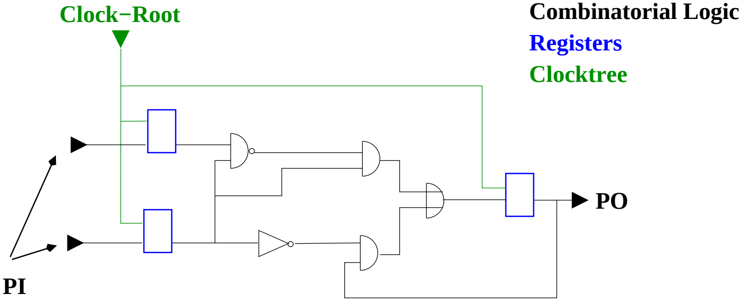
Repeater Trees in VLSI Design

Dieter Rautenbach

Forschungsinstitut für Diskrete Mathematik
Universität Bonn

21. Juni 2006

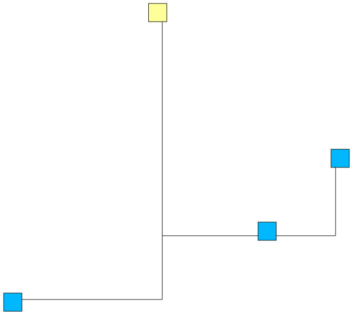
ASIC=Application Specific Integrated Circuit



We see a chiplet with one non-trivial repeater tree.

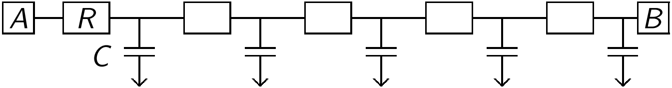
Wiring

The task of the wiring is to distribute a signal from one pin to possibly several other pins.



A pure metal connection is often not good enough.

Delay considerations

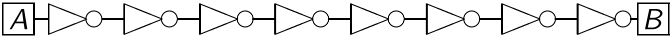


The total resistance $R_t = c_1 l$ and capacitance $C_t = c_2 l$ is distributed over the length l and can be approximated by n resistor/capacitor pairs. The resulting Elmore delay is

$$\begin{aligned} & nRC + (n-1)RC + \ldots + 3RC + 2RC + RC \\ = & \binom{n+1}{2} RC \approx 1/2(nR)(nC) = 1/2 R_t C_t \\ = & \frac{c_1 c_2}{2} l^2, \end{aligned}$$

i.e. the delay grows quadratically with the length l.

Delay considerations, II



If we insert an (even) number of identical inverters in the metal at equal distances, then each inverter has to drive the same capacitance. If the signal slew remains controlled over the sequence of inverters,then each inverter causes the same additional delay.

$$\Rightarrow$$

The overall delay grows linearly with the length.

Technological trends

- feature size shrinking
- $\frac{\text{Interconnect Delay}}{\text{Circuit Delay}}$ keeps growing
- More and more circuits are needed to strengthen the signals
  $20-30\%$ in 90nm technologies
  $30-40\%$ in 65nm technologies
- Success heavily depends on the quality of repeater trees
- On todays designs up to $30 \cdot 10^6$ repeater trees have to be built for Timing Closure

The repeater tree problem

Input

- A root (*root*) r at position $Pl(r)$
- A set S of sinks $s \in S$ with
  - position $Pl(s)$
  - required arrival time $RAT(s)$
    (W.l.o.g. $AT(r) = 0$)
  - required parity $+$ or $-$
  - physical information (input capacitance of the corresponding circuits,...)
- A library $\mathcal{L}$ of available repeaters

Output

A repeater tree T that connects r with all sinks in S, realizes the desired parities and respects all timing restrictions.

$$AT(r) + delay_T(r, s) \leq RAT(s) \; \forall s \in S$$

Optimization goals

- Minimize the total wiring length $\sim l_1$ metric
- Minimize power/area consumption $\sim$ sum of all input capacitances of inserted repeaters
- *Maximize the worst slack* σ_T

$$\sigma_T = \min_{s \in S} RAT(s) - (AT(r) + delay_T(r,s))$$

Previous Work

- "Optimally" insert repeaters into a given topology (embedded tree) on a subsets of a given set of possible positions.
  - Dynamic programming,
    $O(|\mathcal{L}|^2)$ van Ginneken '90,
    $O(|\mathcal{L}| \log(|\mathcal{L}|))$ Shi und Li '03, '05
  - Where do get get topology?
    Where do we get the positions?
- Partition the sinks according to their criticality/parity and determine a suitable topology.
  - What is criticality?
    What is "suitable"?

Our repeater tree algorithm

Our algorithm works in two phases:

- Generation of a “preliminary” topology
- Greedy bottom-up buffering of the preliminary topology

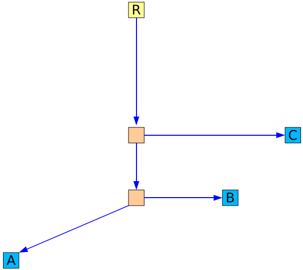
⇒ *During phase one there are no real repeaters. Hence we need an appropriate delay model.*

A preliminary topology

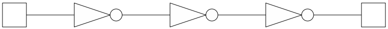
A preliminary topology is an arborescense $T = (V, E)$ with root r and leaves S. All interior nodes

$$u \in V \setminus (\{r\} \cup S)$$

correspond to bifurcations and satisfy $\delta^+_T(u) = 2$. All interior nodes u are assigned a position $Pl(u)$.



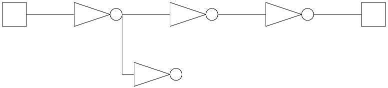
The delay modell



The delay of a two-point connection between u and v into which we optimally inserted equal and equidistant repeaters (inverters) grows linearly with its distance.

$$delay(u, v) = c_{wire} dist(Pl(u), Pl(v))$$

The delay modell, II



Each bifurcation of the topology represents an additional capacitance load.

$$delay(u, v) \leftarrow delay(u, v) + c_{node}$$

Note that, by the degree conditions in the definition of the preliminary topology, the fanout contributes logarithmically to the delay.

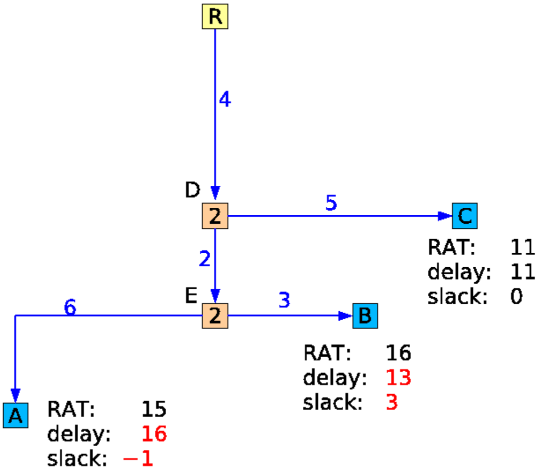
The delay modell, III

In a preliminary topology $T = (V, E)$ we estimate the delay from the root to r to a sink $s \in S$ by the following term.

$$\begin{aligned} delay_T(r,s) &= c_{node}(|E(T_{[r,s]})| - 1) \\ &+ \sum_{(u,v) \in E(T_{[r,s]})} c_{wire} dist(Pl(u), Pl(v)) \end{aligned}$$

Typical values: $c_{node} = 20\text{ps}$ and $c_{wire} = 220\text{ps/mm}$

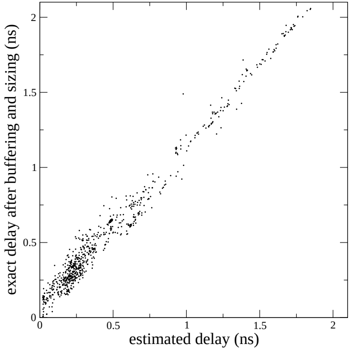
An example for the delay modell



$c_{wire} = 1, c_{node} = 2.$

The modell and the real world

The delay along critical paths estimated with our delay modell and calculated after buffering and exact timing.



The criticality of sinks

The criticality of a sink $s \in S$ is quantified by the slack of an optimally buffered two-point connection from r to s, i.e. a repeater tree with just one sink.

$$RAT(s) - AT(r) - c_{wire} dist(Pl(r), Pl(s))$$

This accounts for $RAT(s)$ as well as $dist(Pl(r), Pl(s))$.
The smaller this number the more critical is the sink.

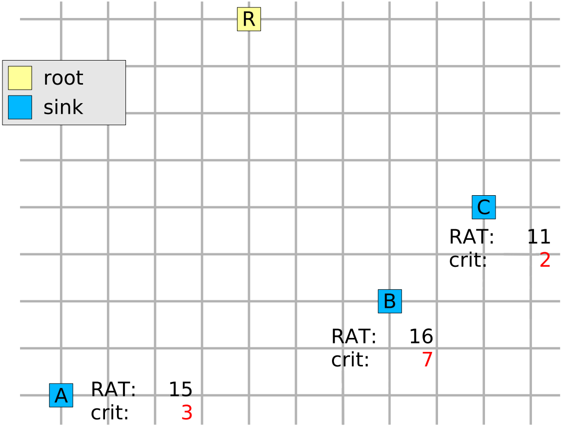
Phase I: Generation of the preliminary topology

- Sort the sinks according non-increasing criticality

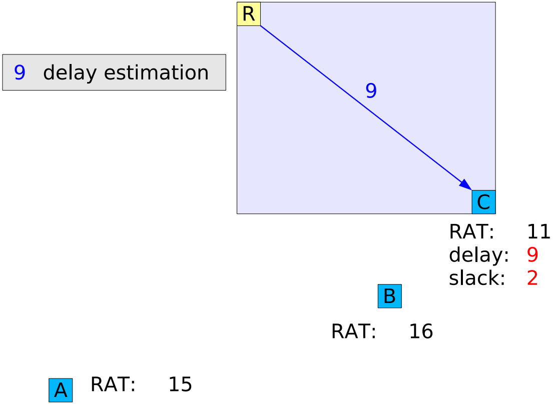
$$s_1, s_2, s_3, ...$$

- The first topology just contains r and s_1.
- Have the sinks $s_1, ..., s_{i-1}$ already been inserted, consider all arcs $e = (u, v)$ of the preliminary topology.
- If we connect s_i to a closest point of e, then we can estimate the resulting worst slack σ_e as well as the aditional wiring l_e.
- Choose e such that σ_e is maximized. Among all such arcs choose e such that l_e is minimized.
- Connect s_i to a closest point on e and iterate.

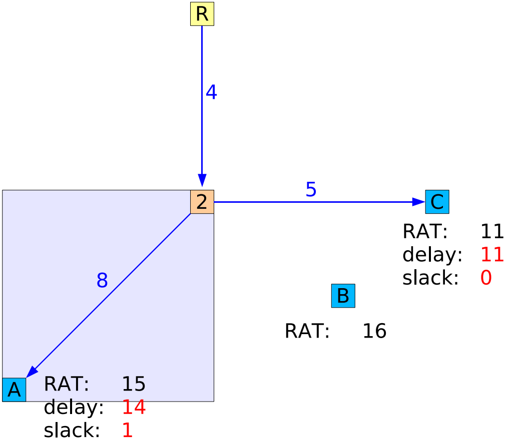
Example for the topology generation



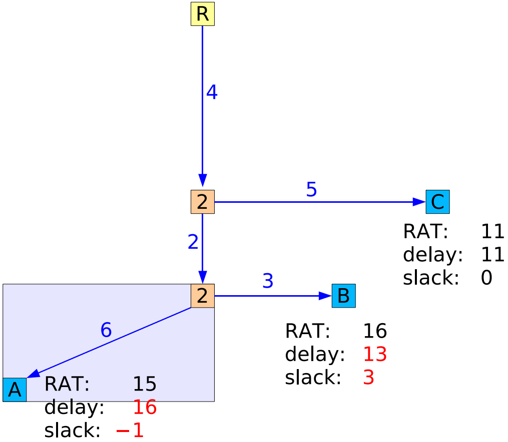
Example for the topology generation



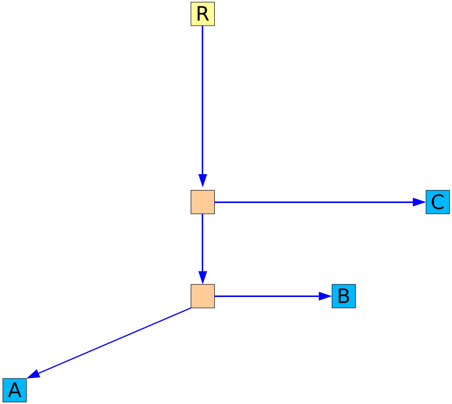
Example for the topology generation



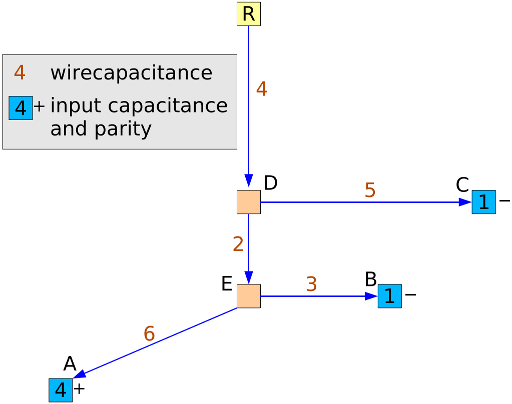
Example for the topology generation



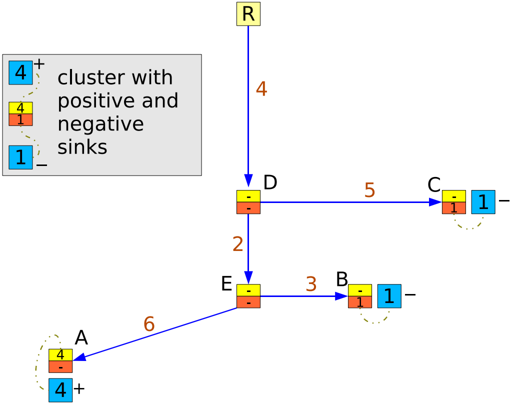
Example for the topology generation



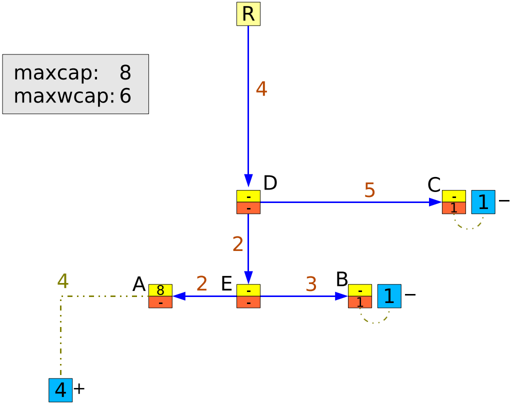
Phase II: Buffering (just an example)



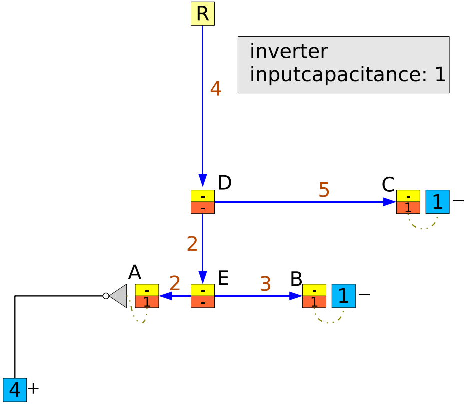
Phase II: Buffering (just an example)



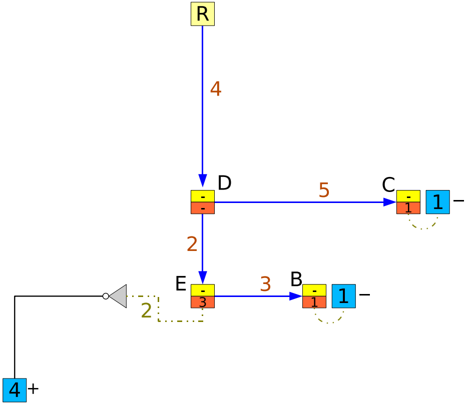
Phase II: Buffering (just an example)



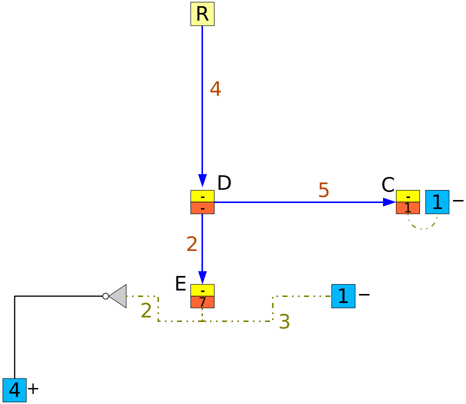
Phase II: Buffering (just an example)



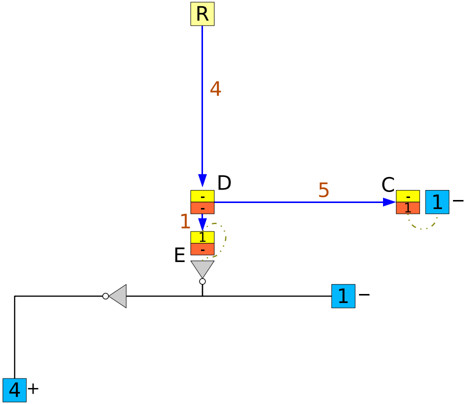
Phase II: Buffering (just an example)



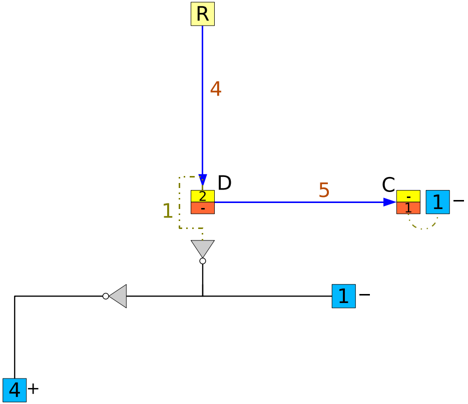
Phase II: Buffering (just an example)



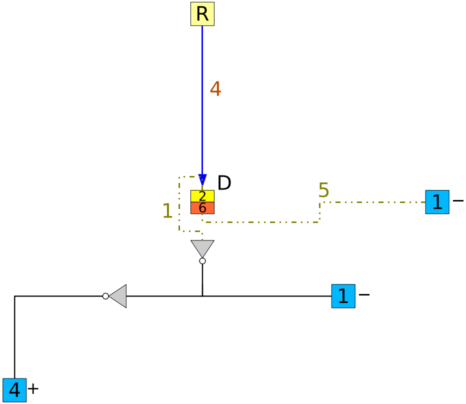
Phase II: Buffering (just an example)



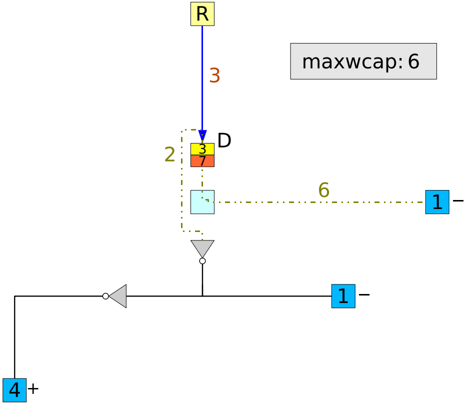
Phase II: Buffering (just an example)



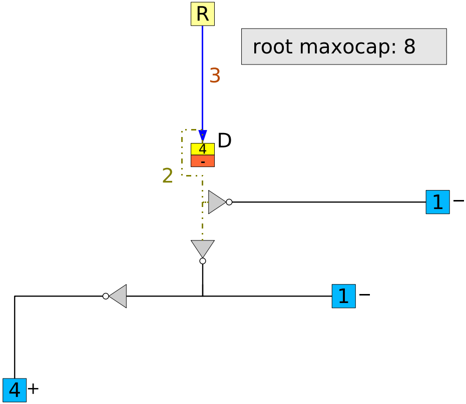
Phase II: Buffering (just an example)



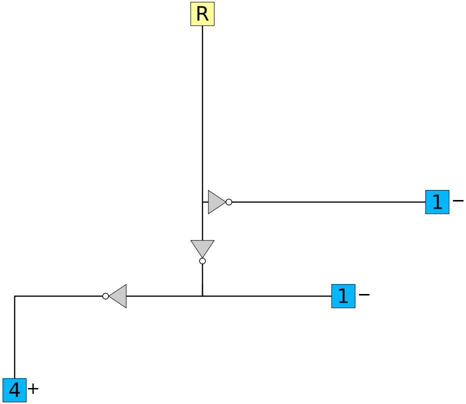
Phase II: Buffering (just an example)



Phase II: Buffering (just an example)



Phase II: Buffering (just an example)



The quality of the topology - bounds

- The maximum reachable worst slack within our delay model is at most

$$-c_{node} \cdot \log \left(\sum_{s \in S} 2^{-\left(\frac{RAT(s) - c_{wire} dist(Pl(r), Pl(s))}{c_{node}} \right)} \right).$$

- The total wiring length of the repeater tree is at least the minimum length of a rectilinear Steiner Tree on the set

$$\{Pl(r)\} \cup \{Pl(s) \mid s \in S\}.$$

Quality of the topology - slack

Theorem

For $c_{wire} = 0$ *and* $c_{node} = 1$ *and given integer values for* AT_r *and* RAT_s, $s \in S$, *the topology constructed by our procedure realizes the maximum possible slack with respect to our delay model, and this slack is*

$$-\left\lceil \log_2 \left(\sum_{s \in S} 2^{-RAT_s + AT_r} \right) \right\rceil .$$

Proof of the theorem

By Kraft's inequality there exists a rooted binary tree with n leaves at depths $l_1, l_2, \ldots, l_n$ if and only if

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1.$$

In order to realize a slack of at least σ in our model we must find a topology in which $RAT_s - AT_r - d_s \geq \sigma$ holds for every sink s where d_s denotes the number of internal nodes on the r-s-path. Note that if we contract the arc incident to the root within our topology, then we obtain a binary tree for which d_s corresponds exactly to the depth of sink s.

Proof of the theorem, II

$\Rightarrow$ The maximum slack is the largest integer $\sigma_{\max}$ that satisfies

$$\sum_{s \in S} 2^{-RAT_s + AT_r + \sigma_{\max}} \leq 1.$$

$\Rightarrow$

$$\sigma_{\max} = -\left\lceil \log_2 \left(\sum_{s \in S} 2^{-RAT_s + AT_r} \right) \right\rceil .$$

Proof of the theorem, III

That our constructed topology realizes this maximum possible slack follows easily by induction on the number of sinks. For one sink the statement is trivial. Now let $s_1, s_2, \dots, s_n$ denote the sinks ordered according to non-increasing criticality, i.e. $RAT_{s_i} \leq RAT_{s_j}$ for $i < j$. By induction, the topology T' containing all but the last sink s_n realizes the maximum possible slack σ' for these sinks. Since the procedure has the option to insert s_n using all arcs of T' leading to sinks, we can assume that all sinks are exactly at the maximum allowed depth $d_s = RAT_s - AT_r - \sigma'$ within T'. Again, by Kraft's inequality, this implies that $\sum_{i=1}^{n-1} 2^{-d_s}$ equals exactly 1. Thus $\sum_{i=1}^{n} 2^{-d_s} > 1$ which implies that the maximum possible slack σ for all n sinks is at least one less that σ' which will clearly be realized by the procedure. □

Quality of the topology - wiring

Theorem

If during the topology generation we always choose the next sink to be inserted as closest to the already constructed topology and just minimize l_e, then the l_1-length of the final topology is at most $\frac{3}{2}$ of the minimum l_1-length of a Steiner Tree on the terminals $\{r\} \cup S$.

The factor $3/2$ comes from Hwang.

Theorem (Hwang '76)

The minimum l_1-length of a spanning tree on a set of points in the plane is at most $\frac{3}{2}$ of the minimum l_1-length of a Steiner Tree on the set.

Proof of the theorem

Let $|S| = n$ and for $i = 1, \ldots, n$ let T_i denote the forest which is the union of the partial topology after the insertion of the first i sinks with the remaining sinks as isolated nodes. Note that $V(T_i)$ consists of $\{r\} \cup S$ together with $i - 1$ internal nodes.
Let F_0 denote a l_1-minimum spanning tree on $V(F_0) = \{r\} \cup S$.
For $i = 1, \ldots, n$ let F_i be a tree with $V(F_i) = V(T_i)$ and $E(T_i) \subseteq E(F_i) \subseteq E(T_i) \cup E(F_0)$. (Going from F_{i-1} to F_i a new sink from S is connected within F_i exactly as specified in T_i. For $i \geq 2$ this implies that some internal node is added to $V(F_{i-1})$ and three new arcs replace one arc which is deleted. In order to turn T_i into the tree F_i exactly $n - i$ arcs from F_0 are used.)

Proof of the theorem, II

By the choice of the order of insertion of the sinks and the arcs of the partial topology into which we insert, we have $l(F_i) \leq l(F_{i-1})$ for $i = 1, \ldots, n$. This property holds because our procedure has all remaining arcs from F_0 as possible choices.

In F_n all internal nodes and all arcs of the final topology have been added. Hence F_n is in fact exactly equal to the final topology T_n and we conclude $l(T_n) = l(F_n) \leq \cdots \leq l(F_0)$. The desired result follows, since, by Hwang's theorem, $l(F_0)$ is at most $\frac{3}{2}$ of the minimum l_1-length of a Steiner tree on the terminals $\{r\} \cup S$. $\square$

Running time and experimental results

- The running time of the topology generation is $O(|S|^2 \cdot \Psi)$ where Ψ is the times needed to determine a shortest path from s to e.
- Only for instances with $|S| > 10^4$ this is too slow and we apply some pre-clustering.
- 4.6 Mio. topolgies were constructed in $\leq$ 100 seconds on a 2.6 GHz Opteron (the buffering took $\leq$ 10 minutes).
- 2.3 Mio. instances with up to 10 000 sinks from a recent 90nm design.
- The wire length is compared to the minimum length of Steiner Trees (optimal for $|S| \leq 30$).
- The worst slack is compared to our bound (evaluated using Huffman coding for numerical stability).

Experimental results

		Wire Length Optimization				Slack Optimization			
		Wirelength Deviation (%)		Slack Deviation (ps)		Wirelength Deviation (%)		Slack Deviation (ps)	
# Sinks	# Instances	avg.	worst	avg.	worst	avg.	worst	avg.	worst
1	1547517	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	319759	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	165448	0.00	0.00	13.89	82.72	12.19	99.60	0.12	20.00
4	86377	0.16	19.65	23.72	312.98	10.93	190.27	0.27	40.00
5	44301	0.16	21.51	33.40	174.51	14.01	188.15	0.34	52.45
6	27854	0.28	23.84	41.92	118.27	14.38	268.06	1.04	52.93
7	20523	0.45	22.24	52.19	285.43	22.26	248.77	0.42	52.51
8	19300	0.44	30.73	64.01	332.29	19.39	268.49	2.08	69.13
9	11085	0.81	26.26	71.11	465.77	29.58	250.04	3.36	60.00
10	11942	0.74	28.68	76.46	367.39	23.61	296.47	1.45	54.87
11-20	38184	1.60	28.00	101.16	427.25	32.57	426.68	1.73	76.80
21-30	11104	3.20	30.80	144.27	520.00	35.86	805.45	2.51	84.18
31-50	8647	2.99	33.16	226.05	793.70	70.29	1091.17	6.55	161.81
51-100	6621	4.06	26.34	344.88	1486.06	105.90	1782.56	12.23	203.48
101-200	1863	5.82	16.91	606.26	2019.90	135.84	1498.34	19.78	351.25
201-500	824	6.22	24.00	920.37	3711.47	209.77	2127.34	26.91	304.92
501-1000	205	7.62	19.40	1686.15	3563.61	569.58	2242.49	48.57	257.65
$>$ 1000	31	6.99	14.74	2929.08	7872.96	211.40	1124.99	17.78	89.88
Total	2321585	**0.66**	33.16	**9.92**	7872.96	**19.35**	2242.49	**0.21**	351.25
$>$ 2 sinks	774068	**1.31**	33.16	**50.69**	7872.96	**38.34**	2242.49	**1.08**	351.25

Things to do...

- Both phases are constantly improved with respect to
  - delay distribution (lopsided version of Kraft's inequality)
  - gate sizing
  - blockages
  - plane assignment
  - better delay modell
  - ...
  - $\rightarrow$ Timing Driven Loop

Further reading

Christoph Bartoschek, Stephan Held, DR, and Jens Vygen, Efficient Generation of Short and Fast Repeater Tree Topologies, ISPD 2006

The end - for today

Thank you again for your attention!!