Datentransfer Synthese → Layout / Plazierung & Verdrahtung
– CADENCE DF II – A. Mäder

In dieser Beschreibung werden die grundlegenden Schritte beschrieben, wie

- eine EDIF-Datei als Ausgabe des SYNOPSYS DESIGN-ANALYZER in eine Datenbasis für CADENCE umgewandelt wird.
- das physikalische Layout (Placement & Routing) mit CADENCE DF II durchgeführt wird.

Dabei sind jeweils nur die einfachsten Schritte gezeigt — bei Problemfällen, die bei größeren Entwürfen (wahrscheinlich) auftreten werden, sei auf die CADENCE Manuals verwiesen...
In folgenden Kurzanleitungen finden sich zusätzliche Informationen:

- „Dokumentation zu CADENCE“ : allgemeine Konzepte, Benutzung des Layout-Editors
- „Synthese von Gatternetzlisten aus VHDL-Beschreibungen“ : EDIF-Datei generieren

Bei der Beschreibung der Benutzereingaben gelten die schon vorher verwendeten Symbole.

1 CADENCE DF II

Hier sind bei der Beschreibung von CADENCE DF II nur die Schritte zur Durchführung einer (automatischen) Plazierung und Verdrahtung bei Benutzung der ES 2-Zellbibliotheken beschrieben.

1. Start des Systems

   > `es2_cdk` [xterm]

   Achtung: Da dieser Befehl ein Skript startet, darf er nicht im Hintergrund ausgeführt werden. Beim ersten Aufruf (in einem Directory) wird nach dem Prozeß -`ecpd..`- und dem Programm -`icfb`- gefragt, dabei sind bei beiden Fragen die voreingestellten Werte zu bestätigen.

1.1 Datentransfer

Der erste Schritt des Datentransfers, die Generierung der Schematics und Ausgabe der EDIF-Datei, wurde schon in der Beschreibung zum Umgang mit den SYNOPSYS Synthesewerkzeugen angesprochen.

Bei dem EDIF Transfer ist es leider nicht möglich Busse als solche im Schematic beizubehalten — Probleme traten bei Aufspaltungen von Bussen auf, da die Abzweigung einzelner Signale über *Ripper* nicht korrekt (automatisch erzeugt) wird. Bei der jetzigen Vorgehensweise bleiben Busse zwar erhalten, sind in den Schematics aber als einzelne Leitungen dargestellt.

1. Einlesen der EDIF-Datei

   □ `ES2 0.7um CMOS - Edifin SYNOPSYS -> DFII` [ES2 0.7um ...]

   ≡ `EDIF input-file` `=<edif file name>` [EDIF data transfer: ...]

↑$_l$ `Apply` [EDIF data transfer: ...]

Anschließend muß man etwas warten während der Datentransfer stattfindet. Dabei erscheinen in dem CADENCE Eingabefenster ([ES2 0.7um ...]) mehrere Ausgaben.

Der Vorgang ist abgeschlossen, wenn die folgende Meldung erscheint:
`-I- Edifin process completed`

↑$_l$ `View Log` [EDIF data transfer: ...]

Die bei der Übersetzung entstandene log-Datei ist unbedingt auf Fehlermeldungen und Warnungen hin zu kontrollieren.

Dabei können folgende Meldungen ignoriert werden:

```
error   ... Port PWR/GND does not exist in view symbol of cell LIB...
warning ... Load tech file named       ... failed because
warning ... library named DESIGNS exists
*WARNING* cannot add reference library ... library already exists
*WARNING* Capability devComp is not active in this session
```

□ `File - Close Window` [<edif file name>.log]

↑$_l$ `Cancel` [EDIF data transfer: ...]

Konnte der Datentransfer nicht richtig durchgeführt werden, so sind die Fehler in der EDIF-Datei, beziehungsweise bei deren Generierung, zu suchen.

Im fehlerfreien Fall wurde der Entwurf (mit der gesamten Hierarchie) in der Bibliothek `DESIGNS` gespeichert. Falls diese Bibliothek beim Start des Datentransfers noch nicht existierte, wurde sie automatisch erzeugt.

2. Top-level Schematic öffnen

□ `Open - Design...` [ES2 0.7um ...]

≡ `Library Name` = `DESIGNS` [Open Design]
`Cell Name` = `<top-level cell>`
`View Name` = `schematic`

Beim Öffnen synthethisch erzeugter Schematics werden folgende Meldungen auftreten: `Info: Breaking .-point wire ... into segments`, diese können ignoriert werden.

3. Test der Daten[1]
Traten bei der EDIF-Ausgabe in SYNOPSYS DESIGN-ANALYZER oder beim Datentransfer Fehler (bzw. unverständliche Meldungen) auf, so empfiehlt es sich den Entwurf dem Schematic-Rule-Check zu unterziehen. Dieser Test kann, ausgehend von dem top-level Schematic, hierarchisch über das gesamte Design laufen.

□ `Check - Hierarchy ...` [Editing:...]

≡ – bestätigen [Check Hierarchy]

Während des SRC werden die Programmeldungen in dem CADENCE Eingabefenster ([ES2 0.7um ...]) ausgegeben.

[1]optional: nicht notwendig, aber immer empfehlenswert

Traten Fehler auf, so sind diese vor der weiteren Bearbeitung des Entwurfs zu beheben. Um die Hierarchie zu traversieren können folgende Befehle benutzt werden:

$\uparrow_l$ `<lower-level cell>` [Editing:...]

Referenziertes Element selektieren, das selektierte Element wird umrahmt dargestellt.

□ `Design - Hierarchy - Descend Edit...` / ⊙ `E` [Editing:...]

≡ – bestätigen [Descend]

Die `schematic`-View der selektierten Zelle wird zum Editieren geöffnet. Anschließend können die „Fehler“ in dem Entwurf korrigiert werden.

□ `Design - Check and Save` / ⊙ `X` [Editing:...]

Das Schematic wird, entsprechend den für den Prozeß geltenden Regeln, geprüft und anschließend gesichert.

□ `Design - Hierarchy - Return` / ⊙ `^e` [Editing:...]

Rücksprung innerhalb der Hierarchie.

1.2 Cadence-Layout

1. Einfügen noch fehlender Padzellen.
   Sofern die Padzellen nicht schon automatisch durch Synopsys eingefügt worden sind, müssen sie jetzt referenziert und angeschlossen werden. Dabei müssen die Ports der Schaltung, die jetzt schon vorhanden sind, über die Padzellen auf die internen Anschlüsse geführt werden.

   Auf alle Fälle sind die noch fehlenden Pads für die Versorgungsspannung in dem Schematic zu ergänzen. Die dafür benötigten Zellen sind unten aufgeführt.

   Achtung: für eine korrekte Plazierung und Verdrahtung ist es zwingend notwendig, eine Instanz von `LIBTOPNETS` in das Schematic einzufügen. Sie sorgt dafür, daß die verscheidenen Vdd- und Gnd-Netze richtig verdrahtet werden.

   □ `Add - Component...` / ⊙ `i` [Editing:...]

   ≡ `Library Name` = `<lib name>` [ES2 ... Library Menu]
   `Cell Name` = `<cell name>`
   `View Name` = `symbol`

   Erzeugen von Instanzen und Plazierung der Zellen im Schematic. Über das Instanziierungsmenü können die benötigten Instanzen erzeugt werden, anschließend wird die Instanziierung durch Eingabe von `Cancel` beendet.

Die benötigten Zellen lassen sich am einfachsten mit dem `Library Browser` auswählen. Sie finden sich unter:

`<lib name>`	`- path`	`<cell name>`	
`PadLib`		`LIBTOPNETS`	alle Vdd / Gnd Netze
`PadLib`	`- supply`	`LIBCRNGNDPY`	Gnd - Corner, für Pads
		`LIBCRNPWRPY`	Vdd - Corner, für Pads
		`LIBGNDCO`	Gnd, für Core
		`LIBGNDPY`	Gnd, für Pads
		`LIBPWRBK`	Vdd, für Megazellen
		`LIBPWRCO`	Vdd, für Core
		`LIBPWRPY`	Vdd, für Pads
`AnaLib`	`- supply`	`ANA...`	I wonder how you did this with SYNOPSYS...

☐ `Design - Check and Save` / ⊙ `X` [Editing:...]

Das Schematic kann nun geprüft und anschließend gesichert werden.

2. Erzeugen einer „flachen" Netzliste für P & R.

☐ `Tools - Floorplan/Schematics` [Editing:...]

☐ `Floorplan - Hierarchy Browser...` [Editing:...]

≡ – bestätigen [Open Hierarchy Browser]

$\uparrow_l$ `<top-level cell>` [Hierarchy Browser:...]

Selektion des „Gesamt"-ICs (Wechsel der Farbdarstellung).

☐ `Hierarchy - Properties - Set Master...` [2] [Hierarchy Browser:...]

≡ `Schematic/Netlist Views` = `schematic netlist pr.sch` [2] [Set Master Prop..]

Diese Property wird für die Generierung der flachen Netzliste bei analogen Zellen benötigt.

☐ `Hierarchy - Generate Physical Hierarchy` [Hierarchy Browser:...]

Der *flatten* Prozeß wird gestartet und erzeugt eine `autoLayout` View der Schaltung.

☐ `Window - Close` [Hierarchy Browser:...]

3. Start von Plazierung und Verdrahtung.

☐ `Design - Open...` [Editing:...]

≡ `Library Name` = `DESIGNS` [Open Design]
`Cell Name` = `<top-level cell>`
`View Name` = `autoLayout`

☐ `Tools - Floorplan/P&R - Cell Ensemble` [Editing:...]

Laden der flachen Netzliste und Initialisierung der P & R-Werkzeuge.

4. Zuordnung der Netzprioritäten
Dieser Schritt ist notwendig, um die korrekte Verdrahtung der Versorgungsleitungen (Breite der Vdd- und Gnd-Verbindungen) zu gewährleisten.

Während es üblicherweise genügt eine entsprechende Datei von den ES 2-Verzeichnissen zu laden, muß bei Verwendung von generierten Blöcken und deren Anschluß über `LIBPWRBK` Pad(s) diese Datei lokal angepaßt werden.

[2] optional: nur bei Verwendung analoger Zellen

☐ `Route - Modify Net - ES2 0.7um CMOS - cp '...'` [Editing:...]

☐ `Route - Modify Net - ES2 0.7um CMOS - edit '...'` [3] [Editing:...]

Für jedes `supply`-Netz einer generierten Zelle sind folgende Zeilen hinzuzufügen:

```
; Save net: <P&R net-name>
   rep=geGetWindowCellView(getCurrentWindow())
   net=dbFindNetByName(rep "<P&R net-name>")
   prsNetPropsUpdateValue(net "criticality" <c-value> "int" 10)
   prsNetPropsUpdateValue(net "connStatus" "needToConnect" "string" "needToConnect")
   prsNetPropsUpdateValue(net "sigType" "supply" "string" "signal")
   prsNetPropsUpdateValue(net "routingWidth" <r-value> "float" 0.000000)
   prsNetPropsUpdateValue(net "estCap" 0.000000 "float" 0.000000)
   prsNetPropsUpdateValue(net "maxLen" 0.000000 "float" 0.000000)
```

☐ `Route - Modify Net - ES2 0.7um CMOS - load '...'` [Editing:...]

≡ – bestätigen [Net Properties File]

5. Initialisierung und Flächenabschätzung

   ☐ `Floorplan - Reinitialize...` [Editing:...]

   ≡ – bestätigen [Initialize Floorplan]

   Die Plazierung wird vorbereitet, dazu werden die benötigten Flächen für Core- und Pad-Bereiche abgeschätzt. Die Ergebnisse werden in dem CADENCE Eingabefenster ([ES2 0.7um ...]) ausgegeben. Für die Plazierung der Standardzellen wird eine Plazierungsregion, entsprechend der Flächenabschätzung, erzeugt.

6. Plazierung der Padzellen
   Die Pad-Plazierung kann entweder automatisch oder über eine Steuerdatei, zur Festlegung einer vorbestimmten Reihenfolge, erfolgen. Nachfolgend werden beide Möglichkeiten kurz vorgestellt:

   (a) Automatische Plazierung der Padzellen.

   ☐ `Floorplan - I/O Place...` [Editing:...]

   ≡ – bestätigen [IO Placer]

   Die Padzellen werden automatisch plaziert. Die *falsche* Orientierung der Pads wird durch spätere Bearbeitungsschritte noch korrigiert und braucht hier nicht weiter beachtet zu werden.

   Im allgemeinen wurden die Padzellen bei der vollautomatischen Plazierung unsinnig angeordnet. Mit dem `move`-Befehl kann eine Nachbearbeitung erfolgen, um beispielsweise dazu Vdd- und Gnd-Pads gleichmäßig zu verteilen und Busse zu ordnen.

   (b) Plazierung der Padzellen über eine Steuerdatei.

   > `vi <IOplace file name>` [xterm]

   Die Datei enthält zeilenweise Einträge der folgenden Form:

   `<cell name>  <side>  <offset>`

[3] optional: nur bei Verwendung von Megazell-Pads `LIBPWRBK`

Beispiel:

```
<cell name>  <side>   <offset>
|I3          left     0
|U192        left     1
|U193        left     2
|I2          bottom   0
|U189        bottom   1
|U190        bottom   2
|I1          right    0
|U191        right    1
|U186        right    2
|I4          top      0
|U187        top      1
|U188        top      2
```

Der `<cell name>` ist der Name der Instanz im Schematic, er kann mit dem Schematic-Editor angesehen werden.

Bei der Beschreibung der Anordnung (`<side>` und `<offset>`) ist die Zählrichtung jeder Seite gegen den Uhrzeigersinn festgelegt.

□ `Place - IO Commands - Read Initial File...` [Editing:...]

≡ `IO Frame File` = `<IOplace file name>` [Build IO Frame]
`Quit if Incomplete` = `<on>`

Achtung: Nach unseren Erfahrungen treten, bei der Plazierung über eine Steuerdatei, Fehler bei der Reihenfolge der Pads auf. Anscheinend sind davon Padzellen für die Peripherie-Spannungsversorgung (LIBPWRPY und LIBGNDPY) betroffen, deshalb ist es unbedingt empfehlenswert, die Plazierung, wie unten angegeben, zu überprüfen.

- Kontrolle der Plazierung und gegebenenfalls deren Umordnung „von Hand".

Hierbei können Padzellen, bzw. Gruppen von Pads (nach vorheriger Selektion), gespiegelt, verdreht und verschoben werden, um die gewünschte Plazierung zu erreichen. Da die „genaue" Ausrichtung erst später vorgenommen wird, genügt hier eine ungefähre Positionierung.

Die Namen der Padzellen lassen sich *nur* über deren Properties feststellen (Selektion und Aufruf der Property Fill-Form).

≡ `Instance` = `<on>` [OSW]
`<all others>` = `<off>`

$\uparrow_l$ `<pad cell>` [Editing:...]

Selektion einer (oder mehrerer) Padzelle(n).

□ `Edit - Properties...` / ⊙ `q` [Editing:...]

Kontrolle des Instanzennamens. Durch anschließende Selektion anderer Padzellen können deren Eigenschaften angesehen werden.

□ `Edit - Move` / ⊙ `m` [Editing:...]

≡ – entsprechend ausfüllen [Move]

Auswahl der Spiegelungen und Rotationen: es sind mehrere Angaben möglich, Rotationen sind 90° pro Auswahl...

[4]optional: nur bei Benutzung generierter Blöcke

7. Plazierung von Megazellen und Neueinteilung der Standardzellregionen[4]
Sind keine Megazellen in dem Entwurf vorhanden, so wurde, entsprechend der Flächenabschätzung schon eine Region für die Plazierung der Standardzellen erzeugt.

   Bei der Verwendung von Megazellen, mußsen die Megazellen von Hand plaziert, die vorhandene Region gelöscht und neue Regionen erzeugt werden.

   ≡ `Instance` = `<on>` [OSW]
   `<all others>` = `<off>`

   ☐ `Edit - Move` / ⊙ `m` [Editing:...]

   ≡ – entsprechend ausfüllen [Move]

   Mit dem `move`-Befehl können die Megazellen (innerhalb der ursprünglichen) Standardzellregion plaziert werden.

   $\uparrow_l$ `<megacell>` [Editing:...]

   ☐ `Edit - Properties...` / ⊙ `q` [Editing:...]

   ≡ `Status` = `placed` [Edit Instance Properties]

   Nach Selektion der neu plazierten Megazelle wird in der Property-Liste der Plazierungsstatus umgesetzt.

   ≡ `Region` = `<on>` [OSW]
   `<all others>` = `<off>`

   $\uparrow_l$ `<region>` / ⊙ `^a` [Editing:...]

   ☐ `Edit - Delete` / ⊙ `del` [Editing:...]

   Sind alle Megazellen plaziert, wird die ursprüngliche Region gelöscht.

   ☐ `Create - Region...` / ⊙ `r` [Editing:...]

   ≡ `Choose Rows From` = `routingRatio` [Create Region]
   `Routing Ratio` = `1`
   `Fit Instance ...` = `<off>`
   `Analyze After Create` = `<off>`

   Neue Regionen für die Plazierung der Standardzellen werden erzeugt, dabei ist für jede Region deren Größe mit der Maus einzugeben.

   $\uparrow_l$ `<region>` / ⊙ `^a` [Editing:...]

   Selektion einer Region.

   ☐ `Analyze - Floorplan Objects...` / ⊙ `i` [Editing:...]

   ≡ – siehe unten [Region Analysis Info]

   Prüfung, ob die Standardzellen in den Regionen plaziert werden können, dabei werden die Standardzellzeilen in die Regionen eingezeichnet. Um die Regionen zu bearbeiten hat man folgende Befehle zur Verfügung:

   - automatische Größenanpassung der Plazierungsregion

     $\uparrow_l$ `Update Region` [Region Analysis Info]

     Die aktuelle Region wird in ihrer Breite den Standardzellzeilen angepaßt.

- manuelle Größenanpassung der Plazierungsregion

  ↑ₗ `<region>` [Editing:...]

  Wird der Cursor an eine Kante der Plazierungsregion gebracht, so verändert sich seine Form und man kann durch Drucken und Festhalten der linken Maustaste die entsprechende kante verschieben.

  ☐ `Analysis - Run` [Region Analysis Info]

  Die Größe der Standardzellzeilen wird neu berechnet.

- Änderung der Zeilenanzahl

  ≡ `Number of Rows` = `<row number>` [Region Analysis Info]

  ↑ₗ `Update Region` [Region Analysis Info]

- Auswahl der nächsten Plazierungsregion

  ↑ₗ `Next` [Region Analysis Info]

- Beenden der Analyse

  ☐ `Window - Close` [Region Analysis Info]

8. Sichern? — es empfiehlt sich, den Entwurf während des Plazierungs- und Verdrahtungsprozesses des öfteren zu sichern, um beim Auftreten von Fehlern, die sich erst später bemerkbar machen, dort wieder aufsetzen zu können.

   ☐ `Design - Save As` [Editing:...]

   ≡ `View Name` = `<initPlaced>` [Save As]

   Hier ist ein geeigneter Name der `Cell View` zu wählen. Um die Arbeit später an dieser Stelle fortzusetzen, kann diese View dann geladen werden:
   ☐ `Open - Design...` [ES2 0.7um ...]

   Bei Beenden der Plazierung & Verdrahtung ist darauf zu achten, daß `autoLayout` *nicht* gesichert wird, da die Sicherung ja inzwischen unter einem anderen Namen erfolgt ist, man aber immer noch `autoLayout` bearbeitet (Fenstertitel: `Editing:...`).

9. Plazierung der Standardzellen

   ☐ `Place - Automatic...` [Editing:...]

   ≡ `Method` = `both` [Automatic Placement]
   `Insert Feedthru` = `<on>`
   `Feedthru Library Name` = `StdLib`
   `Feedthru Master Name` = `LIBFEED`
   `Feedthru Master View` = `abstract`
   `Placement Snap Grid` = `0.1`
   `Mirror Cells` = `<on>`

   Die automatische Plazierung wird gestartet. Beim Auftreten von Fehlern (üblicherweise: Überlappung der Standardzellen mit Pad- oder Megazellen) müssen die Plazierungsregionen neu eingeteilt werden.

10. Einfügen von Ecken (Corner-Cell)

   ☐ `Place - IO Commands - Add Corners...` [Editing:...]

   ≡ `Glue Cell Library Name` = `PadLib` / `<lib name>` [Insert IO Corner Cells]
   `Glue Cell Master Name` = `LIBCORNER` / `<bl> <br> <tr> <tl>`
   `Glue Cell Master View` = `abstract`
   `Net Association` = `matchTermName`

   Die Corner-Zellen werden automatisch eingefügt. Dabei werden normalerweise die Eingaben der zweiten Spalte benutzt außer bei Verwendung spezieller Zellen.

   – Solche extra zu behandelnden Zellen sind: analoge Corner-Zellen und digitale Corner-Versorgungs-Pads. Beim Ausfüllen der Form sind anzugeben:

`<lib name>`	`<..>`	
`PadLib`	`LIBCORNER`	*normale* Corner-Zelle
`AnaLib`	`PRANCORNER`	analoge Corner-Zelle
`PadLib`	`LIBCRNPWRPY`	Vdd-Pad: digital, für Pads
`PadLib`	`LIBCRNGNDPY`	Gnd-Pad: digital, für Pads

   – Digitale Versorgungs-Corner-Pads müssen schon im Schematic vorhanden sein.

   – Sollen analoge und digitale Corner-Zellen gemischt werden, so ist dies u.U. einfacher mit dem Befehl: ☐ `Place - Glue Cell - Add` [Editing:...]

   — ansonsten kann auch hier nur auf die Manuals verwiesen werden...

11. Abstände zwischen Padzellen erzeugen[5]
   Normalerweise ist dieser Schritt nicht notwendig, aber bei Core-bestimmten Entwürfen kann es unter Umständen gewünscht werden (explizit) Platz zwischen Padzellen vorzusehen (z.B. Verpackung in besondere Gehäuse).

   ☐ `Place - Glue Cell - Add...` [Editing:...]

   ≡ `Glue Cell Library Name` = `PadLib` [Add Glue Cell]
   `Glue Cell Master Name` = `LIBSLICE10 / LIBSLICE20`
   `Glue Cell Master View` = `abstract`
   `Number` = `<cell number>`
   `Instance Name Prefix` = `slicer`

   10, bzw. 20 μm breite Zellen werden erzeugt und in der linken unteren Ecke des Layouts plaziert.

   ≡ `Instance` = `<on>` [OSW]
   `<all others>` = `<off>`

   ☐ `Edit - Move` / ⊙ `m` [Editing:...]

   ≡ – entsprechend ausfüllen [Move]

   Mit Hilfe des `move`-Befehls werden die Glue-Zellen und die Pads wie gewünscht angeordnet.

   Sollten solche Zellen glöscht werden, geschieht dies mit:
   ☐ `Place - Glue Cell - Delete` [Editing:...]

[5] optional: nur bei Core-bestimmten Designs mit spezieller Padanordnung

12. Ausrichten der I/O-Pads

   □ `Place - IO Commands - Justify...` [Editing:...]

   ≡ `IO Align Style = free` [Align IO Frame]
   `Align Feature = origin`
   `IO to Core Spacing = 100`
   `IO to IO Spacing = 1`
   `Placement Snap Grid = 0.1`
   `Shift IO Frame to Origin = <on>`

   Endgültiges Ausrichten der Zellen des Padkranzes. Anschließend müssen alle Zellen „richtig" orientiert sein.

   Bei Verwendung analoger Pads, ist darauf zu achten, daß die Abschlußpads korrekt angeordnet sind (4 Anschlüsse zur digitale Seite, 2 zur analogen).

13. Abschluß der Standardzellzeilen
   Alle Standardzellzeilen müssen durch Cap-Zellen abgeschlossen werden. Diese speziellen Zellen sorgen dafür, daß die Spannungsversorgung (Vdd und Gnd) korrekt an die Standardzellzeilen angeschlossen wird.[6]

   ≡ `Region = <on>` [OSW]
   `<all others> = <off>`

   $\uparrow_l$ `<region>` / ⊙ `^a` [Editing:...]

   Da sich da Einfügen der Cap-Zellen nur auf selektierte Plazierungsregionen bezieht, sind vorher *alle* Regionen zu selektieren.

   □ `Place - Power Cell - Add Auto...` [Editing:...]

   ≡ `Max Bar Separation = 100` [Add Power Cells Automatically]
   `Cut Search Range = 1`
   $\uparrow_l$ `Define Power Cells`

   ≡ `*** Cap Cell Information ***` [Power Cell Definition]
   `Library Name = StdLib`
   `Left Cap Cell Name = LIBLCAP`
   `Align Left End Caps = <off>`
   `Right Cap Cell Name = LIBRCAP`
   `Align Right End Caps = <off>`

   Beim Ausfüllen der Fill-Form sind nur die Cap-Zellen einzutragen, alle anderen Zellgruppen sind „leer" zu lassen.

   Nach dem Hinzufügen der linken und rechten Cap-Zellen, sollte das Ergebnis im Layout überprüft werden.

[6]Neben der hier vorgestellten Methode der Core-Spannungsversorgung, gibt es auch noch die Möglichkeit ein *power-grid* zu definieren. Für die Benutzung dieser Option sei auf die CADENCE Online-Dokumentation verwiesen.

14. Ausrichten der Zellen

≡ `Instance` = `<on>` [OSW]
`<all others>` = `<off>`

↑$_l$ `<instance>` / ⊙ `^a` [Editing:...]

Der `snap`-Befehl bezieht sich nur auf selektierte Zellen, dementsprechend dementsprechend sind alle Instanzen im Layout zu selektieren.

☐ `Place - Snap to Grid` [Editing:...]

≡ `Selected Cells` = `<on>` [Snap To Grid]
`Placement Snap Grid` = `0.1`

15. Überprüfen der (fertigen) Plazierung

☐ `Place - Check...` [Editing:...]

≡ `Check Cells Overlap` = `<on>` [Placement Checker]
`Check IO or Pins Overlap` = `<on>`
`Check Gap Between ...` = `<on>`

Achtung: es dürfen keine Fehler auftreten und *alle* Zellen müssen als P&R-Status `placed` besitzen, andernfalls kann nicht verdrahtet werden. Die Ergebnisse werden im CADENCE Eingabefenster (ES2 0.7um ...) angezeigt.

16. Sichern?

☐ `Design - Save As` [Editing:...]

≡ `View Name` = `<finalPlaced>` [Save As]

Zusätzlich kann auch die Plazierung als ASCII Datei gespeichert werden. Damit kann der jetzige Zustand, ausgehend vom unplazierten `autoLayout`, wiederhergestellt werden.

☐ `Floorplan - Floorplan File - Write...` [Editing:...]

≡ `File Name` = `<FP file>` [Save Placement Information]

Sichern des Floorplans — `<FP file>` kann dann analog dazu wieder eingelesen werden: ☐ `Floorplan - Floorplan File - Read...` [Editing:...]

17. Verdrahtungskanäle generieren

☐ `Route - Channels - Create...` [Editing:...]

≡ – bestätigen [Create Channels]

Die Verdrahtungskanäle werden erzeugt. Sollten später nach der Kanalverdrahtung Probleme mit dem *Alignment* der Padzellen auftreten (s.u. Probleme), so kann der Wert `Initial Cut` explizit festgelegt werden.

Während die Kanäle generiert werden, erhält man im CADENCE Eigabefenster einige Meldungen: `WARNING Contact...`; sie können ignoriert werden.

18. Verdrahtung der Versorgungsspannung im Core-Bereich[7]
Mit diesem Schritt wird für die Plazierungsregionen festgelegt, von welcher Seite (links oder rechts) die Versorgungsspannungen angeschlossen werden.

≡ `Region`	= `<on>`		[OSW]
`<all others>`	= `<off>`		

↑ₗ `<region>` / ⊙ `a` [Editing:...]

Da das Power Routing für jede Region getrennt durchgeführt werden muß, ist eine Plazierungsregion zu selektieren.

Wenn mehrere Regionen gleichartig behandelt werden sollen, so können sie gemeinsam selektiert werden. Die nachfolgenden Befehle beziehen sich jeweils auf die selektierte Auswahl.

□ `Route - Power Route - Create...` [Editing:...]

≡ `Net Name(s)`	= `vdd!/gnd!`	`gnd!/vdd!`	[Power Route]
`Route Power Cell Columns`	= `<off>`	`<off>`	
`Route to Left Ends ...`	= `<on>`	`<off>`	
`Route to Right Ends ...`	= `<off>`	`<on>`	

Entsprechend der Anordnung der Padzellen `LIBPWRCO` und `LIBGNDCO` empfiehlt es sich (für jede Plazierungsregion) Vdd von der einen Seite und Gnd von der gegenüberliegenden Seite anzuschließen. So werden unnötige Kontakte vermieden und die Leitungslänge wird minimiert.

Gegebenenfalls kann die Topologie der Versorgungsspannungen mit dem nachfolgend beschriebenen Schritt: „Test der globalen Verdrahtung" kontrolliert werden.

19. Globale Verdrahtung

□ `Route - Global Route - Automatic...` [Editing:...]

≡ `Method` = `both` [Global Route Method]
↑ₗ `Automatic`

≡ `Use Stup Routing` = `<on>` [Automatic Global Route]

≡ `Method` = `both` [Global Route Method]
↑ₗ `Optimizer`

≡ `Remove Unused Feedthru`	= `<on>`	[Global Route Optimizer]
`Feedthru Library Name`	= `StdLib`	
`Feedthru Master Name`	= `LIBFEED`	
`Feedthru Master View`	= `abstract`	

Globale Verdrahtung des ICs, die Ausgaben des Programms erscheinen im CADENCE Eingabefenster.

Treten Probleme bei der globalen Verdrahtung auf, so können die beiden Teilschritte `auto` und `optimize` getrennt ausgeführt werden.

[7] optional: um entsprechend der Plazierung der Core-Powerpads eine „gute" Verdrahtung zu erzeugen ist dieser Schritt empfehlenswert!

20. Test der globalen Verdrahtung[8]
Es empfiehlt sich, gerade bei Entwürfen mit analogen Komponenten, die Vdd- und Gnd-Netze zu überprüfen. Dies sind folgende Netze:

`VSSR1!`	`VCCR1!`
`VSSR2!`	`VCCR2!`
`ANAVSSR2!`	`ANAVCCR2!`
`gnd!`	`vdd!`

☐ `Route - Global Route - Topology Display...` [Editing:...]

≡ `Net Name` = `<net name>` [Net Topology Display]

Ein Netz `<net name>` wird angezeigt. Durch die Eingabe von ↑$_l$ `remove` kann die Anzeige gelöscht werden.

Entsprechend lassen sich auch andere Netze anzeigen... Sind die Ergebnisse der globalen Verdrahtung nicht zufriedenstellend, so müssen einzelne Netze interaktiv gelegt werden (s. Manual).

21. Kanalverdrahtung

☐ `Route - Detail Route - Automatic...` [Editing:...]

≡ `Compaction Mode` = `automatic` [Route And Compact All Channels]
`Contact Style` = `offCentered`
`Add Conditional Via` = `off`
`Routing Layers` = `2`
↑$_l$ `Compact Options`

≡ `Placement Snap Grid` = `0.1` [Compact Channels]

Verdrahtung der einzelnen Kanäle, die Ausgaben des Programms erscheinen im Cadence Eingabefenster.

- Probleme / Rückgängigmachen von Verdrahtungsschritten
Eines der am häufigsten auftretenden Probleme ist, daß, bei *padbestimmten*- Entwürfen, Pads nach der Kanalverdrahtung in den Core-Bereich hineingezogen werden. Dazu zwei (mögliche) Abhilfen:

-1- andere Kanalverdrahtung

☐ `Route - Detail Route - Automatic...` [Editing:...]

≡ `Compaction Mode` = `rigidIO` [Route And Compact All Channels]
...weiter wie oben

-2- Generierung der Kanäle mit anderer Vorzugsrichtung beim ersten Schnitt

☐ `Route - Channels - Create...` [Editing:...]

≡ `Initial Cut` = `vertical / horizontal` [Create Channels]

[8]optional: bei Verwendung analoger Komponenten sowie nach dem „Power Routing“ empfehlenswert!

Um Verdrahtungsoperationen rückgängig zu machen, gilt die entsprechend umgekehrte Reihenfolge.

☐ `Route - Detail Route - Delete...` [Editing:...]

☐ `Route - Global Route - Delete...` [Editing:...]

☐ `Route - Channels - Delete` [Editing:...]

▷ `channelDelete` [ES2 0.7um ...]

Bei der Kanalgenerierung werden entsprechende Views für die Kanäle in der Bibliothek (`DESIGNS`) erzeugt, diese können durch diese SKILL-Prozedur gelöscht werden. **Achtung:** ES 2 empfiehlt, vor dem Aufruf die Datenbasis zu sichern! (bugs?)

Achtung: da durch die Routing-Operationen auch die Kanäle verändert werden, sollten, um zum plazierten Layout zurückzugelangen, alle drei Schritte (`... - Delete`) durchgeführt werden — am einfachsten ist es meist, vorher gesicherte Versionen, bzw. eine Floorplan-Datei zu laden.

22. Sichern

☐ `Design - Save As` [Editing:...]

≡ `View Name` = `<routed>` [Save Design]

23. Extraktion der Leitungskapazitäten für die Post-Layout Simulation

☐ `Analyze - ES2 0.7um CMOS - Extract Parasitic Capacitance...` [Editing:...]

≡ – bestätigen [Extract Wiring Length]

Die Leitungskapazitäten werden extrahiert und in eine Datei `pr.cap` geschrieben, die später für die Post-Layout Simulation benötigt wird.

24. ...that's all folks — CADENCE DF II kann jetzt beendet werden.

☐ `Open - Quit...` [ES2 0.7um ...]

≡ – bestätigen [Quit]

≡ – entsprechend ausfüllen! [Save Cellviews]

Achtung: für alle noch nicht gesicherten Designs wird gefragt, was mit ihnen geschehen soll. Dabei ist darauf zu achten, daß `<top-level cell> autoLayout` *nicht* gesichert wird, da diese View (vorher) einen anderen Namen bekommen hat (letzter Schritt: `Design - Save As...`).

≡ `Save Techfile Changes?` = `discard changes` [Save Technology File Changes?]

Änderungen in der Technologiedatei für `DESIGNS` brauchen nicht gesichert zu werden.