

ERSTELLEN VON HARDMAKROS UND AUFBAU EINER ZELLBIBLIOTHEK UNTER VERWENDUNG DES ES2 - LIBRARY - KITS

Detlef Pouw

ASIC Design Center, Fachhochschule Offenburg
Leitung: Prof.Dr.-Ing. Dirk Jansen

Es wird eine Anleitung zur Erstellung von Hardmakros mit der Mentor Graphics - Software gegeben. Die Hardmakros werden mit Standardzellen aus der ES2-Bibliothek der Firma EUROCHIP aufgebaut. Die Hardmakros werden in eine eigenständige Bibliothek abgelegt und können in neuen Chip-Designs verwendet werden.

1. EINLEITUNG

An der Fachhochschule Offenburg wurde im Jahr 1989 unter Leitung von Professor Jansen das ASIC - Design - Center mit 5 APOLLO Workstations und Software der Firma MENTOR GRAPHICS [MG] eingerichtet.

Es wurden (im Zusammenhang mit dem Wahlfach ASIC) ein elektronischer Würfel und als Diplomarbeit der Empfangsteil eines GPS-Empfängers (Global Position System) entwickelt und in Standardzellen - Layout [SZ] hergestellt.

Die ASIC-Entwürfe beschränkten sich bis dato auf zwei Techniken. Zum einen das normale SZ-Layout, in dem Zellen aus der ES2-Bibliothek platziert und geroutet werden, zum anderen ein Layout, das sich aus Softmakros zusammensetzt.

Soft-Makros werden aus ES2-Zellen aufgebaut und in das Gesamt-Layout integriert. Der Nachteil dieser Entwurfstechnik liegt darin, daß Softmakros sich beim Integrieren in größere Designs, d.h. das Makro wird bis auf die Standardzellen hinunter geflattet und diese beliebig im Design platziert und geroutet.

Somit fallen Softmakros in jedem Design unterschiedlich aus. Dies impliziert jedoch bei jedem

Design neue physikalische Eigenschaften (Timing) des verwendeten Softmakros.

Um diese negativen Eigenschaften zu vermeiden, besteht mit der MG-Software die Möglichkeit, hierarchisch aufgebaute Hardmakros zu entwerfen, die einmal richtig ausgetestet und in einer Bibliothek abgelegt werden.

Beim Routen des Chip-Design muß das Hardmakro nur an seinen Ein/Ausgangsport verdrahtet werden; das Hardmakro selbst ist bereits komplett platziert und geroutet. Dies bringt neben bekanntem, definiertem Timing des Hardmakros den Vorteil kürzerer Routingzeiten mit sich.

Im folgenden wird ein Überblick über die MG-Software sowie eine kurze Anleitung zum Aufbau eines Makros gegeben.

2. DESIGN ÜBERBLICK

Um mit der **MG-Software V7.0** und dem **ES2-Kit** Hardmakros aufzubauen, müssen zuerst (einmalig) einige Konfigurationsdateien editiert werden, eigene Signal- und Powerports erstellt, und in eine Bibliothek integriert werden.

Dann ist es möglich, eigene Designs mit dem schematischen Editor zu erstellen und dieses automatisch in eine Hardmakrozelle verwandeln zu lassen. Eine Optimierung ist interaktiv während jedes Designs - Schritts möglich.

Schließlich können die neuen Zellen in eine Bibliothek integriert werden.

Alle Programme des ES2-Kits folgen der Syntax
es2_programm-name design.

Nach der Eingabe eines Schaltplanes mit dem schematischen Editor **NETED** muß zunächst mit dem Symboleditor **SYMED** ein Symbol für diesen Schaltplan entworfen werden.

Abbildung 1 zeigt die Ausgabedateien für das schematische Design.

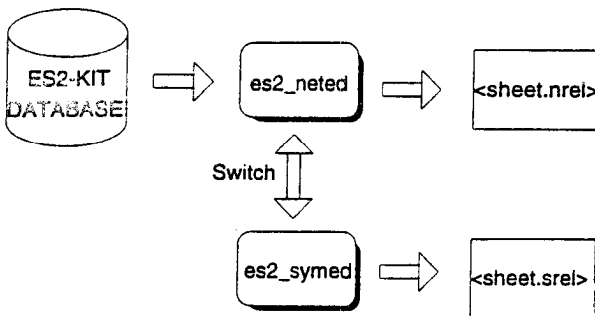


Abb. 1 NETED und SYMED schreiben ihre Daten in die Dateien <sheet.nrel> bzw. <sheet.srel>

Da NETED und SYMED die Möglichkeit besitzen, hierarchische Designs zu erstellen, müssen die schematischen Daten vor Erstellung einer Netzliste geflattet werden.



Abb. 2 ES2_COMP_EXPAND schreibt Daten nach <comp.erel>

ES2_EXPAND_COMP sorgt dafür, daß alle Properties, die für die Netzliste benötigt werden, expandiert und in die Datenbasis <comp.erel> integriert werden (Abb. 2).

ES2_LOGIC_ENTRY erstellt die physikalische Daten Basis aus dem expandierten logischen Design-File <comp.erel>.

Die erstellten Dateien <design.xxx> werden von den Standardzellen-Layout-Programmen CELLGRAPH und CHIPGRAPH benötigt.



Abb. 3 ES2_LOGIC_ENTRY wandelt die logischen, expandierten Daten eines Design in physikalische Daten um.

In Abb. 4 ist das Zusammenspiel der drei Programme Floorplan-Generator, Platzierer und Router dargestellt.

ES2_CELLFLOOR generiert automatisch einen Floorplan für das angegebene Design. Daten werden nach <floor.chp> und <floor.dsn> geschrieben.

ES2_CELLPLACE platziert automatisch sämtliche Makros des Designs. Es benutzt und erneuert Daten des physikalischen Design Files <design.dsn>, generiert von LOGIC_ENTRY. Daten werden nach <place.chp> und <place.dsn> geschrieben.

ES2_CELLROUTE routet das Design. Alle Makros müssen zuvor gültig platziert worden sein. Daten werden nach <route.chp> und <route.dsn> geschrieben.

Diese drei Programme lassen sich über die Parameterdatei <cadi.prm> (editiert von <edit_parms>) steuern.

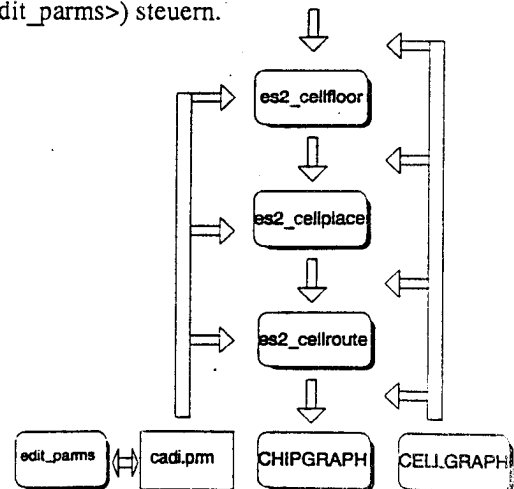


Abb. 4 CELLFLOOR, CELLPLACE und CELLROUTE beziehen ihre Parameter aus der Datei <cadi.prm>. CELLGRAPH kann in jedem Design-Schritt aufgerufen werden.

Um Zellen in eine Bibliothek aufnehmen zu können, muß zunächst entweder **ES2_CHIPGRAPH** oder **ES2_CHIPGRAPH_OUTPUT** aufgerufen werden, um die CHIPGRAPH Cell Data File-Base zu schreiben.

Ein Technologie-Verzeichnis (beim ES2-Kit ist eines `.../es2/ecpd15/sc_layout_data`) besteht mindestens aus drei Dateien:

- <cadi.blc> Binärdatei, die die physikalischen Makro- Daten enthält; erzeugt von <build_lib>
- <cadi.chp> Chip-Technologie-Datei; erzeugt von <tdf_chip_input>
- <cadi.prm> ASCII-Parameterdatei in Technology Definition Format (TDF); wird mit <edit_parms> editiert

TDF_CHIP_INPUT compiliert die Chip-Technologie-Datei <chip.tdf> unter Verwendung von Parametern aus <cadi.prm> und schreibt seine Daten nach <cadi.chp>.

BUILD_LIB nimmt die CHIPGRAPH-Daten unter Verwendung der Daten aus <cadi.chp> in die Bibliotheksdatei <cadi.blc> auf.

Abbildung 5 zeigt die Verbindung der Cell Data Base mit den Programmen **TDF_CHIP_INPUT** und **BUILD_LIB**.

Eine Zusammenfassung der wichtigsten Abläufe zur Generierung eines Hardmakros zeigt Abb. 6.

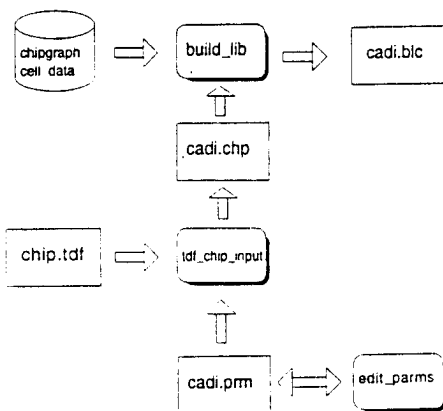


Abb. 5 Die Technologiedateien, die Bibliotheksdatei <cadi.blc> und ihre Erzeugung durch die Programme <build_lib> und <tdf_chip_input>

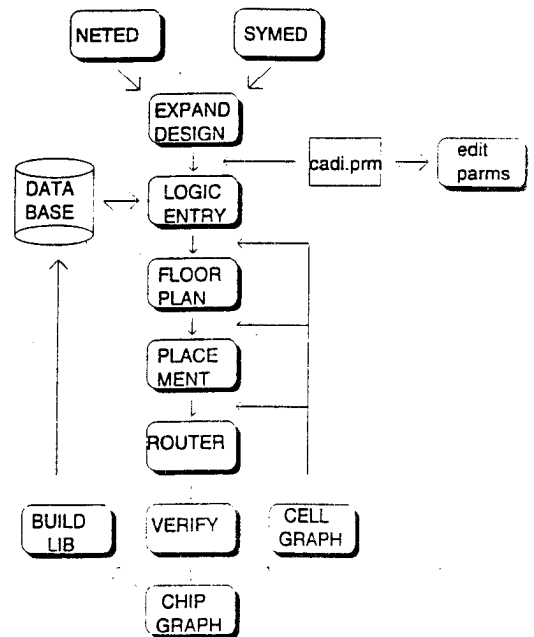


Abb. 6 Ablaufplan (mit interaktivem CELLGRAPH) der wichtigsten MG-Programme zur Erzeugung eines Hardmakros

3. POWER - UND SIGNAL - PORTS ERSTELLEN

Bevor eigene Zellen erstellt werden können, müssen Signal-I/O-Ports und Powerports definiert und in der Bibliothek aufgenommen werden.

Abbildung 7 und 8 zeigen die Ports (Metal2.i und Metal2.e) mit allen Properties.

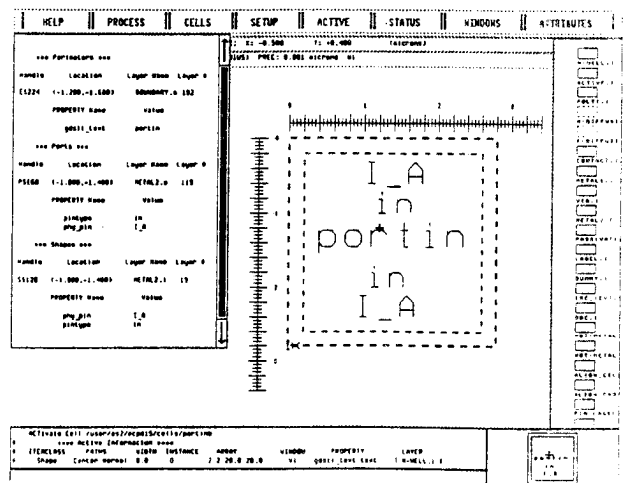


Abb. 7 Signal-Eingangsport in CHIPGRAPH mit Statusfenster für die Property-Anzeige

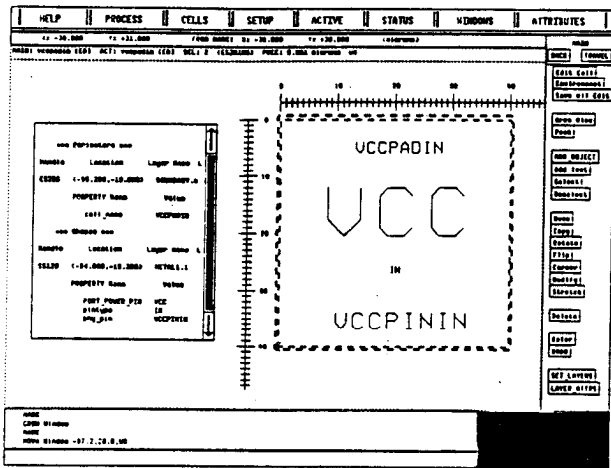


Abb. 8 VCC-Port in CHIPGRAPH mit Statusfenster für die Property-Anzeige

Die Zellen werden mit `<build_lib> sc_layout_data` in die Bibliotheksdatei `<cadi.blc>` aufgenommen.

In NETED werden Powerports aus der ES2-Zellbibliothek einfach modifiziert, d.h. es werden neue Properties vergeben (Abb. 9).

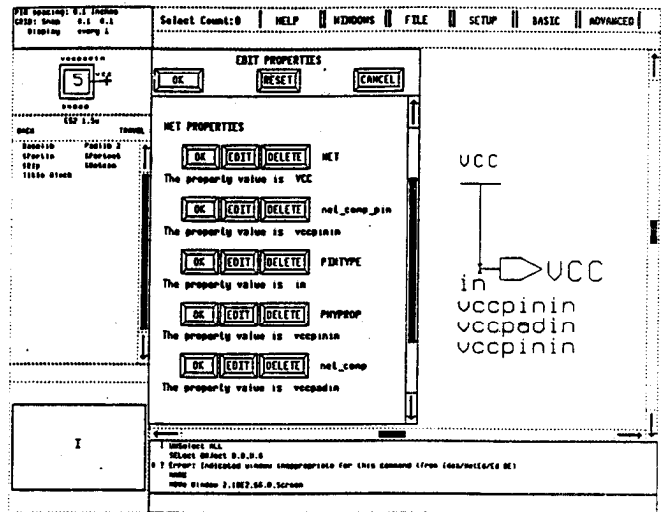


Abb. 10 NETED mit den NET-Properties für vccpadin
Auf gleiche Weise wird der VSS-Port erstellt.

Um die Verbindung vom Neted-Design zu den erstellten Ports zu schaffen, werden in NETED folgende Properties an das Design vergeben (Abb. 11):

PINTYPE IN IN für Eingangssignale

NET_COMP Portname Verbindung von NETED Sportin zum selbst erstellten Signalport

NET_COMP_PIN I_A Physikalische Pin Bezeichnung

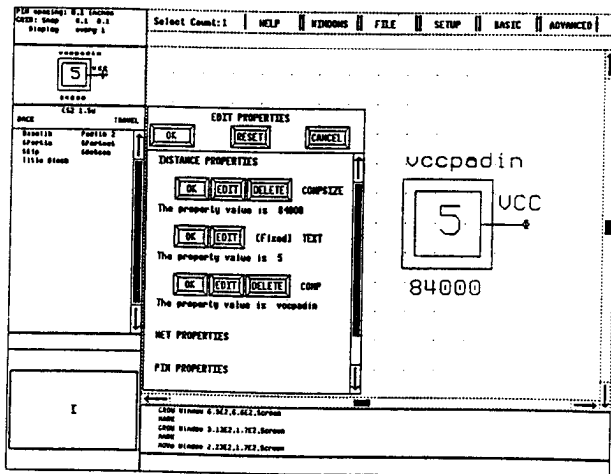


Abb. 9 NETED mit Spannungsport-Symbol und Properties

Die Sheet-Daten diesem Symbol müssen ebenfalls mit neuen Properties versehen werden (Abb. 10).

Mit dem Programm `<es2_autocell>` können nun die Sheet-Daten in physikalische Daten gewandelt werden. Abb. 12 zeigt das optimierte Hardmakro der ALU.

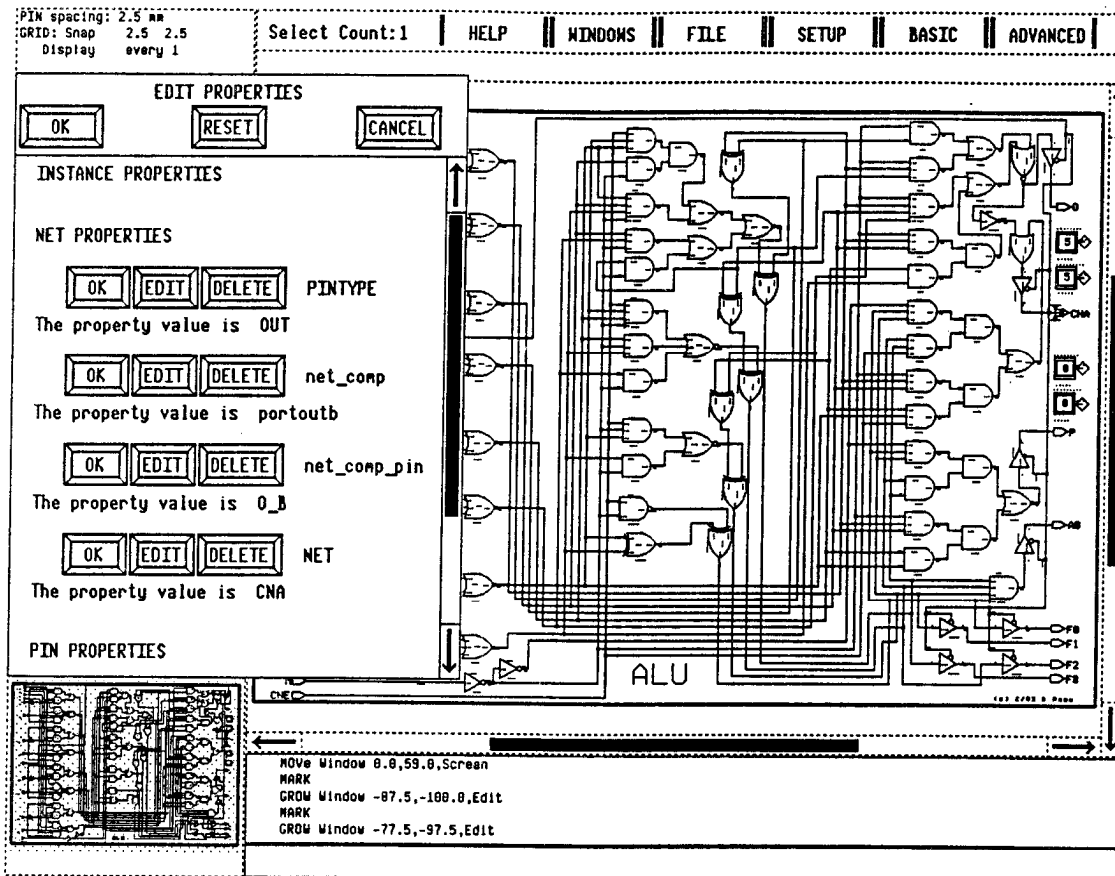


Abb. 11 NETED-SHEET einer ALU mit selektiertem Ausgangsport CNA und Fenster für Properties

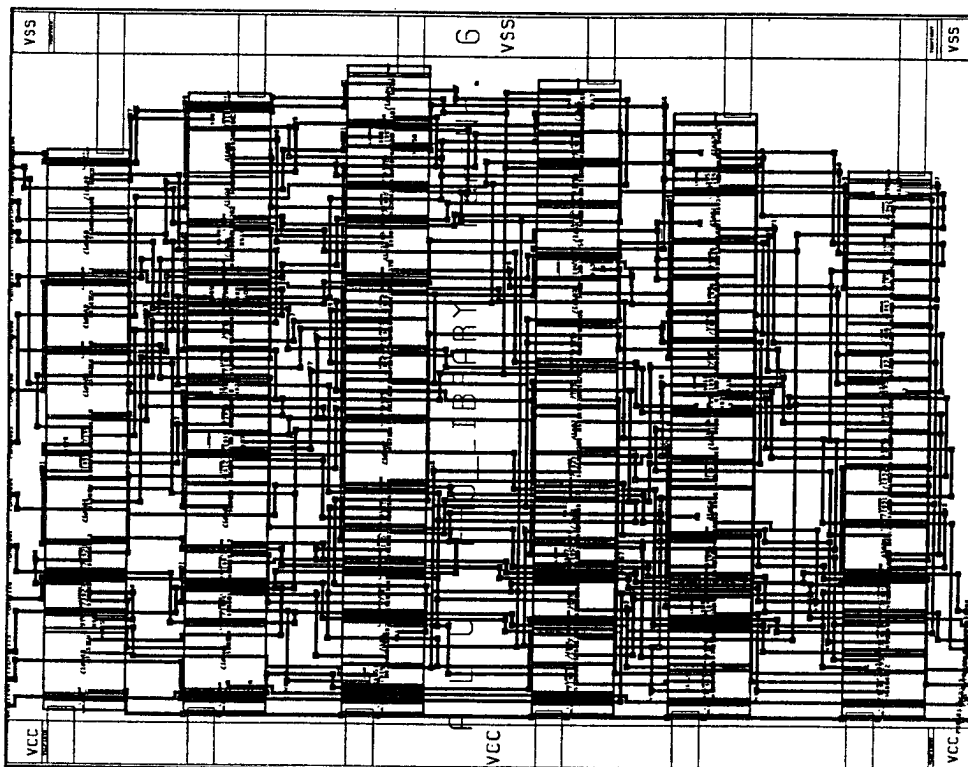


Abb. 12 HARDMAKRO der ALU in CHIPGRAPH

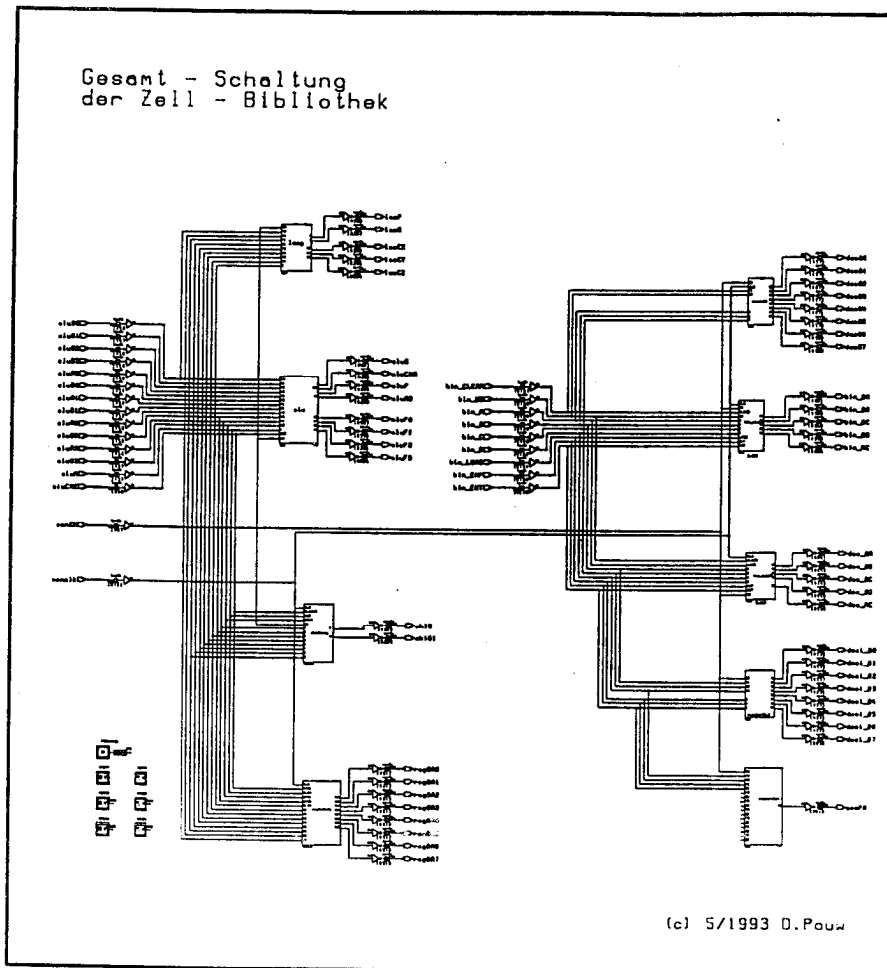


Abb. 13 Gesamtlayout des Testchips in NETED

Es wurden an der Fachhochschule Offenburg neun Hardmakros erzeugt und in eine eigene Bibliothek abgelegt. Abbildung 13 zeigt die Gesamtschaltung in NETED (um Eingangspins und damit Chipfläche zu sparen, wurden mehrer Eingänge zusammengeschaltet), Abbildung 14 zeigt den fertigen Chip. Dieser wurde gefertigt, um die Funktionalität und das Timing der Hardmakros zu testen.

4. ZUSAMMENFASSUNG

Nachdem alle Vorarbeiten (Ports erstellen und aufnehmen) in CHIPGRAPH und NETED durchgeführt sind, kann jeder neuer Schaltplan (mit den entsprechenden Properties) mittels <es2_autocell> in ein Hardmakro gewandelt und mit <build_lib> in eine Bibliothek aufgenommen werden.

Probleme treten in erster Linie beim Programm <es2_cellplace> auf, das die Hardmakros nicht plazieren kann. Die Plazierung muß in CELLGRAPH erfolgen mit anschließendem Routen durch das Programm <es2_cellroute>.

Die automatische Erstellung von Hardmakros mittels <es2_autocell> erbringt keine zufriedenstellenden Ergebnisse. Jedes Makro muß anschließend manuell in CHIPGRAPH optimiert werden (Flächenoptimierung).

Auch der Autorouter bringt öfter unbefriedigende Ergebnisse (Signalpfade werden als Schleifen ausgeführt, obwohl genug Platz für eine direkte Verbindung besteht usw.). So sollte man das gesamte Design auch bezüglich des Routens überprüfen und optimieren.

Das fertige Chip-Layout wird mit dem Programm <translate> write in das GDS2-Format übersetzt und kann via E-Mail an GMD gesendet werden. Hier wird mit Design-Rule-Checkern das Layout auf Fehler überprüft und der Chip gefertigt.

Die Version 8.0 der MENTOR GRAPHICS Software ist wesentlich anwenderfreundlich und zur Erstellung von Hardmakros besser geeignet.

Eine exakte Beschreibung der Vorgehensweise zum Erstellen von Hardmakros kann über die
Fachhochschule Ulm
 bezogen werden.

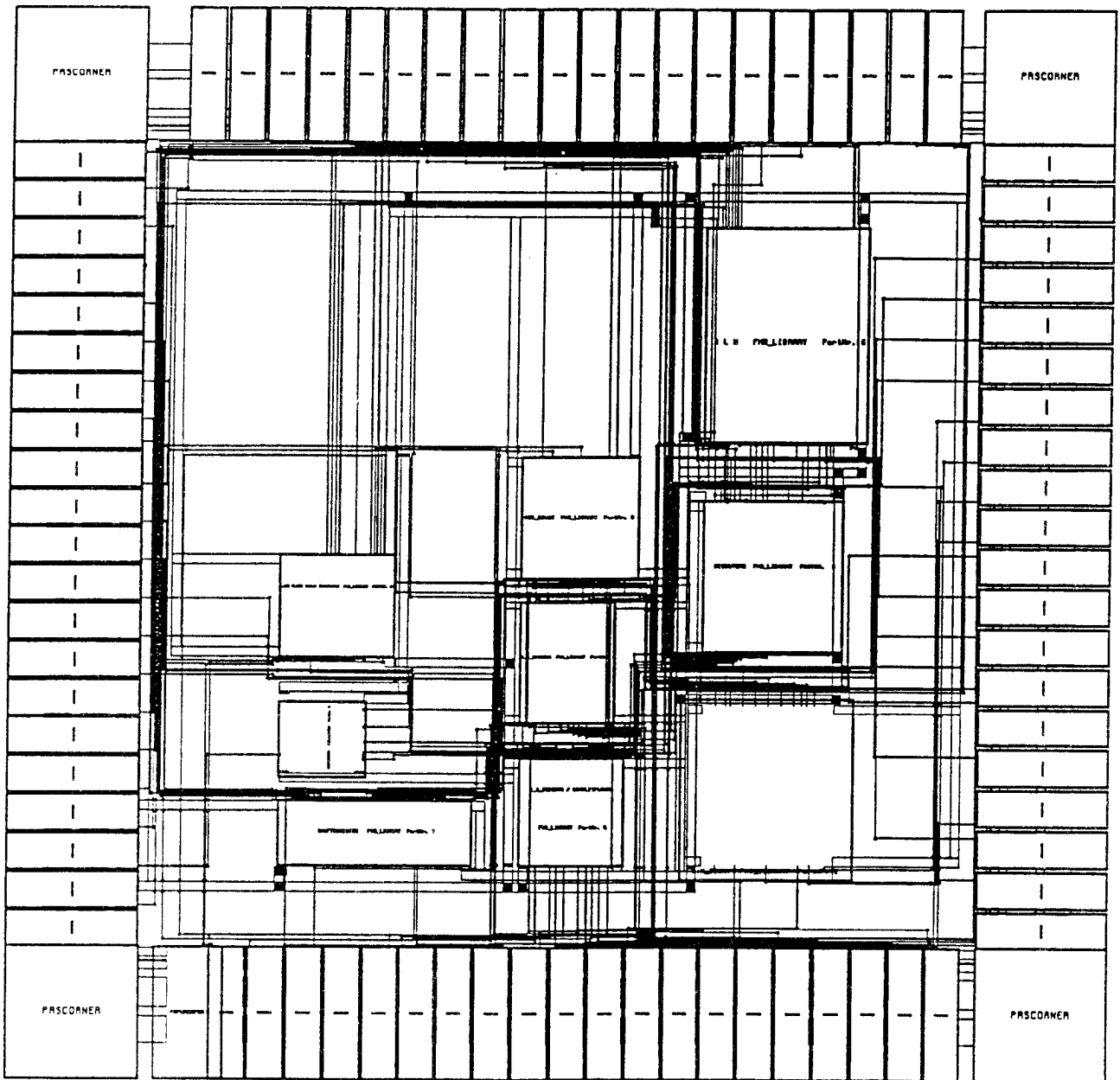


Abb. 14 Der fertige Chip in CHIPGRAPH mit allen Hardmakros