Clock distribution network
Quick summary...
Clock distribution network
Flip-flop = edge triggered copy
Flip-flop = edge triggered copy

--- FF
FF: process (clk, D)
begin
  if (clk'event and clk = '1') then
    Q <= D;
  end if;
end process FF;
Timing: constraints

$D$  xxxxxxxxx  xxxxxxxxxx

$clk$

$Q$

stable $Q$  stable $Q$

$T_{setup}$  $T_{hold}$
Timing: guarantees

- $T_{setup}$
- $T_{hold}$
- clk2Q delay
Goal: small skew
Clocked Design

\[ D \quad Q \]

\[ \text{stable } t(A) \quad \text{stable } t(t(A)) \]

\[ \text{stable } t(t(A)) \]

\[ \text{stable } A \quad \text{stable } t(A) \quad \text{stable } t(t(A)) \]
A stable t(A) and stable t(t(A)) are shown in the diagram. The launch clk path is indicated by red arrows, and the capture clk path is shown in green.

The combinational path connects the launch reg to the capture reg, and the clk signals are properly aligned with the D and Q signals.

The timing diagram shows the stable A and D signals over time, with the clk signal triggering the proper state transitions.
stable t(t(A))

0

1

2

3

4

5

6

capture clk path

capture reg

combinational path

launch reg

launch clk path

T_{cyc}

clock

$t^k_{skew}(5, 3)$
The setup constraint

$T_{cyc}$

$d_{r2r}(3, 5)$

$t_{skew}^{k+1}(3, 5)$

$T_{setup}$

$D \rightarrow Q$

launch reg

$D \rightarrow Q$

capture reg

launch clk path

capture clk path

$k$

$T_{cyc}$

clk
The setup constraint

large pos. skew is bad
The hold constraint

$T_{cyc}$

$t_{skew}^k(3, 5)$

$d_{r2r}(3, 5)$

$T_{hold}$

$D$ $Q$

launch reg

$D$ $Q$

capture reg

$T_{cyc}$

capture clk path

launch clk path

$D$

1

2

$T_{cyc}$

$clk$

$D$

4

5

6

$3$

$5$
The hold constraint

large neg. skew is bad
Clock distribution network
Its skew
Minimum path
Maximum path
But ...
... faults
New requirements

Guarantee skew among some clock outputs despite faults
Pulse Synchronization

In pulse synchronization, for each $i \in \mathbb{N}$, every (correct) node $v \in V_g$ generates pulse $i$ exactly once.

Let $p_{v,i}$ denote the time when $v$ generates the $i$-th pulse. We require that there are $S, P_{\text{max}}, P_{\text{min}} \in \mathbb{R}_{>0}$ satisfying

1. skew: $\sup_{i \in \mathbb{N}, u, w \in V_g} \{|p_{v,i} - p_{w,i}|\} = S$

2. per-1: $\inf_{i \in \mathbb{N}} \{ \min_{v \in V_g} p_{v,i+1} - \max_{v \in V_g} p_{v,i} \} \geq P_{\text{min}}$

3. per-2: $\sup_{i \in \mathbb{N}} \{ \max_{v \in V_g} p_{v,i+1} - \min_{v \in V_g} p_{v,i} \} \leq P_{\text{max}}$
Lower bounds

Number of faults $f$. Then necessarily:

Global: $n > 3f$

Local: $\text{degree} > 2f$
Ideas?
Chapter 11 (2)
Low-degree
clock distribution networks

Matthias Fuegger and Christoph Lenzen
Faults

\[ D \rightarrow Q \] (Blue)

\[ D \rightarrow Q \] (Green)

\[ \text{clk} \]

\[ 0 \]

\[ 1 \] (Red)

\[ 2 \] (Red)
Pulse Synchronization

In pulse synchronization, for each $i \in \mathbb{N}$, every (correct) node $v \in V_g$ generates pulse $i$ exactly once.

Let $p_{v,i}$ denote the time when $v$ generates the $i$-th pulse. We require that there are $S, P_{\text{max}}, P_{\text{min}} \in \mathbb{R}_{>0}$ satisfying

1. skew: $\sup_{i \in \mathbb{N}, u, w \in V_g} \{|p_{v,i} - p_{w,i}|\} = S$

2. per-1: $\inf_{i \in \mathbb{N}} \{\min_{v \in V_g} p_{v,i+1} - \max_{v \in V_g} p_{v,i}\} \geq P_{\text{min}}$

3. per-2: $\sup_{i \in \mathbb{N}} \{\max_{v \in V_g} p_{v,i+1} - \min_{v \in V_g} p_{v,i}\} \leq P_{\text{max}}$
Last time: ideas session
Idea 1: only LW algorithm

no tree, only LW

3f+1
Idea 1

5x5 grid

properties:
- fault tolerance?
- cost?
- skew?
properties:
- fault tolerance?

- cost?

- skew?
Idea 2: redundant trees

replicate the clock source, vote on output

2f+1

3f+1
Idea 2

properties:
- fault tolerance?
- skew?
- cost?
Fault tolerance

replicate the clock source, vote on output

1. at most f fail
Fault tolerance

replicate the clock source, vote on output

2. at most $f$ trees fail
Fault tolerance

replicate the clock source, vote on output

2. at most \( f \) trees fail
Idea 2

5x5 grid

properties:
- fault tolerance?
- skew?
- cost?
Skew
Skew
Skew
Skew

\[ \text{med}(p_{v1,k}, p_{v2,k}, p_{v3,k}) \leq \text{med}(p_{r1,k} + D, p_{r2,k} + D, \infty) \leq \max(p_{r1,k} + D, p_{r2,k} + D) \leq \max(p_{r1,k}, p_{r2,k}) + D \]
Skew

\[ \leq \max(p_{r_1,k}, p_{r_2,k}) + D \]
Skew

\[ D \rightarrow Q \rightarrow \text{med} \]

\[ \geq D - u \]

\[ \geq D - u \]

\[ \geq 0 \]

\[ S \]
Skew

\[ \geq \min(p_{r_1,k}, p_{r_2,k}) + D - u \]
\[
\leq \max(p_{r_1,k}, p_{r_2,k}) + D \quad \geq \min(p_{r_1,k}, p_{r_2,k}) + D - u
\]
Skew

$|\cdot| \leq S + u$

\[
\begin{align*}
\geq D - u \\
\geq D - u \\
\geq 0
\end{align*}
\]
Idea 2

properties:
- fault tolerance?
- skew?
- cost?
Cost

3f+1 roots & 2f+1 trees

\[ \text{med} \]

\[ p_{v_1,k} \quad p_{v_2,k} \quad p_{v_3,k} \]
Depends on the environment

3 faults
Q: other solutions?
Idea 3: interlinked trees

median at each stage of the tree
Skew

median at each stage of the tree
Skew

median at each stage of the tree
Idea 4: interlinked trees v2

median at each stage of the tree
Ex 11.1

\[ P(\geq 2 \text{ faults}) = 1 - P(\leq 1 \text{ faults}) \approx 0.80 \]

\[ 1 - \sum_{i=0}^{1} \binom{300}{i} p^i (1-p)^{300-i} \text{ where } p = 0.01 \]

\[ P(\geq 100 \text{ faults}) = 1 - P(\leq 99 \text{ faults}) \approx 6.9 \cdot 10^{-15} \]

\[ 1 - \sum_{i=0}^{99} \binom{300}{i} p^i (1-p)^{300-i} \text{ where } p = 0.01 \]
100 faults

global constraint (idea 1, LW):

\[ \Pr(\geq 100 \text{ faults}) = 1 - \Pr(\leq 99 \text{ faults}) \approx 6.9 \cdot 10^{-15} \]

local constraints (idea 2, redundant trees):

\[ \Pr(\text{correct}) = \Pr(\leq 1 \text{ faulty tree}) \]

\[ \{ p = 0.01, c = (1 - p)^{100}, c^3 + 3c(1 - c)^2 \} \]

\[ c(4c^2 - 6c + 3) \approx 0.490383 \]

local constraints (idea 3, interlinked trees):

\[ \Pr(\text{correct}) = \Pr(\leq 1 \text{ fault per triple}) \]

\[ \{ p = 0.01, c = (1 - p)^3 + 3(1 - p)^2 p, c^{100} \} \]

\[ c^{100} \approx 0.970635 \]
Chapter 11 (3)

Low-degree clock distribution networks

Matthias Fuegger and Christoph Lenzen
Clocking a grid with root(s) & tree(s)

5x5 grid

properties:
- fault tolerance
- skew
- cost
Clocking a grid with root(s) & tree(s)

properties:

- fault tolerance

- skew (g & l):
  \[ | \cdot | \leq S + U \]

- cost
Clocking a grid with root(s) & tree(s)

- skew (local):
  \[ | \cdot | \leq \mathcal{S} + U \]
  \[ U = f(u) \]
Limits

$k \times k$ grid, $n = k^2$
with $1 \times 1$ cells

Show:
In any tree spanning the points,
exist two grid-neighboring nodes with
tree-distance $d = \Omega(k)$. 
Tree:

degree $\leq 3$ (in+out degree)
Exists edge e st.

after removing e: both components size $\geq (n-1)/3 = \Omega(n)$
Exists edge $e \in E$ such that
after removing $e$: both components size $\geq (n-1)/3 = \Omega(n)$

$n = 9 \rightarrow$
subtree size $\geq 8/3 = 2.6..$
Pick e.

Case 1: $|T1|, |T2| \geq (n-1)/3$. Done.
Pick e.

Case 1: $|T_1|, |T_2| \geq (n-1)/3$. Done.

Case 2: $|T_1| < (n-1)/3$.

$\rightarrow |T_2| = n - |T_1| \geq n - (n-1)/3 = 2(n-1)/3 + 1$
Case 2: $|T_1| < \frac{(n-1)}{3}$.

$\rightarrow |T_2| = n - |T_1| \geq n - \frac{(n-1)}{3} = \frac{2(n-1)}{3} + 1$

$w := \text{endpoint of } e \text{ in } T_2$
Case 2: $|T_1| < (n-1)/3$.

$\rightarrow |T_2| = n - |T_1| \geq n - (n-1)/3 = 2(n-1)/3 + 1$

$w := \text{endpoint of } e \text{ in } T_2$

$\text{delete } w$

$\rightarrow \text{at most } 2 \text{ components of } T_2 \text{ (degree } \leq 3)$
Case 2: $|T_1| < (n-1)/3$.

$\Rightarrow |T_2| = n - |T_1| \geq n - (n-1)/3 = 2(n-1)/3 + 1$

$w := \text{endpoint of } e \text{ in } T_2$

delete $w$

$\Rightarrow \text{at most 2 components of } T_2 \text{ (degree } \leq 3): T_2' \text{ & } T_2''$

$\Rightarrow \text{exists component } T_2' \text{ with size } \geq (|T_2|-1)/2$

$\geq (n-1)/3$
-> exists component $T_2'$ with size $\geq (|T_2| - 1)/2$
$\geq (n-1)/3$

delete edge from component $T_2'$ to $w$ in $T$
exists component $T_2'$ with size $\geq (|T_2| - 1)/2$

$\geq (n-1)/3$

dele edge from component $T_2'$ to $w$ in $T$

$\rightarrow T_1'$ and $T_2'$ with

$|T_1'| > |T_1|$

$|T_2'| \geq (n-1)/3$
\[ |T1'| > |T1| \]
\[ |T2'| >= (n-1)/3 \]

if both \( >= (n-1)/3 \):
   Done.
else:
   Repeat with \( T1' \) and \( T2' \) instead of \( T1 \) and \( T2 \).
-> partition tree into 2 trees with Omega(k^2) nodes
-> partition tree into 2 trees with Omega(k^2) nodes

boundary node :=
grid-neighbors in both sets

dist(u,v) := dist in grid
d(u,v) := dist in tree

-> exist boundary nodes of T1 (and of T2) with dist = Omega(k) [geometric arg.]
-> partition tree into 2 trees with Omega(k^2) nodes

-> exist boundary nodes of T1 (and of T2) with dist = Omega(k)

u, v two such nodes in T1
exist boundary nodes of T1 (and of T2) with dist = Omega(k)
u, v two such nodes in T1

one (e.g. u) of a := dist = Omega(k) to T1 root (triangle)
u has d >= a = Omega(k) to its neighbor in T2
Q last time

5 x 5 grid

reg i-1

>=1

reg i

reg U(i)

reg i+1

reg B(i)
-> live session