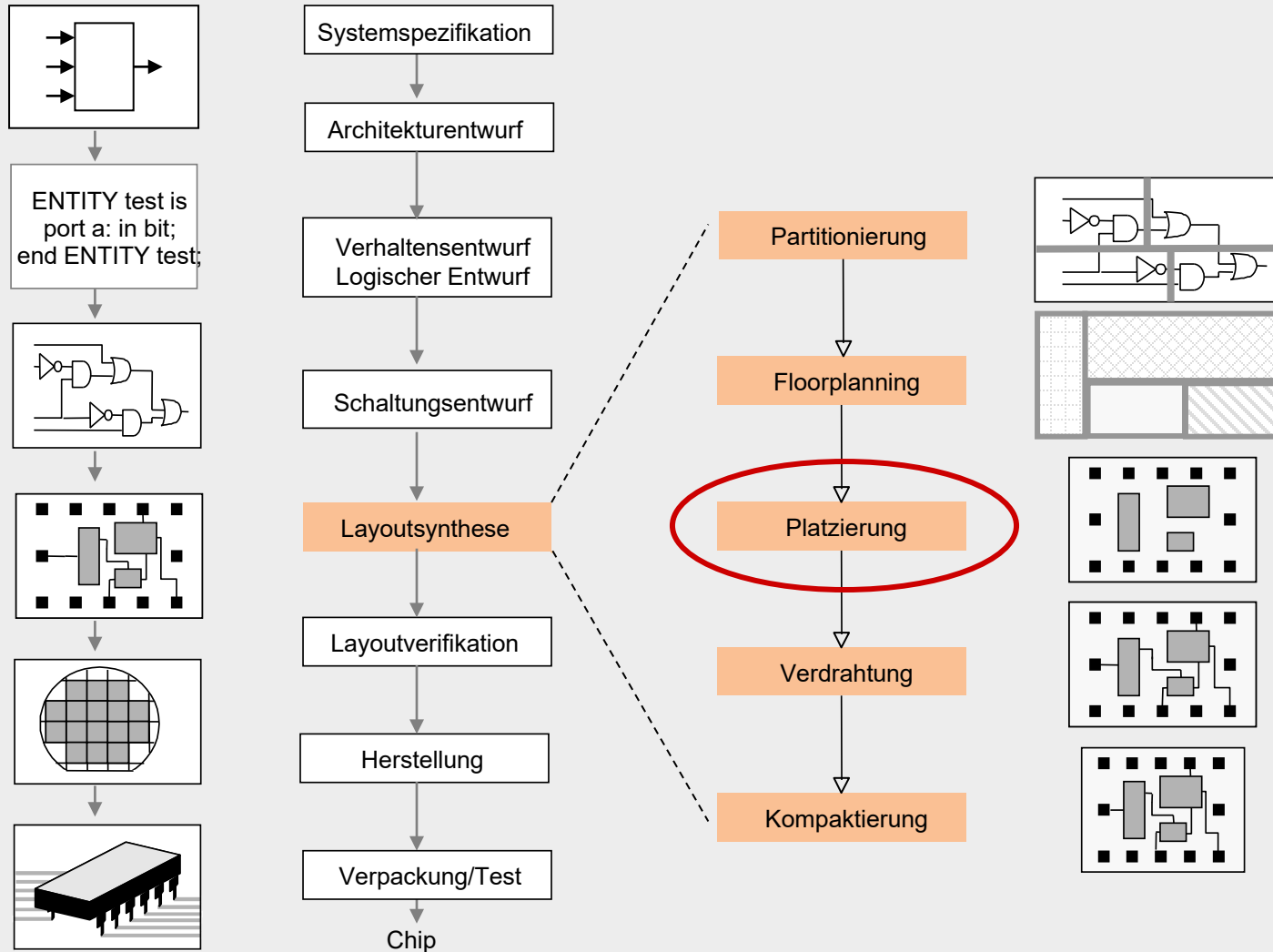


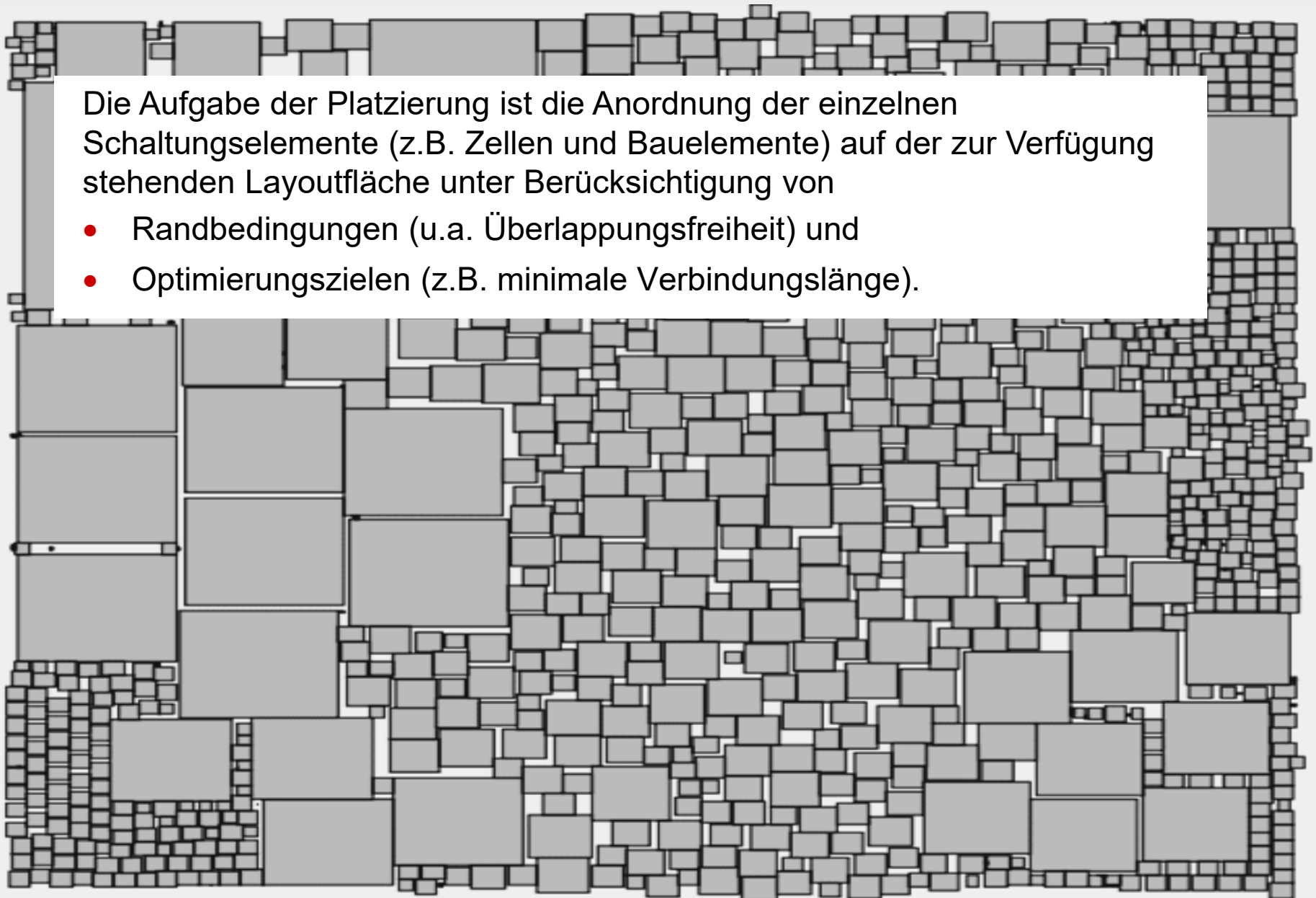
- 4.1 Einführung
- 4.2 Optimierungsziele
  - 4.2.1 Gewichtete Gesamtverbindungslänge
  - 4.2.2 Maximale Schnittanzahl
  - 4.2.3 Lokale Verdrahtungsdichte
  - 4.2.4 Signalverzögerungen
- 4.3 Platzierungsalgorithmen
  - 4.3.1 Min-Cut-Platzierung
  - 4.3.2 Min-Cut-Platzierung mit Anschlussfestlegung
  - 4.3.3 Quadratische Platzierung
  - 4.3.4 Kräfteplatzierung mittels ZFT-Position
  - 4.3.5 Simulated Annealing
  - 4.3.6 Weitere Platzierungsalgorithmen
- 4.4 Aktuelle Platzierungswerkzeuge

# 4.1 Einführung



Die Aufgabe der Platzierung ist die Anordnung der einzelnen Schaltungselemente (z.B. Zellen und Bauelemente) auf der zur Verfügung stehenden Layoutfläche unter Berücksichtigung von

- Randbedingungen (u.a. Überlappungsfreiheit) und
- Optimierungszielen (z.B. minimale Verbindungslänge).



1. Globale Platzierung (Global Placement)  
Verdrahtungslänge, Verdrahtbarkeit usw.
2. Legalisierung (Legalization, optional)  
Überlappungen beseitigen
3. Detaillierte Platzierung (Detailed Placement)  
Lösungsverbesserung

4.1 Einführung



4.2 Optimierungsziele

4.2.1 Gewichtete Gesamtverbindungslänge

4.2.2 Maximale Schnittanzahl

4.2.3 Lokale Verdrahtungsdichte

4.2.4 Signalverzögerungen

4.3 Platzierungsalgorithmen

4.3.1 Min-Cut-Platzierung

4.3.2 Min-Cut-Platzierung mit Anschlussfestlegung

4.3.3 Quadratische Platzierung

4.3.4 Kräfteplatzierung mittels ZFT-Position

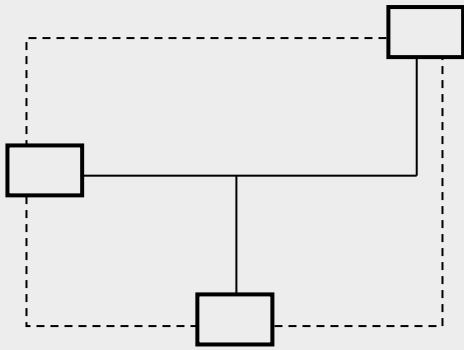
4.3.5 Simulated Annealing

4.3.6 Weitere Platzierungsalgorithmen

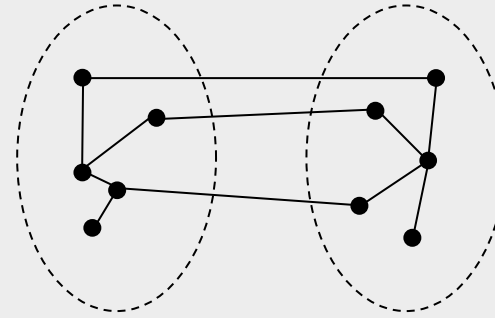
4.4 Aktuelle Platzierungswerkzeuge

- Vorgriff auf das zu erwartende Verdrahtungsergebnis
- Platzierung nur bei 100%iger Verdrahtung sinnvoll + weitere Optimierungsziele
- Einfache, schnell zu ermittelnde Parameter, die auf Verdrahtungseigenschaften schließen lassen

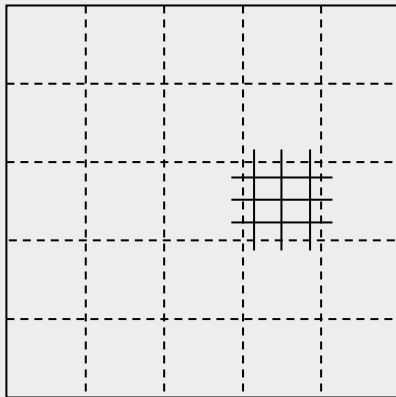
## 4.2 Optimierungsziele



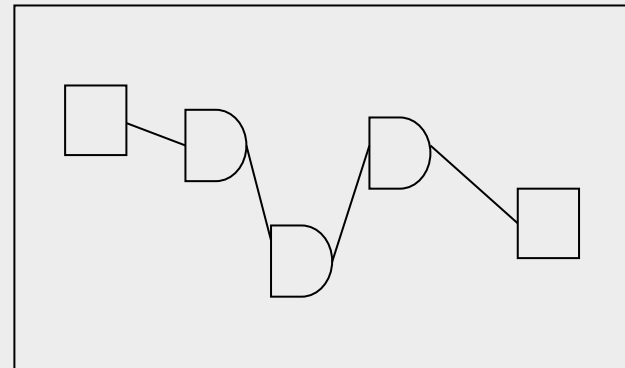
Gesamtverbindungs-länge



Anzahl der geschnittenen Netze



Lokale  
Verdrahtungsdichte



Signalverzögerungen

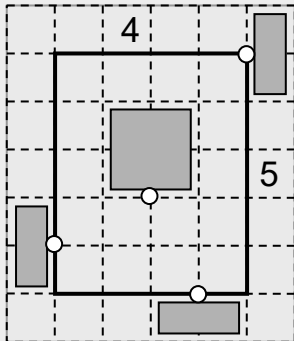
## 4.2.1 Optimierungsziel: Gesamtverbindungslänge

- Oberste Priorität: Schnelligkeit des Algorithmus
- Annahme: Netze werden in Manhattan-Geometrie verdrahtet. D.h., ein **Zweipunkt-Netz**, welches die beiden Zellen  $i$  und  $j$  verbindet, hat eine Manhattan-Länge von  $x_{ij} + y_{ij}$ , wobei  $x_{ij}$  der Horizontalabstand der beiden Zellen  $i$  und  $j$  ist und  $y_{ij}$  deren Vertikalabstand.
- Abschätzung der Verdrahtungslängen bei **Mehrpunkt-Netzen** nach verschiedenen Möglichkeiten.



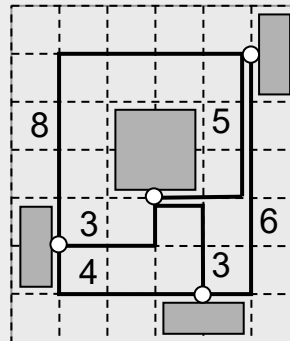
### Abschätzung der Verdrahtungslängen bei Mehrpunktnetzen

Halber Umfang des minimal umschließenden Rechtecks  
(Semi-perimeter method)



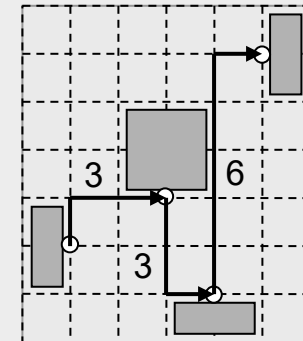
Länge halber Netz-  
umfang = 9

Kompletter Graph  
(Complete graph)



Länge kompletter Graph  
\* 2 / Pinanzahl = 14,5

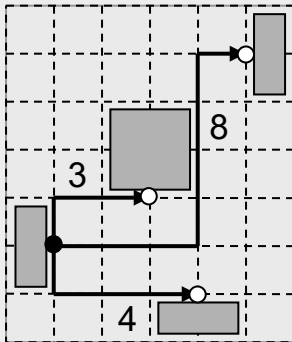
Minimale Kette  
(Minimum chain)



Kettenlänge = 12

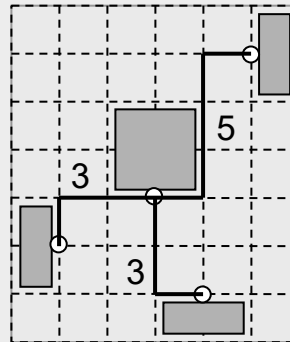
### Abschätzung der Verdrahtungslängen bei Mehrpunktnetzen (2)

Quelle-Senken-  
Verbindung  
(Source to sink  
connection)



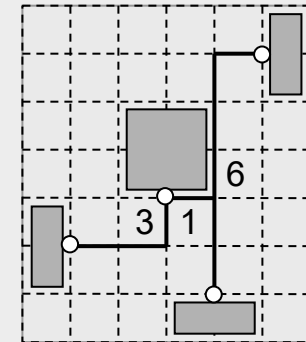
Quelle-Senken-  
Länge = 15

Minimaler rektilinear  
Spannbaum  
(Minimum rectilinear  
spanning tree)



Spannbaum-  
Länge = 11

Steinerbaum-  
Abschätzung  
(Steiner tree  
approximation)

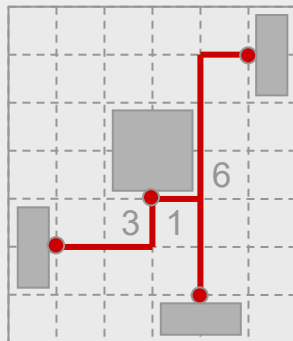


Steinerbaum-  
Länge = 10

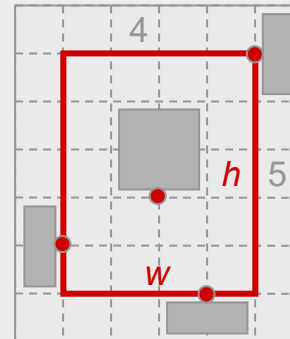
### Abschätzung der Verdrahtungslängen bei Mehrpunktnetzen (3)

Bevorzugte Methode: Halber Umfang des minimal umschließenden Rechtecks (Semi-perimeter method)

- schnell berechenbar (10 x schneller als Steinerbaum)
- entspricht der Länge des Steinerbaums bei Zwei- und Dreipunkt-Netzen
- für eine komplette Schaltung: Fehler zum Steinerbaum etwa 8% [Chu, ICCAD 04]



Steinerbaum-Länge



Halber Umfang des umschließenden Rechtecks

$$L_{\text{HPWL}} = w + h$$

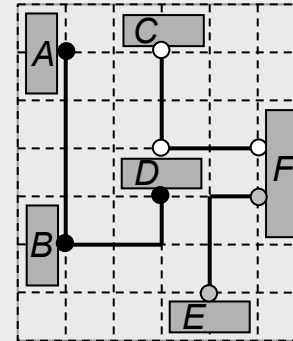
- Zur Einbeziehung von Netzeigenschaften (z.B. kritische Netze) werden **Netzgewichte** eingeführt
- Netzgewicht: numerischer Faktor, der „Wichtigkeit“ eines Netzes ausdrückt  
Beispiel: 1 = Normalnetz, 2 = Netz mit doppelt-wichtiger Bedeutung usw.
- Kostenfunktion: Minimierung der gewichteten Gesamtverbindungslänge  $L(P)$  aller Signalnetze

$$L(P) = \sum_{n \in N} w_n \cdot L_n$$

wobei  $L_n$  die geschätzte Länge des Netzes  $n$ ,  $w_n$  das Gewicht (Weight) des Netzes  $n$  und  $N$  die Menge aller Netze sind.

## 4.2.1 Optimierungsziel: Gewichtete Gesamtverbindungs­länge – Beispiel

Netze	Gewicht
$N_1 = (A, B, D_1)$	$w_1 = 2$
$N_2 = (C, D_2, F_1)$	$w_2 = 4$
$N_3 = (F_2, E)$	$w_3 = 1$



$$L(P) = \sum_{n \in N} w_n \cdot L_n = 2 \cdot 7 + 4 \cdot 4 + 1 \cdot 3 = 33$$

## 4.2.2 Optimierungsziel: Anzahl der geschnittenen Netze

- Die Anzahl der Netze, welche bei einer bestimmten Platzierung  $P$  von der vertikalen Schnittlinie  $x_i$  geschnitten werden, sei mit  $\psi_P(x_i)$  gekennzeichnet.
- Analog ist  $\psi_P(y_j)$  die Menge von Netzen, die eine horizontale Schnittlinie  $y_j$  schneiden.
- Kostenfunktion: Minimierung von 
$$L(P) = \sum_i \psi_P(x_i) + \sum_j \psi_P(y_j)$$

Annahme: Gittereinheit = 1 Längeneinheit

## 4.2.2 Optimierungsziel: Anzahl der geschnittenen Netze – Beispiel

Netze

$$N_1 = (A, B, D_1)$$

$$N_2 = (C, D_2, F_1)$$

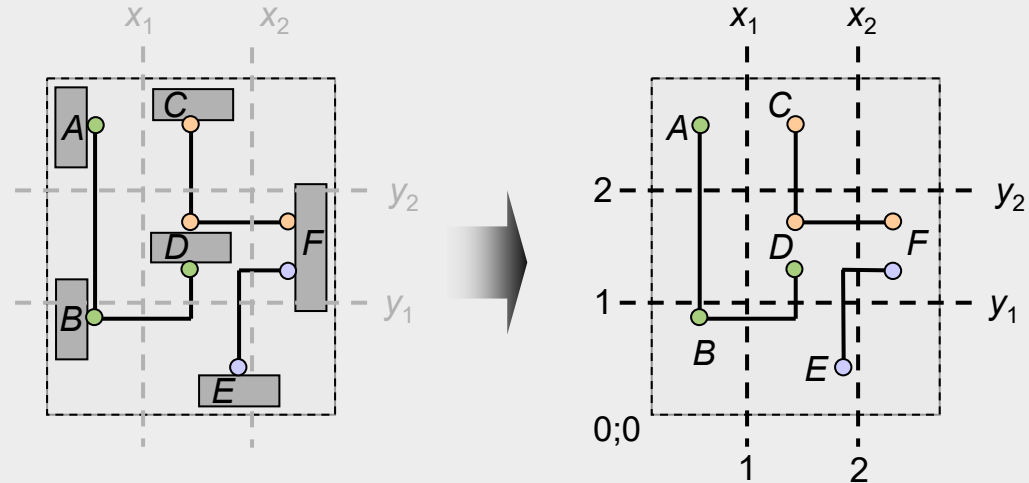
$$N_3 = (F_2, E)$$

$$\psi_P(x_1) = 1$$

$$\psi_P(x_2) = 2$$

$$\psi_P(y_1) = 3$$

$$\psi_P(y_2) = 2$$



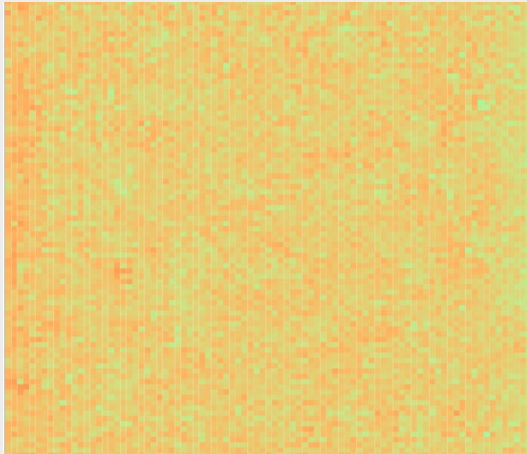
Zielfunktion (Gesamtverbindungslänge):

$$L(P) = \sum_i \psi_P(x_i) + \sum_j \psi_P(y_j) = \psi_P(x_1) + \psi_P(x_2) + \psi_P(y_1) + \psi_P(y_2) = 1 + 2 + 3 + 2 = 8$$

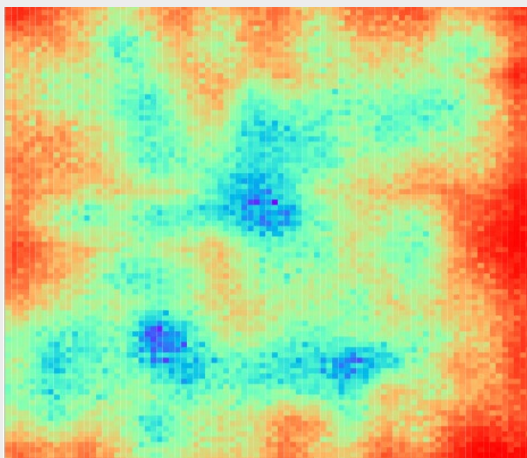
## 4.2.3 Optimierungsziel: Lokale Verdrahtungsdichte

### 1. Platzierung (Zellen)

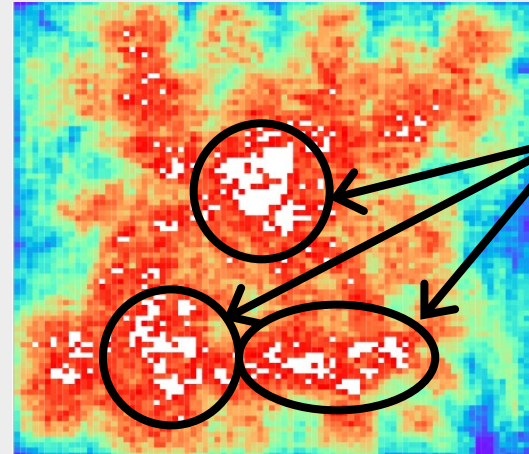
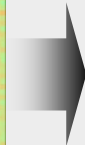
Vernachlässigung der Verdrahtbarkeit



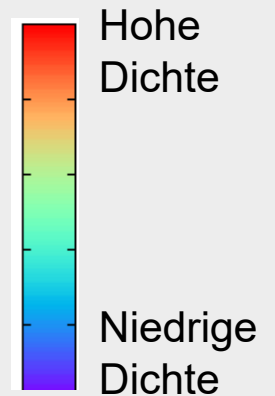
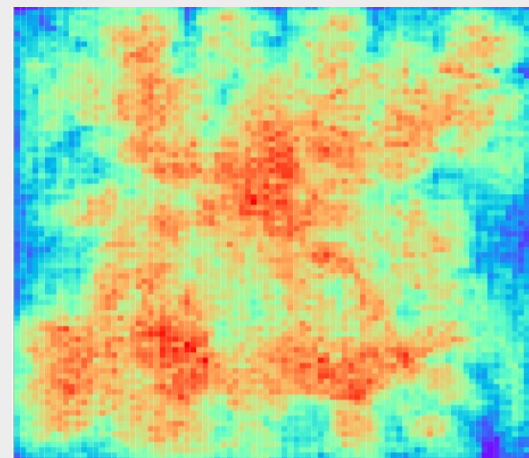
Berücksichtigung der Verdrahtbarkeit



### 2. Verdrahtung (Verbindungen)

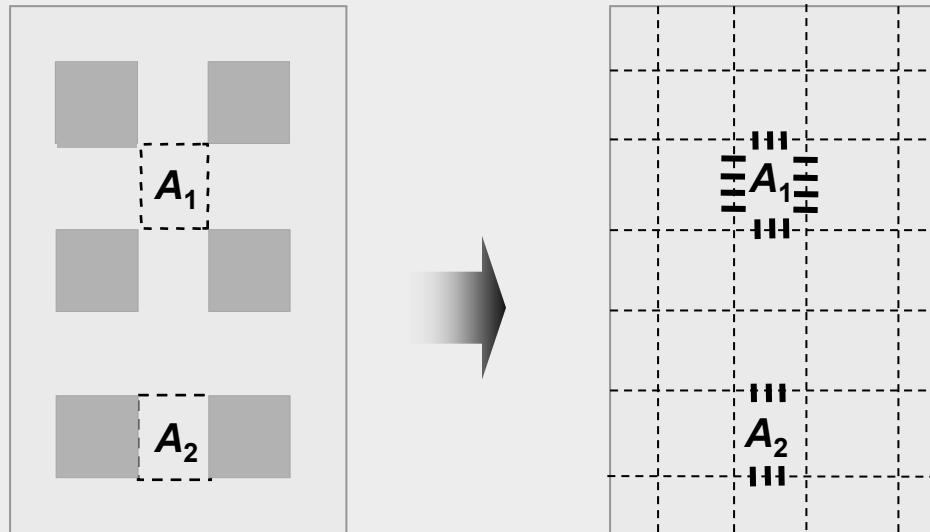


Congestions  
→ nicht verdrahtbar





## 4.2.3 Optimierungsziel: Lokale Verdrahtungsdichte



## 4.2.3 Optimierungsziel: Lokale Verdrahtungsdichte

- Für jede Platzierung  $P$  lässt sich die Netzanzahl, die eine bestimmte Kante durchqueren, abschätzen. Dabei repräsentieren
  - $e_i$  die Kante des jeweiligen Verdrahtungsbereiches,
  - $\eta_P(e_i)$  die geschätzte, sie durchquerende Netzanzahl, und
  - $\phi_P(e_i)$  die Verdrahtungskapazität der Kante  $e_i$ .
- Damit lässt sich die **lokale Verdrahtungsdichte** der Kante  $e_i$  definieren als

$$d_P = \frac{\eta_P(e_i)}{\phi_P(e_i)}$$

- Die Verdrahtbarkeit über die Kante  $e_i$  ist nur gegeben, wenn  $d_P \leq 1$

- Durch Betrachtung der maximalen lokalen Verdrahtungsdichte über alle Kanten lässt sich die Verdrahtbarkeit einer Platzierung  $P$  mittels der **Dichtefunktion  $D(P)$**  definieren als

$$D(P) = \max_i [d_P(e_i)]$$

- Damit ist  $D(P)$  während der Platzierung zu minimieren.

## 4.2.3 Optimierungsziel: Lokale Verdrahtungsdichte – Beispiel

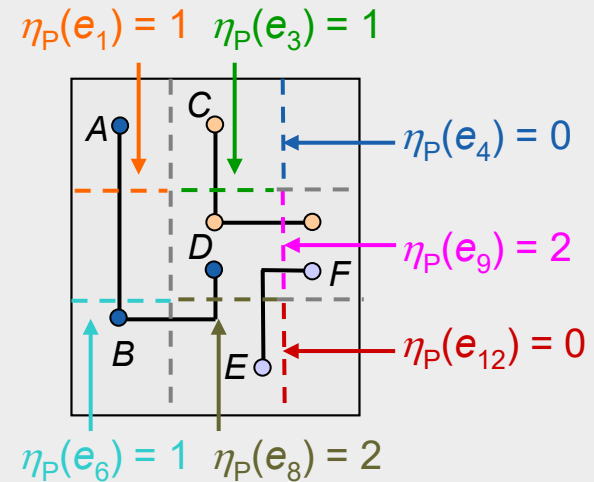
Netze

$$N_1 = (A, B, D_1)$$

$$N_2 = (C, D_2, F_1)$$

$$N_3 = (F_2, E)$$

Jede Kante besitzt eine Kapazität  $\phi_P(e_i) = 3$ .



Maximum  $\eta_P(e_i) = 2$ ,  $\phi_P(e_i) = 3$ , womit  $D(P) = 2/3$ , d.h.  $D(P) \leq 1$ , die Platzierung  $P$  also einfach zu verdrahten sein sollte.

## 4.2.4 Optimierungsziel: Signalverzögerungen

- Oft Berücksichtigung von maximal möglichen Signalverzögerungen während Layouterzeugung notwendig
- Netze werden vor der Layouterzeugung mit maximaler Verzögerungszeit gekennzeichnet, diese Werte werden während Platzierung (und Verdrahtung) mittels Wichtungsfaktoren in der Zielfunktion berücksichtigt
- Nach erfolgter Platzierung ermittelt man entweder unter Nutzung eines Timing-Analysis-Tools oder durch Anwendung einfacher Berechnungsmethoden, wie z.B. dem Elmore-Delay, die Signalverzögerungen der Netze, um sie mit den maximal erlaubten zu vergleichen.

4.1 Einführung

4.2 Optimierungsziele

4.2.1 Gewichtete Gesamtverbindungslänge

4.2.2 Maximale Schnittanzahl

4.2.3 Lokale Verdrahtungsdichte

4.2.4 Signalverzögerungen

**→** 4.3 Platzierungsalgorithmen

4.3.1 Min-Cut-Platzierung

4.3.2 Min-Cut-Platzierung mit Anschlussfestlegung

4.3.3 Quadratische Platzierung

4.3.4 Kräfteplatzierung mittels ZFT-Position

4.3.5 Simulated Annealing

4.3.6 Weitere Platzierungsalgorithmen

4.4 Aktuelle Platzierungswerkzeuge

- **Partitionierende Algorithmen (rekursive Algorithmen):**
  - Optimierung der Platzierungsanordnung mittels rekursiver und dabei immer feinerer Partitionierung der Netzliste und des Platziergebiets
  - Nutzung von Graphenpartitionierern (KL, FM)
  - Beispiel: Min-Cut-Platzierung
- **Analytische Vorgehensweisen:**
  - Nutzung von mathematischen Methoden (z.B. lineare Gleichungssysteme) zur Abbildung und Optimierung des Platzierungsproblems
  - Beispiel: Quadratische Platzierung
- **Stochastische Algorithmen:**
  - Mit Hilfe von stochastischen Methoden wird das Minimum einer beliebigen Kostenfunktion gesucht
  - Einbeziehung von Zufallsentscheidungen, womit bei gleicher Aufgabenstellung unterschiedliche Lösungen erzeugt werden
  - Beispiel: Simulated Annealing

## 4.3 Platzierungsalgorithmen

Partitionierend

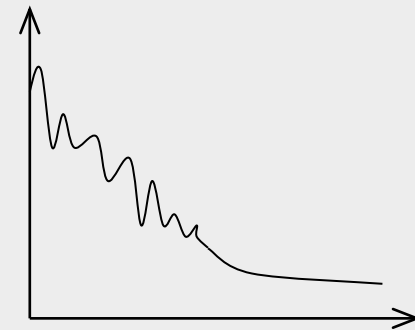
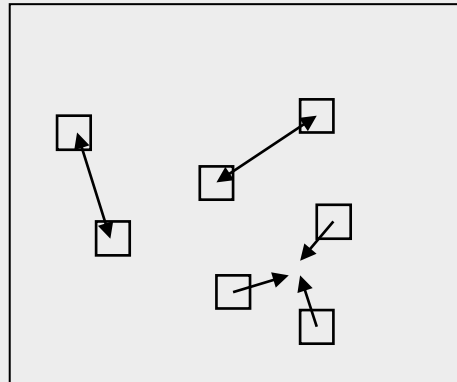
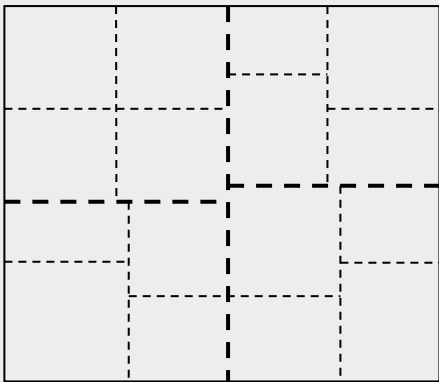
Analytisch

Stochastisch

Min-Cut-Platzierung

Quadratische Platzierung

Platzierung mit SA

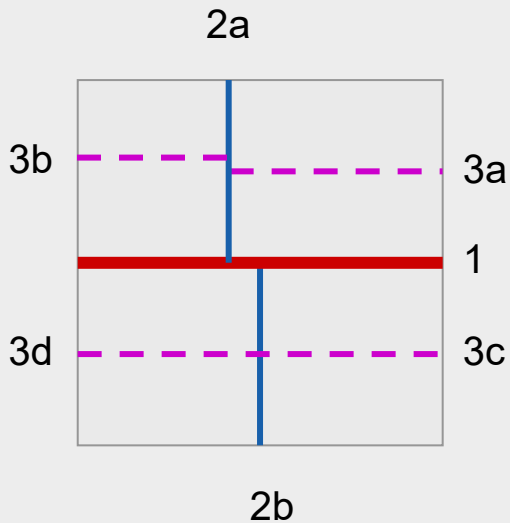




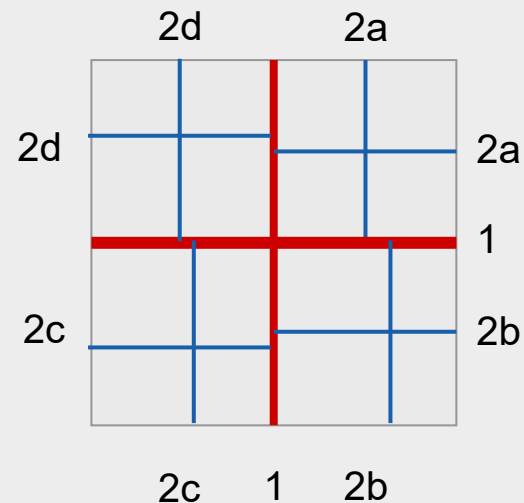
## 4.3.1 Min-Cut-Platzierung

- Platzierungsfläche sequentiell mit Schnittlinien durchzogen, bis die Schnittflächen so klein sind, dass sie nur noch wenige/eine Zelle(n) einschließen
- Bei jedem Schnitt werden die Zellen z.B. so auf die beiden entstehenden Teilflächen aufgeteilt, dass am Ende die Anzahl der die Schnittlinien  $c_r$  kreuzenden Netze  $\psi_P(c_r)$  minimiert ist
- Algorithmen zur Minimierung von  $\psi_P(c_r)$  sind oft der Kernighan-Lin-Algorithmus (KL-Algorithmus) sowie der Fiduccia-Mattheyses-Algorithmus (FM-Algorithmus)

## Bisection-Platzierung (jeweils zwei Partitionen)



## Quadrisection-Platzierung (simultane horizontale und vertikale Schnitte, d.h. jeweils vier Partitionen)

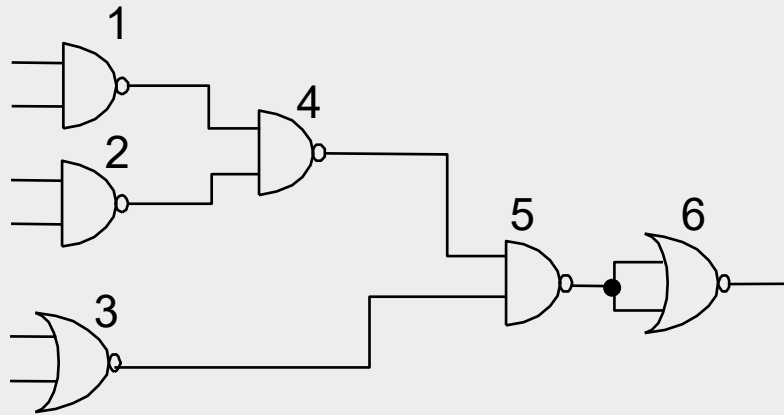


### Min-Cut-Algorithmus

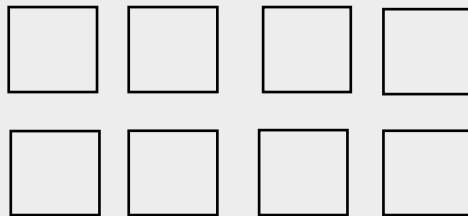
1. Aufteilung der Layoutfläche in zwei Teilflächen mit senkrechter oder horizontaler Schnittrichtung.
2. Anwendung eines geeigneten Algorithmus, z.B. des KL- oder FM-Algorithmus, zur optimierten Verteilung der Zellen auf die beiden Teilflächen.
3. Aufteilung in neue Teilflächen und jeweils Initialzuordnung der Zellen auf diese. Alternierender Wechsel zwischen senkrechter und horizontaler Schnittrichtung.
4. ENDE, falls jede Teilfläche genau eine Zelle enthält, sonst weiter mit Schritt 2.

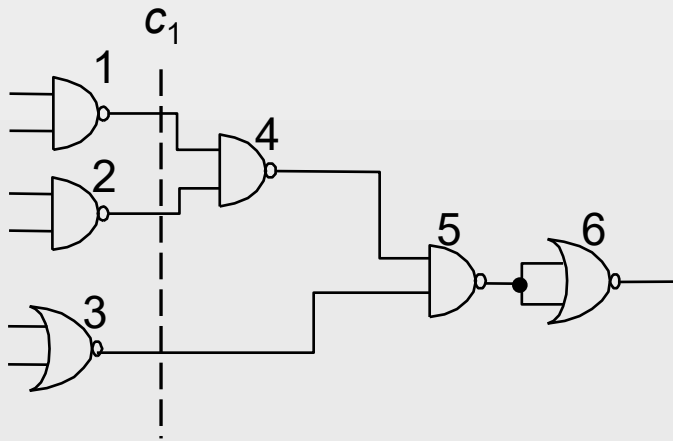
## 4.3.1 Min-Cut-Platzierung: Beispiel

Gegeben:

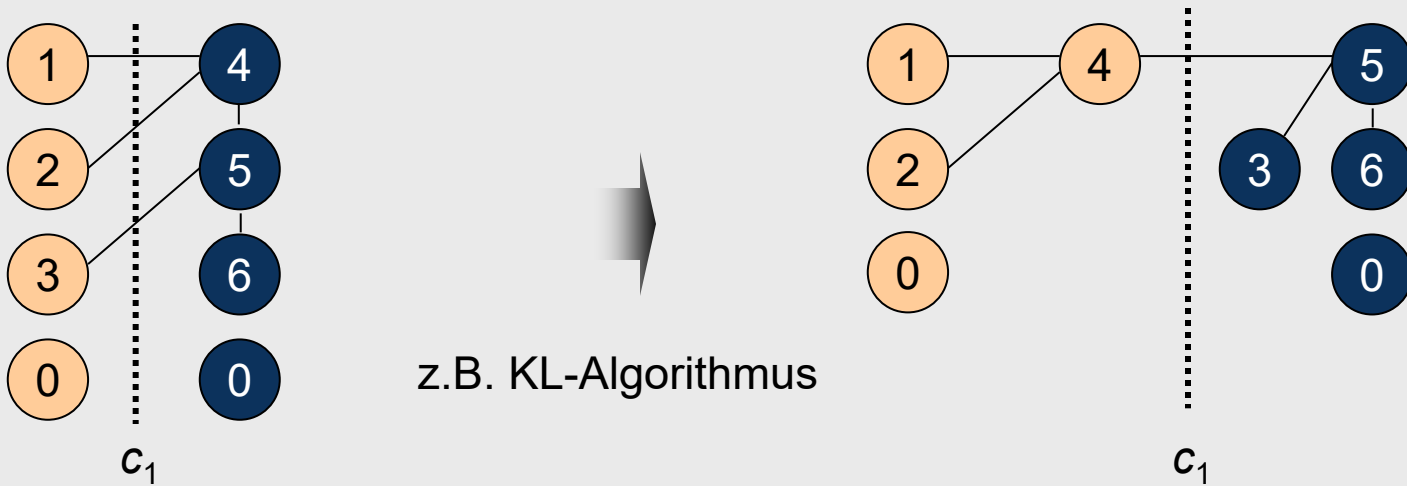


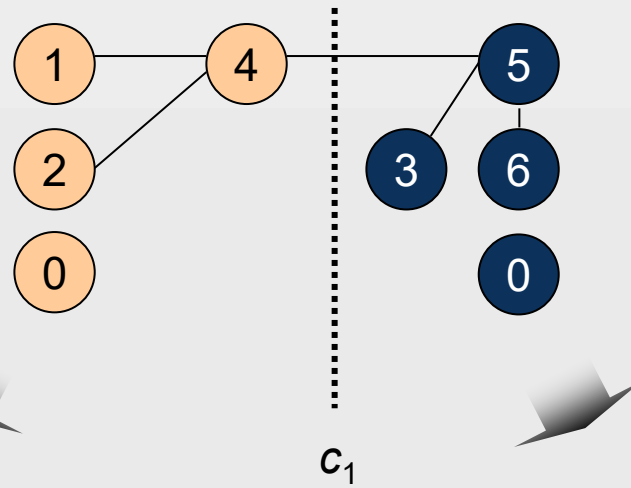
Gesucht: 4 x 2 Platzierung mit minimaler Netzlänge





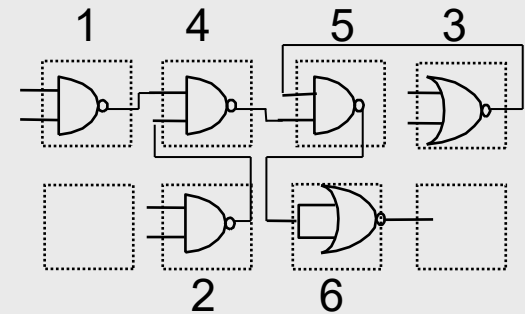
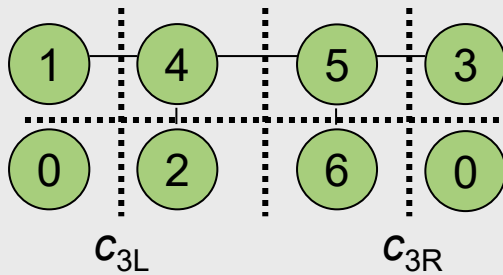
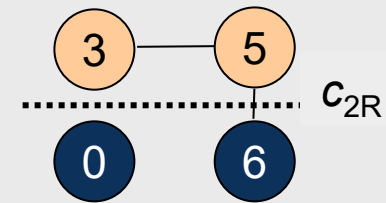
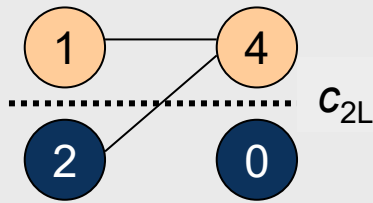
Vertikaler Initialschnitt  $c_1$ :  $L=\{1,2,3\}$ ,  $R=\{4,5,6\}$





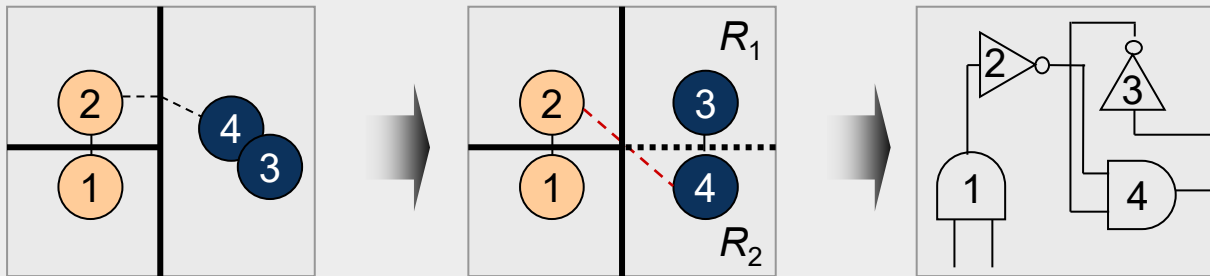
Horizontaler Schnitt  $c_{2L}$ :  $T=\{1,4\}$ ,  $B=\{2,0\}$

Horizontaler Schnitt  $c_{2R}$ :  $T=\{3,5\}$ ,  $B=\{6,0\}$

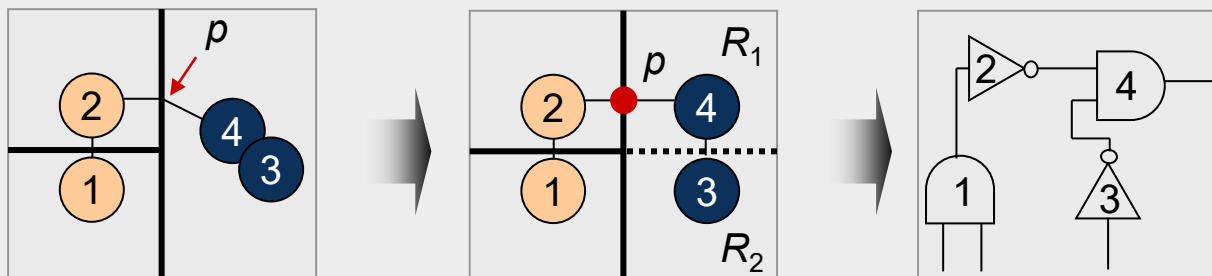


- Vorteile:
  - Sehr schnell
  - Kostenfunktion kann beliebig erweitert werden, d.h. auch für Timing-driven Placement anwendbar
  - Von Natur aus hierarchisch, daher für große Schaltungen nutzbar
- Nachteile:
  - Viele Verschiebungen ohne Auswirkungen
  - Oft Zufallsfaktoren eingeschlossen, daher nicht immer deterministisch
  - Unterhalb bestimmter Partitionsgröße andere Ansatz zur Platzierung sinnvoll
- ➔ Nur sequentielle Optimierung, d.h. die Optimierung bezieht sich immer nur auf die Zuordnung zur jeweils betrachteten Schnittlinie

## 4.3.2 Min-Cut-Platzierung mit Anschlussfestlegung



Mit Anschlussfestlegung (Terminal Propagation)



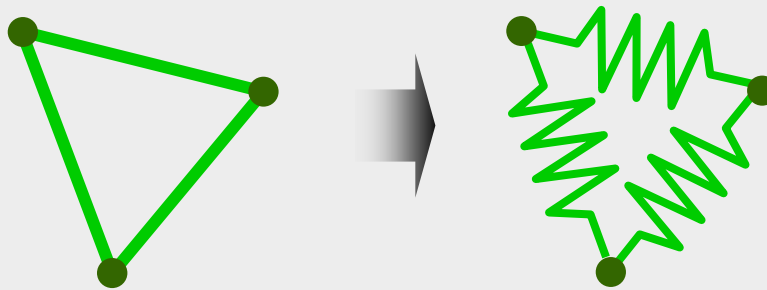


### 4.3.3 Quadratische Platzierung

- Euklidische Verbindungslänge geht quadratisch in Kostenfunktion ein:

$$L(P) = \frac{1}{2} \sum_{i,j=1}^n c_{ij} \left( (x_i - x_j)^2 + (y_i - y_j)^2 \right)$$

- Netze dazu in Zweipunkt-Verbindungen zerlegt, Kostenfunktion ist die Summe der gewichteten quadratischen Abstände, Minimierung dieser Summe
- Analogie: Federmodell



### 4.3.3 Quadratische Platzierung

- Euklidische Verbindungslänge geht quadratisch in Kostenfunktion ein:

$$L(P) = \frac{1}{2} \sum_{i,j=1}^n c_{ij} \left( (x_i - x_j)^2 + (y_i - y_j)^2 \right)$$

- Netze dazu in Zweipunkt-Verbindungen zerlegt, Kostenfunktion ist die Summe der gewichteten quadratischen Abstände, Minimierung dieser Summe
- Analogie: Federmodell
  - jede quadratische Zweipunkt-Länge entspricht Energie einer Feder zwischen beiden Punkten (Energie einer Feder ist proportional zum Quadrat ihrer Auslenkung)
  - quadratische Kostenfunktion verkörpert Gesamtenergie des Federsystems; deren Ableitung ist die Gesamtkraft des Systems
  - System sucht Zustand minimaler Energie, also minimale Summe der Abstandsquadrate
  - damit Zellen im Kräftegleichgewicht hinsichtlich der die Verdrahtung repräsentierenden Kräfte

### 4.3.3 Quadratische Platzierung

- Euklidische Verbindungslänge geht quadratisch in Kostenfunktion ein:

$$L(P) = \frac{1}{2} \sum_{i,j=1}^n c_{ij} \left( (x_i - x_j)^2 + (y_i - y_j)^2 \right)$$

- Vorgehensweise:

- Vektor-/Matrixschreibweise:  $L(P) = \frac{1}{2} [X^T E X + Y^T E Y] - X^T K_x - Y^T K_y + k$

- mit  $X^T$  und  $Y^T$  Vektoren der Dimension  $n$  der  $x$ -/ $y$ -Koordinaten der  $n$  Zellen,  $E$  Gesamtkostenmatrix,  $K_x$  und  $K_y$  Vektoren mit  $K_x [i]$  bzw.  $K_y [i] =$  Summe der  $x$ -/ $y$ -Koordinaten der mit der Zelle  $i$  verbundenen, nicht verschiebbaren Zellen/Außenanschlüsse, sowie eine Konstante  $k$

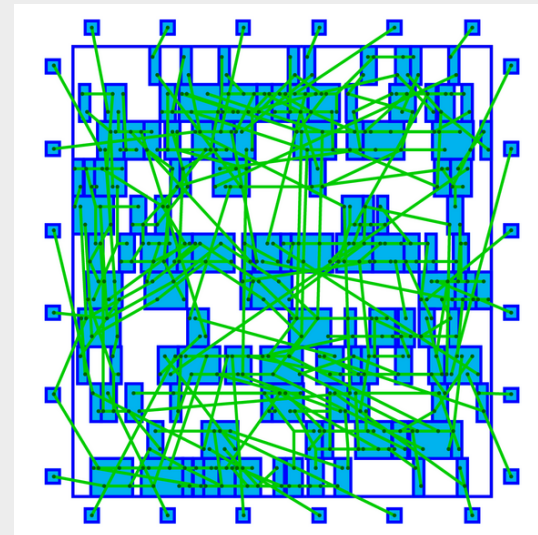
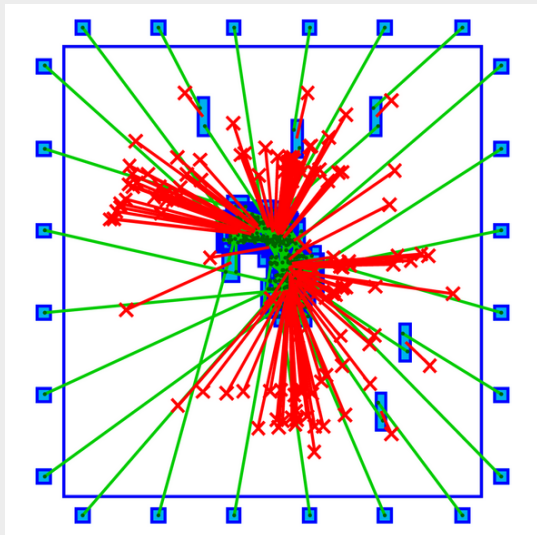
- Globales Minimum und damit platzierungsoptimale  $x$ - und  $y$ -Koordinaten der Zellen lassen sich durch partielle Ableitung von  $L(P)$  bestimmen:

$$\frac{\partial L(P)}{\partial X} = EX - K_x = 0 \quad \text{und} \quad \frac{\partial L(P)}{\partial Y} = EY - K_y = 0$$

- Ergebnis: viele Zellenüberlappungen

### 4.3.3 Quadratische Platzierung

- Quadratische Platzierung lässt sich weiter unterteilen, je nachdem, wie die Zellenüberlappungen beseitigt/vermieden werden



### 4.3.3 Quadratische Platzierung

- Quadratische Platzierung lässt sich weiter unterteilen, je nachdem, wie die Zellenüberlappungen beseitigt/vermieden werden



Kraftbasierte quadratische Platzierung

Überlappungsfreiheit:  
zusätzliche Kräfte



Quadratische Platzierung mit Schwerpunktsnebenbedingungen

Überlappungsfreiheit:  
Verfeinerung von Schwerpunktsnebenbedingungen (führt zu „Quadratic Programming“) und damit rekursive Zerteilung in Platziergebiete

- Vorteile:
  - Schnelle analytische Lösung
  - Auch für große Problemgrößen geeignet
- Nachteile:
  - Pads notwendig (liefert triviale Lösung ohne Pads)
  - Hierarchischer Ansatz schwierig zu realisieren

### Kräfteplatzierung

- Bei der Kräfteplatzierung werden die zu platzierenden Zellen analog einem mechanischen System aus mit Federn verbundenen Körpern betrachtet.
- Dabei üben miteinander verbundenen Körper (Zellen) eine Anziehungskraft zueinander aus, wobei diese Kraft direkt proportional zur Entfernung zwischen den Körpern ist.
- Ergebnis: Kräftegleichgewicht bzw. energieminimaler Zustand

### Kräfteplatzierung und quadratische Platzierung

- Energie einer Feder ist proportional zum Quadrat ihrer Auslenkung
- Ermittlung der energieminimalen Positionen von Zellen, die mit Federn verbunden sind, ist daher identisch zur Minimierung der Summe der Quadrate der euklidischen Abstände („Quadratische Platzierung“)
- Unterscheidung in der Zellenbetrachtung: Während bei der quadratischen Platzierung die Zellenplatzierungen durch gleichzeitige Berücksichtigung aller Zellen in einem aus der quadratischen Kostenfunktion abgeleiteten Gleichungssystem ermittelt werden (Überlappungen!), erfolgt die Zellenplatzierung bei der Kräfteplatzierung mittels ZFT-Position durch sequentielle Zellenverschiebungen



## 4.3.4 Kräfteplatzierung mittels ZFT-Position

- Angenommen, eine Zelle  $a$  ist mit einer Zelle  $b$  verbunden. Die Anziehungskraft zwischen beiden Zellen ergibt sich aus dem Produkt von Wichtung und Länge der Verbindung

$$\vec{F} = w_{ab} \cdot \vec{d}_{ab}$$

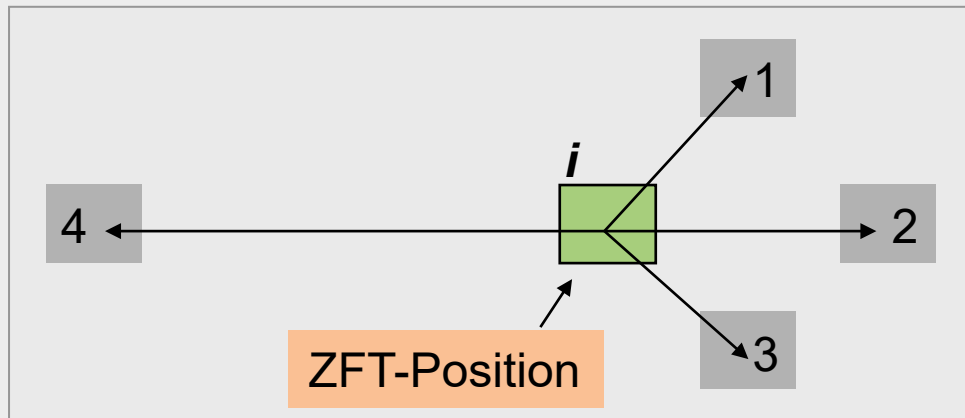
(bzw.  $F = w_{ab} \cdot d_{ab}$ , da  $\vec{F}$  parallel  $\vec{d}_{ab}$  ist).

- Analog gilt für eine Zelle  $i$ , die mit mehreren Zellen  $1 \dots j$  verbunden ist

$$\vec{F}_i = \sum_j (w_{ij} \cdot \vec{d}_{ij})$$

wobei  $w_{ij}$  die Wichtung der Verbindung und  $d_{ij}$  deren Länge sind.

### Zero-Force-Target (ZFT)-Position einer Zelle $i$



$$\vec{F}_i = w_{i1} \cdot \vec{d}_{i1} + w_{i2} \cdot \vec{d}_{i2} + w_{i3} \cdot \vec{d}_{i3} + w_{i4} \cdot \vec{d}_{i4} \rightarrow \min$$

## 4.3.4 Kräfteplatzierung mittels ZFT-Position

Anwendung bei der Platzierung:

- Für jede Zelle werden die auf sie wirkenden Kräfte berechnet, um diese Zelle  $i$  dann in ihrer jeweiligen ZFT-Position  $(x_i^0, y_i^0)$  zu platzieren.
- Diese lässt sich ermitteln, indem die in  $x$ - und in  $y$ -Richtung wirkenden Kräfte zu Null gesetzt werden:

$$\sum_j w_{ij} \cdot (x_j - x_i^0) = 0 \quad \sum_j w_{ij} \cdot (y_j - y_i^0) = 0$$

- Die Umstellung dieser Gleichungen nach  $x_i^0$  und  $y_i^0$  liefert

$$x_i^0 = \frac{\sum_j w_{ij} \cdot x_j}{\sum_j w_{ij}} \quad y_i^0 = \frac{\sum_j w_{ij} \cdot y_j}{\sum_j w_{ij}}$$

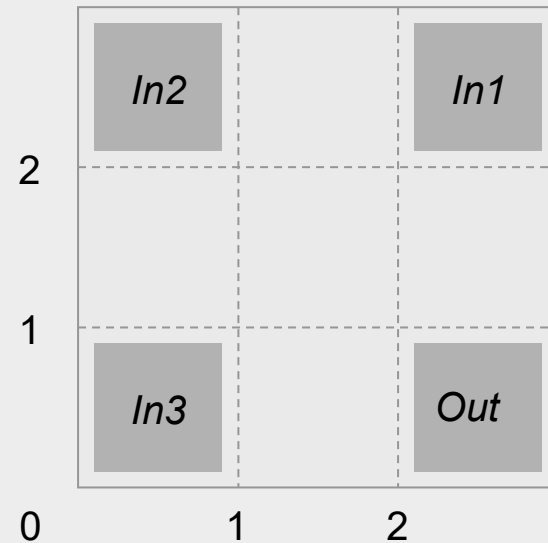
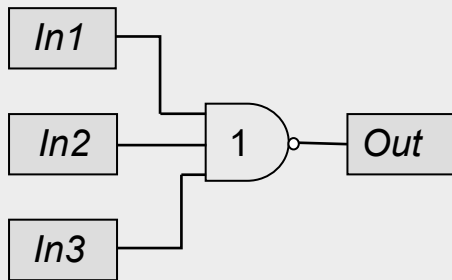
Berechnung der ZFT-Position einer Zelle  $i$ , welche mit den Zellen  $1 \dots j$  verbunden ist

## 4.3.4 Kräfteplatzierung mittels ZFT-Position: Beispiel (ZFT-Position)

Gegeben:

- Schaltung mit einer NAND-Zelle 1 und vier I/O-Padzellen auf einem 3 x 3 Raster
- Positionen der Padzellen:  $In1$  (2;2),  $In2$  (0;2),  $In3$  (0;0),  $Out$  (2;0)
- Wichtigkeit der einzelnen Verbindungen:  $w_{1In1} = 8$ ,  $w_{1In2} = 10$ ,  $w_{1In3} = 2$ ,  $w_{1Out} = 2$

Gesucht: ZFT-Position der Zelle 1 (Rasterposition)



## 4.3.4 Kräfteplatzierung mittels ZFT-Position: Beispiel (ZFT-Position)

Gegeben:

- Positionen der Padzellen:  $In1 (2;2)$ ,  $In2 (0;2)$ ,  $In3 (0;0)$ ,  $Out (2;0)$
- Wichtung der einzelnen Verbindungen:  $w_{1In1} = 8$ ,  $w_{1In2} = 10$ ,  $w_{1In3} = 2$ ,  $w_{1Out} = 2$

Lösung:

$$x_1^0 = \frac{\sum_j w_{1j} \cdot x_j}{\sum_j w_{1j}} = \frac{w_{1In1} \cdot x_{In1} + w_{1In2} \cdot x_{In2} + w_{1In3} \cdot x_{In3} + w_{1Out} \cdot x_{Out}}{w_{1In1} + w_{1In2} + w_{1In3} + w_{1Out}} = \frac{8 \cdot 2 + 10 \cdot 0 + 2 \cdot 0 + 2 \cdot 2}{8 + 10 + 2 + 2} = \frac{20}{22} \approx 0,9$$

$$y_1^0 = \frac{\sum_j w_{1j} \cdot y_j}{\sum_j w_{1j}} = \frac{w_{1In1} \cdot y_{In1} + w_{1In2} \cdot y_{In2} + w_{1In3} \cdot y_{In3} + w_{1Out} \cdot y_{Out}}{w_{1In1} + w_{1In2} + w_{1In3} + w_{1Out}} = \frac{8 \cdot 2 + 10 \cdot 2 + 2 \cdot 0 + 2 \cdot 0}{8 + 10 + 2 + 2} = \frac{36}{22} \approx 1,6.$$

## 4.3.4 Kräfteplatzierung mittels ZFT-Position: Beispiel (ZFT-Position)

Gegeben:

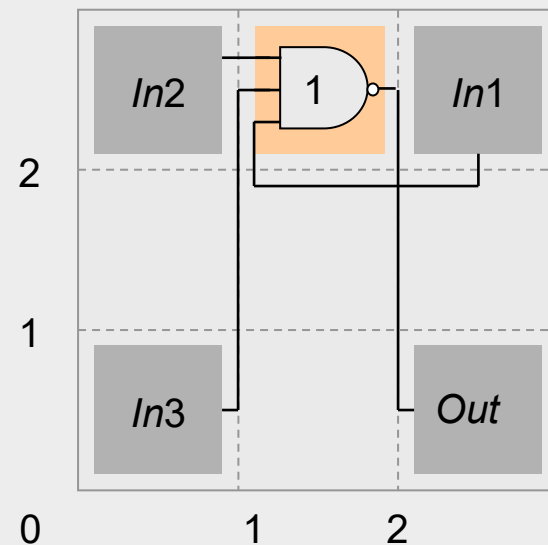
- Positionen der Padzellen:  $In1 (2;2)$ ,  $In2 (0;2)$ ,  $In3 (0;0)$ ,  $Out (2;0)$
- Wichtung der einzelnen Verbindungen:  $w_{1In1} = 8$ ,  $w_{1In2} = 10$ ,  $w_{1In3} = 2$ ,  $w_{1Out} = 2$

Lösung:

$$x_1^0 = \frac{\sum_j w_{1j} \cdot x_j}{\sum_j w_{1j}} = \frac{w_{1In1} \cdot x_{In1} + w_{1In2} \cdot x_{In2} + w_{1In3} \cdot x_{In3} + w_{1Out} \cdot x_{Out}}{w_{1In1} + w_{1In2} + w_{1In3} + w_{1Out}} = \frac{8 \cdot 2 + 10 \cdot 0 + 2 \cdot 0 + 2 \cdot 2}{8 + 10 + 2 + 2} = \frac{20}{22} \approx 0,9$$

$$y_1^0 = \frac{\sum_j w_{1j} \cdot y_j}{\sum_j w_{1j}} = \frac{w_{1In1} \cdot y_{In1} + w_{1In2} \cdot y_{In2} + w_{1In3} \cdot y_{In3} + w_{1Out} \cdot y_{Out}}{w_{1In1} + w_{1In2} + w_{1In3} + w_{1Out}} = \frac{8 \cdot 2 + 10 \cdot 2 + 2 \cdot 0 + 2 \cdot 0}{8 + 10 + 2 + 2} = \frac{36}{22} \approx 1,6.$$

Damit entspricht die Rasterposition (1;2) der ZFT-Position der Zelle 1.



### Algorithmus zur Kräfteplatzierung mittels ZFT-Position

1. Ermitteln einer willkürlichen Anfangsplatzierung
2. Auswahl einer Zelle (z.B. diejenige mit maximalem Verbindungsgrad) und Berechnen ihrer ZFT-Position
  - wenn ZFT-Position frei, dann Verschiebung zu dieser
  - wenn ZFT-Position belegt, Anwendung einer der nachfolgenden Belegungsoptionen
3. Weiter mit Schritt 2 und neuer Zelle, bis Abbruchkriterium erreicht ist.

### Optionen bei bereits erfolgter Belegung einer ZFT-Position

( $p$ : zu verschiebende Zelle,  $q$ : Zelle in der ZFT-Position)

- Verschieben von  $p$  zu einer freien Zellenposition möglichst nahe zu  $q$ .
- Berechnen der Kostenveränderung bei Austausch von  $p$  mit  $q$ . Sollten sich die Gesamtkosten, wie z.B. die gewichtete Gesamtverbindungslänge  $L(P)$  verringern, werden  $p$  und  $q$  in ihren Positionen vertauscht.



## 4.3.4 Kräfteplatzierung mittels ZFT-Position: Beispiel

Gegeben:

Netze

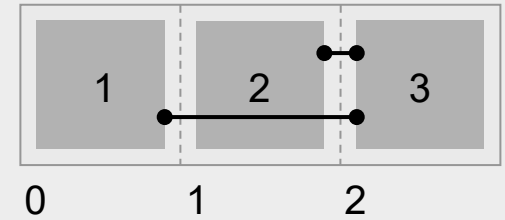
$$N_1 = (1, 3)$$

$$N_2 = (2, 3)$$

Gewicht

$$w_1 = 2$$

$$w_2 = 1$$



## 4.3.4 Kräfteplatzierung mittels ZFT-Position: Beispiel

Gegeben:

Netze

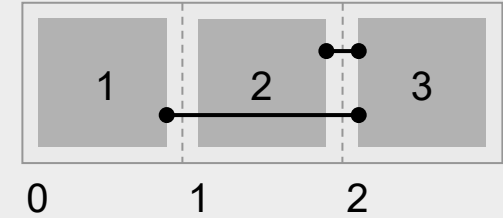
$$N_1 = (1, 3)$$

$$N_2 = (2, 3)$$

Gewicht

$$w_1 = 2$$

$$w_2 = 1$$



Zu ver-  
schiebende  
Zelle  $p$

ZFT-Position  
von Zelle  $p$

Zelle  $q$

$L(P)$  vor  
Vertauschung

$L(P)$  /Anordnung  
nach Vertauschung

3

$$x_3^0 = \frac{\sum_j w_{ij} \cdot x_j}{\sum_j w_{ij}} = \frac{2 \cdot 0 + 1 \cdot 1}{2 + 1} \approx 0$$

1

$$L(P) = 5$$

## 4.3.4 Kräfteplatzierung mittels ZFT-Position: Beispiel

Gegeben:

Netze

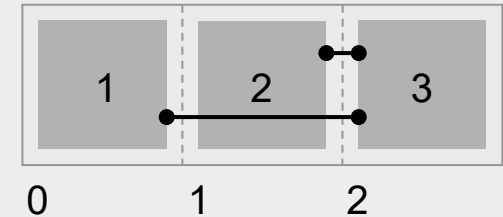
$$N_1 = (1, 3)$$

$$N_2 = (2, 3)$$

Gewicht

$$w_1 = 2$$

$$w_2 = 1$$



Zu ver-  
schiebende  
Zelle  $p$

ZFT-Position  
von Zelle  $p$

Zelle  $q$

$L(P)$  vor  
Vertauschung

$L(P)$  /Anordnung  
nach Vertauschung

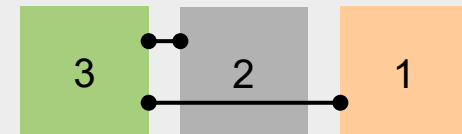
3

$$x_3^0 = \frac{\sum_j w_{ij} \cdot x_j}{\sum_j w_{ij}} = \frac{2 \cdot 0 + 1 \cdot 1}{2 + 1} \approx 0$$

1

$$L(P) = 5$$

$$L(P) = 5$$



Damit keine Vertauschung von 3 und 1.

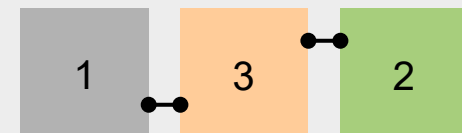
2

$$x_2^0 = \frac{\sum_j w_{ij} \cdot x_j}{\sum_j w_{ij}} = \frac{1 \cdot 2}{1} = 2$$

3

$$L(P) = 5$$

$$L(P) = 3$$

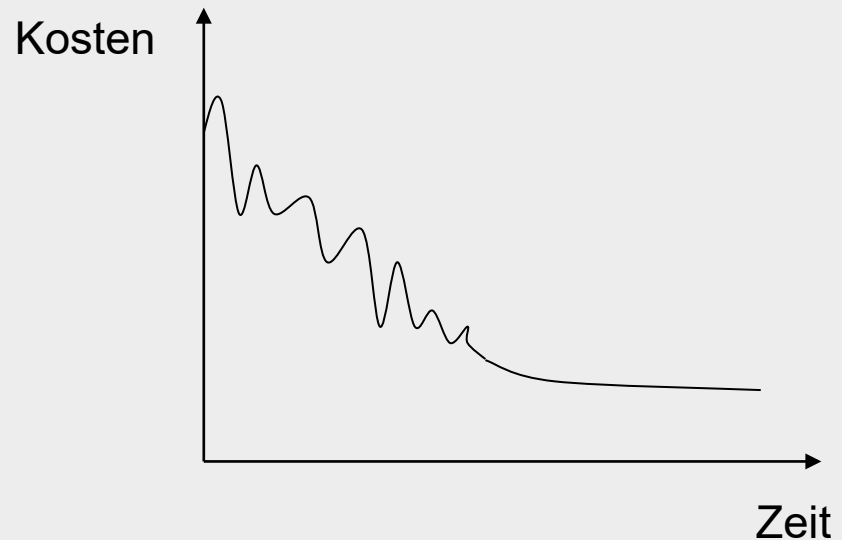
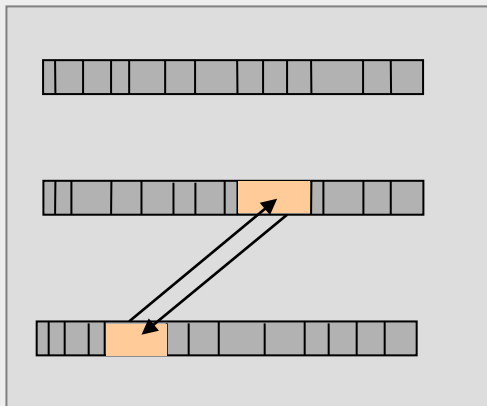


Vertauschung von 2 und 3.

- Vorteile:
  - Einfach zu implementieren
  - Gut geeignet zur Detailplatzierung
- Nachteile:
  - Nicht für große Problemgrößen geeignet (da im Gegensatz zur quadratischen Platzierung immer nur eine Zelle betrachtet wird)
  - Hierarchischer Ansatz schwierig zu realisieren

## 4.3.5 Simulated Annealing

- Analogie zum Abkühlungsprozess von metallischen Schmelzen (energieminimales Atomgitter)
- Modifikation einer Anfangsplatzierung durch Tausch von zufällig ausgewählten Zellen



- Wenn Kosten verbessert werden, wird Tausch ausgeführt
- Bei keiner Kostenverbesserung wird Tausch mit temperaturabhängiger (d.h. abnehmender) Wahrscheinlichkeit ausgeführt

## 4.3.5 Simulated Annealing

### Simulated-Annealing-Platzierungsalgorithmus

**begin**

$T = T_0$

*/\* Anfangstemperatur \*/*

$P = \text{init\_placement}$

*/\* Anfangsplatzierung \*/*

**repeat**

**repeat**

$\text{NewP} = \text{PERTURB}(P)$

$\Delta\text{cost} = \text{COST}(\text{NewP}) - \text{COST}(P)$

**if** ( $\Delta\text{cost} < 0$ ) **then**

*/\* wenn Verbesserung \*/*

$P = \text{NewP}$

*/\* Übernahme neuer Platzierung \*/*

**else**

*/\* ansonsten \*/*

$r = \text{RANDOM}(0,1)$

*/\* Zufallszahl zwischen 0 und 1 \*/*

**if** ( $r < e^{-\frac{\Delta\text{cost}}{T}}$ ) **then**

*/\* bedingte Übernahme \*/*

$P = \text{NewP}$

*/\* der neuen Platzierung \*/*

**until** ( Abbruchkriterium, z.B. Gleichgewicht bei  $T$ , erreicht )

$T = \alpha * T$  */\*  $0 < \alpha < 1$  \*/*

*/\* Temperatur-Reduktion \*/*

**until** ( $T < T_{\min}$ )

**end.**

- Vorteile:
  - Kann globales Optimum finden (bei „genügend“ Zeit)
  - Kann mehrere Optimierungsziele berücksichtigen (Wichtungsfaktoren)
  - Gut geeignet zur Detailplatzierung
- Nachteil:
  - Sehr langsame Lösungskonvergenz

## 4.3.6 Weitere Platzierungsalgorithmen

- Zuordnung zu Einbauplätzen (z.B. bei Gate-Arrays)
- Neuronale Netzwerke
- Evolutionäre Algorithmen
- Timing-driven Placement / Performance-driven Placement



- 4.1 Einführung
- 4.2 Optimierungsziele
  - 4.2.1 Gewichtete Gesamtverbindungslänge
  - 4.2.2 Maximale Schnittanzahl
  - 4.2.3 Lokale Verdrahtungsdichte
  - 4.2.4 Signalverzögerungen
- 4.3 Platzierungsalgorithmen
  - 4.3.1 Min-Cut-Platzierung
  - 4.3.2 Min-Cut-Platzierung mit Anschlussfestlegung
  - 4.3.3 Quadratische Platzierung
  - 4.3.4 Kräfteplatzierung mittels ZFT-Position
  - 4.3.5 Simulated Annealing
  - 4.3.6 Weitere Platzierungsalgorithmen
- 4.4 Aktuelle Platzierungswerkzeuge