

Algorithmen für Chip-Entwurfswerkzeuge

Floorplanning



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vorlesung
WS 2018/2019

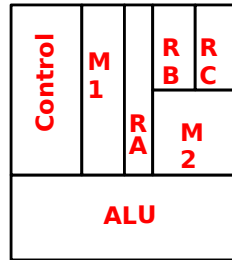
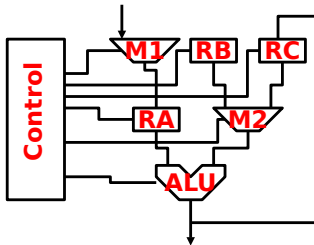
Florian Stock

Eingebette Systeme und Anwendungen
Technische Universität Darmstadt



- ▶ Auf unteren Entwurfsebenen
 - ▶ Ausreichend Details vorhanden:
Fläche, Verdrahtung
 - ▶ Layout leicht zu berücksichtigen
- ▶ Auf höheren Entwurfsebenen
 - ▶ Details fehlen
 - ▶ Frühe Abschätzungen erforderlich z.B. für
 - ▶ Fläche
 - ▶ Verdrahtungsmuster

- ▶ *Topologische* Anordnung
- ▶ **Floorplanning**: Bestimme optimale Form und Anordnung
- ▶ Vereinfachungen:
 - ▶ Relative Anordnungen statt absoluter Position
 - ▶ Abschätzungen z.B. für Fläche und Verdrahtungslänge



Floorplanning

Ziele



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Möglichst früh im Entwurf Abschätzungen
- ▶ Flächenminimierung
- ▶ Abstrakte Module

- Gegeben:**
- ▶ Liste von Module $B = \{b_1, \dots, b_m\}$
 - ▶ Jedes Module hat Höhe h_i , Breite w_i und Fläche a_i
- Gesucht:**
- ▶ (x_i, y_i) bezeichnet die linke untere Koordinate des Moduls b_i auf dem Chip
 - ▶ Ein **Floorplan** ist eine Bestimmung der (x_i, y_i) , so dass
 1. Die Module sich nicht überlappen
 2. Eine Kostenfunktion minimiert wird.
Üblicherweise
 - ▶ Fläche
 - ▶ Verdrahtungslänge
 - ▶ Aber auch andere (Verdrahtbarkeit, Hitze, ...)
 - ▶ Die Module dürfen dabei auch rotiert werden!



- ▶ Floorplanning ist im allgemeinen Fall NP-schwer
- ▶ Benutzte Lösungsalgorithmen:
 - ▶ Simulated Annealing
Mit kombinierter HPWL- und Flächenkostenfunktion
 - ▶ Mixed Integer Linear Programming

Fertig?



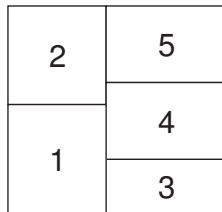
- ▶ Wie sieht der Lösungsraum aus?
 - ▶ Wie sieht die Nachbarschaft aus?
- ⇒ Wie modelliert man den Floorplan?



- ▶ Darstellung eines Floorplans soll
 - ▶ Schnell aufbaubar sein
 - ▶ Leicht zu manipulieren sein (SA-Tausch)
 - ▶ Den Suchraum einschränken
(\Rightarrow Lösungen gehen verloren!)
- ▶ 4 Arten von Floorplans haben sich herausgebildet
 - ▶ Geschnittene (Slicing) Floorplans
 - ▶ Kompakte (Compact) Floorplans
 - ▶ Mosaik (Mosaic) Floorplans
 - ▶ Allgemeine (General) Floorplans

Geschnittene (Slicing) Floorplans

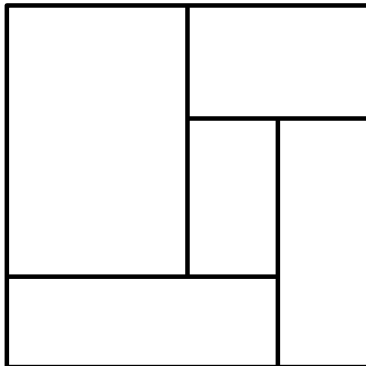
- ▶ Wiederholt werden Gebiete horizontal (H) oder vertikal (V) geschnitten
⇒ Nicht alle Floorplans können dargestellt werden
- ▶ Darstellbar als:
 - ▶ Schnittbaum (Slicing-Tree)
 - ▶ Normalisierten Umgekehrte Polnischen Ausdrücken (Normalized Reverse Polish Expression, NRPE)
- ▶ Nicht alle Floorplans sind Slicing Floorplans



Ungeschnittener (Non-Slicing) Floorplan

Beispiel

- ▶ Ab 5 Modulen möglich

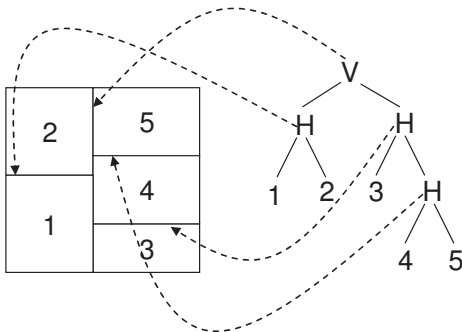


Floorplan Repräsentation

Schnittbaum (Slicing-Tree)

- ▶ Module sind die Blätter eines Baumes
- ▶ Innere Knoten sind entsprechend des teilenden Schnittes markiert (H oder V)
- ▶ Darstellung nicht eindeutig
⇒ Ein Floorplan kann durch verschiedene Bäume dargestellt werden
- ▶ Eindeutig: Skewed Slicing Trees
 - ▶ Schnittbäume, bei denen der Eltern- und rechter Kindknoten nicht gleich geschnitten sind
 - ▶ Kleinere Suchraum

Schnittbaum Beispiel



Quelle: Y.-W. Chang

Floorplan Repräsentation

Normalisierte Umgekehrte Polnische Ausrücke

- ▶ Wong, 1986
- ▶ Umgekehrte Polnische Ausrücke (Reverse Polish Expressions):
 - ▶ Generation: Postorder-Durchlauf des Schnittbaumes
 - ▶ Sequenz aus Modulnummern $1, \dots, n$ (Operanden) und den Symbolen V und H (Schnitt-Operatoren)
 - ▶ Länge $2n - 1$
 - ▶ Normalisierte PA benutzen Skewed Slicing Tree
⇒ Keine HH oder VV in Sequenz
- ▶ Umkehrung: Auswerten des Ausdrucks mittels Stack

Floorplan Repräsentation

NRPE – SA-Züge

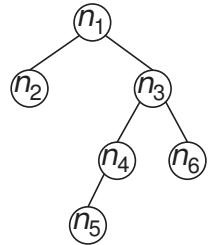
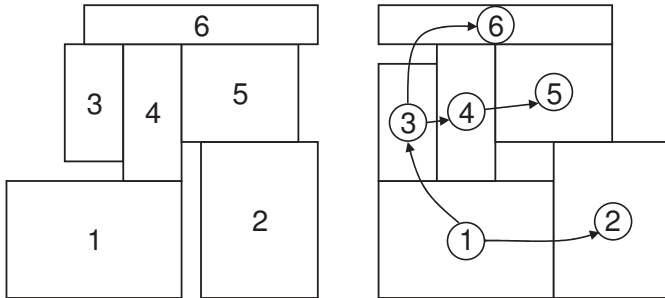
- ▶ Züge für SA bei Benutzung von NRPE als Darstellung:
 - ▶ Tausch von zwei nebeneinanderliegenden Operatoren (Immer legal (evtl. nicht mehr normalisiert))
 - ▶ In einer ununterbrochenen Folge von Operatoren, alle invertieren (Immer legal (evtl. nicht mehr normalisiert))
 - ▶ Einen nebeneinanderliegenden Operator und Operand tauschen (Nicht immer legal: In jedem Prefix der Sequenz muß gelten: $\#Operatoren < \#Operanden$)



- ▶ Kompakte Floorplan:
 - ▶ Chang, 2000
 - ▶ Keine Module können nach links oder unten verschoben werden
 - ▶ Geordnete binäre Bäume
 - ▶ 1-zu-1-Beziehung: Kompakter Floorplan und B*-Baum
⇒ Keine Doppelten im Suchraum
- ▶ SA-Operationen:
 - ▶ Ein Module im Baum rotieren
 - ▶ Ein Knoten im Baum bewegen (löschen, wiedereinfügen)
 - ▶ Zwei Knoten tauschen

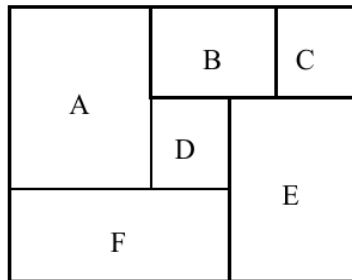
B*-Baum Konstruktion

- ▶ Wurzel-Knoten entspricht Modul unten links
- ▶ Linker Teilbaum entspricht Modulen rechts davon
- ▶ Rechter Teilbaum entspricht Modulen darüber



Quelle: Y.-W. Chang

- ▶ Ähnlich mächtig wie allgemeine
 - ▶ Lässt keine *Lücken* zu
 - ▶ Lässt nur T-Kreuzungen zu
- ▶ Datenstrukturen zur Darstellung
 - ▶ Corner Block List
 - ▶ Twin Binary Sequence





- ▶ Allgemeiner Floorplan
- ▶ Keine Einschränkungen
- ▶ Beliebteste Darstellung:
 - ▶ Sequenz Paar (Sequence Pair)
 - ▶ Murata, 1995

Allgemeine (General) Floorplans

Sequenz-Paar Darstellung

- ▶ Darstellung als Paar von Sequenzen $(\Gamma_+, \Gamma_-) = (m_{f_1} \dots m_{f_M}, m_{s_1} \dots m_{s_M})$
- ▶ Jedes Paar Sequenzen beschreibt einen gültigen Floorplan
- ▶ Einfache SA-Operationen
 - ▶ Ein Modul rotieren
 - ▶ Zwei Module in einer Sequenz tauschen
 - ▶ Zwei Module in beiden Sequenzen tauschen

Sequenz Paar Darstellung

Sequenz-Paar \rightarrow Floorplan



- ▶ Relation der Module zu einander:
 1. $(\dots i \dots j \dots, \dots i \dots j \dots)$
 \Rightarrow Modul i ist links von Modul j
 2. $(\dots j \dots i \dots, \dots i \dots j \dots)$
 \Rightarrow Modul i ist unter Modul j
- ▶ Anschaulich:
 - ▶ $n \times n$ -Gitter um 45° gedreht
 - ▶ Je eine Dimension für Γ_+ und Γ_-
 - ▶ Kreuzungspunkte des gleichen Moduls beschreiben seine Position
- ▶ Algorithmisch:
 - ▶ Horizontaler und vertikaler Einschränkungsgraph (azyklisch)
(Details zu Einschränkungsgraph in Block 7)
 - ▶ Modulkordinaten mittels längsten Pfad in $\mathcal{O}(n^2)$

Suchraumgröße (Asymptotisch)

Geschnittene

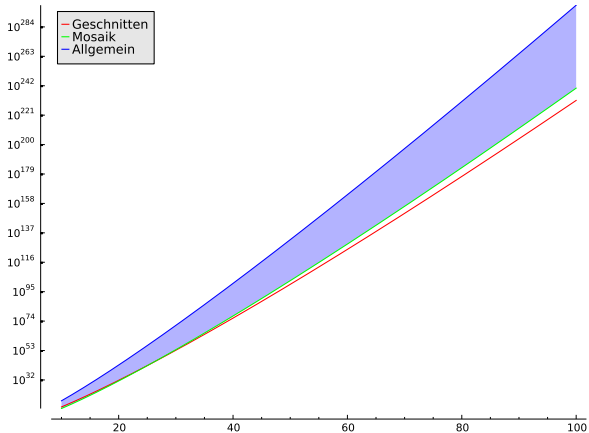
$$\rightarrow n! \frac{2^{2.543n}}{n^{1.5}}$$

Mosaik

$$\rightarrow n! \frac{2^{3n}}{n^4}$$

Allgemeine

$$\rightarrow n! \frac{2^{3n}}{n^4} \text{ bis } n! \frac{2^{5n}}{n^{4.5}}$$





- ▶ Weitere Darstellungen
 - ▶ Polare Graphen
 - ▶ Adjazenz-Graphen
 - ▶ Channel-Intersection-Graphen
 - ▶ Transitive-Closure-Graph
 - ▶ O-Tree
 - ▶ Corner-Block-List
 - ▶ Bounded-Sliceline-Grid
 - ▶ Sequence Triple
 - ▶ ...

MILP

Formulierung



- ▶ Analytischer Ansatz
 - ▶ Zwei Aspekte Berücksichtigen:
 - ▶ Überlappungsfreiheit
⇒ Abgebildet mittels Restriktionen
 - ▶ Flächenminimierung
 - ▶ Eigentliches Ziel: $x_g \cdot y_g$
 - ▶ Nichtlinear!
- Lösung: Breite W fixieren und Höhe y_g minimieren

MILP

Formulierung Überlappungsfreiheit

m_i Relation m_j	Ungleichung	p_{ij}	q_{ji}
links von	$x_i + w_i \leq x_j$	0	0
▶ über	$y_i + h_i \leq y_j$	0	1
rechts von	$x_i - w_j \geq x_j$	1	0
unter	$y_i - h_j \geq y_j$	1	1

▶ p_{ij}, q_{ij} binäre Variablen um obigen Fall auszuwählen

▶ Restriktionen (H, W Obergrenze für Floorplangröße):

$$x_i + w_i \leq x_j + W(p_{ij} + q_{ij})$$

$$y_i + h_i \leq y_j + H(1 + p_{ij} + q_{ij})$$

$$x_i - w_j \geq x_j - W(1 - p_{ij} + q_{ij})$$

$$y_i - h_j \geq y_j - H(2 - p_{ij} + q_{ij})$$

$$x_i + w_i \leq W$$

$$y_i + h_i \leq Hy_g$$

- ▶ $\mathcal{O}(n^2)$ Ungleichungen
- ▶ $\mathcal{O}(n)$ reelle Variablen, $\mathcal{O}(n^2)$ ganzzahlige Variablen
- ▶ Zur Reduzierung der Komplexität:
 - ▶ Schrittweise einige wenige Module hinzufügen
 - ▶ Vorheriges Resultat durch (wenige) überdeckende Rechtecke darstellen \Rightarrow Großes Problem wird in mehrere einfachere zerlegt (natürlich Suboptimal)



- ▶ Bisher nur feste (Hard) Module (und Rotation)
- ▶ Wie behandelt man flexible (Soft) Module?
 - ▶ Nicht nur 90° drehbar
 - ▶ Verschiedene Formen
- ▶ Feste Fläche, variables Seitenverhältnis
(\Rightarrow Bei SA eine weitere Operation: Größe der Zelle ändern)
- ▶ Bei geschnittenen Floorplans: Formfunktionen
(Module werden verformt, um leere Flächen zu füllen)



- ▶ Fixed-Die / Fixed-Outline Floorplan
(Feste vorgegebene Größe statt Flächenminimierung)
- ▶ Für große Schaltkreise:
 - ▶ Hierarchischer Ansatz (Ähnlich Multilevel Partitionierung)
- ▶ Mehrdimensionalität
 - ▶ 3D-Chips
 - ▶ Neue Datenstrukturen
 - ▶ Sequence Triple, Sequenze Quintuple
 - ▶ K-Tree, T-Tree, 3D-subTCG
 - ▶ ...
- ▶ Berücksichtigung weiterer Aspekte:
 - ▶ Buffer
 - ▶ Busse
 - ▶ Power-/Ground-Netzwerke
 - ▶ ...



- ▶ Floorplanning
- ▶ Lösungsalgorithmen:
SA, MILP
- ▶ Wichtige Darstellungen:
 - ▶ Geschnittene Floorplans (NPE, Schnitt-Bäume)
 - ▶ Kompakte Floorplans (B*-Bäume)
 - ▶ Allgemeine Floorplans (Sequenz-Paar-Darstellung)
- ▶ MILP-Formulierung
- ▶ Ausblick