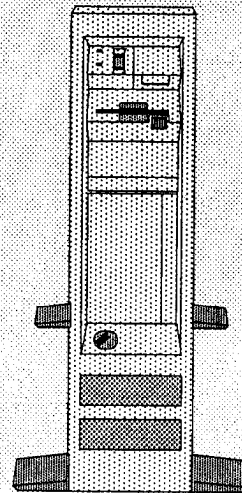
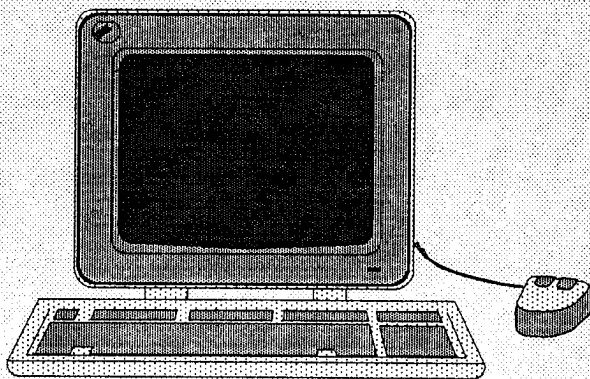


# **ERFAHRUNGEN MIT DER BOARD-STATION VON MENTOR-GRAPHICS**



**Elke Mackensen**

**Fachhochschule Offenburg  
Im Februar 1993**

## Erfahrungen mit der BOARD-Station von MENTOR-Graphics

Elke Mackensen

Fachhochschule Offenburg

Mit zunehmend komplexer werdenden Schaltungen wachsen auch die Anforderungen an die Entwicklung einer entsprechenden Leiterplatte. Mit der BOARD-Station von MENTOR-Graphics können professionelle Leiterplatten entwickelt werden.

Im Rahmen dreier Entwicklungsprojekte an der Fachhochschule Offenburg wurden mehrere aufwendige Layoutentwürfe mit der BOARD-Station in verschiedenen Diplomarbeiten durchgeführt.

Im Folgenden wird über die dabei gewonnenen Erfahrungen berichtet.

### 1. Vorstellung der Projekte

In den letzten drei Jahren wurde die BOARD-Station in mehreren Projekten erfolgreich eingesetzt.

Bei dem ersten Projekt handelt es sich um die Entwicklung eines GPS-Empfängers (GPS = Global Positioning System) für den Einsatz im KFZ-Bereich. Dieser GPS-Empfänger besteht aus einer Antenneneinheit, einem Tuner, einem Netzteil, einem Analogteil und einem Digitalteil. Bei dem schaltungstechnisch recht aufwendigen Analog- bzw. Digitalteil wurden die Platinen mit der BOARD-Station entwickelt (Abb.1-1 u. Abb.1.-2). Es wurden folgende Anforderungen an die Platinen gestellt:

Platzmässig ist man mit den Platinen auf die Größe eines Autoradios beschränkt, wobei man den Platzbedarf des Tuners und des Netzteiles abziehen muß. Wegen dieser Platzeinschränkung und des schaltungstechnisch großen Umfanges wurden

- die Platinen in SMD-Technik und als Multilayerplatinen ausgeführt.
- große Teile der Schaltung in LCA's, GAL's und in einen ASIC integriert, und diese bei der Platinenentwicklung miteinbezogen.

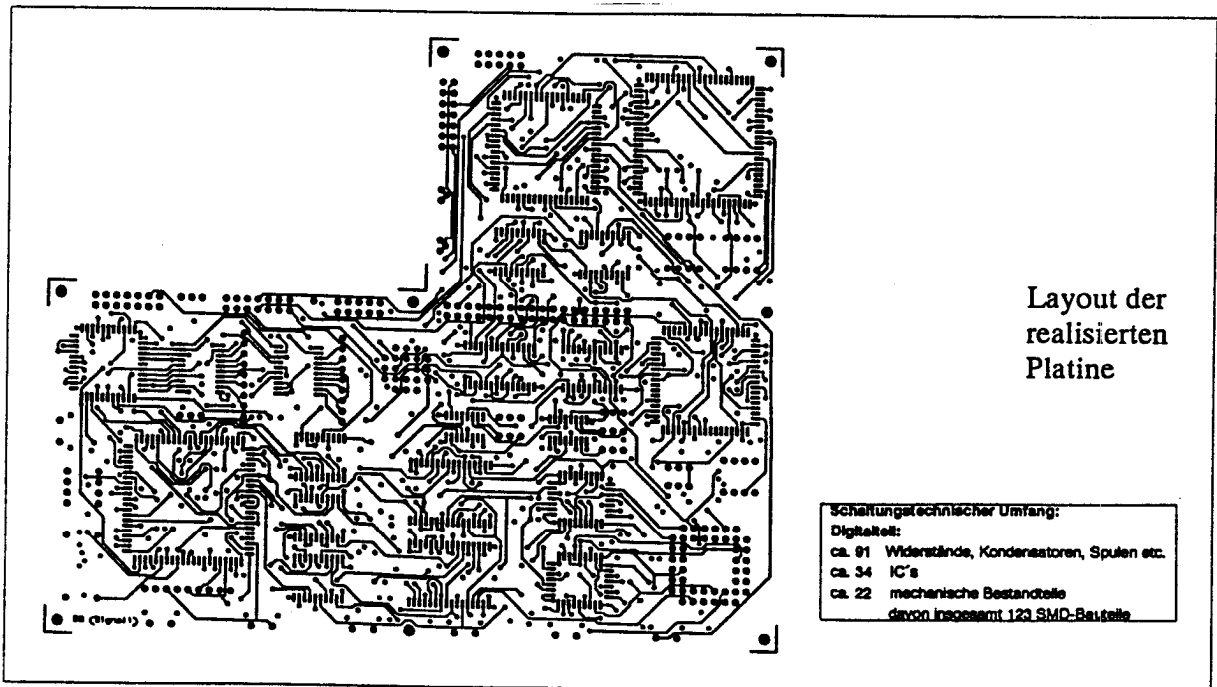


Abb.1-1: Das GPS-Projekt (Projekt 1): Digitalteil

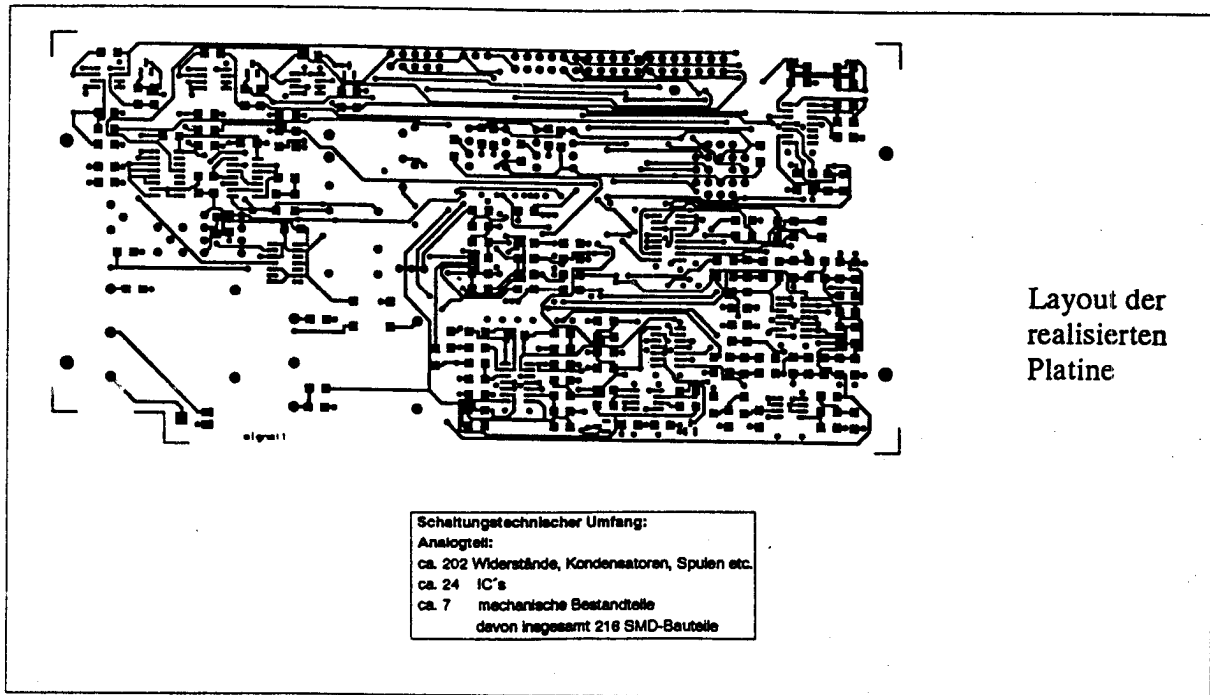


Abb.1-2: Das GPS-Projekt (Projekt 1): Analogteil

Bei dem zweiten Projekt handelt es sich um ein Industriegerät mit Display, Tastatur und IEC-Bus zur Auswertung der Daten eines optischen Wegsensors, der zur Messung von Abständen zu einem Objekt benutzt wird. (Abb.1-3). An die Platine wurden folgende Anforderungen gestellt:

- Wegen des schaltungstechnisch großen Umfangs (siehe Abb.1-3) und wegen hoher Taktfrequenzen Fertigung der Platine in SMD-Technik und als Multilayerplatine.
- Unterbringung der Schaltung auf einer Platine mit doppeltem Europakarten-Format, also nicht etwa Trennung von Analog- und Digitalteil.

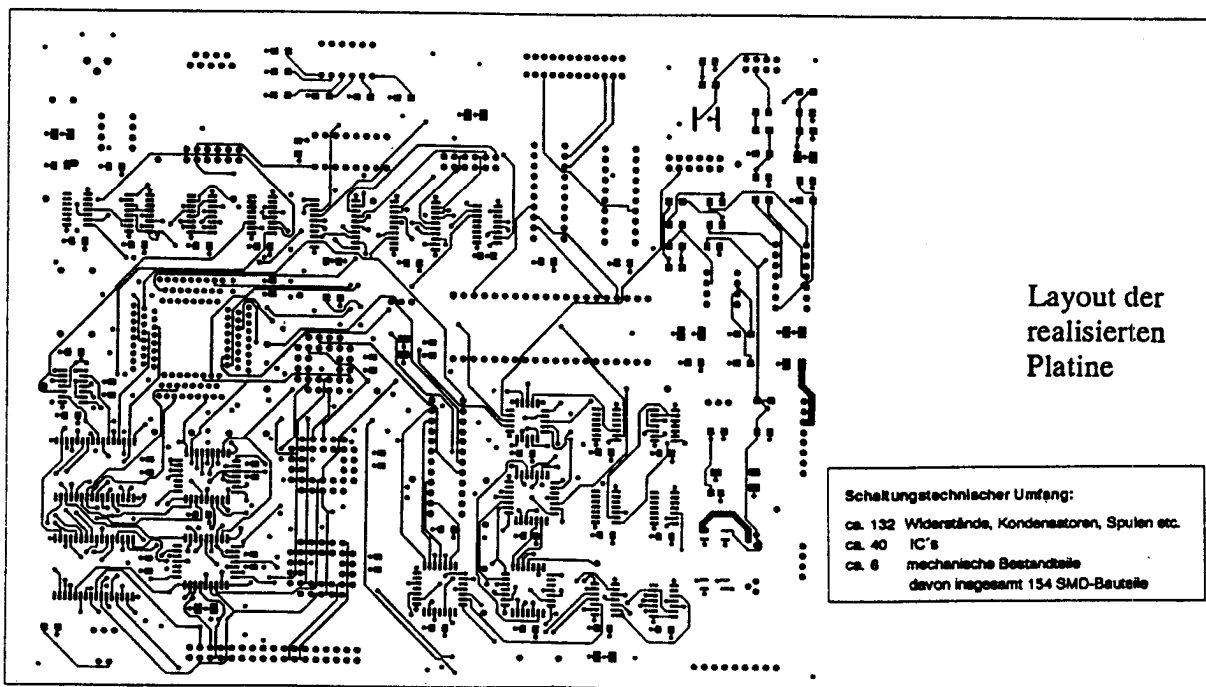


Abb.1-3: Industriegerät zur optischen Wegmessung (Projekt 2)

Bei dem dritten Projekt handelt es sich um die Entwicklung einer Elektronik für optische Datenübertragung in einem Drehbearbeitungszentrum zur Vermessung von Distanzen, Kanten und Oberflächen von Getriebewellen. Die Anforderungen an die Platine in diesem Falle waren:

- Erstellung einer runden Platine mit einem  $\phi$  von 16 cm
- Erstellung einer zweilagigen Platine, um die Platine noch mit Mitteln der Fachhochschule fertigen zu können.
- Wegen des Schaltungstechnisch großen Umfangs auch weitestgehend Fertigung in SMD-Technik und Einsatz von Logic-Cell-Array's (LCA's).

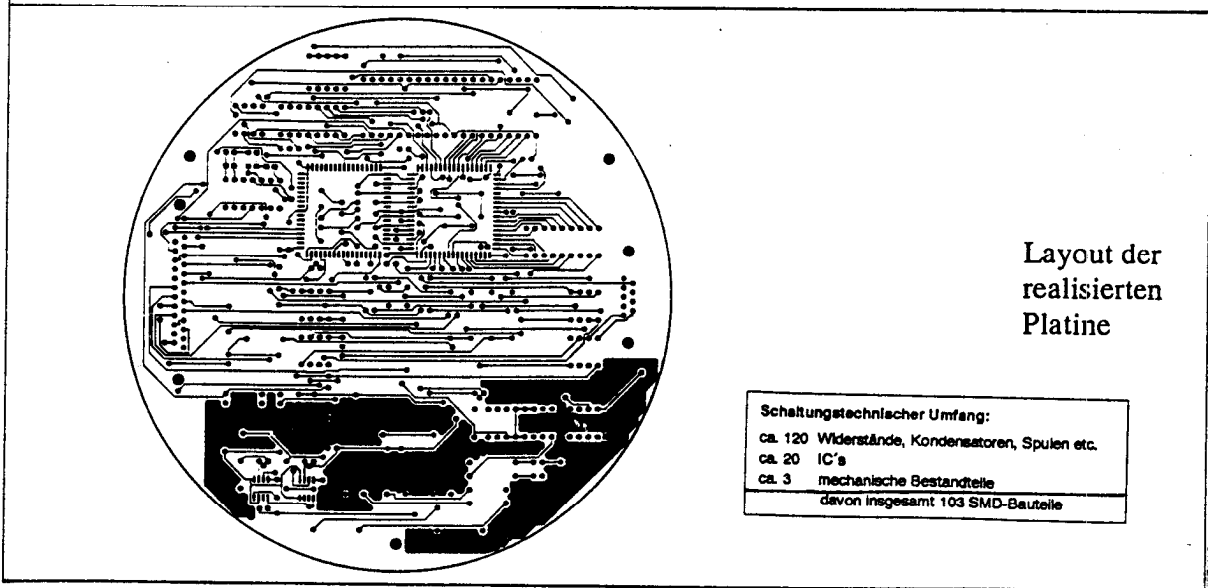


Abb.1-4: Messung von Distanzen bei einer Drehbearbeitung (Projekt 3)

## 2. Entwicklungsgang auf der BOARD-Station

Der prinzipielle Entwicklungsgang auf der BOARD-Station, um von dem bestehenden Schaltungsentwurf zur fertigen Platine zu kommen, ist in Abb.2-1 dargestellt. Es müssen 8 verschiedene Programmmodule durchlaufen werden:

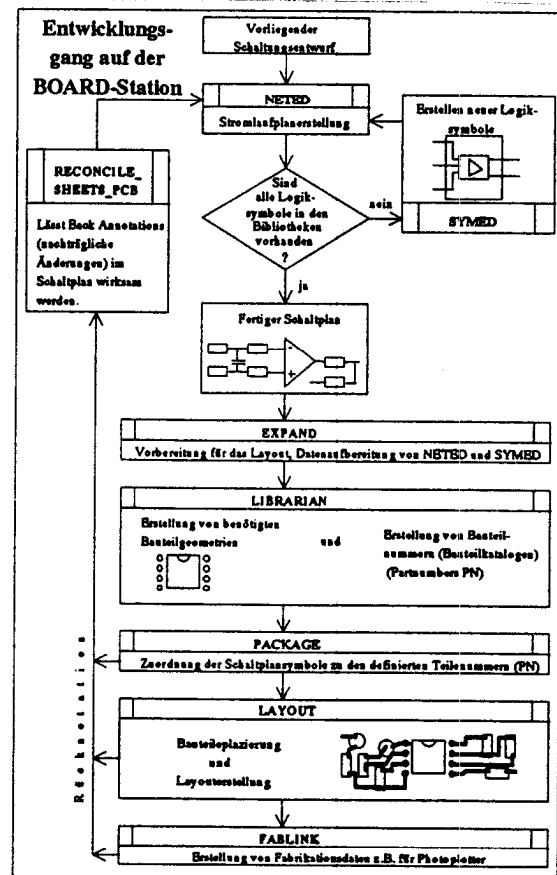


Abb.2-1: Entwicklungsgang auf der BOARD-Station

### 3. Erfahrung mit den einzelnen Programmodulen

#### 3.1. NETED/SYMED

Der erste Schritt für den Layoutentwickler ist die Eingabe seines Stromlaufplanes in den Schaltplaneditor NETED. Zur Schaltplanerstellung stehen dem Entwickler im sogenannten STATIC-Menü in

55 Libraries ca. 1800 Symbole für Bauteile

zur Verfügung. Ist man sich jedoch der Vielzahl real existierender Bauteile bewußt, so kann man sich vorstellen, daß gerade bei größeren Schaltplaneingaben viele Symbole neu erstellt bzw. bestehende Symbole abgeändert werden müssen. Hierzu steht dem Anwender der Symboleditor SYMED zur Verfügung. Die neu erstellten Symbole speichert man in ein Unterverzeichnis z.B. *symbols\_pcb* ab, aus dem sie jederzeit aufrufbar sind.

Für unserere Projekte mußten trotz der angebotenen Symbole eine recht hohe Anzahl eigener Symbole erzeugt werden:

Projekt 1: Analogteil:	33 neue Symbole
Digitalteil:	42 neue Symbole
Projekt 2:	40 neue Symbole
Projekt 3:	47 neue Symbole

Ein gravierendes Problem von NETED ist, daß die Einbindung eigener Symbole schlecht unterstützt wird. Unter dem Menü PARTS → PARTS BY NAME im EDIT-Fenster kann das entsprechende selbsterstellte Symbol unter Angabe seines gesamten Pfades ausgewählt werden:

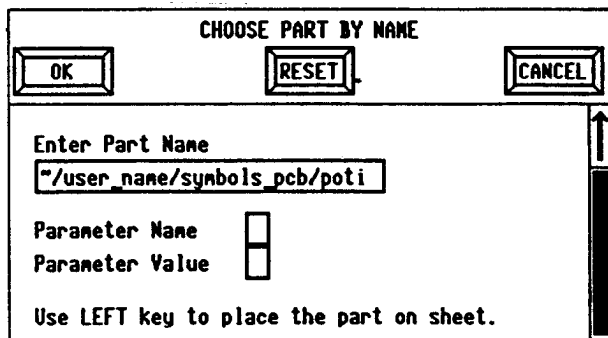


Abb.3-1: Aufruf eines eigenen Symbols unter NETED

Die Nachteile liegen auf der Hand:

- zeitaufwendige Arbeit,
- komplizierte Prozedur für viele selbsterstellte Symbole,
- man muß sich zu lange Pfadnamen merken.

Die Lösung dieses Problems haben wir durch die Erstellung einer eigenen Library gefunden, die nach Aufruf von NETED wie die anderen Standard-Libraries im STATIC-Menü erscheint. Als Beispiel ist hier die Erstellung der *fho\_lib* dargestellt. Ein Symbol Name als "Potentiometer" soll in der Library als "Poti" auswählbar sein. Die entsprechenden Befehle müssen in NETED in der Command-Line eingegeben werden:

1. Voraussetzung:

Das Symbol ist mit SYMED erstellt worden und liegt in einem Unterverzeichnis z.B. *symbols\_pcb*.

2. Definition der betreffenden Library:

```
DEFine MEnu "fho_lib"
```

3. Definition des Symbol, das unter der Library stehen soll (Poti) und Angabe, wo das Symbol zu finden ist (*symbols\_pcb*):

```
DEFine Item "fho_lib/poti" "transcript off; act comp ~/user_name/symbols_pcb/potentiometer -Hidden -NOUNdo
```

4. Schreiben der Library bzw. eines compilierten Binärfiles:

```
WRite MEnu fho_menu -replace
```

Um nun die erzeugte Library auch nach Aufruf von NETED im STATIC-Menü erscheinen zu lassen, muß nur noch die Startup-Datei für NETED (*neted.startup3*) abgeändert werden, wie dies in Abb.3-2 dargestellt ist. Dieses Startup-File muß sich unter dem Userspezifischen Verzeichnis *user\_data* befinden und enthält schließlich die zusätzliche Befehlszeile:

```
READ MEnu fho_menu
```

Der Aufruf von NETED erfolgt schließlich mit:

```
NETED filename -st
```

In NETED müssen zudem weitere Informationen an Hand von sogenannten Properties für die spätere Bearbeitung der Schaltung eingebracht werden. Dies sind z.B. elektr.Daten, Pinbelegung, Symbolname usw.. Eine Vergabe bzw. Abänderung solcher Properties ist durch eine gute Menü-Führung in NETED leicht möglich. Eine Erläuterung, der für die Erstellung eines Layoutes benötigten Properties, ist in dem "IDEA Series Properties Reference Manual" übersichtlich dargestellt.

```

#-----
#
# FILE:   neted.startup3
#
# SOURCE:  -/elke/user_data
#
# HISTORY: 17/10/90 FHO Reorganized in preparation for v6.1
#
#-----
IF ^($Easy_Interface = $TRUE) THEN
  EXecute /idea/sys/hi/startup/easy_neted.startup ^$arg_1 ^$arg_2 ^$arg_3 ^$arg_4 ^$arg_5 ^$arg_6 ^$arg_7
  ^$arg_8 ^$arg_9 # compiled version
ELSE
  DO /idea/sys/hi/startup/original_neted.startup ^$arg_1 ^$arg_2 ^$arg_3 ^$arg_4 ^$arg_5 ^$arg_6 ^$arg_7
  ^$arg_8 ^$arg_9
END IF

# Eigenes STATIC-MENU einlesen und anzeigen
#-----
READ MEnu fho_menu
  
```

Abb.3-2: Neues Startup-File für NETED

### 3.2. EXPAND

Das Programm EXPAND, das die Datenstrukturen von NETED und SYMED für die weitere Bearbeitung aufbereitet, erzeugt nach Aufruf von EXPAND\_PCB automatisch ein Design-File *pcb\_design.erel*. Dieses Programm ist unproblematisch, setzt aber voraus, daß unter NETED keine Fehler gemacht wurden. Falls also ein EXPAND nicht richtig durchgeführt wird, so liegt die Ursache nur an Fehlern in NETED.

### 3.3. LIBRARIAN

Einer der recht zeitaufwendigsten und abstraktesten Programme ist LIBRARIAN.

#### 3.3.1 Aufgaben von Librarian

a) Benötigte Bauteil-Geometrien aus vorhandenen Bibliotheken in eine eigene Bibliothek zu kopieren bzw. abzuspeichern. Hierzu stehen dem Layoutentwickler von MENTOR, wie auch schon in NETED, unter dem Menü LIBRARIES → VIEW PARTS LIBRARIES → MENTOR

7 Bibliotheken mit ca. 110 Bauteilgeometrien

zur Verfügung.

Der Anwender sollte sich jedoch nicht zu früh über diese recht große Anzahl von Bauteilgeometrien freuen. Wir mußten bei unseren Entwürfen immer wieder feststellen, daß man die vorhandenen Geometrien mit den Geometrien des realen Bausteines vergleichen sollte. Viele vorhandenen Geometrien stimmen z.B. im Rastermaß nicht mit denen des

Datenblattes überein.

Beispiel: Die Geometrie *jedec\_sq\_84* (PLCC84-Gehäuse) liegt nicht im Zoll-Raster

Oder die Nummerierung der Pins der vorhandenen Geometrien stimmt nicht mit der Zählweise des Herstellers überein.

Beispiel: Die Geometrie *jedec\_rec\_32* weicht von der Nummerierung mit der Zählweise eines INTEL-Flash-Speicher ab.

Bei der Übernahme von vorhandenen Bauteilgeometrien sollte also immer ein Vergleich mit dem Datenblatt des betreffenden Bauteiles erfolgen !!!

b) Erstellung eigener Bauteilgeometrien, Pad's und Via's. Dies ist mit dem Menü LIBRARIES → CREATE COMPONENT → GENERIC COMPONENT möglich.

c) Erstellung eines Bauteilekataloges, d.h. man ordnet den Logiksymbolen aus dem Schaltplan ein oder mehrere Bauteilgeometrien zu. Dies läßt sich auch mit dem Menü LIBRARIES → CREATE PART NUMBER relativ leicht realisieren, wie dies in Abb.3-3 gezeigt ist.

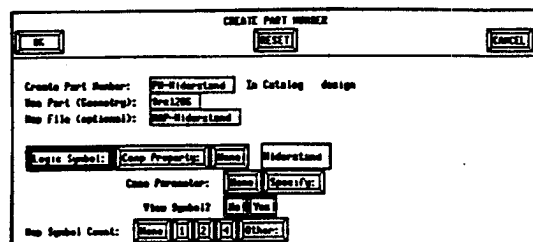


Abb.3-3: Erstellung eines Bauteilekataloges

Was hier als besonders Positiv auffiel, war die Möglichkeit, einem Gehäuse mehrere unterschiedliche Symbole aus NETED zuzuordnen. Dies bezeichnet man als sogenanntes "Inhomogenes Zusammenpacken" bzw. als "Non-homogeneous Packaging" (Abb.3-4). Diese Möglichkeit mußten wir bei dem Analogteil des Projektes 1 nutzen. Hier wurden für den verwendeten ASIC vier verschiedene Symbole erstellt, da auf dem ASIC ein ZF-Verstärker, drei verschiedene Multiplizierer und ein Addierer integriert waren.

An Hand dieser Option des inhomogenen Zusammenpackens kann man in NETED/SYMED mehrere Symbole für einen Baustein erstellen, um im Schaltplan die Übersicht zu wahren. Wenn man daran denkt, daß in zunehmendem Maße LCA's, ASIC's usw. in Schaltungen eingesetzt werden, erweist sich dies als vorteilhaft.

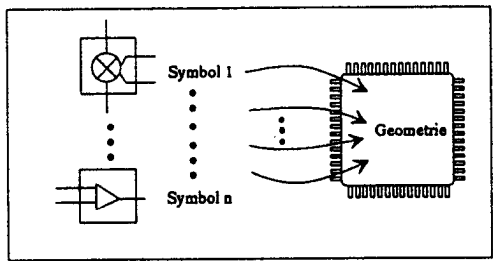


Abb.3-4: Inhomogenes Zusammenpacken.

#### d) Erstellung des Leiterplattenboardes.

### 3.3.2. Gründe für die Komplexität von LIBRARIAN

a) Aufwendige Struktur des Programmes durch viele verschiedene Menüs, die gerade für den Einsteiger zur Verwirrung führen.

b) Es müssen viele Informationen für die spätere Bearbeitung unter PACKAGE, LAYOUT und FABLINK eingebracht werden:

Diese Informationen vergibt man mit sogenannten Attributen, die entweder zwingend oder optional sind. Hier gibt es genau 74 verschiedene Attribute. Auf die richtige Vergabe der Attribute muß geachtet werden, da einige in der späteren Bearbeitung des Designs nicht mehr abänderbar sind, andere wiederum in der späteren Bearbeitung wesentlich einfacher zu vergeben sind.

Mit einem Attribut kann man folgende Dinge festlegen:

- Platzierungs-Eigenschaften
- Routing-Eigenschaften
- Eigenschaften für eine thermische Analyse

Man unterscheidet die Attribute in

- BOARD-Attribute, die sich auf die Leiterplatte beziehen.
- Component-Attribute, die sich auf die Bauteilgeometrien beziehen.
- Pin-Padstack-Attribute, die sich auf alle Arten von Pad's beziehen.
- Via-Padstack-Attribute, die sich auf alle Arten von VIA's beziehen.

Diese Attribute unterteilt man wiederum in Graphical-Attributes und Non-Graphical-Attributes. Ein Graphical Attribute ist in der Zeichnung des Bauteiles oder des Boardes direkt sichtbar. Ein Non-Graphical-Attribute wird zusätzlich vergeben, ist jedoch nicht aus der Zeichnung ersichtlich.

In Abb.3-5 ist die Erstellung einer Spulen-Geometrie und ihrer Attribute dargestellt. Bei der Vergabe von Component-Attributes sollte man nun folgendes beachten:

Eine Bauteilgeometrie besitzt 2 zwingende Attribute und je nach Anwendung 16 optionale Attribute. Die unbedingt erforderlichen Attribute sind Graphical-Attributes und werden beim Zeichnen der Geometrie angelegt:

- Pin-Definition  
(Component\_Pin\_Definition: Attr.3+4)
- Bauteile-Umrandung  
(Component\_Placement\_Definition: Attr.5)

Hierbei tritt folgendes Problem auf:

Optionale Attribute von Components lassen sich in der späteren Bearbeitung in LAYOUT nicht mehr abändern, obwohl die meisten von ihnen dort erst wirksam werden.

Folge: Vergessene Attribute in LIBRARIAN bedeuten einen großen Zeitaufwand für den Layoutentwickler, da er wieder in LIBRARIAN zurück und die Bauteile erneut aufrufen und bearbeiten muß. Besonders nachteilig ist dies natürlich bei vielen selbsterzeugten Geometrien.

#### Beispiel:

Bei den Layoutentwürfen der Fachhochschule entschloss man sich bei den meisten Platinen eine beidseitige Platzierung der Bauteile durchzuführen. In LIBRARIAN läßt sich nun dafür das optionale Attribut COMPONENT\_LAYOUT\_SURFACE 'BOTH' vergeben (Attr.9 in Abb.3-5), was dem Bauteil die Berechtigung gibt, auf beiden Seiten einer Platine platziert zu werden. Vergißt man dieses Attribut, so lassen sich die Bauteile nur auf einer Seite der Platine platzieren.

Deshalb kann hier die Empfehlung gegeben werden, sich alle optionalen Attribute der Bauteilgeometrien genau zu betrachten, um auftretende Probleme in LAYOUT zu vermeiden.

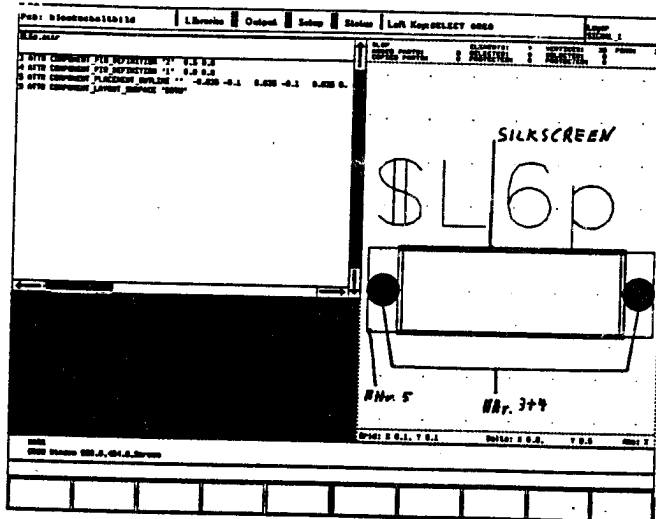


Abb.3-5: Erstellung einer Spulen-Geometrie

Bei der Vergabe von Board-Attributen kann man 12 zwingende und 24 optionale Attribute vergeben. Im Gegensatz zu den Component-Attributen sind die meisten optionalen Attribute eines Boardes in LAYOUT abänderbar, da diese im Wesentlichen auf den Routing-Vorgang bzw. die Design-Rules für den Routing-Prozess bezogen sind. So sollte man auf die Vergabe von optionalen Attributen bei einem Board verzichten, da diese in LAYOUT wesentlich einfacher zu vergeben sind.

c) Die Fertigungsmöglichkeiten des Platinenherstellers müssen miteinbezogen werden:

Während der Layout-Entwicklung unserer Projekte sind drei gravierende Dinge aufgefallen, die sich erst nach der Fertigung bemerkbar machten:

- Man hat die Möglichkeit, in LIBRARIAN eine Siebdruckvorlage (SILKSCREEN) um die Bauteile zu zeichnen (siehe Abb.3-5). Verwendet man auf der Platine hauptsächlich SMD-Bauteile, so muß man hier auf Folgendes achten: Überdeckt der Positionsdruck SMD-Pads, so lassen sich die SMD-Bauteile nur noch von Hand anlöten, ein Reflowlöten der Bauteile ist nicht mehr möglich, da der Siebdrucklack während des Lötvorganges nicht verdrängt wird und somit eine gute Lötung der SMD-Bauteile nicht mehr gewährleistet ist (siehe Abb.3-6). Diese Erfahrung mußten wir bei 2 Platinen machen. Bei der Digitalplatine von Projekt 1 mußte während der Fertigungszeit nochmals eine völlig neue Siebdruckvorlage erstellt werden.

Bei der Analogplatine von Projekt 1 wurde die Siebdruckvorlage etwas zu dick erstellt und wurde der Siebdruck während der Fertigung leicht versetzt, so daß zahlreiche SMD-Pads von dem Siebdruck überdeckt waren. Diesen Siebdruck konnte man nur noch durch vorsichtiges Abkratzen entfernen.

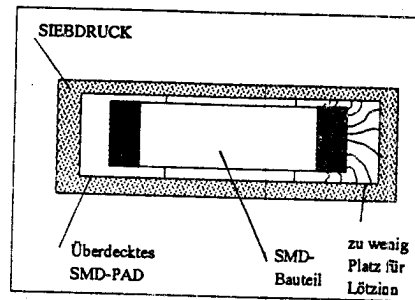


Abb.3-6: Schlechter Siebdruck bei SMD-Pad's

- Bei der Erstellung der Lötstopmaske (SOLDER\_MASK) von Pad's und Via's sollte man auf folgendes achten:

Die Lötstopmaske wird wie in Abb.3-7.a dargestellt etwas größer als das eigentliche Pad bzw. Via gewählt, da man davon ausgeht, daß die Positionierung der Lötstopmaske nicht 100% genau ist. Hier muß jedoch darauf geachtet werden, daß die Lötstopmaske nicht zu groß gewählt wird, wie das z.B. bei unserem dritten Projekt passiert ist. Dadurch wurden an die Pad's grenzende Leiterbahnen freigelegt (Abb.3-7.b), was schließlich beim Löten zu möglichen Kurzschlüssen führen kann, gerade wenn minimale Abstände von Leiterbahn zu Pad gewählt wurden.

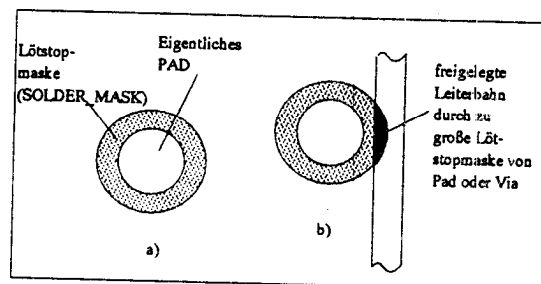


Abb.3-7: Lötstopmaske bei Pad's und Via's

- Bevor man an die Definition der Leiterplatte geht sollte man wissen, wieviele Lagen die Platine haben soll.

Mit der BOARD-Station ist eine Definition beliebiger Anzahl von Lagen möglich. Es gibt jedoch viele Firmen, die aus fertigungstechnischen Gründen nur geradzahlige Multilayerplatinen fertigen können. Ein Beispiel hierfür ist die Erstellung der Analogplatine aus Projekt 1. Diese war als 5-lagige Platine konzipiert worden. Gefertigt werden konnten jedoch entweder nur 4 oder 6 Lagen. Deshalb mußte bei dieser Platine eine Lage verdoppelt werden.



Daraus ergaben sich folgende Nachteile:

- Größerer Kostenaufwand bei mehr Lagen.
- Wäre man gleich von 6 Lagen ausgegangen, so hätte man eine bessere Aufteilung der Lagen vornehmen können.

**d) Es muß bei der Erstellung von Geometrien und Board's die richtige Einstellung des Grids gewählt werden:**

Die Einstellung des Grids hat später Auswirkung bei der Platzierung der Bauteile und beim Routen im Programm LAYOUT.

Bei der Einstellung des richtigen Grids sollte man sich an der Praxis orientieren. Die meisten Bauteile sind in INCH-Maßen gefertigt. Das Standard Zoll-Raster beträgt  $0,1'' = 2,54 \text{ mm}$  (z.B. Abstände bei den Pins an einem IC).

Grundsätzlich kann man sagen, daß die Einstellung des Grids zur Erstellung der Bauteile ein Vielfaches des Standard-Zoll-Rasters betragen sollte. Ein Beispiel für eine falsche Grideinstellung ist in Abb.3-8 dargestellt. Pin 1 liegt im Raster des Routing-Grids, Pin 2 liegt nicht mehr im Routing-Grid. Hier ist die richtige Anbindung des Pins beim Routing-Vorgang nicht mehr garantiert.

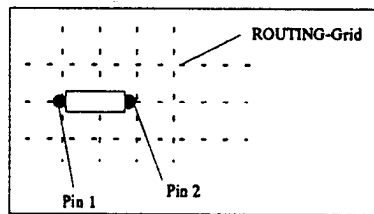


Abb.3-8: Falsche Grideinstellung

**5. Die Erstellung von Leiterplatten, die eine abweichenden Form von der eines Rechteckes haben, wird von LIBRARIAN schlecht unterstützt:**

Die Definition eines Boardes ist recht einfach, wenn die Platine eine rechteckige Form hat, denn dann geschieht die Erstellung des Boardes mit dem Menü LIBRARIES → CREATE NEW PART → CREATE BOARD. Hier werden die für das BOARD zwingenden und die optionalen Attribute durch systematische Abfrage automatisch vergeben. Ebenso geschieht die Definition der Physikalischen Lagen automatisch. Sobald man jedoch ein nicht rechteckiges BOARD erzeugen möchte, muß man alles von Hand einstellen trotz der großen Anzahl vorhandener Menüs.

Zunächst sollte man beachten, daß ein Board kein Bauteil ist und deswegen während des Designs nur

einmal vorkommen kann. Um überhaupt ein BOARD erstellen zu können muß man in die Command-Line den Befehl

Create BOARD

eingeben.

Ein Punkt aus dem man schließen kann, daß die BOARD-Station nicht dazu geschaffen ist, jedes Platinenformat zu definieren, war die Tatsache, daß die graphischen Attribute BOARD\_PLACEMENT\_OUTLINE (Gebiet, in dem Komponenten platziert werden dürfen) und BOARD\_ROUTING\_OUTLINE (Gebiet in dem geroutet werden darf) nur in Form von Polygonen mit  $90^\circ$ - oder  $45^\circ$ -Linien vergeben werden können. Dies machte vor allen Dingen bei der Erstellung der runden Platine aus Projekt 3 Schwierigkeiten (Abb.3-9). Hier mußte ein sehr feines Grid gewählt werden, um eine annähernd runde Placement\_Outline bzw. Routing\_Outline zu erstellen.

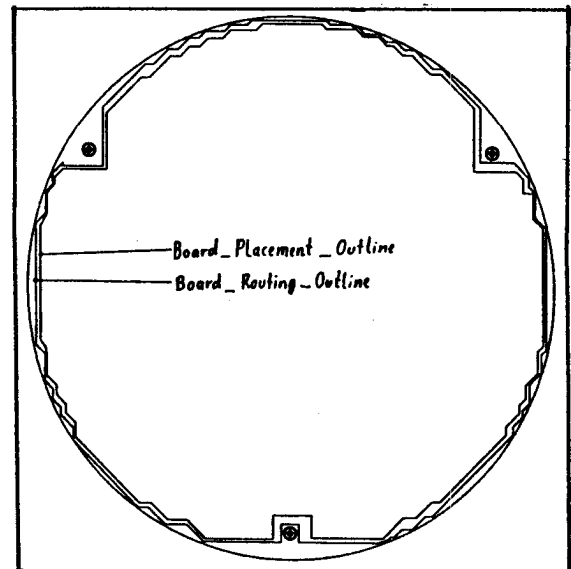


Abb.3-9: Erstellung einer runden Platine

Ein wichtiger Punkt ist die richtige Definition der einzelnen Lagen bei einer Platine, falls man das BOARD manuell erstellt. Dies geschieht mit dem Menü SETUP → PHYSICAL LAYERS. Man ordnet hier den sogenannten Physical-Layers (Physical\_1 ... Physical\_n) bestimmte Logical-Layers zu. Logical Layers sind die Lagen, die man sich auf dem Bildschirm anzeigen lassen kann, z.B. Signallage 1..n, Padlage 1 und 2, Power-Lagen 1..n, Board\_Outline usw..Insgesamt kann der Anwender hier **99 verschiedene Lagen** definieren.

Ein Problem, auf das wir während des Entwurfes der Leiterplatte bei Projekt 2 gestoßen sind, ist, daß die Definition der Physikalischen Lagen nicht den realen physikalischen Lagen entspricht. In Abb.3-10 ist die Definition der Physical-Layers am Beispiel der Platine



aus Projekt 2 erklärt. Erstellt werden sollte eine 6-lagige Multilayerplatine, wobei die Anordnung der Lagen entsprechend der Abb.3-10 vorgenommen werden sollte. Bei den mittleren Lagen wollte man eine sogenannte SPLITTING POWER PLANE durchführen, daß heißt die Power-Lagen nicht nur mit einer Spannung zu belegen, sondern aus zwei Spannungen zu kombinieren, in diesem Beispiel pos12V und VCC bzw. GND und GNDa.

Bei der Definition der Physical-Layers kann man nun beiden äußeren Lagen mehr als ein Logical-Layer zuordnen. Bei der Definition der inneren Lagen darf man nur einen Logical-Layer vergeben. D.h. man kann also nicht folgende Definition durchführen:

Physical\_3 = "Power\_1" "Power\_2"

Deshalb muß man hier mehr physikalische Lagen definieren, als in Wirklichkeit vorhanden sind. Darauf sollte man bei der Definition der Physical Layers unbedingt achten.

### 3.4. PACKAGE

PACKAGE ist wiederum eines der Programmmodule, das ohne größere Probleme zu überwinden ist. Hier wird allen vorhandenen Logiksymbolen mit der Funktion BUILD eine Geometrie aus dem Bauteilekatalog zugewiesen.

Aber auch hier kann gerade der Anfänger Fehler machen. Wie man in Abb.3-11 sieht, kann man in Package nicht nur Geometrien zuordnen, sondern auch die Platzierung (Spalte LOCATION) auf dem BOARD abändern.

Das Problem ist, daß in LAYOUT alle Bauteile platziert werden, ein BACK-ANNOTATION gemacht wird und man feststellt, daß dem einen oder anderen Bauteil nochmals eine andere Geometrie gegeben werden muß.

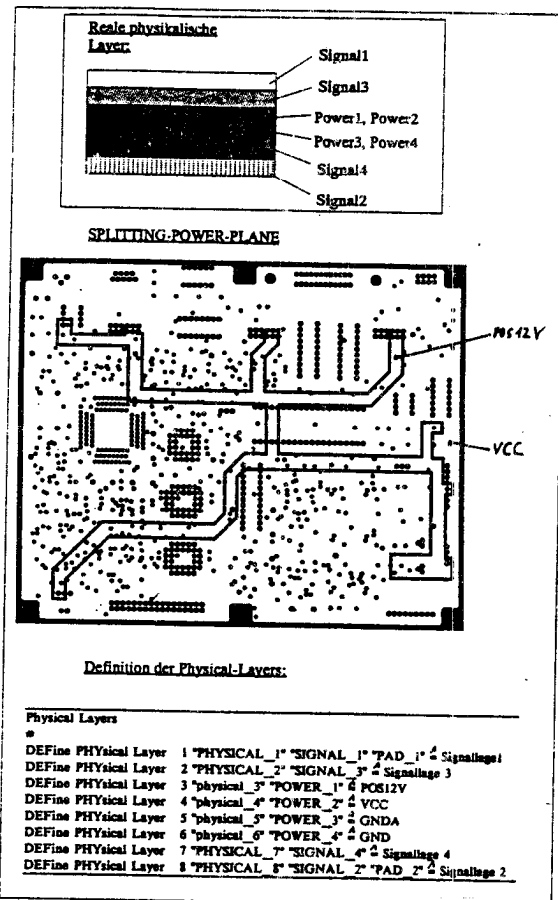


Abb3-10: Definition der Physical-Layer

Wenn der Anwender bei diesem Schritt nicht zuerst in NETED geht und ein anschließendes EXPAND durchführt ist trotz Back-Annotation nach einem erneuten Build in PACKAGE die ganze Platzierung verloren, denn eine Back-Annotation wird nur durch Aufruf von NETED wirksam. D.h. bei aufwendigen Layoutentwürfen sind möglicherweise mehrere Tage Arbeit zunichte geworden.

PACKAGE EDIT SYMBOL									
LINE	FLG	SCHEMATIC SYMBOL	GEOMETRY	MAPPING	REFERENCE	INSTANCE	PART NUMBER	LOCATION	SYMBOL PROPERTY VALUES
1	A	CAPACITOR	\$rc8885	NAP-C8885	C1	/I\$121/C1	PN-C8885	64.77 66.84 2 188 MM	6477888 6684888
2	A	CAPACITOR	\$rc8885	NAP-C8885	C2	/I\$121/C2	PN-C8885	64.77 62.855 2 188 MM	6477888 6285588
3	A	CAPACITOR	\$rc1286	NAP-C1286	C3	/I\$121/C3	PN-C1286	64.77 59.69 2 188 MM	6477888 5969888
4	A	CAPACITOR	\$rc1286	NAP-C1286	C4	/I\$121/C4	PN-C1286	49.53 52.785 1 278 MM	4953888 5278588
5	A	CAPACITOR	\$rc1286	NAP-C1286	C5	/I\$121/C5	PN-C1286	74.93 52.785 2 98 MM	7493888 5278588
6	A	CAPACITOR	\$rc1286	NAP-C1286	C6	/I\$121/C6	PN-C1286	86.36 58.165 1 8 MM	8636888 5816588
7	A	CAPACITOR	\$rc1286	NAP-C1286	C7	/I\$121/C7	PN-C1286	86.36 43.815 1 8 MM	8636888 4381588
8	A	CAPACITOR	\$rc1286	NAP-C1286	C8	/I\$121/C8	PN-C1286	86.36 48.64 1 8 MM	8636888 4864888
9	A	CAPACITOR	\$rc1286	NAP-C1286	C9	/I\$22/C9	PN-C1286	95.885 43.815 1 8 MM	9588588 4381588

Abb.3-11: PACKAGE-Möglichkeiten

### 3.5. LAYOUT

Im Programm LAYOUT wird schließlich das eigentliche LAYOUT der Leiterplatte erstellt. Der Entwickler kann sein Layout interaktiv oder per AUTO-Router entwerfen. Bei größeren Entwürfen sollte man nur mit dem Autorouter arbeiten. Ein interaktives Routing ist zu aufwendig.

Die Probleme in Zusammenhang mit dem Programm LAYOUT liegen vor allen Dingen in einer sinnvollen Platzierung der Bauteile, und der Wahl der Voreinstellungen, damit ein reibungsloser Routing-Prozeß abläuft.

#### a) Platzieren der Bauteile:

Hier gibt es zwei Möglichkeiten:

##### - Automatisches Platzieren der Bauteile:

Bei allen drei Projekten erwies sich dies jedoch nicht als sinnvoll, da z.B.

- alle Widerstände, IC's usw. Gruppenweise platziert wurden.
- eine viel zu enge Platzierung der Bauteile vorgenommen wurde.
- eine funktional schlechte Platzierung durchgeführt wurde.

##### - Interaktives Platzieren der Bauteile:

Hierbei sollte man vor allen Dingen auf die Platzierungsdichte der Bauteile achten. Ab einer Platzierungsdichte von ca. 60% kann der Router keine Lösungen mehr finden. Die Platzierungsdichte kann man sich mit einem Histogramm anzeigen lassen. Abb.3-12 zeigt als Beispiel die Digital-Platine von Projekt 1. Die Platzierung nimmt einen recht großen Zeitraum in Anspruch.

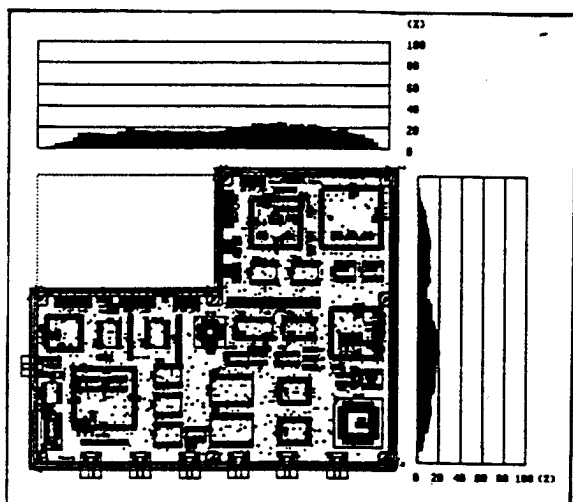


Abb.3-12: Histogramm für Platzierungsdichte

#### b) Grideinstellung:

Es gibt drei verschiedene Gridarten in LAYOUT einzustellen:

##### a) DISPLAY-Grid:

Grideinteilung auf dem Bildschirm; kann beliebig gewählt werden.

##### b) PLACEMENT-Grid:

Grid auf dem die Bauteile beim Platzieren einrasten. Sollte genauso groß wie das ROUTING-Grid eingestellt werden, damit gewährleistet ist, daß die Pins auf dem Routing Grid liegen und somit ein exaktes Routen stattfindet.

##### c) ROUTING-Grid:

Aufgeteilt in Wire-Grid, Via-Grid und Pad-Grid. Grid auf dem geroutet wird.

Aus Erfahrungswerten während der Entwürfe hat sich folgende Routinggrideinstellung als günstig erwiesen:

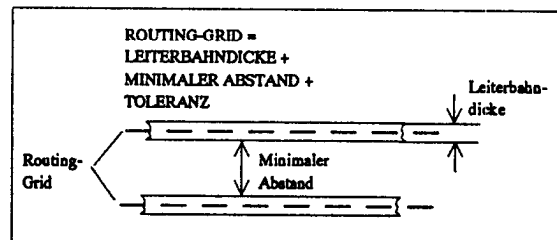


Abb.3-13: ROUTING-Grid-Einstellung

Im LAYOUT-User-Manual ist eine Formel angegeben, die zudem Pad- und Viagröße zur Berechnung des Routing-Grids miteinbezieht. Dies erwies sich jedoch als nicht so günstig.

Ist das Board in Librarian im mm-Maßstab gefertigt worden, so werden hier alle spezifischen Werte für die Grideinstellung auch in mm eingegeben. Aber auch hier gilt wiederum, nur Werte als Vielfaches von 0,1" bzw. 2,54 mm anzugeben.

### c) Einstellung günstiger Design-Rules

Die Design-Rules für das Routen müssen mit dem Menü AUTO → SETUP ROUTER eingegeben werden. Hier werden folgende Dinge festgelegt:

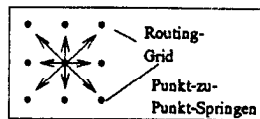
- Leitungsdicke (Wire-Width)
- Min. Abstände zwischen Leitungen, Leitungen und Vias (Track-Clearance)
- Min. Anstände zw. Pads, Pads und Vias, und Pads und Leitungen (Pad-Clearance)
- Benutzung von VIA's
- Benutzung von T-Junctions beim Routen
- Diagonales Routen
- Bevorzugte Routing-Richtung auf den Signallagen (horizontal oder waagrecht)
- Nach welchen Gesichtspunkten geroutet werden soll, z.B. nach dem Gesichtspunkt erst der "kürzesten Verbindungen" zu routen.

Bevor man die Design-Rules einstellt sollte man sich über die Herstellmöglichkeiten der Firma klar sein, d.h. wie dünn dürfen Leiterbahnen sein, wie groß müssen minimale Abstände sein usw..

Z.B. wurden bei der Digitalplatine von Projekt 1 Leitungsdicken von 0,254mm eingestellt. Bei der Analogplatine von Projekt 1 wurden minimale Abstände von 0,212 mm gewählt. Hiermit war man schon an der Grenze der Fertigungsmöglichkeit einer Firma.

### d) Richtige Einstellung des Routers

Der hier verwendete Router ist ein MAZE-Runner, der von Grid-Punkt zu Grid-Punkt springt und nach der kürzesten Verbindung zwischen Anfangs- und Endpunkt eines Netzes sucht.



Der Ablauf des AUTO-Routings ist in Abb.3-14 dargestellt. Dem Layoutentwickler wird hier die sinnvolle Auswahl der 3 verschiedenen Routing-Mechanismen überlassen. Wichtig ist auch die Auswahl der Anzahl der Durchgänge (PASSES) der Routing-Mechanismen. Hier sollte man die max. Anzahl der Passes gerade bei komplexen Layoutentwürfen nutzen. Die verschiedenen Routing-Mechanismen kann man übrigens auch mehr als einmal aufrufen.

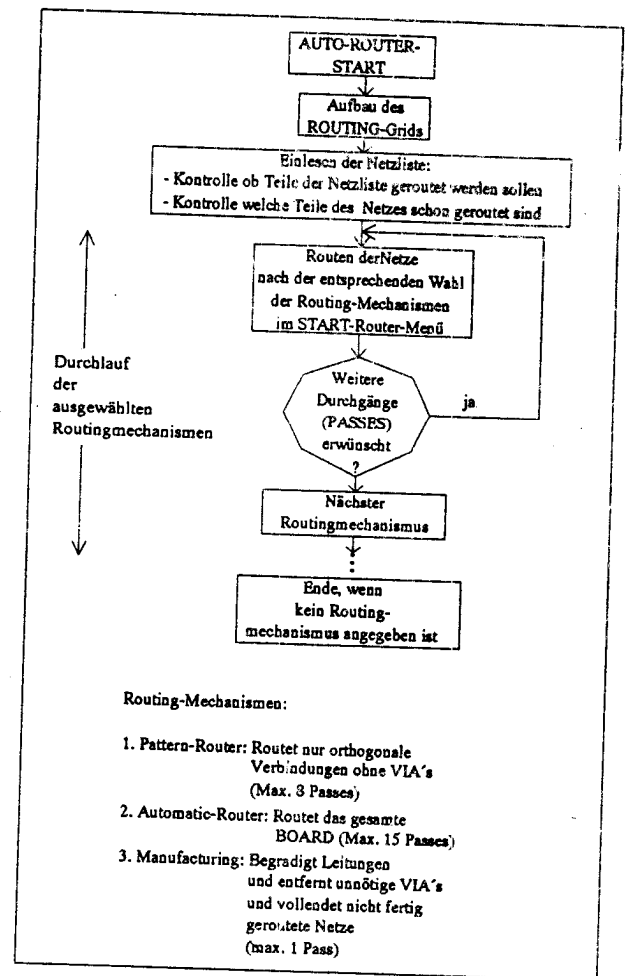


Abb.3-14: Ablauf des Autoroutings

### e) Vorrouten bestimmter Netze

Bestimmte Netze sollte man sich vorrouten lassen, so z.B. die Versorgungslagen der Platinen GND, VCC usw., da der Router dann direkt von einem Pin an dem betreffenden Netz mit einem VIA nach unten auf die Versorgungslage geht. Lässt man alle Netze auf einmal routen, so verbaut sich der Router selbst Wege durch zu lange Leitungsführung bei den Netzen, die mit einem VIA direkt an eine Lage angebunden werden könnten.

Bei der richtigen Einstellung der Gridarten, der Design-Rules und des Routers hat man im Routing-Durchlauf sehr gravierende Unterschiede bemerkt: Im Allgemeinen benötigt der Auto-Router für solch aufwendige Layoutentwürfe wie unsere **ca. 2-4 Stunden Zeit zum Routen**.

Bei falschen Voreinstellungen benötigt der Router einen **Zeitraum von ca. 5 Stunden** und weist danach immer noch **offene Netze** auf.

Nachträglich mußte man bei den Layoutentwürfen der Fachhochschule keine manuelle Bearbeitung durchgeführt werden; das Routing-Ergebnis war sehr gut.

### 3.5. FABLINK

In Fablink werden die Fabrikationsdaten für die Fertigung der Platine erstellt:

1. Die Konfiguration zur Erzeugung der Photoplotdaten im Gerber-Format und die Erstellung der Blendetabellen (Aperture table) für den Blendeteller des Photoplotters.
2. Die Konfiguration für Fräsdaten im Excellon- bzw. Sieb & Meyer-Format
3. Die Konfiguration für Bohrdaten im Excellon- bzw. Sieb & Meyer-Format

Dies geschieht in FABLINK weitestgehend automatisch.

Bevor man die Vorlagen für die Photoplots erstellt (Artworks), muß man diese definieren. Wie auch schon bei den Physical-Layers weist man den Artworks verschiedene Logical-Layers zu, die sich auf den Photoplots befinden sollen.

Ein Beispiel für die Erstellung von Photoplot-Definitionen ist in Abb.3-15 dargestellt.

ARTWORK LAYER	ARTWORK FILE	LOGICAL LAYERS
1	artwork_1	signal_1,pad_1,placement_keepout
2	artwork_2	signal_2,pad_2,placement_keepout
3	artwork_3	silkscreen_1,board_outline
4	artwork_4	silkscreen_2,board_outline
5	artwork_5	power_1,board_outline
6	artwork_6	power_2,board_outline
7	artwork_7	power_3,board_outline
8	artwork_8	solder_mask_1,board_outline
9	artwork_9	solder_mask_2,board_outline
10	artwork_10	paste_mask_1,placement_keepout
11	artwork_11	paste_mask_2,placement_keepout

*Handwritten annotations:*  
 } Signallagen (rows 1-2)  
 } SIEBDRUCK (rows 3-4)  
 } Versorgungslagen (rows 5-7)  
 } Lötstopmasken (rows 8-9)  
 } Klebmasken für SMD-Bauteile (rows 10-11)

Abb.3-15: Definition der Photoplotvorlagen

Bevor man nun die Photoplotdaten erstellt, sollte man sich die Definition der Blendetabellen für den Photoplotter betrachten. Abb.3-16 stellt den Auszug aus der Definition einer Blendetabelle dar.

In Bezug auf die spätere Fertigung sollte man sich die Einstellung der Blenden etwas genauer betrachten. Benutzt man z.B. für den Siebdruck eine zu große Blende, so entsteht ein unschöner, breiter Druckstrich mit Gefahr des Verlaufens.

Es gibt kreisförmige (Circle) oder rechteckförmige (Rectangular) Blenden. Zum Einen gibt es Blenden, die blitzartig auf- und zugemacht werden (Flash) und somit dafür sorgen, daß es unterschiedliche Pad's oder Via's gibt. Zum Anderen gibt es Blenden, die über den Film gezogen werden (Trace) und somit Linien, Text oder Leiterbahnen erzeugen. Verfügt eine Firma über eine Laserplotter, so ist man in der Zusammenstellung der Blenden vollkommen uneingeschränkt, d.h. man kann jede Art von Blende an jeder Blendentellerposition erzeugen. Wird kein Laserplotter verwendet, so ist man auf die Firmenüblichen vordefinierten Blendenteller angewiesen, und muß sich mit seinem Layout daran orientieren.

#	DEFine APerture	Position	Shape	Type	Height(Y)/Diameter	Width(X)	Kommentar: zugehöriges Symbol
DEFine	APerture	1	Circle	Trace	0.254000	0.000000	Leiterbahnen,Silkscreen(Umrisse)
DEFine	APerture	2	Circle	Flash	1.000000	0.000000	Via1(Signallage)
DEFine	APerture	3	Circle	Flash	1.500000	0.000000	Via1(Power-Antip.,Lötst.),Pad1(Sign.)
DEFine	APerture	4	Circle	Flash	1.100000	0.000000	Pad taste (Signal)
DEFine	APerture	5	Circle	Flash	2.600000	0.000000	Befestigungslöcher für Platine
DEFine	APerture	6	Circle	Trace	0.127000	0.000000	Board_Outline,Silkscreen(Text)
DEFine	APerture	7	Rectangular	Flash	2.032000	0.635000	SMD-Pad 80x25(Signal,Paste-Mask)

Abb.3-16: Auszug einer Blendentellerdefinition für den Photoplotter



#### 4. Entwicklungszeiten

In Tabelle 4.1 sind die Entwicklungszeiten der einzelnen Projekte dargestellt.

Tabelle 4.1: Entwicklungszeiten

	Reine Entwicklungszeit
<b>Projekt 1</b>	
Digitalteil	ca. 8 Wochen
Analogteil	ca. 6-8 Wochen
<b>Projekt 2</b>	ca. 6 Wochen
<b>Projekt 3</b>	ca. 4-6 Wochen

Hierbei muß man erwähnen, daß die Projekte jeweils im Rahmen einer Diplomarbeit durchgeführt wurden, und die Diplomanden vorher noch nicht mit dem System und der BOARD-Station vertraut waren.

Die meiste Zeit bei der Layoutentwicklung mit der BOARD-Station mußte also in die Einarbeitung der verschiedenen Programme investiert werden.

Den Zeitraum, den die Arbeit an den einzelnen Programmen in Anspruch nimmt, kann man nicht genau angeben, da dieser immer anwenderspezifisch ist. Denn ja nach Projekt müssen einmal mehr Geometrien erstellt oder ein anderes Mal ein komplexeres Layout geroutet werden usw..

Sicherlich kann man jedoch sagen, daß man durch Berücksichtigung der gewonnenen Erfahrungen, den gesamten Layoutentwicklungsprozeß an der BOARD-Station verkürzen kann.