03M1 Lecture
Chip Design

Martin Grötschel

Beijing Block Course
"Combinatorial Optimization at Work"
September 25 – October 6, 2006

Martin Grötschel

- Institut für Mathematik, Technische Universität Berlin (TUB)
- DFG-Forschungszentrum "Mathematik für Schlüsseltechnologien" (MATHEON)
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

groetschel@zib.de http://www.zib.de/groetschel

Contents

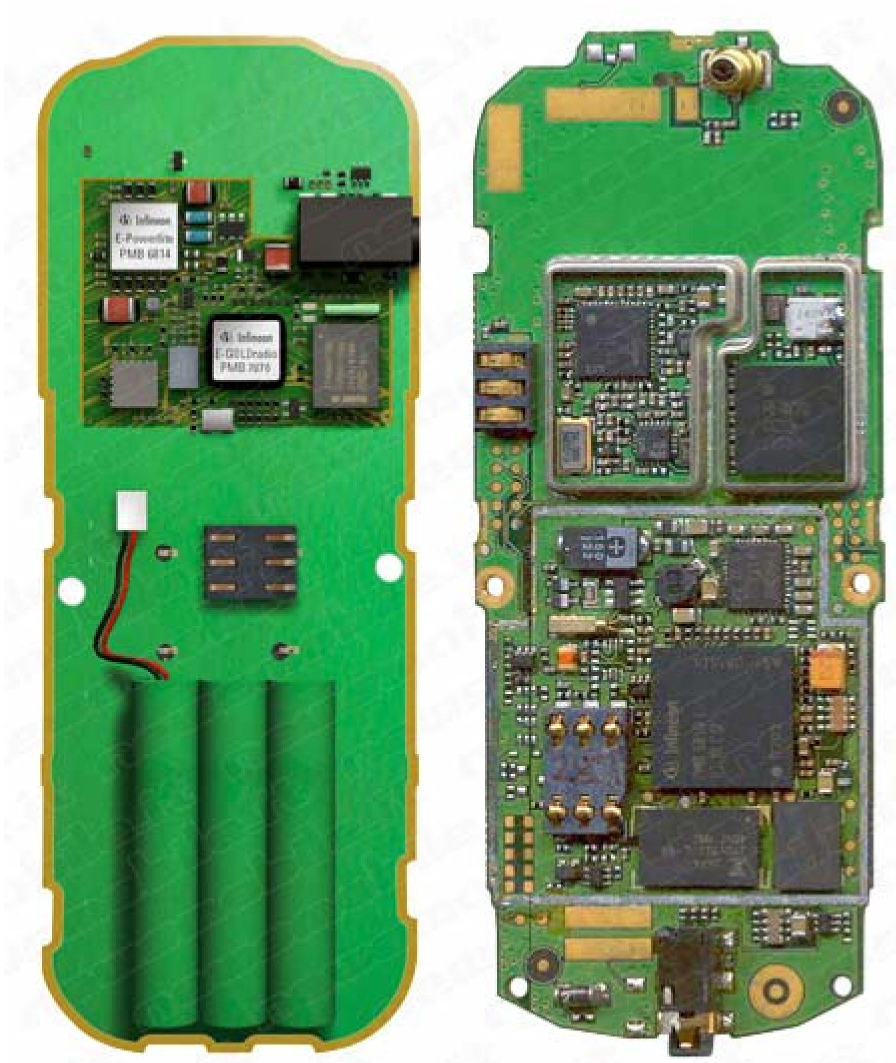
Martin Grötschel

Contents

Integrated Circuits, Chips, Microprocessors

- An **integrated circuit (IC)** is a thin **chip** consisting of at least two interconnected semiconductor devices, mainly transistors, as well as passive components like resistors. As of 2004, typical chips are of size 1 cm^2 or smaller, and contain millions of interconnected devices, but larger ones exist as well.
- Among the most advanced integrated circuits are the **microprocessors**, which drive everything from computers to cellular phones to digital microwave ovens. Digital memory chips are another family of integrated circuits that are crucially important in modern society.

The interior of a cellular phone

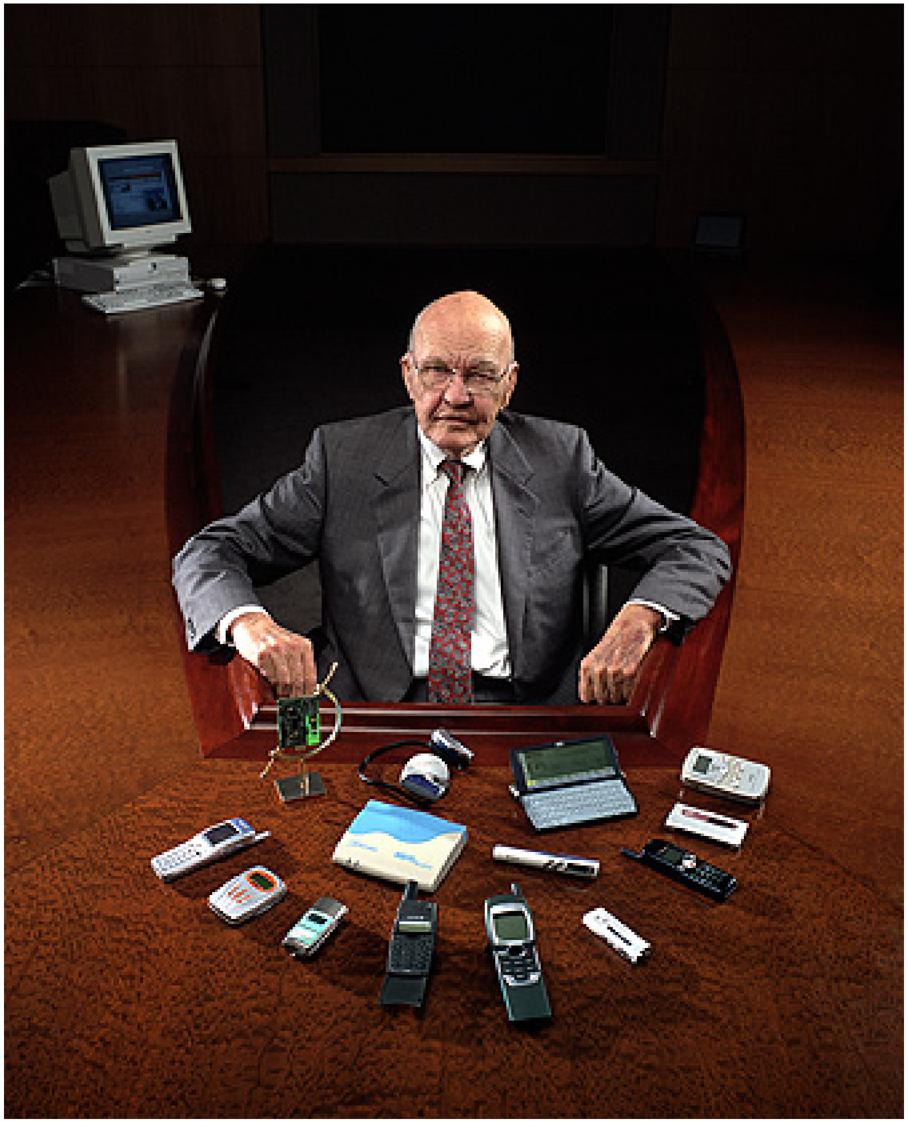


Martin Grötschel

Integrated Circuits: History

- The integrated circuit was first conceived by a radar scientist, Geoffrey W.A. Dummer (born 1909), working for the Royal Radar Establishment of the British Ministry of Defence, and published in Washington DC on May 7, 1952. Dummer unsuccessfully attempted to build such a circuit in 1956.
- The first integrated circuits were manufactured independently by two scientists:
- Jack Kilby of Texas Instruments filed a patent for a "Solid Circuit" made of germanium on February 6, 1959. Kilby received patents US3138743, US3138747, US3261081, and US3434015. He received the physics Nobel Prize in 2000, (See the Chip that Jack built (http://www.ti.com/corp/docs/kilbyctr/jackbuilt.shtml) for more information.)
- Robert Noyce of Fairchild Semiconductor was awarded a patent for a more complex "unitary circuit" made of Silicon on April 25, 1961.
- Noyce credited Kurt Lehovec of Sprague Electric for the *principle of dielectric isolation* caused by the action of a p-n junction (the diode) as a key concept behind the IC.

Kilby & Noyce



Jack Kilby (1923 – 2005)

Robert Noyce (1927 – 1990)
cofounder of Fairchild and Intel

Leslie Berlin wrote a biography about Noyce in June 2005 entitled "The Man Behind the Microchip: Robert Noyce and the Invention of Silicon Valley".

Integrated Circuits:Growth of Size

- **SSI:** The first integrated circuits had only a few transistors. Called "Small-Scale Integration", they used circuits containing transistors numbering in the tens.
- **MSI:** The next step in the development of integrated circuits, taken in the late 1960s, introduced devices which contained hundreds of transistors on each chip, called "Medium-Scale Integration" (MSI).
- **LSI:** Further development, driven by the same economic factors, led to "Large-Scale Integration" in the 1970s, with tens of thousands of transistors per chip.
- **VLSI:** The final step in the development process, starting in the 1980s and continuing on, was "Very Large-Scale Integration" (VLSI), with hundreds of thousands of transistors and now well past several million.
- **WSI:** The most extreme integration technique is wafer-scale integration (WSI). Attempts to take this step commercially in the 1980s (e.g. by Gene Amdahl) failed.
- **SOC:** Advances in semiconductor manufacturing allowed for another attack on the IC complexity: System-on-Chip (SOC) design. In this approach, components traditionally manufactured as separate chips to be wired together on a printed circuit board, are designed to occupy a single chip that contains memory, microprocessor(s), peripheral interfaces, Input/Output logic control, data converters, etc., i.e., the whole electronic system

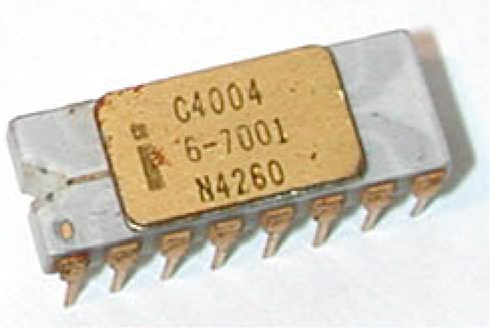
Integrated Circuits: Complexity

- Digital integrated circuits can contain anything from one to millions of logic gates, flip-flops, multiplexers, etc. in a few square millimeters. The small size of these circuits allows high speed, low power dissipation, and reduced manufacturing cost compared with board-level integration.
- The growth of complexity of integrated circuits follows a trend called "Moore's Law", first observed by Gordon Moore of Intel. Moore's Law in its modern interpretation states that the number of transistors in an integrated circuit doubles every two years. By the year 2000 the largest integrated circuits contained hundreds of millions of transistors. It is difficult to say whether the trend will eventually slow down.
- The integrated circuit is one of the most important inventions of the 20th century. Modern computing, communications, manufacturing and transport systems, including the Internet, all depend on its existence.

Martin Grötschel

Integrated Circuit:Examples

The **Intel 4004**, a 4-bit CPU, was the world's first single-chip microprocessor, as well as the first commercial one. The "gold and white with gray traces" specimen shown belongs to the initial CERDIP type series manufactured in 1971.



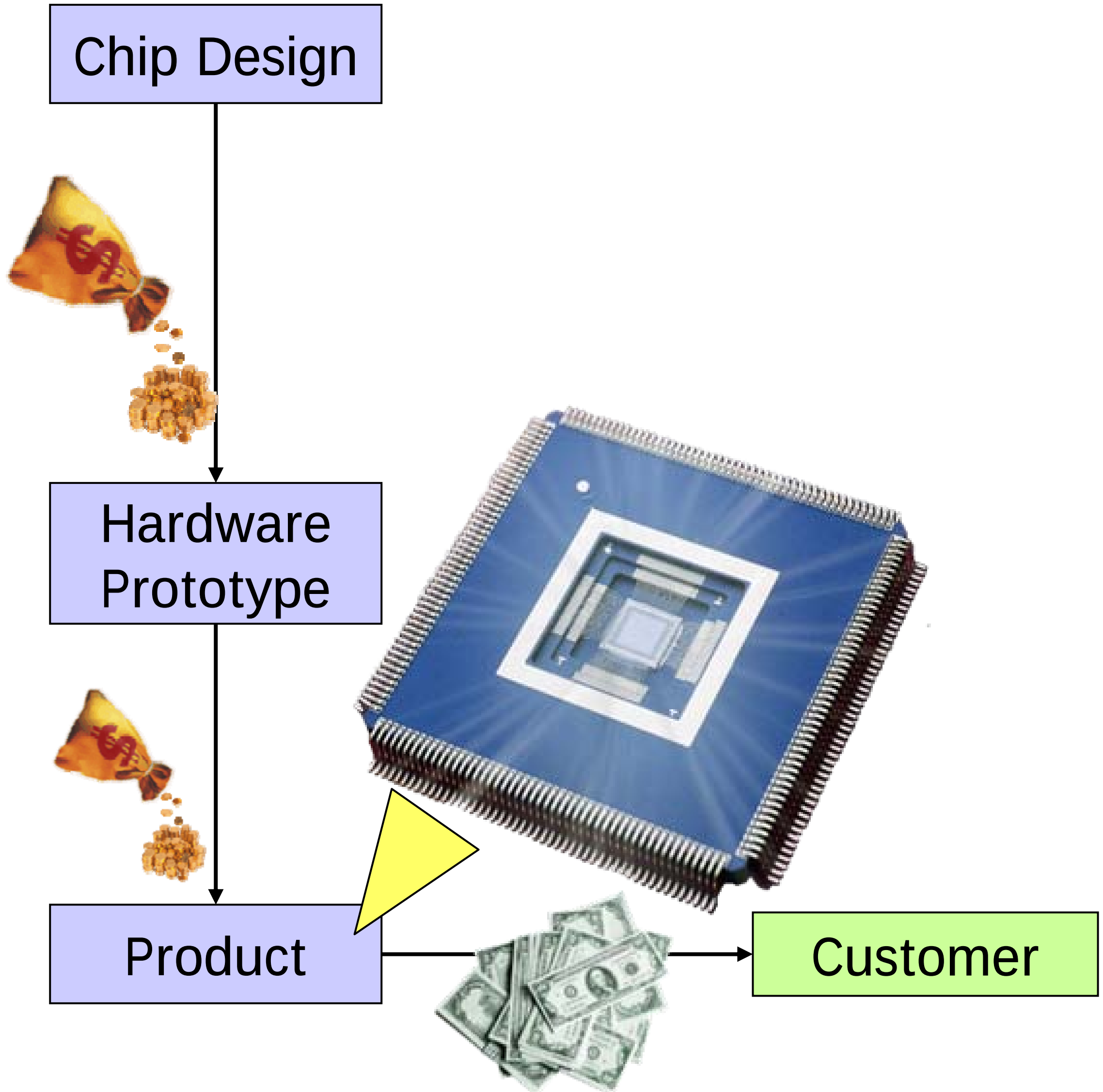
Contents

Martin Grötschel

The “Logic Phase”

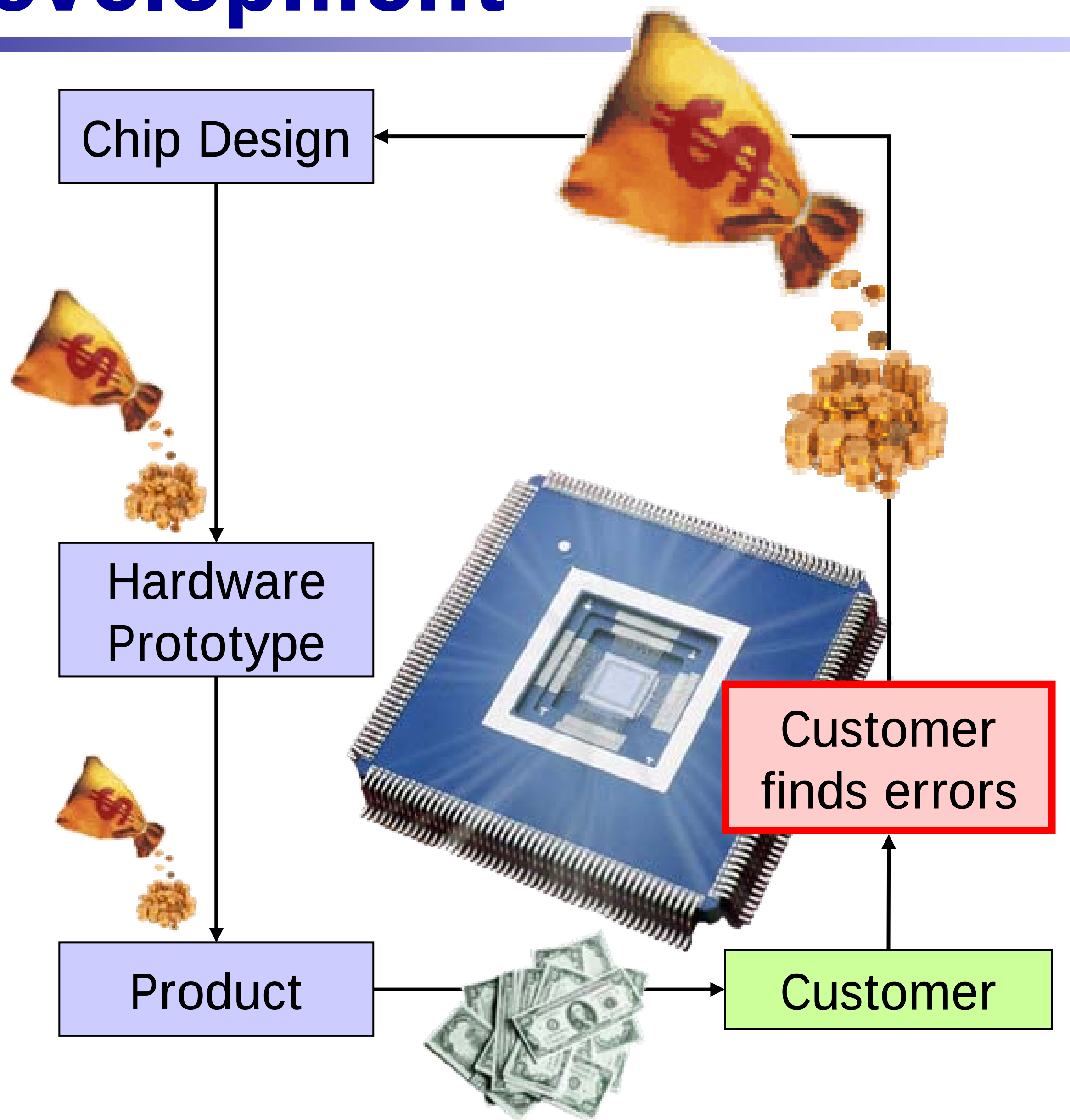
- Decision on tasks to be addressed
- Task partitioning (PhD Thesis Carlos Ferreira, TU Berlin 1994)
- Rough logic design
- Decision on chip technology
- Detailed logic design based on components libraries
- Logic verification (Ph.D Thesis Tobias Achterberg, TU 2006)

Chip Development



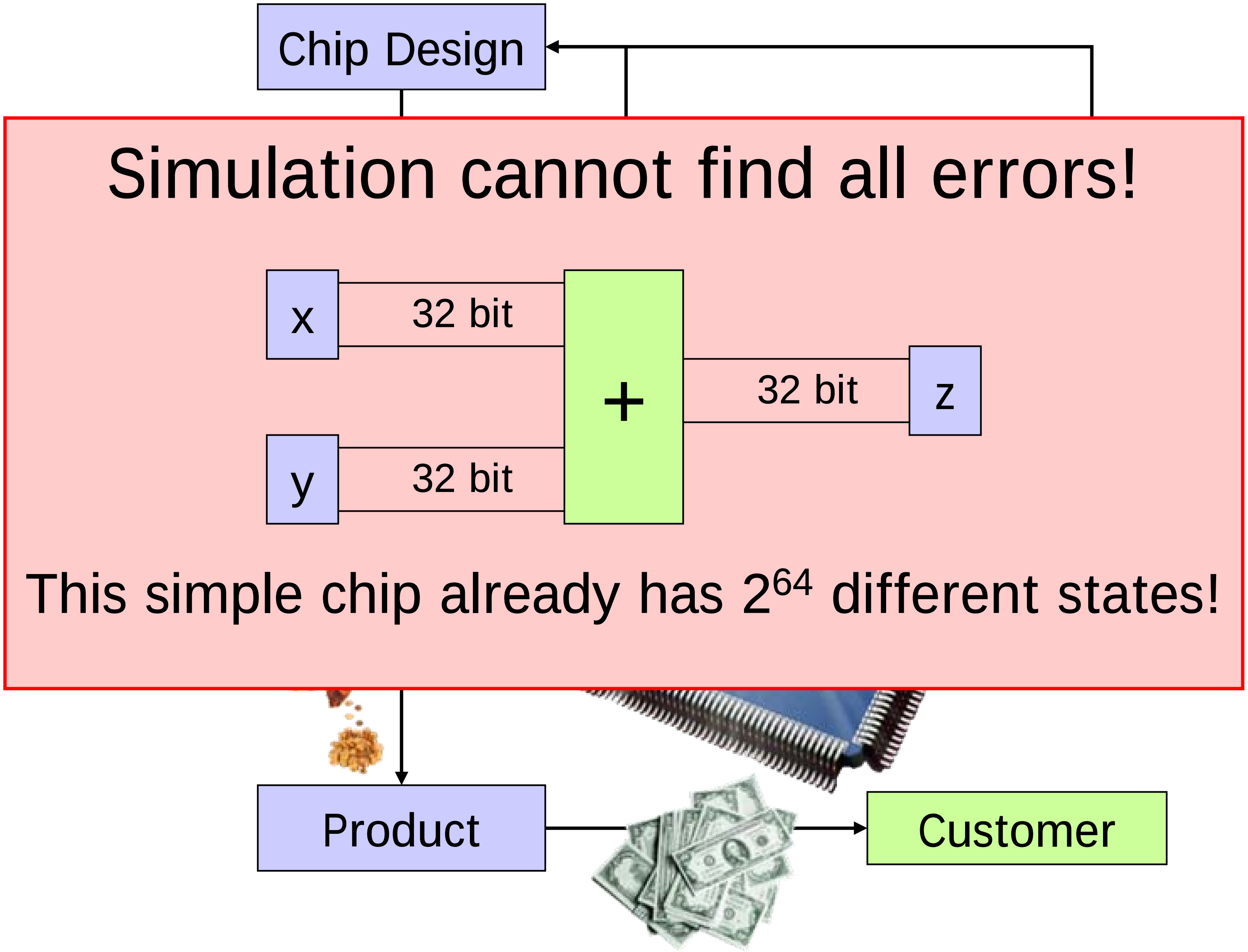
Martin Grötschel

Chip Development

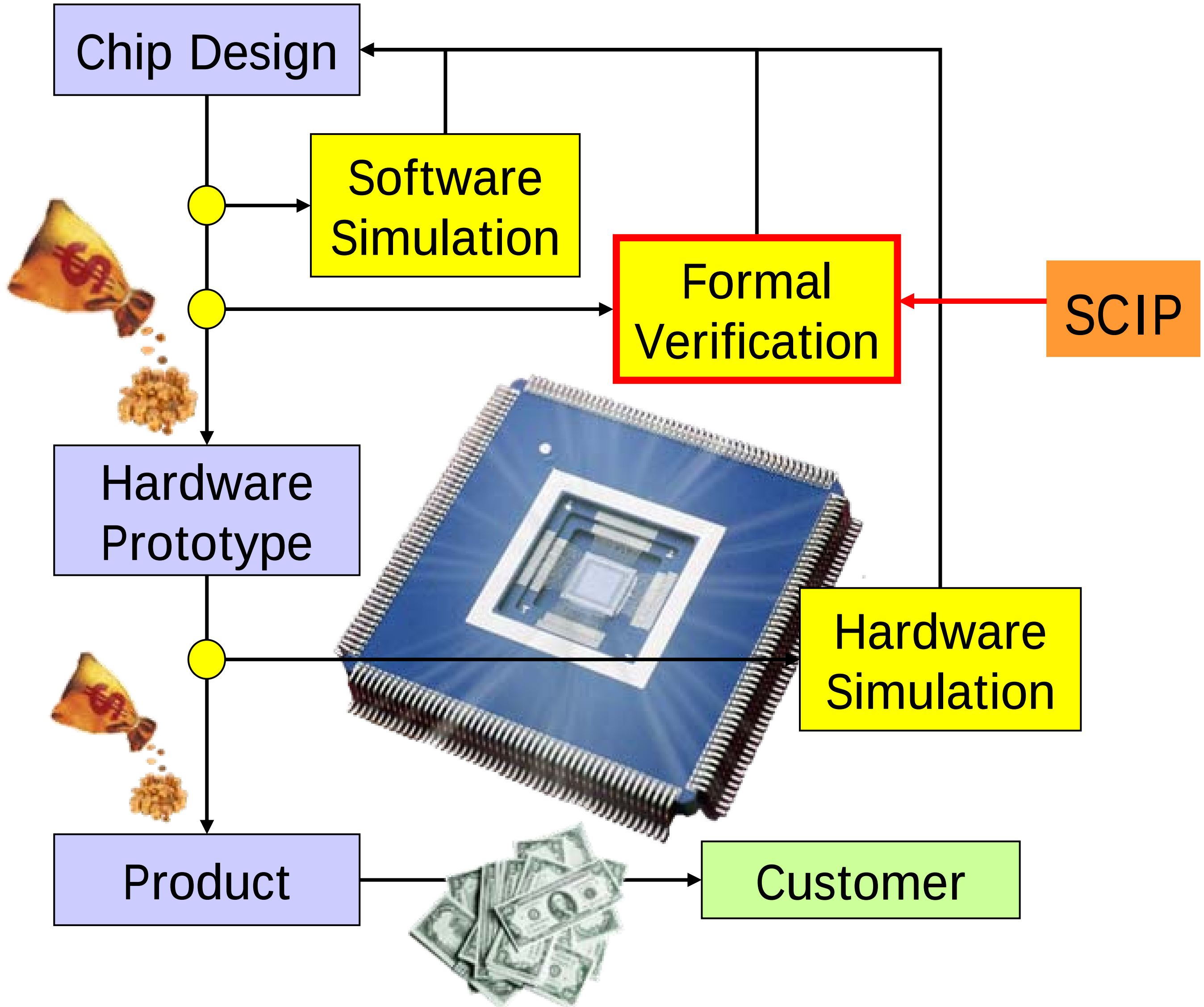


Martin Grötschel

Chip Verification: Simulation



Formal Chip Verification



Chip Design

Chip technology has been chosen, logic design exists:

- Global placement
- Local placement
- Global (homotopic) routing
- Local routing
- Layer assignment & via minimization
- Compactification

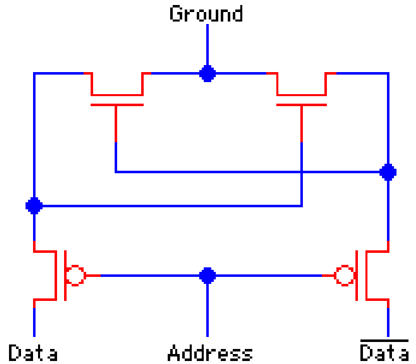
} combinatorial optimization

- logic simulation
- runtime simulation

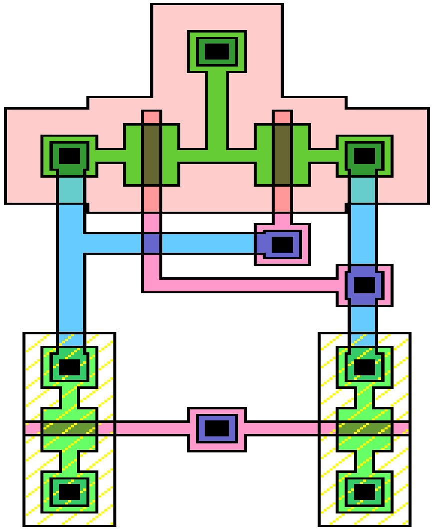
} differential equations

Martin Grötschel

Chip-Design

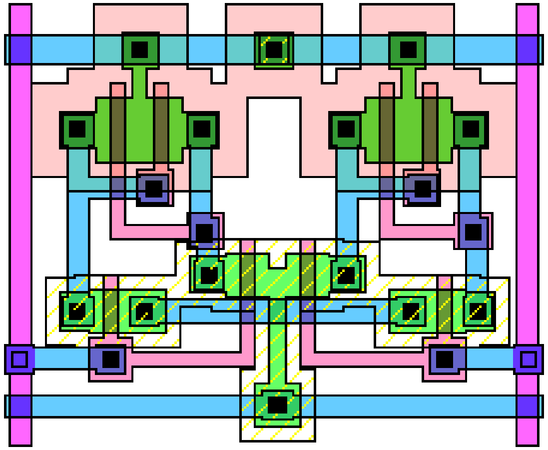


Schematic for four-transistor static-memory cell.

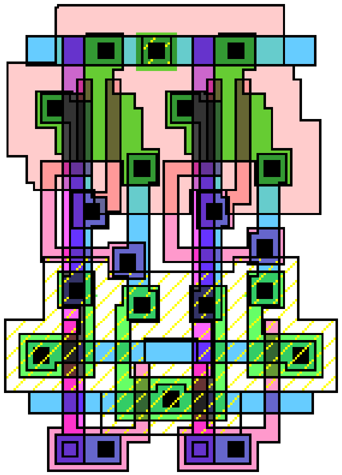


CMOS layout for four-transistor static-memory cell

placement
routing
compactification



CMOS layout for two four-transistor static-memory cells.



Compacted CMOS layout for two four-transistor static-memory cells.

Chip Production

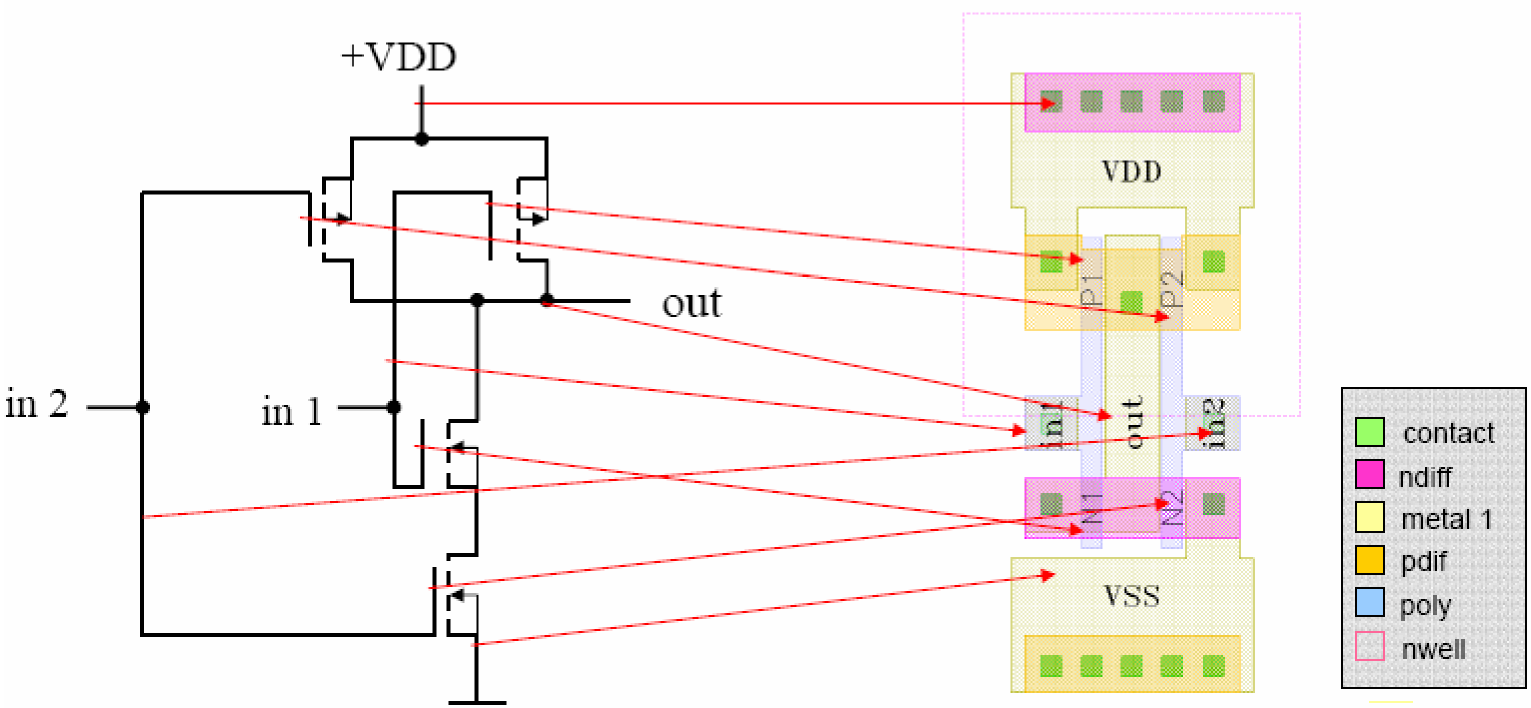
Problems depend on technology chosen, typical issues:

- Wafer production, e.g. crystal growing
- Mask drawing
- Sequencing of the production line
- Online control of the production line
- Control and optimization of various machines
- Optimization of the material flow
- Physical testing: Design of test sequences

From Logic to Geometry &Function

how to come from a simple transistor circuit plan

such a well dimensioned geometrical plan



Martin Grötschel

Silicon from Burghausen am Inn

Siltronic is a leading Manufacturer and global Supplier of hyperpure, electronic Grade Silicon to the Semiconductor Industry

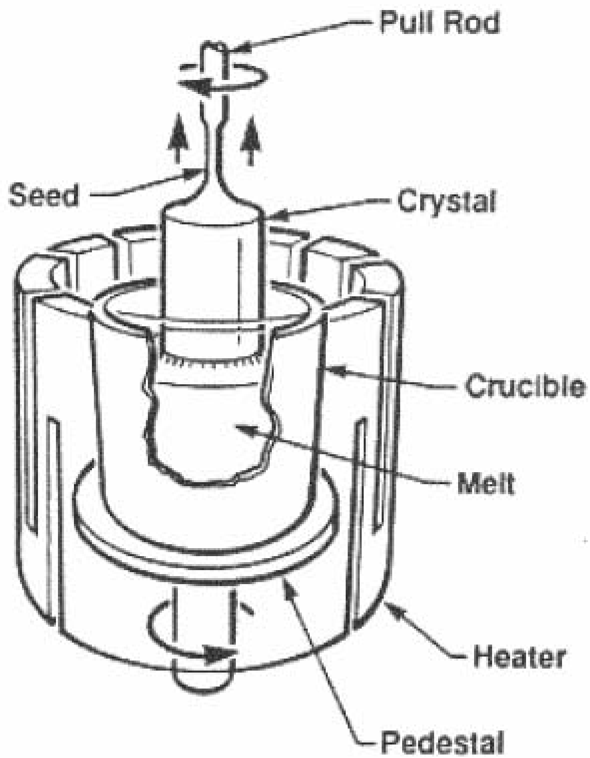
- 100% part of Wacker Group, Germany
- Global Sales in 2003: 871 Mio EUR
- Capital Investments 164 Mio EUR
- 6158 Employees worldwide
- 14 Sales offices
- 8 Production Fabs in 5 Locations around the World
- Products: Chlorosilanes; polycrystalline silicon, monocrystalline ingots; as cut, lapped, etched, polished, annealed or epitaxial wafers from 100mm to 300mm diameter

Single Crystal Growth

Czochralski single crystal growth (CZ)

- Named after J. Czochralski (1918)
- The CZ Process was modified by Teal and Little (1950)
  - Using a seed to define the crystal orientation
  - Diameter-control by the heating power and pulling rate
  - Control of the doping variation by the crystal rotation and pulling velocity
- Since 1970 the CZ method is the common method for IC applications
- Diameters up to 300 mm in standard production

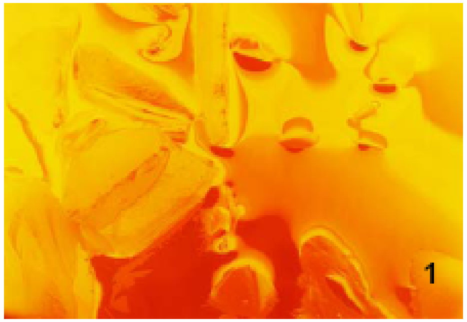
CO at Work



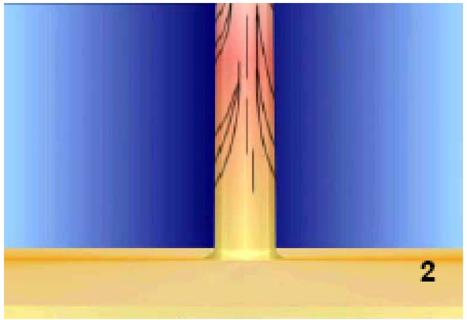
Martin Grötschel

Czochralski Growth

Quartz crucible filled with polycrystalline silicon

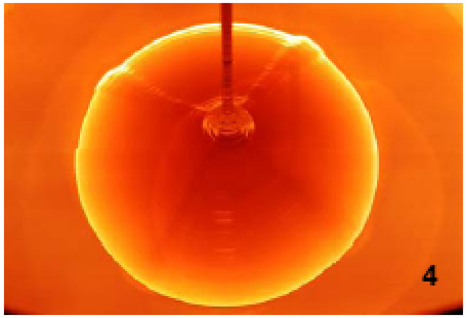


Heat up to liquid silicon



Crystal neck pulling

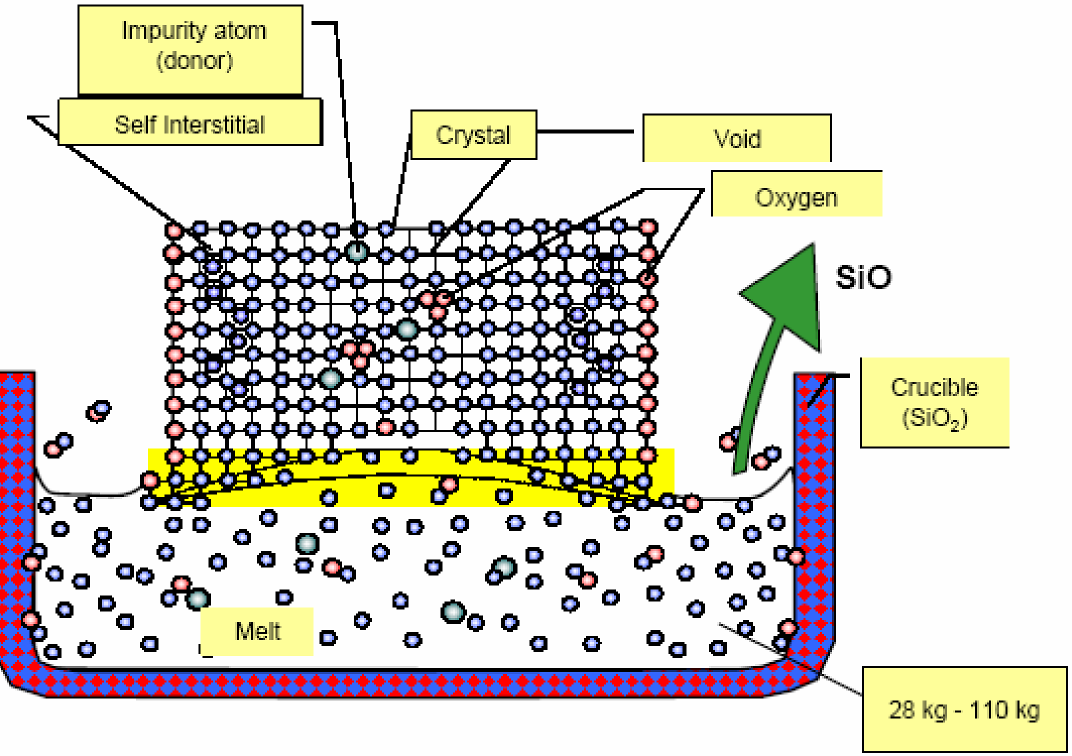
Increasing diameter



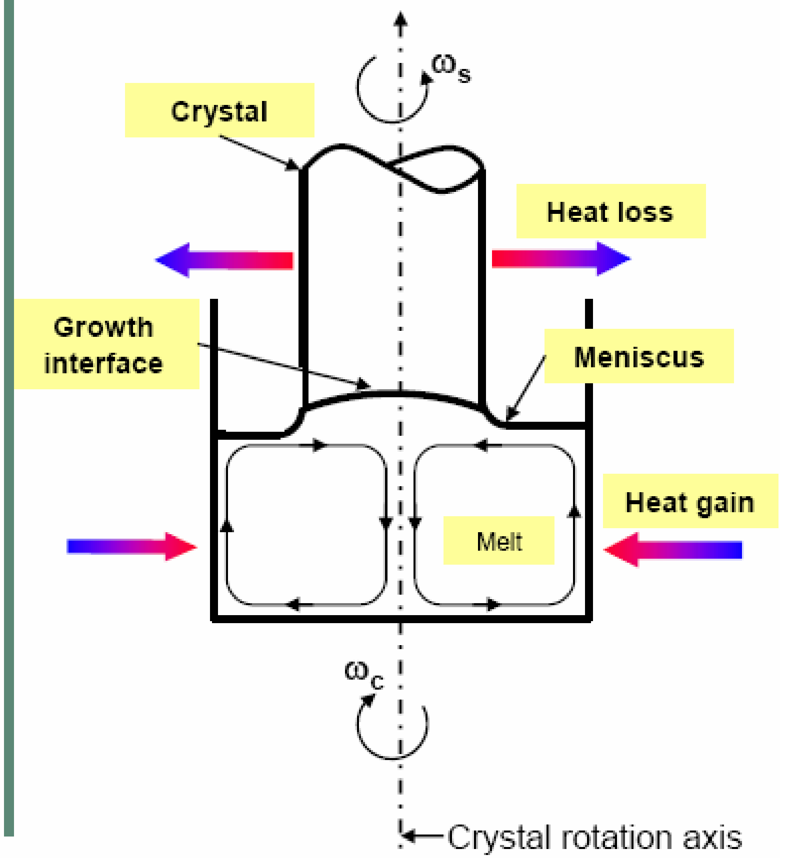
Crystal pulling

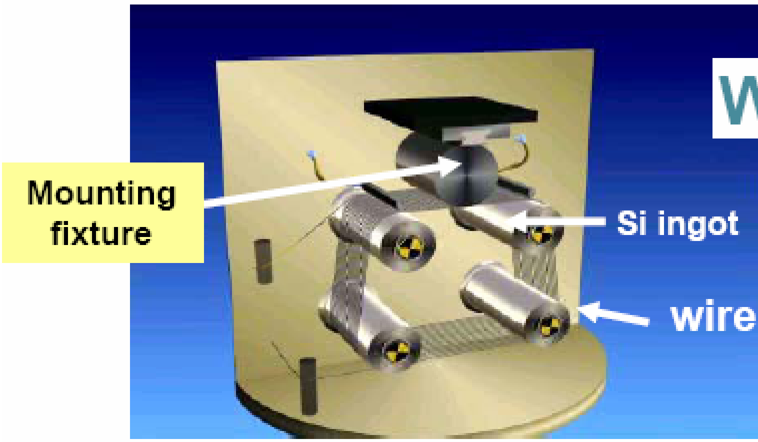
Single crystal silicon after CZ growth

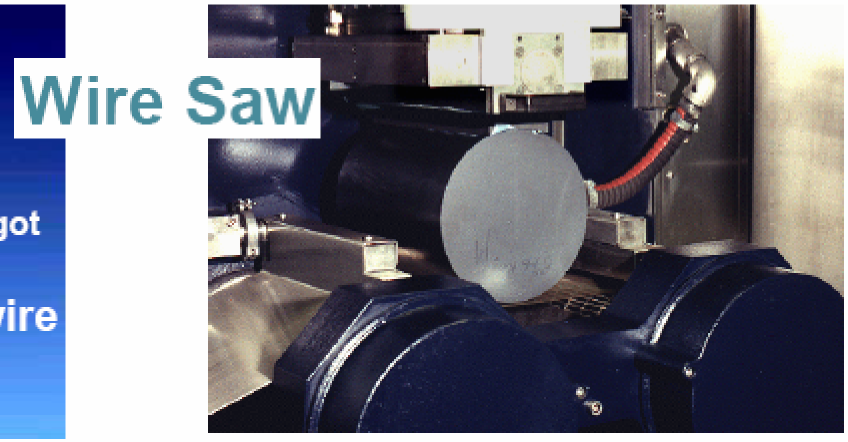
Czochralski Growth: schematic view

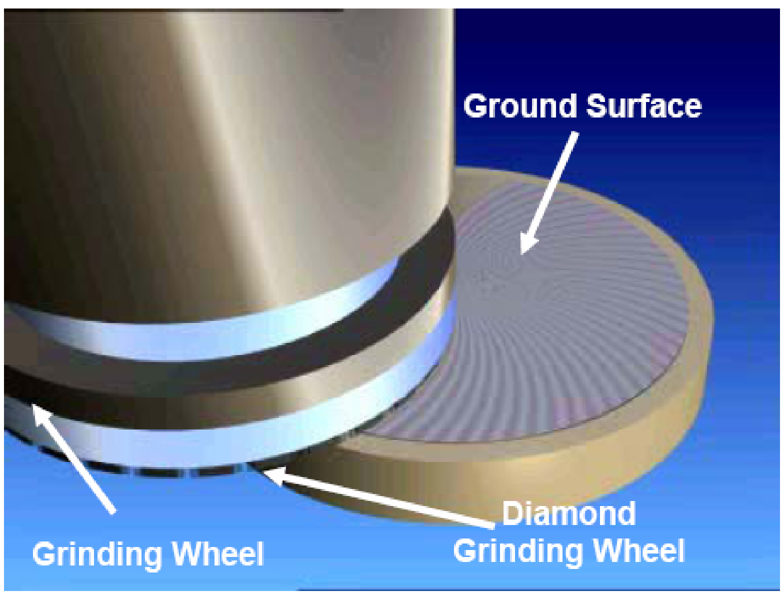


Schematic view of the growth process

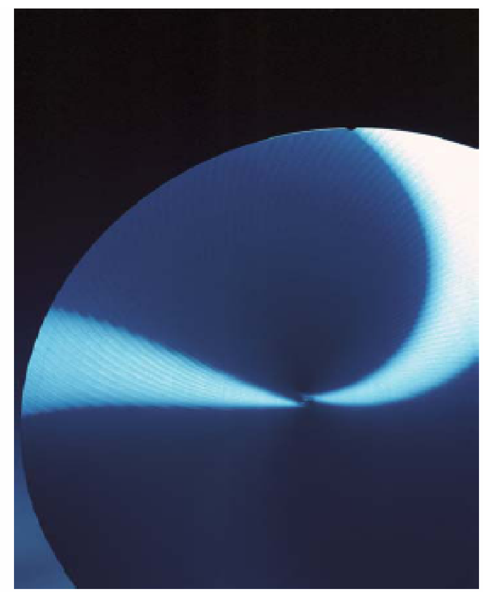








Ground in principle

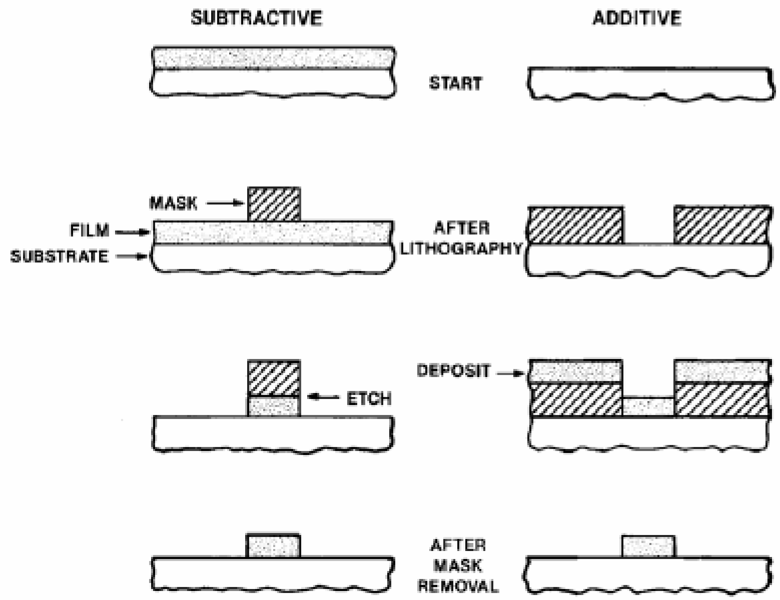


Ground silicon wafer

Martin Grötschel

IC Fabrication

Basic process sequence at IC fabrication

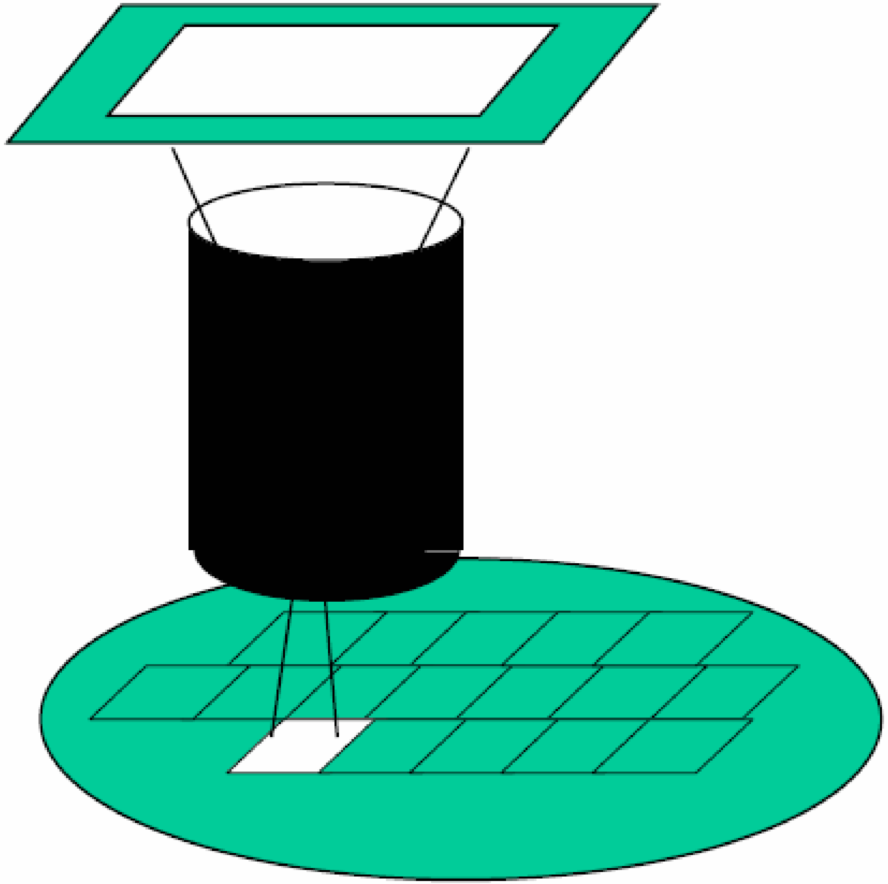


Subtractive	Additive
Deposition	
Photo-lithography	Photo-lithography
Etch	Deposition
Resist - Strip	„Lift Off"

Technique mostly used today is: **Subtractive processing** or:.......

Lithography

Stepper Principal



Principal:

The whole Mask is exposed onto the Wafer at a time

Then the Wafer steps to the next Position and exposes the Mask at the new Position

Martin Grötschel

Lithography

Mask drawing (Siemens uni2)

uni2

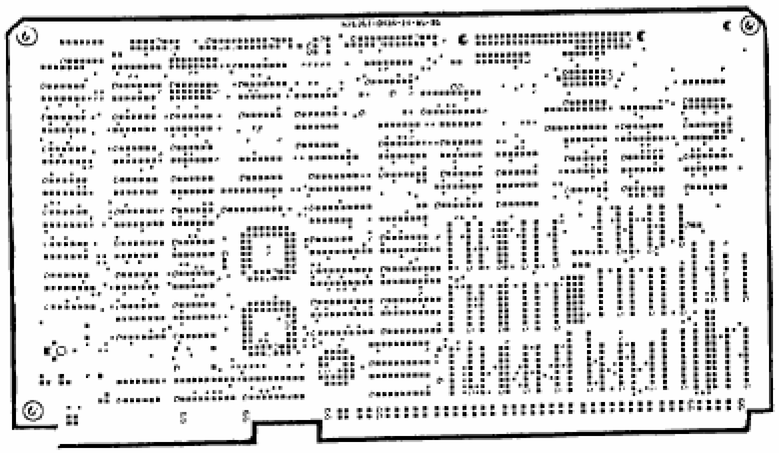
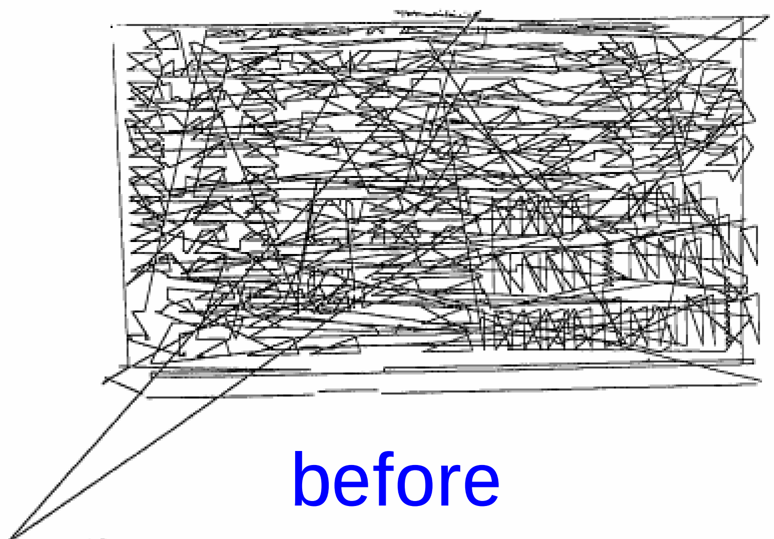


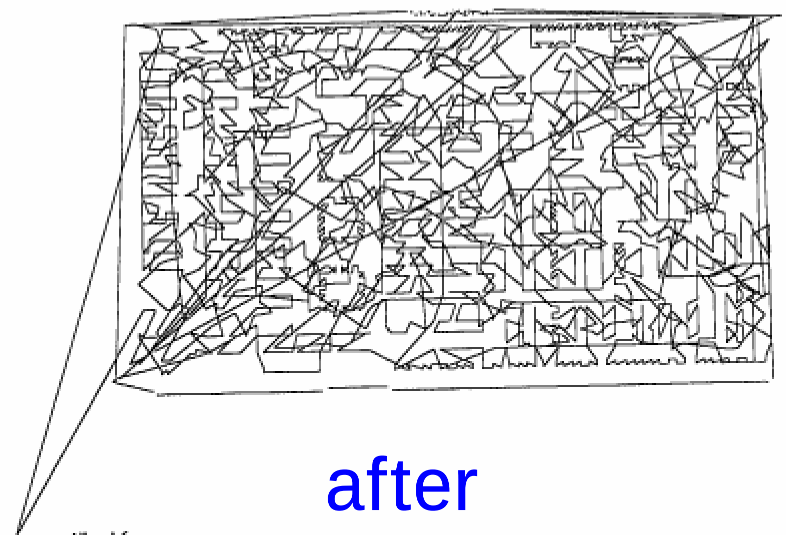
Fig. A2.

uni2



before

Fig. A5



after

Fig. A6

Martin Grötschel

Mask drawing (Siemens uni1)

uni1

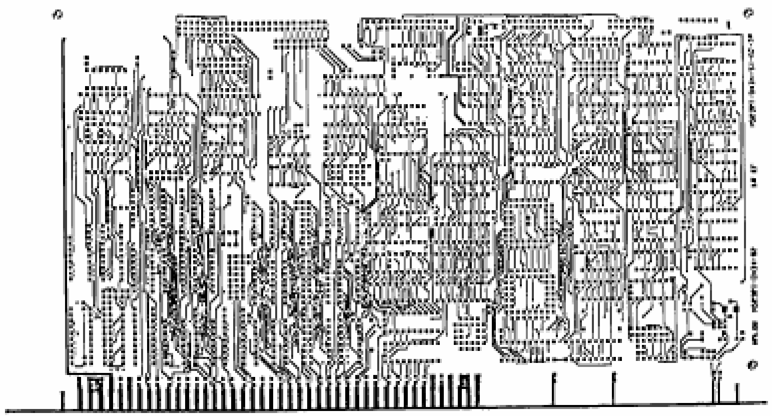


Fig. A1.

uni1

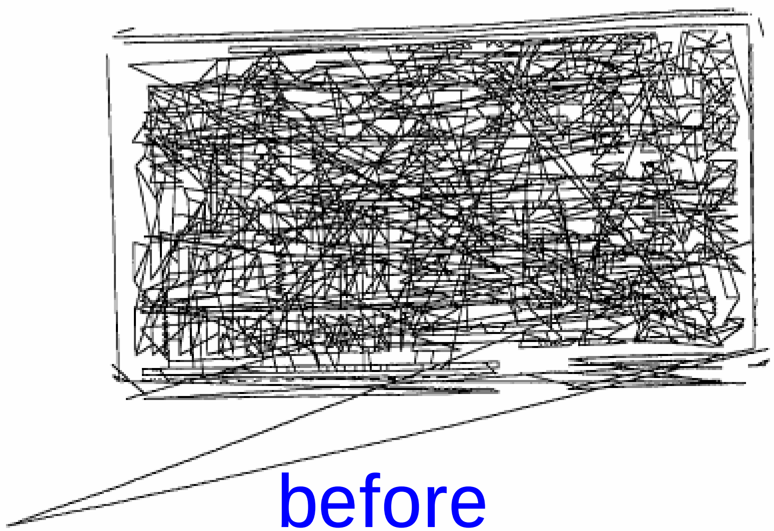


Fig. A3.

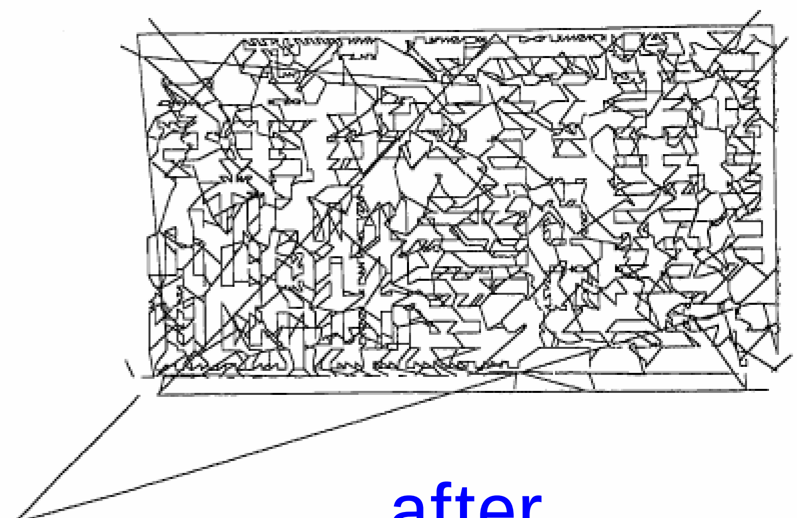


Fig. A4.

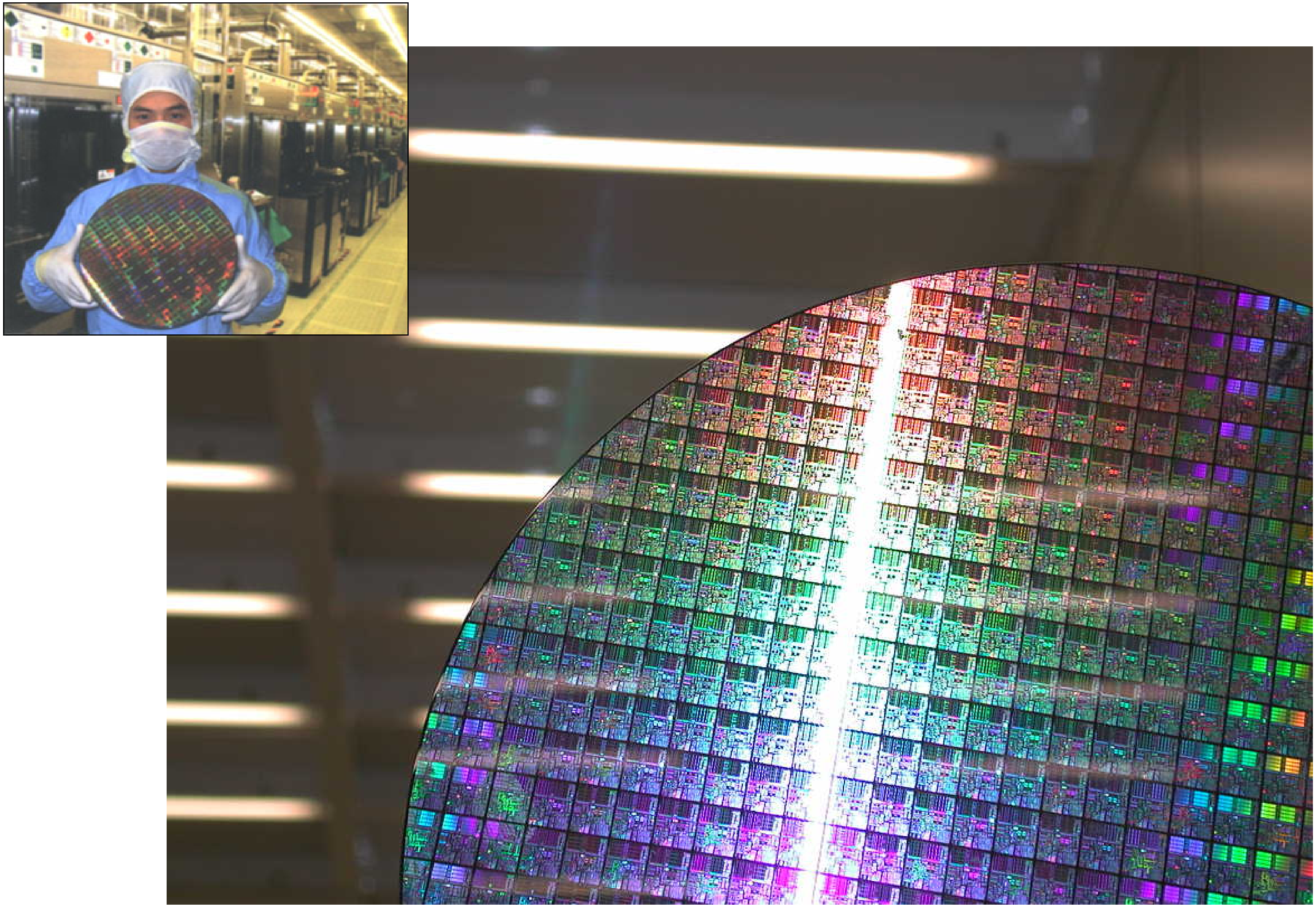
Typical Problems at Siemens

	uni1	uni2	uni3	uni5
Number of lines	6139	869	1360	28621
Number of points	2157	2496	1477	1060
Number of apertures	7	9	5	5

Fast Heuristics

	uni1	uni2	uni3	uni5
CPU time (min:sec)	4:33	2:36	1:37	1:19
Improvement in %	57.05	38.19	14.19	83.24

Chips on a wafer



Martin Grötschel

Contributions of Mathematics

Chip design and production without mathematics:

* reduced efficiency
* smaller capacity
* lower production quality
* lower speed
* much higher cost

Contents

Some Chip (Layout) Technologies

- Semi-Custom versus Full-Custom Layout

- Standard cells
- Gate arrays
- Sea of gates
- General cells

- Floorplanning

Martin Grötschel

(i) Entwurf für Allgemeine Zellen

Bei diesem Entwurfsstil können die Zellen beliebig auf dem Master angeordnet werden. Auch die Ausmaße der Zellen sind keinerlei Einschränkungen unterworfen. Ziel ist es, die Zellen so anzuordnen und die Netze so zu verdrahten, daß die resultierende Fläche möglichst klein ist.

(ii) Standard-Zellen Entwurf

Hier ist der Master unterteilt in einen Plazierungs- und Verdrahtungsbereich. Der Plazierungsbereich ist in Reihen gleicher Höhe eingeteilt. Die Zellen haben alle eben diese Höhe, können jedoch in der Breite differieren. Die Zellen müssen nun auf die Reihen so verteilt werden, daß zum Beispiel die längste Reihe möglichst kurz ist oder die Verdrahtungslänge minimiert wird. Die anschließende Verdrahtung erfolgt in den zwischen den Reihen liegenden Kanälen.

(iii) Gate-Array Entwurf

Im Gegensatz zu den beiden ersten Entwurfsstilen ist hier die Größe des Masters fest vorgegeben, jedoch erfolgt wieder eine Einteilung in Plazierungs- und Verdrahtungsbereich. Der Plazierungsbereich ist matrixförmig unterteilt in sogenannte *Basiszellen*. Die Breite (Höhe) der zu plazierenden Zellen ist ein Vielfaches der Breite (Höhe) dieser Basiszellen. Die Verdrahtung erfolgt in dem a-priori festgelegten Verdrahtungsbereich.

(iv) Sea-of-cells Entwurf sea-of-gates

Dieser Stil unterscheidet sich von letzterem nur dadurch, daß keine Einteilung in Plazierungs- und Verdrahtungsbereich erfolgt. Das bedeutet, daß der gesamte Master matrixförmig in Basiszellen unterteilt ist. Das Verdrahtungsgebiet bilden diejenigen Basiszellen, die nicht von den plazierten Zellen überdeckt sind.

Martin Grötschel

Min-Cut Placement (Heuristic)

- By pictures

Quadratic 0/1-Optimization

Weismantel,Robert:

Plazieren von Zellen: Theorie und Lösung eines quadratischen 0/1 Optimierungsproblems, 1992 (awarded with the Carl-Ramsauer-Preis of the AEG-Aktiengesellschaft), PhD Thesis at TU Berlin

3.1 Das Modell (P)

Eine Instanz des Plazierungsproblems für den "Sea of cells"-Entwurfsstil ist charakterisiert durch folgende Eingabedaten:

die Anzahl von Zellen;
Breite und Höhe jeder Zelle;
die Zulässigkeitsmenge jeder Zelle;
die Anzahl von Netzen und die Netzliste;
Breite und Höhe des Masters;

Variablen

Zur Modellierung des Problems führen wir für $i = 1, \ldots, n$ und für $k \in Z(i)$ Variablen ein. Diese lassen sich wie folgt interpretieren.

$$x_{ik} = \begin{cases} 1, & \text{wenn die linke untere Ecke der Zelle } i \text{ der} \\ & \text{Basiszelle } k \text{ zugewiesen wird.} \\ 0, & \text{sonst.} \end{cases}$$

Nebenbedingungen

(1) Jede Zelle muß genau einmal plaziert werden. Diese Nebenbedingung kann formal folgendermaßen beschrieben werden:

$$\sum_{k \in Z(i)} x_{ik} = 1 \text{ für alle } i = 1, \ldots, n.$$

(2) Alle Variablen müssen entweder den Wert 0 oder den Wert 1 annehmen:

$$x_{ik} \in \{0, 1\} \text{ für alle } i = 1, \ldots, n, \; k \in Z(i).$$

An dieser Stelle verzichten wir auf eine Modellierung der Überlappungsfreiheit. Diese Nebenbedingung wird in der Zielfunktion berücksichtigt.

Zielfunktion

Die Zielfunktion besteht aus einer gewichteten Summe. Der erste Summand entspricht einer Abschätzung der Gesamtverdrahtungslänge. Der zweite Summand zählt die Anzahl von Basiszellen, welche von mehr als einer Zelle überdeckt werden.

Zur Abschätzung der Gesamtverdrahtungslänge gehen wir folgendermaßen vor:
Zunächst verwenden wir das in Kapitel 1 vorgestellte Modell des vollständigen Graphen in leicht abgeänderter Form, um die Länge eines Netzes abzuschätzen. Den Abstand zwischen zwei Pins des Netzes bestimmen wir nicht nach der L_1 Norm. Stattdessen berechnen wir die "shortest Manhattan"-Distanz zwischen den beiden Zellen der Pins. Die **"shortest Manhattan"-Distanz** zwischen zwei Zellen ist definiert als der minimal Abstand, gemessen in der L_1-Norm, zwischen allen Paaren von Punkten auf den Rändern der zwei Zellen. Wie in Kapitel 1 dargestellt, werden die Zellabstände mit dem Faktor $\frac{1}{\alpha_t - 1}$ gewichtet, damit Netze der Kardinalität größer zwei nicht allzusehr überbewertet werden (die Zahl α_t wurde zu Anfang des Kapitels eingeführt und bezeichnet die Kardinalität des Netzes t).

Ferner führen wir **Affinitätskoeffizienten** ein. Diese sind zwischen allen Paaren von Zellen i und j, $i \neq j$ definiert und bestimmen sich folgendermaßen:

$$c_{ij} = \begin{cases} 0, & \text{wenn kein Netz existiert, welches Zellen } i \text{ und } j \text{ beide enthält.} \\ \sum_{t=1 \mid i \in A_t, j \in A_t}^{z} \frac{1}{\alpha_t - 1}, & \text{sonst.} \end{cases}$$

Gegeben sei eine zulässige Lösung des Plazierungsproblems und es bezeichne k_i die Basiszelle, welcher eine logische Zelle oder Randzelle i zugewiesen wurde. Ferner kürzen wir den "shortest Manhattan"-Abstand, wenn Zelle i auf Platz k und Zelle j auf Platz l ist, mit $d(i,k,j,l)$ ab.
Dann entspricht der Ausdruck

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k \in Z(i)} \sum_{l \in Z(j)} c_{ij} d(i,k,j,l) x_{ik} x_{jl} + \sum_{i=1}^{n} \sum_{k \in Z(i)} \sum_{r \in R} c_{ir} d(i,k,r,e(r)) x_{ik}$$

der nach dem Modell des vollständigen Graphen abgeschätzten Gesamtverdrahtungslänge.

The Quadratic 0/1-Minimization Model

Das vollständige Modell läßt sich damit folgendermaßen angeben:

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k \in Z(i)} \sum_{l \in Z(j)} c_{ij} d(i,k,j,l) x_{ik} x_{jl}$$

$$+ \quad \sum_{i=1}^{n} \sum_{k \in Z(i)} \sum_{r \in R} c_{ir} d(i,k,r,e(r)) x_{ik}$$

$$+ \quad \lambda_o \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k \in Z(i)} \sum_{l \in Z(j)} o(i,k,j,l) x_{ik} x_{jl},$$

so daß $\sum_{k \in Z(i)} x_{ik} = 1$, für alle $i = 1, \ldots n$.

$x_{ik} \in \{0,1\}$, für alle $i = 1, \ldots n$, $k \in Z(i)$.

Das **Floorplaning-Problem** stellt eine Verallgemeinerung des Plazierungsproblems dar. Insbesondere steht für die Zellen mehr als eine Realisierung zur Verfügung. Desweiteren ist beim Floorplaning-Problem der Master nicht fest vorgegeben. Als weiteres Optimalitätskriterium bleibt deshalb die für den Entwurf benötigte Fläche zu berücksichtigen. In Kapitel 1 dieser Arbeit haben wir bereits darauf hingewiesen, daß der Packungsaspekt (Fläche der Plazierung) bei anderen Entwurfsstilen ebenfalls von Bedeutung ist. Wir wollen daher in diesem Unterpunkt zeigen, wie unser Modell modifiziert werden muß, um die von der Plazierung benötigten Fläche in unseren Ansatz zu integrieren.

Martin Grötschel

Floorplaning Model

Bezeichne $\lambda_F \in \mathbb{R}$ den Bestrafungsparameter für die Fläche, dann stellt sich eine Modellierung des Floorplaning-Problems so dar:

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k\in Z(i)} \sum_{l\in Z(j)} \sum_{a\in A(i)} \sum_{c\in A(j)} c_{ij} d(i,k,a,j,l,c) x_{ik}^a x_{jl}^c \quad +$$

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k\in Z(i)} \sum_{l\in Z(j)} \sum_{a\in A(i)} \sum_{c\in A(j)} \lambda_o o'(i,k,a,j,l,c) x_{ik}^a x_{jl}^c$$

$$+ \sum_{i=1}^{n} \sum_{k\in Z(i)} \sum_{a\in A(i)} \sum_{r\in R} c_{ij} x_{ik}^a d(i,k,a,r,e(r),a(r))$$

$$+ \lambda_F \sum_{u=1}^{h_{max}-h_{min}+1} \sum_{s=1}^{w_{max}-w_{min}+1} (s+w_{min}-1)(u+h_{min}-1) v^s h^u$$

$$+ \sum_{i=1}^{n-1} \sum_{k\in Z(i)} \sum_{a\in A(i)} \sum_{u=1}^{h_{max}-h_{min}+1} g^h(u,i,k,a) x_{ik}^a h^u$$

$$+ \sum_{i=1}^{n-1} \sum_{k\in Z(i)} \sum_{a\in A(i)} \sum_{s=1}^{w_{max}-w_{min}+1} g^v(s,i,k,a) x_{ik}^a v^s.$$

so daß $\sum_{k\in Z(i)} \sum_{a\in A(i)} x_{ik}^a = 1$, für alle $i = 1,\dots n$.

$$\sum_{s=1}^{w_{max}-w_{min}+1} v^s = 1, \qquad \sum_{u=1}^{h_{max}-h_{min}+1} h^u = 1$$

$x_{ik}^a \in \{0,1\}$, für alle $i = 1,\dots n$, $k \in Z(i)$, $a \in A(i)$,

$v^s \in \{0,1\}$, $h^u \in \{0,1\}$.

Results

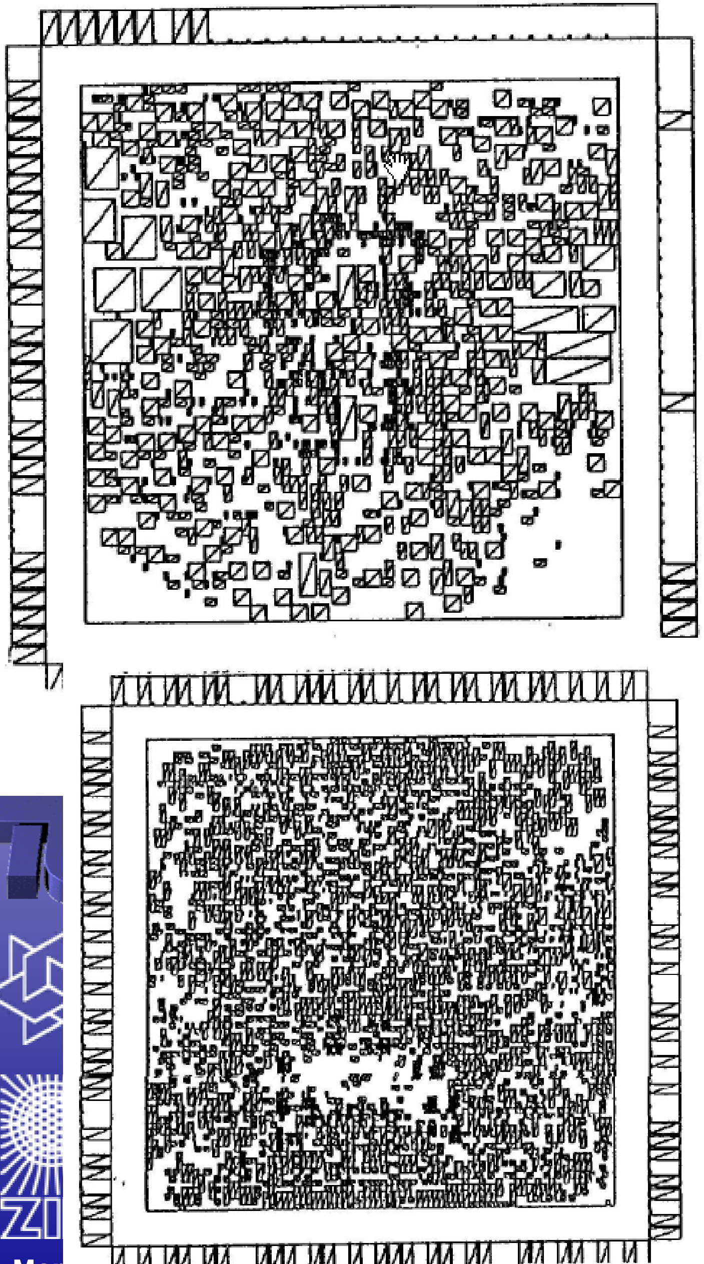


Abbildung 8.8

soc_1	gesch. Verdrahtungsl.	nicht verbunden	CPU
Min-Cut	212258	-	-
Gordian	184045	9	-
Quazo	176851	14	12:12

Tabelle 8.1

soc_2	gesch. Verdrahtungsl.	nicht verbunden	CPU
Min-Cut	194732	24	4:25
Gordian	189683	19	3:43
Quazo	185592	14	28:47

Tabelle 8.2

soc_3	gesch. Verdrahtungsl.	nicht verbunden	CPU
Min-Cut	796622	33	15:54
Gordian	652129	51	3:13
Quazo	553575	48	129:39

Tabelle 8.3

soc_4	gesch. Verdrahtungsl.	nicht verbunden	CPU
Min-Cut	623159	551	21:04
Gordian	506160	253	6:52
Quazo	497285	260	159:23

Tabelle 8.4

Contents

Martin Grötschel

Approaches to Routing

Martin, Alexander:

Packen von Steinerbäumen: Polyedrische Studien und Anwendung, 1992

Koch, Thorsten:

Rapid Mathematical Programming, 2004

PhD Theses at TU Berlin

Steiner Trees and Steiner Tree Packing

The (weighted) Steiner Tree Problem:

Given a graph $G=(V,E)$ with edge weights $c(e)$, $e \in E$, and a subset T of V, called terminal nodes. Find a tree in G spanning T of minimum weight.

The Steiner Tree Packing Problem:

Given a graph $G=(V,E)$ with edge weights $c(e)$, $e \in E$, and N subsets $T_1,\dots,T_N$ of V, called nets. Find trees $S_1,\dots,S_N$ in G spanning the terminal nodes $T_1,\dots,T_N$, respectively, of total minimum weight. The trees $S_1,\dots,S_N$ have to satisfy, in addition, (application specific) disjointness/intersection conditions.

(The graph G usually has special properties.)

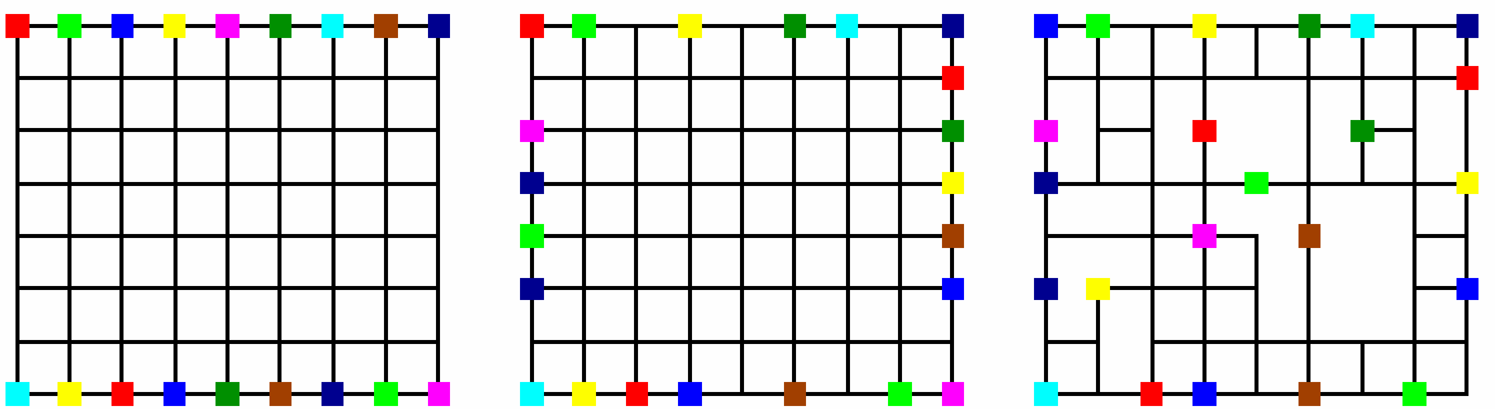
Martin Grötschel

Examples of Routing Problems

Channel routing (10.1a) Here, we are given a complete rectangular grid graph. The terminals of the nets are exclusively located on the lower and upper border. It is possible to vary the height of the channel. Hence, the size of the routing area is not fixed in advance. Usually all nets have only two terminals, i. e., $|T_i| = 2$.

Switchbox routing (10.1b) Again, we are given a complete rectangular grid graph. The terminals may be located on all four sides of the graph. Thus, the size of the routing area is fixed.

General routing (10.1c) In this case, an arbitrary grid graph is considered. The terminals can be located arbitrarily (usually at some hole of the grid).



(a) Channel routing (b) Switchbox routing (c) Global routing

Martin Grötschel

Intersection / Disjointness conditions

Intersection model

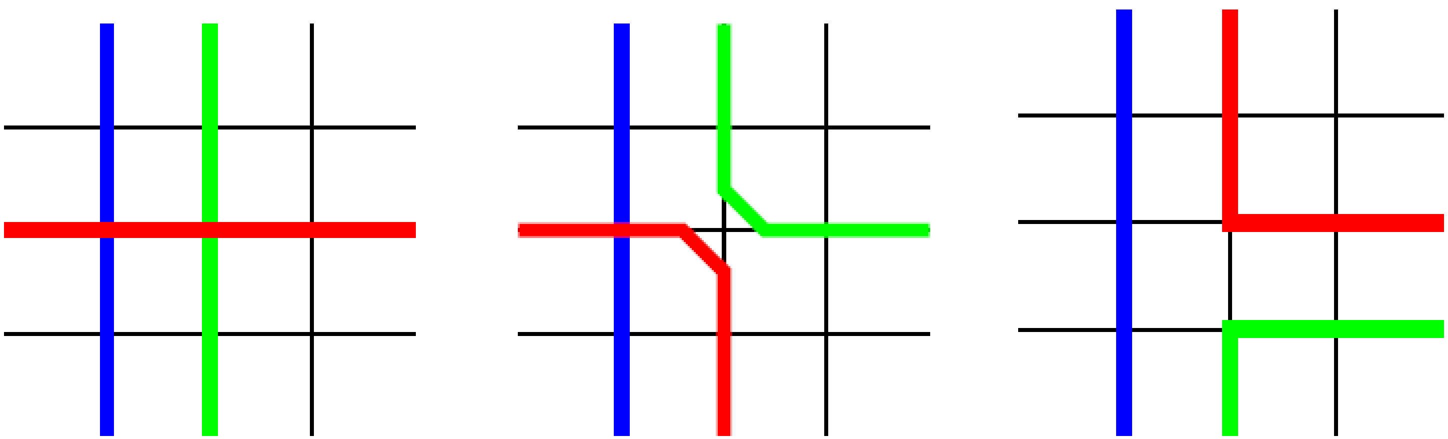
The intersection of the nets is an important point in Steiner tree packing. Again three different models are possible.

Manhattan (10.2a) Given same (planar) grid graph. The nets must be routed in an edge disjoint fashion with the additional restriction that nets that meet at some node are not allowed to bend at this node, i. e., so-called *Knock-knees* are not allowed. This restriction guarantees that the resulting routing can be laid out on two layers at the possible expense of causing long detours.

Knock-knee (10.2b) Again, some (planar) grid graph is given and the task is to find an edge disjoint routing if the nets. In this model Knock-knees are possible. Very frequently, the wiring length of a solution in this case is smaller than in the Manhattan model. The main drawback is that the assignment to layers is neglected.

Brady and Brown (1984) have designed an algorithm that guarantees that any solution in this model can be routed on four layers. It was shown by Lipski (1984) that it is $\mathcal{NP}$-complete to decide whether a realization on three layers is possible.

Node-disjoint (10.2c) The nets have to be routed in a node disjoint fashion. Since no crossing of nets is possible in a planar grid graph, this requires a multi-layer model.



(a) Manhattan model (b) Knock-knee model (c) Node disjoint model

Martin Grötschel

Multiple Layers

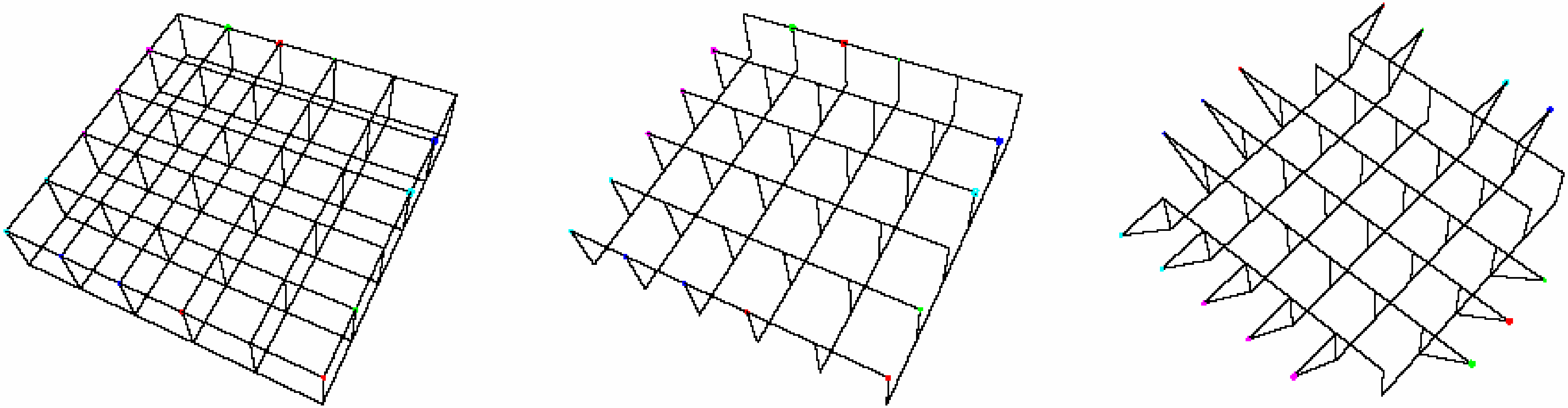
Multiple layers

While channel routing usually involves only a single layer, switchbox and general routing problems are typically multi-layer problems. Using the Manhattan and Knock-knee intersection is a way to reduce the problems to a single-layer model. Accordingly, the multi-layer models typically use node disjoint intersection. While the multi-layer model is well suited to reflect reality, the resulting graphs are in general quite large. We consider two[2] possibilities to model multiple layers:

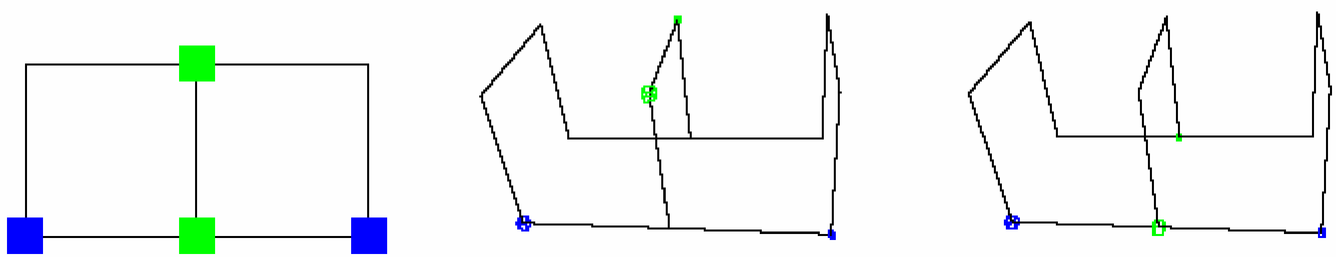
k-**crossed layers** (6.3a) There is given a k-dimensional grid graph (that is a graph obtained by stacking k copies of a grid graph on top of each other and connecting corresponding nodes by perpendicular lines, so-called *vias*), where k denotes the number of layers. This is called the k-layer model in Lengauer (1990).

k-**aligned layers** (6.3b) This model is similar to the crossed-layer model, but in each layer there are only connections in one direction, either east-to-west or north-to-south. Lengauer (1990) calls this the *directional* multi-layer model. Korte et al. (1990) indicate that for $k = 2$ this model resembles the technology used in VLSI-wiring best. Boit (2004) mentions that current technology can use a much higher number of layers.

Martin Grötschel



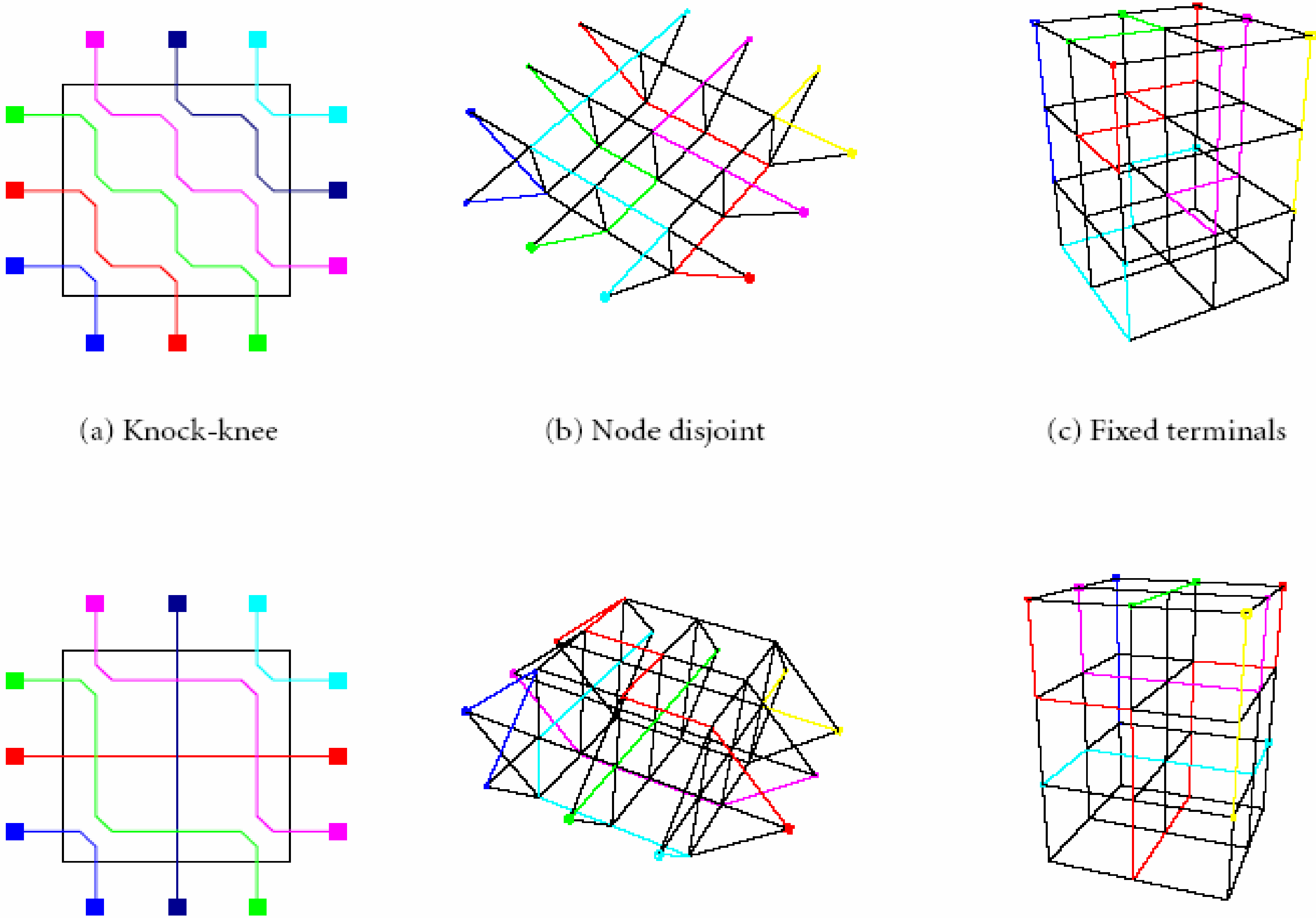
(a) Multi-crossed layers (b) Multi-aligned layers (c) With connectors



(a) Manhattan one-layer (b) Feasible (c) Infeasible

Figure 6.4: Manhattan one-layer vs. Node disjoint two-aligned-layer

Martin Grötschel



(a) Knock-knee

(b) Node disjoint

(c) Fixed terminals

(d) Knock-knee

(e) Node disjoint

(f) Fixed terminals

Figure 6.5: Number of layers needed to route a Knock-knee one-layer solution

Martin Grötschel

6.2.1 Undirected partitioning formulation

This formulation is used in Grötschel et al. (1997). Given a weighted grid graph $G = (V, E, c)$, and terminal sets $T_1, \dots, T_N$, $N > 0$, $\mathcal{N} = \{1, \dots, N\}$, we introduce binary variables x^n_{ij} for all $n \in \mathcal{N}$ and $(i,j) \in E$, where $x^n_{ij} = 1$ if and only if edge $(i,j) \in S_n$. We define $\delta(W) = \{(i,j) \in E | (i \in W, j \notin W) \vee (i \notin W, j \in W)\}$ with $W \subseteq V$.

The following formulation models all routing choices for the Knock-knee one-layer model:

$$\min \quad \sum_{n \in \mathcal{N}} \sum_{(i,j) \in E} c_{ij} x^n_{ij}$$

$$\sum_{(i,j) \in \delta(W)} x^n_{ij} \geqslant 1 \qquad \text{for all } W \subset V, W \cap T_n \neq \emptyset, (V \setminus W) \cap T_n \neq \emptyset, n \in \mathcal{N} \tag{6.1}$$

$$\sum_{n \in \mathcal{N}} x^n_{ij} \leqslant 1 \qquad \text{for all } (i,j) \in E \tag{6.2}$$

$$x^n_{ij} \in \{0, 1\} \qquad \text{for all } n \in \mathcal{N}, (i,j) \in E \tag{6.3}$$

In order to use Manhattan intersection another constraint is needed to prohibit Knock-knees. Let (i,j), (j,k) be two consecutive horizontal (or vertical) edges. Then,

$$\sum_{n \in N_1} x^n_{ij} + \sum_{m \in N_2} x^m_{jk} \leqslant 1 \text{ for all } j \in V, N_1 \subset \mathcal{N}, N_2 \subset \mathcal{N}, N_1 \cap N_2 = \emptyset, N_1 \cup N_2 = \mathcal{N} \tag{6.4}$$

is called *Manhattan inequality*.[3] The model can be further strengthened with several valid inequalities as described in Grötschel et al. (1996a,b), Grötschel et al. (1997).

$$\min \sum_{n\in\mathcal{N}} \sum_{(i,j)\in A} c_{ij}^n \bar{x}_{ij}^n$$

$$\sum_{(i,j)\in\delta_j^-} y_{ij}^t - \sum_{(j,k)\in\delta_j^+} y_{jk}^t = \left\{ \begin{array}{rl} 1 & \text{if } j = t \\ -1 & \text{if } j = r_{\sigma(t)} \\ 0 & \text{otherwise} \end{array} \right\} \quad \text{for all } j \in V, t \in T \setminus R \qquad (6.5)$$

$$0 \leqslant y_{ij}^t \leqslant \bar{x}_{ij}^{\sigma(t)} \quad \text{for all } (i,j) \in A, t \in T \setminus R \qquad (6.6)$$

$$\sum_{n\in\mathcal{N}} (\bar{x}_{ij}^n + \bar{x}_{ji}^n) \leqslant 1 \quad \text{for all } (i,j) \in A \qquad (6.7)$$

$$\bar{x}_{ij}^n \in \{0,1\} \quad \text{for all } n \in \mathcal{N}, (i,j) \in A \qquad (6.8)$$

To use node disjoint intersection we have to add:

$$\sum_{n\in\mathcal{N}} \sum_{(i,j)\in\delta_j^-} \bar{x}_{ij}^n \leqslant \left\{ \begin{array}{ll} 0 & \text{if } j \in R \\ 1 & \text{otherwise} \end{array} \right. \quad \text{for all } j \in V \qquad (6.9)$$

It is possible to strengthen (6.11) by subtracting the incoming arc anti-parallel to the outgoing arc in question, giving the following valid inequality:

$$\bar{x}_{jk}^n + \bar{x}_{kj}^n \leqslant \sum_{(i,j)\in\delta_j^-} \bar{x}_{ij}^n \quad \text{for all } j \in V \setminus R, (j,k) \in \delta_j^+, n \in \mathcal{N} \;, \qquad (6.12)$$

Martin Grötschel

Corollary 1. *The* LP *relaxation of the multicommodity flow formulation of the node disjoint two-aligned-layer model is strictly stronger than the* LP *relaxation of the partitioning formulation of the Manhattan one-layer model.*

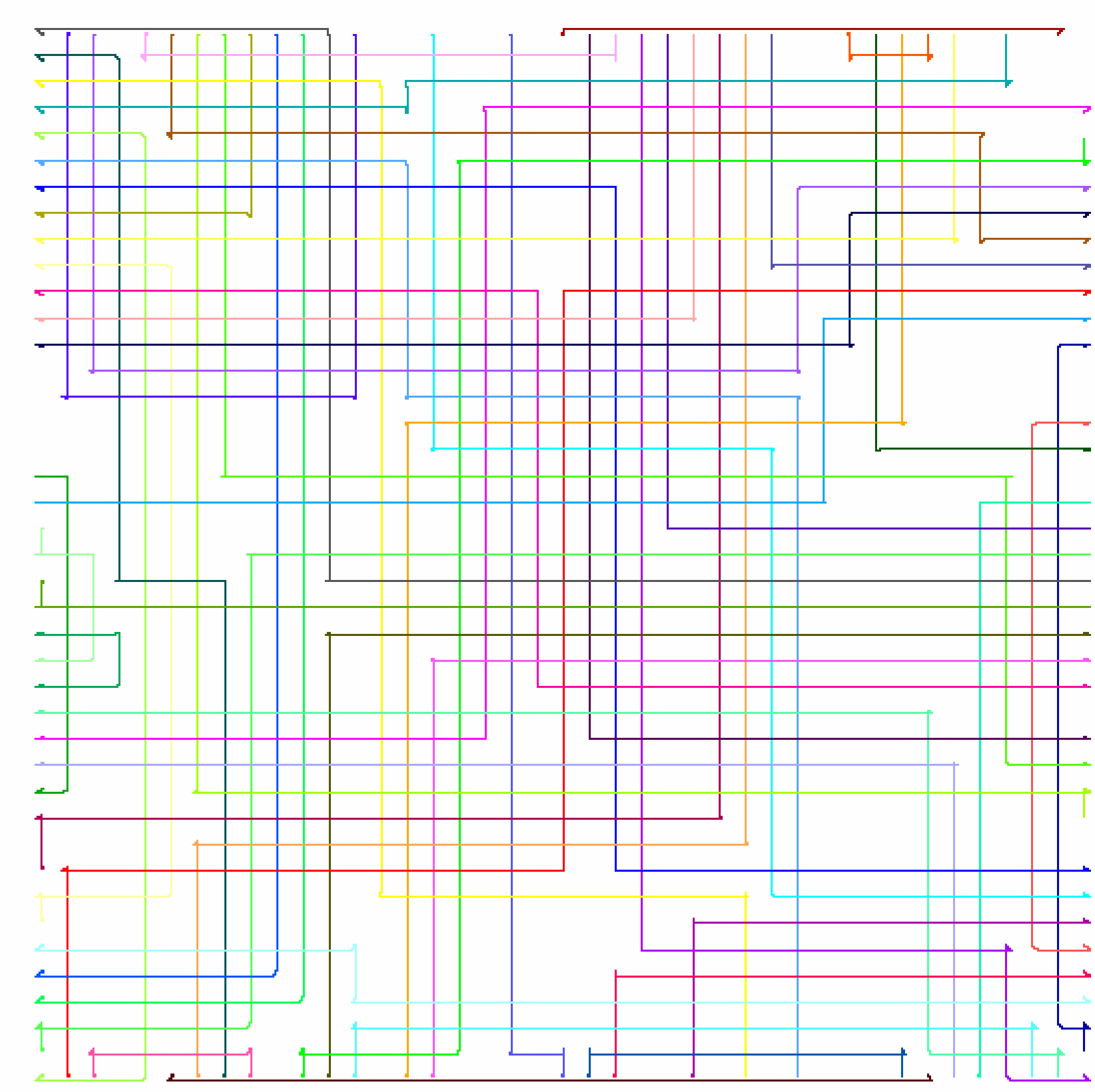
Performance

Grötschel et al. (1997) report solution times for the Manhattan one-layer model on a SUN IPX 4/50 with 40 megahertz. Of course, any comparison of CPU times between different processors is highly inaccurate and debatable. Nevertheless, we will make some educated guesses. The results for the node disjoint two-aligned-layer model in Table 6.5 were computed on a 3,200 megahertz computer. This gives us a factor of 80. If we compare our best solution times with the ones reported, the geometric mean of the speed-up for all five solvable instances is 1,526. This is nearly twenty times faster than what we would have expected from the megahertz figure. Furthermore, this is the comparison between a special purpose code with preprocessing, separation routines and problem specific primal heuristics with a *generate the whole model and feed it into a- tandard solver* approach without any problem specific routines. We can conclude from the value of the root LP relaxation that the partitioning formulation with additional strengthening cuts and the directed multicommodity flow formulation are about equally strong in practice. It should be noted, though, that for *moredifficult-2*, the only instance where the flow formulation is weaker, we also have the least improvement by only a factor of 246, while for *pedabox-2*, the only instance where the flow formulation is stronger, we have the highest improvement by a factor of 22,416. The rest of the speed-up seems to come from CPLEX.[12] The numbers are compatible with those given in Bixby et al. (2000) and Bixby (2002), keeping in mind that the improvement in hardware speed of 80 times is only a gross approximation.

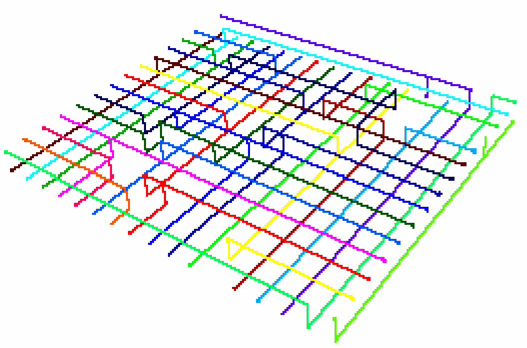
Name	CS	B&B Nodes	Time [s]	Via-cost	Vias	Arcs	Vias +Arcs
sb11-20-7	B	1	16,437	1	107	486	593
	E	1	65,393	1	107	486	593
sb3-30-26d	B	1	1,455	1	130	1286	1416
	E	1	47,335	1	130	1286	1416
sb40-56	B	1	3,846	1	166	2286	2452
	D	1	518	1	166	2286	2452
	E	3	776	1	166	2286	2452
taq-3	B	71	931	1	66	371	437
	D	4	385	1	66	371	437
	E	19	346	1	66	371	437
alue-4	B	124	18,355	1	117	668	785
	D	1	3,900	1	117	668	785
	E	4	1,825	1	117	668	785

Table 6.6: Results for the node disjoint multi-aligned-layer model (part 2)

Dissertation Thorsten Koch



optimal solution of routing a problem with simultaneous via-minimization



03M1 Lecture
Chip Design

The End

Martin Grötschel
▪ Institut für Mathematik, Technische Universität Berlin (TUB)
▪ DFG-Forschungszentrum "Mathematik für Schlüsseltechnologien" (MATHEON)
▪ Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)
groetschel@zib.de
http://www.zib.de/groetschel