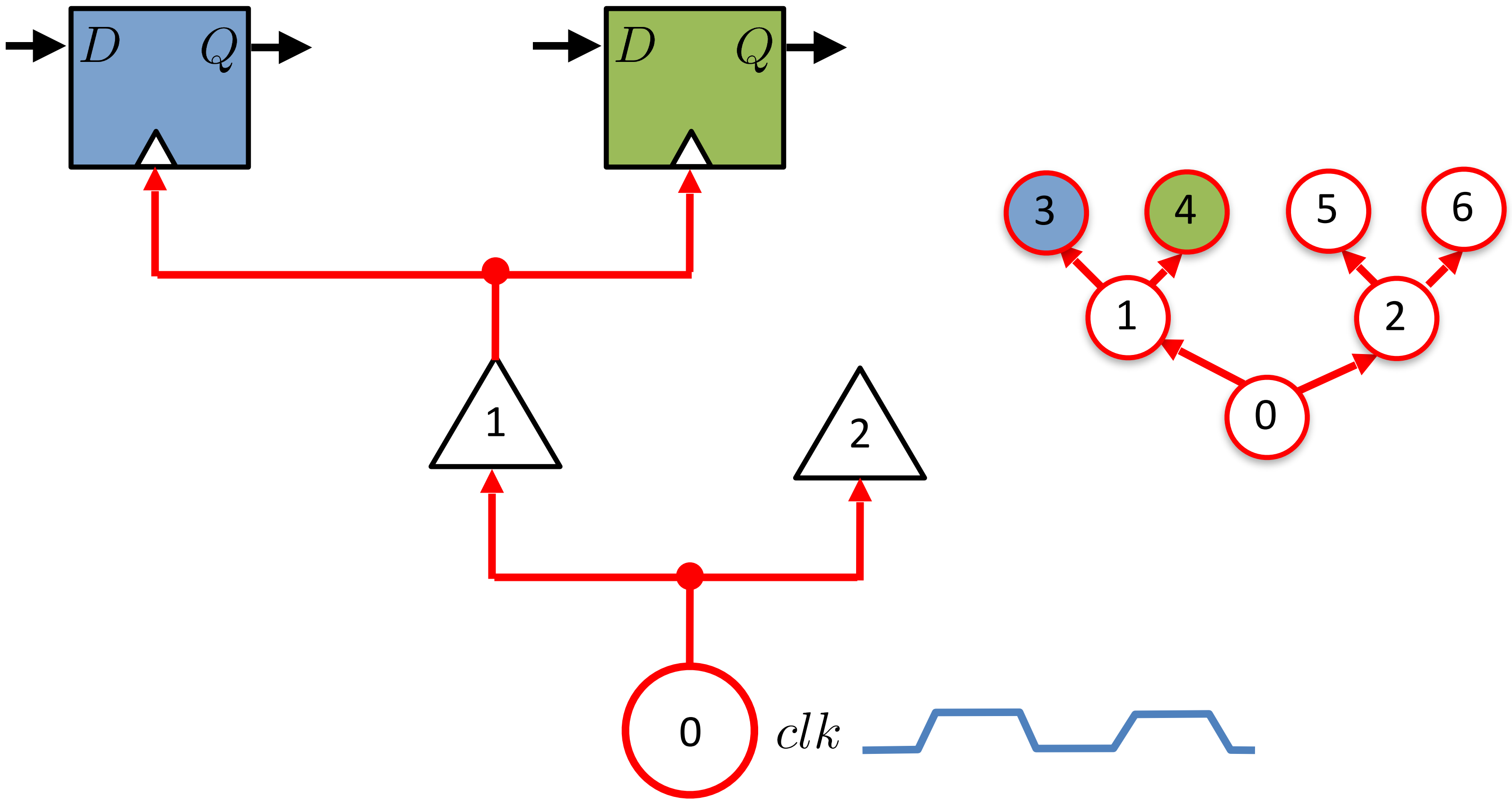
Chapter 11

Low-degree clock distribution networks

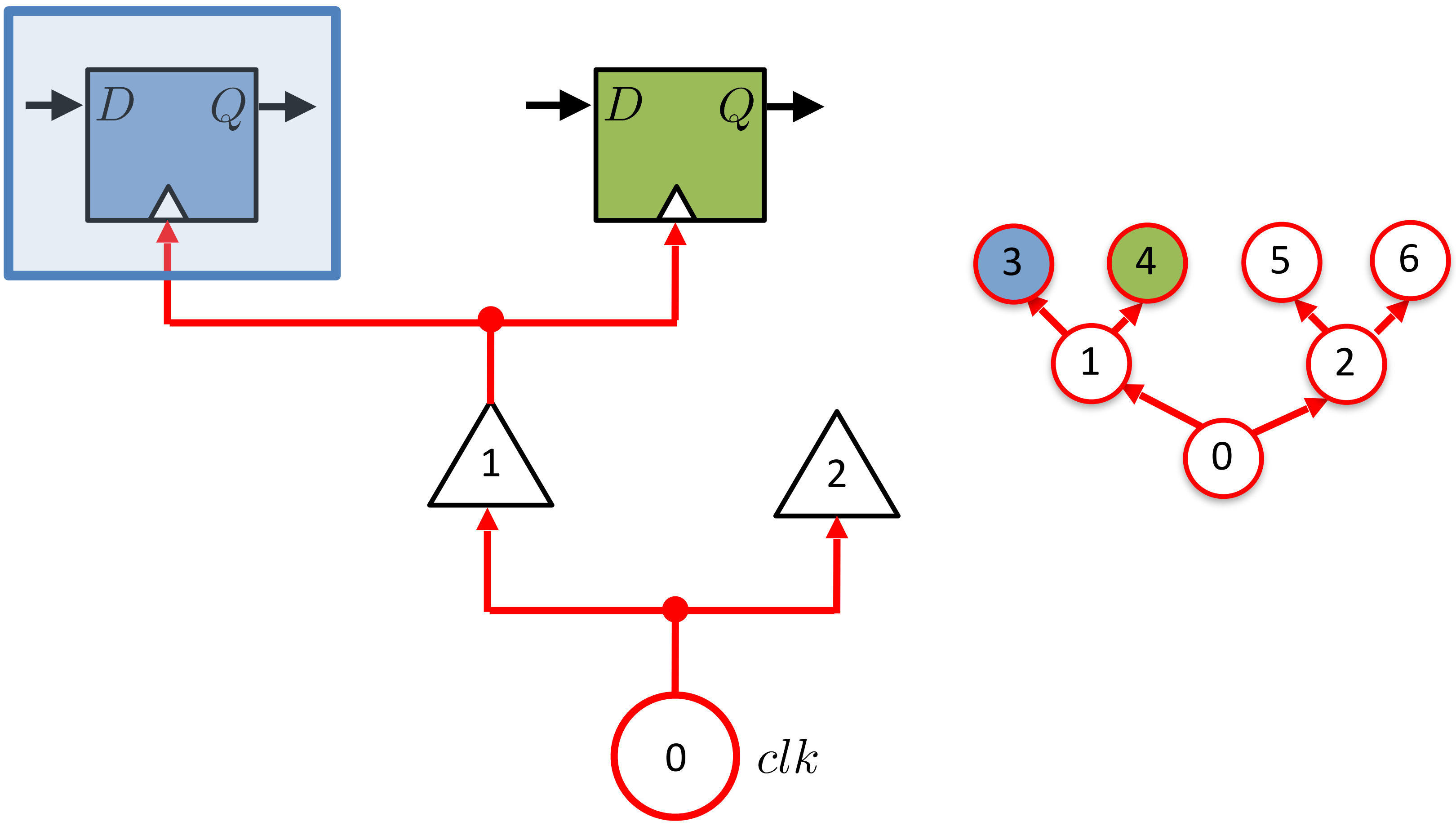
Matthias Fuegger and Christoph Lenzen

Clock distribution network

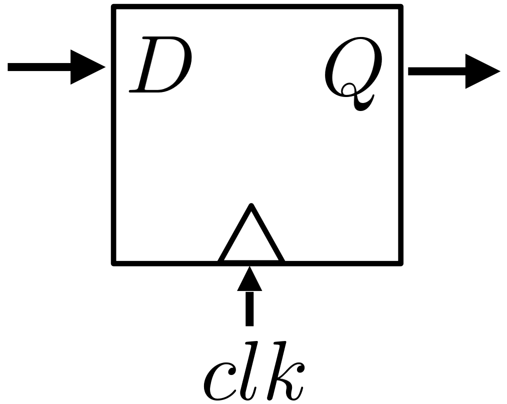


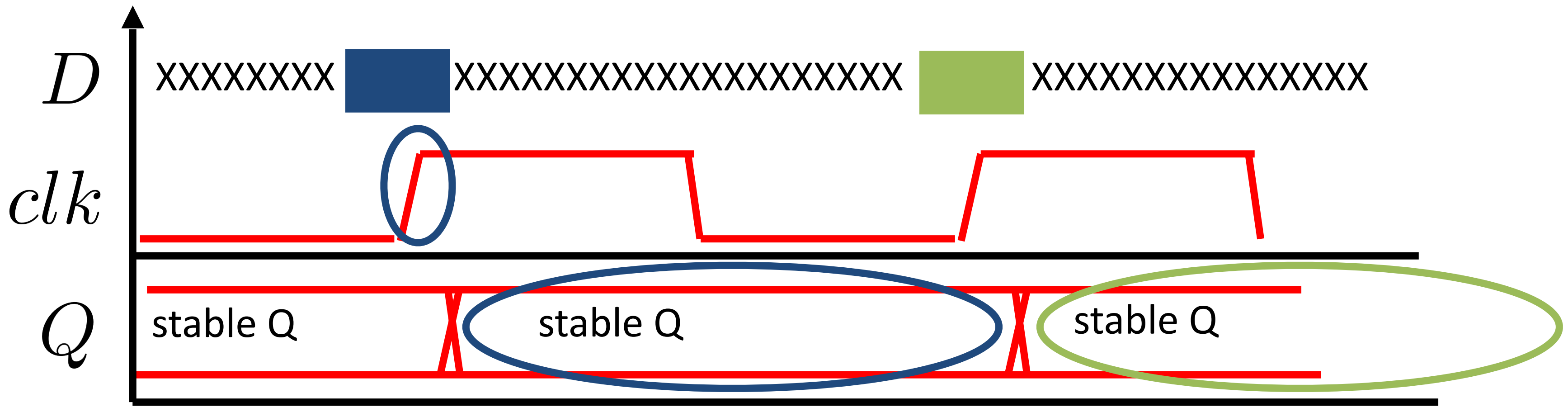
Quick summary...

Clock distribution network

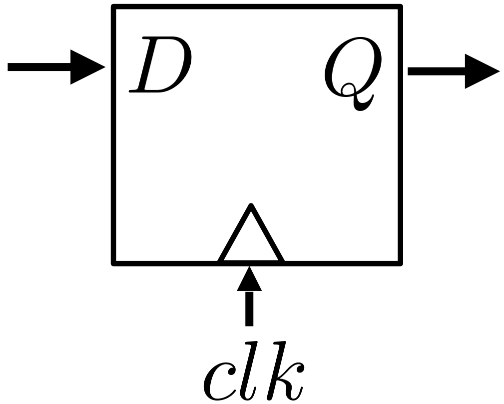


Flip-flop = edge triggered copy



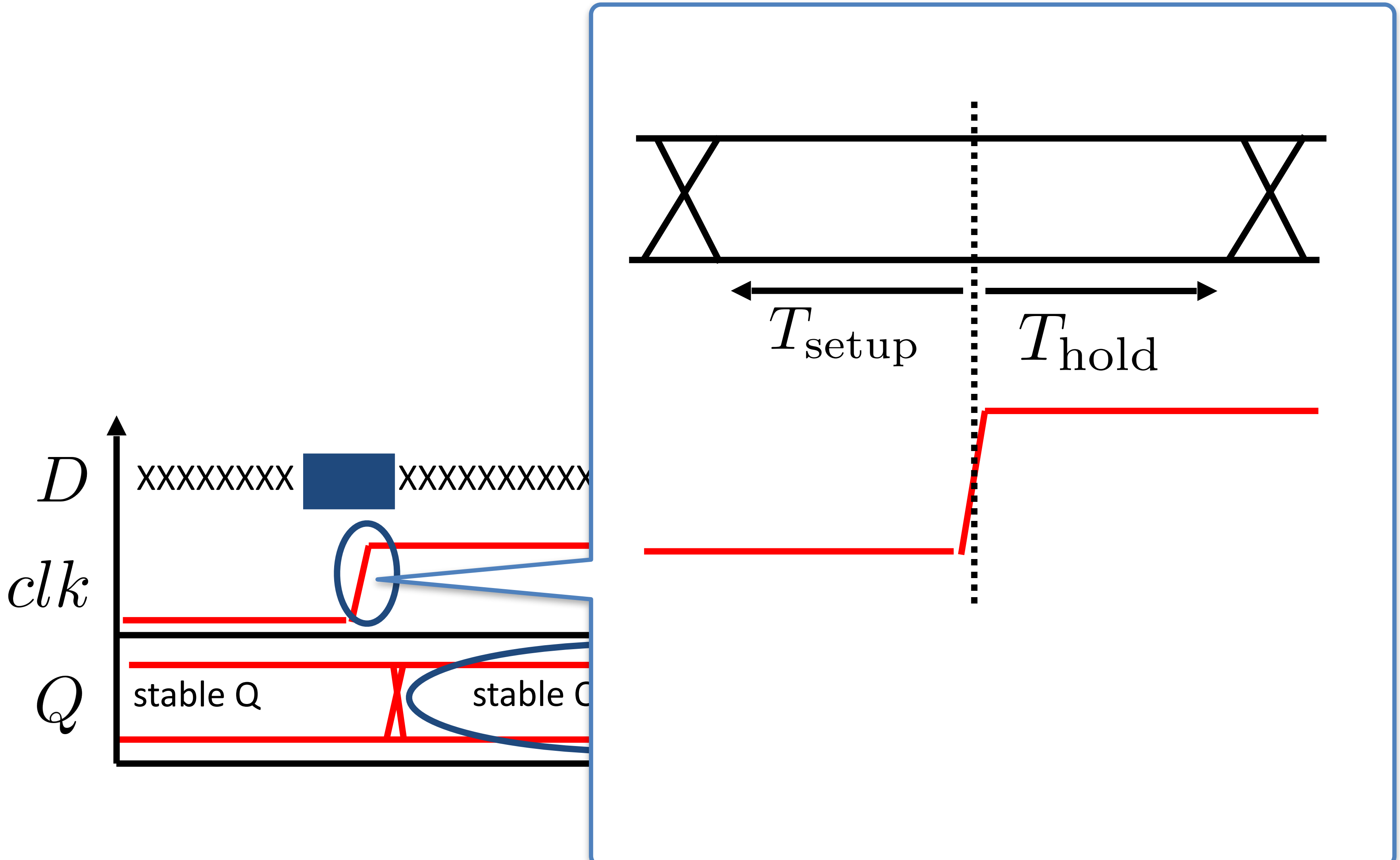


Flip-flop = edge triggered copy

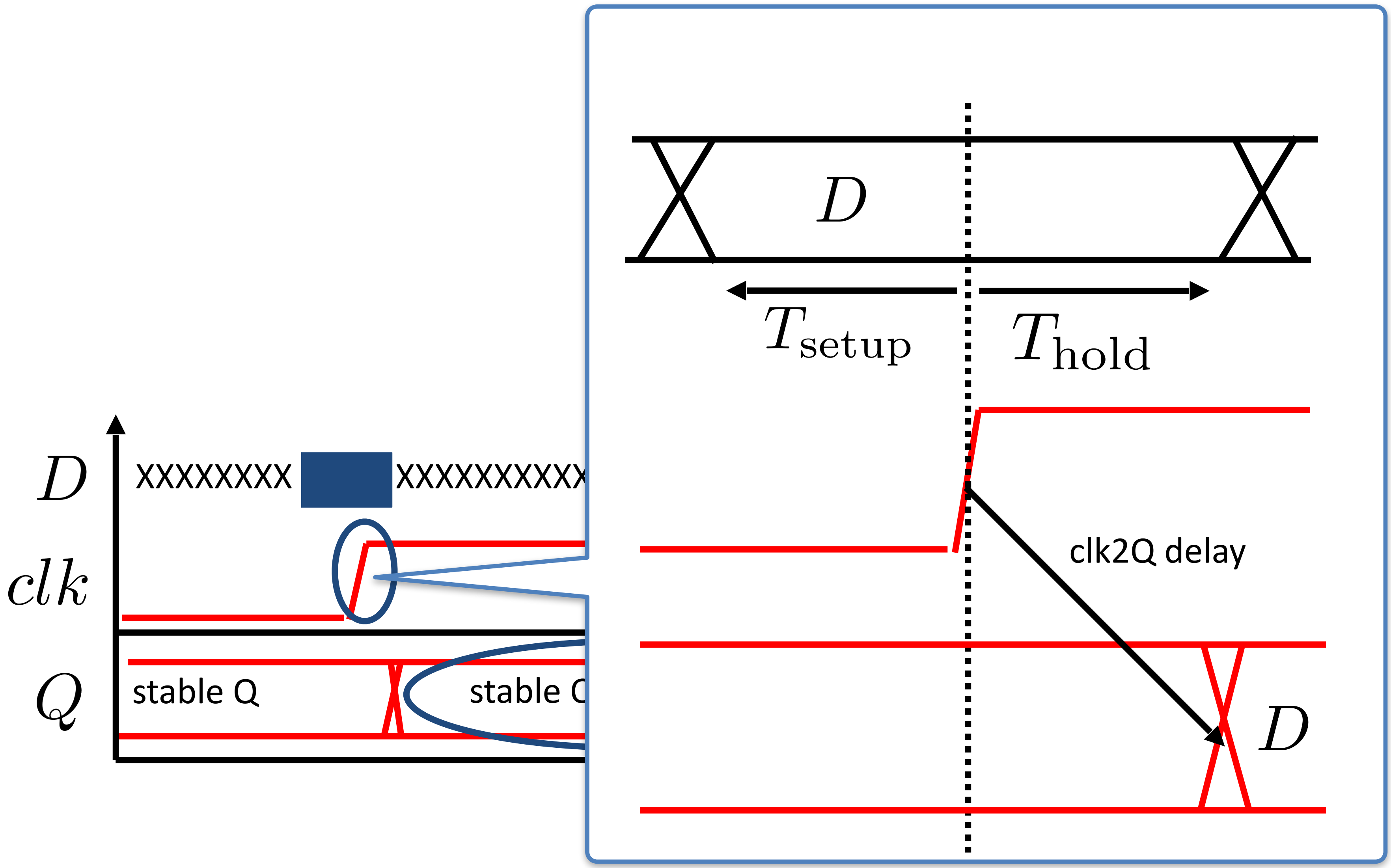



```
-- FF
FF: process (clk, D)
begin
    if (clk'event and clk = '1') then
        Q <= D;
    end if;
end process FF;
```

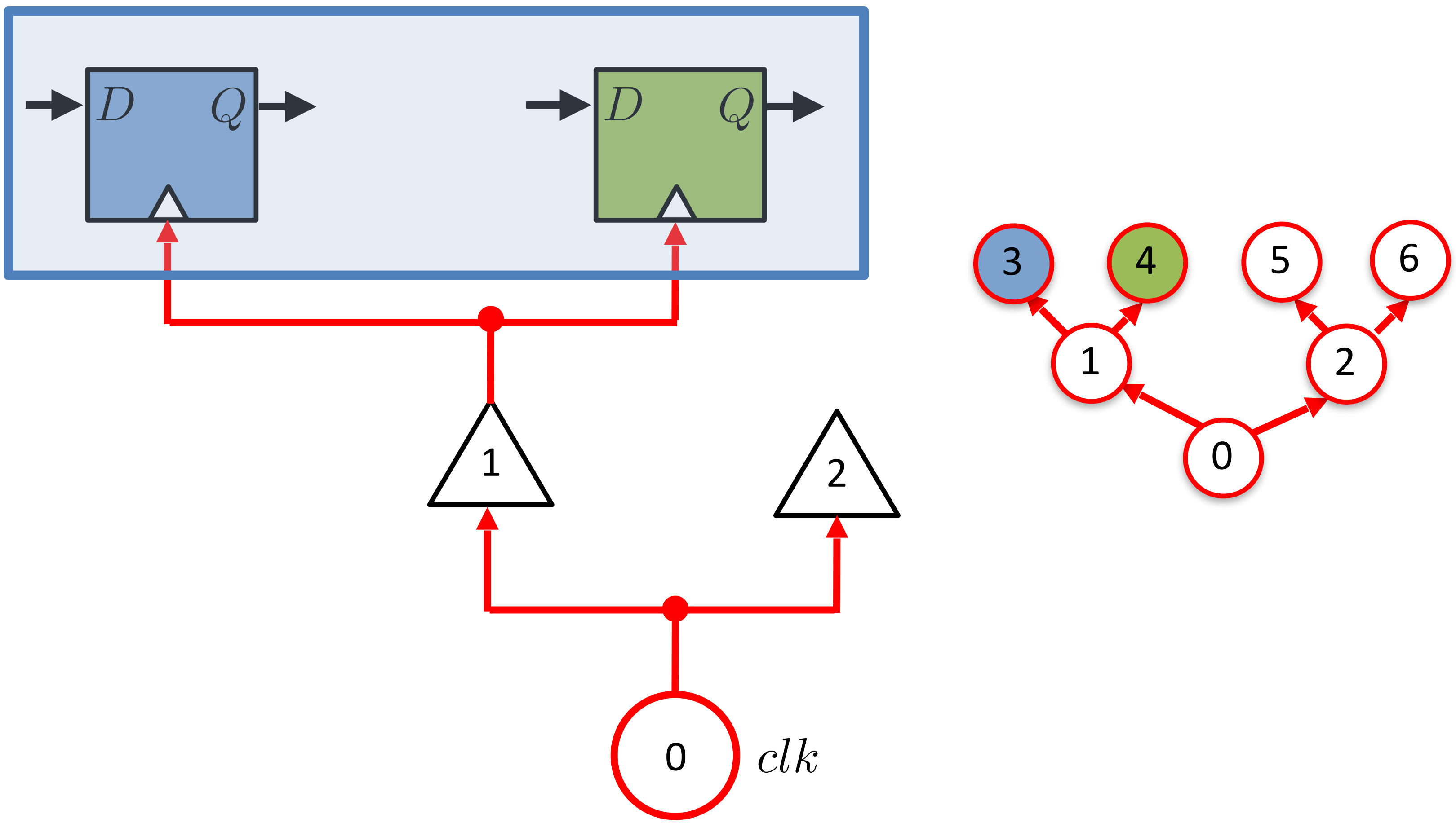
Timing: constraints



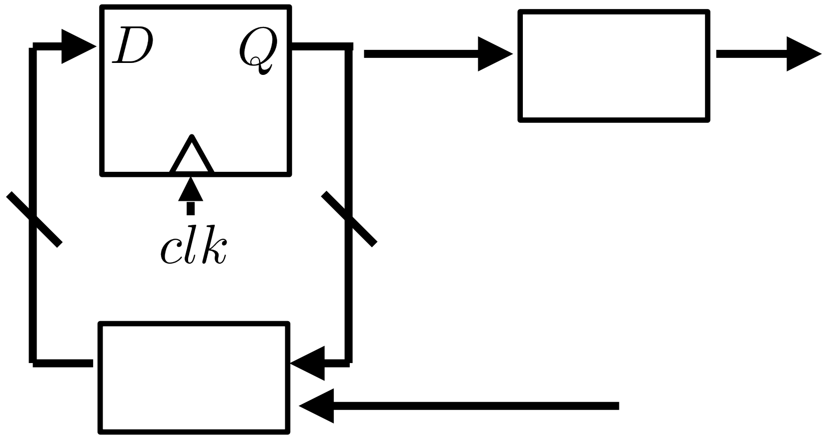
Timing: guarantees

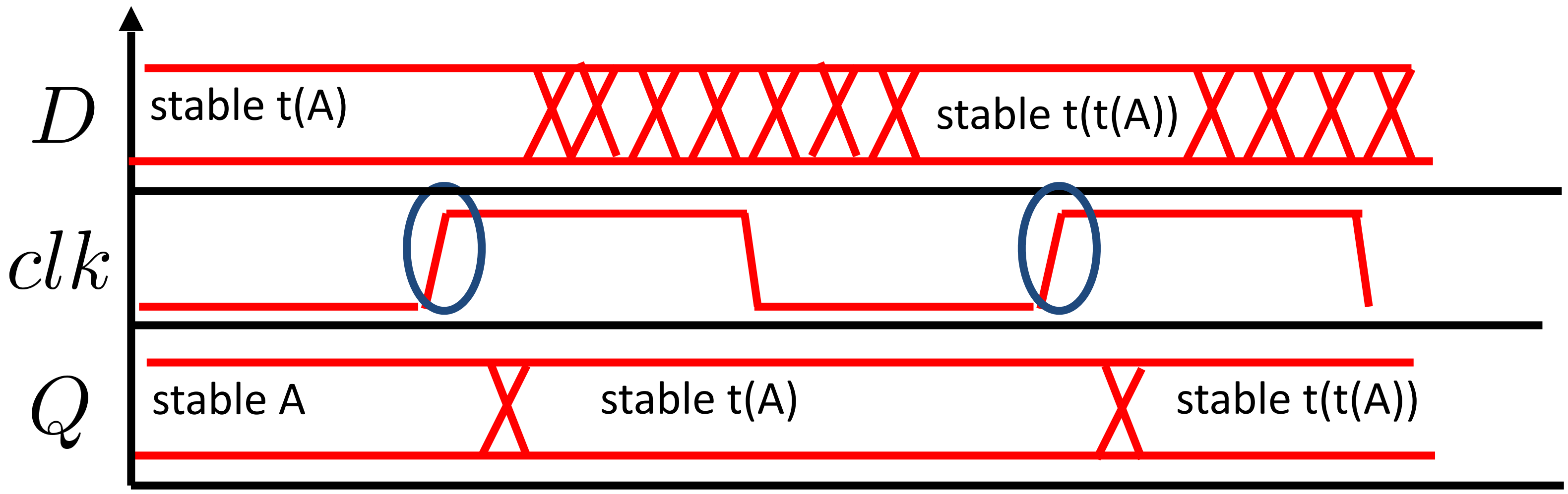


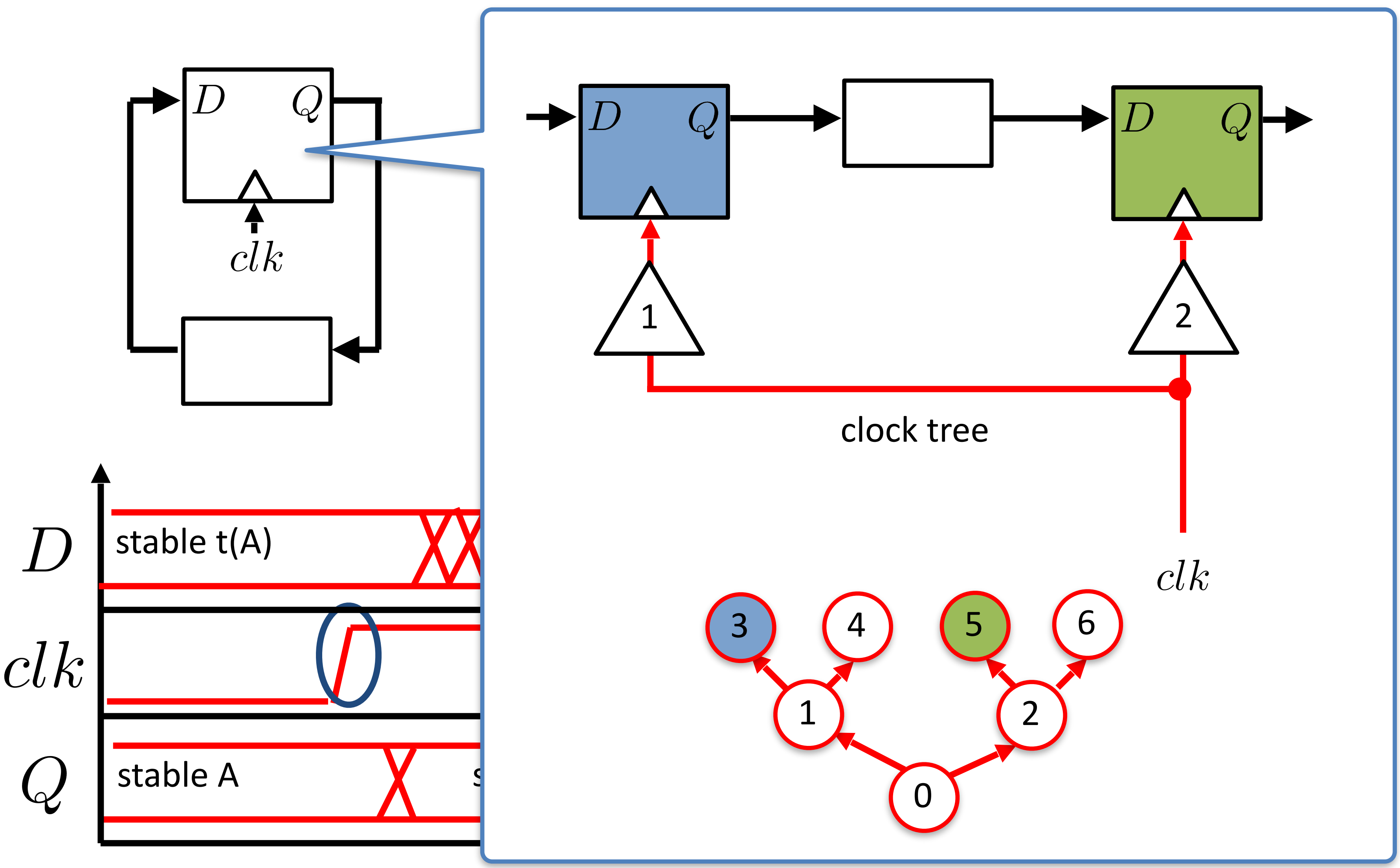
Goal: small skew



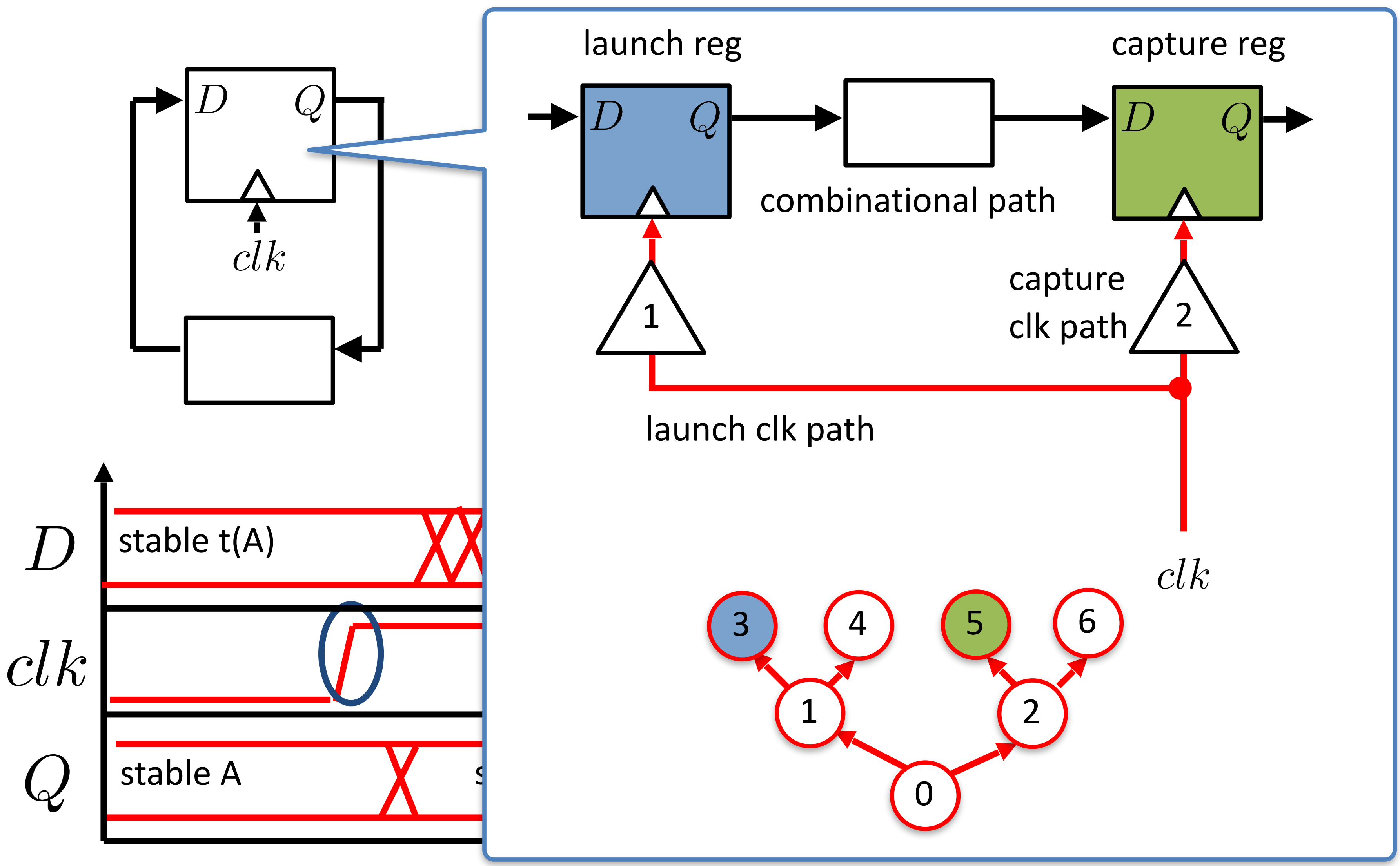
Clocked Design



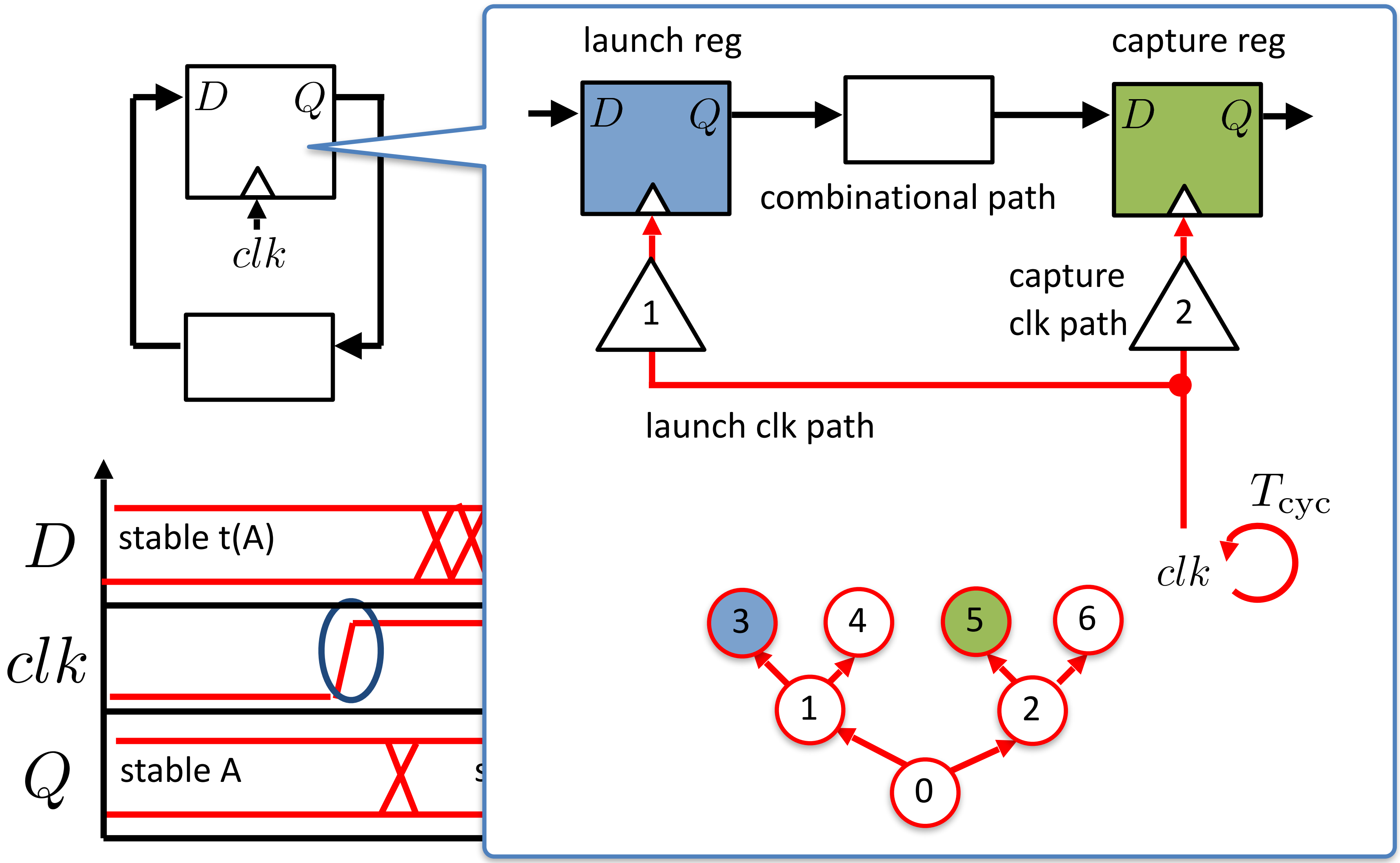


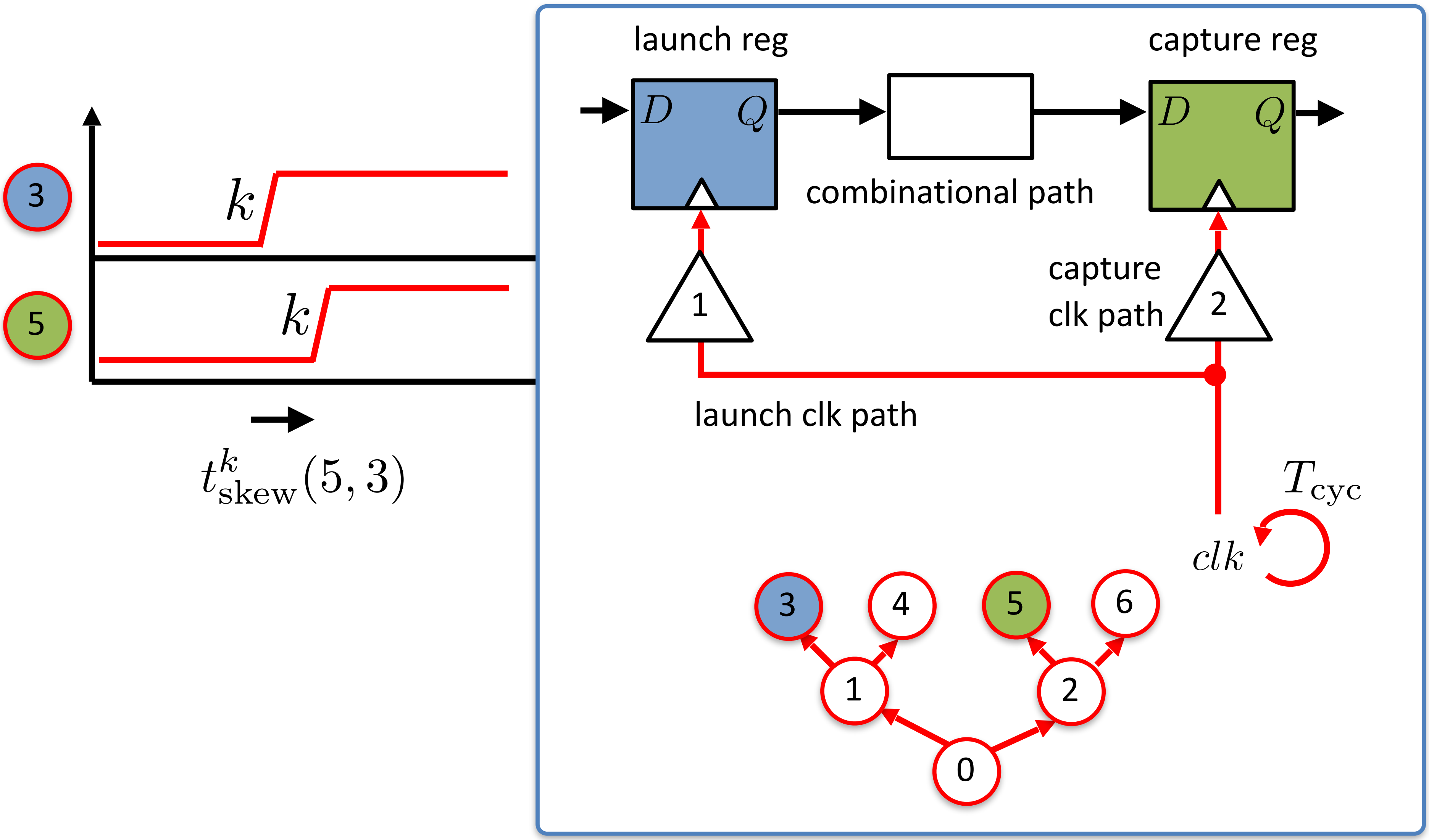


D
Q
clk
stable t(A)
clk
Q
stable A
D
Q
D
Q
1
2
clock tree
clk
3
4
5
6
1
2
0

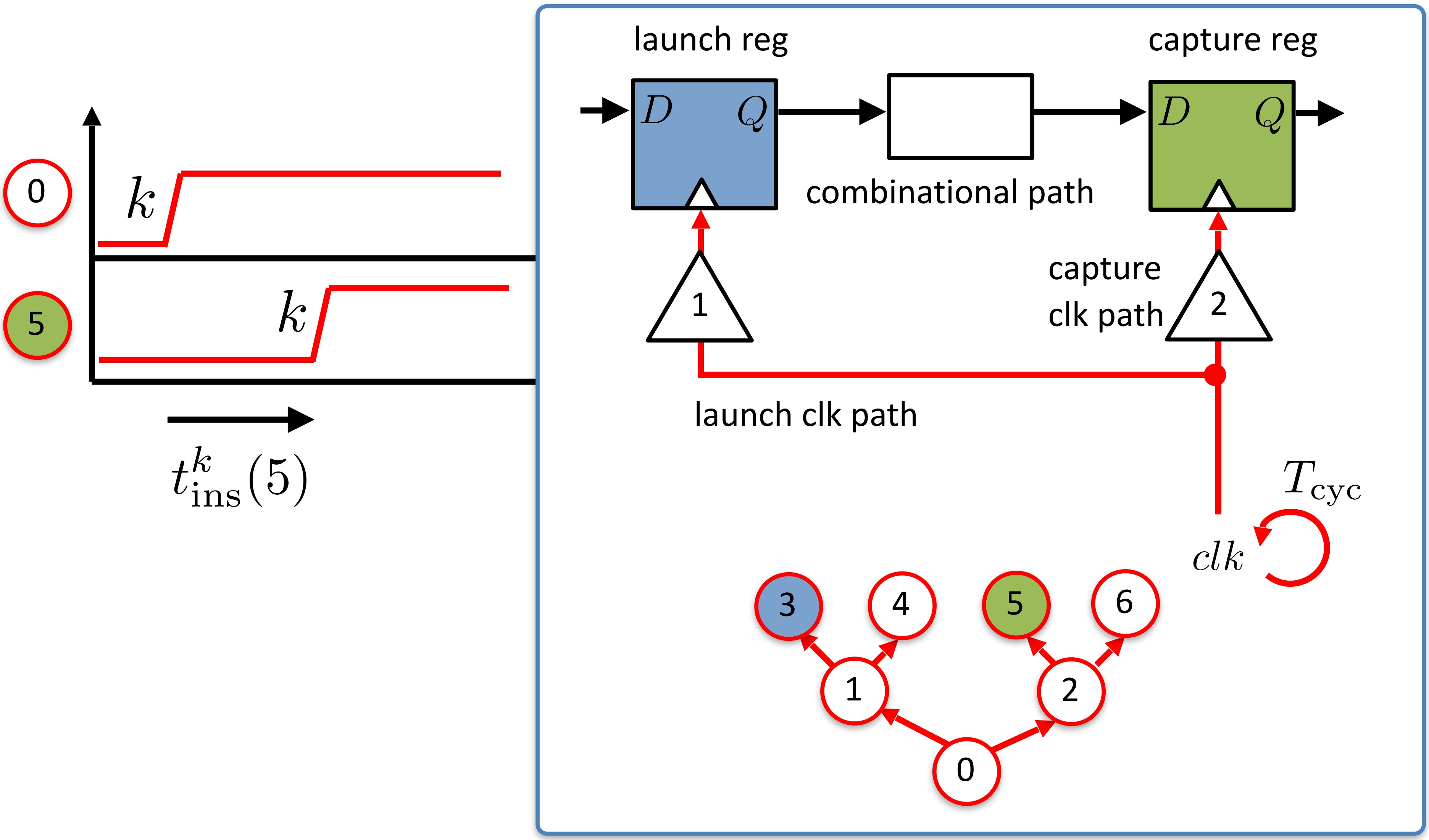


D
Q
clk
launch reg
capture reg
D
Q
D
Q
combinational path
1
capture
clk path
2
launch clk path
clk
3
4
5
6
1
2
0
D
stable t(A)
clk
Q
stable A

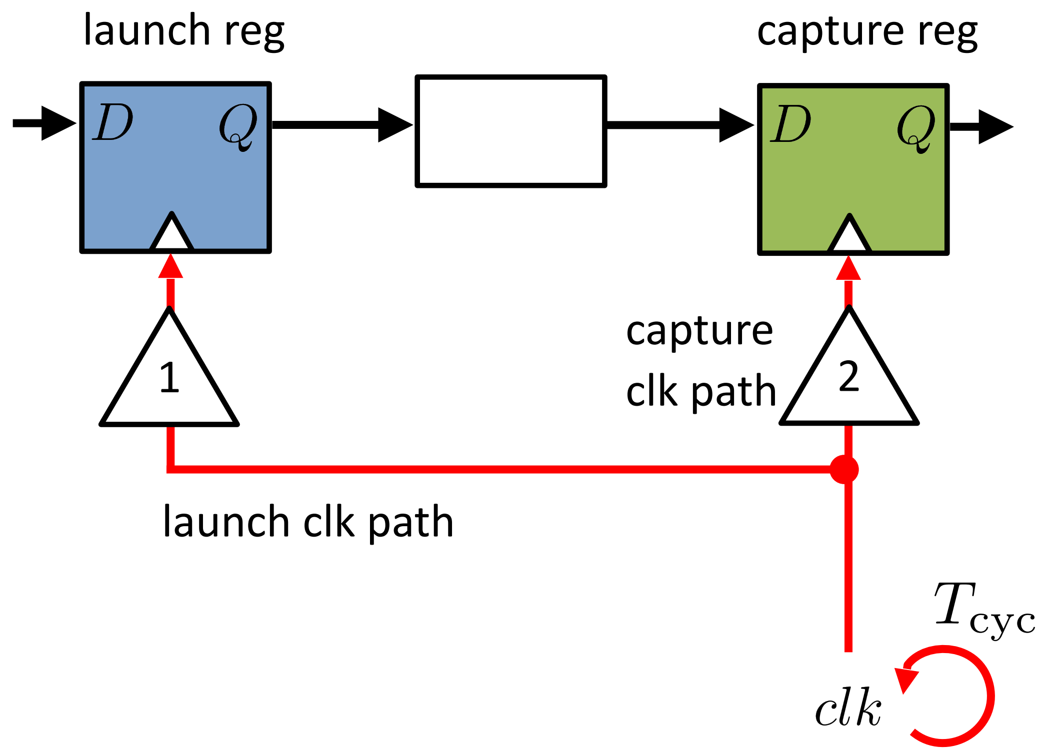

D
Q
clk
D
stable t(A)
clk
Q
stable A
launch reg
capture reg
D
Q
D
Q
combinational path
1
capture
clk path
2
launch clk path
T_{cyc}
clk
3
4
5
6
1
2
0

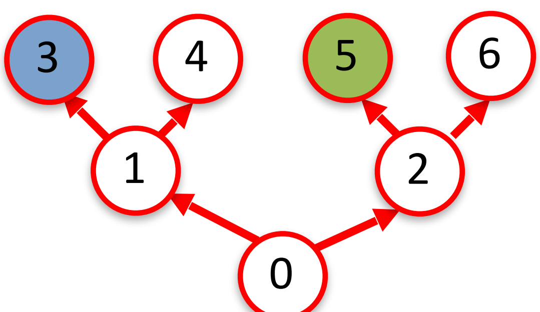


3
5
k
k
$t^k_{\text{skew}}(5,3)$
launch reg
capture reg
D
Q
D
Q
combinational path
1
2
capture
clk path
launch clk path
T_{cyc}
clk
3
4
5
6
1
2
0

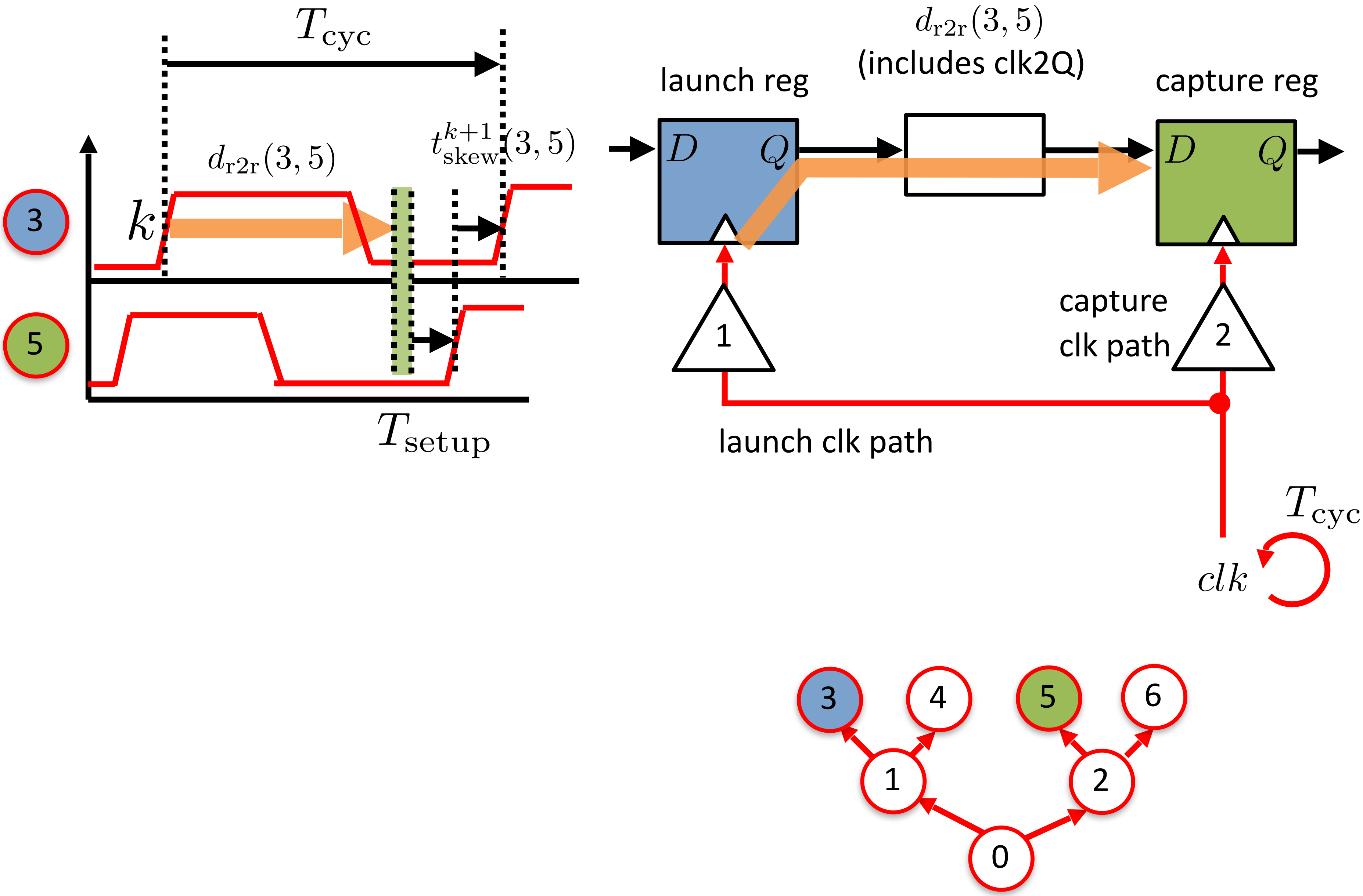


0
5
k
k
$t^k_{\text{ins}}(5)$
launch reg
capture reg
D
Q
D
Q
combinational path
1
capture
clk path
2
launch clk path
T_{cyc}
clk
3
4
5
6
1
2
0

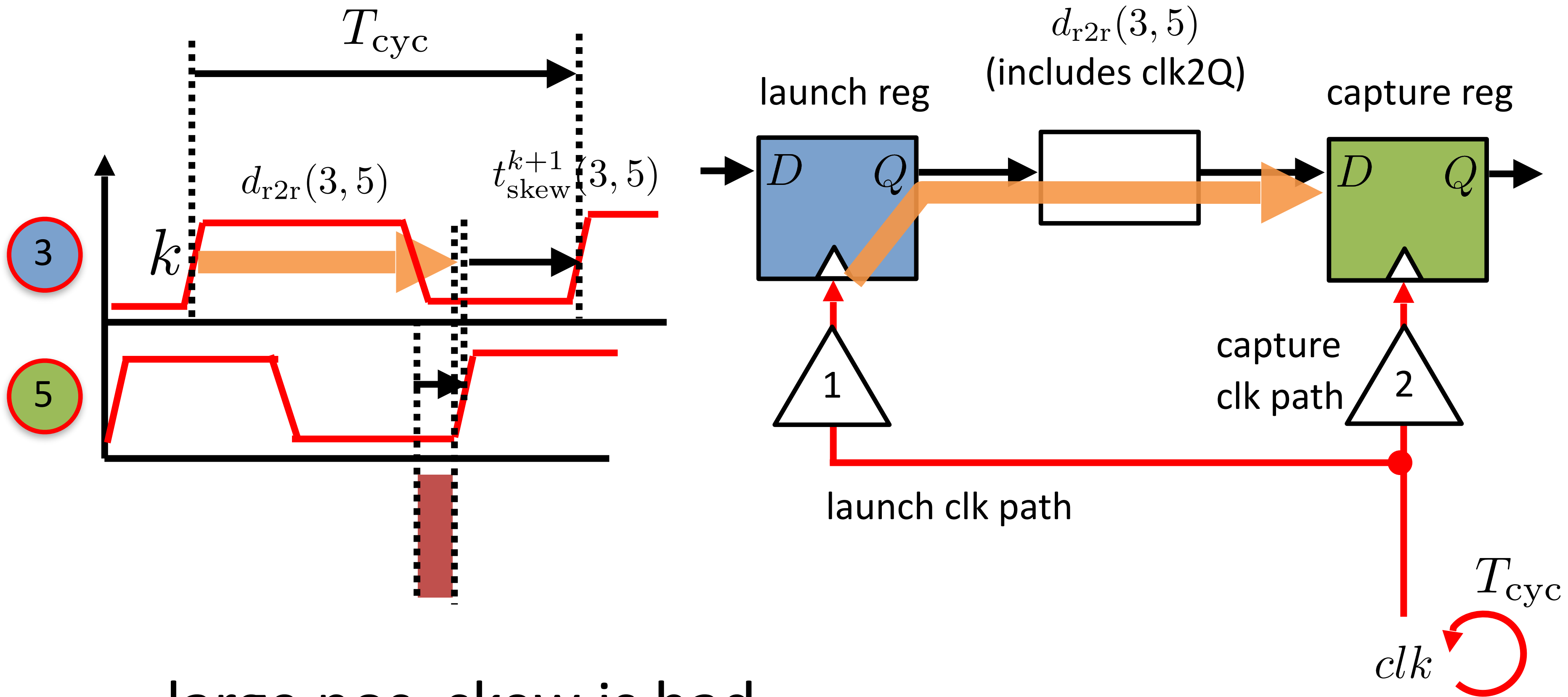

launch reg
capture reg
D
Q
D
Q
1
2
capture
clk path
launch clk path
T_{cyc}
clk


3
4
5
6
1
2
0

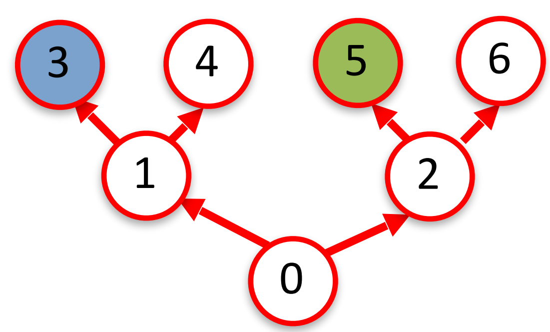
The setup constraint



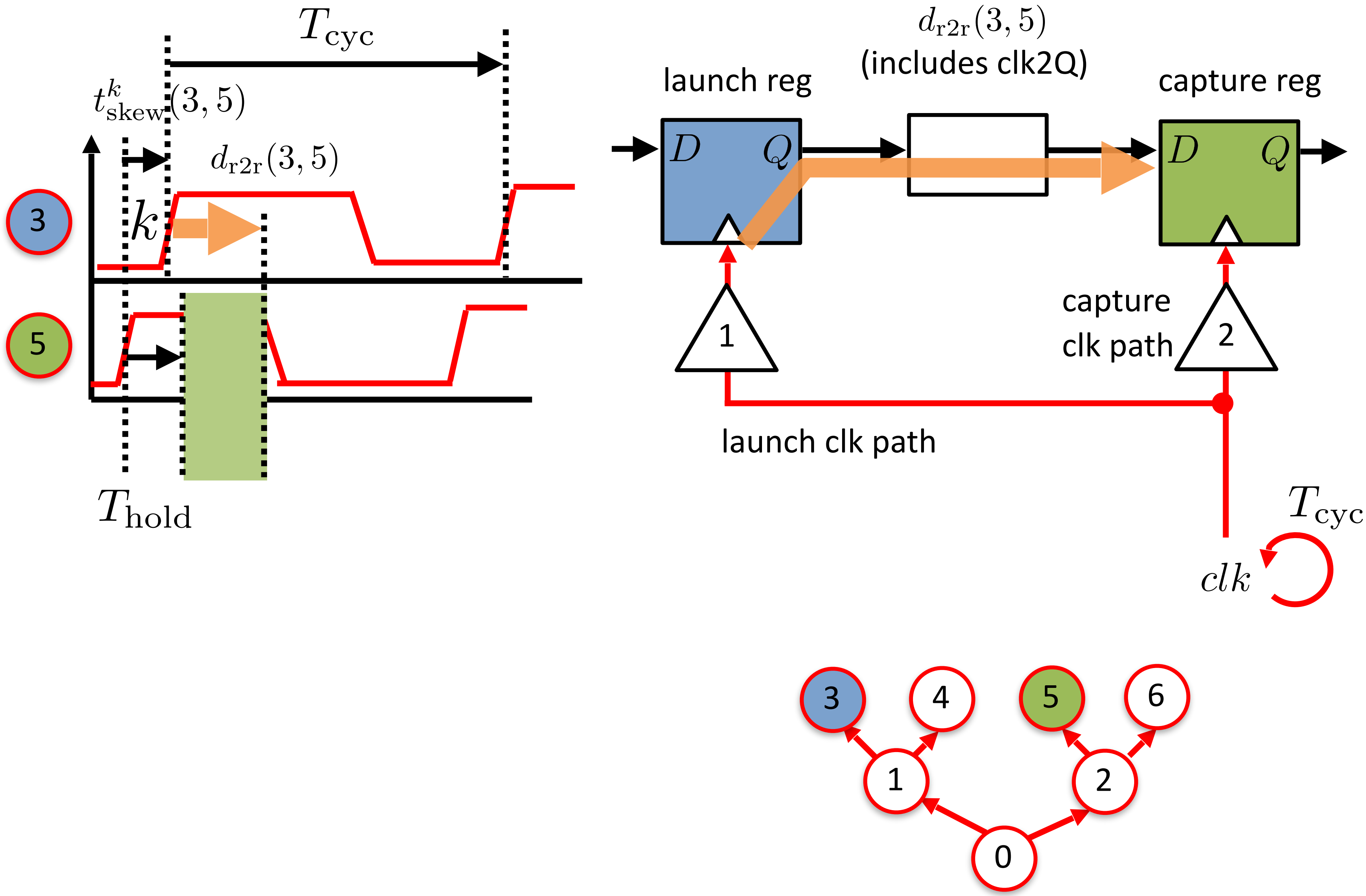
The setup constraint



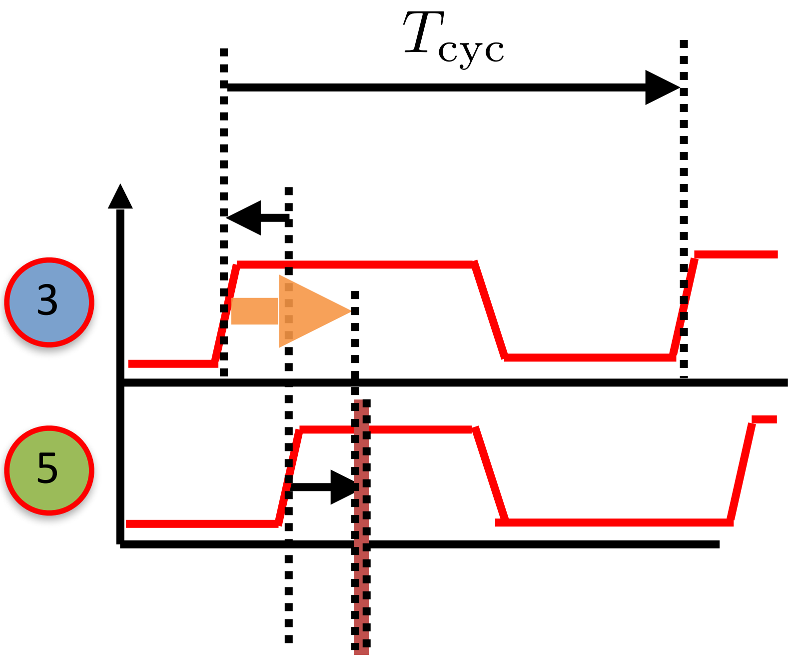
large pos. skew is bad

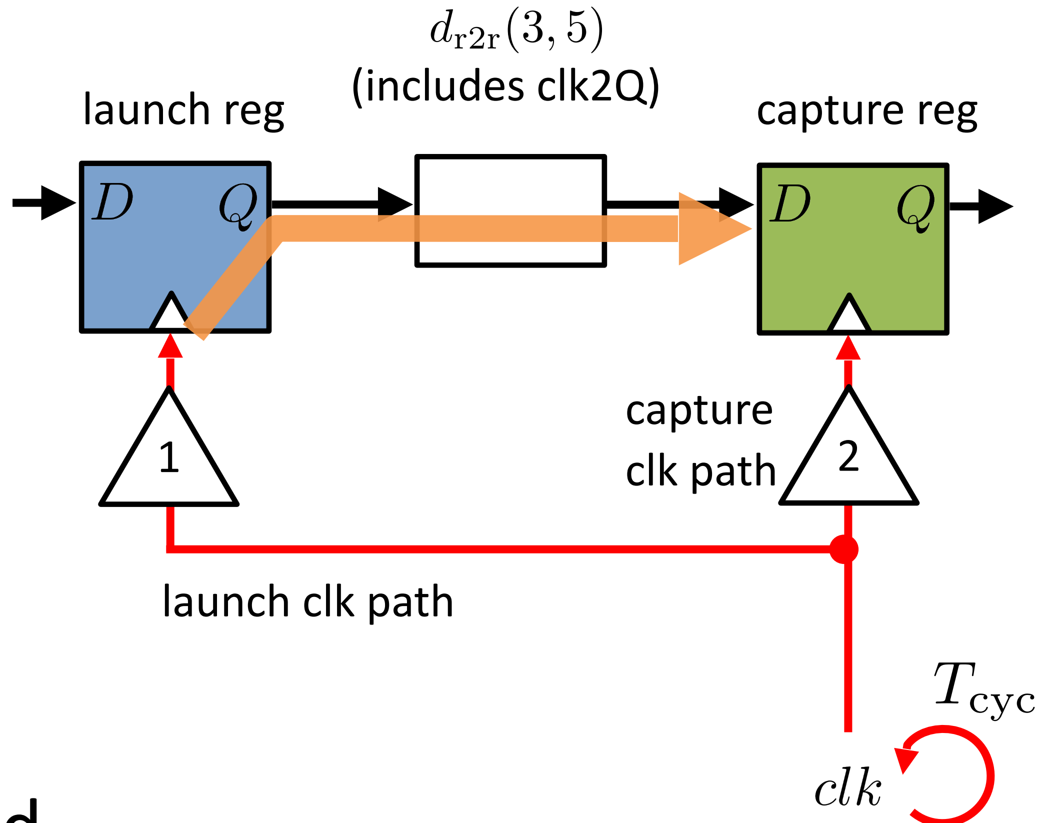


The hold constraint

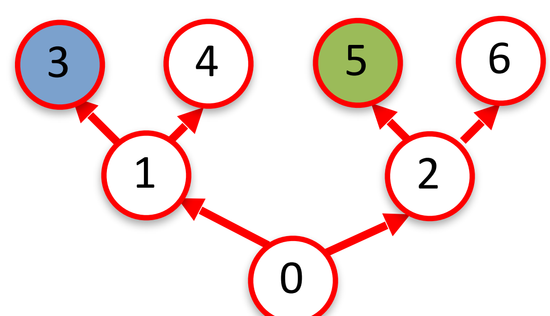


The hold constraint

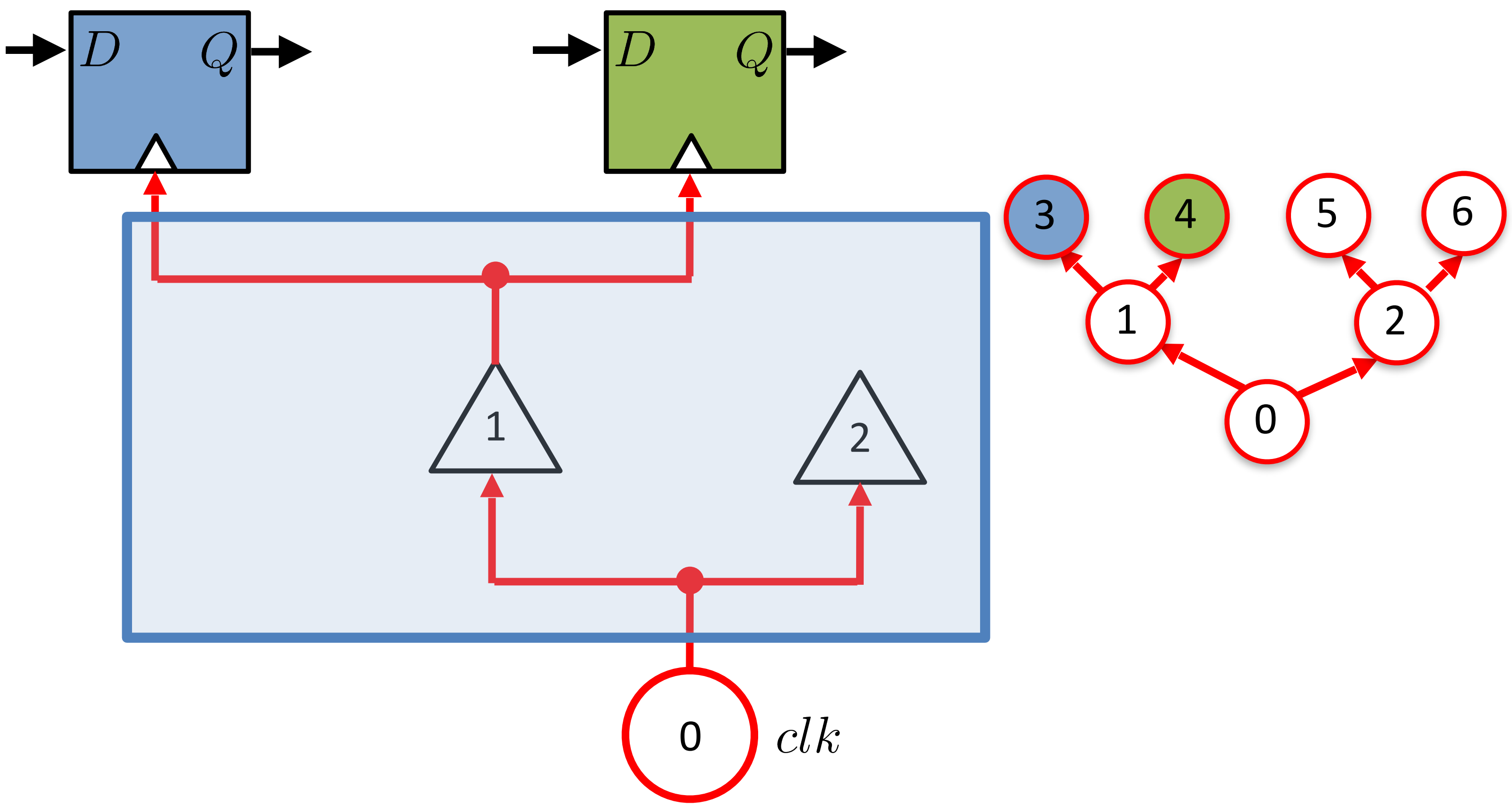




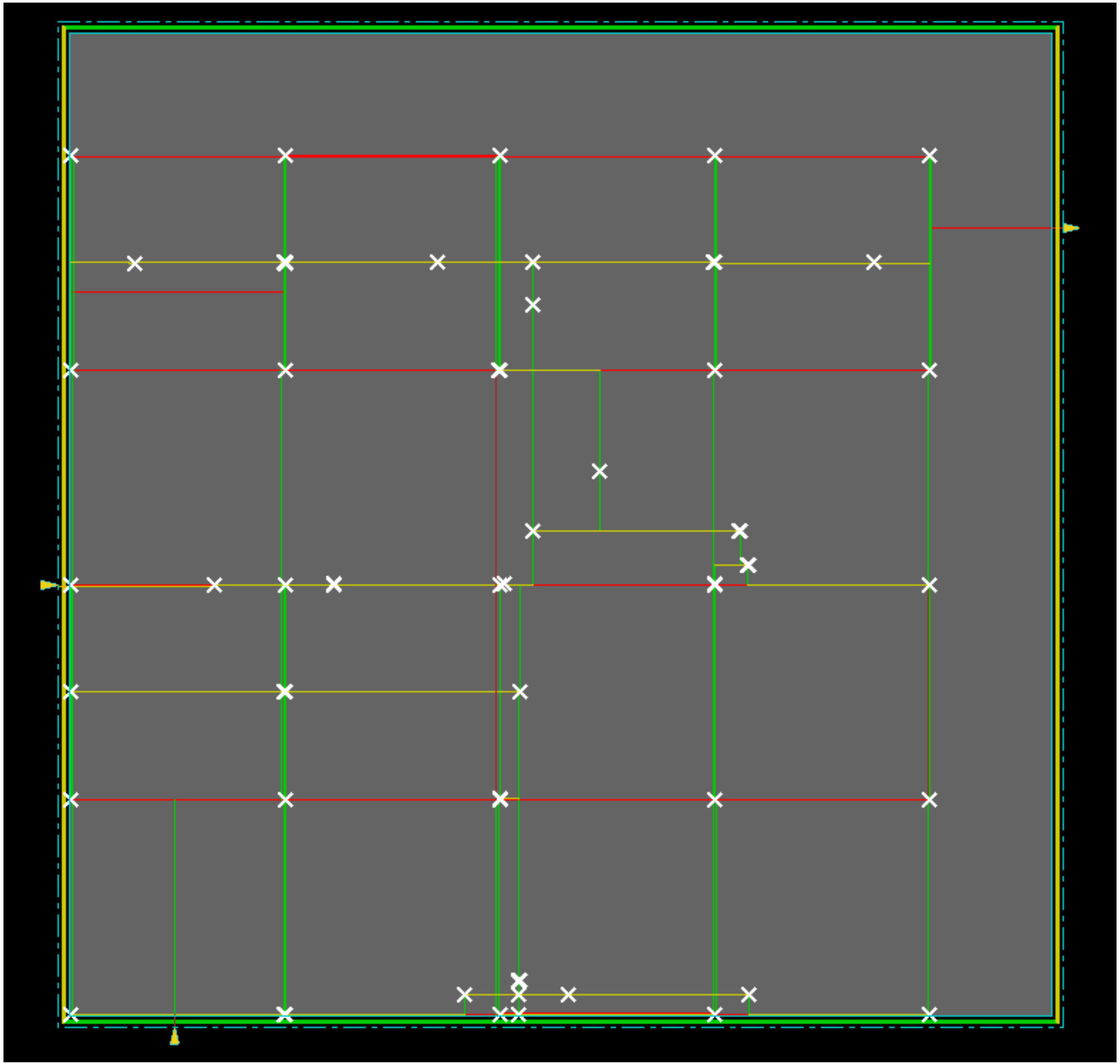
large neg. skew is bad



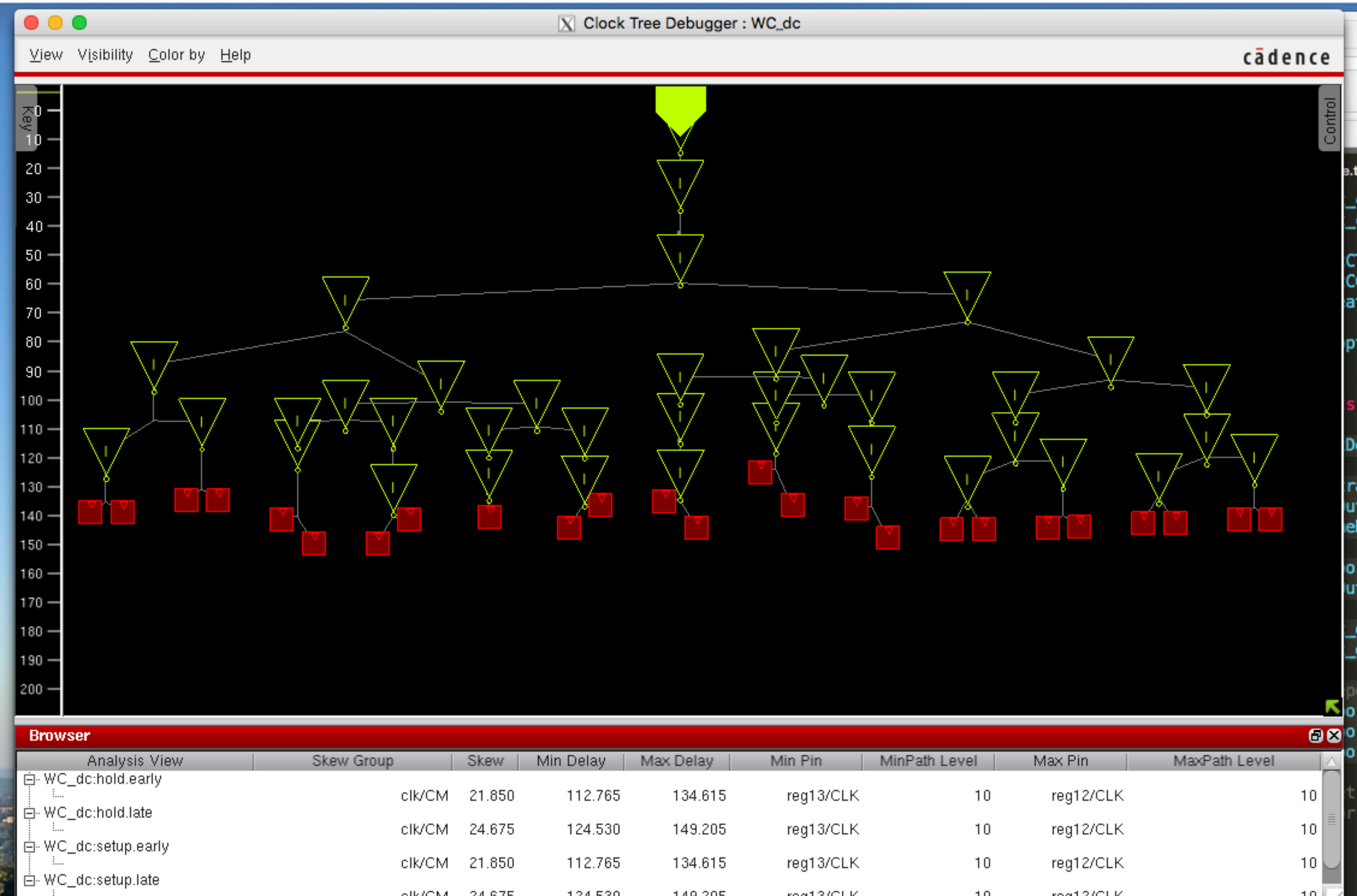
Clock distribution network



The clock

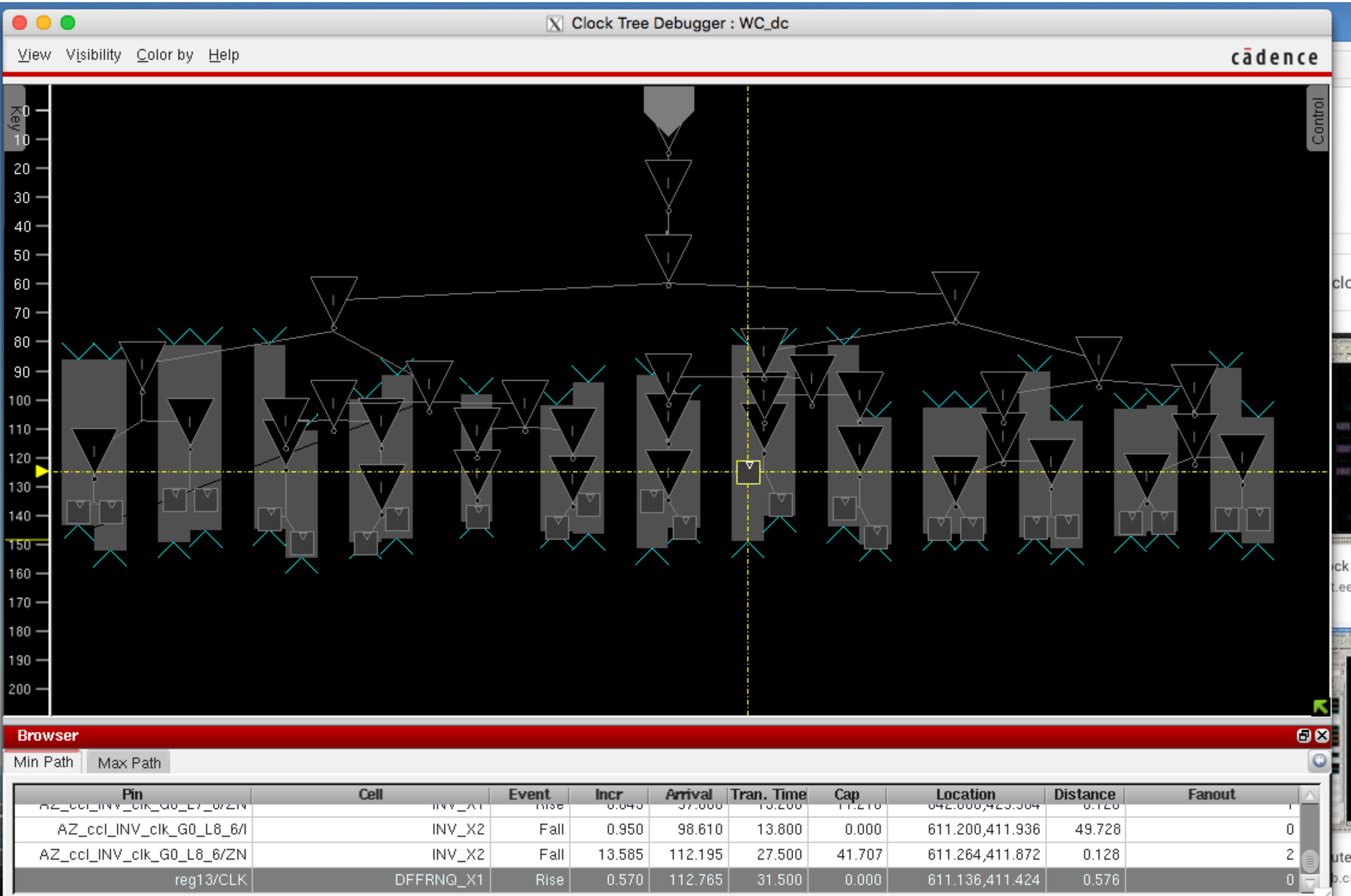


Its skew

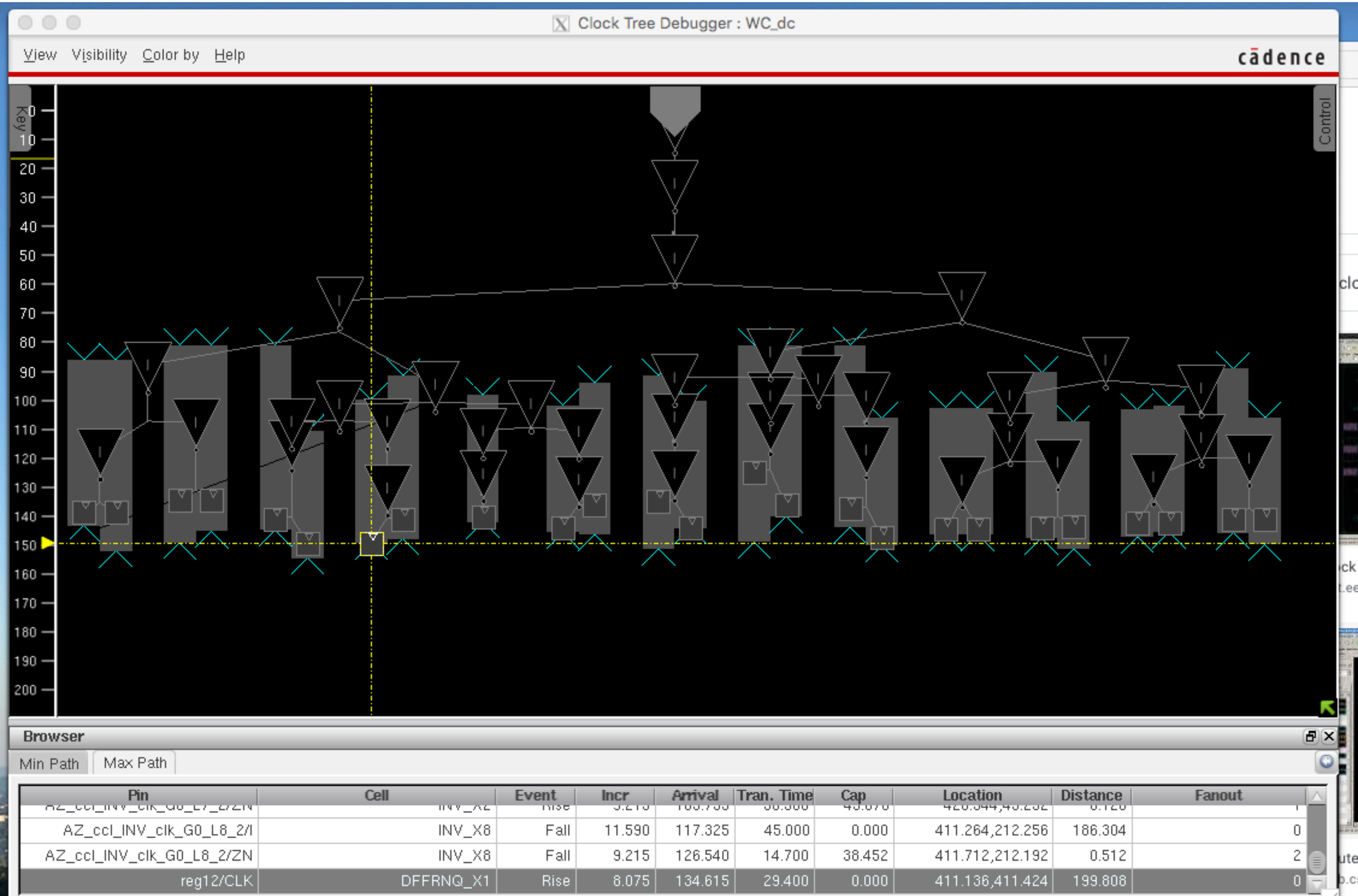


Clock Tree Debugger : WC_dc
View Visibility Color by Help
cādence
Key
Control
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
Browser
Analysis View
Skew Group
Skew
Min Delay
Max Delay
Min Pin
MinPath Level
Max Pin
MaxPath Level
WC_dc:hold.early
clk/CM 21.850 112.765 134.615 reg13/CLK 10 reg12/CLK 10
WC_dc:hold.late
clk/CM 24.675 124.530 149.205 reg13/CLK 10 reg12/CLK 10
WC_dc:setup.early
clk/CM 21.850 112.765 134.615 reg13/CLK 10 reg12/CLK 10
WC_dc:setup.late

Minimum path

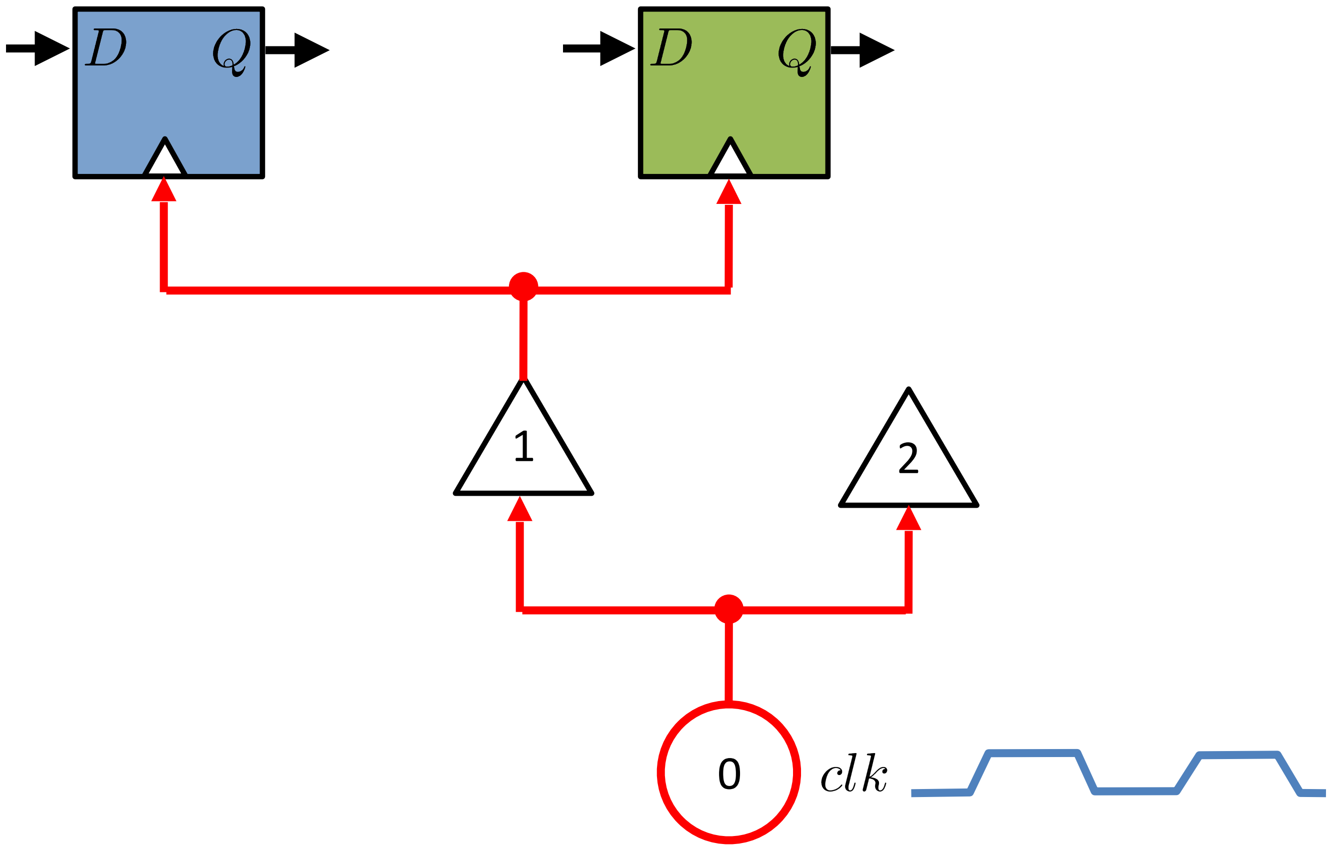


Maximum path

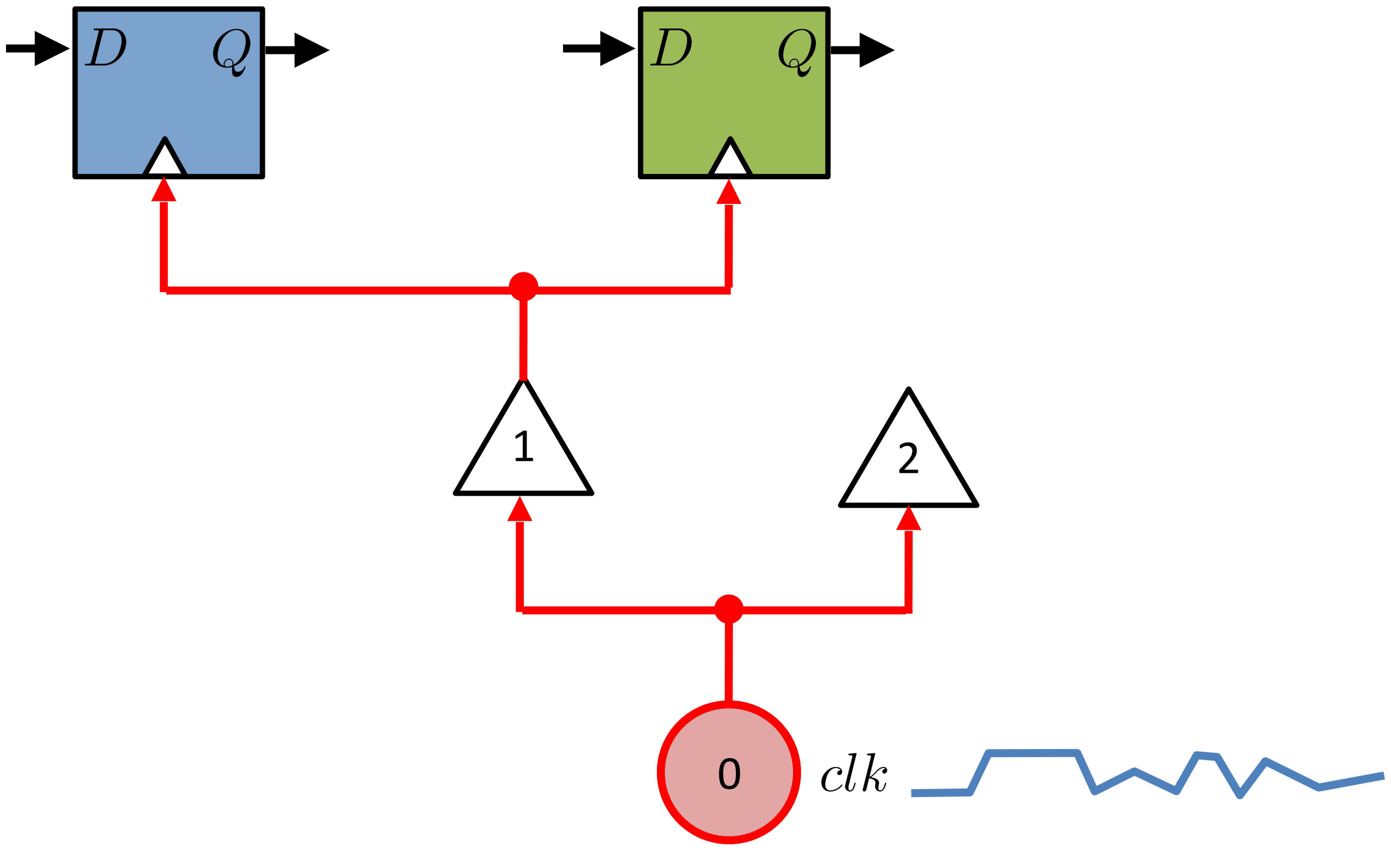


Pin	Cell	Event	Incr	Arrival	Tran. Time	Cap	Location	Distance	Fanout
AZ_ccl_INV_clk_G0_L8_2/I	INV_X8	Fall	11.590	117.325	45.000	0.000	411.264,212.256	186.304	0
AZ_ccl_INV_clk_G0_L8_2/ZN	INV_X8	Fall	9.215	126.540	14.700	38.452	411.712,212.192	0.512	2
reg12/CLK	DFFRNQ_X1	Rise	8.075	134.615	29.400	0.000	411.136,411.424	199.808	0

But ...

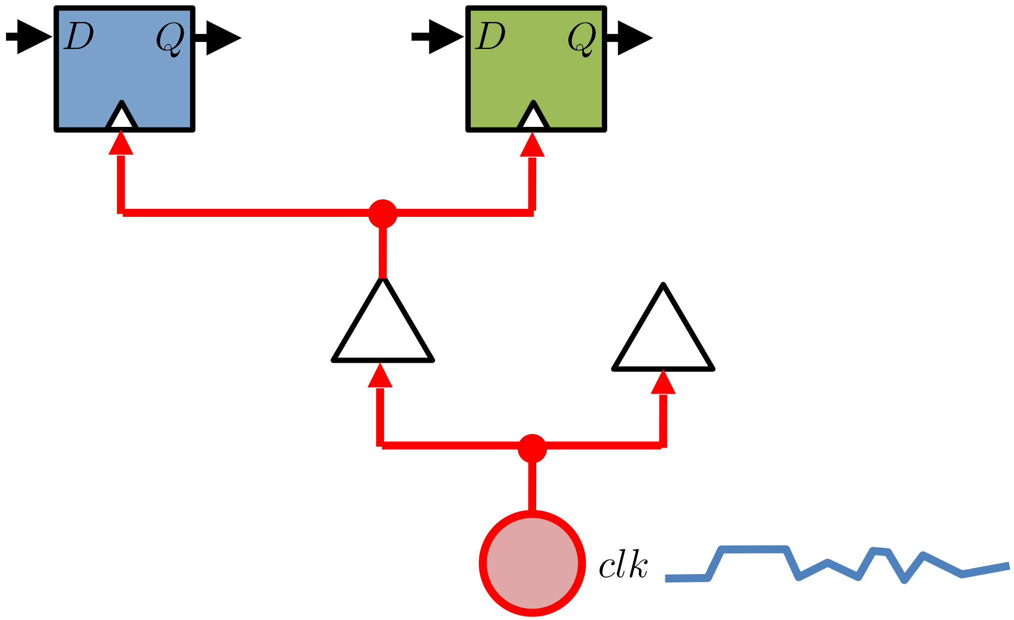


... faults

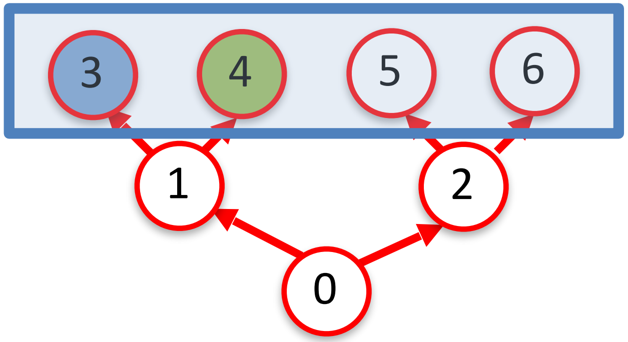


New requirements

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* $\mathcal{S}, P_{\max}, P_{\min} \in \mathbb{R}_{>0}$ *satisfying*

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

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

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

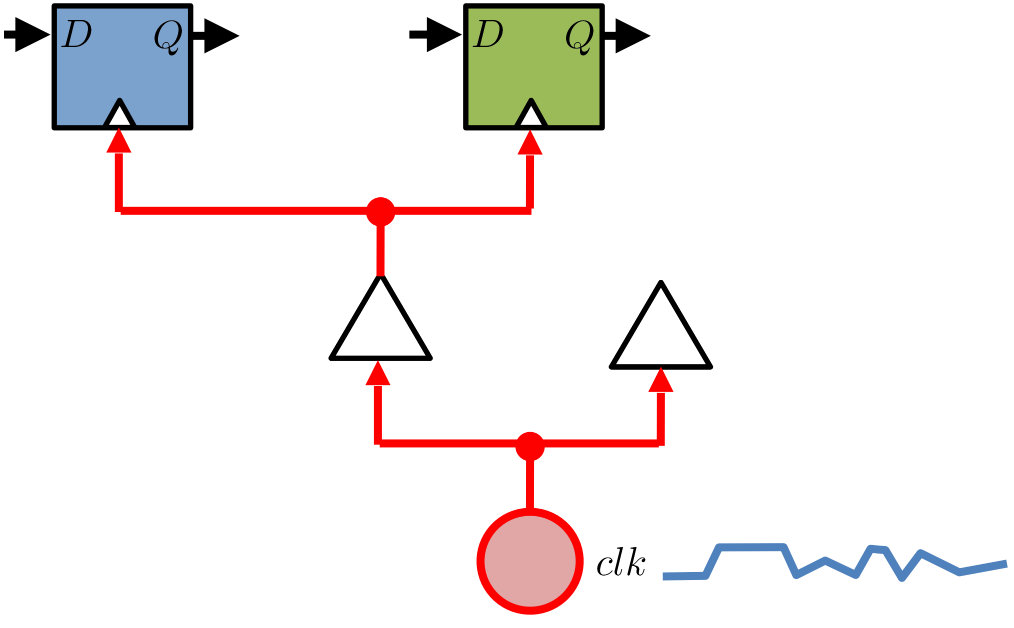
Lower bounds

Number of faults f. Then necessarily:

Global: $n > 3f$

Local: degree $> 2f$

Ideas?

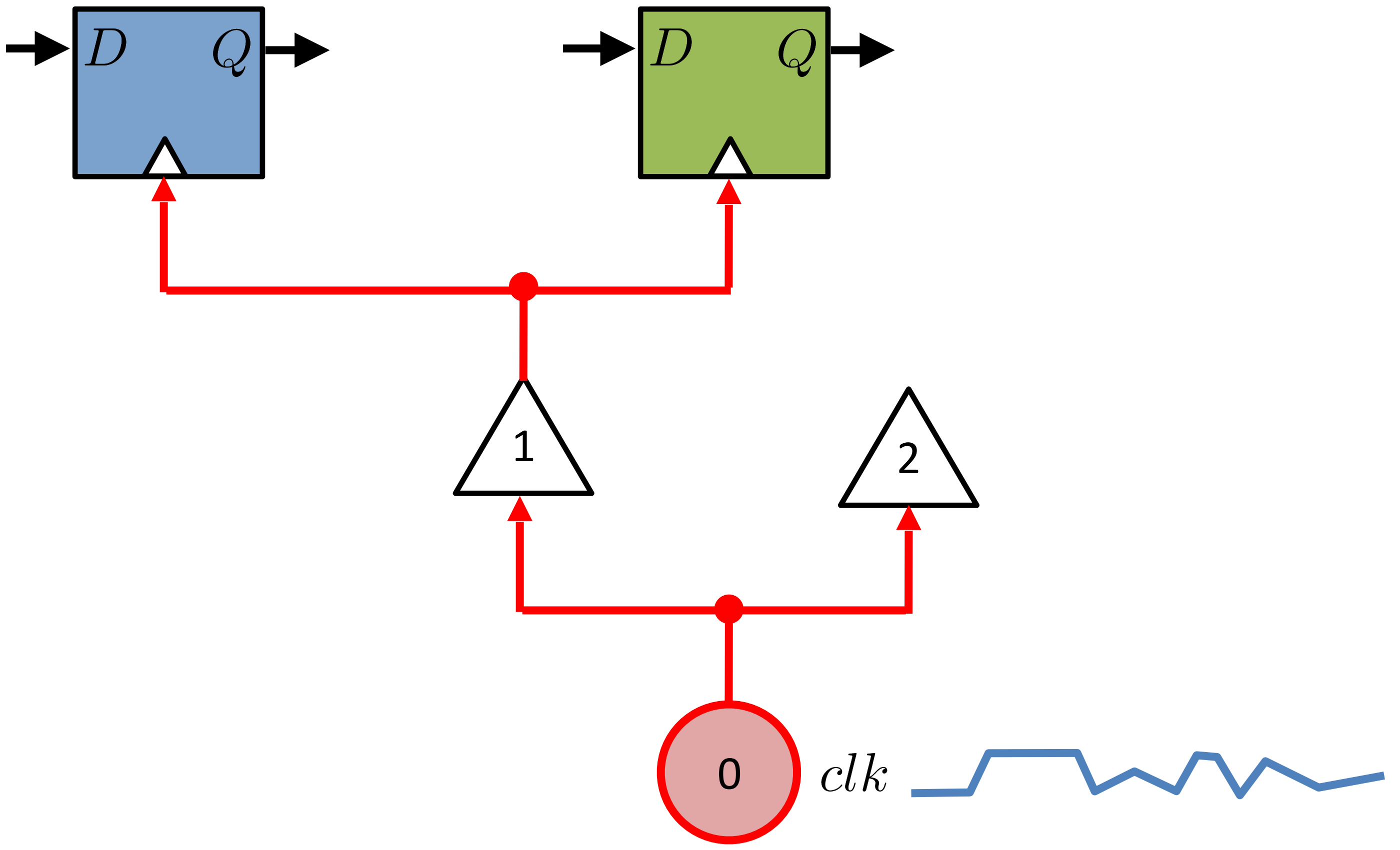


Chapter 11 (2)

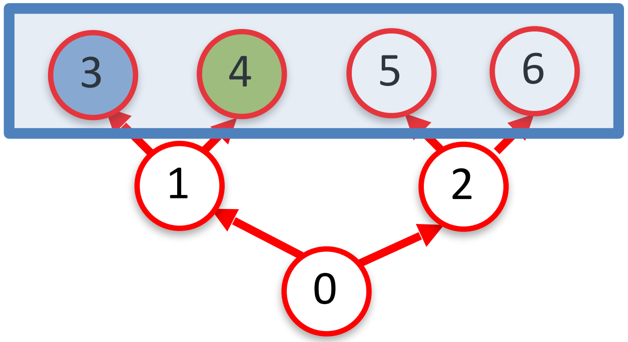
Low-degree clock distribution networks

Matthias Fuegger and Christoph Lenzen

Faults



Pulse Synchronization



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

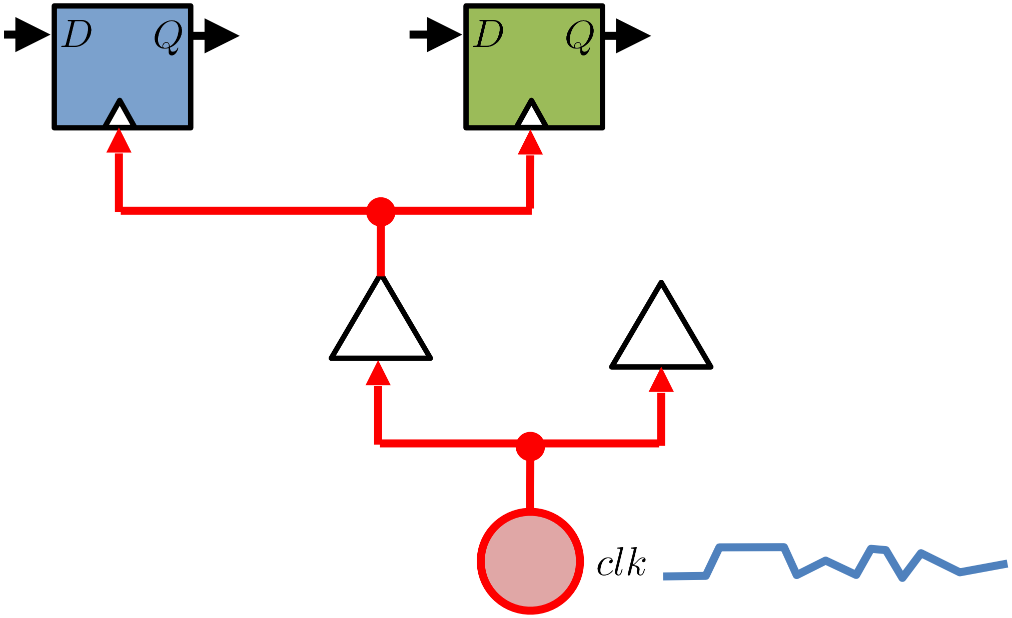
Let $p_{v,i}$ *denote the time when* v *generates the i-th pulse*. *We require that there are* $\mathcal{S}, P_{\max}, P_{\min} \in \mathbb{R}_{>0}$ *satisfying*

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

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

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

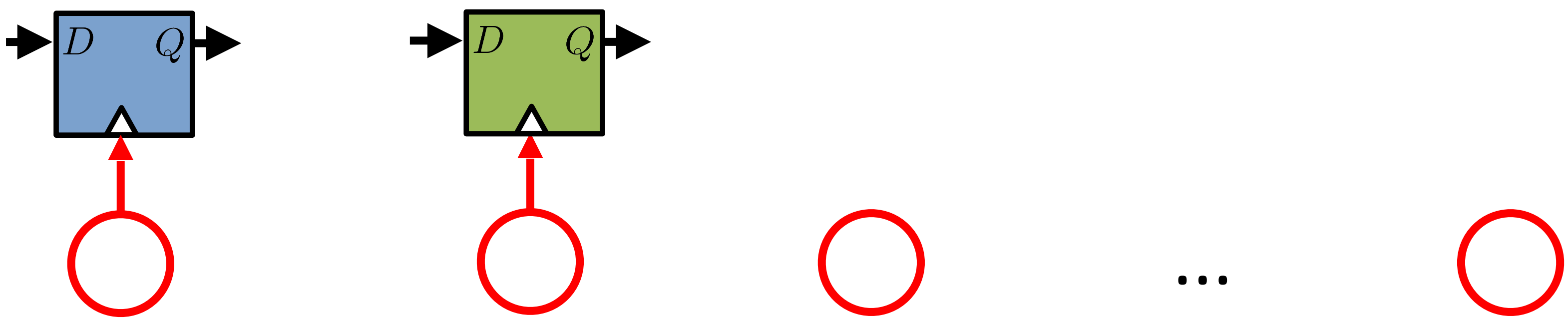
Last time: ideas session



Idea 1: only LW algorithm

no tree, only LW

3f+1

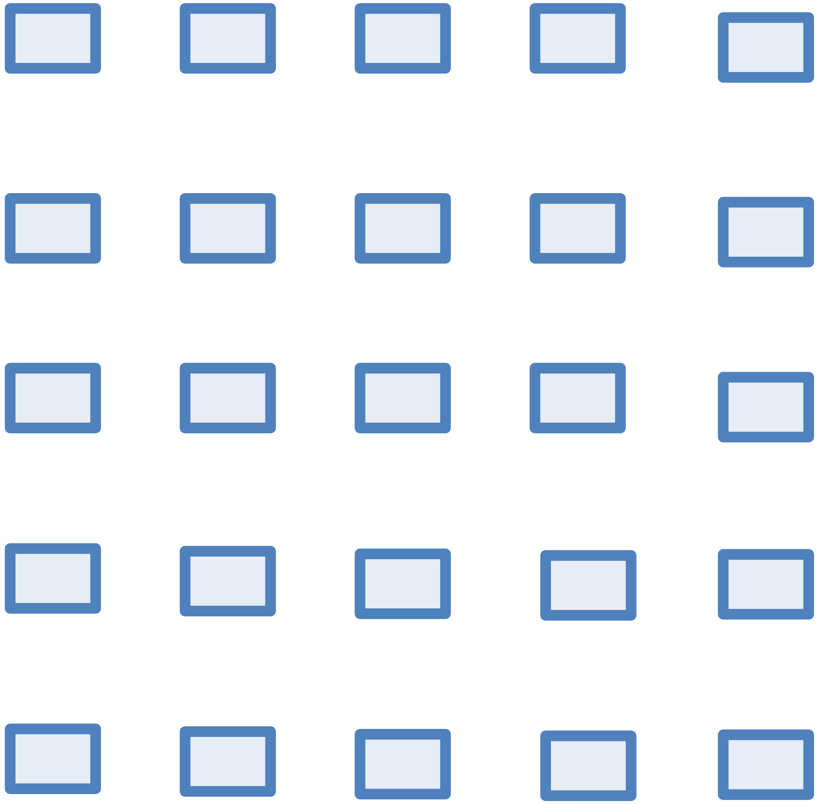


Idea 1

properties:

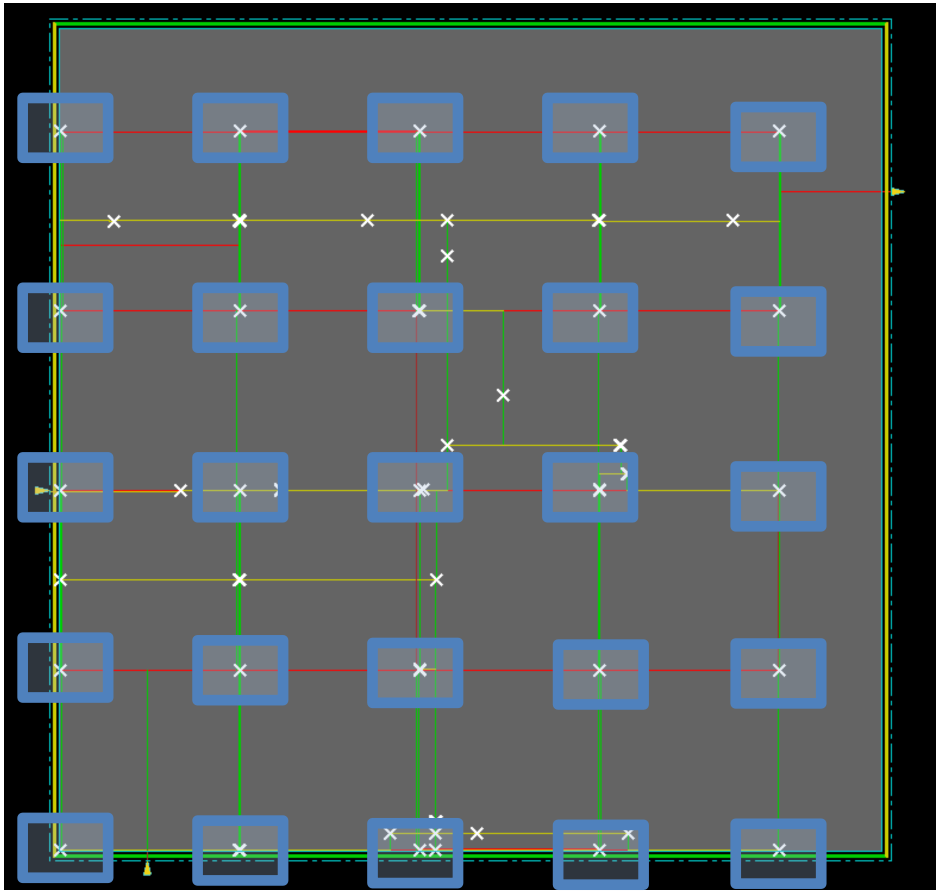
- **fault tolerance?**
- cost?
- skew?

5x5 grid



Idea 1

5x5 grid

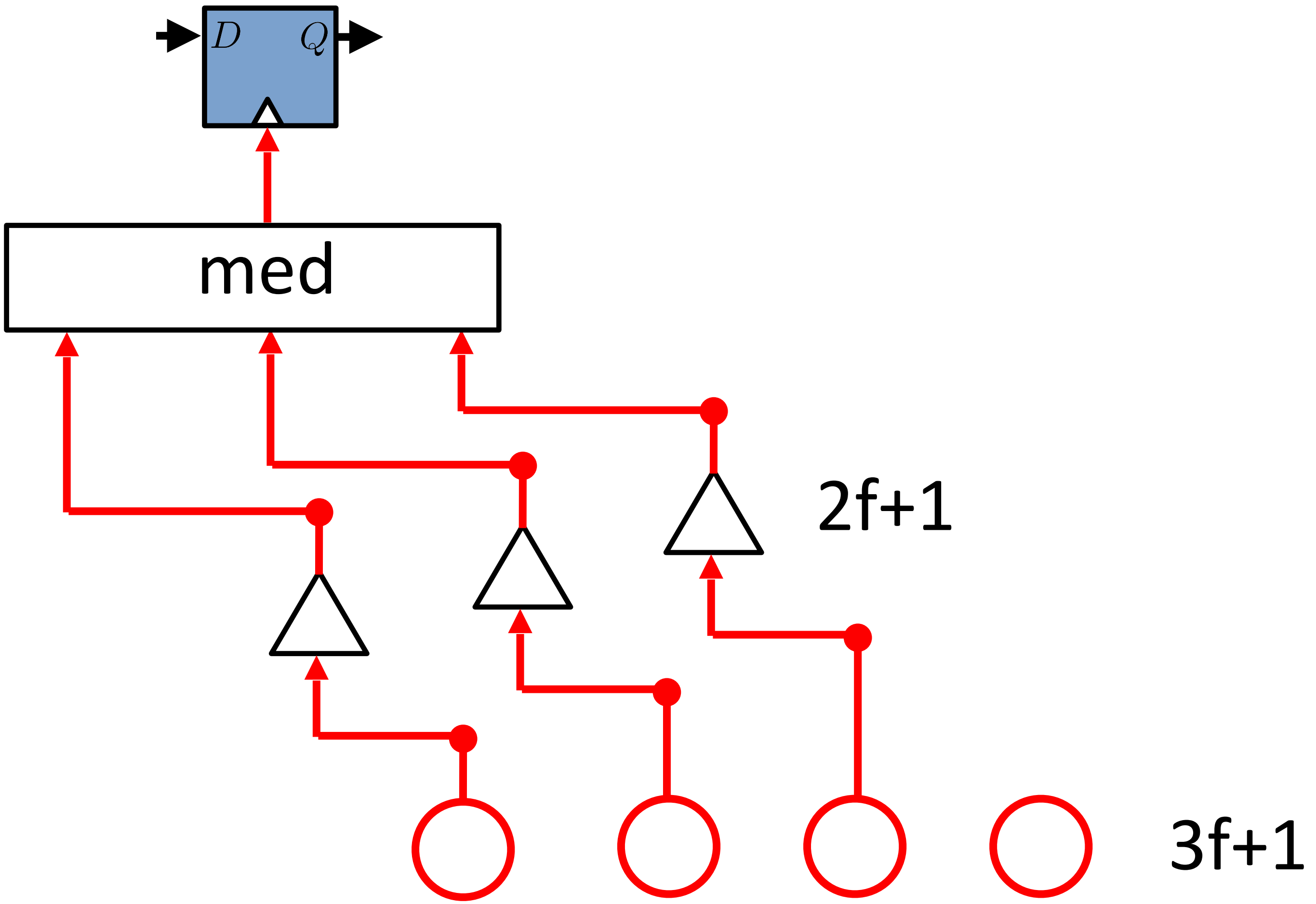


properties:

- fault tolerance?
- cost?
- skew?

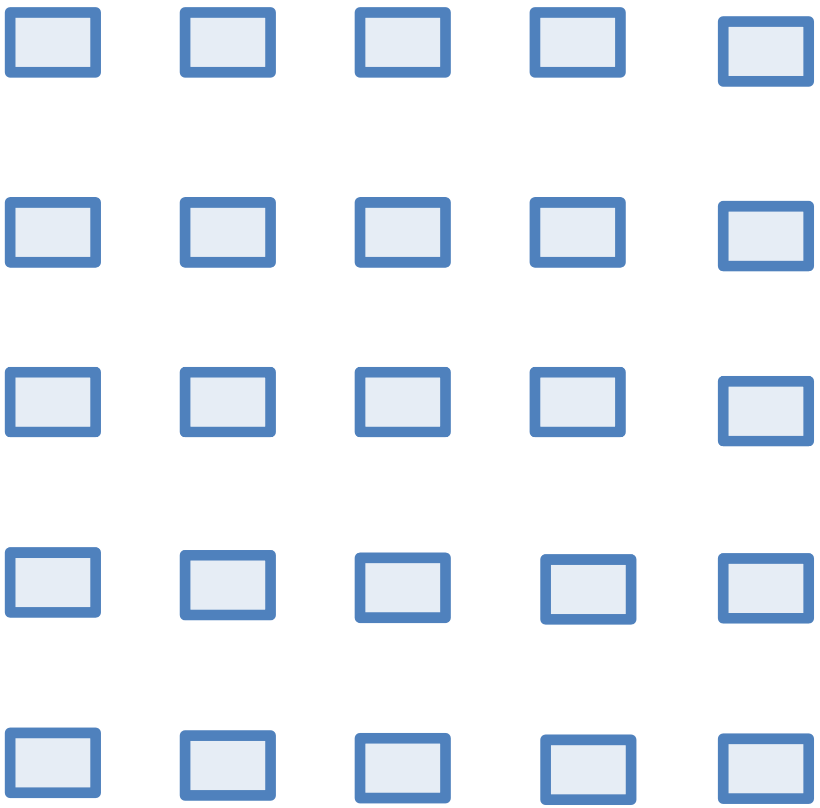
Idea 2: redundant trees

replicate the clock source, vote on output



Idea 2

5x5 grid



properties:

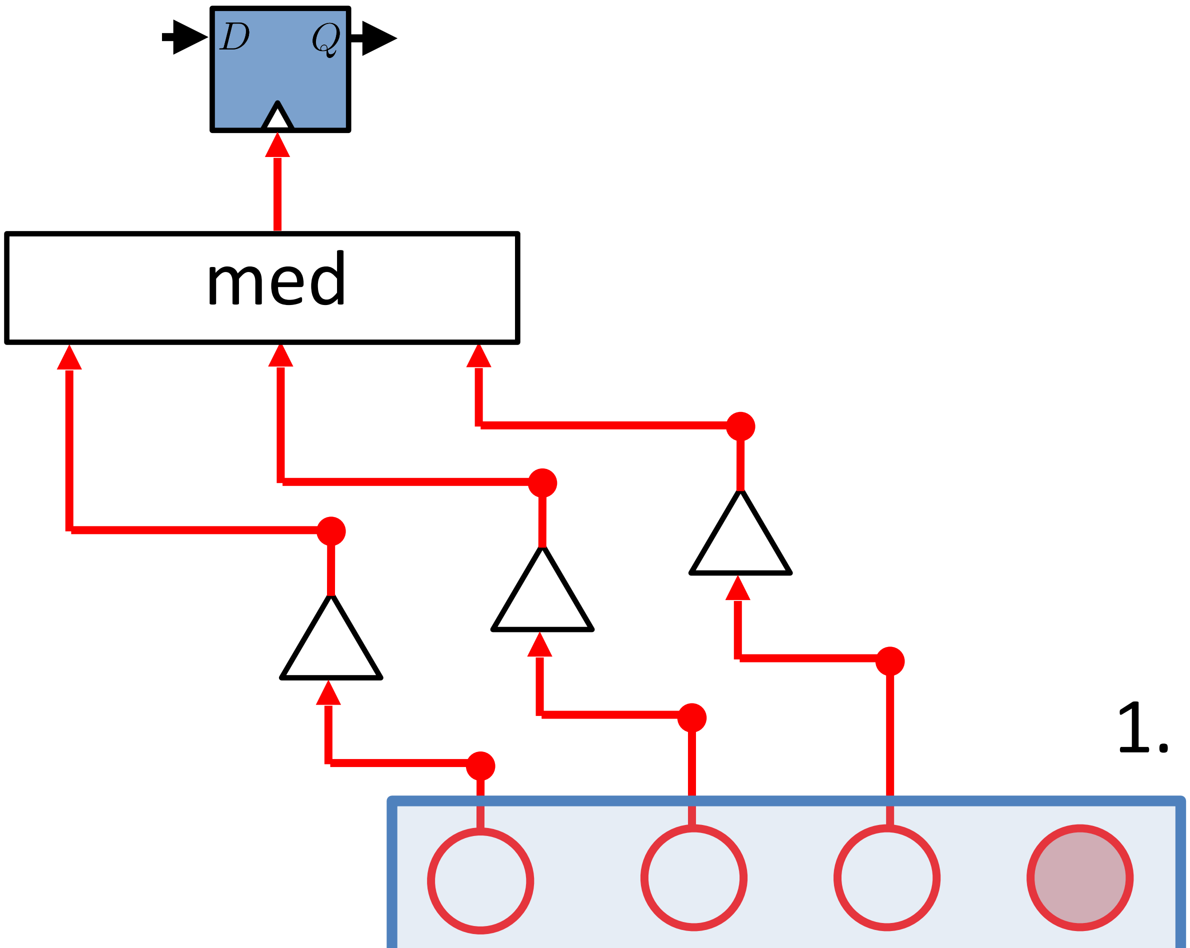
- fault tolerance?

- skew?

- cost?

Fault tolerance

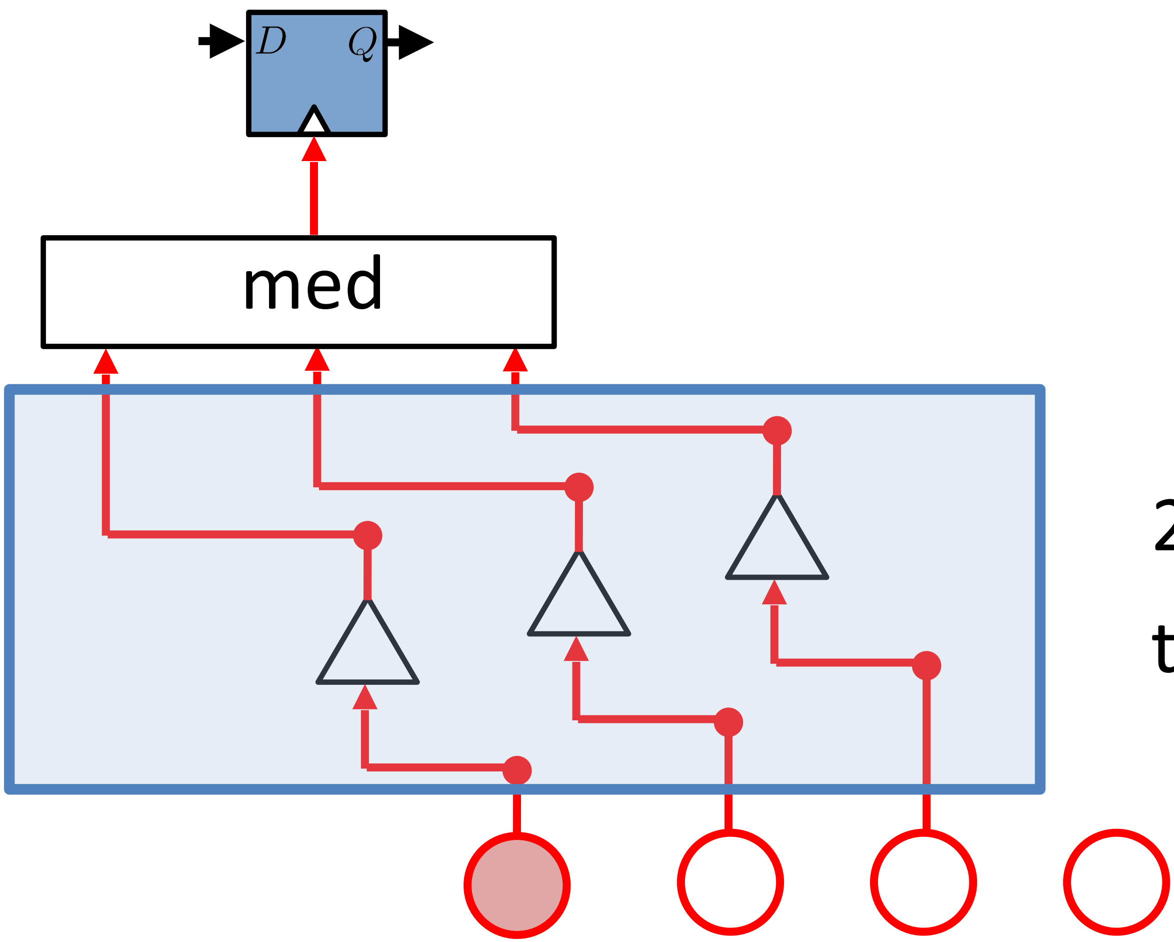
replicate the clock source, vote on output



1. at most f fail

Fault tolerance

replicate the clock source, vote on output

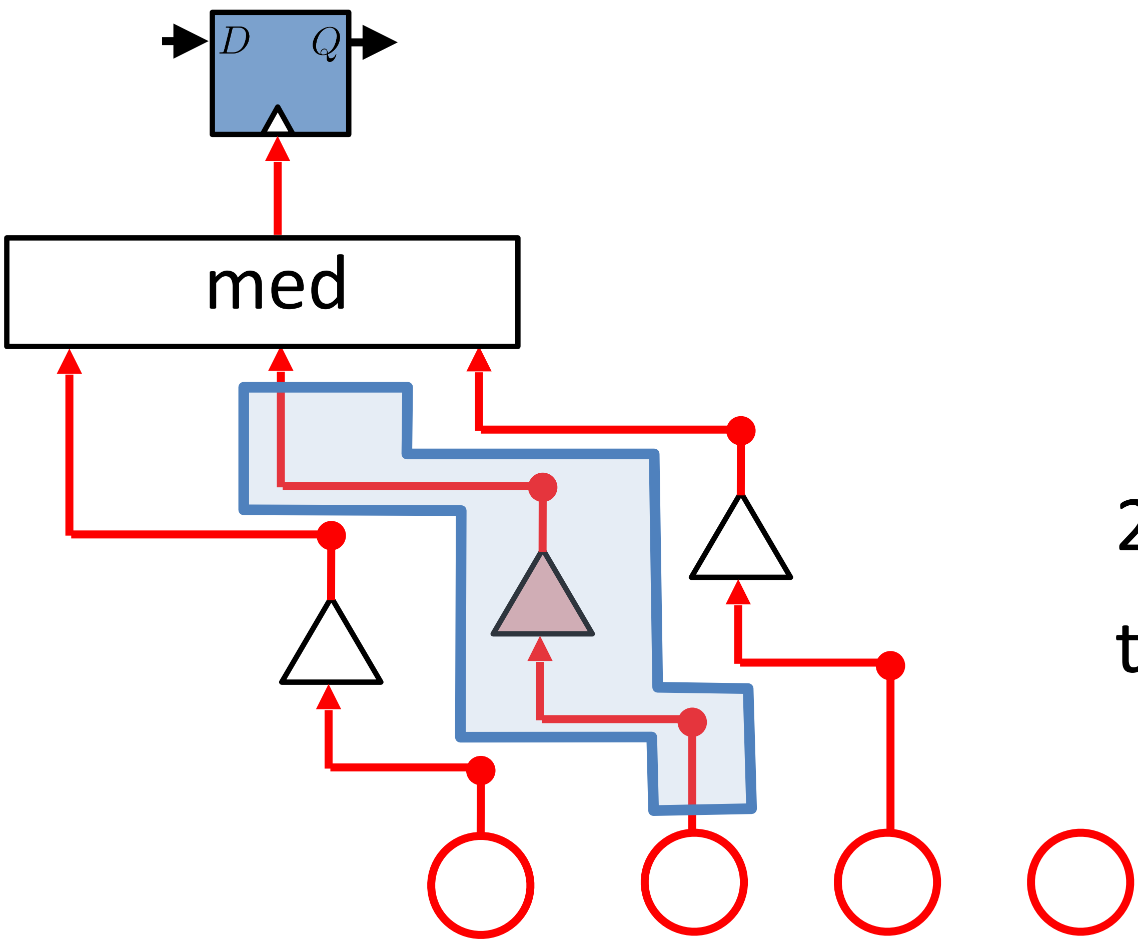


2. at most f

trees fail

Fault tolerance

replicate the clock source, vote on output



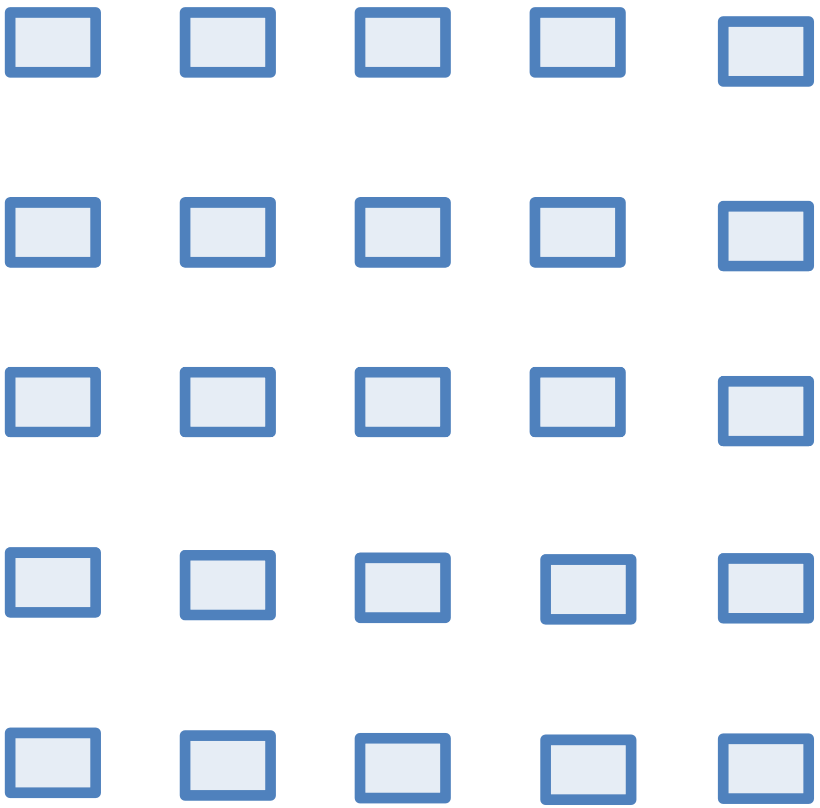
2. at most f trees fail

Idea 2

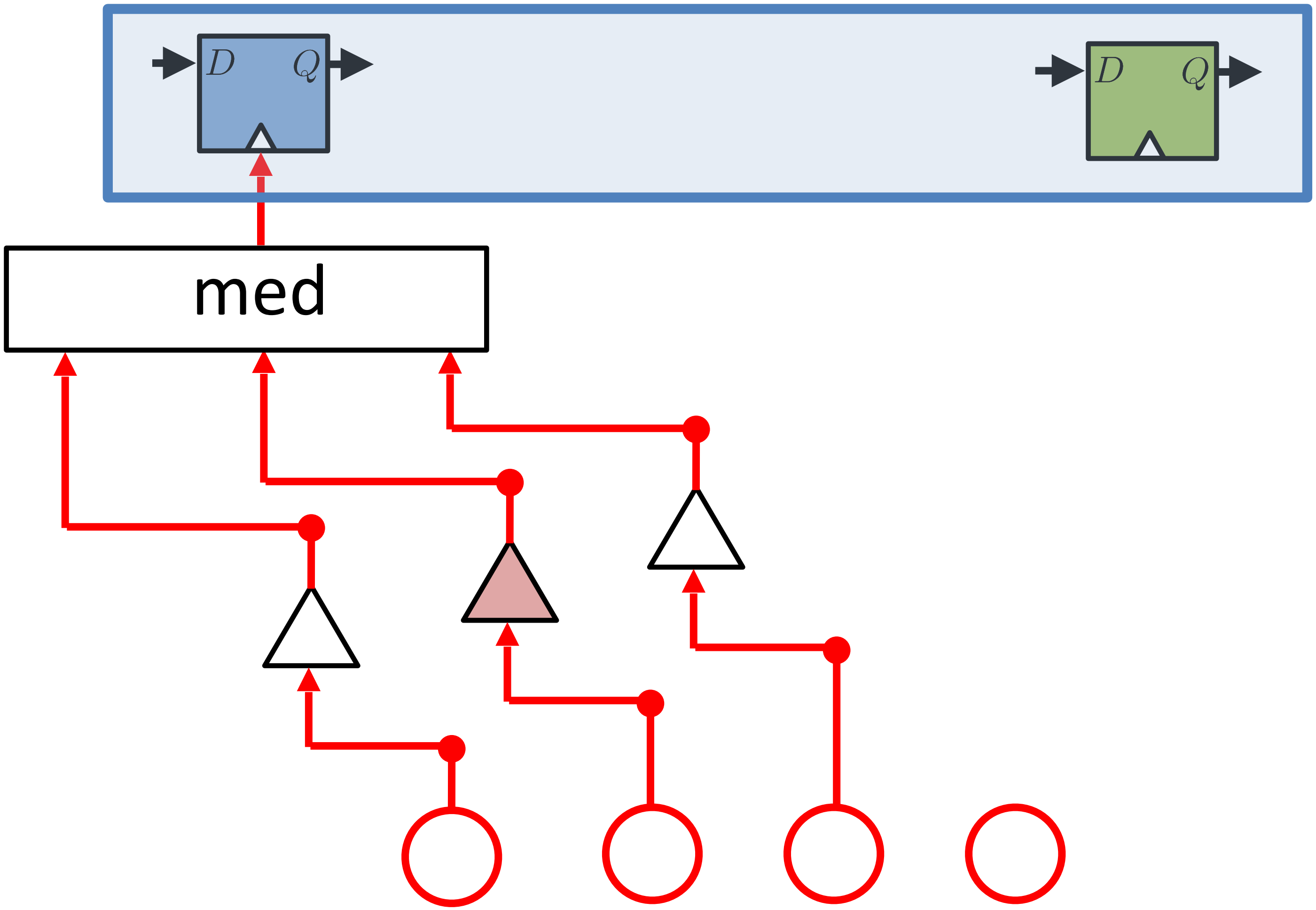
5x5 grid

properties:

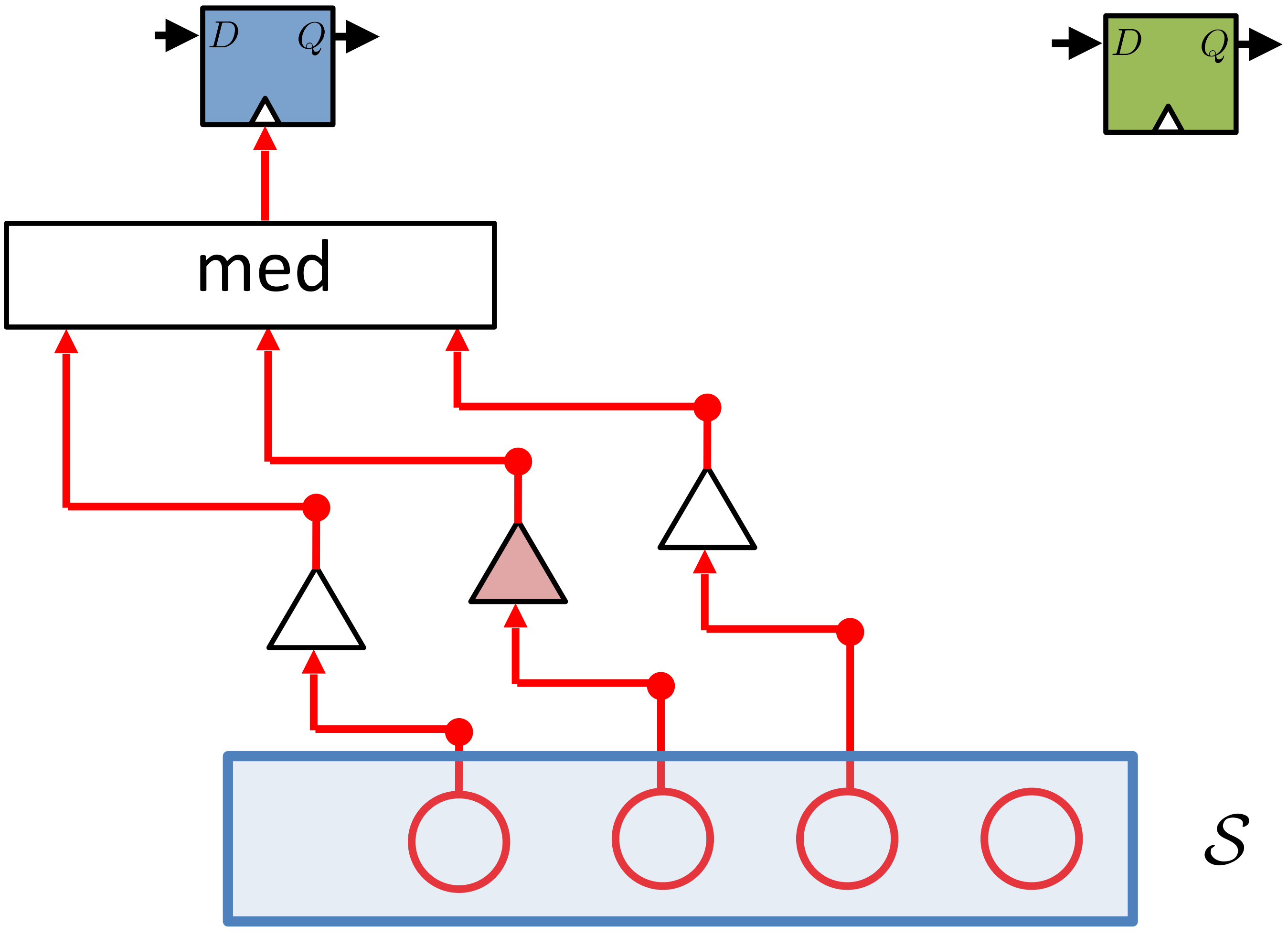
- fault tolerance?
- **skew?**
- cost?



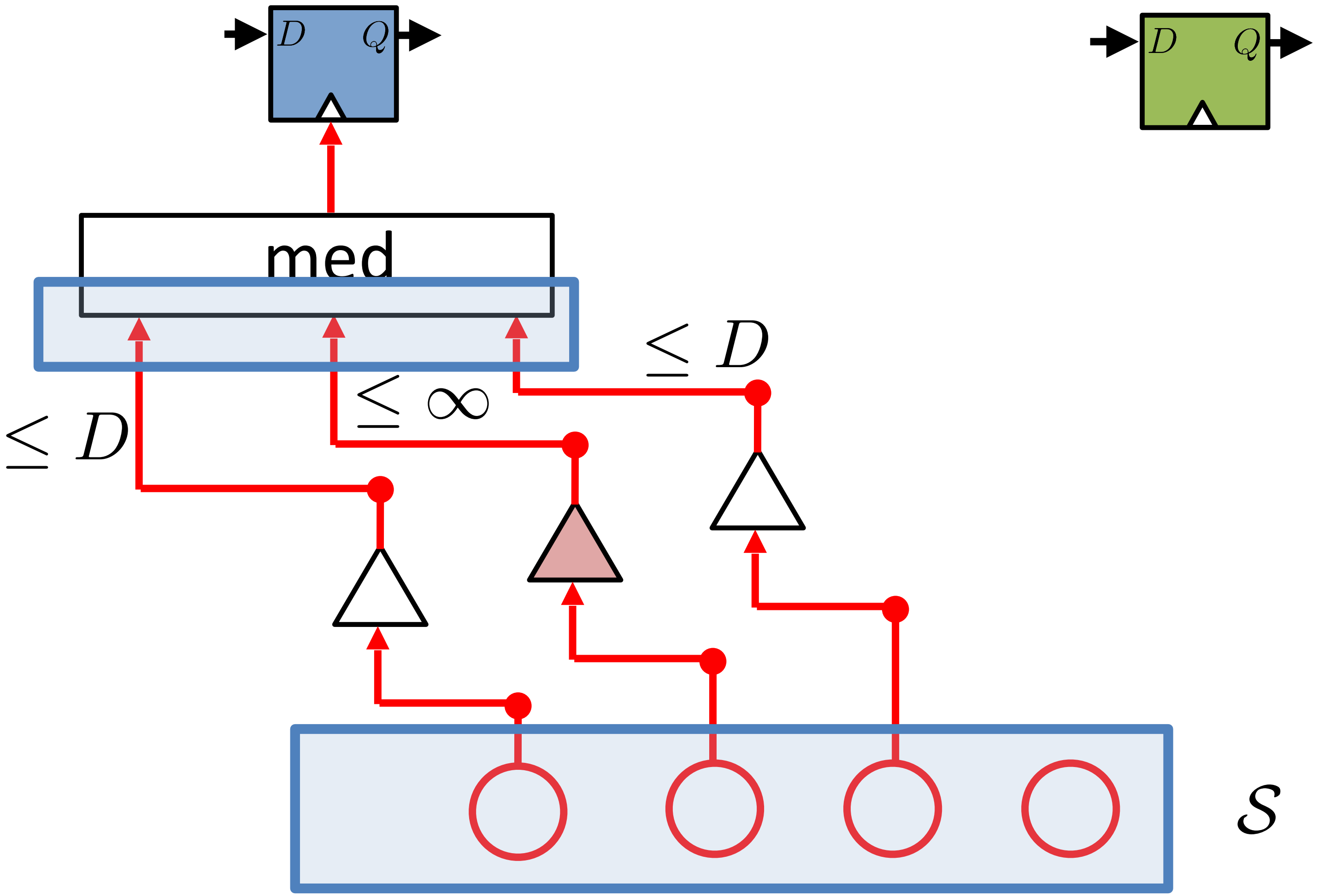
Skew



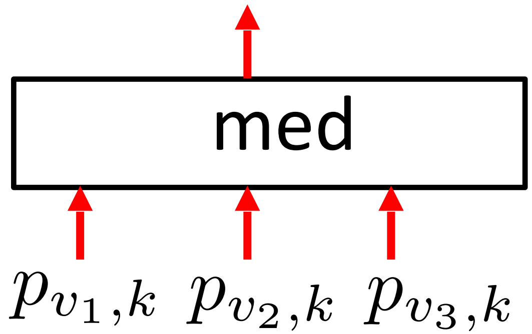
Skew



Skew

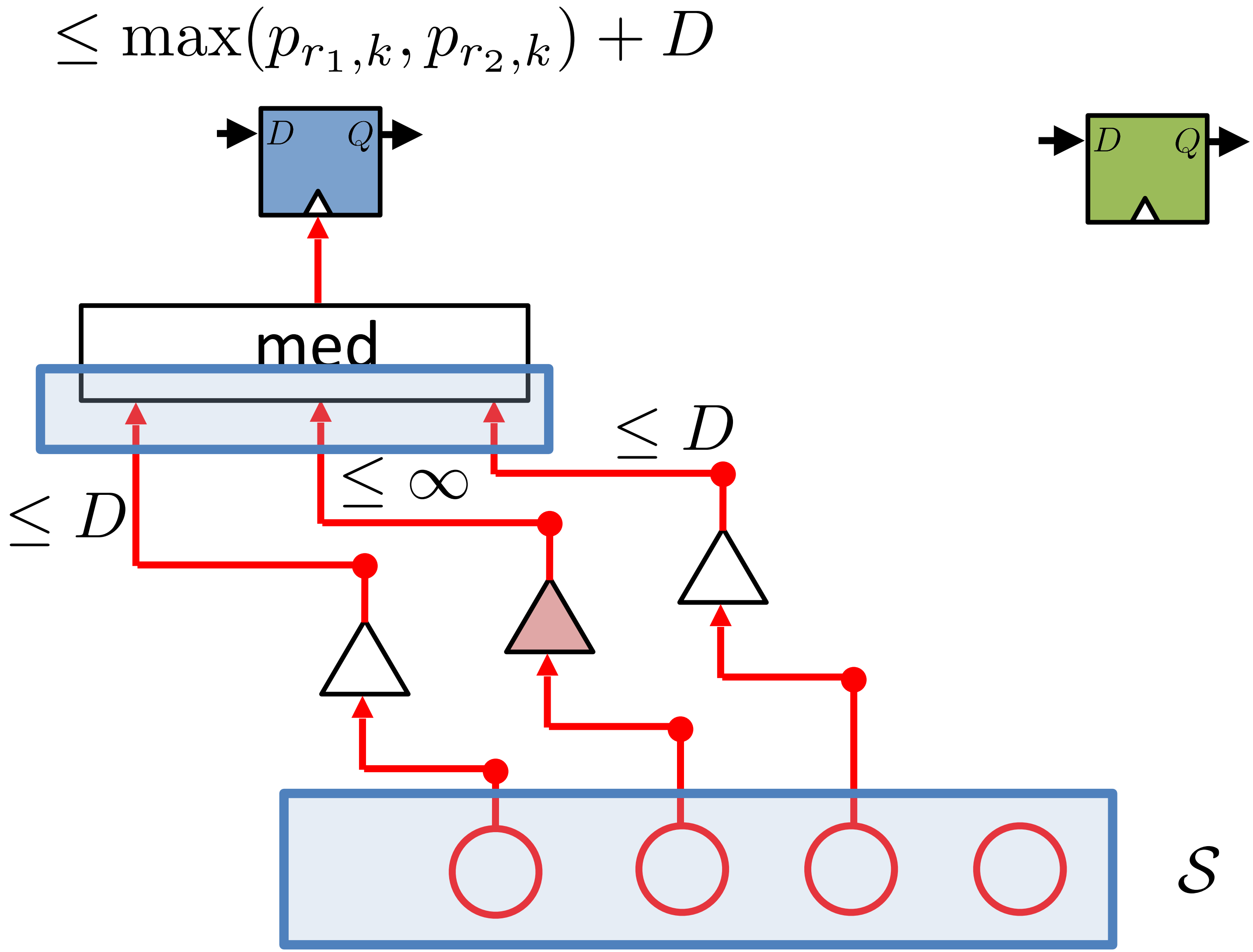


Skew

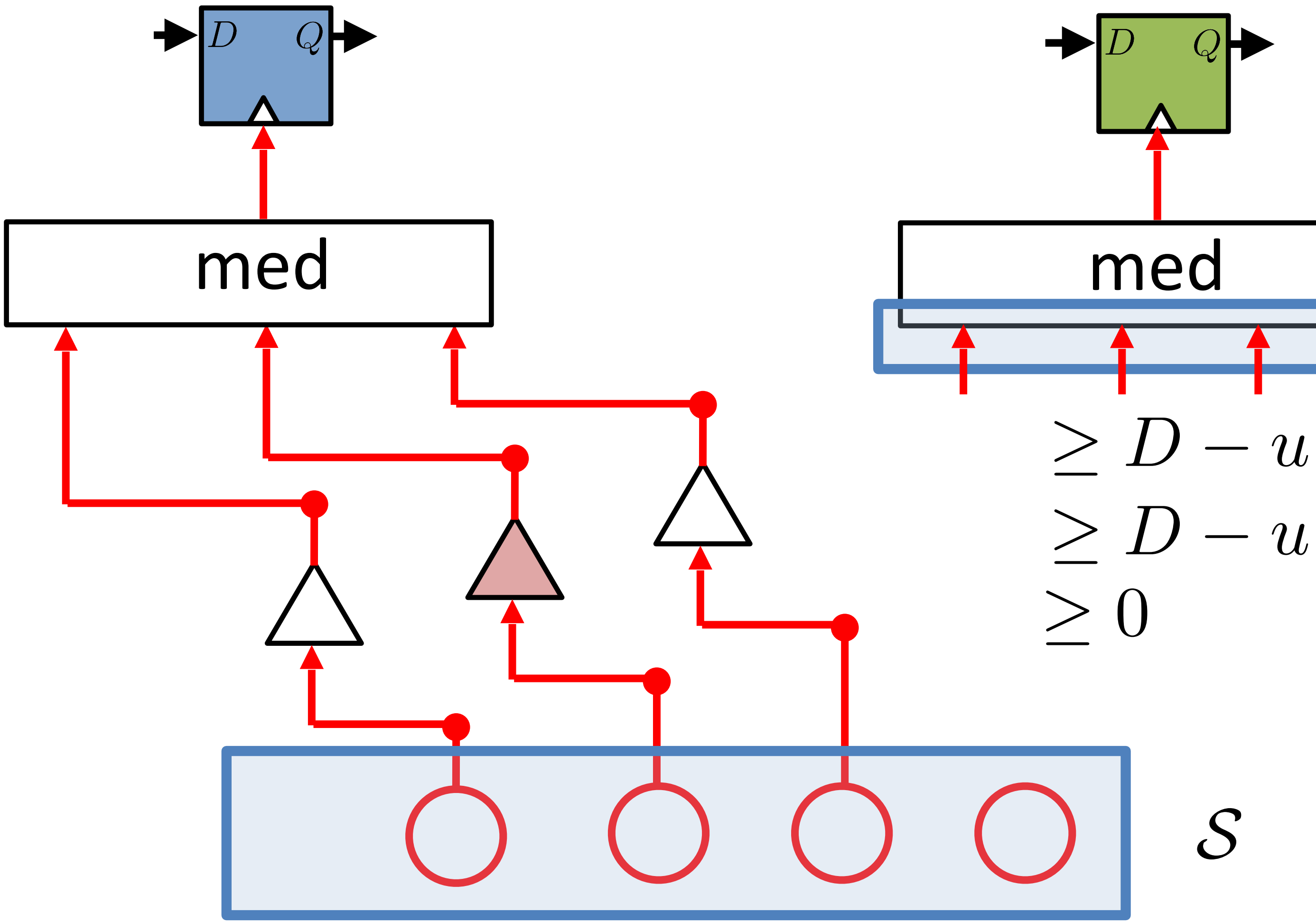


$$\begin{aligned}\mathrm{med}(p_{v_1,k}, p_{v_2,k}, p_{v_3,k}) &\le \mathrm{med}(p_{r_1,k} + D, p_{r_2,k} + D, \infty) \\ &\le \max(p_{r_1,k} + D, p_{r_2,k} + D) \\ &\le \max(p_{r_1,k}, p_{r_2,k}) + D\end{aligned}$$

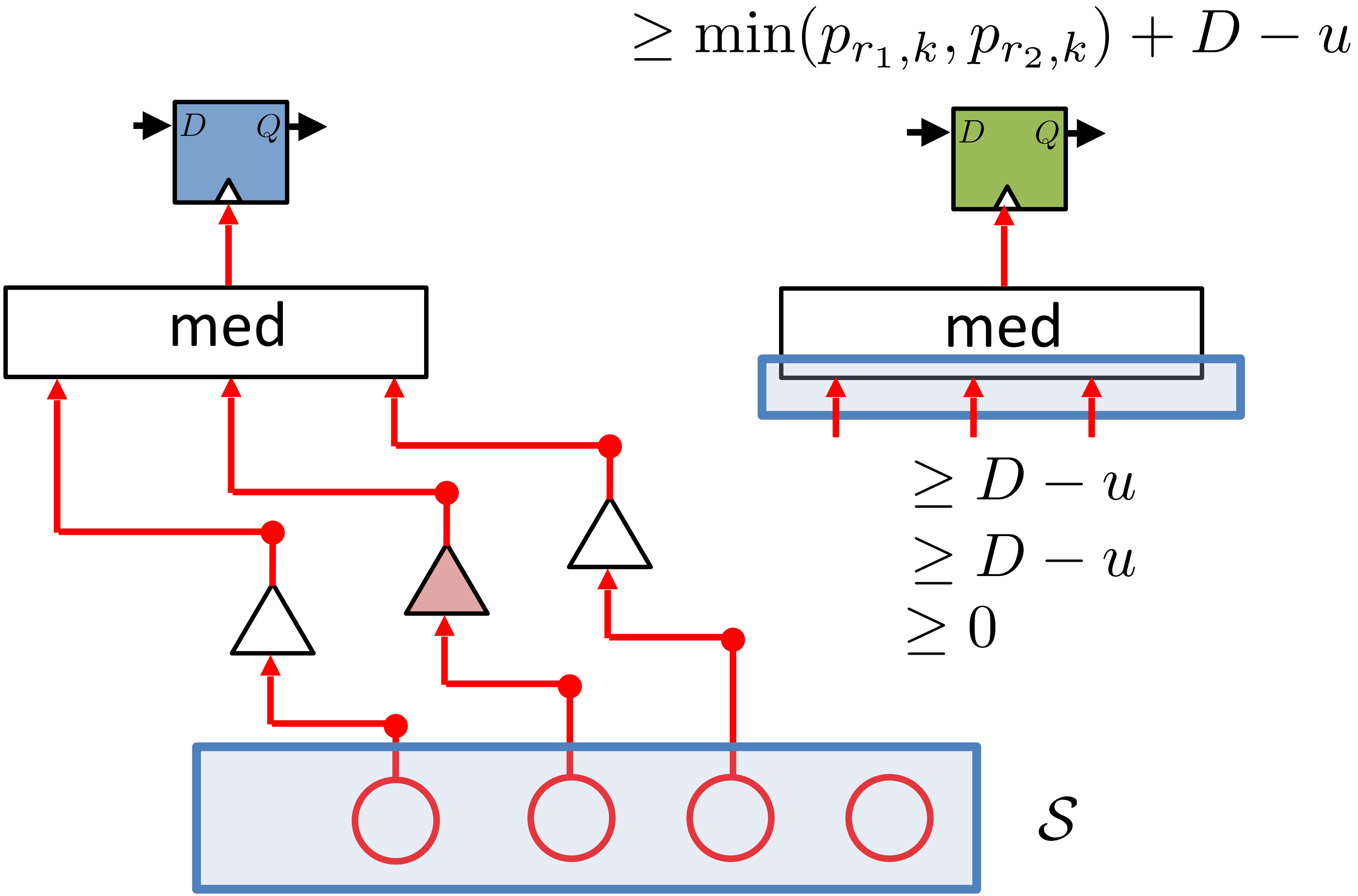
Skew



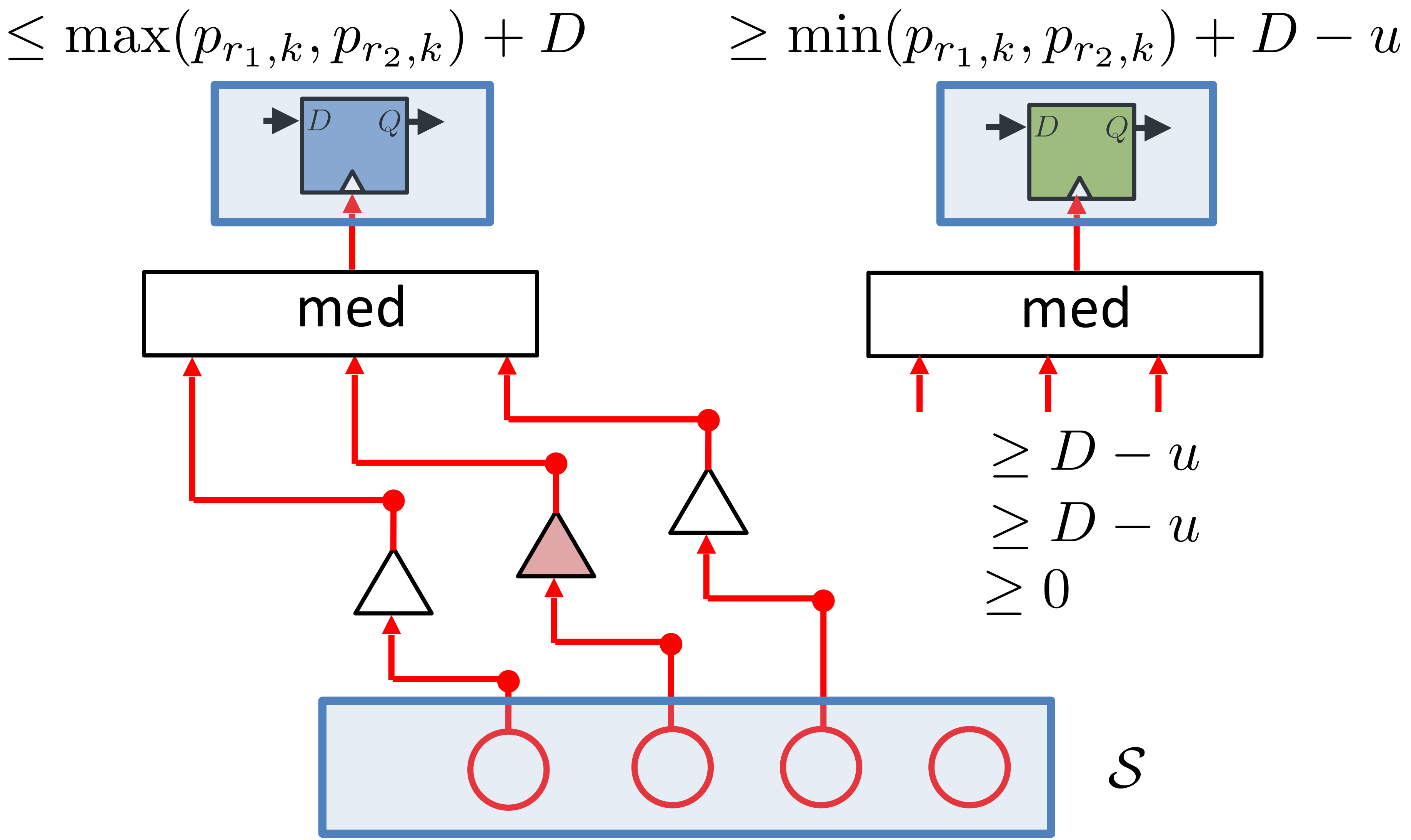
Skew



Skew

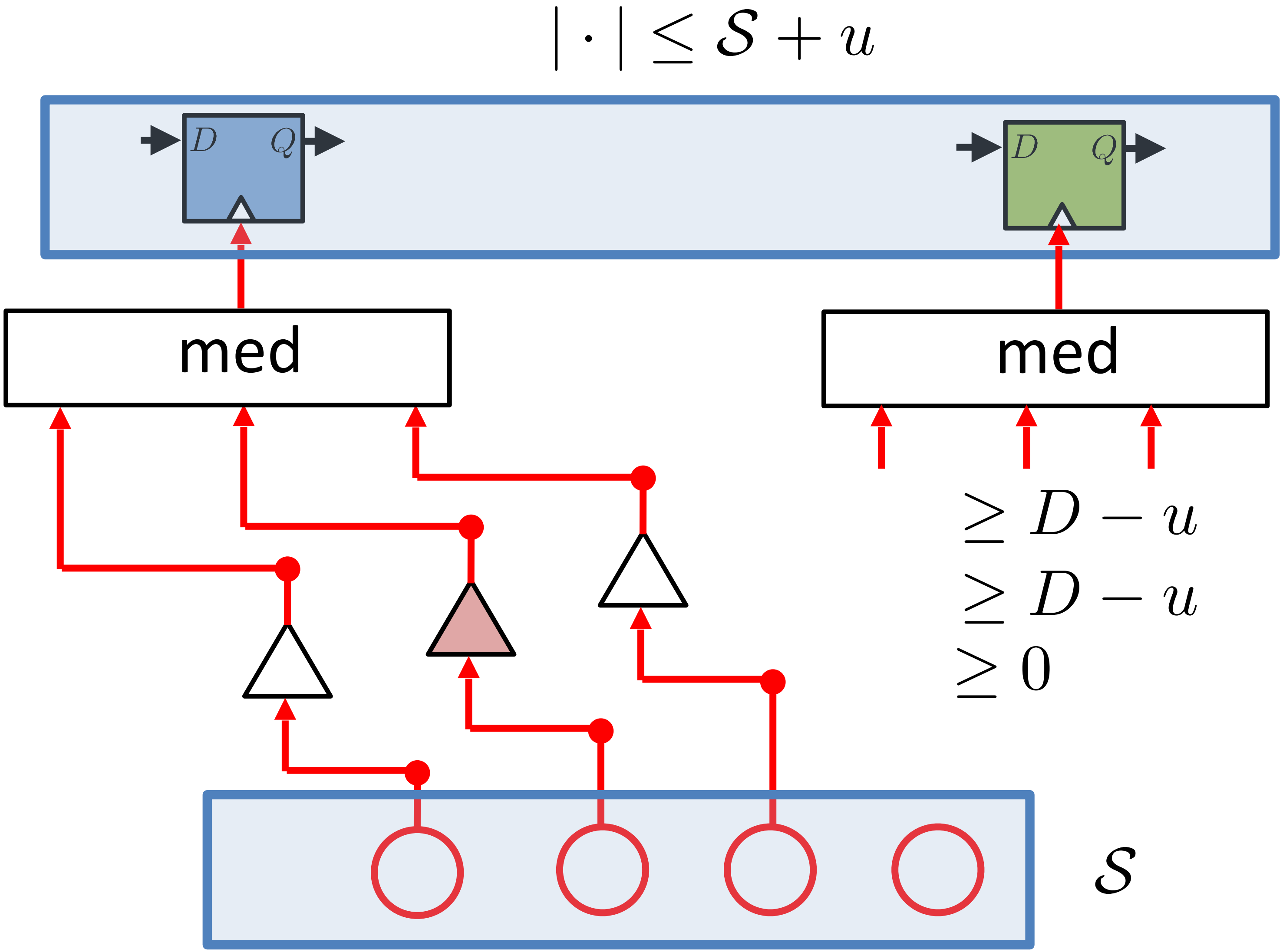


Skew



$\leq \max(p_{r_1,k}, p_{r_2,k}) + D$
$\geq \min(p_{r_1,k}, p_{r_2,k}) + D - u$
D
Q
D
Q
med
med
$\geq D - u$
$\geq D - u$
≥ 0
$\mathcal{S}$

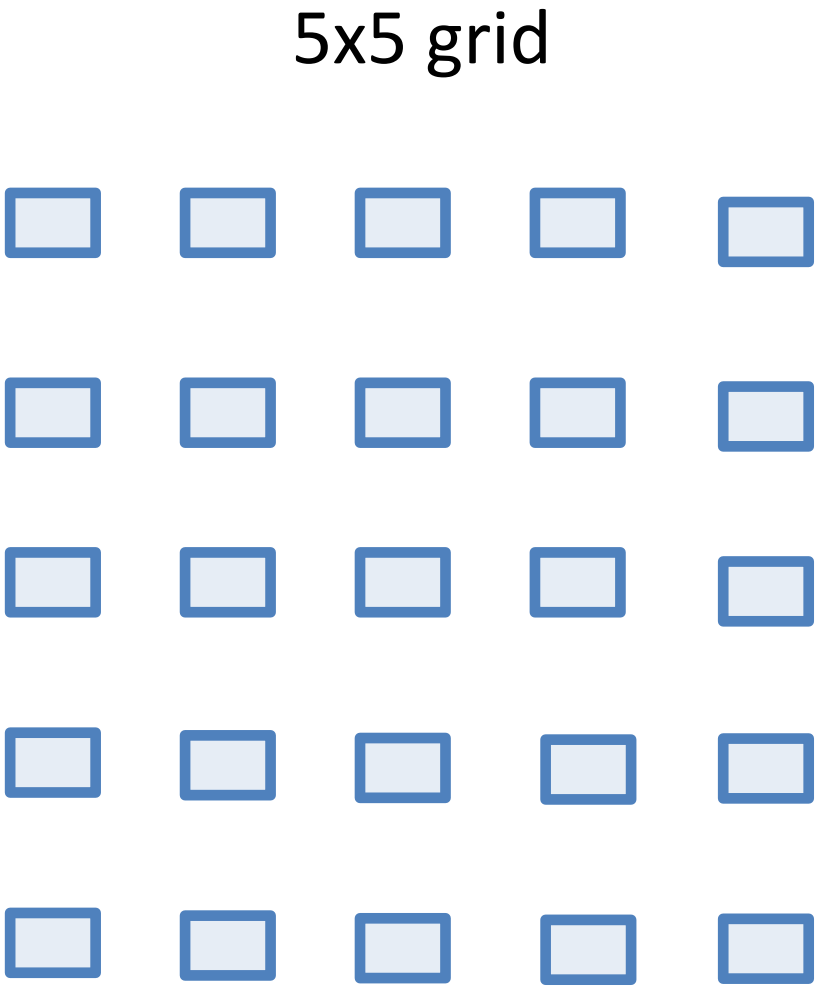
Skew



Idea 2

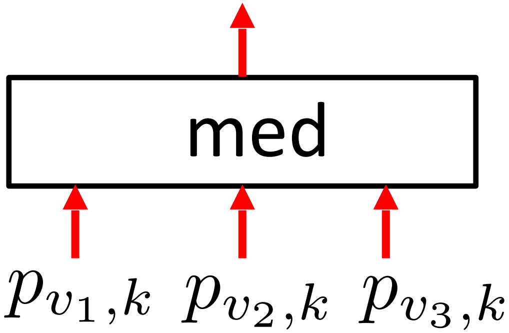
properties:

- fault tolerance?
- skew?
- **cost?**

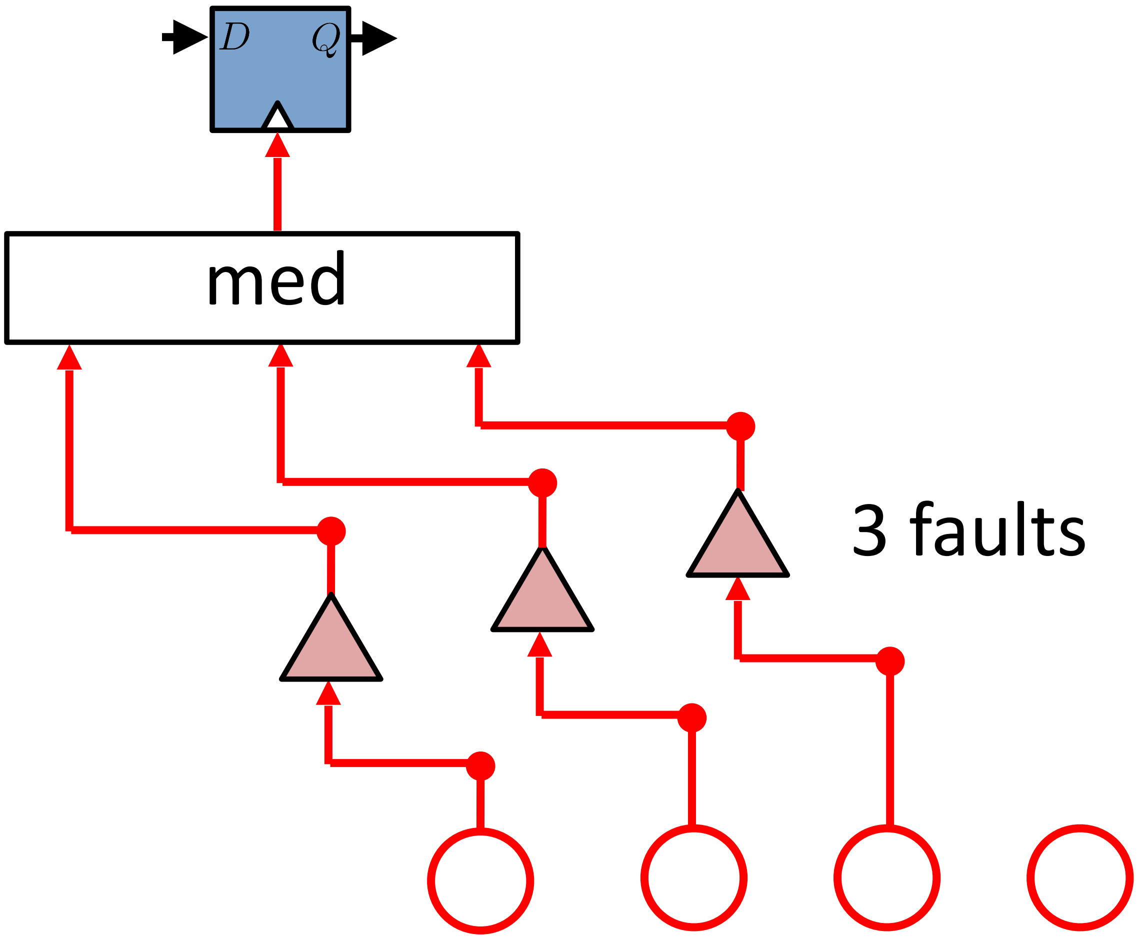


Cost

3f+1 roots & 2f+1 trees



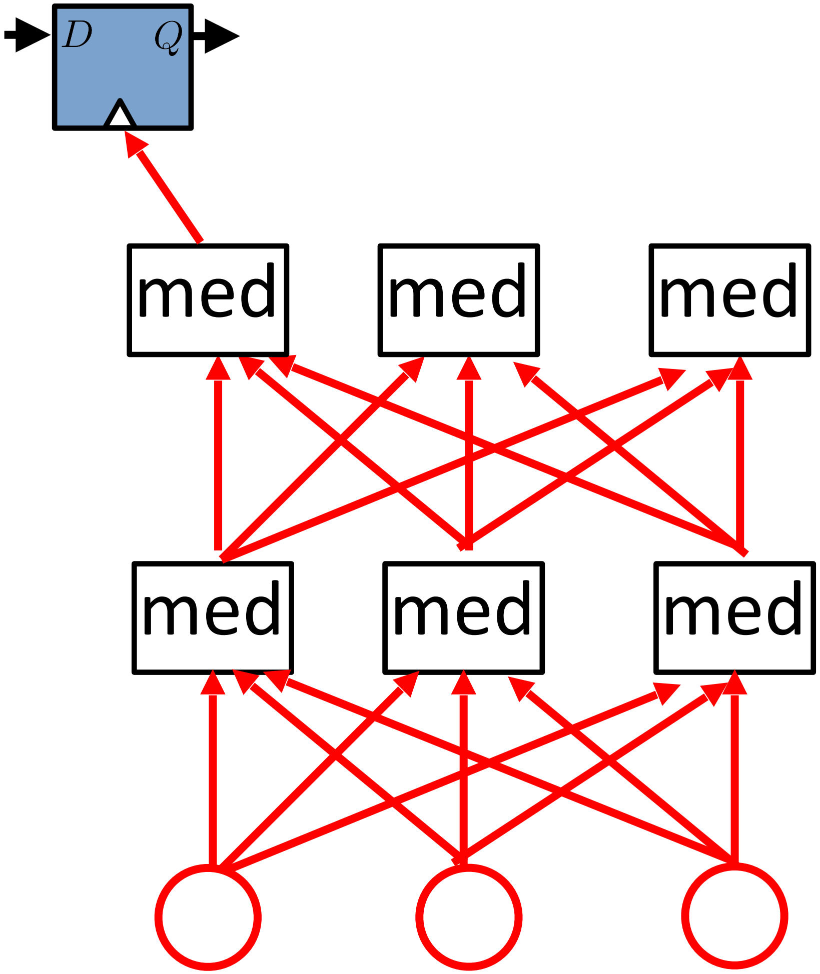
Depends on the environment



Q: other solutions?

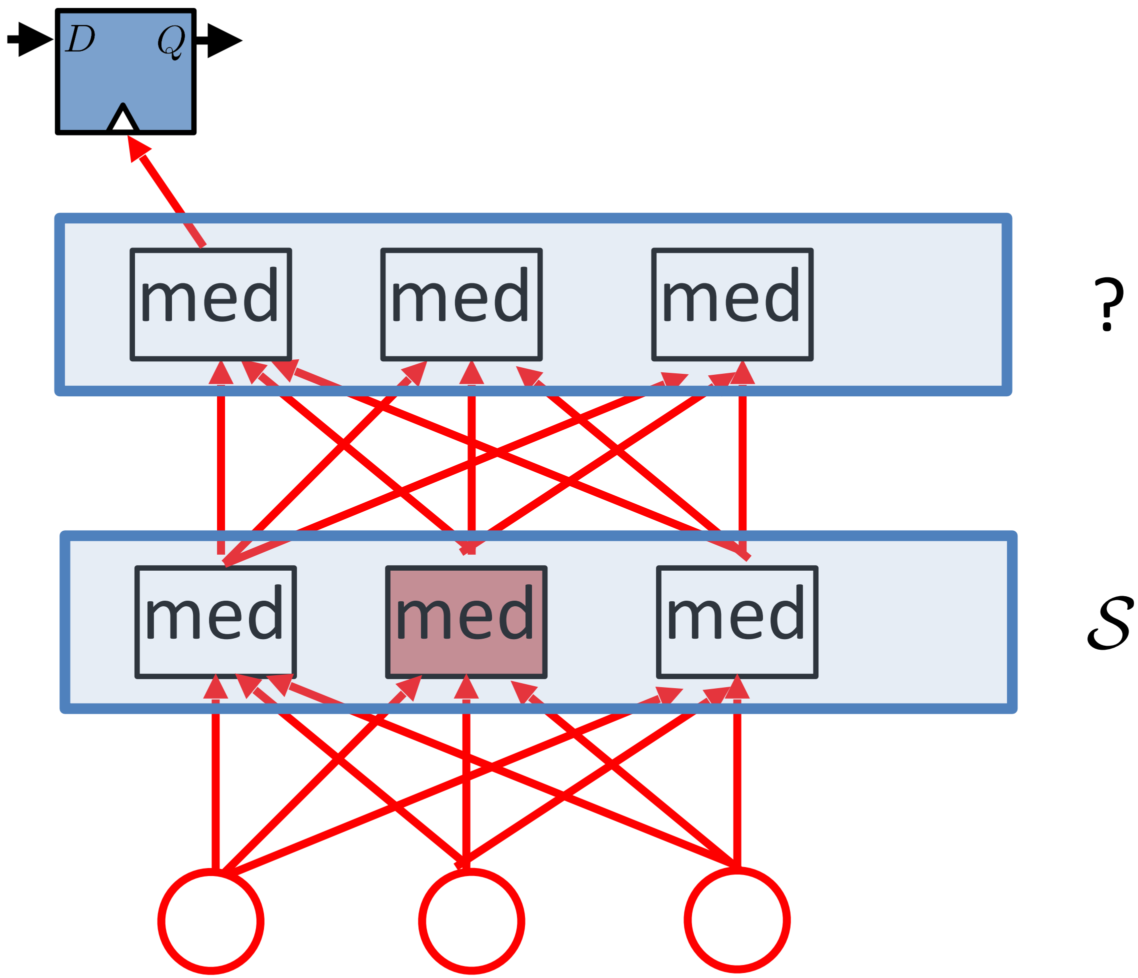
Idea 3: interlinked trees

median at each stage of the tree



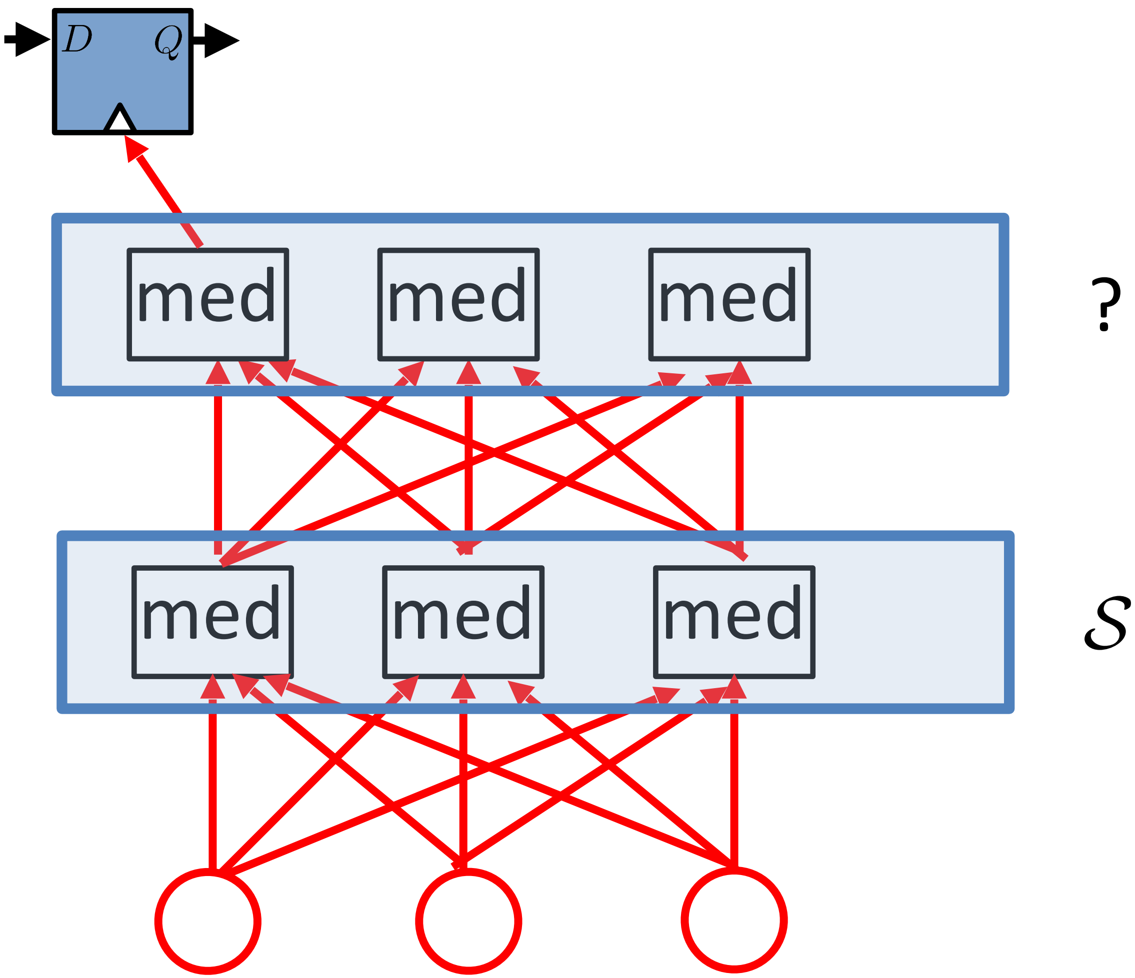
Skew

median at each stage of the tree



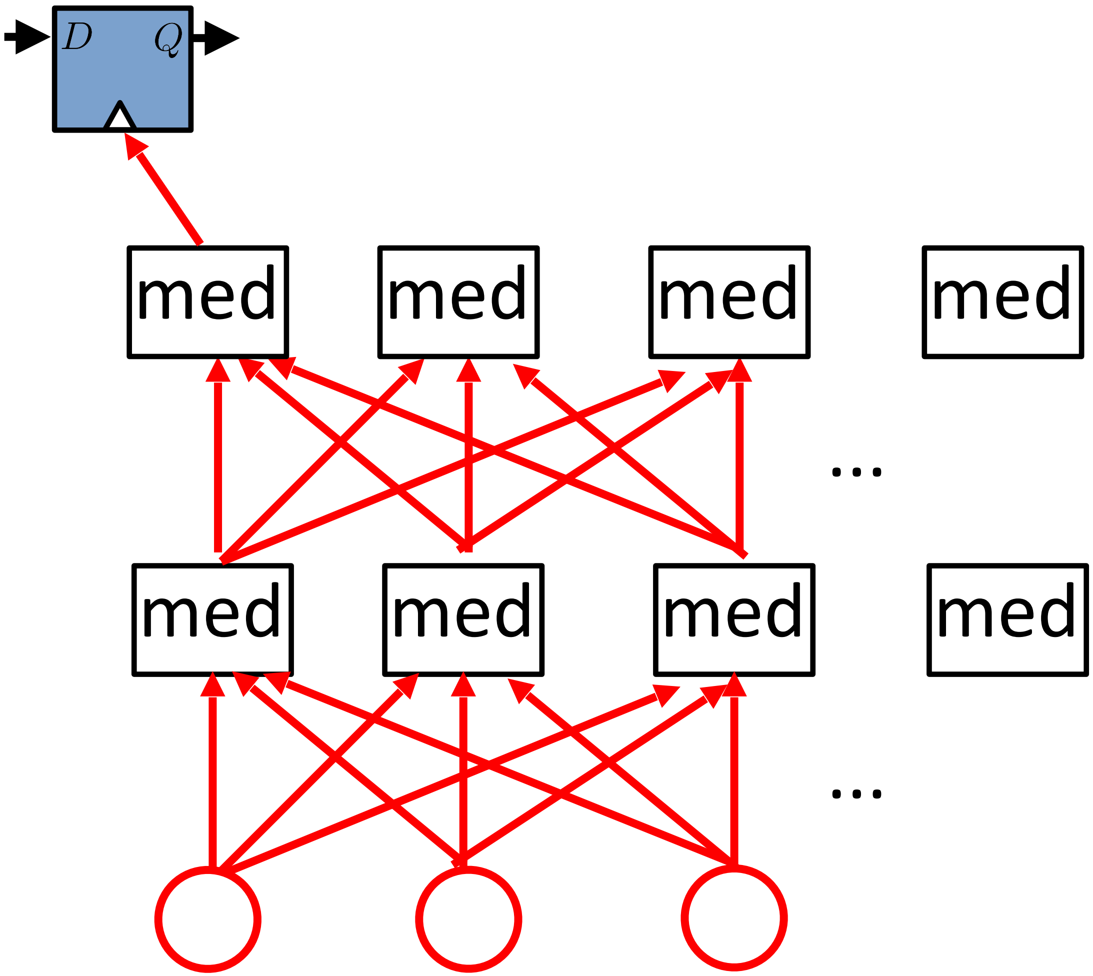
Skew

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):

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

local constraints (idea 2, redundant trees):

$$P(\text{correct}) = P(\leq 1 \text{ faulty tree})$$

$$\left\{p = 0.01,\, c = (1-p)^{100},\, c^3 + 3\,c\,(1-c)^2\right\}$$

$$c\left(4\,c^2 - 6\,c + 3\right) \approx 0.490383$$

local constraints (idea 3, interlinked trees):

$$P(\text{correct}) = P(\leq 1 \text{ fault per triple})$$

$$\left\{p = 0.01,\, c = (1-p)^3 + 3\,(1-p)^2\,p,\, c^{100}\right\} \qquad c^{100} \approx 0.970635$$

Chapter 11 (3)

Low-degree clock distribution networks

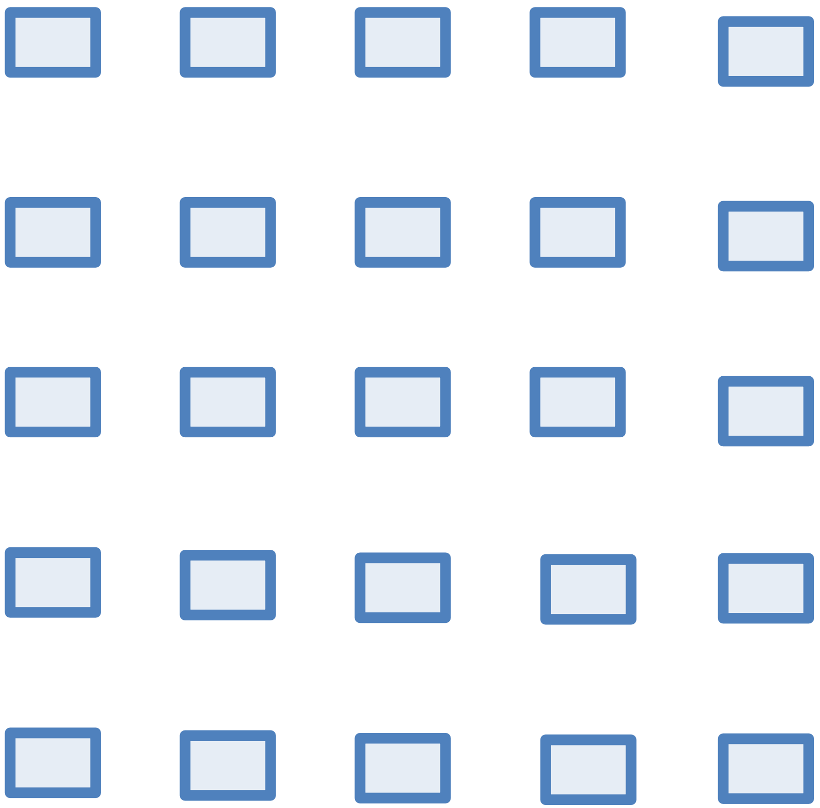
Matthias Fuegger and Christoph Lenzen

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

properties:

- fault tolerance
- skew
- cost

5x5 grid



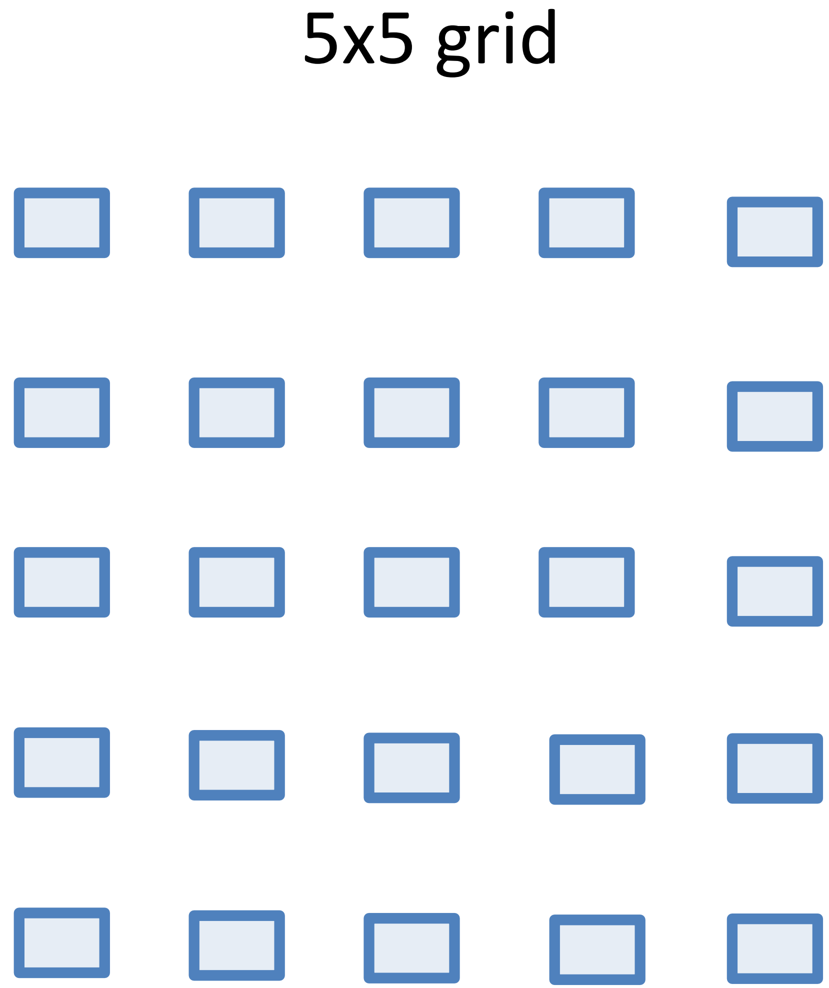
Clocking a grid with root(s) & tree(s)

properties:

- fault tolerance

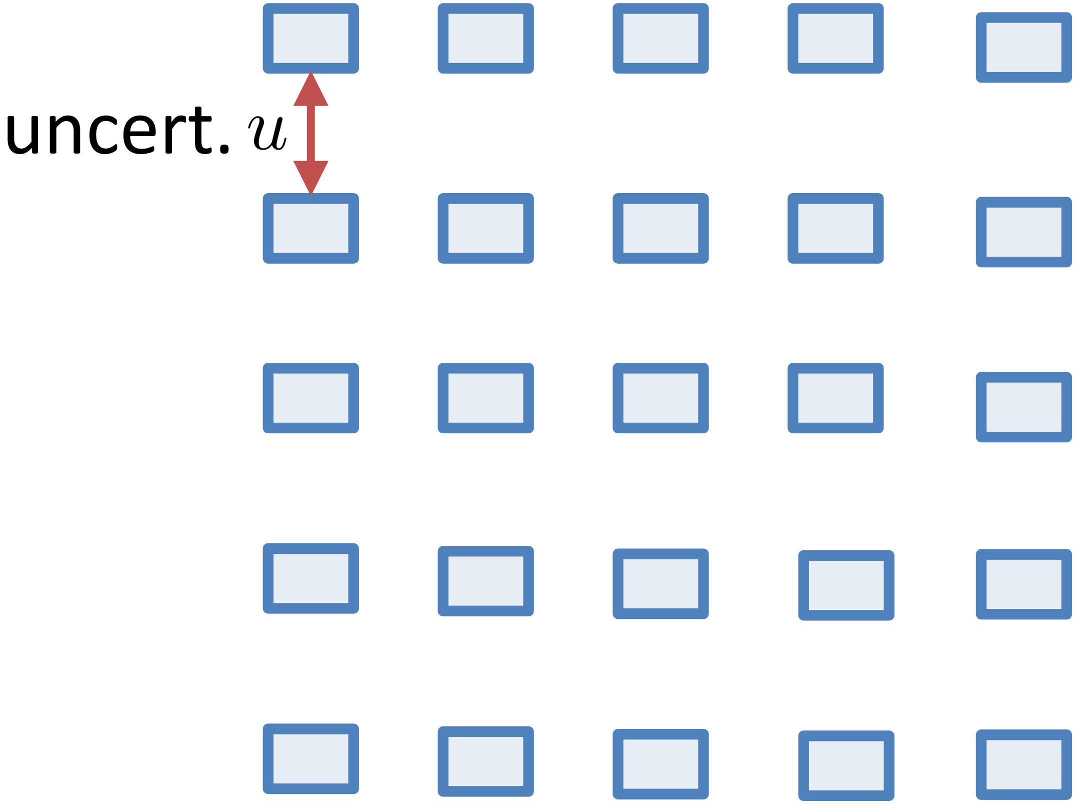
- **skew (g & l):**
  $|\cdot| \leq \mathcal{S} + U$

- cost



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

5x5 grid



- **skew (local):**

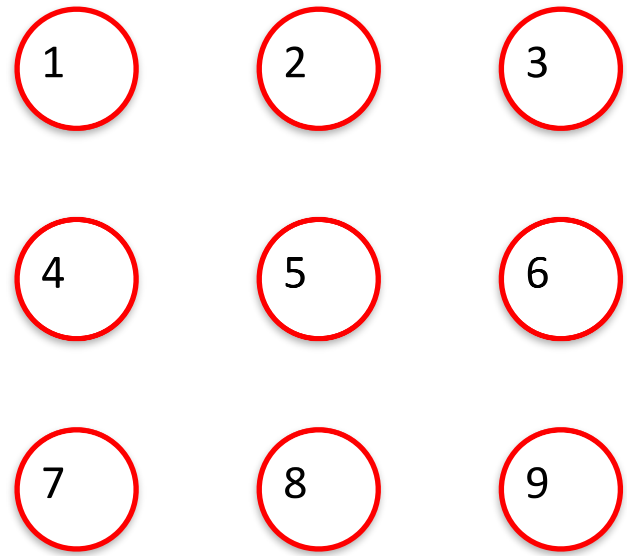
$$|\cdot| \leq \mathcal{S} + U$$

$$U = f(u)$$

Limits

k x k grid, n = k^2

with 1 x 1 cells



Show:

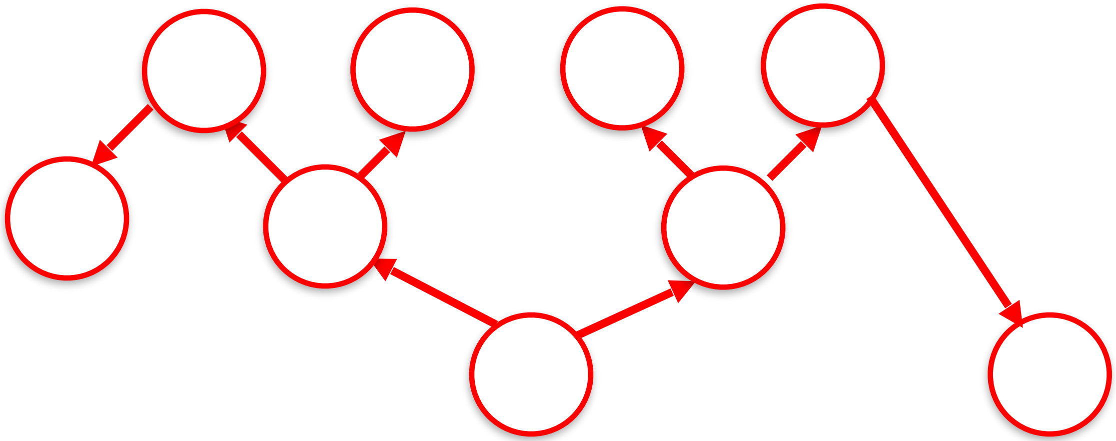
In any tree spanning the points,

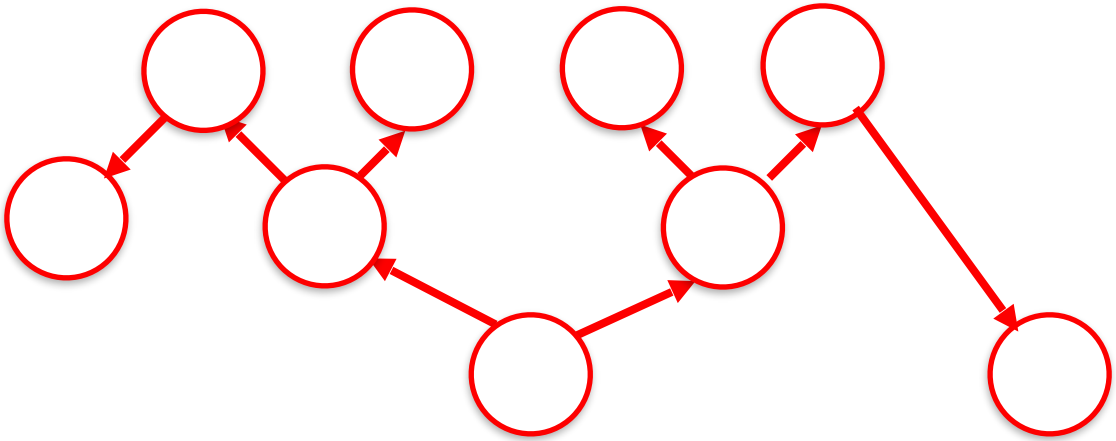
exist two grid-neighboring nodes with

tree-distance d = Omega(k).

Tree:

degree <= 3 (in+out degree)

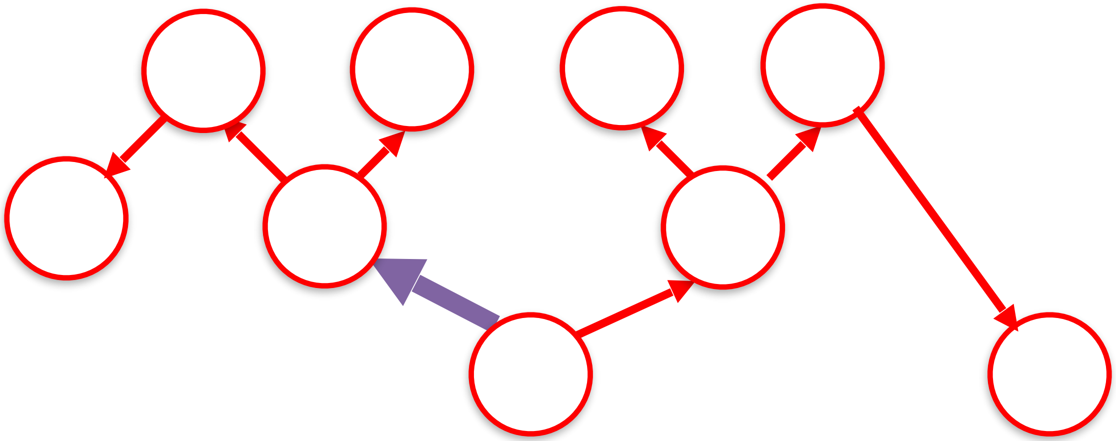




Exists edge e st.

after removing e: both components size >=

(n-1)/3 = Omega(n)



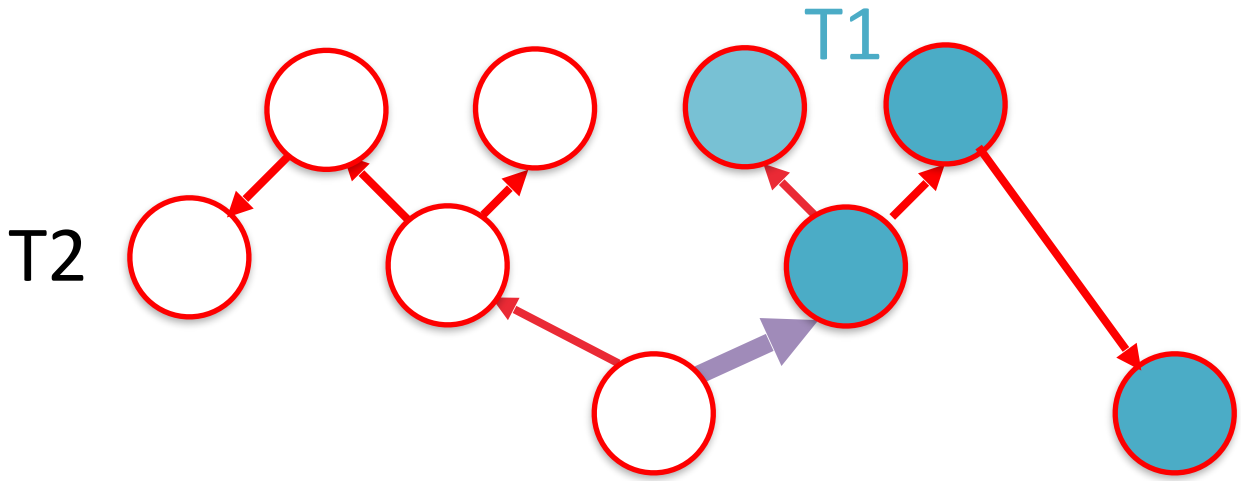
Exists edge e st.

after removing e: both components size >=

(n-1)/3 = Omega(n)

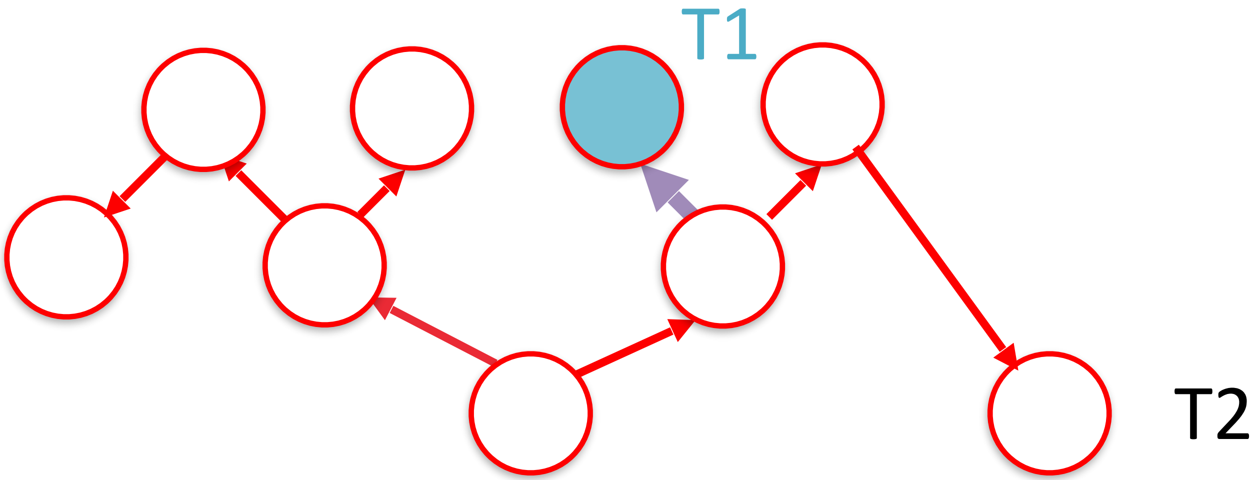
n = 9 ->

subtree size >= 8/3 = 2.6..



Pick e.

Case 1: |T1|, |T2| >= (n-1)/3. Done.

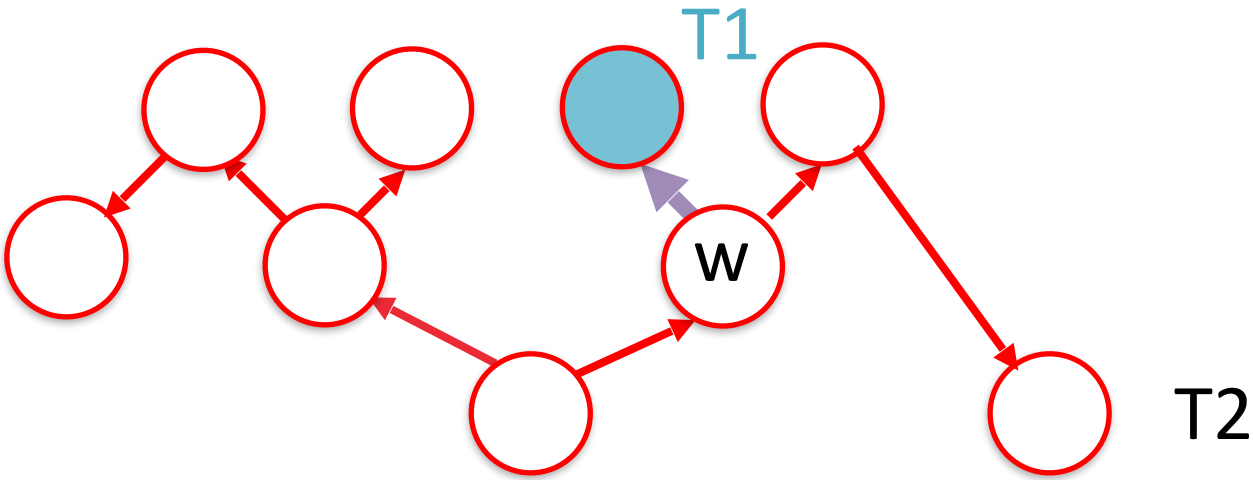


Pick e.

Case 1: |T1|, |T2| >= (n-1)/3. Done.

Case 2: |T1| < (n-1)/3.

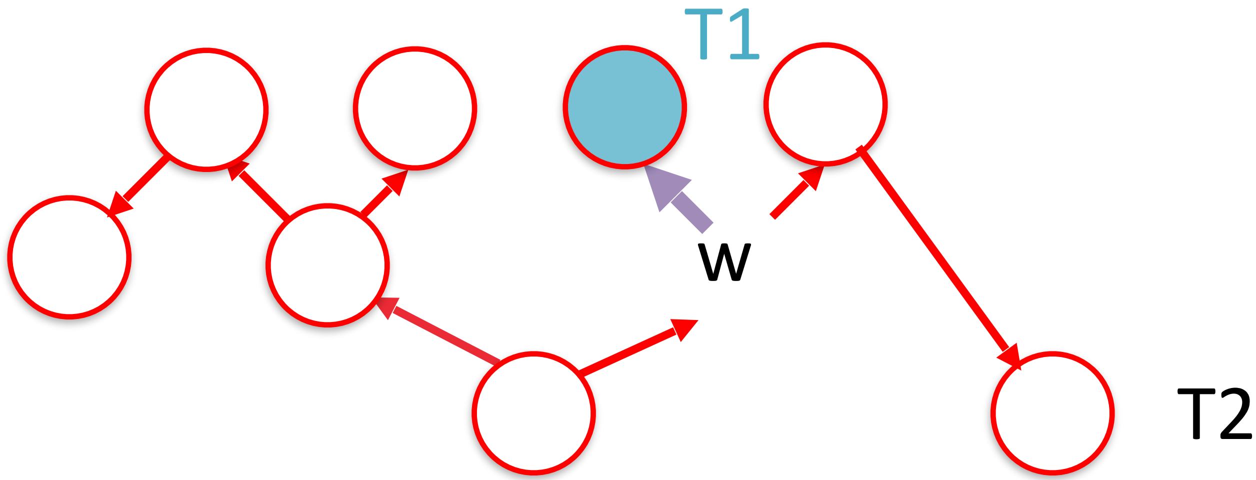
-> |T2| = n - |T1| >= n - (n-1)/3 = 2(n-1)/3 + 1



Case 2: |T1| < (n-1)/3.

-> |T2| = n - |T1| >= n - (n-1)/3 = 2(n-1)/3 + 1

w := endpoint of e in T2



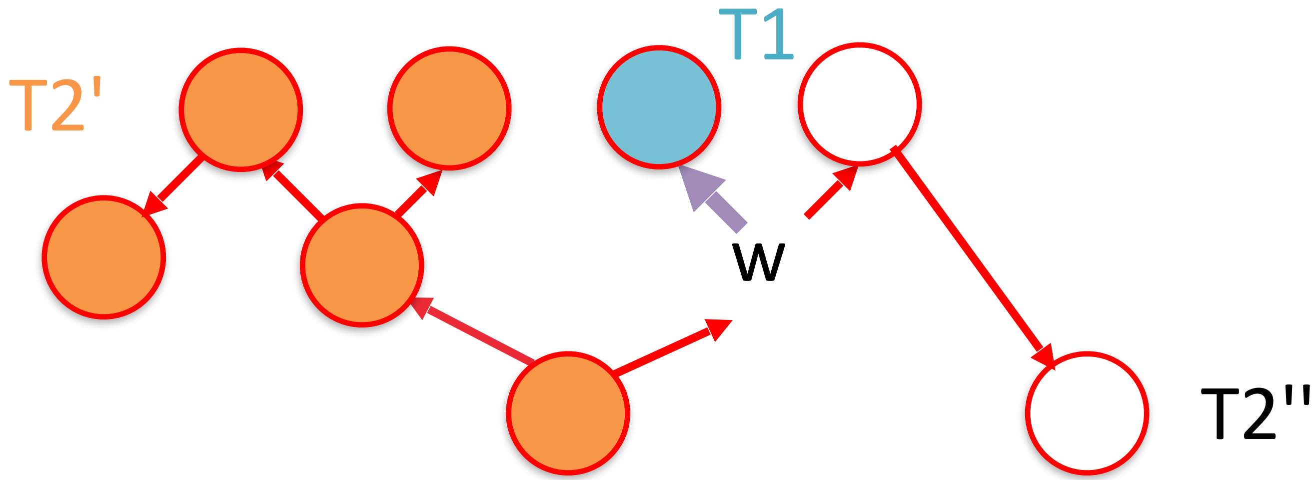
Case 2: |T1| < (n-1)/3.

-> |T2| = n - |T1| >= n - (n-1)/3 = 2(n-1)/3 + 1

w := endpoint of e in T2

delete w

-> at most 2 components of T2 (degree <= 3)



Case 2: |T1| < (n-1)/3.

-> |T2| = n - |T1| >= n - (n-1)/3 = 2(n-1)/3 + 1

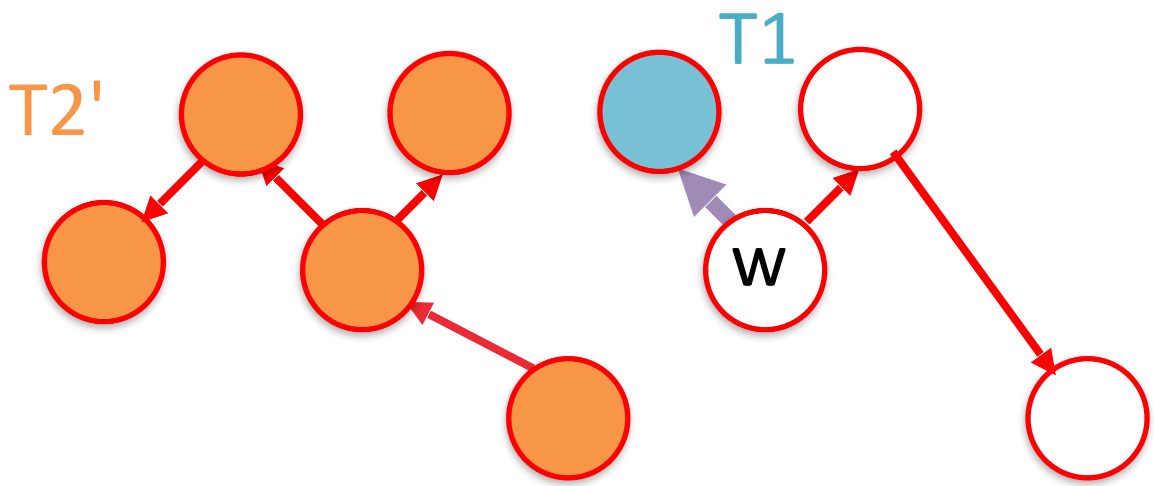
w := endpoint of e in T2

delete w

-> at most 2 components of T2 (degree <= 3): T2' & T2''

-> exists component T2' with size >= (|T2|-1)/2

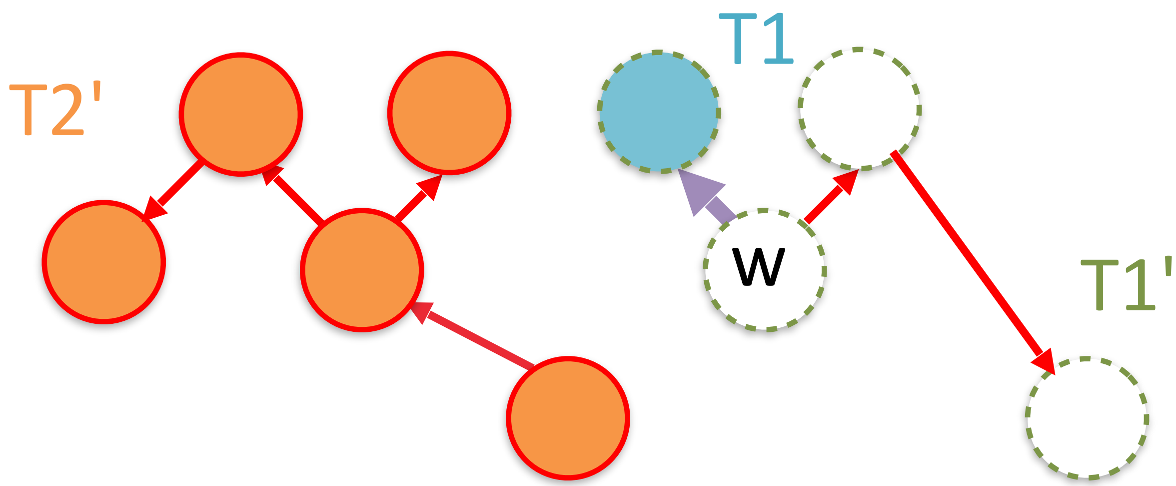
>= (n-1)/3



-> exists component T2' with size >= (|T2|-1)/2

>= (n-1)/3

delete edge from component T2' to w in T



-> exists component T2' with size >= (|T2|-1)/2

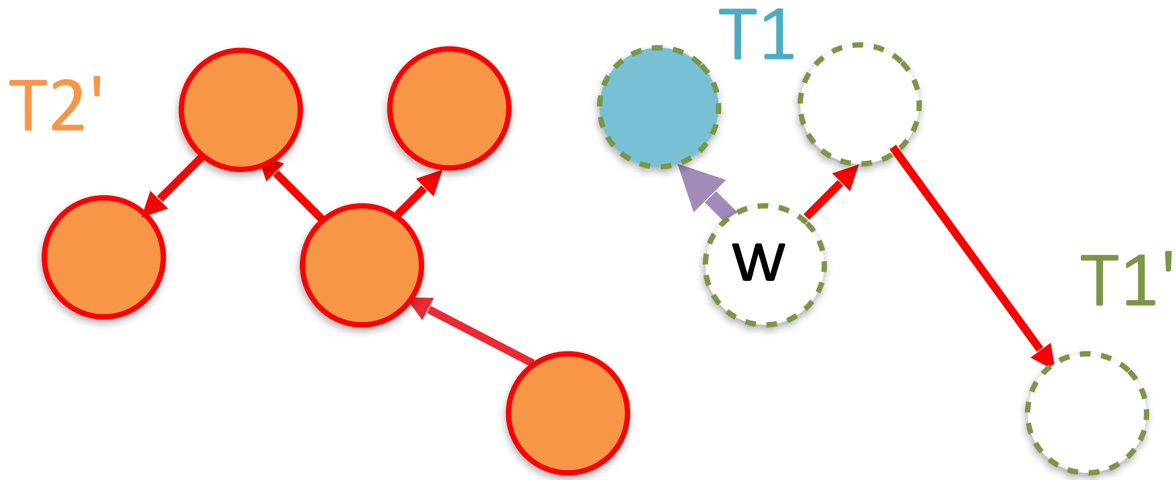
>= (n-1)/3

delete edge from component T2' to w in T

-> T1' and T2' with

|T1'| > |T1|

|T2'| >= (n-1)/3



|T1'| > |T1|

|T2'| >= (n-1)/3

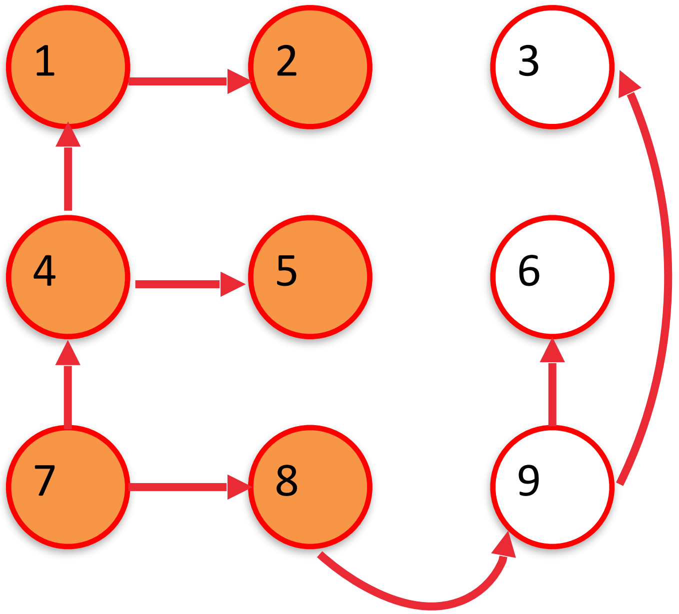
if both >= (n-1)/3:

  Done.

else:

  Repeat with T1' and T2' instead of T1 and T2.

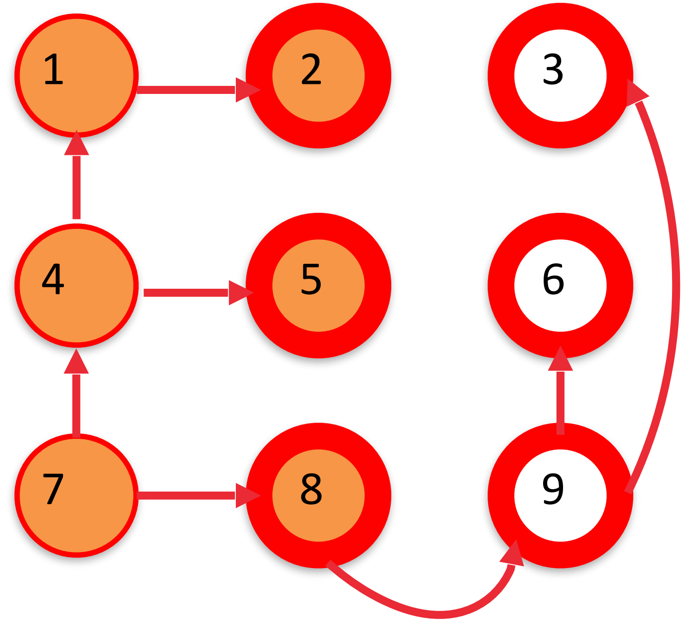
-> partition tree into 2 trees with Omega(k^2) nodes



-> partition tree into 2 trees with
Omega(k^2) nodes

boundary node :=
grid-neighbors in both sets

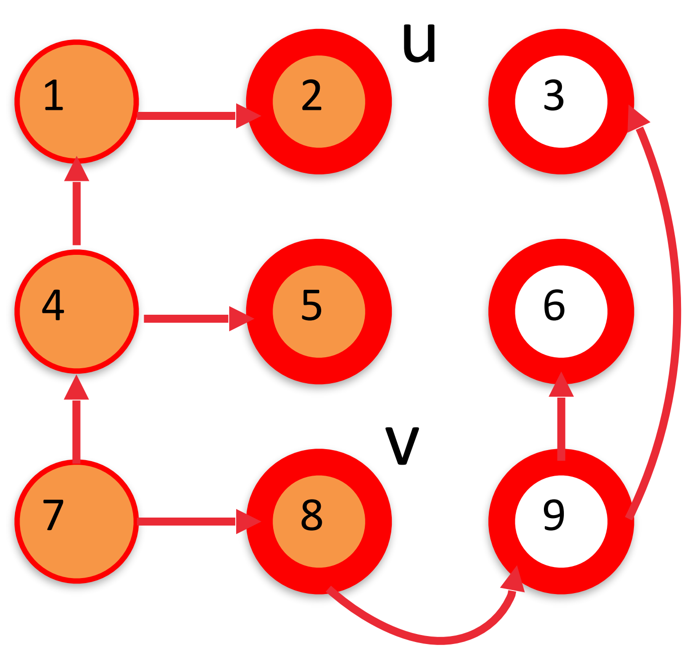
dist(u,v) := dist in grid
d(u,v) := dist in tree



-> 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

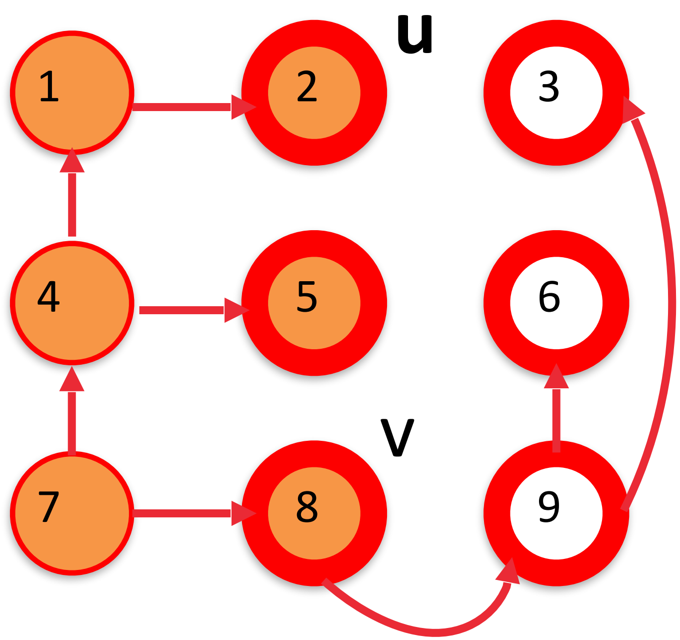
T1 (and of T2) with dist = Omega(k)

u, v two such nodes in T1

-> exist boundary nodes of

T1 (and of T2) with dist = Omega(k)

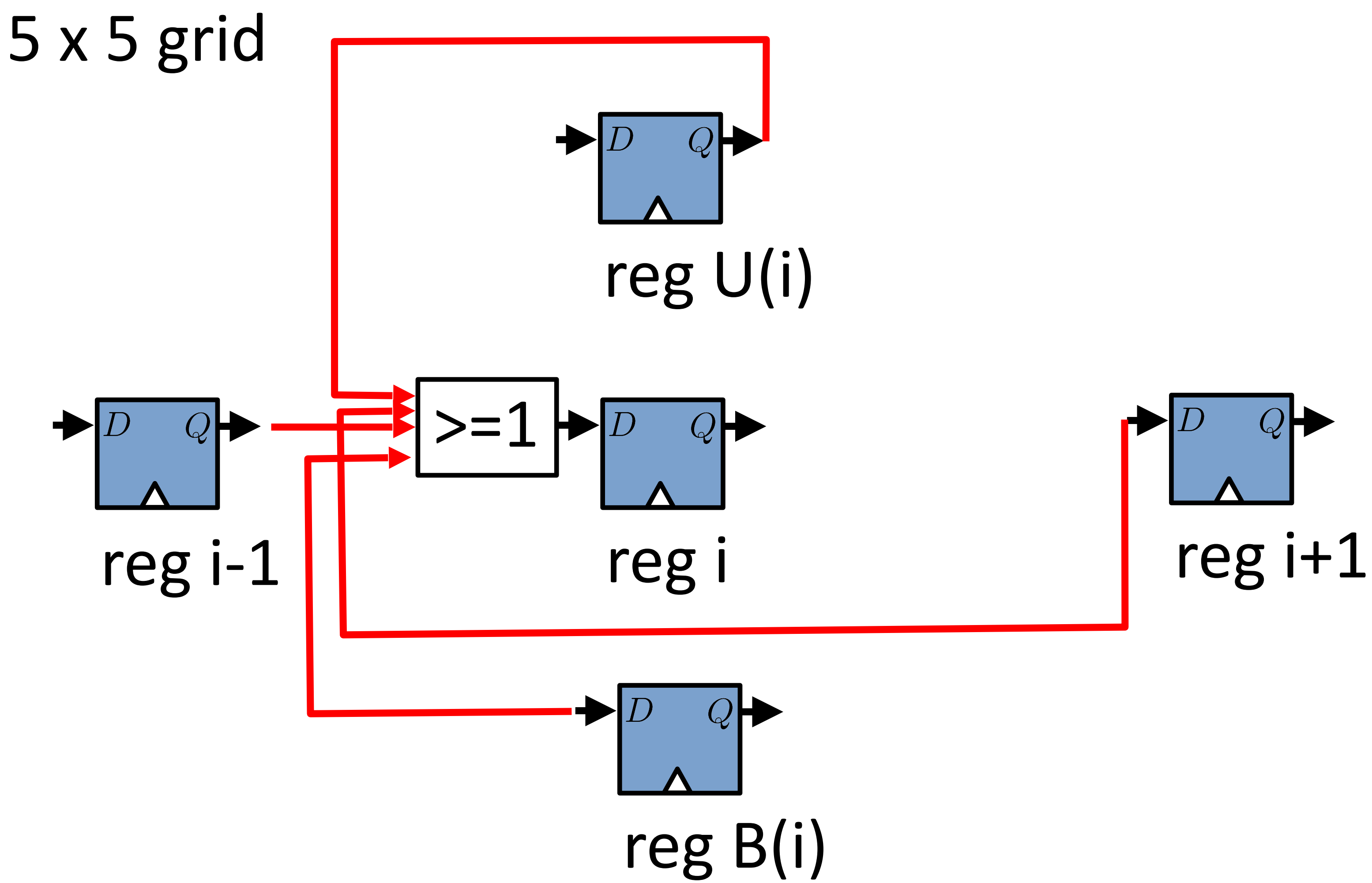
u, v two such nodes in T1



-> 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



-> live session