virtuoso Schematic...]

$\odot$   $\hat{z}$  Zoom Out [Virtuoso Schematic...]

#### weiteres

$\square$  Window - Pan [Virtuoso Schematic...]

Die Cursorposition wird der neue Fenstermittelpunkt.

$\odot$  v [Virtuoso Schematic...]

Für eine Punkt des Designs wird angegeben, wo er in dem Fenster platziert werden soll (relative Pan).

$\square$  Window - Fit/ $\odot$  f [Virtuoso Schematic...]

Das Design wird verkleinert/vergrößert, so dass es vollständig im Fenster sichtbar ist.

### 4.5 Eingabehilfen

**Undo / Redo** Die jeweils letzten 5 Befehle können wieder rückgängig gemacht werden.

- Edit - Undo/⊙ u [Virtuoso Schematic...]
- Edit - Redo/⊙ U [Virtuoso Schematic...]

**Selektion** Die Selektierbarkeit kann über einen Filter kann die Selektierbarkeit (Selektion 4.3) auf bestimmte Objektgruppen eingeschränkt werden — ähnlich der Selektierbarkeit von Layern im Layout-Editor.

- Edit - Select - Filter.../⊙ ^f [Virtuoso Schematic...]
- ≡ Area Partial Selection =off [Schematic Selection Filter]
- Schematic Objects =wire|pin|instance|...
- Instance Objects =name|pin|...

Ein globale Suchfunktion für das aktuelle Schematic:

- Edit - Select - All... [Virtuoso Schematic...]
- ≡ Schematic Objects =wire|pin|instance|... [Schematic Select All]
- Instance Objects =name|pin|...

Eine Selektion ist auch über die Properties der Elemente möglich. Dabei werden Ausdrücke gebildet, über die alle dementsprechenden Objekte des Schematic selektiert, bzw. deselektiert werden:

- Edit - Select - By Property... [Virtuoso Schematic...]
- ≡ Search for =⟨propId⟩ ⟨op⟩ ⟨propVal⟩ [Schematic Select By ...]
- Form Action =select|deselect
- Schematic Objects =wire|pin|instance|...
- Instance Objects =name|pin|...

**Suchfunktionen** Die beiden Suchfunktionen arbeiten jeweils in zwei Schritten. Zuerst werden Objekte (innerhalb der Hierarchie) gesucht. Anschließend können die gefundenen Objekte des Suchergebnisses einzeln selektiert (Find) oder verändert (Replace) werden.

Suche nach Objekten und deren Selektion:

- Edit - Search - Find... [Virtuoso Schematic...]
- ≡ – entsprechend ausfüllen [Schematic Find]
- ⟨propId⟩ ⟨op⟩ ⟨propVal⟩ wonach wird gesucht
- Search Scope wo wird gesucht
- Object Filter
- Previous | Next
- Select
- schränkt Suche auf bestimmte Objekte ein
- wechselt zwischen gefundenen Elementen
- selektiert ein gefundenes Objekt

Suche nach Objekten und Modifikation durch Veränderung von Properties:

|   |  |
|---|--|
| <input type="checkbox"/> Edit - Search - Replace... | [Virtuoso Schematic...]                  |
| ≡ - entsprechend ausfüllen                          | [Schematic Replace]                      |
| <propId> <op> <propVal>                             | wonach wird gesucht                      |
| Search Scope  | wo wird gesucht                          |
| Object Filter                                       | schränkt Suche auf bestimmte Objekte ein |
| Replace with: <propId> <propVal>                    | wodurch wird ersetzt                     |
| Replace   | ersetzt aktuelles Objekt                 |
| Skip  | keine Ersetzung                          |
| Replace All   | ersetzt alle gefundenen Objekte          |

### 4.6 Schematic zeichnen

Ein Schematic besteht aus Komponenten-Symbolen (Zellview symbol) und deren Verbindung untereinander. Diese Symbole können

1. aus vorgegebenen Zellbibliotheken kommen (Gatterbibliotheken mit Standardzellen, Bibliotheken mit elektrischen Bauteilen...).
2. aus selbst entworfenen Schematics generiert worden sein. Die Verwendung solcher *eigener* Symbole entspricht dem Aufbau einer Hierarchie im Design.

Fast alle Befehle des Schematic-Editors sind so lange aktiv, bis sie explizit abgebrochen werden, entweder mit Esc oder **Cancel** in dem entsprechenden Menü.

#### Symbole instanziiieren

|  |                           |
|--|---------------------------|
| <input type="checkbox"/> Add - Instance.../⊙ i | [Virtuoso Schematic...]   |
| ≡ Library                                      | = <libId> [Add Component] |
| Cell   | = <cellId>                |
| View   | = symbol                  |
| Names  | = <instIdLis>             |
| Rows   | = <nr>                    |
| Columns  | = <nr>                    |
| Rotate   Sideways   Upside Down                |                           |

Die Komponentenauswahl geht am einfachsten grafisch mit **Browse**.

**Verbindungen erzeugen** Über Leitungen werden die Anschlüsse der Instanzen miteinander verbunden. Für die Darstellung von Bussen verwendet man dabei üblicherweise breitere Leitungen. Die einzelnen Punkte der Leitung werden der Reihe nach eingegeben. Geht eine Leitung an einen Anschluss eines Symbols, so wird sie abgesetzt, ansonsten muss der gleiche Punkt zweimal eingegeben werden um die Leitung zu beenden.

|  |                            |
|--|----------------------------|
| <input type="checkbox"/> Add - Wire (narrow)/⊙ w | [Editing]                  |
| <input type="checkbox"/> Add - Wire (wide)/⊙ W   | [Editing]                  |
| ≡ Draw Mode                                      | = route   ... [Add Wire]   |
| Route Method                                     | = full   direct   flight   |
| Width  | = 0   0.0625 (narrow/wide) |

- ⊙ s [Editing]  
 Wenn sich eine Leitung in der Nähe von Anschlusspunkten oder anderen Leitungen befindet, dann wird durch ein Rautensymbol ein *möglicher* Anfangs- oder Endpunkt gekennzeichnet. Durch Eingabe des Bindkeys wird die Leitung dort angeschlossen.

**Netznamen** Sollen explizite Namen für Netze vergeben werden, so wird erst das Label platziert und anschließend einer Leitung zugeordnet.

- Add - Wire Name.../⊙ l [Virtuoso Schematic...]
- ≡ Names =  $\langle netIdLis \rangle$  [Add Wire Name]
- Bus Expansion = off | on
- Placement = single | multiple
- Purpose = label | alias
- Rotate

**Texte** Als Merkhilfe für den Designer.

- Add - Note - Note Text.../⊙ L [Virtuoso Schematic...]
- ≡ Note Text =  $\langle text \rangle$  |  $\langle string \rangle$  [Add Note Text]
- Rotate

#### 4.7 Schematic verändern

Nach der Eingabe der Befehle ist immer auszuwählen, welche Objekte bearbeitet werden sollen. Dies kann durch eine „normale“ Selektion geschehen, es ist aber auch möglich eine schon vorher selektierte Gruppe zu benutzen (Selektion 4.3).

Bei den Befehlen move, copy, stretch wird, vor dem endgültigen Absetzen, die Wirkung des Befehls durch eine helle Umrandung dargestellt.

**Verschieben** Der Stretch-Befehl verschiebt Elemente (Symbole), wobei Leitungen die an die Symbole angeschlossen sind, mitgeführt (verlängert, bzw. neu gelegt) werden.

- Edit - Stretch/⊙ m [Virtuoso Schematic...]
- ≡ Snap Mode = anyAngle | diagonal | orthogonal [Stretch]
- Route Method = full | direct | flight
- Rotate | Sideways | Upside Down

Im Gegensatz dazu verschiebt der Move-Befehl Elemente, lässt Leitungen aber liegen.

- Edit - Move/⊙ M [Virtuoso Schematic...]
- ≡ Snap Mode = anyAngle | diagonal | orthogonal [Move]
- Rotate | Sideways | Upside Down

#### Rotieren

- Edit - Rotate [Virtuoso Schematic...]
- ≡ Rotate = on | off [Rotate]
- UpsideDown = on | off
- SideWays = on | off



**Kopieren**

- Edit - Copy/⊙ c [Virtuoso Schematic...]
- ≡ Snap Mode = anyAngle | diagonal | orthogonal [Copy]
- Columns =  $\langle nr \rangle$
- Rows =  $\langle nr \rangle$
- Rotate | Sideways | Upside Down

**Löschen**

- Edit - Delete/⊙ del [Virtuoso Schematic...]

**4.8 Hierarchie**

**Erzeugen** Die eigentliche Instanziierung durch die Verwendung zuvor generierter Symbole wurde oben schon erläutert.

**Kennzeichnung der Anschlüsse** Die Anschlüsse der Schaltung, die in einer Hierarchie verwendet werden, müssen in dem Schematic als Pins gekennzeichnet werden. Bei einer Simulation der Schaltung können nur diese Pins angesprochen werden.

- Add - Pin.../⊙ p [Virtuoso Schematic...]
- ≡ Pin Names =  $\langle pinIdLis \rangle$  [Add Pin]
- Direction = input | output | inputOutput | switch
- Usage = schematic
- Bus Expansion = on | off
- Placement = single | multiple
- Rotate | Sideways | Upside Down

**Symbolgenerierung** Um Symbole für den Aufbau von Hierarchien zu erzeugen, gibt es mehrere Möglichkeiten, die im einzelnen in den Abschnitten 4.8.1 und 4.8.2 vorgestellt sind.

**Traversieren** Ausgehend von dem aktuellen Schematic kann die Hierarchie durchlaufen, und dort sogar Änderungen vorgenommen werden.

- Design - Hierarchy - Descend Edit.../⊙ E [Virtuoso Schematic...]
- oder
- Design - Hierarchy - Descend Read.../⊙ e [Virtuoso Schematic...]
- ≡ View Name = schematic | symbol | layout | ... [Descend]
- Lädt das selektierte Design in den entsprechenden Editor — dies wird normalerweise ein Schematic sein.
- Design - Hierarchy - Return/⊙ ^ e [Virtuoso Schematic...]
- Rückkehr innerhalb der Hierarchie zur nächsthöheren Ebene.

### 4.8.1 Bottom-up Design

Aus einer vorhandenen Zellview `schematic` kann, über die Information der Pins, automatisch ein Symbol generiert werden, das dann in der nächsthöheren Hierarchieebene benutzt werden kann.

|                          |   |   |
|--------------------------|---|---|
| <input type="checkbox"/> | Design - Create Cellview - From Cellview... | [Virtuoso Schematic...]                               |
| ≡                        | Library Name                                | = <code>&lt;libId&gt;</code> [Cellview From Cellview] |
|                          | Cell Name                                   | = <code>&lt;cellId&gt;</code>                         |
|                          | From View Name                              | = <code>schematic</code>                              |
|                          | Tool / Data Type                            | = <code>Composer-Symbol</code>                        |
|                          | Display Cellview                            | = <code>off on</code>                                 |
|                          | Edit Options                                | = <code>off on</code>                                 |

Edit Options öffnet ein Formularfenster ([Symbol Generation Options]) in der man beispielsweise die Anordnung der Pins ändern kann.

Display Cellview startet anschließend den Symboleditor Virtuoso-Symbol.<sup>4</sup>

### 4.8.2 Top-down Design

Wenn kein Schematic vorhanden ist, wie beim top-down Entwurf oder wenn ein Symbol zu einer layout-View generiert werden soll, kann ein Symbol aus einer Textliste der Pins generiert werden.

|                          |   |  |
|--------------------------|---|--|
| <input type="checkbox"/> | Design - Create Cellview - From Pin List... | [Virtuoso Schematic...]                                  |
| ≡                        | Input Pins                                  | = <code>&lt;pinIdLis&gt;</code> [Cellview From Pin List] |
|                          | Output Pins                                 | = <code>&lt;pinIdLis&gt;</code>                          |
|                          | IO Pins                                     | = <code>&lt;pinIdLis&gt;</code>                          |
|                          | Switch Pins                                 | = <code>&lt;pinIdLis&gt;</code>                          |
|                          | Library Name                                | = <code>&lt;libId&gt;</code>                             |
|                          | Cell Name                                   | = <code>&lt;cellId&gt;</code>                            |
|                          | Tool / Data Type                            | = <code>Composer-Symbol</code>                           |
|                          | Display Cellview                            | = <code>off on</code>                                    |
|                          | Edit Options                                | = <code>off on</code>                                    |

Edit Options öffnet ein Formularfenster ([Symbol Generation Options]) in der man beispielsweise die Anordnung der Pins ändern kann.

Display Cellview startet anschließend den Symboleditor Virtuoso-Symbol.<sup>4</sup>

---

<sup>4</sup>Der Symboleditor ist im Rahmen dieser Kurzeinführung nicht weiter beschrieben, da die automatisch generierten Symbole „im Regelfall“ ohne Probleme benutzt werden können — ansonsten muss auf die CADENCE Online-Dokumentation verwiesen werden.

Eine zweite Möglichkeit einen top-down Entwurf durchzuführen hat man mit dem Block-Befehl. Mit seiner Hilfe können im Schematic automatisch `symbol`-Views für referenzierte Komponenten erzeugt und in dem gerade aktuellen Schematic instanziiert und untereinander verbunden werden.

- Add - Block.../⊙ b [Virtuoso Schematic...]
- ≡ Library = <libId> [Add Block]
- Cells = <cellId>
- View = symbol
- Names = <instIdLis>
- Pin Name Prefix = <pinId>
- Block Shape = freeform|...

Nach seiner Generierung hat der Block noch keinerlei Ein- und Ausgänge; wird der Block in dem Schematic an Leitungen angeschlossen, so werden die Anschlüsse (Pins) automatisch generiert. Dabei werden die Namen der Pins aus <pinId> und einer Nummer gebildet. Um „sinnvolle“ Namen zu vergeben ist eine Nachbearbeitung des Symbols mit dem Symboleditor notwendig.

Eine schematic-View dieses Blocks muss man dann später noch erzeugen/bearbeiten.

### 4.9 Properties

Die Arbeitsweise vieler CADENCE-Programme wird durch Eigenschaften der Designs, bzw. deren Elemente, beeinflusst. Diese Properties können angesehen und modifiziert werden.

#### Schematic-Objekte

- Edit - Properties - Objects.../⊙ q [Virtuoso Schematic...]
- ≡ -ansehen oder ändern [Edit Object Properties]

Für Elemente des Schematic (Instanzen, Leitungen, Pins, Label...). Sind mehrere Elemente selektiert, so kann in dem Fenster über `Next` und `Previous` direkt zwischen der selektierten Elementen umgeschaltet werden.

**Design** Die Properties des aktuellen Designs kann man sich mit folgendem Befehl ansehen:

- Edit - Properties - Cellview.../⊙ Q [Virtuoso Schematic...]
- ≡ -ansehen oder ändern [Edit Cellview Properties]

**Properties verändern** In den entsprechenden Formularfenstern lassen sich neben den vordefinierten auch eigene Properties eintragen und verändern.

- ≡ Add [Edit Object/Cellview Properties]
- ≡ Name = <propId> [Add Property]
- Type = int | float | string | ...
- Value = <propVal>
- Choices = <val1 val2 val3...>

|                             |                                       |                                   |
|-----------------------------|---------------------------------------|-----------------------------------|
| ≡ <b>Modify</b>             |                                       | [Edit Object/Cellview Properties] |
| ≡ $\langle propSel \rangle$ |                                       | [Modify Property]                 |
| Type                        | = int   float   string   ...          |                                   |
| Name                        | = $\langle propId \rangle$            |                                   |
| Value                       | = $\langle propVal \rangle$           |                                   |
| Choices                     | = $\langle val1 val2 val3... \rangle$ |                                   |
|                             |                                       |                                   |
| ≡ <b>Delete</b>             |                                       | [Edit Object/Cellview Properties] |
| ≡ $\langle propSel \rangle$ |                                       | [Delete Property]                 |
| Delete All                  | = off   on                            |                                   |