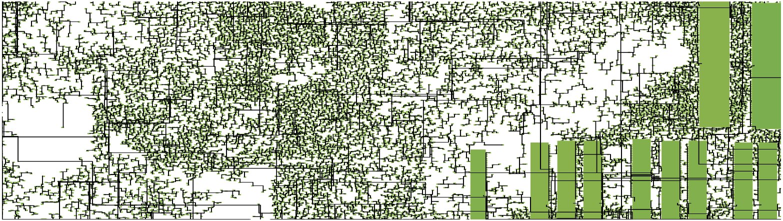
# Steiner Trees in Chip Design

Jens Vygen

Hangzhou, March 2009

# Introduction

- A digital chip contains millions of gates.
- Each gate produces a signal (0 or 1) once every cycle.
- The output signal of a gate is input to other gates.
- For each gate we need a network that distributes the signal from the root (output of this gate) to the given set of sinks.
- In the simplest case the network is a Steiner tree.
- A set of pins that need to be connected is called a net.



# Constraints and objectives

A feasible Steiner tree for a net (=set of pins)

- consists of horizontal and vertical wires on the wiring planes
- and vias connecting wires on different planes
- such that the network of wires, vias and pins is a tree,
- each wire and via has at least a certain minimum width and sufficient distance to blockages and other wires and vias,
- and obeys certain (local) ground rules (for manufacturing).

A Steiner tree is good if it

- consumes little area, avoids congested regions,
- has small electrical capacitance,
- allows for fast signal transmitting from the source to the (critical) sinks,
- can be manufactured well (small yield loss).

# Steiner trees at various design stages

- Everywhere:
  - shortest rectilinear Steiner trees
- Placement:
  - estimates (hypergraph models)
- Timing optimization:
  - RC-optimal trees
  - buffered trees
- Clock tree design:
  - balanced trees
  - clustering sinks with bounds on Steiner tree lengths
- Routing:
  - packing Steiner trees
  - fast search for paths and Steiner trees in huge grids

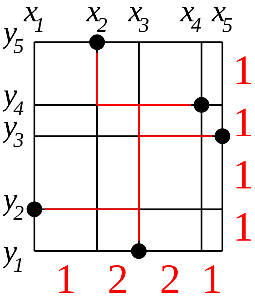
# Characteristics of the instances

- Third dimension very small, can often be neglected
- Number of terminals mostly small, but some very large instances (with millions of terminals)
- Completely blocked regions do not occur often
- Billions of instances must be solved

# Shortest rectilinear Steiner trees

- *NP*-hard (Garey, Johnson [1977])
- approximation scheme (Arora [1998])
- theorems of Hanan [1966] and Hwang [1976]
- exact algorithm for up to 10000 terminals:
  GeoSteiner (Warme, Winter, Zachariasen [2000])
- fast exact algorithm for up to 9 terminals:
  FLUTE (Chu, Wong [2008])
- fast approximation algorithm: modification of Prim's algorithm
- many heuristics ...

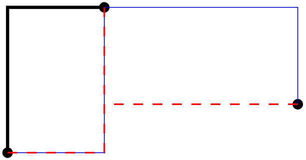
# FLUTE (Chu, Wong [2008])




- Let $I = \{(x_1, y_{\pi_1}), \dots, (x_n, y_{\pi_n})\}$ with $\{\pi_1, \dots, \pi_n\} = \{1, \dots, n\}$, $x_1 \leq \dots \leq x_n$, and $y_1 \leq \dots \leq y_n$.
- For each permutation $\pi$ there is a finite set of Steiner trees that are part of the Hanan grid.
- For such a tree $T$ let $\chi(T) := (X_1, \dots, X_{n-1}, Y_1, \dots, Y_{n-1})$, where $X_j$ is the number of edges $\{(x_j, y), (x_{j+1}, y)\}$ for some $y$, and $Y_j$ is defined analogously ($j = 1, \dots, n-1$).
- Store all minimal vectors $\chi(T)$ in a table. For $n = 9$ there are about $10^7$ such vectors.
- By simple reductions and symmetry this can be reduced significantly and the vector with the smallest scalar product with $(x_2 - x_1, \dots, x_n - x_{n-1}, y_2 - y_1, \dots, y_n - y_{n-1})$ can be found very fast (for $n \leq 9$).

# Modification of Prim's algorithm

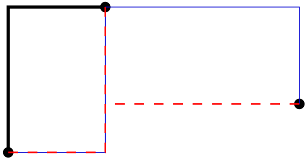
- Start with a single terminal $s$, $T = (\{s\}, \emptyset)$.
- For a terminal $t \notin V(T)$ and an edge $\{v, w\} \in E(T)$ let $d(t, v, w) := \min_{z \in \mathbb{R}^2}(||t - z||_1 + ||v - z||_1 + ||w - z||_1 - ||v - w||_1)$.
- Insert $t$ via $z$ into $\{v, w\}$ where the minimum (over all $t, v, w, z$) is attained.
- Iterate until all terminals are inserted.



## Theorem (folklore)

*The resulting Steiner tree is at most 1.5 times longer than optimal.*

# Proof of performance guarantee



## Theorem (folklore)

*The resulting Steiner tree is at most 1.5 times longer than optimal.*

## Proof

- Let $T_i$ be the forest $T$ after $i$ iterations of the algorithm ($i = 1, \ldots, n-1$).
- Let $Z_0$ be a minimum spanning tree for the terminals.
- For $i = 1, \ldots, n-1$ let $Z_i$ be a tree with $V(Z_i) = V(T_i)$ and $E(T_i) \subseteq E(Z_i) \subseteq E(T_i) \cup (E(Z_{i-1} \cap E(Z_0))$.
- Then $c(E(Z_{i-1})) \geq c(E(Z_i))$ for $i = 1, \ldots, n-1$.
- Note that $Z_{n-1} = T_{n-1}$ and $c(E(Z_0))$ is at most $\frac{3}{2}$ times the cost of an optimum Steiner tree. $\square$

# Number of instances and running times

| # terminals | # instances | total runtime |
|---:|---:|---:|
| 2 | 3726352 | 11.095 sec |
| 3 | 598625 | 2.303 sec |
| 4 | 294251 | 1.282 sec |
| 5 | 145700 | 0.741 sec |
| 6 | 75444 | 0.577 sec |
| 7 | 43516 | 0.394 sec |
| 8 | 27528 | 0.301 sec |
| 9 | 26779 | 0.464 sec |
| 10 | 19972 | 0.282 sec |
| $\leq$ 100 | 130358 | 8.500 sec |
| $\leq$ 1000 | 1392 | 1.917 sec |
| $\leq$ 10000 | 53 | 5.015 sec |
| $\leq$ 100000 | 21 | 11.806 sec |
| $\leq$ 1000000 | 3 | 34.749 sec |

Instances up to 9 terminals solved optimally. Other trees $< 2\,\%$ longer on average. Total length $< 0.1\,\%$ longer than optimum.

# Placement: modeling hyperedges (multi-terminal nets)

Let $N$ be a finite set of points in the plane. Define net models:

- $\textsc{Steiner}(N)$ := length of an optimum rectilinear Steiner tree for $N$. This is expected to be close to the actual routing length.
- $\textsc{BB}(N) := \max_{p \in N} x(p) - \min_{p \in N} x(p) + \max_{p \in N} y(p) - \min_{p \in N} y(p)$.
- $\textsc{MST}(N)$ := length of a minimum spanning tree for $N$, where edge weights are rectilinear distances.
- $\textsc{Clique}(N) := \frac{1}{|N|-1} \sum_{p,p' \in N} (|x(p)-x(p')|+|y(p)-y(p')|)$.
- $\textsc{Star}(N) := \min_{(x',y') \in \mathbb{R}^2} \sum_{p \in N} (|x(p)-x'|+|y(p)-y'|)$.

# Worst case ratios of various net models

Entry $(r, c)$ is $\sup \frac{c(N)}{r(N)}$ over all point sets $N$ with $|N| = n \in \mathbb{N}$.

| | BB | STEINER | MST | CLIQUE | STAR |
|---|---|---|---|---|---|
| BB | $1$ | $1$ | $1$ | $1$ | $1$ |
| STEINER | $\frac{n-1}{\lceil\sqrt{n}\rceil+\left\lceil\frac{n}{\lceil\sqrt{n}\rceil}\right\rceil-2}$ $\cdots$ $\frac{\lceil\sqrt{n-2}\rceil}{2}+\frac{3}{4}$ | $1$ | $1$ | $\begin{cases} \frac{9}{8} & (n=4) \\ 1 & (n \neq 4) \end{cases}$ | $1$ |
| MST | $\left\lfloor\frac{\sqrt{2n-1}+1}{2}\right\rfloor$ $\cdots$ $\frac{\sqrt{n}}{\sqrt{2}}+\frac{3}{2}$ | $\frac{3}{2}$ | $1$ | $1+\Theta\left(\frac{1}{n}\right)$ $\cdots$ $\frac{3}{2}$ | $\begin{cases} \frac{4}{3} & (n=3) \\ \frac{3}{2} & (n=4) \\ \frac{6}{5} & (n=5) \\ 1 & (n>5) \end{cases}$ |
| CLIQUE | $\frac{\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor}{n-1}$ | $\frac{\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor}{n-1}$ | $\frac{\lceil\frac{n}{2}\rceil\lfloor\frac{n}{2}\rfloor}{n-1}$ | $1$ | $1$ |
| STAR | $\lfloor\frac{n}{2}\rfloor$ | $\lfloor\frac{n}{2}\rfloor$ | $\lfloor\frac{n}{2}\rfloor$ | $\frac{n-1}{\lceil\frac{n}{2}\rceil}$ | $1$ |

(Hwang [1976], Brenner, V. [2001], Rautenbach [2004])

# Net models in placement

- STEINER is best, but *NP*-hard to compute
- all others can be computed in $O(n)$ time (BB, STAR) or in $O(n \log n)$ time (MST, CLIQUE).
- in quadratic placement:

$$\min \sum_{e=\{v,w\} \in E(G)} \left( (x_v - x_w)^2 + (y_v - y_w)^2 \right)$$

  CLIQUE and STAR are used
- BB is often used as a simple measure. As most nets have few pins, this is not too bad.

# Clique is the best topology-independent net model

**Theorem**

*For $n \geq 2$, a connected graph $G$ with $\{1, \ldots, n\} \subseteq V(G)$, $c : E(G) \to \mathbb{R}_{>0}$, and $p : \{1, \ldots, n\} \to \mathbb{R}^2$ let*
$\mathcal{M}_{(G,c)}(p) :=$

$$\min\left\{ \sum_{e=\{v,w\} \in E(G)} c(e) ||p(v) - p(w)||_1 \;\middle|\; p : V(G) \setminus \{1, \ldots, n\} \to \mathbb{R}^2 \right\}.$$

*Then the ratio of supremum and infimum of*

$$\left\{ \mathcal{M}_{(G,c)}(p) \middle| p : \{1, \ldots, n\} \to \mathbb{R}^2, \ \textsc{steiner}(\{p(1), \ldots, p(n)\}) = 1 \right\}$$

*is minimum for the complete graph $K_n$ with unit weights.*

(Brenner, V. [2001])

# Steiner trees in timing optimization

Instance:

- a root $r \in \mathbb{R}^2$,
- a finite set $S \subset \mathbb{R}^2$ of sinks,
- for each sink $s \in S$ a maximal feasible delay $d_{\max}(s)$

Task: Compute

- an arborescence $A$ rooted at $r$ whose set of sinks is $S$, and
- $\psi : V(A) \setminus (\{r\} \cup S) \to \mathbb{R}^2$,

such that $t(r) := \min\{0, \min_{s \in S}(d_{\max}(s) - \mathit{delay}_{(A,\psi)}(r, s))\}$ is maximum, and the total length is minimum.

# Unbuffered ("RC-optimal") trees

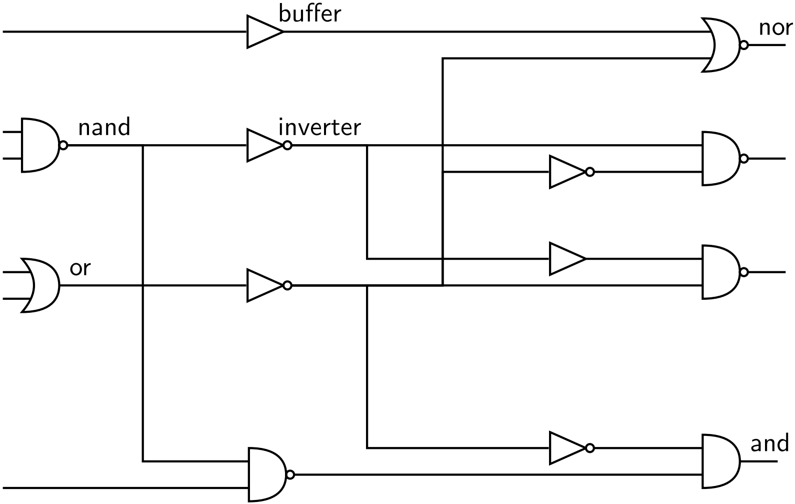
Standard delay model:

- capacitance $c_e$ and resistance $r_e$ of an edge $e$ proportional to its length
- downstream capacitance $C_v$ of a vertex $v$ given for sinks and recursively defined by $C_v := \sum_{e=(v,w)\in\delta^+(v)}(c_e + C_w)$.
- resistance $R$ of source given.
- $delay_{(A,\psi)}(r,s) = RC_r + \sum_{e=(v,w)\in A_{[r,s]}} r_e(\frac{1}{2}c_e + C_w)$, where $A_{[r,s]}$ is the $r$-$s$-path in $A$.

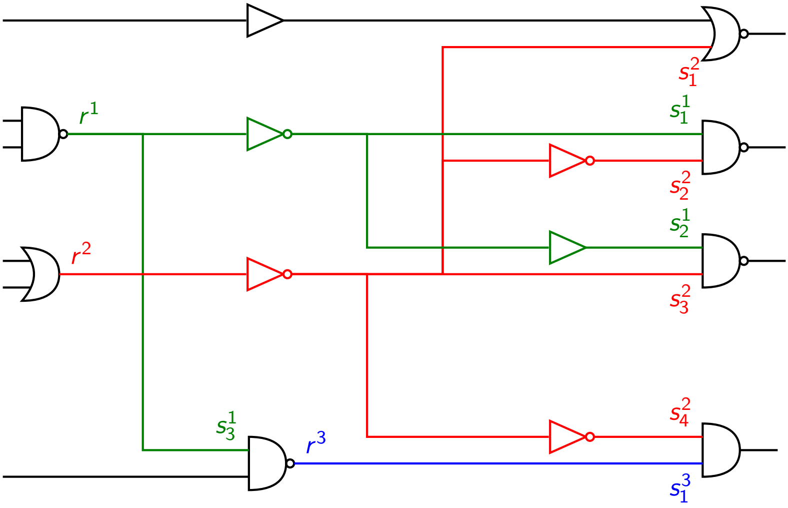
(Elmore [1948])

- *NP*-hard (Boese, Kahng, McCoy, Robin [1994])
- in general no optimal solution is part of the Hanan grid.
- Kadodi [1999] and Peyer [2000] gave algorithms for $n \leq 4$.
- no finite algorithm known in general.

# Buffered trees: using inverters as repeaters



# Buffered trees: an inverter tree

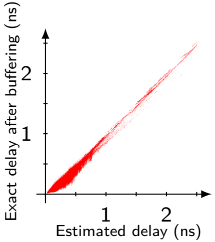


# Fast and short repeater tree topologies

New delay model:

$$delay_{(A,\psi)}(r,s) = \sum_{(v,w)\in A[r,s]} \left( dist(v,w) + (|\delta^+(v)| - 1) \right),$$

where *dist* denotes $\ell_1$-distance.




Fact 1: Huffman coding yields optimum latency, but with length $\sum_{s\in S} dist(r,s)$.

Fact 2: Starting with an isolated root and successively inserting a closest sink is a $\frac{3}{2}$-approximation for the Steiner tree problem.

# Fast and short repeater tree topologies

## Proposed Algorithm:

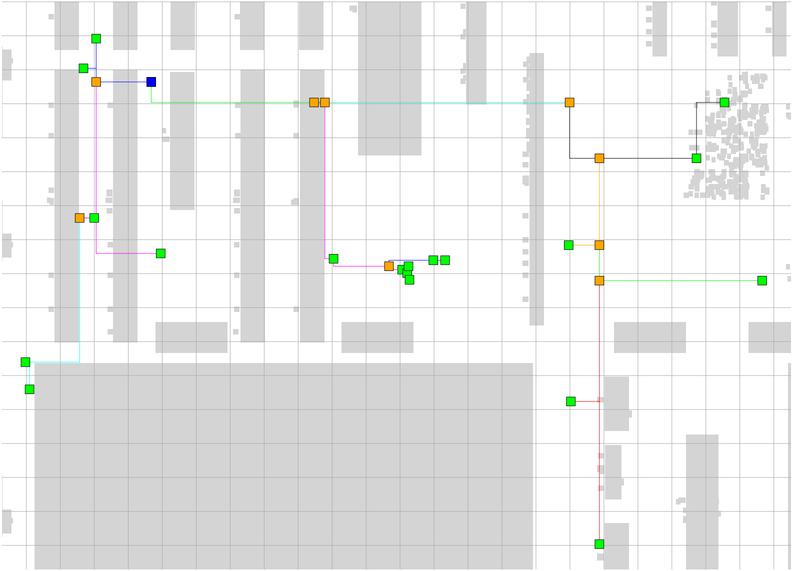
- Sort the sinks by $d_{\max}(s) - dist(r,s)$, in nondecreasing order.
- Start by connecting the first sink to the root.
- Then successively insert the sinks in the above order. Insert $s$ into edge $e \in E(A)$ such that $\min\{d_{\max}(s') - delay_A(r,s') : s' \in V(A)\}$ is maximum, or the total length is minimum, or a linear combination.

## Theorem

*If all distances are zero, this also results in the optimum, namely* $t(r) = -\left\lceil \log_2 \left( \sum_{s \in S} 2^{-d_{\max}(s)} \right) \right\rceil$.

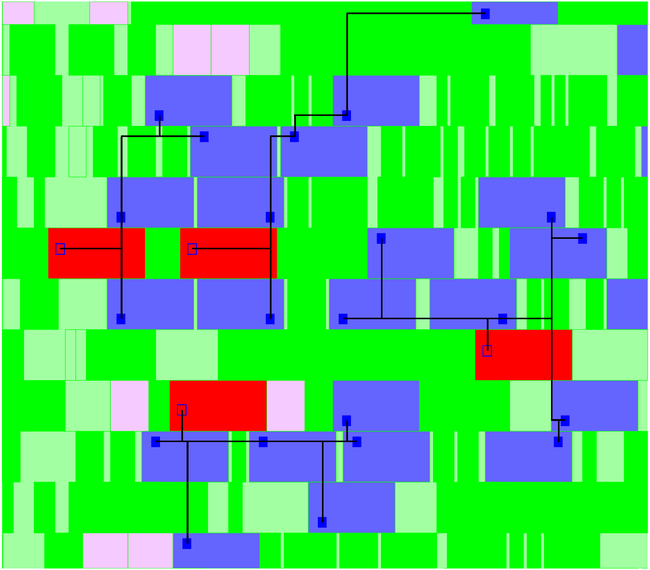
Experimental results show that in average, these trees are 0.66% longer or 0.22ps worse than the optimum.
(Bartoschek, Held, Rautenbach, V. [2006])

# Example of a real inverter tree



blue: source, green: 19 sinks, orange: 9 inverters colored lines: nets

# Distributing a signal to many terminals: sink clustering

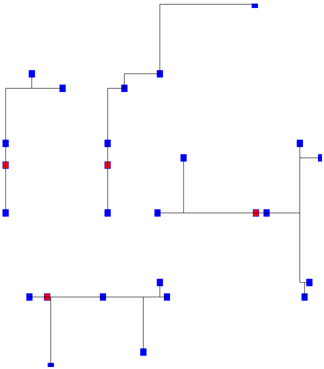


blue: sinks (terminals, clients) red: drivers (facilities)

# Sink clustering problem

## Instance:

- metric space $(V, c)$,
- finite set $\mathcal{D} \subseteq V$ (terminals/clients),
- demands $d : \mathcal{D} \to \mathbb{R}_+$,
- facility opening cost $f \in \mathbb{R}_+$,
- capacity $u \in \mathbb{R}_+$.



Task: Find a partition $\mathcal{D} = D_1 \dot\cup \cdots \dot\cup D_k$ and Steiner trees $T_i$ for $D_i$ $(i = 1, \ldots, k)$ with

$$c(E(T_i)) + d(D_i) \leq u$$

for $i = 1, \ldots, k$ such that

$$\sum_{i=1}^{k} c(E(T_i)) + k \cdot f$$

is minimum.

# Approximation algorithms

## Proposition

- *There is no* $(1.5 - \epsilon)$*-approximation algorithm (for any* $\epsilon > 0$*) unless* $P = NP$*.*
- *There is no* $(2 - \epsilon)$*-approximation algorithm (for any* $\epsilon > 0$*) for any class of metrics where the Steiner tree problem cannot be solved exactly in polynomial time.*

## Theorem

- *There is a polynomial-time 4.099-approximation algorithm for general metric spaces.*
- *There is an* $O(n \log n)$*-time 4-approximation algorithm for the rectilinear plane.*

(Maßberg and V. [2005])

# Extensions

- wires must avoid routing blockages
- repeaters cannot be placed on macros
- thus, unbuffered trees may cross most macros, but not by a long distance
- different wiring planes, with different electrical properties, should be considered
- routing congestion should be avoided
- placement space is limited, too

# Steiner trees in routing

- Now compute the exact layout for each net
- Wire shapes must follow certain rules for manufacturability
- Wires for different nets must be apart from each other
- Take previously computed information (e.g., layer assignment) into account
- Observe timing constraints, optimize power consumption or yield

# Steiner trees in routing: general approach

- Split task into global and detailed routing
- Global routing includes global optimization, packing Steiner trees
- Detailed routing considers one net at a time
- model routing space by a kind of 3-dimensional grid graph ("track graph", with currently up to $10^{11}$ vertices)
- vertex and edge weights
- Steiner tree algorithms (like Dreyfus-Wagner): too slow
- compose Steiner trees of paths
- Dijkstra in standard form: too slow
- very fast variants of Dijkstra's algorithm are used here

# Summary

- Steiner trees ubiquious in chip design
- minimum length Steiner trees is just one subproblem
- different objectives in placement, timing optimization, routing
- early estimates should match final realization
- many instances
- most, but not all, have only few terminals