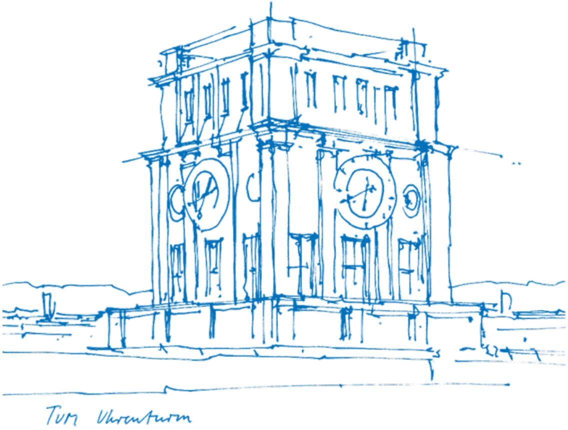
Übung 14: SAT und Physical Design

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

30. Januar 2026



Keine Garantie für die Richtigkeit der Tutorfolien.

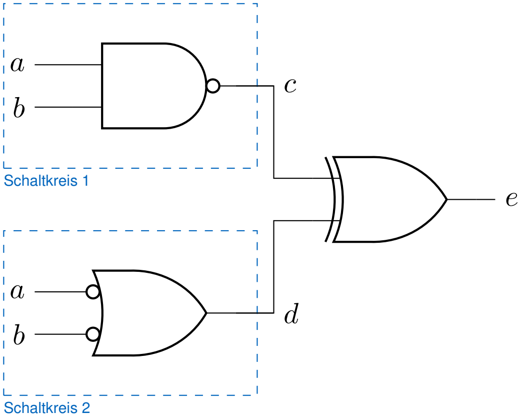
Bei Unklarheiten/Unstimmigkeiten haben VL/ZÜ-Folien recht!

SAT

- „Satisfiability", d.h. Erfüllbarkeit einer boolschen Funktion herausfinden:
  - „Existiert eine Belegung $x_0, \ldots, x_n$ sodass $f(x_0, \ldots, x_n) \equiv 1$?"
  - „Gibt es in der Wahrheitstabelle mindestens eine Zeile mit Ergebnis 1?"
- moderne Solver können sichere Aussage über Erfüllbarkeit bzw. Nicht-Erfüllbarkeit treffen, ohne alle Variablenbelegungen durchzuprobieren
- Übertragung verschiedenster Probleme auf SAT, Formulierung als KNF[1]: OR in den Klauseln, AND zwischen Klauseln

[1] Konjunktive Normalform, CNF

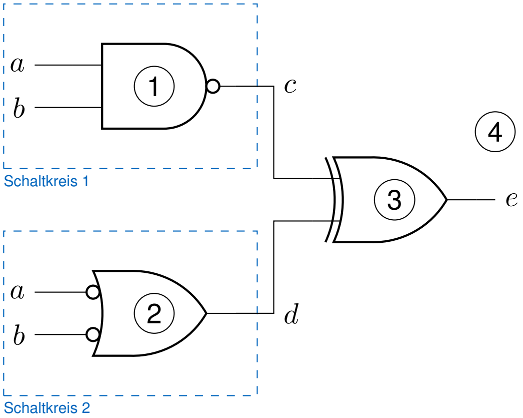
SAT: Schaltkreisäquivalenz



c	d	$e = c \oplus d$
0	0	0
0	1	1
1	0	1
1	1	0

- existiert eine Belegung von a, b sodass $e = 1$, dann sind die beiden Schaltkreise für diese Belegung nicht äquivalent
- eine solche Schaltung heißt Miter
- KNF kann durch Tseitin-Transformation aufgestellt werden

Tseitin-Transformation



1 $\overline{(a \wedge b)} \leftrightarrow c \;\; \wedge$

2 $(\overline{a} \vee \overline{b}) \leftrightarrow d \;\; \wedge$

3 $(c \oplus d) \leftrightarrow e \;\; \wedge$

4 e

nach Umformung zu KNF und Berechnung mittels eines SAT-Solvers erhalten wir `UNSAT`, die Schaltkreise sind also äquivalent[2]

[2] mittels De-Morgan einfach nachprüfbar: $\overline{a \cdot b} \equiv \overline{a} + \overline{b}$

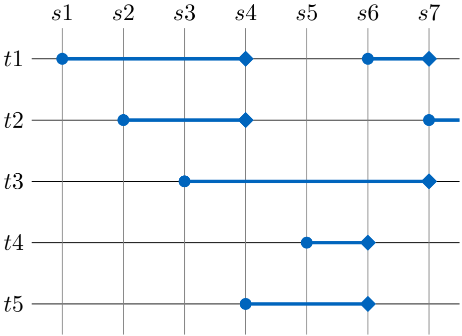
SAT: Registerallokation

```
s1: t1 = 1
s2: t2 = 5
s3: t3 = 7
s4: t5 = t1 + t2
s5: t4 = t3 + t5
s6: t1 = t4 + t5
s7: t2 = t1 + t3
```

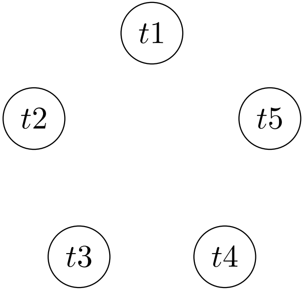
	$s1$	$s2$	$s3$	$s4$	$s5$	$s6$	$s7$
$t1$							
$t2$							
$t3$							
$t4$							
$t5$							

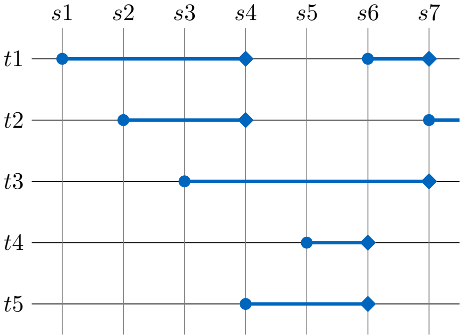
SAT: Registerallokation

```
s1: t1 = 1
s2: t2 = 5
s3: t3 = 7
s4: t5 = t1 + t2
s5: t4 = t3 + t5
s6: t1 = t4 + t5
s7: t2 = t1 + t3
```

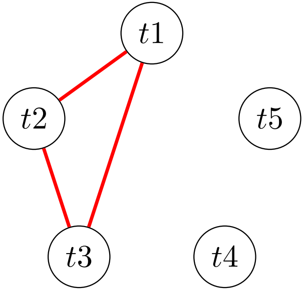


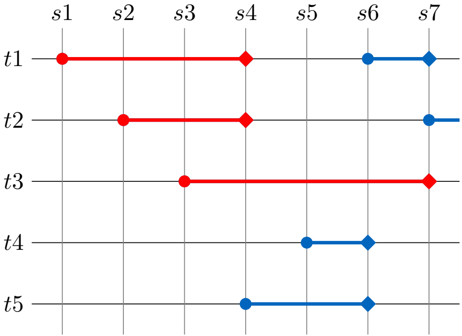
SAT: Registerallokation



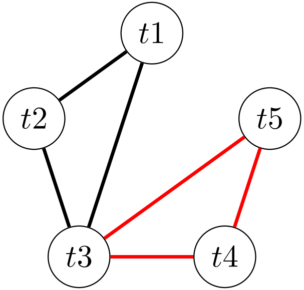


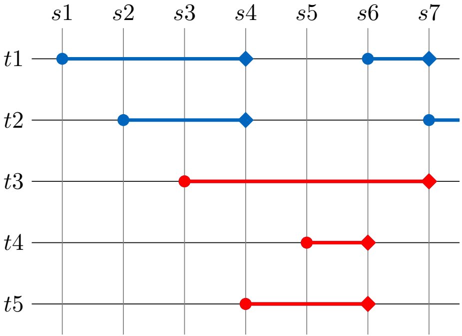
SAT: Registerallokation



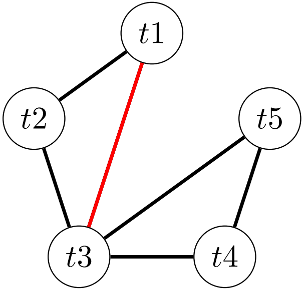


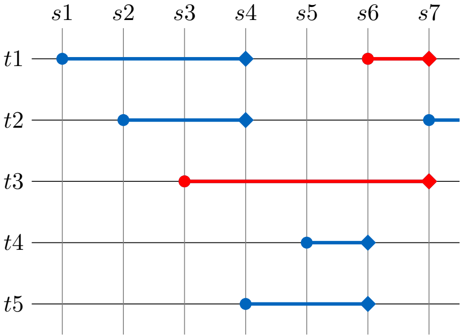
SAT: Registerallokation



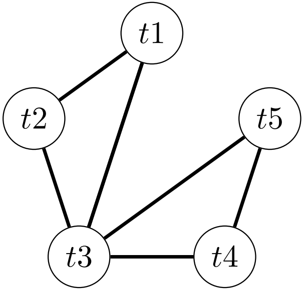


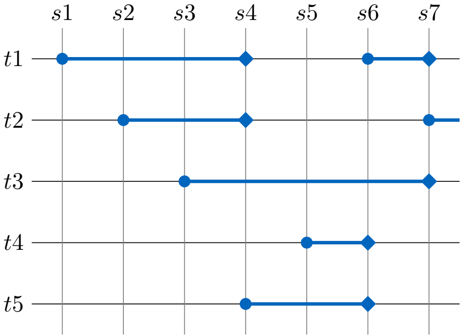
SAT: Registerallokation



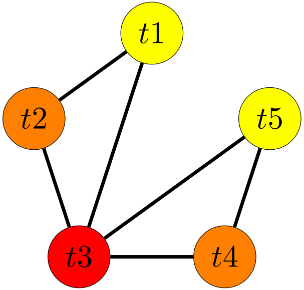


SAT: Registerallokation



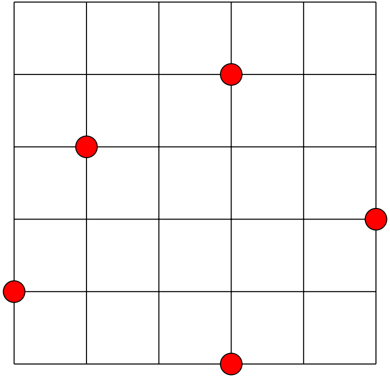


SAT: Registerallokation



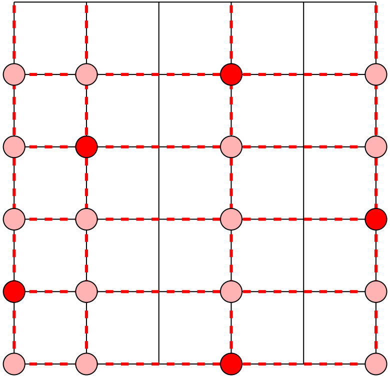
- In Register ● $a0$: $t1, t5$
- In Register ● $a1$: $t2, t4$
- In Register ● $a2$: $t3$

Single-Net Routing



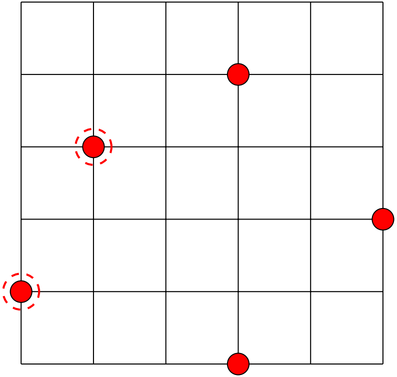
- Ziel: Verbindung von Terminalen mit kürzesten Pfaden
- rektilinear (geradlinig): nur horizontale/vertikale Verbindungen

Single-Net Routing



- Hanan-Punkte: mögliche Steinerknoten (Abzweigungen im Steinerbaum)
- Schnittpunkte von Geraden durch Terminalknoten
- Reduziert Menge an Abzweigungspunkten, die betrachtet werden müssen

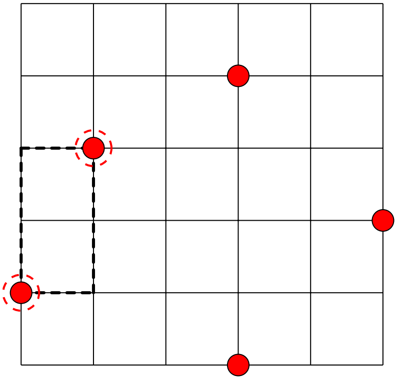
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$

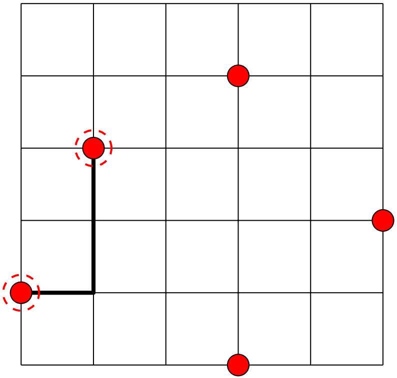
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)

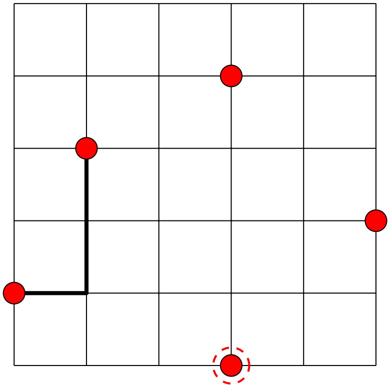
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat

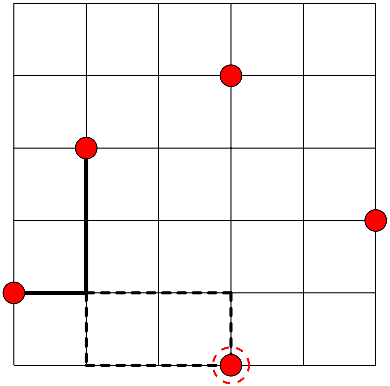
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

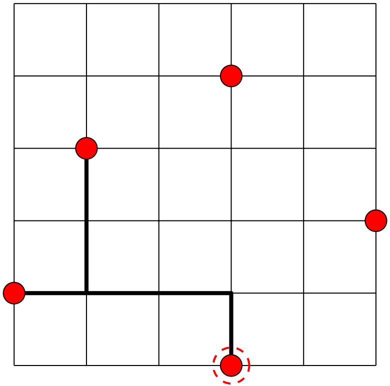
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

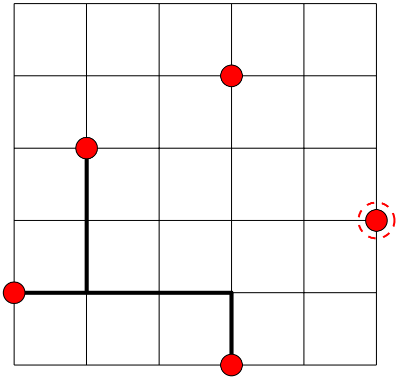
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

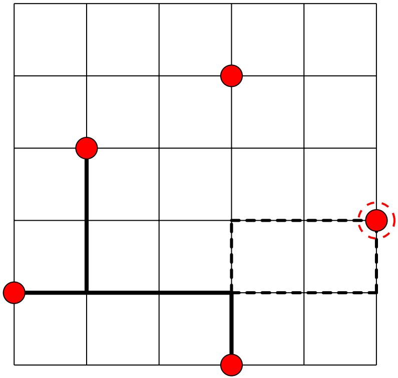
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

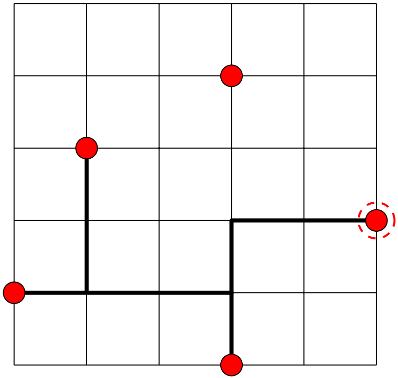
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

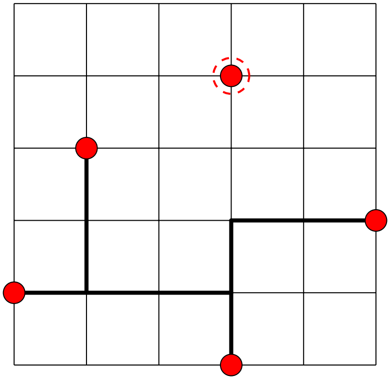
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

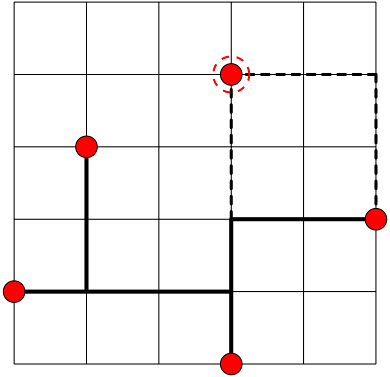
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

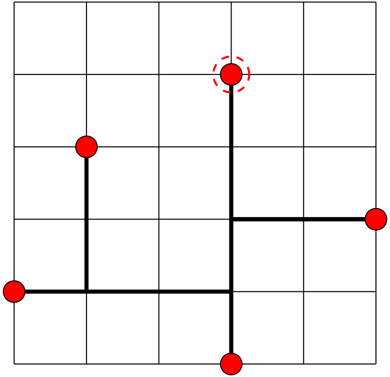
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

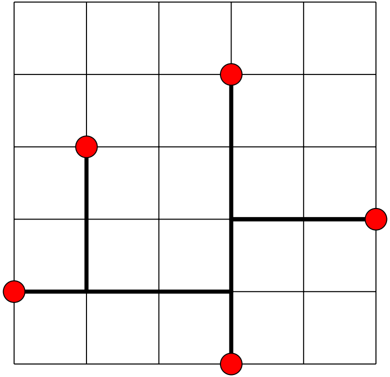
Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

Single-Net Routing



Konstruktion des Steinerbaums:

1. Finde Terminale mit minimaler Manhattan-Distanz $\delta = \Delta x + \Delta y$
2. Konstruiere die kürzesten Verbindungen (bounding box)
3. Wähle die Verbindung, deren Hanan-Punkte den geringsten Abstand zu einem der anderen Terminalknoten hat
4. Füge diese Verbindung dem Steinerbaum hinzu und fahre mit Schritt 2 fort

Fragen?

Die Slides zur Registerallokation wurden von Bjarne Hansen übernommen

Links

- Zulip: „ERA Tutorium – Mi-1600-3“ bzw. „ERA Tutorium – Fr-1500-1“
- ERA-Moodle-Kurs
- ERA-Artemis-Kurs

Übung 14: SAT und Physical Design

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

30. Januar 2026