International Symposium on Physical Design

Soft-Clustering Driven Flip-flop Placement Targeting Clock-induced OCV

Dimitrios Mangiras Giorgos Dimitrakopoulos

Democritus University of Thrace, Xanthi, Greece

Pavlos Mattheakis Pierre-Olivier Ribet

Mentor, a Siemens Business, Grenoble, France

ISPD 2020, Taipei, Taiwan

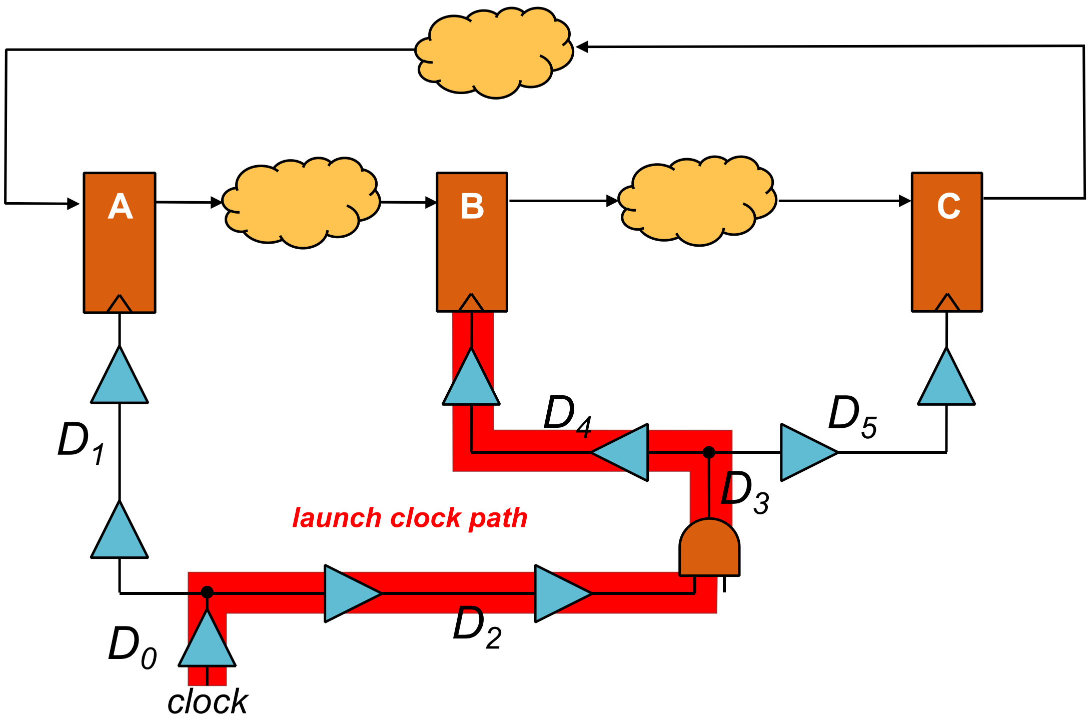
On-Chip Variations

- Variability is inherent to semiconductor manufacturing
- Variations make cells in both data and clock paths to be slower or faster than expected
- Temperature and voltage drops affect dynamically the timing of the design
- Extra margins needed to cover OCV degradations
  - Limits potential timing performance

- **Our focus is to alleviate the effect of clock induced OCV**
  - The variability in clock latency and how this is affected by the structure of the clock tree

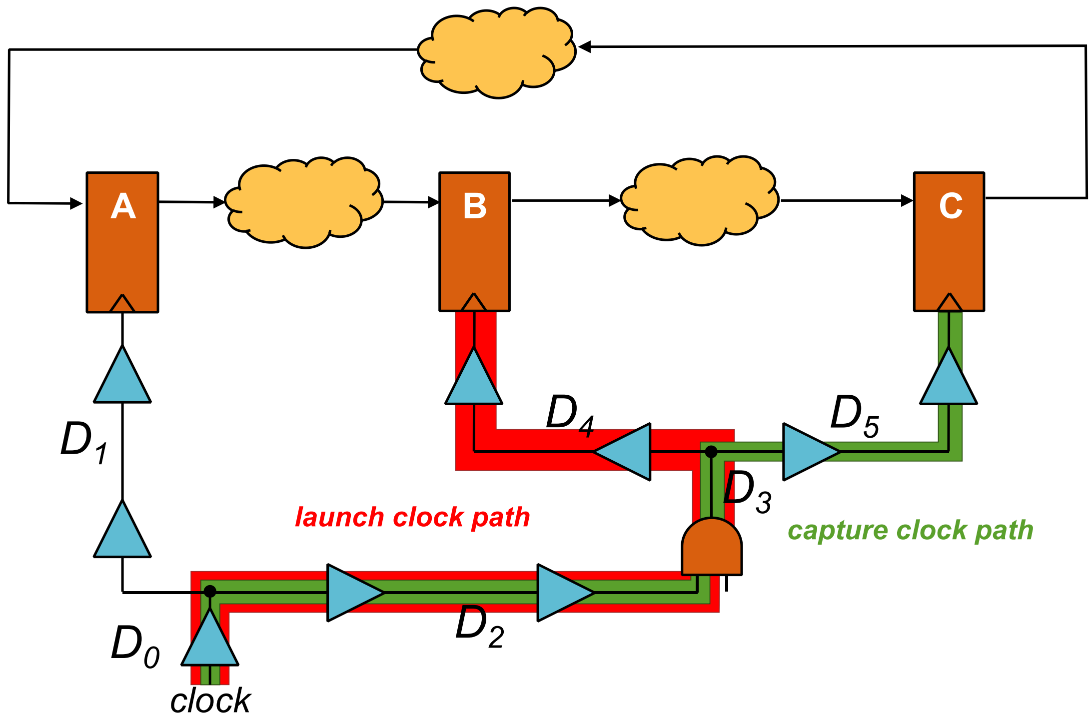
Common Path Pessimism Removal - Setup analysis

- OCV can make the launch clock path to be slower



Common Path Pessimism Removal - Setup analysis

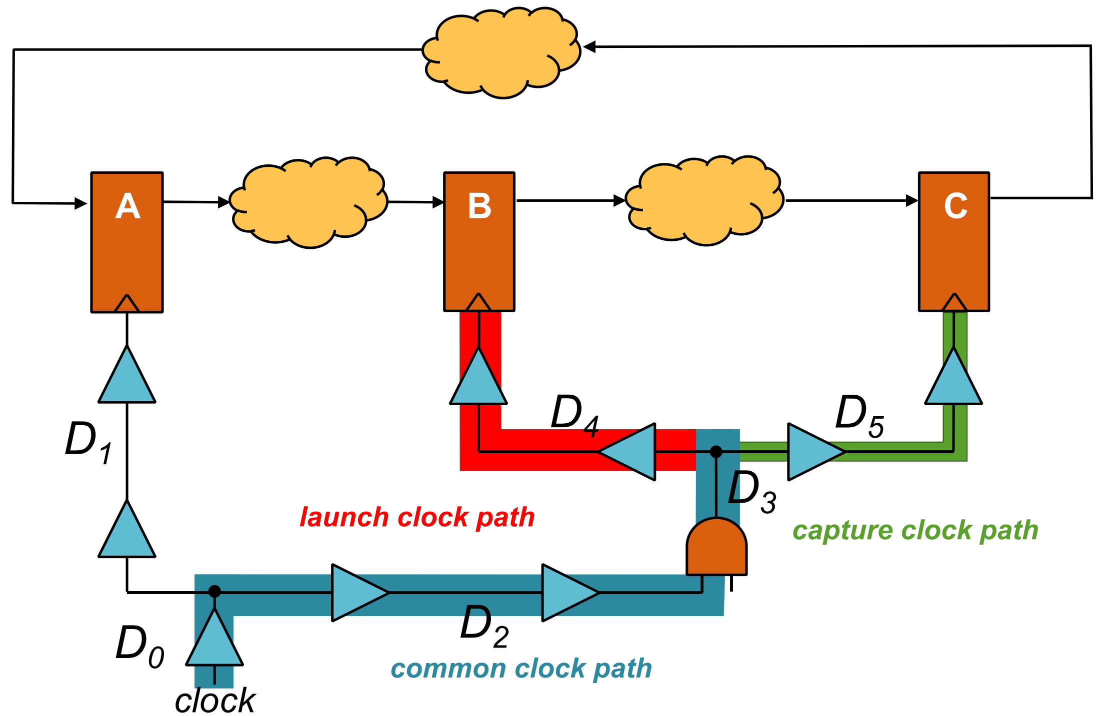
- OCV can make the launch clock path to be slower
- OCV can make the capture clock path to be faster



Common Path Pessimism Removal - Setup analysis

- OCV can make the launch clock path to be slower

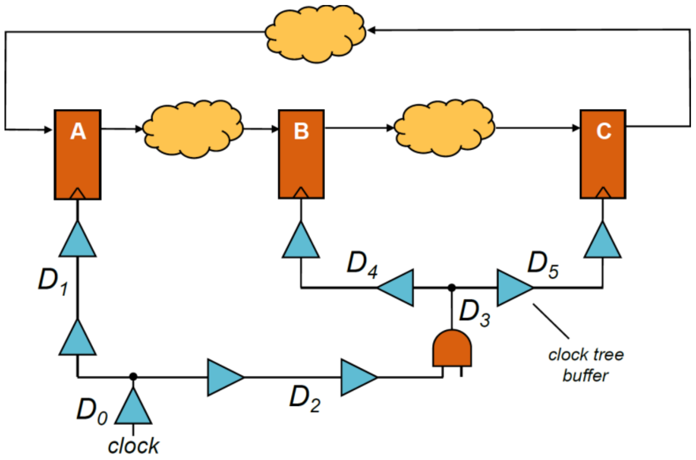
- OCV can make the capture clock path to be faster



- Cells in the common clock path can <u>not</u> be slower and faster <u>simultaneously</u> for both launch and capture paths

- CPPR discards this pessimism in STA

Main idea

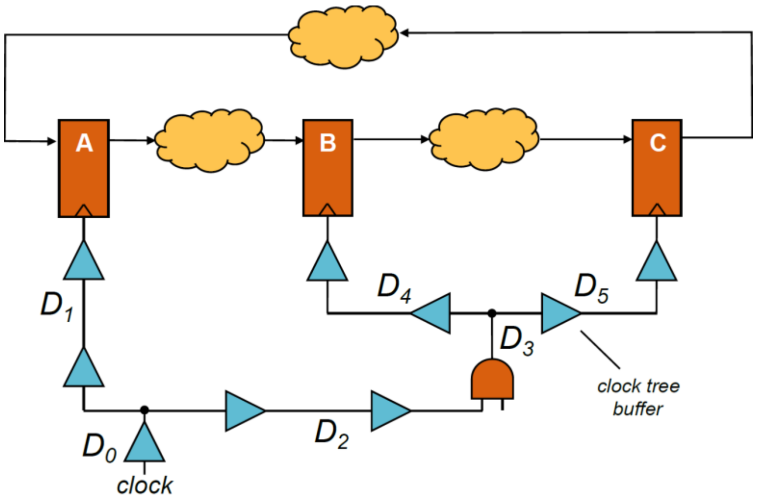


- Produce clock trees with as many common paths as possible
- Closely placed cells are most probably driven by the same clock path
- **Move relevant clocked cells closer**
  - CTS will drive them with the same buffer paths
  - Reduce implicitly the effect of timing degradation due to OCV

- Skew = launch_path_delay – capture_path_delay
- A → B: $skew_{A \to B}^{setup} = D_1^{max} - (D_2^{min} + D_3^{min} + D_4^{min})$
- B → C: $skew_{B \to C}^{setup} = D_4^{max} - D_5^{min}$
- C → A: $skew_{C \to A}^{setup} = (D_2^{max} + D_3^{max} + D_5^{max}) - D_1^{min}$

Main idea

$$A \rightarrow B: skew^{setup}_{A\rightarrow B} = D_1^{max} - (D_2^{min} + D_3^{min} + D_4^{min})$$
$$B \rightarrow C: skew^{setup}_{B\rightarrow C} = D_4^{max} - D_5^{min}$$
$$C \rightarrow A: skew^{setup}_{C\rightarrow A} = (D_2^{max} + D_3^{max} + D_5^{max}) - D_1^{min}$$

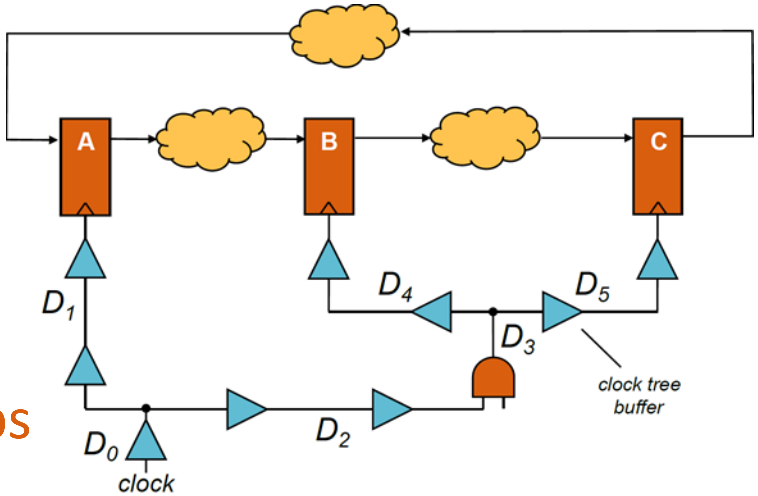


- $D_i^{max} = D_i(1+\gamma)$ and $D_i^{min} = D_i(1-\gamma)$
  - D_i = delay **without** OCV
  - γ is the derating factor

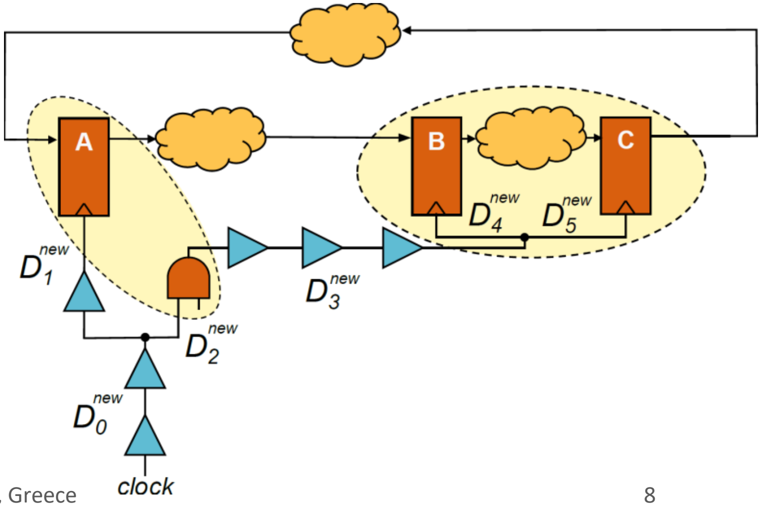
- $A \rightarrow B: skew^{setup}_{A\rightarrow B} = (D_1 - D_2 - D_3 - D_4) + \gamma(D_1 + D_2 + D_3 + D_4)$
- $B \rightarrow C: skew^{setup}_{B\rightarrow C} = (D_4 - D_5) + \gamma(D_4 + D_5)$
- $C \rightarrow A: skew^{setup}_{C\rightarrow A} = (D_2 + D_3 + D_5 - D_1) + \gamma(D_2 + D_3 + D_5 + D_1)$

How to reduce OCV impact

$$A \rightarrow B: \text{skew}^{\text{setup}}_{A\rightarrow B} = (D_1 - D_2 - D_3 - D_4) \quad + \gamma(D_1 + D_2 + D_3 + D_4)$$
$$B \rightarrow C: \text{skew}^{\text{setup}}_{B\rightarrow C} = \qquad (D_4 - D_5) \qquad + \gamma(D_4 + D_5)$$
$$C \rightarrow A: \text{skew}^{\text{setup}}_{C\rightarrow A} = (D_2 + D_3 + D_5 - D_1) + \gamma(D_2 + D_3 + D_5 + D_1)$$

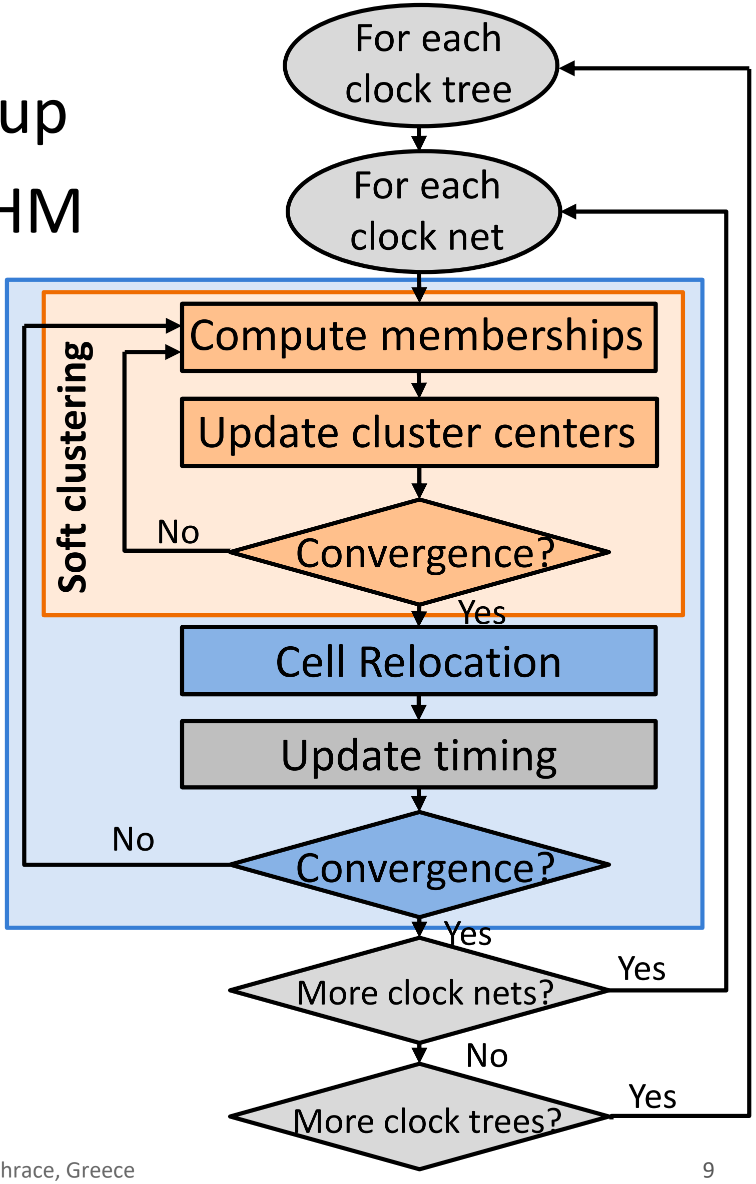


- Minimize D_4 and D_5 by moving flip-flops B and C closer
  - This movement guides CTS to drive flip-flops B and C with a common clock path increasing their common clock delay
- Decrease D_1 and D_2 clock path delays
  - Move the clock gater closer to flip-flop A
  - CTS will also produce a longer common path for A and the clock gater



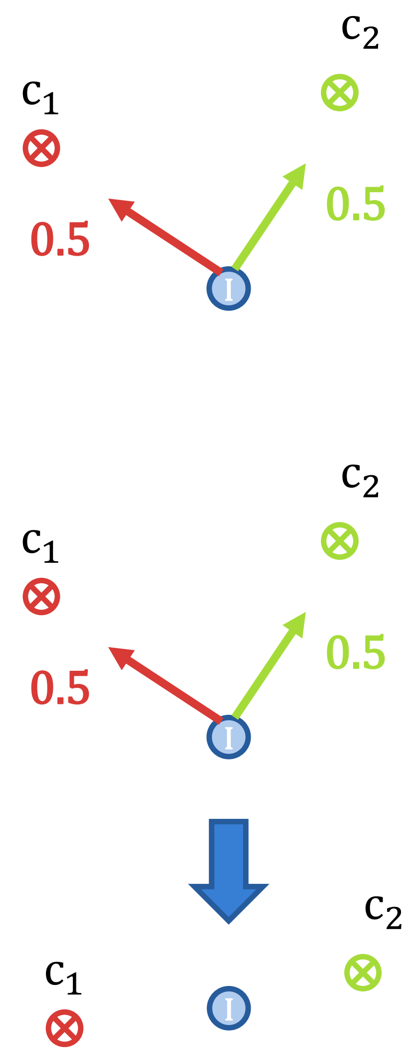
Overview of the proposed algorithm

- Traverse clock hierarchy bottom-up
- Clock cells are clustered using k-HM soft clustering
  - Compute cells-to-cluster memberships
  - Cluster centers are updated
- Clock cells are relocated to approach the new position
- Update routing and timing information



Why soft clustering

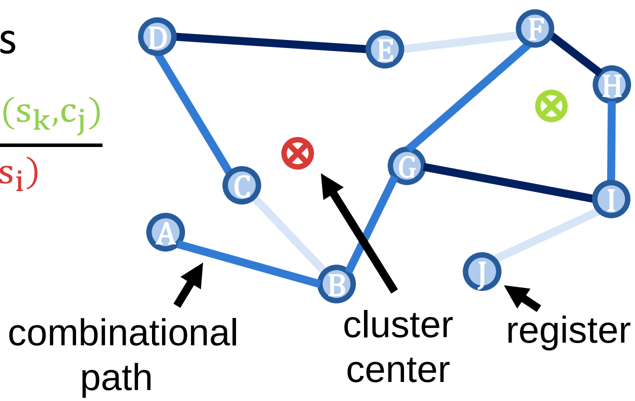
- **Hard** clustering : each point belongs to <u>only one</u> cluster
  - Cell moving towards only one cluster
  - Ping-pong effect choosing the best cluster
  - Unable to handle cases when two clusters have the same cost
    - Slow convergence
- **Soft** clustering : each point belongs to <u>every</u> cluster
  - All clusters contribute to the cell's relocation
    - New location is the weighted mean of all cluster centers
  - Each cell approaches both two clusters having the same cost
    - <u>No need</u> to choose one
    - More trusted convergence due to less oscillations



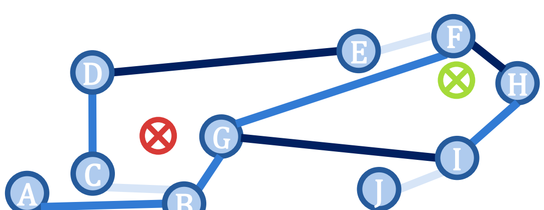
K-harmonic means soft clustering

- **Hard** clustering : each point belongs to only one cluster
- **Soft** clustering : each point belongs to every cluster
  - Basic: $m(s_i, c_j) = d(s_i, c_j)$
    - Only physical distance between points and clusters
  - Ours: $m(s_i, c_j) = \alpha \cdot d(s_i, c_j) + (1 - \alpha) \cdot \frac{\sum_{\forall s_k} t(s_k, s_i) d(s_k, c_j)}{\sum_{\forall s_k} t(s_k, s_i)}$
    - Physical distance of cell to cluster
    - Physical proximity of cell's neighbors to cluster
    - How critical are the neighbors

$$d(s_i, c_j) = \frac{\left\| s_i - c_j \right\|^{-p-2}}{\sum_{k=1}^{K} \left\| s_i - c_j \right\|^{-p-2}}$$

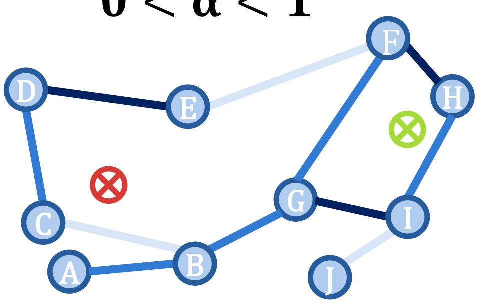


$\alpha = 1$



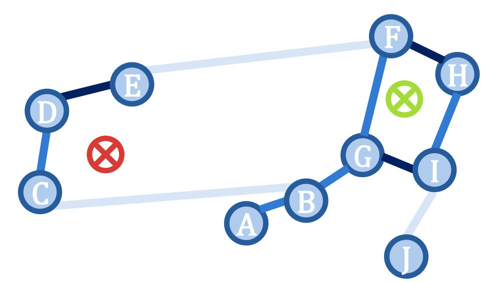
Similar to register clumping - only physical distance

$0 < \alpha < 1$



Intermediate state

$\alpha = 0$



Clusters based only on physical proximity of neighbors

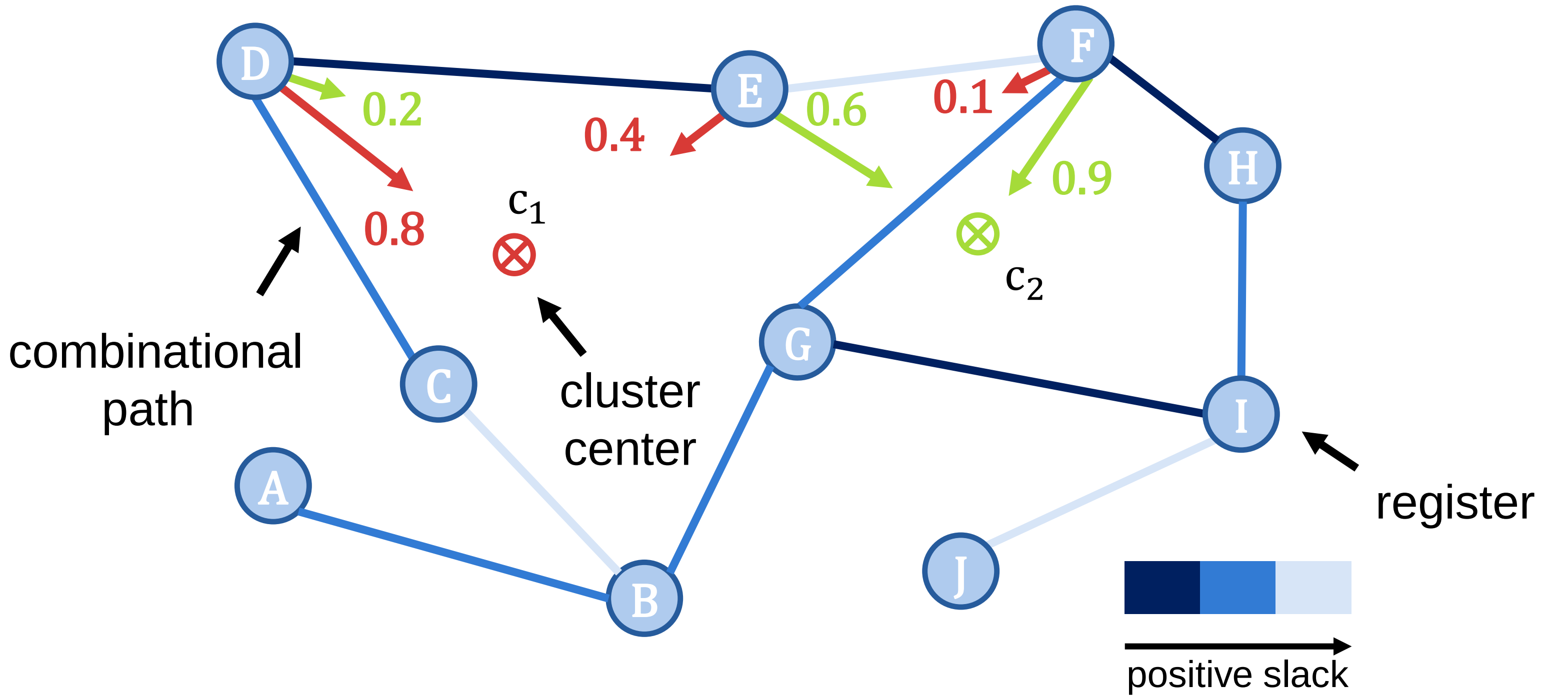
Compute membership weights

- Start computing the physical distance probability of each cell s_i to each cluster c_j center

$$d(s_i, c_j) = \frac{\left\|s_i - c_j\right\|^{-p-2}}{\sum_{k=1}^{K} \left\|s_i - c_j\right\|^{-p-2}}$$

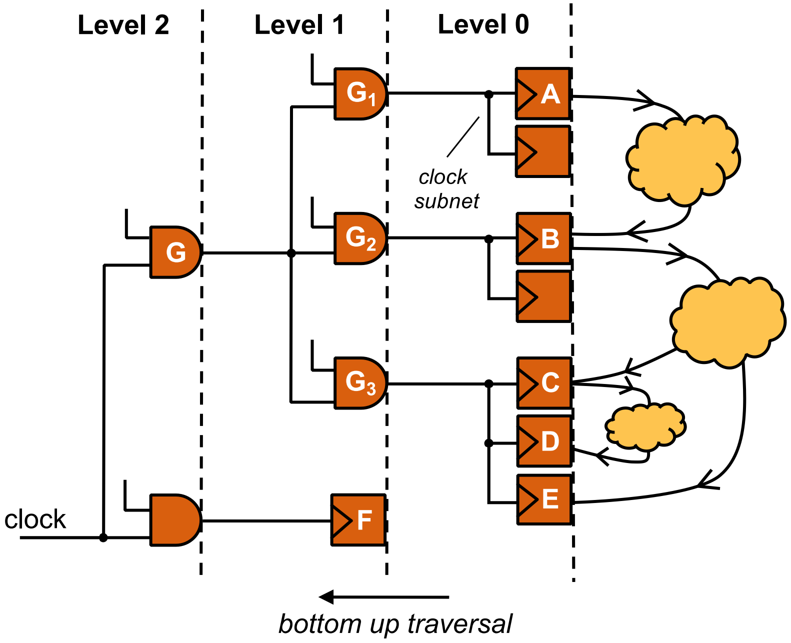
$d(s_D, c_1) = 0.8 \quad d(s_E, c_1) = 0.4 \quad d(s_F, c_1) = 0.1 \quad \ldots$

$d(s_D, c_2) = 0.2 \quad d(s_E, c_2) = 0.6 \quad d(s_F, c_2) = 0.9 \quad \ldots$



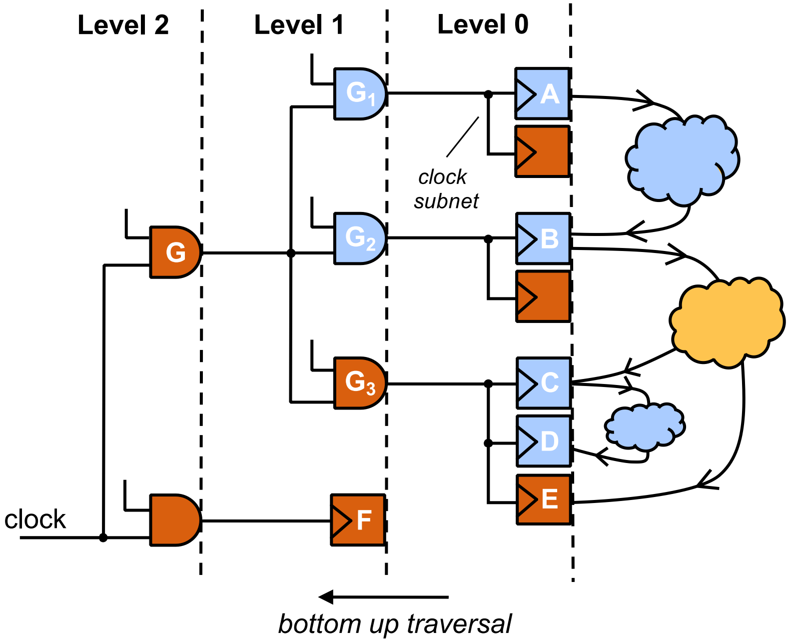
Identifying which clock cells should be placed closer

- **Timing neighbors** : Two clock cells with clock pins on the same net and belong to the launch and capture parts of a constrained timing path



Identifying which clock cells should be placed closer

- Timing neighbors : Two clock cells with clock pins on the same net and belong to the launch and capture parts of a constrained timing path



- **Flip-flop C is a timing neighbor of flip-flop D** but not of flip flop B

- Clock gater G1 is a timing neighbor of clock gater G2

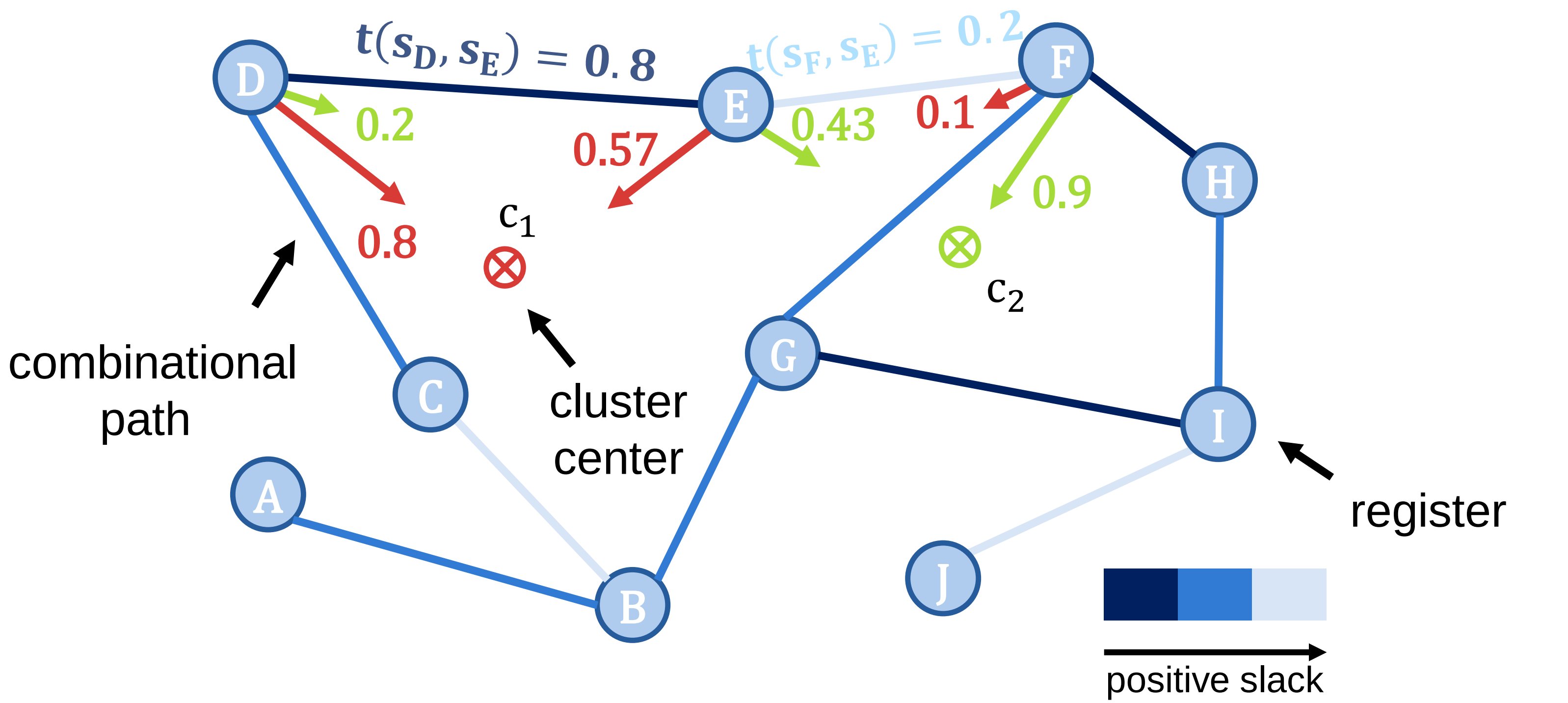
Compute membership weights

- Registers F and D are the timing neighbors of E
- Path DE is **more critical** than EF
  - $t(s_D, s_E) = 0.8$ and $t(s_F, s_E) = 0.2$

$$m(s_i, c_j) = \alpha \cdot d(s_i, c_j) + (1 - \alpha) \cdot \frac{\sum_{\forall s_k} t(s_k, s_i) d(s_k, c_j)}{\sum_{\forall s_k} t(s_k, s_i)}$$

$$m(s_E, c_1) = 0.35 \cdot d(s_E, c_1) + 0.65 \cdot \left(t(s_D, s_E) \cdot d(s_D, c_1) + t(s_F, s_E) \cdot d(s_F, c_1)\right) = 0.57$$

$$m(s_E, c_2) = 0.35 \cdot d(s_E, c_2) + 0.65 \cdot \left(t(s_D, s_E) \cdot d(s_D, c_2) + t(s_F, s_E) \cdot d(s_F, c_2)\right) = 0.43$$



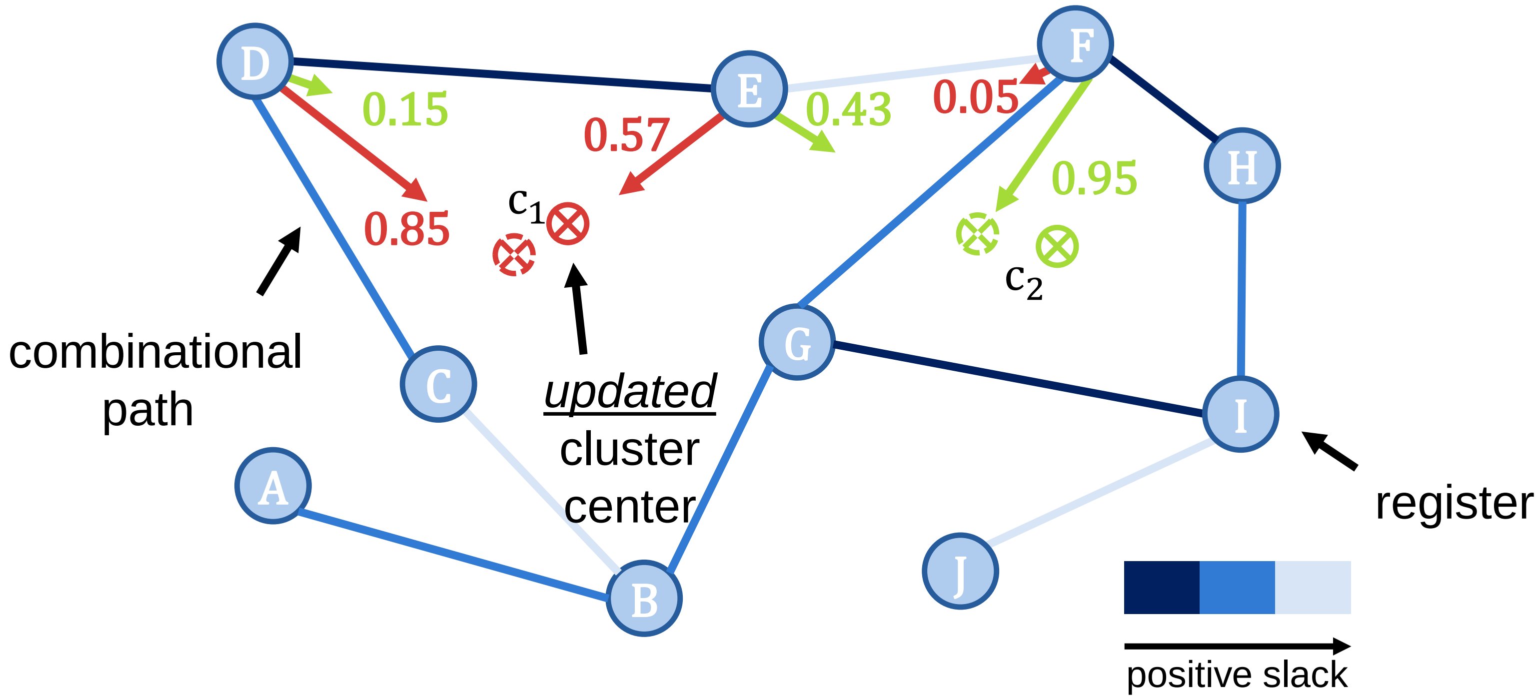
Update cluster centers

- All clock cells membership weights are updated
- Cluster centers are recomputed using kHM formulas

$$w(s_i) = \frac{\sum_{k=1}^{K} \left\| s_i - c_j \right\|^{-p-2}}{\left(\sum_{k=1}^{K} \left\| s_i - c_j \right\|^{-p}\right)^2}$$

$$x_{c_j} = \frac{\sum_{\forall s_i} m(s_i, c_j) w(s_i) x_{s_i}}{\sum_{\forall s_i} w(s_i)}$$

$$y_{c_j} = \frac{\sum_{\forall s_i} m(s_i, c_j) w(s_i) y_{s_i}}{\sum_{\forall s_i} w(s_i)}$$

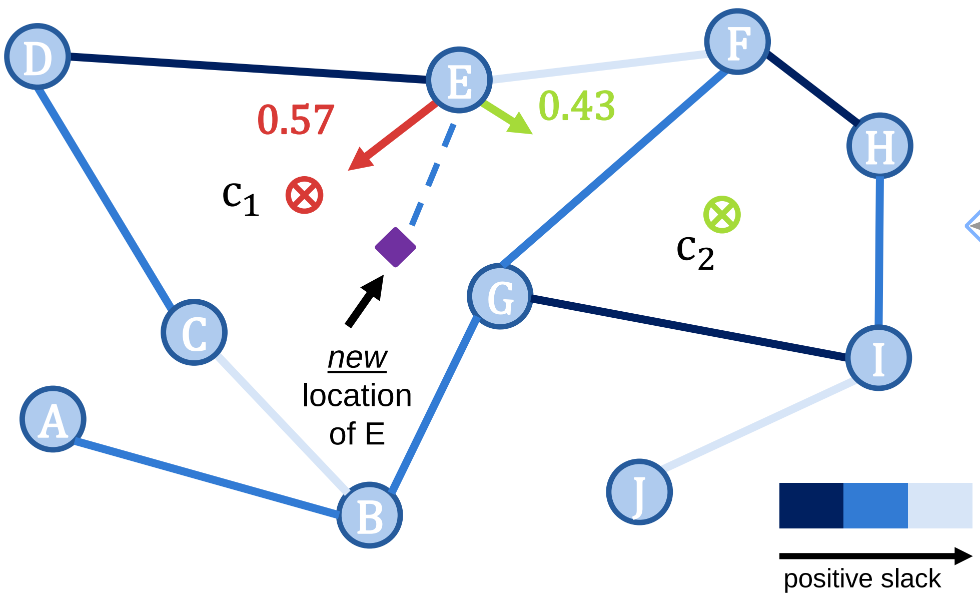


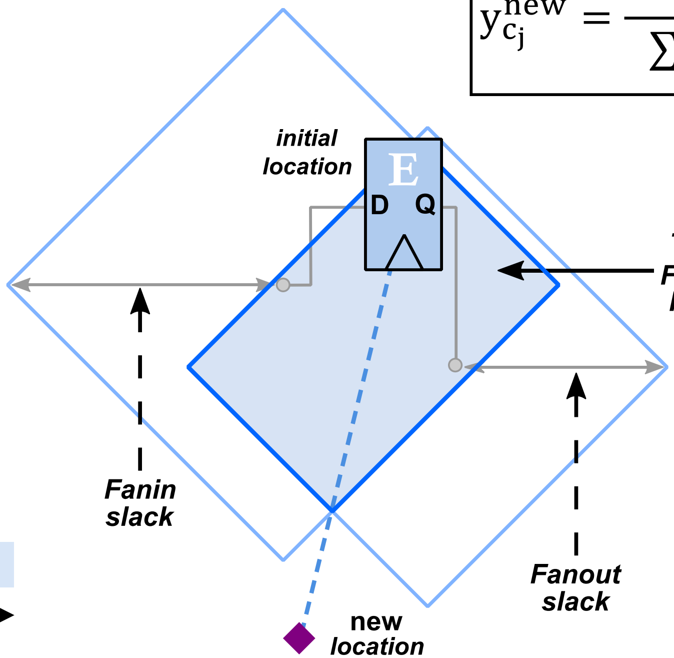
Cell relocation

- Cell approaches the weighted mean location of all neighbor clusters
  - Closer to cluster centers that clock cell has higher membership weights for them
- Cell movement is limited by :
  - Its Timing Feasible Region (TFR) to prevent timing degradation
  - A maximum allowed displacement

$$x_{c_j}^{new} = \frac{\sum_{\forall c_j} m(s_i, c_j) x_{s_i}}{\sum_{\forall c_j} m(s_i, c_j)}$$

$$y_{c_j}^{new} = \frac{\sum_{\forall c_j} m(s_i, c_j) y_{s_i}}{\sum_{\forall c_j} m(s_i, c_j)}$$



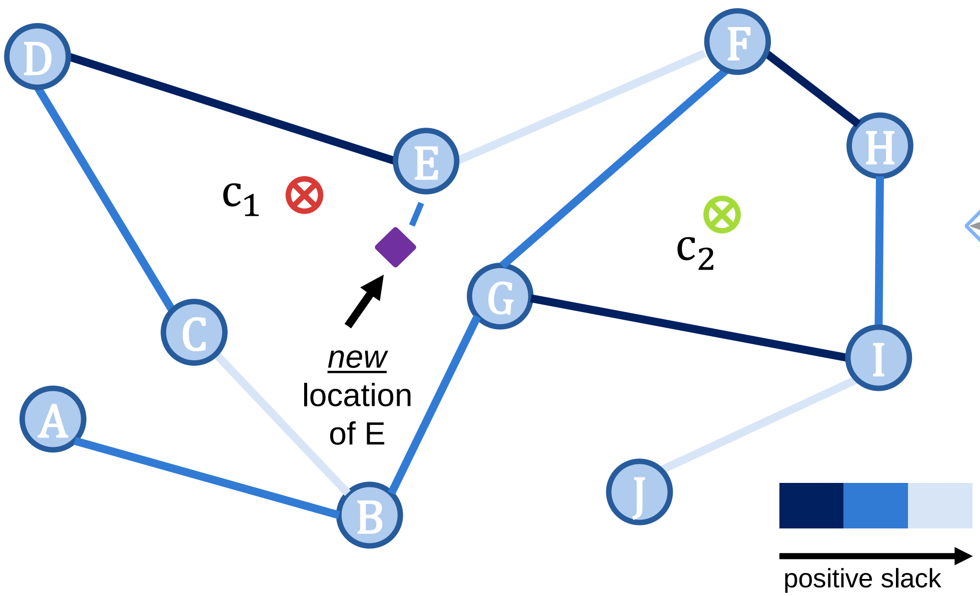


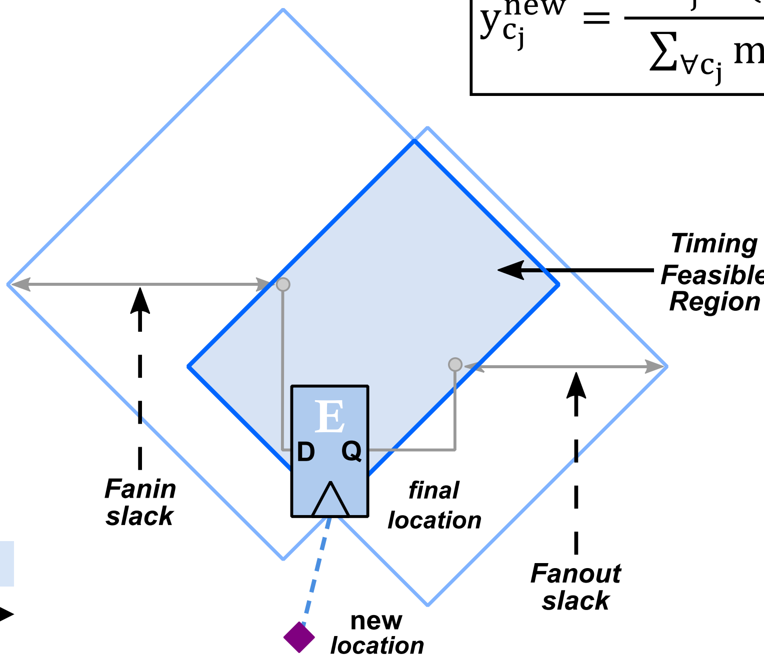
Cell relocation

- Cell approaches the weighted mean location of all neighbor clusters
  - Closer to cluster centers that clock cell has higher membership weights for them
- Cell movement is limited by :
  - Its Timing Feasible Region (TFR) to prevent timing degradation
  - A maximum allowed displacement

$$x_{c_j}^{new} = \frac{\sum_{\forall c_j} m(s_i, c_j) x_{s_i}}{\sum_{\forall c_j} m(s_i, c_j)}$$

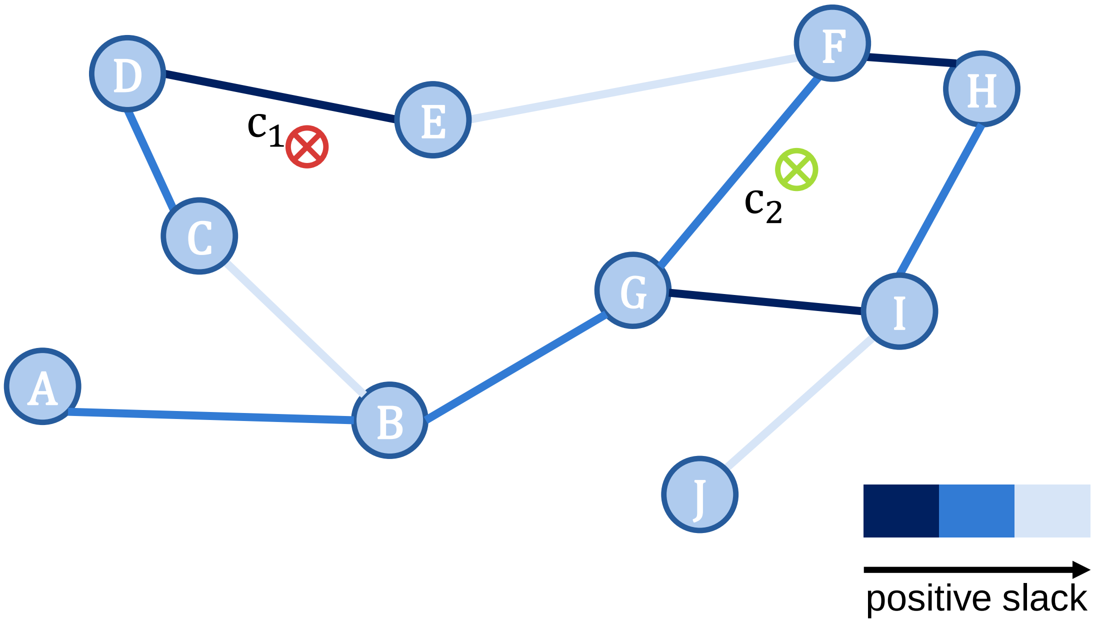
$$y_{c_j}^{new} = \frac{\sum_{\forall c_j} m(s_i, c_j) y_{s_i}}{\sum_{\forall c_j} m(s_i, c_j)}$$





Cell relocation

- Once a cell is relocated, it's *new* physical probabilities **change the membership weights** of every timing neighbor
  - The membership weights are updated
  - Cells approach even more clusters where their more timing critical neighbors are closer



Experimental Setup

- The proposed method has been integrated in Mentor's Nitro-SoC P&R tool
  - Executed after global placement and data-path optimization and before CTS
- Tested on six real industrial designs (82K – 1.54M cells)
  - In all designs (except D2) Advanced OCV derates are set
    - D2 has simple OCV derates
- For comparison 2 flows are used:
  - The "Base" flow is the industrial quality flow
  - The "Cluster" runs the physical register clustering of Wu *et al.* DAC 2016 that targets low power clock trees

Experimental Results – Timing comparison (1/3)

- Timing reports are collected at the end of post-CTS optimizations

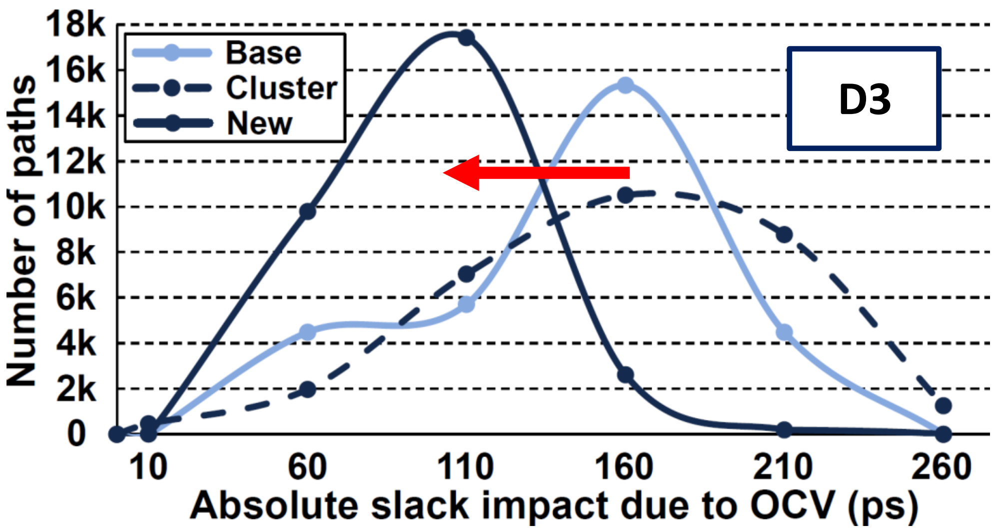
Design		Setup		Hold	
		WNS (ps)	TNS (ns)	WHS (ps)	THS (ns)
D1 – 14nm 82K cells 4.5K regs	Base	-337.2	-29.7	0.0	0.0
	Cluster	-320.0	-28.8	0.0	0.0
	New	**-297.0**	**-25.2**	**0.0**	**0.0**
D2 – 28nm 199K cells 16K regs	Base	-396.0	-885.0	-134.0	-0.6
	Cluster	-409.0	-1148.1	-104.0	-7.5
	New	**-368.0**	**-768.2**	**-1.0**	**-0.1**
D3 – 16nm 542K cells 35K regs	Base	-43.0	-0.6	-15.0	-0.6
	Cluster	-137.0	-0.9	-17.0	-0.1
	New	**-24.0**	**-0.3**	**-14.0**	**-0.1**

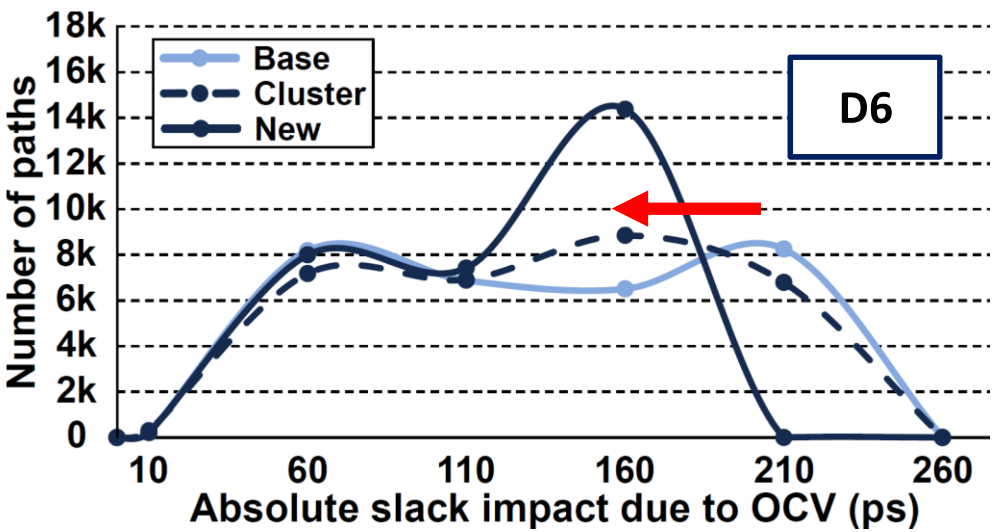
Experimental Results – Timing comparison (2/3)

- Proposed method achieves the best WNS and TNS for all designs and for both setup and hold analysis
- **Setup TNS** reduced by **42%** and **hold THS** by **73%**, on average

Design		Setup		Hold	
		WNS (ps)	TNS (ns)	WHS (ps)	THS (ns)
D4 – 22nm 557K cells 47K regs	Base	-232.0	-564.2	0.0	0.0
	Cluster	-288.0	-677.0	0.0	0.0
	New	**-223.0**	**-392.5**	**0.0**	**0.0**
D5 – 16nm 611K cells 45K regs	Base	-802.0	-442.9	-35.0	-1.4
	Cluster	-668.0	-487.0	-49.0	-0.9
	New	**-379.0**	**-100.6**	**-30.0**	**-0.6**
D6 – 14nm 1545K cells 71K regs	Base	-103.0	-41.1	-93.0	-6.0
	Cluster	-68.0	-20.6	-170.0	-20.2
	New	**-59.0**	**-16.4**	**-68.0**	**-1.9**

Experimental Results – Timing comparison (3/3)





- Measured the difference of the path's late slack with and without OCV derates for the 30K most critical paths
- Split the impact values to bins and created the histograms
- Proposed method **restructured most heavily affected paths** by OCV
  - In D3, the peak moved from 160ps to around 60-110ps
  - In D6, the histogram's peak moved from 210ps to 160ps

Experimental Results – Clock tree complexity (1/2)

- Average clock latency is reported
- There are insignificant differences in the clock tree QoR

Design		Buffers	WL(mm)	Cap (pF)	Lat (ps)	Skew (ps)
D1	Base	64	18.7	8.4	352	88
	Cluster	65	19.0	8.5	320	88
	New	**64**	**18.1**	**8.2**	**341**	**108**
D2	Base	342	103.6	33.6	635	162
	Cluster	300	102.6	33.3	604	134
	New	**303**	**99.5**	**32.4**	**535**	**124**
D3	Base	1285	211.9	85.5	690	164
	Cluster	1201	210.8	84.7	740	140
	New	**1216**	**211.2**	**84.2**	**677**	**166**

Experimental Results – Clock tree complexity (2/2)

Design		Buffers	WL(mm)	Cap (pF)	Lat (ps)	Skew (ps)
D4	Base	6650	326.1	150.0	661	98
	Cluster	6688	338.2	151.5	599	110
	New	**6637**	**327.2**	**149.8**	**679**	**123**
D5	Base	5719	250.4	238.3	1642	143
	Cluster	6009	267.1	250.9	1749	142
	New	**5611**	**253.5**	**239.9**	**1646**	**112**
D6	Base	9463	569.6	774.0	1911	197
	Cluster	9540	580.7	778.5	1808	236
	New	**9650**	**571.1**	**776.0**	**1540**	**173**

Conclusions

- Clocked cell relocation increases the common clock paths on timing paths highly affected by OCV
  - Selected cells move closer their timing neighbors $\Longrightarrow$ CTS can share clock paths for them
  - An iterative soft-clustering algorithm guides the overall process
- The method is evaluated on six real industrial designs and achieved the best QoR
  - Setup improved by 28% and hold by 45% on average
  - The overall clock tree complexity remained the same