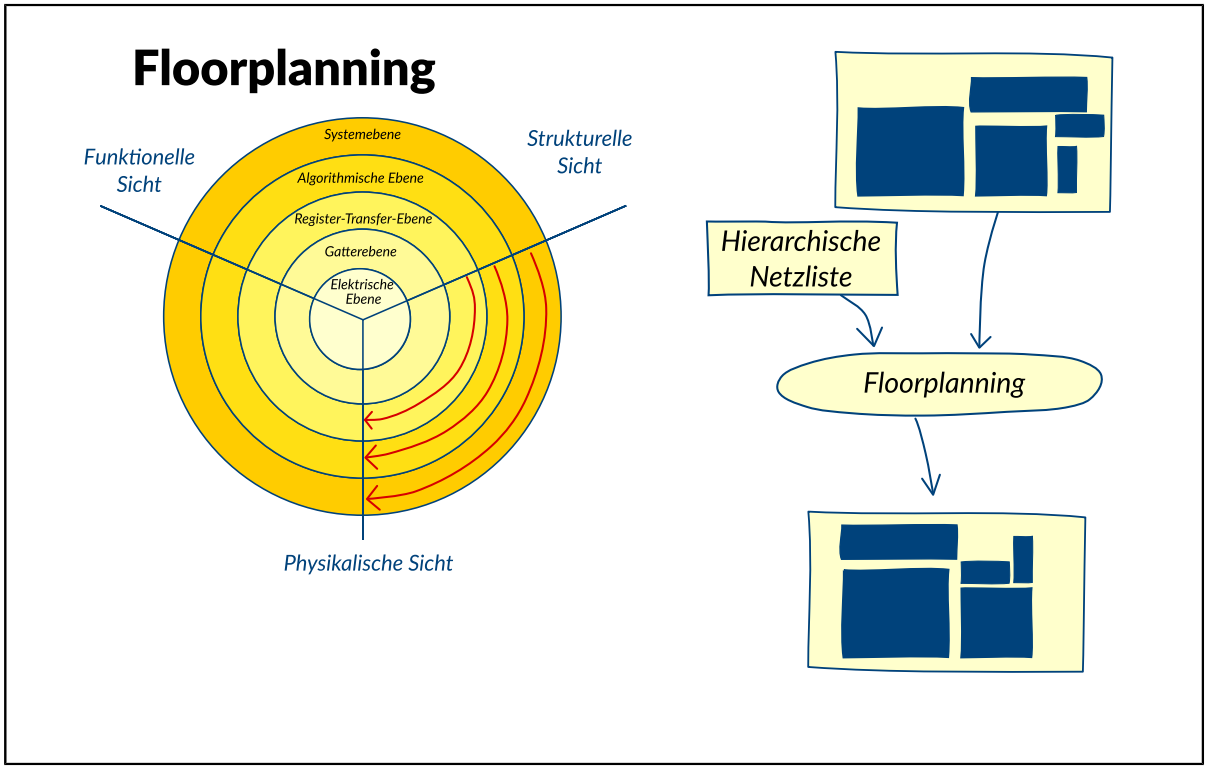
Electronic Design Automation (EDA)

Floorplanning

Floorplanning
Floorplanning als Optimierungsproblem
Abschätzung der Optimierungsziele
Slicing Floorplan
Schnittbaum, umgekehrte polnische Notation(UPN)
Algorithmen (1): Simulated Annealing
...Pseudo-Code
...Akzeptanz
...Überwinden lokaler Minima
...Anwendung auf das Floorplanning-Problem
Algorithmen (2): Graphen-Dualisierung
Vom Graph zum Floorplan
Algorithmen (3): Lineare Optimierung
Lineare Optimierung (2)
Lineare Optimierung (3)

Date of generation: 26.10.2026

Floorplanning: Floorplanning



Floorplanning ist ein verallgemeinertes Platzierungsproblem. Ausgehend von einer hierarchischen Netzliste und einer Bibliothek von möglichen geometrischen Ausprägungen der einzelnen Zellen wird eine Grobplanung des Chip-Layouts angefertigt. Dabei werden sowohl die Fläche des Chips festgelegt als auch die Positionen der Makro-, Block- oder Modulzellen auf dem Chip bestimmt.

Die Schwierigkeit beim Floorplanning resultiert aus dem zusätzlichen Freiheitsgrad, der auf der flexiblen Form der Zellen beruht. Daher benötigt das Floorplanning zusätzlich Bibliotheksinformationen über die möglichen geometrischen Realisierungen der einzelnen Zellen. Diese können sehr vage sein und lediglich Angaben über die Gesamtfläche einer Zelle enthalten, Grenzen für das Seitenverhältnis beinhalten oder aber eine oder mehrere exakt vorgegebene Alternativen für die Form und Größe der Zelle vorgeben.

Grundsätzlich kann die Form einer Zelle ganz beliebig sein. Auf Grund der Komplexität des Problems werden jedoch von den meisten Algorithmen nur rechteckige Zellen zugelassen, teilweise gibt es Erweiterungen für Zellen mit L- oder T-Form oder sogar beliebigen rechtwinkligen Formen. Im folgenden wird jedoch, wie allgemein üblich, von rechteckigen Formen der Zellen ausgegangen.

Floorplanning: Floorplanning als Optimierungsproblem

Floorplanning als Optimierungsproblem

Optimierungskriterien:

- Fläche
- Gesamtverdrahtungslänge
- Länge des kritischen Pfads
- Temperaturverteilung
- Power

Randbedingungen:

- Seitenverhältnis des Chips
- ...

Floorplanning ist ein Optimierungsproblem. Ziel ist es, die Positionen und Formen der Zellen derart zu bestimmen, dass die Chipfläche und die Verdrahtungslänge der globalen Netze, welche die Zellen miteinander verbinden, minimiert werden, um geringe Fertigungskosten sowie hohe Performance zu bewirken. Dazu sollten die Zellen so angeordnet werden, dass möglichst wenig nutzbare Fläche verloren geht und stark verbundene Zellen möglichst dicht bei einander liegen.

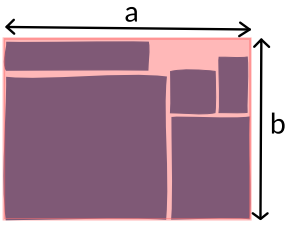
Für einen gegebenen Floorplan ist der Verschnitt und somit die Chipfläche eindeutig bestimmbar, während die Verdrahtungslänge nur abgeschätzt werden kann, da die echte Verdrahtung erst später vorgenommen wird. Dabei kann entweder die Abschätzung der Gesamtlänge aller Netze oder die Länge des längsten Netzes berücksichtigt werden. Daneben können weitere Optimierungskriterien und Randbedingungen auftreten, wie z.B. die Temperaturverteilung des Chips oder ein gewünschtes Seitenverhältnis.

Floorplanning: Abschätzung der Optimierungsziele

Abschätzung der Optimierungsziele

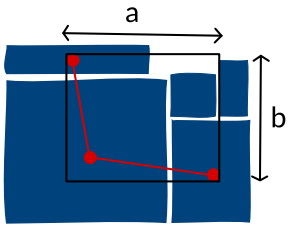
Fläche:

Kleinstes umschließendes Rechteck



Verdrahtungslänge:

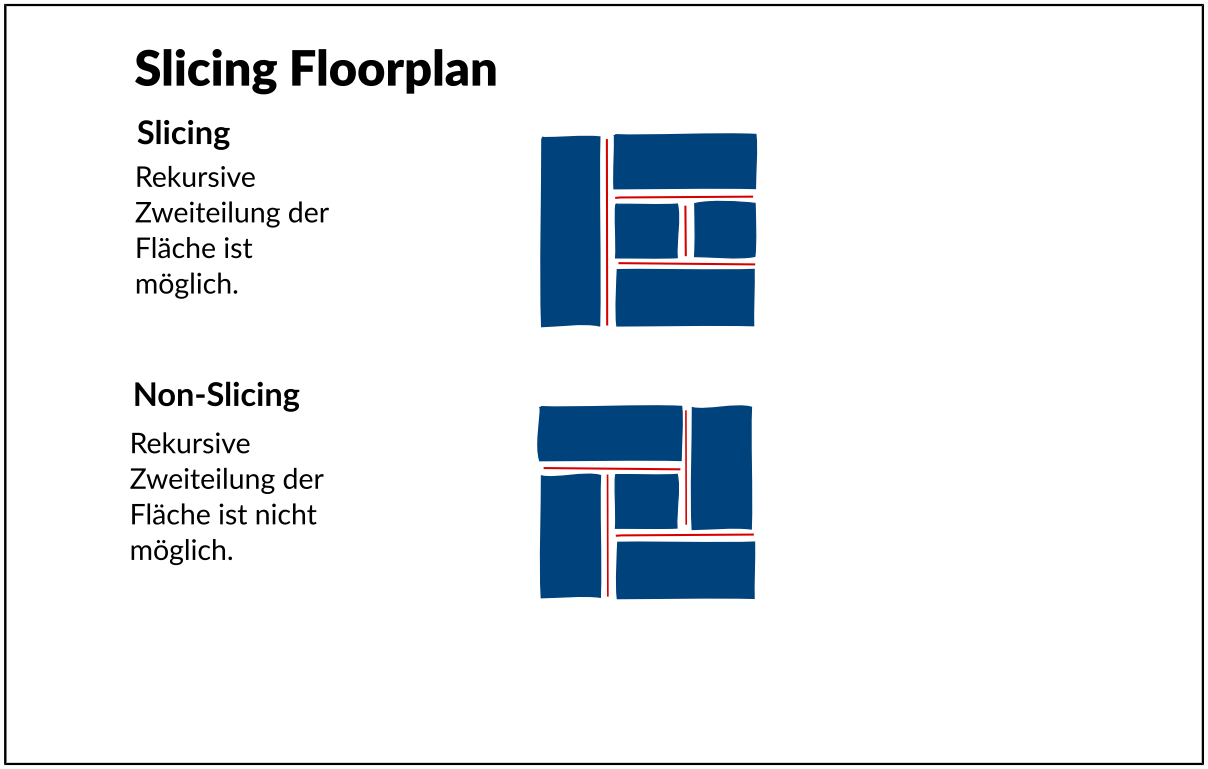
Halber Umfang des umschließendes Rechtecks



Zur Bewertung eines Floorplans werden insbesondere die Chipfläche und die Gesamtverdrahtungslänge betrachtet. Da im Allgemeinen nur rechteckige Chips gefertigt werden können, wird die Chipfläche definiert über ein Rechteck, das alle Zellen beinhaltet, das heißt über die Fläche des umschließenden Rechtecks.

Für die Verdrahtungslänge werden beim Floorplanning die globalen Netze betrachtet, welche die Zellen miteinander verbinden. Da es sich beim Floorplanning um ein sehr frühes Entwurfsstadium handelt, in dem nur wenig Informationen über das endgültige Layout vorhanden sind, muss eine grobe Abschätzung der Verdrahtungslängen genügen. Dafür eignet sich der halbe Umfang der Bounding-Box. Die Bounding-Box stellt das umschließende Rechteck aller Anschlüsse eines Netzes dar.

Floorplanning: Slicing Floorplan

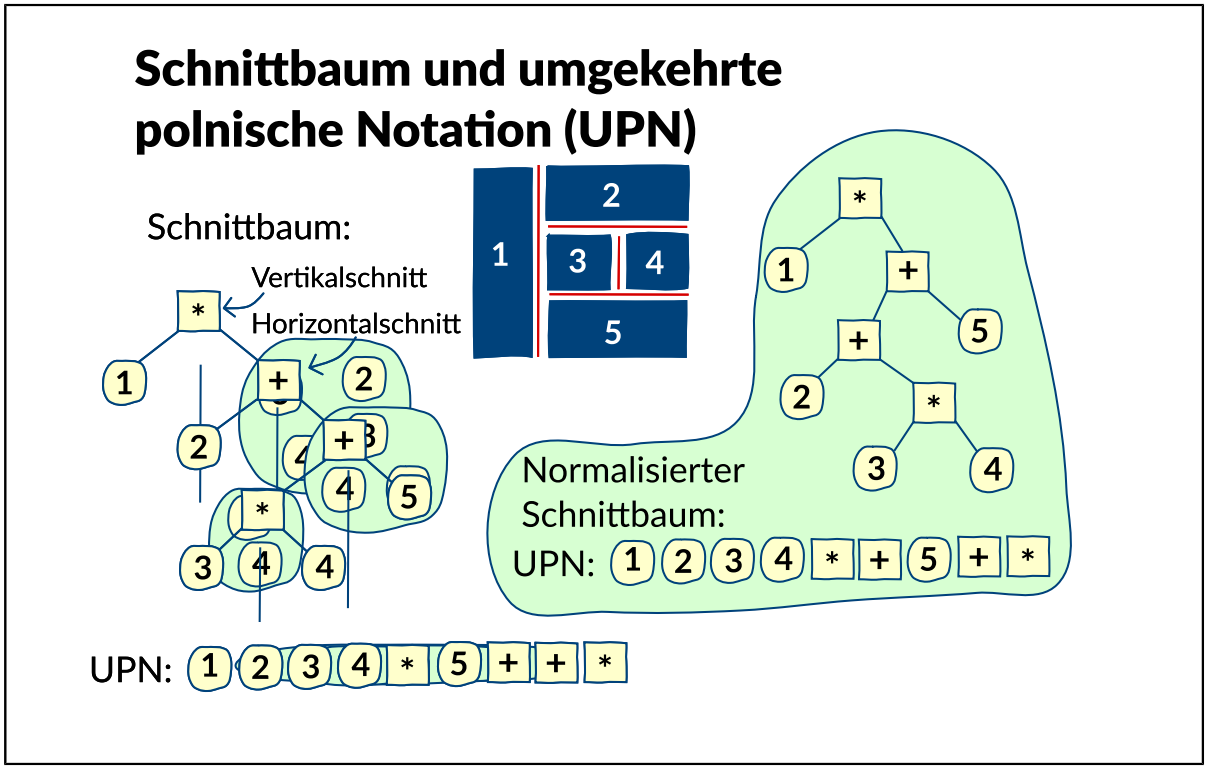


Aufgrund der verschiedenen Datenstrukturen, mit denen Floorplans repräsentiert und gespeichert werden können, ergibt sich eine Unterteilung in zwei verschiedene Klassen von Floorplans: Slicing Floorplans und Non-Slicing Floorplans.

Floorplans, die sich durch eine rekursive Zweiteilung der Layoutfläche erhalten lassen, werden Slicing Floorplans genannt. Alle anderen, allgemeinen Floorplans ohne Einschränkung gehören zu den Non-Slicing Floorplans. Ein Beispiel für einen Non-Slicing Floorplan ist das so genannte Rad (s. Abb.).

Beschränkt man sich durch die Wahl der Datenstruktur auf Slicing Floorplans, so wird auch der Raum der möglichen Lösungen eingeschränkt und damit kann das globale Optimum eventuell nicht erreicht werden. Andererseits können durch diese Einschränkung die Daten sehr effizient gespeichert und der beschränkte Lösungsraum wesentlich effizienter durchsucht werden. Daher finden beide Arten von Floorplans Verwendung.

Floorplanning: Schnittbaum, umgekehrte polnische Notation(UPN)



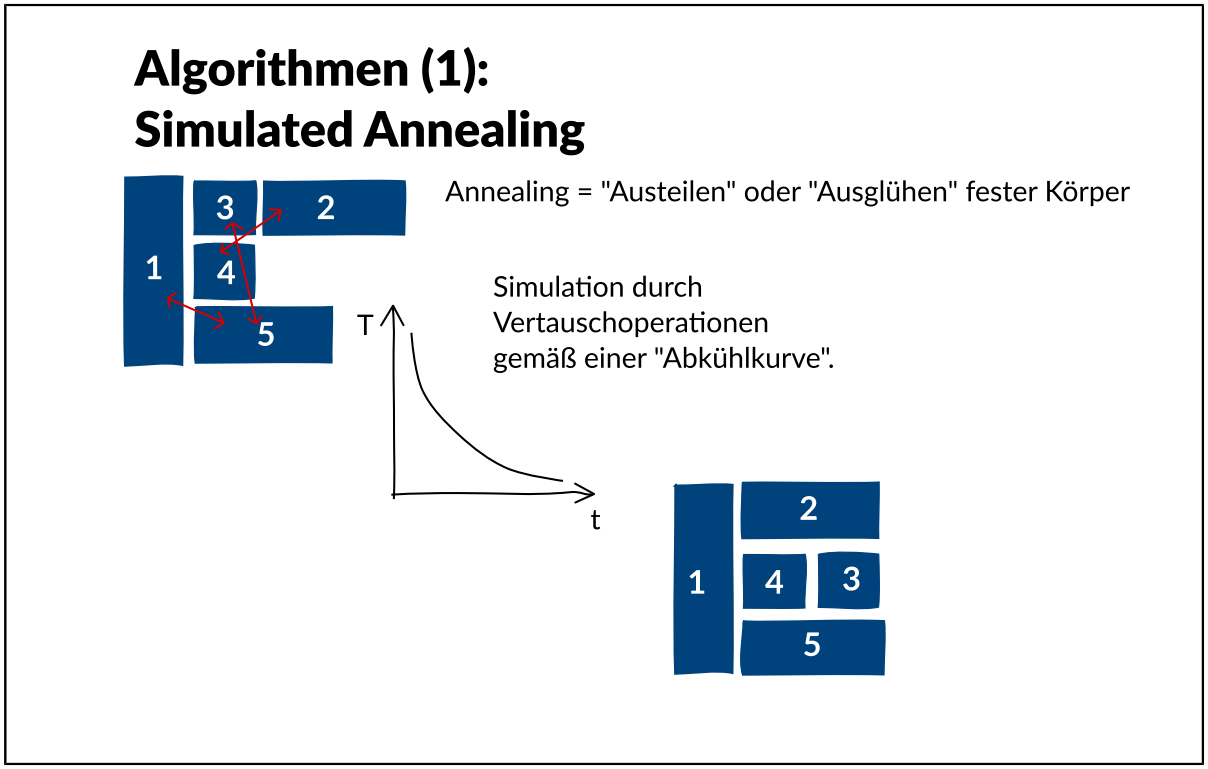
Die am häufigsten verwendete Datenstruktur für das Floorplanning sind so genannte Schnittbäume, welche durch die umgekehrte polnische Notation (UPN) beschrieben werden können. Schnittbäume sind geordnete, vollständige, binäre Bäume. In einem Schnittbaum werden die einzelnen Module als Blätter des Baums dargestellt, während die Verzweigungsknoten Schnittlinien repräsentieren. Dabei enthält jeder Knoten ein Label für die Schnittrichtung. Diese gibt an, ob die beiden Teilbäume dieses Knotens durch einen horizontalen oder einen vertikalen Schnitt miteinander verbunden sind, d.h. ob deren Teilflächen nebeneinander oder übereinander liegen. Durch Schnittbäume können lediglich Slicing Floorplans dargestellt werden.

Durchläuft man einen Schnittbaum mit einem Post-Order-Durchlauf, so erhält man den zu dem Schnittbaum äquivalenten UPN-Ausdruck. Die Zellen sind die Operanden, die durch ihre Nummer gekennzeichnet sind. "+" bzw. "*" sind Operatoren für einen horizontalen bzw. vertikalen Schnitt.

Es existieren i.A. mehrere verschiedene Schnittbäume bzw. UPN-Ausdrücke, welche den gleichen Floorplan beschreiben. Um zu einer normalisierten Form zu kommen, wird eine zusätzliche Bedingung an die Datenstruktur geknüpft:

In einem normalisierten Schnittbaum muss die Schnittrichtung des rechten Teilbaums immer verschieden sein von der seines Vorgängerknotens. Für die normalisierte UPN folgt, dass keine zwei aufeinander folgenden Operatoren gleich sein dürfen.

Floorplanning: Algorithmen (1): Simulated Annealing



Simulated Annealing ist ein Verfahren, das das Ausglühen (Annealing) von festen Körpern nachbildet. Dabei werden Spannungen in festen Körpern entfernt. Das Ergebnis ist ein Zustand minimaler Energie.

Beim Ausglühen, wie auch beim Algorithmus; sind verschiedene Aspekte zu berücksichtigen, um ein möglichst gutes Ergebnis zu erzielen. Zum einen muss die Starttemperatur, also die Temperatur, bei der das Ausglühen beginnen soll, hoch sein, damit die Moleküle in starke Schwingungen versetzt werden. Entsprechend einer zu definierenden Abkühlkurve (Annealing Schedule) wird die Temperatur gesenkt und die Moleküle ordnen sich in einem spannungsfreien Verbund an.

Floorplanning: ...Pseudo-Code

Simulated Annealing: Pseudo-Code

```
T = Anfangstemperatur;
Initiallösung;
BEWERTE (Initiallösung);
while(! Gefrierpunkt erreicht)
{
   while(! Gleichgewicht für T erreicht)
   {
      VERÄNDERE (aktuelle Lösung);
      BEWERTE (neue Lösung);
      if(AKZEPTANZ)
      {
          Übernehme neue Lösung;
      }
   }
   Senke Temperatur T;
}
```

Simulated Annealing wird für verschiedene Optimierungsprobleme eingesetzt. Dem Körper entspricht im Algorithmus eine mögliche Lösung, die Schwingung der Moleküle wird durch Störungen bzw. VERÄNDERUNGEN der aktuellen Lösung zu einer neuen realisiert.

Zu jeder Temperatur werden so viele VERÄNDERUNGEN durchgeführt, bis ein Gleichgewichtszustand für diese Temperatur erreicht ist. Das kann z.B. nach einer festen Anzahl von Veränderungsschritten der Fall sein, oder wenn sich die Qualität der Lösung über eine bestimmte Anzahl von Schritten nicht mehr verändert hat. Als Kostenfunktion zur BEWERTUNG wird im Allgemeinen die gewichtete Summe von Fläche und Verdrahtungslänge betrachtet.

Floorplanning: ...Akzeptanz

Simulated Annealing: Akzeptanz

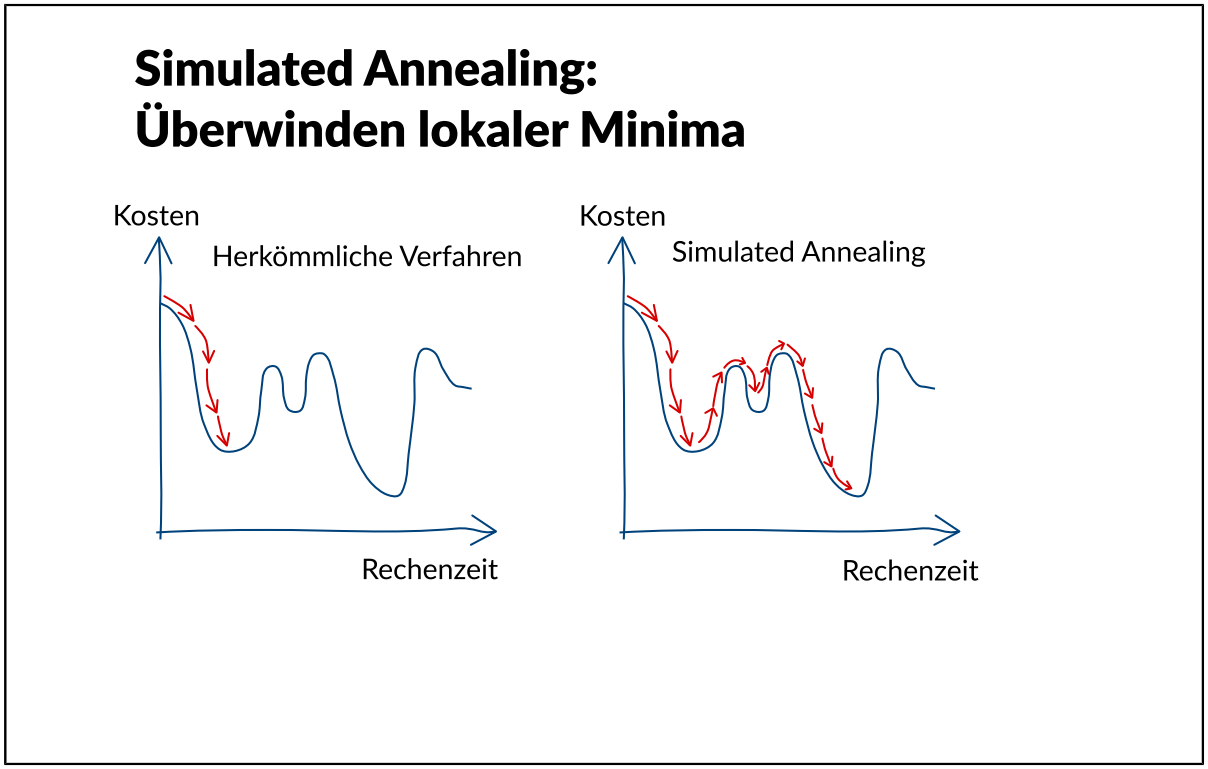
```
ΔK = NeueKosten - AlteKosten;

if ((ΔK < 0)  or  (RANDOM (0,1) < EXP(-ΔK/kT)))
   {
      Akzeptanz der neuen Lösung
   }
```

Entsprechend der Temperatur schwingen die Moleküle zunächst mehr, später weniger stark. Dementsprechend werden je nach Temperatur mehr oder weniger starke Verschlechterungen der aktuellen Lösung akzeptiert. Veränderungen, die zu einer Verbesserung führen, werden immer akzeptiert. Neben der Temperatur und der Differenz der Kosten der Lösungen hängt die Akzeptanz einer Veränderung auch noch vom Zufall ab. Dieser sorgt dafür, dass es sich beim Simulated Annealing um einen stochastischen, nicht-deterministischen Algorithmus handelt. Je geringer die Temperatur ist, desto geringer ist auch die Wahrscheinlichkeit, dass eine Verschlechterung akzeptiert wird. Je stärker sich die Kosten erhöhen, desto unwahrscheinlicher wird die Akzeptanz dieser Veränderung. Meistens wird dieser Zusammenhang durch die Boltzmann-Wahrscheinlichkeitsfunktion exp((NeueKosten-AlteKosten) / (k*Temperatur)) modelliert, wobei k die Boltzmann-Konstante ist.

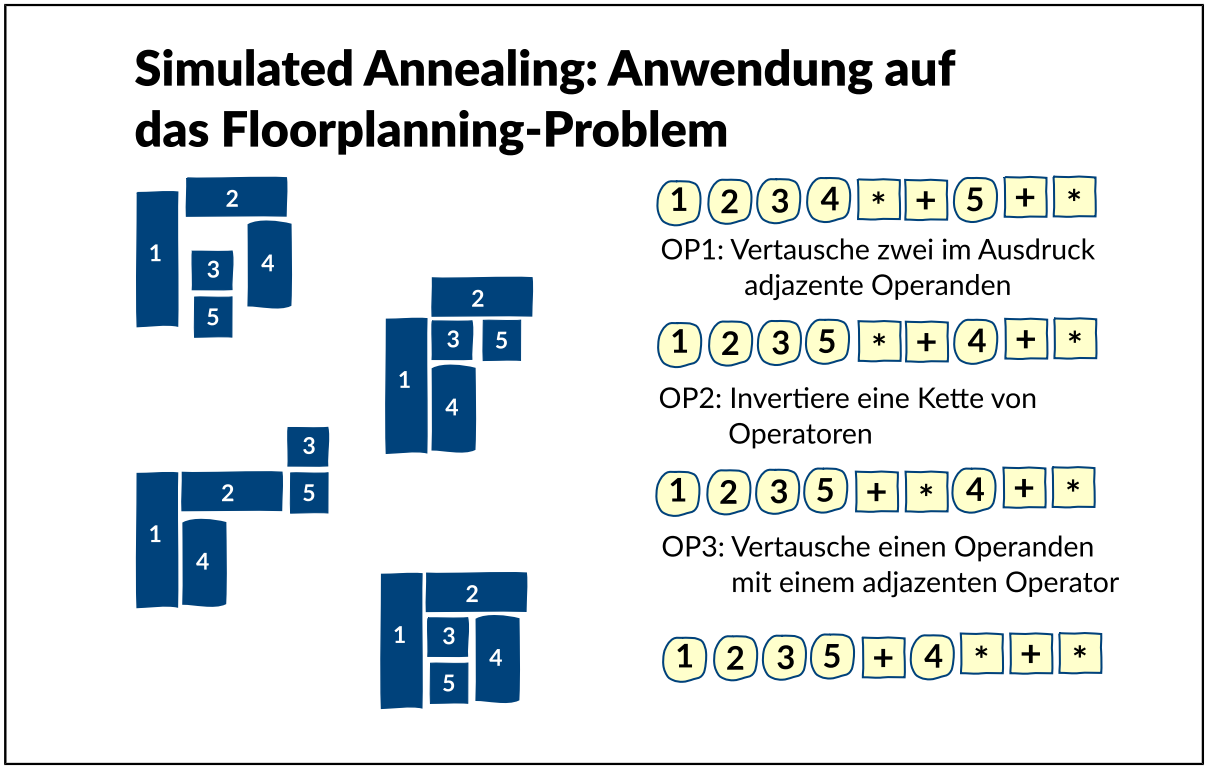
Im Allgemeninen wird die letzte bekannte beste Lösung noch einmal seperat abgespeichert.

Floorplanning: ...Überwinden lokaler Minima



Das Verfahren wird in verschiedenen EDA-Bereichen angewandt. Es ist einfach zu implementieren, liefert gute Ergebnisse, ist aber sehr rechenzeitintensiv. Das Verfahren hat die Eigenschaft, dass lokale Minima überwunden werden können. Der Grund hierfür liegt in der Tatsache, dass auch Platzierungsergebnisse, die höhere Kosten als die vorhergehende Platzierung erzeugen, akzeptiert werden können.

Floorplanning: ...Anwendung auf das Floorplanning-Problem



Beim Floorplanning ist Simulated Annealing das derzeit am meisten verwendete Verfahren. Ein bekannter Floorplanning-Algorithmus basierend auf Simulated Annealing arbeitet dabei mit der umgekehrten polnischen Notation (UPN). Die Startlösung wird im Allgemeinen zufällig generiert. Entscheidend für den Algorithmus sind der Satz der erlaubten Veränderungen und die Kostenfunktion.

Veränderungen:

Ausgehend von einem normalisierten UPN-Ausdruck wird eine neue Lösung, die wiederum eine normalisierte UPN ist, durch eine der folgenden drei Operationen bestimmt:

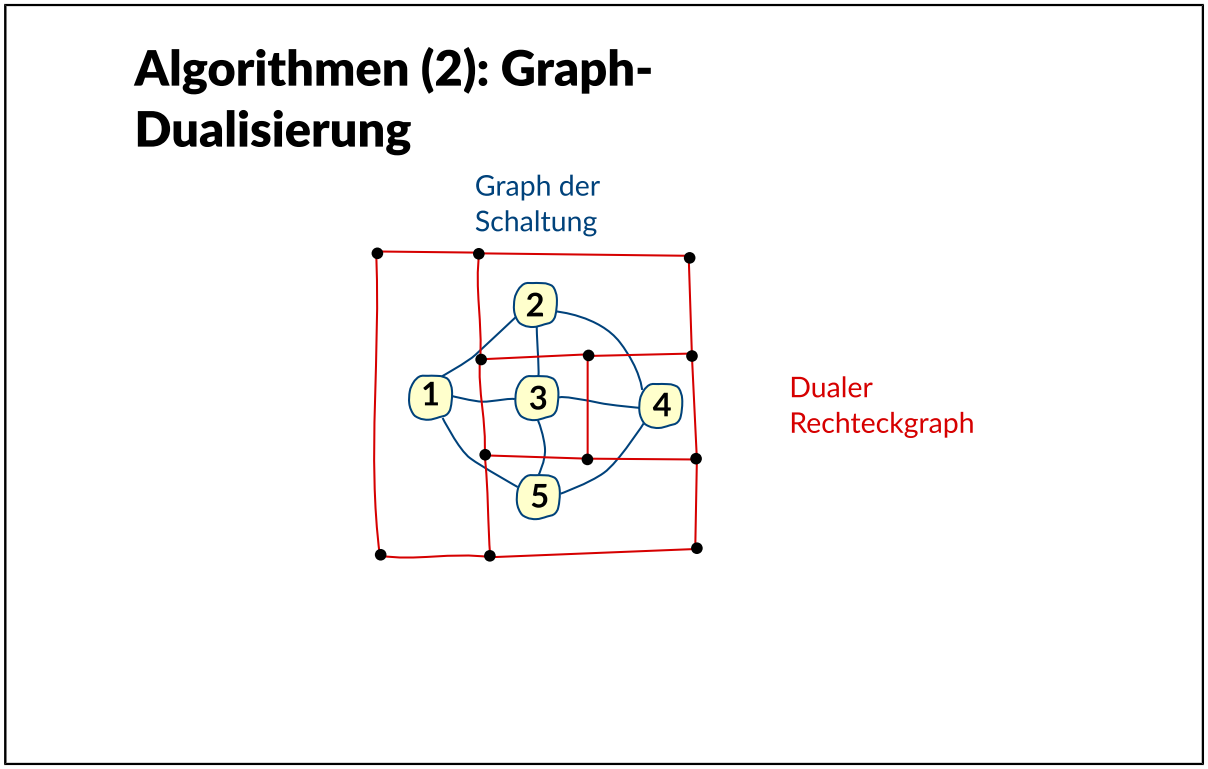
OP1: Vertausche zwei im Ausdruck adjazente Operanden

OP2: Invertiere eine Kette von Operatoren (* wird zu + und umgekehrt)

OP3: Vertausche einen Operanden mit einem adjazenten Operator, falls das Ergebnis wieder eine normalisierter UPN-Ausdruck ist.

Dieser Satz von Veränderungen ist vollständig, d.h. ausgehend von jedem beliebigen normalisierten UPN-Ausdruck kann jeder beliebige andere über eine Folge dieser Operationen erreicht werden.

Floorplanning: Algorithmen (2): Graphen-Dualisierung



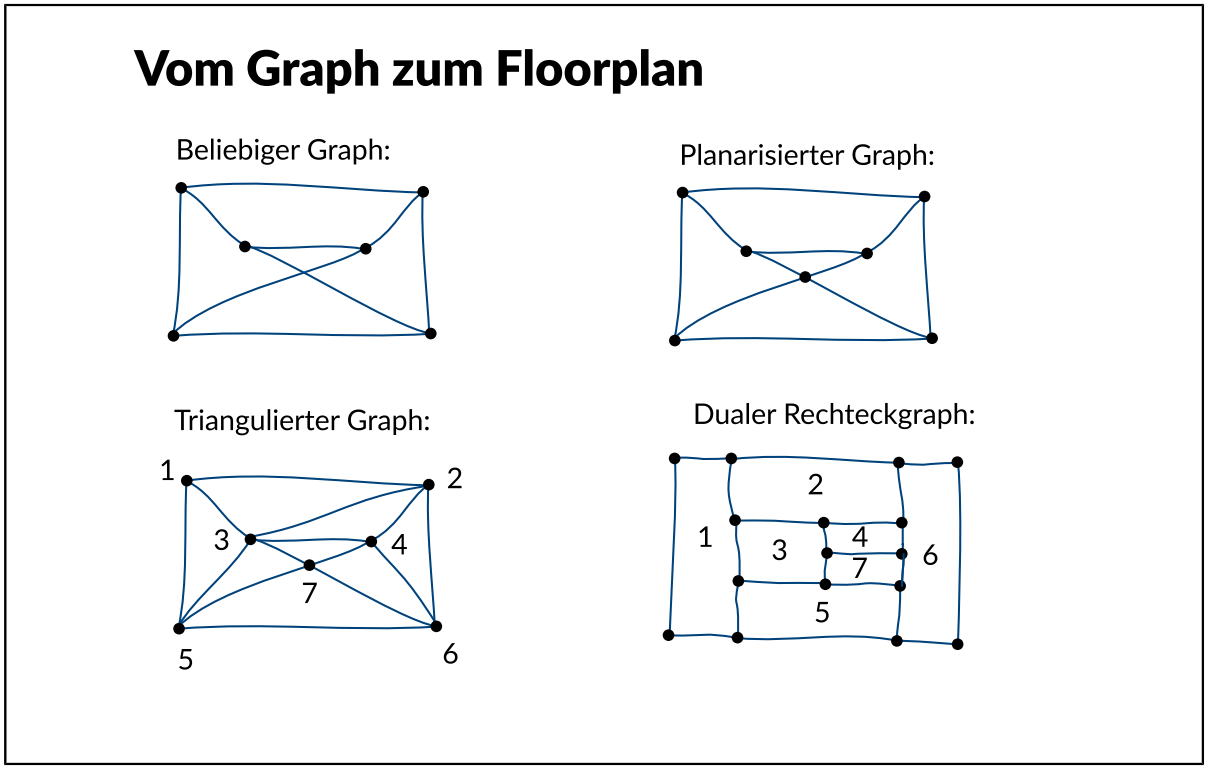
Bei der Graphen-Dualisierung wird die Schaltung zunächst durch einen Graphen G dargestellt, dessen Knoten die Zellen der Schaltung und die Kanten deren Verbindungen untereinander darstellen. Ein Floorplan kann daraus ermittelt werden, indem der zu G duale Rechteckgraph (DRG) aufgestellt wird. Dieser besteht aus nicht-überlappenden Rechtecken, welche die folgenden Bedingungen erfüllen:

1. Jeder Knoten v_i entspricht einem Rechteck R_i.

2. Für jede Kante (v_i, v_j) sind die Rechtecke R_i und R_j adjazent.

Die Graph-Dualisierung führt zur Maximierung von Adjazenzen von Blöcken, die stark (z.B. durch mehtfache Verbindungen) miteinander verbunden sind.

Floorplanning: Vom Graph zum Floorplan



Nicht jeder Graph besitzt einen dualen Rechteckgraphen. Allerdings besitzen planare, triangulierte Graphen (PTG) einen dualen Rechteckgraphen. Um aus einem beliebigen Graphen einen PTG zu machen, sind die folgenden Schritte notwendig:

1. Planarisierung: An Kreuzungspunkten werden Knoten hinzugefügt oder es werden überlappende Kanten entfernt.

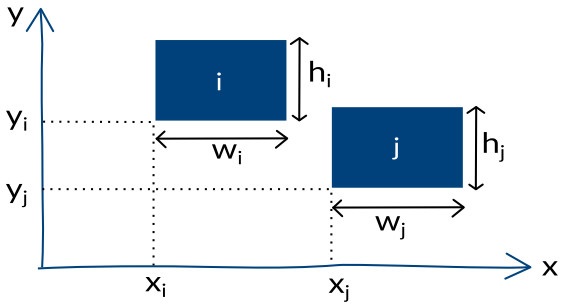
2. Triangulierung: In Gebiete, die keine Dreiecke sind, werden zusätzliche Kanten eingefügt.

Zusätzliche Knoten entsprechen im Floorplan Platzhalter-Rechtecken, zusätzliche Kanten bedingen Adjazenzen von Zellen, die keine Verbindungen haben, durch das Entfernen von Kanten werden Verbindungen von Zellen ignoriert. Daher sollen bei diesen Schritten so wenig Veränderungen wie nötig durchgeführt werden, um eine gute Ergebnisqualität zu erreichen.

Für PTGs existieren Algorithmen, um die Topologie, die in dem planaren Graphen enthalten ist, in einen dualen Rechteckgraphen umzusetzen, der allerdings nicht eindeutig sein muss.

Floorplanning: Algorithmen (3): Lineare Optimierung

Algorithmen (3): Lineare Optimierung (1)



Formulierung des Problems durch ein System von Ungleichungen:

Horizontale Beziehungen: $x_i + w_i \leq x_j$ (i ist links von j)
$x_j + w_j \leq x_i$ (i ist rechts von j)

Vertikale Beziehungen: $y_i + h_i \leq y_j$ (i ist unterhalb von j)
$y_j + h_j \leq y_i$ (i ist oberhalb von j)

Lineare Optimierung ist ein analytisches Verfahren, mit dem sich das Floorplanning-Problem zurückführen lässt auf das Lösen eines linearen Ungleichungssystems. Für diese Aufgabe existieren aus der Mathematik zahlreiche Lösungsverfahren. Hauptaufgabe ist also die Formulierung des Floorplanning-Problems durch einen Satz von linearen Ungleichungen für die Randbedingungen und eine lineare Kostenfunktion.

Die Kostenfunktion (beim Floorplanning insbesondere die Größe der Chipfläche) ist eine nichtlineare Funktion (A=X*Y), kann aber leicht linearisiert werden, indem eine Variable (z.B. die Chipbreite) als fest angenommen wird und die andere minimiert wird. Im folgenden werden die wichtigsten Randbedingungen durch lineare Ungleichungen beschrieben.

Randbedingung Überlappungsfreiheit:

In einem gültigen Floorplan dürfen sich keine zwei Zellen überlappen. Für zwei Zellen i und j, deren linke untere Ecke jeweils an der Koordinate (xi, yi) bzw. (xj, yj) liegt und die die Höhe hi (bzw. hj) und Breite wi (bzw. wj) aufweisen, ist diese Bedingung erfüllt, wenn mindestens eine der abgebildeten Gleichungen gültig ist.

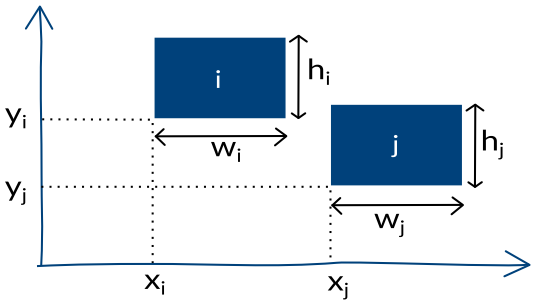
Floorplanning: Lineare Optimierung (2)

Lineare Optimierung (2)

Es muss immer nur eine Gleichung erfüllt sein. Daher: Einführung von Hilfsgrößen:

W: Obergrenze für Gesamtbreite
H: Obergrenze für Gesamthöhe

p_{ij}, q_{ij}: aus der Menge {0,1} steuern die Auswahl der gültigen Bedingungen



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

$$y_i + h_i \leq y_j + H * (1 + p_{ij} - q_{ij})$$
$$y_j + h_j \leq y_i + H * (2 - p_{ij} - q_{ij})$$

Zum Beispiel:
$p_{ij} = 0$ und $q_{ij} = 1$
$y_i + h_i \leq y_j$

Da es ausreicht, dass eine dieser Gleichungen erfüllt ist, werden für jedes Paar von Zellen i und j zwei zusätzliche Variablen pij und qij eingeführt, die nur die Werte 0 und 1 annehmen können und die Auswahl der jeweiligen Bedingung darstellen. Außerdem werden zwei zusätzliche Größen W und H eingeführt, welche die oberen Grenzen der Breite und der Höhe der gültigen Gesamtlösung angeben.

Dadurch ergibt sich das dargestellte Ungleichungssystem. Zum Beispiel muss für pij=0 und qij=1 die Ungleichung yi + hi <= yj erfüllt sein, während alle anderen Gleichungen auf Grund des zusätzlichen Terms W bzw. H keine Rolle spielen. Mindestens eine der ursprünglichen Gleichungen ist also bei jeder Wahl von p und q erfüllt.

Floorplanning: Lineare Optimierung (3)

Lineare Optimierung (3)

Floorplanningproblem als
nichtlineares Optimierungsproblem:

min A = min X * Y

Linearisierung durch Vorgabe
einer festen Gesamtbreite X:

min Y

Alle Zelle-Koordinaten sind positiv.

$x_i, y_i \geq 0$

Alle Zellen liegen innerhalb der Layoutfläche.

$x_i + w_i \leq X$
$y_i + h_i \leq Y$

Zellen dürfen sich nicht überlappen.

$x_i + w_i \leq x_j + W * (p_{ij} + q_{ij})$ $\quad p_{ij}, q_{ij} = 0$ oder 1
$x_j + w_j \leq x_i + W * (1 - p_{ij} + q_{ij})$

$y_i + h_i \leq y_j + H * (1 + p_{ij} - q_{ij})$
$y_j + h_j \leq y_i + H * (2 - p_{ij} - q_{ij})$

für alle $1 \leq i \leq n$
bzw. $1 \leq i < j \leq n$

Für einen gültigen Floorplan müssen die folgenden Bedingungen zutreffen:

1. Alle Zelle-Koordinaten sind positiv, d.h. 0 <= x_i und 0 <= y_i

2. Jede Zelle befindet sich vollständig innerhalb des den Floorplan umgebenden Rechtecks mit Breite X und Höhe Y, d.h. $x_i + w_i$ <= X und $y_i + h_i$ <= Y.

3. Keine zwei Zellen überlappen sich.

Ziel ist es, die Fläche A=X*Y des Floorplans zu minimieren. Dies ist jedoch eine nichtlineare Bedingung. Eine Möglichkeit ist es, die Breite X festzuhalten und die Höhe Y zu minimieren. Insgesamt folgt dann für den linearen Optimierungsalgorithmus der angegebene Satz von Bedingungen.

Erlaubt man den Zellen eine flexible Form, so sind h_i und w_i ebenfalls unbestimmte Variablen, für die weitere Gleichungen eingeführt werden können. Auch die Abschätzung der Verdrahtungslänge kann durch einen Satz von geeigneten Gleichungen berücksichtigt werden.

Neben den vorgestellten Verfahren werden für das Floorplanning auch noch genetische Algorithmen, Min-Cut- basierte Algorithmen, Cluster-Verfahren und andere verwendet. Diese werden auch bei der Platzierung eingesetzt.