



Erschließung einer 0.35µm Technologie für die Bearbeitung mit Mentor Tools

Dipl.-Ing. (FH) Markus Striebel, Prof. Dr.-Ing. Dirk Jansen, ASIC Design Center
 Fachhochschule Offenburg, Badstraße 24, 77652 Offenburg
 Tel. 0781/205-274, Fax 0781/205-174
 markus.striebel@fh-offenburg.de

Im ASIC-Design-Center der FH-Offenburg wird ein Design Kit für die AMI 0.35µm Technologie erstellt. Der Design-Kit ist eine Zusammenstellung aller Dateien, die für ein Chipdesign unter Verwendung der Mentor Tools für den AMI 0.35µm Prozess benötigt werden.

1. Einführung

Der EKG-Datenlogger (Abb. 1), der derzeit in der 0.5µm Technologie der Firma AMI vorliegt, soll nun in der 0.35µm Technologie entworfen werden. Für den Entwurf des Digitalteils werden somit Standardzellen der Firma AMI verwendet. Für die 0.5µm Technologie stand ein kompletter Design-Kit zur Verfügung. Dieser Design-Kit (EP_MTC_CMOS05_v1.0) wurde von Europractice entwickelt. Für die 0.35µm Technologie ist derzeit kein kompletter Design-Kit verfügbar. Erhältlich sind GDS2 -Beschreibungen der Zellen, Technologie-Dateien für Synopsis, Vital-Codes für die Simulation, und Rules-Files für Verifikation der gerouteten Zelle mit Calibre. Wichtige Dinge wie z. B. Prozess-Dateien für das Routen oder auch Schaltplansymbole fehlen.

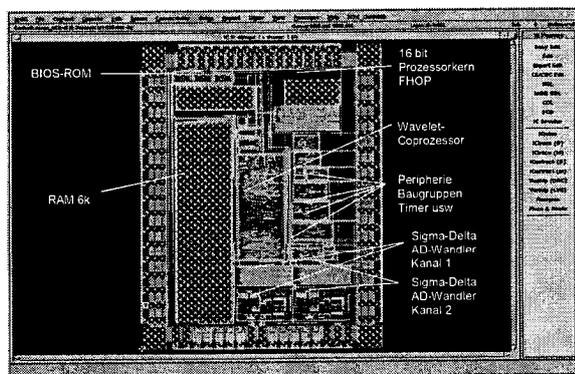


Abbildung 1: EKG-Datenlogger Chip mit Standardzellenentwurf in der 0.5µm Technologie von AMI.

Der Design-Kit FHO_MTC_CMOS035_v1.0, der all diese Dateien enthält, wird derzeit an der Fachhochschule Offenburg entworfen.

2. Technologievergleich

Ein wesentlicher Vorteil der AMI 0.35µm Technologie ist, dass 5 Metallisierungslagen zur Verfügung stehen. Damit lässt sich ein sehr viel kompakteres Routen durchführen. In diesem Zusammenhang ist es nun auch von Vorteil, direkte Durchkontaktierungen (stacked vias) von einer tieferliegenden Metallschicht (z. B. Metall 1) in eine höherliegende Metallschicht (z. B. Metall 5) in der Prozess-Datei zu definieren. Somit lassen sich auch neue Konzepte für das Routen des „TOP-Sheets“ erstellen. Auf dieser obersten Ebene kann nun Metall 5 und Metall 4 nur für das „Power-Routen“ benutzt werden. Bemerkenswert ist auch, dass die internen Stromversorgungsleitungen der Zellen nur noch eine Breite von 1.4µm aufweisen. Bei der AMI 0.5µm Technologie betrug dieser Wert 4.7µm. Deshalb ist es nun wichtig, in regelmäßigen Abständen einen Metallisierungstreifen („power-bar“) für die Stromversorgung in das Design einzubauen.

	Mietec 0.5	Mietec 0.35
Anzahl der Metallisierungslagen	3	5
Einsatz von Stacked vias	Kein Vorteil Beim Routen	Routingergebnis Sehr viel besser
Breite der Powerleitungen Internal row	4.7 µm	1.4 µm
Minimale Leiterbahnenbreite Metal1	0.7 µm	0.5 µm
Abstand	0.8 µm	0.6 µm
Steilheit GM _{max} NMOS TYP (V _{DS} =0.1V, V _{bs} =0V, 27 degrees C)	W/L = 10/0.5 274 µAV	W/L = 10/0.35 578 µAV

Tabelle 1: Technologievergleich zwischen der 0.5µm und der 0.35µm von AMI [1], [2].



Auch die Analog-Zellen müssen im Zuge einer solchen Prozessumstellung modifiziert werden. Da sich die Steilheit G_{mmax} der Transistoren ändert, müssen alle Transistoren, die für z. B. Stromquellschaltungen oder Stromspiegelschaltungen eingesetzt werden, neu dimensioniert werden. Die Leiterbahnenbreite in den verschiedenen Metallisierungsschichten oder Abstände zwischen den Metallisierungsschichten werden kleiner. In Tabelle 1 ist der Wert für Metall 1 aufgeführt.

3. Entwurfsablauf

Anhand des Entwurfsablaufs (Abb. 2) soll gezeigt werden, welche Programme zum Einsatz kommen und welche Dateien für die einzelnen Programme benötigt werden. Für die Synthese mit dem Design Analyzer von Synopsis werden die Technologiebeschreibungen der Logikgatter verwendet. Diese Dateien (Bibliotheken) sind auf der Europractice CD vorhanden, sowohl als ASCII Dateien, so z. B. die Datei MTC45000.lib, die die Beschreibung für alle Standardzellen enthält, sowie in kompilierter Form (MTC45000.db). Entsprechend sind auch die Bibliotheken für die PAD-Zellen vorhanden. Für die Timing-Simulation nach erfolgter Synthese mit dem Programm Modelsim, werden die VHDL-Beschreibungen der Logikgatter benötigt, die sogenannten Vital-Codes. Auch diese sind auf der Europractice CD verfügbar.

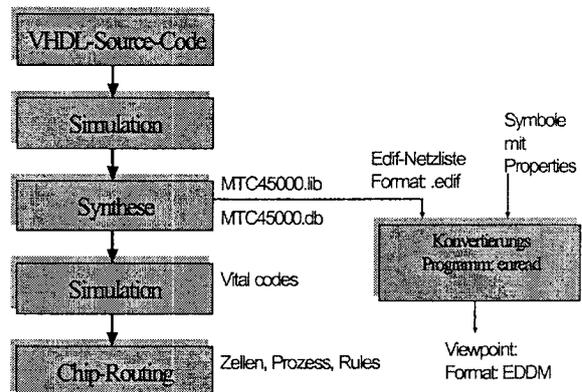


Abbildung 2: Entwurfsablauf beim Chipdesign.

Die Edif-Netzliste (edif-Format), die man als Syntheseergebnis erhält, kann nicht direkt zum Routen des Chips verwendet werden. Zunächst muss man mit dem Programm „enread“ das edif-Format (ASCII-Datei) in ein EDDM Format (Erzeugen eines Viewpoints) konvertieren (Abb. 2, Abb. 3). Dieser Schritt ist keine reine Konvertierung. Benötigt wird hierfür ein Verzeichnis, indem sich die Symbole und Schaltpläne der Logikgatter befinden. Diese Symbole dienen als Träger bestimmter Eigenschaften (Properties).

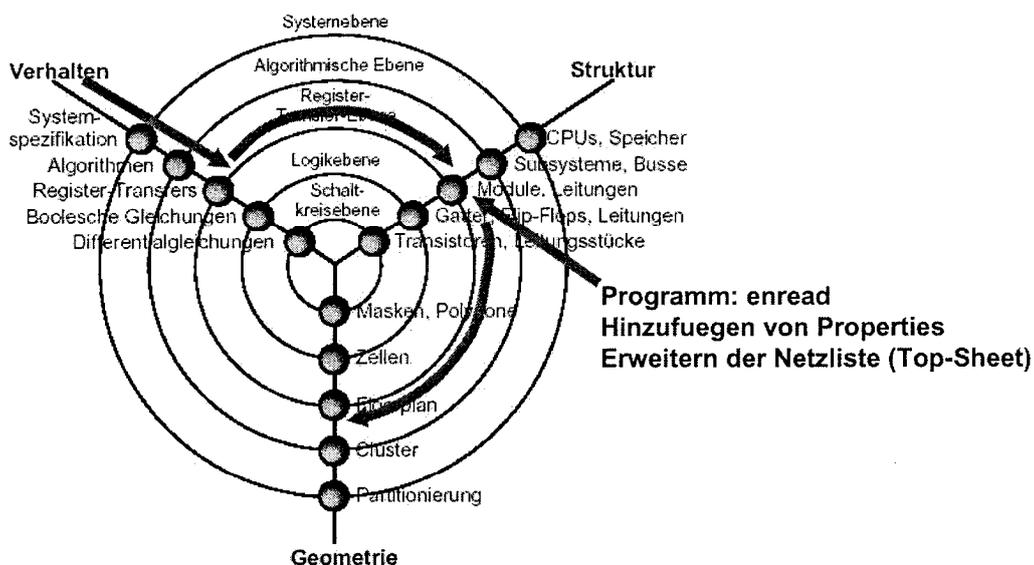


Abbildung 3: Entwurfsablauf beim Chipdesign mit Standardzellen im Gajski-Y verdeutlicht.



Konvertiert man also die Edif-Netzliste, so wird ein Viewpoint erzeugt (Binary Datei), der neben der reinen logischen Verknüpfung der Gatter (Informationsquelle ist die Edif-Netzliste) noch weitere Informationen enthält (Informationsquelle sind die Symbole mit den Properties).

Die Symbole und Schaltpläne der Logikgatter sowie der PAD-Zellen sind nicht erhältlich. Diese mussten mit Hilfe des Design-Architects von Mentor-Graphics entworfen werden.

Mit Hilfe der Viewpoint-Beschreibung seines Designs kann man mit dem Chip-Routen auf der IC-Station von Mentor-Graphics fortfahren. Für das Routen werden die Zellen im IC-Graph-Format benötigt. Vorhanden sind die Zellen jedoch nur im GDS2-Format. Die Zellen müssen mit Hilfe des Programms IC-Link von Mentor Graphics konvertiert werden. Dann müssen sie mit der IC-Station nachbearbeitet werden, da wichtige Dinge wie z. B. Portdefinitionen fehlen.

Selbst geschrieben werden mussten auch die Prozessdateien, die sämtliche Informationen für das Routen enthalten, wie z. B. verwendete Routing-Layer). Die Rules-Dateien (LVS, DRC) die für die Verifikation des gerouteten Designs mit dem Programm Calibre benötigt werden, waren vorhanden. Für einen analogen Designentwurf werden Schaltplansymbole für Transistoren, Widerstände und Kondensatoren benötigt. Auch diese müssen selbst entworfen werden. Die Properties für die Simulation mit dem Programm Eldo, und die Properties für die Device Generatoren, welche in der Prozess-Datei definiert werden, müssen eingefügt werden. Die Spicefiles der Transistoren (BSIM3) sind verfügbar.

4. FHO_MTC_CMOS035_v1.0, der Design Kit für die AMI 0.35µm Technologie

In Abb. 4 ist der Aufbau des Design-Kit dargestellt. Der Ordner *autocells* enthält die Dateien, die für das Routen mit dem Programm Autocells benötigt werden. Das ist zum einen die Autocells Technologie Datei **mtc45000** (ASCI-Format) und die Bibliothek-Datei mit den Zellen-Beschreibungen **mtc45000.L** (ASCI-Format). Außerdem die Datei **Iconvt.opt** (ASCI-Format) mit den Pfaden zu den oben erwähnten Dateien und die Datei **Autocells.PAR_user** (ASCI-Format), in der man die Möglichkeit hat, bestimmte Vorgaben für den Router anzugeben.

Der Ordner *GDS2* enthält die GDS2-Beschreibungen der Zellen.

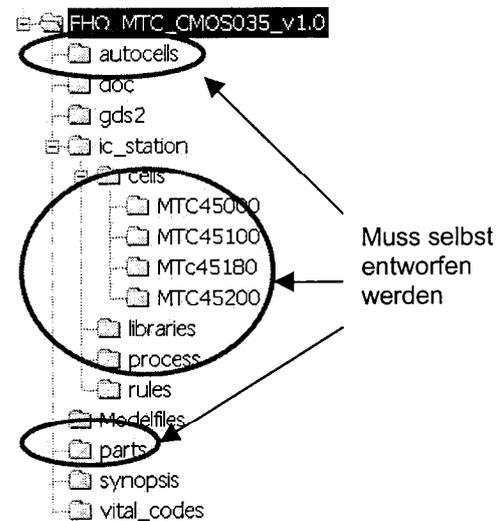


Abbildung 4: Bestandteile des Design-Kits FHO_MTC_CMOS035_v1.0.

Der Ordner *ic_station* enthält Unterordner und Dateien, die für das Routen mit der IC-Station benötigt werden. Der Ordner *cells* enthält den Unterordner **MTC45000** mit den für die IC-Station aufbereiteten Standardzellen, den Ordner **MTC45100** mit den MID-Profile-IO-Zellen, den Ordner **MTC45180** mit den PAD-Limited-IO-Zellen, und den Ordner **MTC45200** mit den Core-Limited-IO-Zellen. Der Ordner *libraries* enthält entsprechend die Bibliotheken **mtc45000**, **mtc45100**, **mtc45180**, **mtc45200**. Der Ordner *process* enthält die Prozesse **block_process**, **top_process** und **analog_process**. Der Ordner *Rules* enthält die Dateien für die Verifikation des Design mit dem Programm Calibre. Eine Datei für LVS und eine Datei für DRC.

Für die Anlogsimulation ist der Ordner *Modelfiles* vorhanden, in dem sich die Spice-Beschreibungen für die verschiedenen Transistor-Modelle befinden. Das sind die Dateien **ntyp.dat**, **ptyp.dat**, **nslow.dat**, **pslow.dat**, **nfast.dat** und **pfast.dat**.

Der Ordner *parts* enthält die gleichen Unterordner wie der Ordner *cells*, jeder Ordner enthält die entsprechenden Symbole und Schaltpläne. Außerdem ist der Unterordner *devices* vorhanden, der die Symbole für die Analog-Devices enthält (Transistoren, Widerstand, Kondensator).

Der Ordner *Synopsis* enthält die Technologiebeschreibungen der Gatter für die Synthese mit Synopsis.



Der Ordner *vital_codes* enthält die vhd-Beschreibungen der Logikgatter für die Simulation mit Modelsim.

5. Entwurf der Schaltplansymbole für den neuen Design Flow mit ADMS

Die Schaltplansymbole und die Schaltpläne der Logikgatter müssen mit dem Programm Design-Architect gezeichnet werden. Die Symbole müssen die richtigen Properties für die Simulation mit ADMS (Advanced Mixed Signal Simulator) und für das Routen mit der IC-Station erhalten. Für die Simulation sind die beiden Properties **View** und **Model** [2] von Bedeutung. Für jedes Symbol können mehrere Modelle im Design Architect registriert werden. Für ein Modell gibt es drei verschiedene Beschreibungsmöglichkeiten:

- eine compilierte HDL (VHDL, VHDL-AMS, Verilog),
- eine Spice-Netzliste für die Simulation mit dem ELDO-Kernel, der in das Programm ADMS integriert ist,
- oder ein Schaltplan.

Über das Property **Model** kann dann gewählt werden, welches Modell für die Simulation verwendet werden soll. Mit Hilfe des Properties **View** kann man auswählen, welche der drei Beschreibungsmöglichkeiten bei einem Doppelklick auf das Symbol angezeigt werden soll. Für die Digitalsymbole wird also ein Modell mit dem compilierten Vital-code erzeugt. Die Analog-Devices müssen mit den richtigen Properties für die Simulation mit dem **Eldo**-Kernel versehen werden.

Der Vorteil des neuen Design-Flows mit ADMS ist, dass eine Schaltung simuliert werden kann, bestehend aus einem analogen Block, der im Spice-Format beschrieben ist, und einem digitalen Block, der in einer HDL-Beschreibung vorliegt. An das Programm ADMS wird dann eine Top-Spice-Netzliste übergeben. Für den Digitalteil wird dann der integrierte Modelsim-Kernel verwendet, für den Analogteil der integrierte Eldo-Kernel. Mit Hilfe des Viewers *Xelga* wird das Endergebnis angezeigt.

Entsprechend müssen alle Symbole noch mit den Properties für die IC-Station versehen werden.

6. Zellenaufbereitung für die IC-Station

Nach der Konvertierung der Zellen vom GDS2-Format in das IC-Graph-Format müssen die Zellen nachbearbeitet werden. Zunächst muss ein Prozess für die Zelle geladen werden. Dann müssen Ports definiert werden, Flächen, die keine Ports sind, müssen mit einem Blockage-Layer versehen und der Shape-Aspect aller Strukturen muss von *internal* auf *internal/external* umgestellt werden. Damit erst werden die Strukturen in einer höheren Hierarchie-Stufe sichtbar. Eine andere Möglichkeit wäre gewesen, die Zellen im LEF-Format (Cadence) zu importieren. Dies funktioniert, jedoch haben die Zellen andere Strukturen, wie Abb. 5 zeigt. Es wurde deshalb entschieden, die GDS2-Beschreibungen zu verwenden.

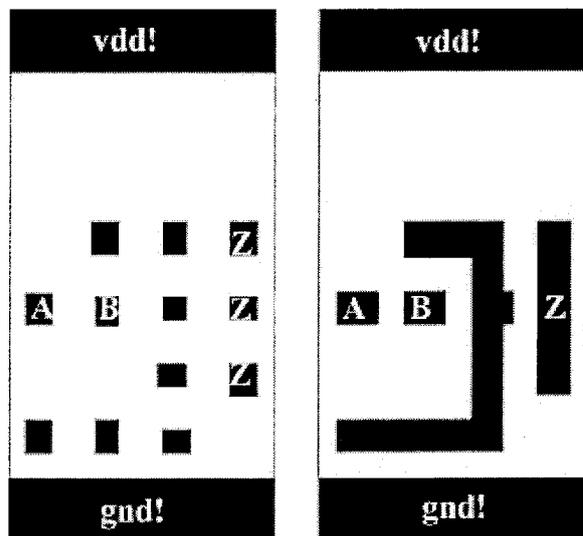


Abbildung 5: GDS2 konvertierte Zelle (AN2) und LEF-Zelle im Vergleich.

7. Entwurf der Prozess-Dateien

Für das Routen mit der IC-Station müssen die Prozess-Dateien geschrieben werden. Eine Datei für das analoge Routen (Abb. 6), eine Datei für das Block-Routen und eine Datei für das Routen der Top-Zelle. In der Prozess-Datei stehen alle wichtige Vorgaben für den Router. Hier wird definiert, welche Layer für das Routen verwendet werden oder wie groß der Abstand zwischen den Metall-Lagen sein muss. In der Prozess-Datei für das analoge Routen werden außerdem die Device-Generatoren definiert. Dadurch ist es möglich, sich in der IC-Station ein analoges Device automatisch aus dem Schaltplan-Symbol erzeugen zu lassen. Ein Transistor-Symbol verfügt über die Properties Gatelänge und Gatebreite (*l_ic* und *w_ic*), alle



anderen Spezifikationen für den Entwurf des Transistors sind in der Prozess-Datei enthalten. Diese Spezifikationen können aus den Design-Rules abgeleitet werden. Geschrieben wurden Device-Generatoren für PMOS, NMOS, Kapazität, Widerstand, Stacked-Vias und Guard-Band.

```

////////////////////////////////////
// File: analog_process.puf
//gekürzte Ausschnitte
////////////////////////////////////
extern ic_process@@$process_name = "MTC45000";
extern ic_process@@$layer_spacing = [
["POLYSILICON", "POLYSILICON", 0.55],
["POLYSILICON", "polysilicon.blkg", 0.55],
["POLYSILICON", "ACTIVE_AREA", 0.3], ];
extern ic_process@@$port_height = 0.7;
extern ic_process@@$routing_level = [
["POLYSILICON", @vertical, 0.5, "POLYSILICON",
"polysilicon.blkg"],
["METAL1", @horizontal, 0.5, "METAL1", "metal1.blkg"],
["METAL2", @vertical, 0.7, "METAL2", "metal2.blkg"],];

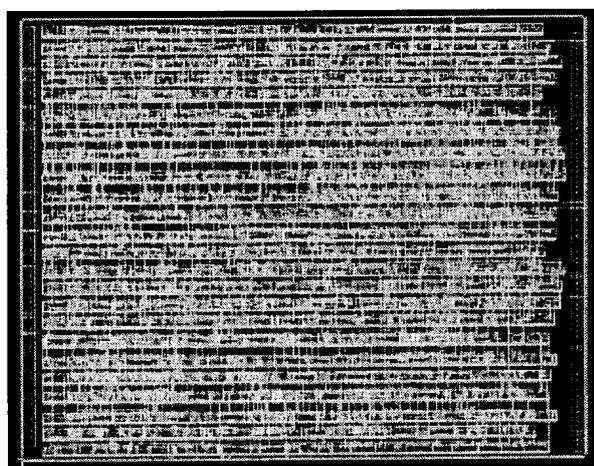
////////////////////////////////////
//PMOS Device Definitions, Device-Generator
//gekürzte Ausschnitte
////////////////////////////////////
extern ic_process@@$dev_layer_mos_mp = [
["contact_layer", "CONTACT"]
["gate_layer", "POLYSILICON"] ];
extern ic_process@@$dev_rule_mos_mp = [
["contact_size", [0.4,0.4]],
["gate_to_gate", 0.55] ];
extern ic_process@@$dev_info_mos_mp = [
["mos_flip", @noflip]
["mos_rotate", 0] ];
    
```

Abbildung 6: Ausschnitt aus dem AMI 0.35 Prozess für analoges Routen

Verifikation des Design-Kits anhand der Entwicklung des Lottozahlengenerator-Chips

Die durchgeführten Arbeiten wurden am Entwurf eines Lottozahlengenerator-Chips verifiziert. Die Funktionsweise des Chips orientiert sich am Lotteriespiel 6 aus 49, wie es im deutschen Fernsehen zu sehen ist.

Bei Betätigung eines Tasters wird durch das Vorhandensein eines Zufallsgenerators eine Zahl ausgewählt, die durch das Aufleuchten einer LED angezeigt wird. Sind sieben Zahlen ausgewählt (sechs + Zufallszahl) so ertönt eine Melodie. Die 32kHz Frequenz für den Digitalteil (Abb. 7) wird durch den analogen Entwurf eines 32kHz Oscillators (Abb. 8) realisiert. Durch diesen Mixed-Signal-Chip (Abb. 9), der gefertigt wird, können alle geleisteten Arbeiten verifiziert werden.



Gesamtgröße:
623 µm/493 µm

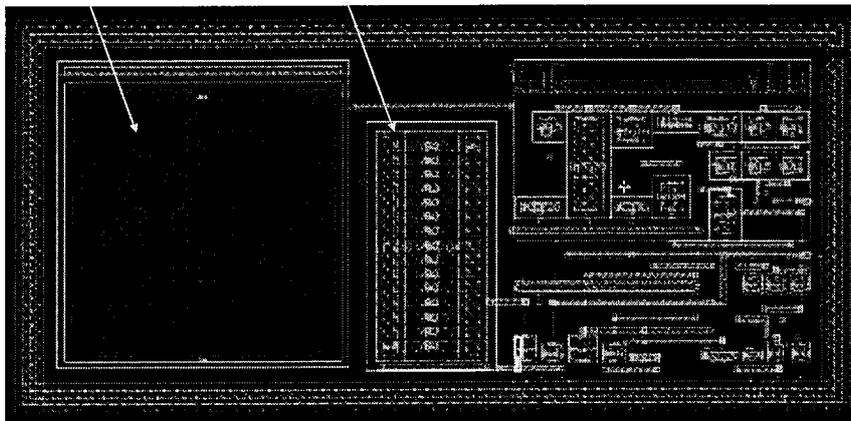
Standardzellenentwurf

Zellenanzahl: 1134

Abbildung 7: Digitalteil des Lottozahlengenerator-Chip mit Standardzellenentwurf der AMI 0.35µm Technologie.

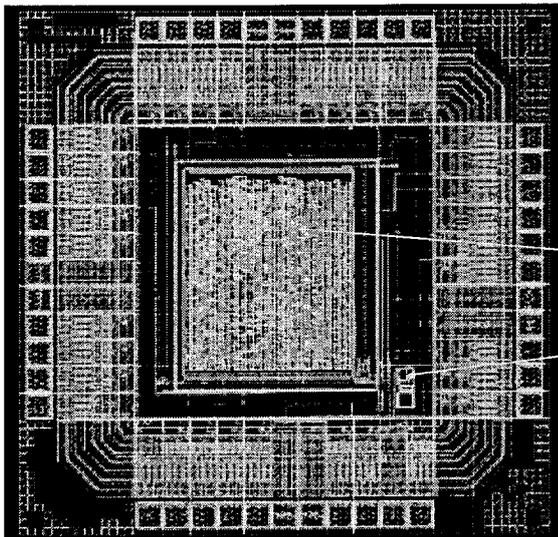


Kondensator 1.5pF Widerstand 132k Gesamtgröße: 119 µm/56 µm



Entwurf mit Devicegeneratoren

Abbildung 8: Analogteil des Lottozahlengenerator-Chip, 32kHz Oszillator.



Gesamtgröße :
1553 µm/1553 µm

Digitalteil mit
Standardzellen

Analogteil
OSCI32kHz

Abbildung 9: Der Lottozahlengenerator-Chip mit Digitalteil, 32kHz Oszillator und für die IC-Station aufbereiteten Padzellen.

8. Zusammenfassung

An der Fachhochschule Offenburg wird der Design-Kit **FHO_MTC_CMOS_035_v1.0** erstellt. Mit Hilfe dieses Kits lassen sich Designs in der AMI 0.35µm Technologie entwerfen. Alle durchgeführten Arbeiten werden durch den Entwurf eines Lottozahlengenerator-Chips verifiziert, der gefertigt wird. Damit sind alle wesentlichen Schritte bekannt, die für die Aufbereitung eines Design-Kits für beliebige Technologien für die Mentor-Tools erforderlich sind. Der Design-Kit wird für alle MPC-Mitglieder freigegeben, die eine NDA für AMI bei Europractice unterzeichnet haben.

9. Referenzen

[1] AMI : C035M-A Design-Rule Manual, Document DS13337

[2] AMI : C035M-D Design Rule Manual, Document DS13330

[3] IC-Flow Bookcase