# Sequential Delay Budgeting with Interconnect Prediction

**Chao-Yang Yeh and Malgorzata Marek-Sadowska**

Department of Electrical and Computer Engineering,
University of California, Santa Barbara

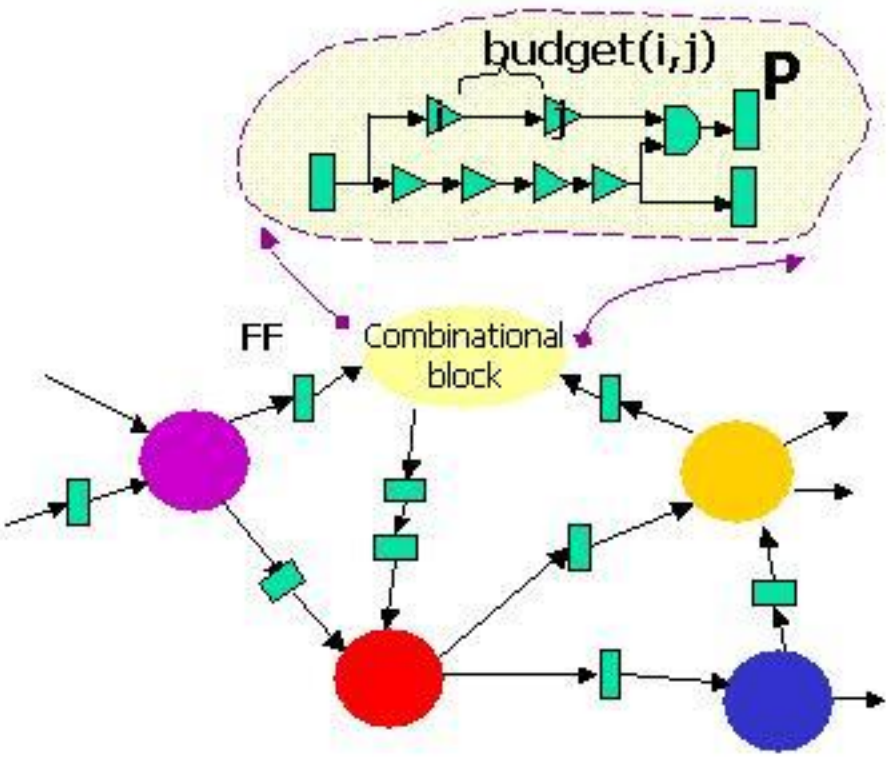
# Motivation

- Placement of circuit elements has a major impact on wire length/delay
- Timing-driven placement
  - Path-based algorithm
    - Analyze path delay explicitly during placement
  - Net-based algorithm
    - Net weight
    - 

# Net Delay Budgeting

Combinational budgeting

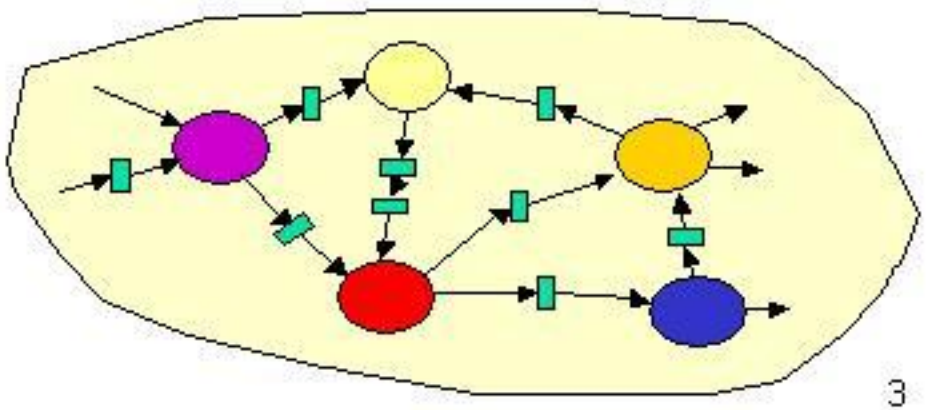
- Treat FFs like POs, PIs
- Optimize each block individually




Sequential Budgeting

- Solve the circuit as a whole by combining budgeting with retiming

1. Solve the sequential budgeting formulation. (Obtain clock skew for each FF)
2. Use skew-retiming equivalence relation to realize FF movement

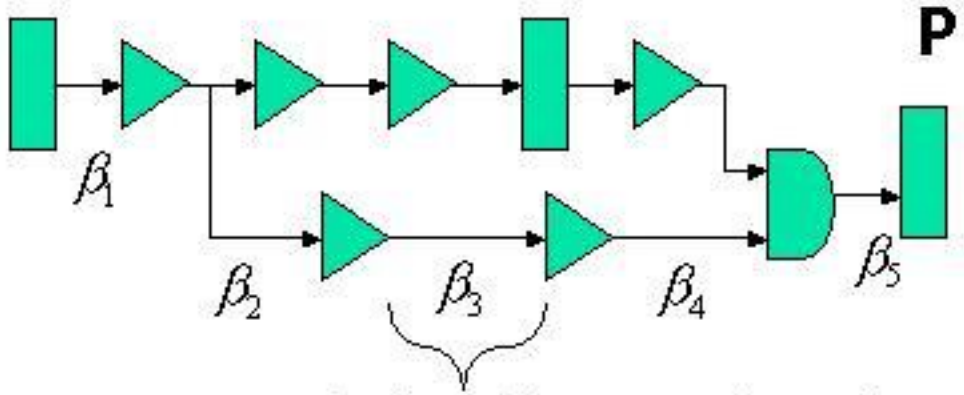


# Motivation (cont)

- Interconnect Prediction
  - Predicted Statistically
  - Predicted Analytical – hierarchical average length estimation (Donth79')
- New Ideas
  - Using combinational budgeting as a delay prediction method (predict individual net length)
  - Use prediction methods for sequential budgeting
    - Generate weight : [redacted], Average length estimation
    - Delay prediction : combinational budgeting

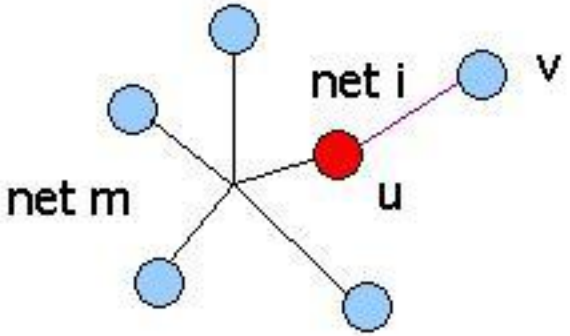
# Combinational Budgeting for interconnect delay prediction

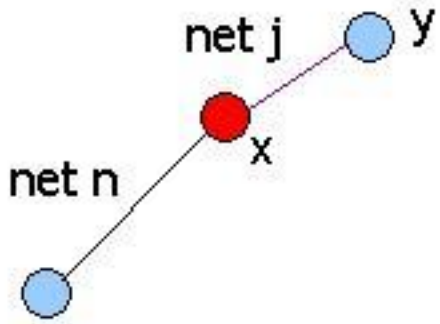
- Combinational Delay budgeting:
  - Predict individual net length/delay



# Mutual Contraction

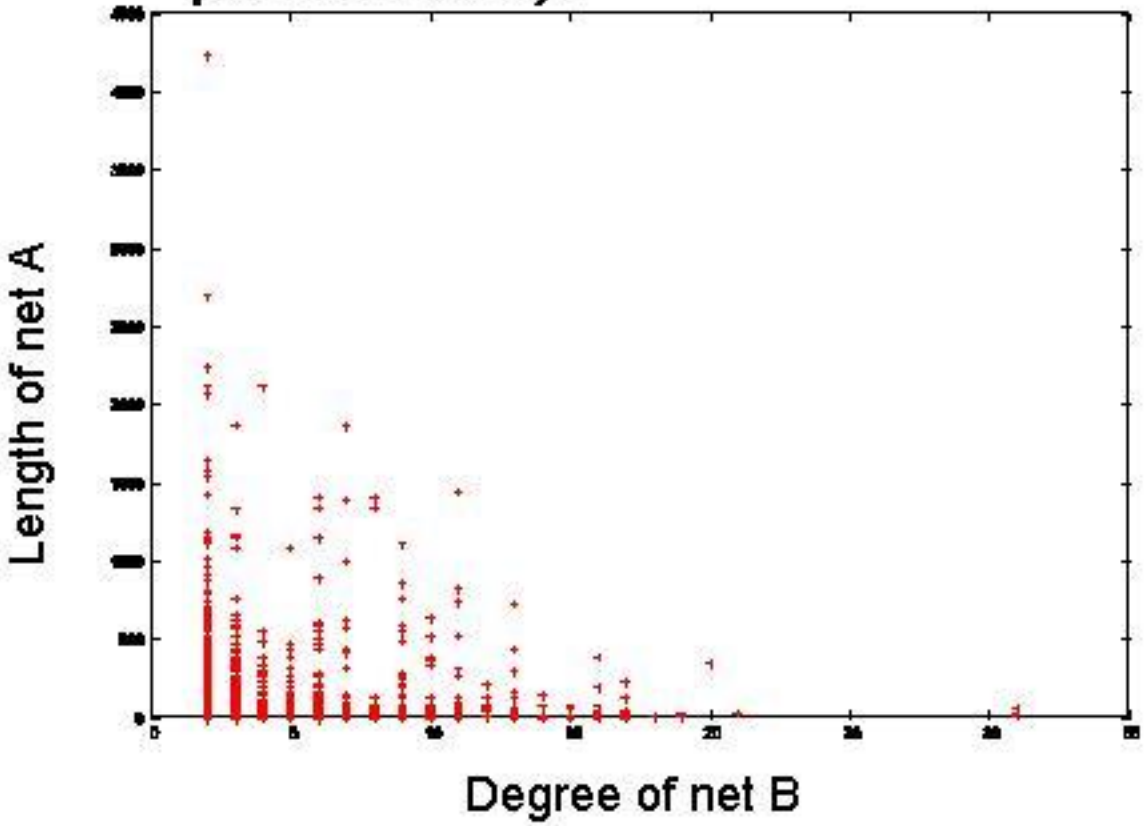
- **Mutual Contraction** (A new prediction method)
  - Placer tends to place strongly connected nodes in close proximity.
  - Placer puts more optimization effort on smaller-fanout nets
- **Derive mutual contraction metrics - estimate net length for a pair of nodes.**

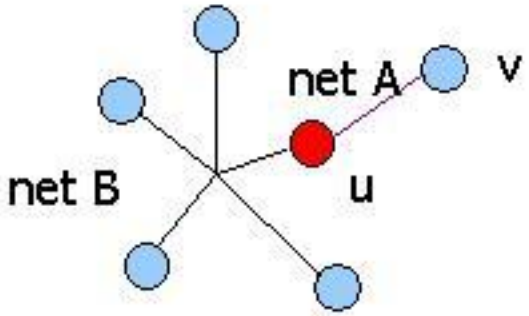


# Mutual Contraction experiments (1)

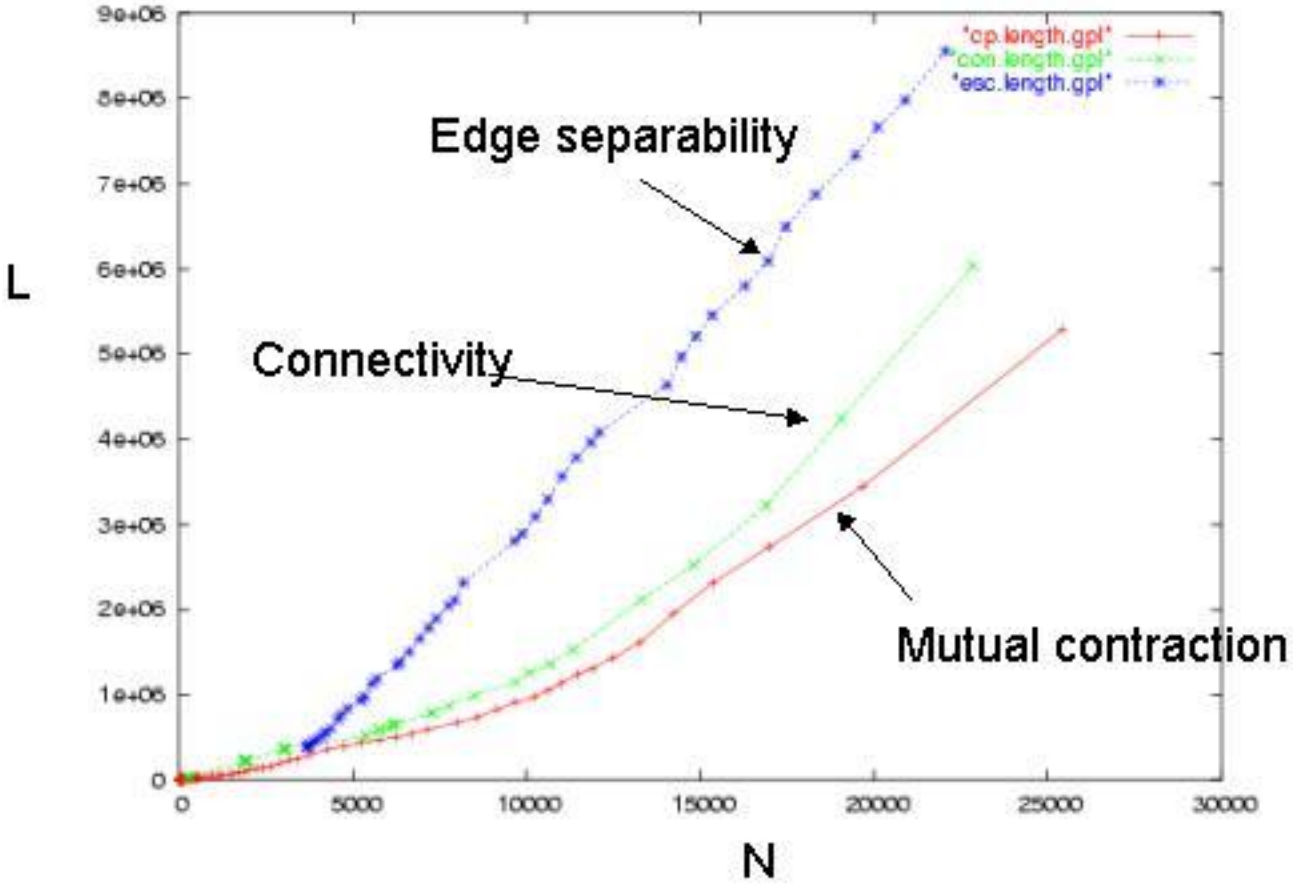
- Length distribution of 2-pin nets (in golden placement):



# Mutual Contraction experiments (2)

- Connection length prediction:



# Weighting function

- **Average Interconnect Lengths**
  - Predict average edge length for each partition level

$$L_k = \frac{4\times\left(\frac{4\lambda}{3}-\frac{1}{3\lambda}\right)+4\lambda}{6},\ \lambda = 2^{H-k}$$

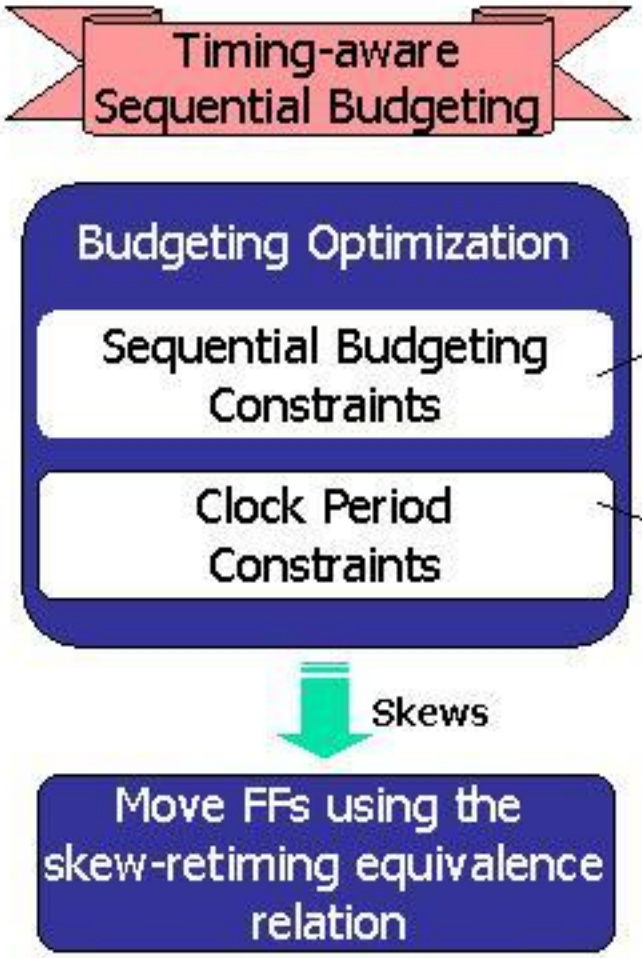
- Assign weight for each net
  - Assign more budget for edges predicted long

$$\alpha_{ij} = \left(\frac{1+e^{-w_N}}{2}\right)^{5/4} \times \left(2-e^{-L_k}\right)^5,\ e_{ij} \in net\ N, N\ is\ at\ level\ k$$

Mutual Contraction

Average Interconnect length

# Timing-aware Sequential Budgeting




Sequential Budgeting constraints:

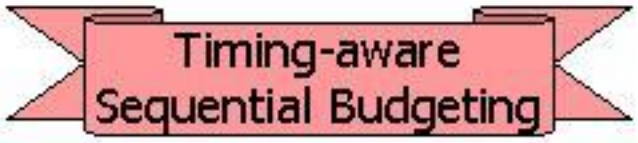
- Transform Path constraints to edge constraints
- Assign fan-in arrival time to each gate
- Add FFs into consideration

Clock period constraints:

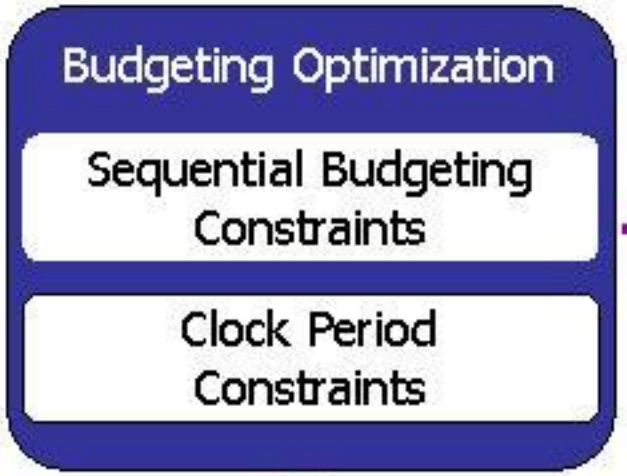
- Budgeting doesn't violate timing
- Give larger budget for paths predicted long

**Find the best budgeting that meet timing requirement**

# Including interconnect prediction



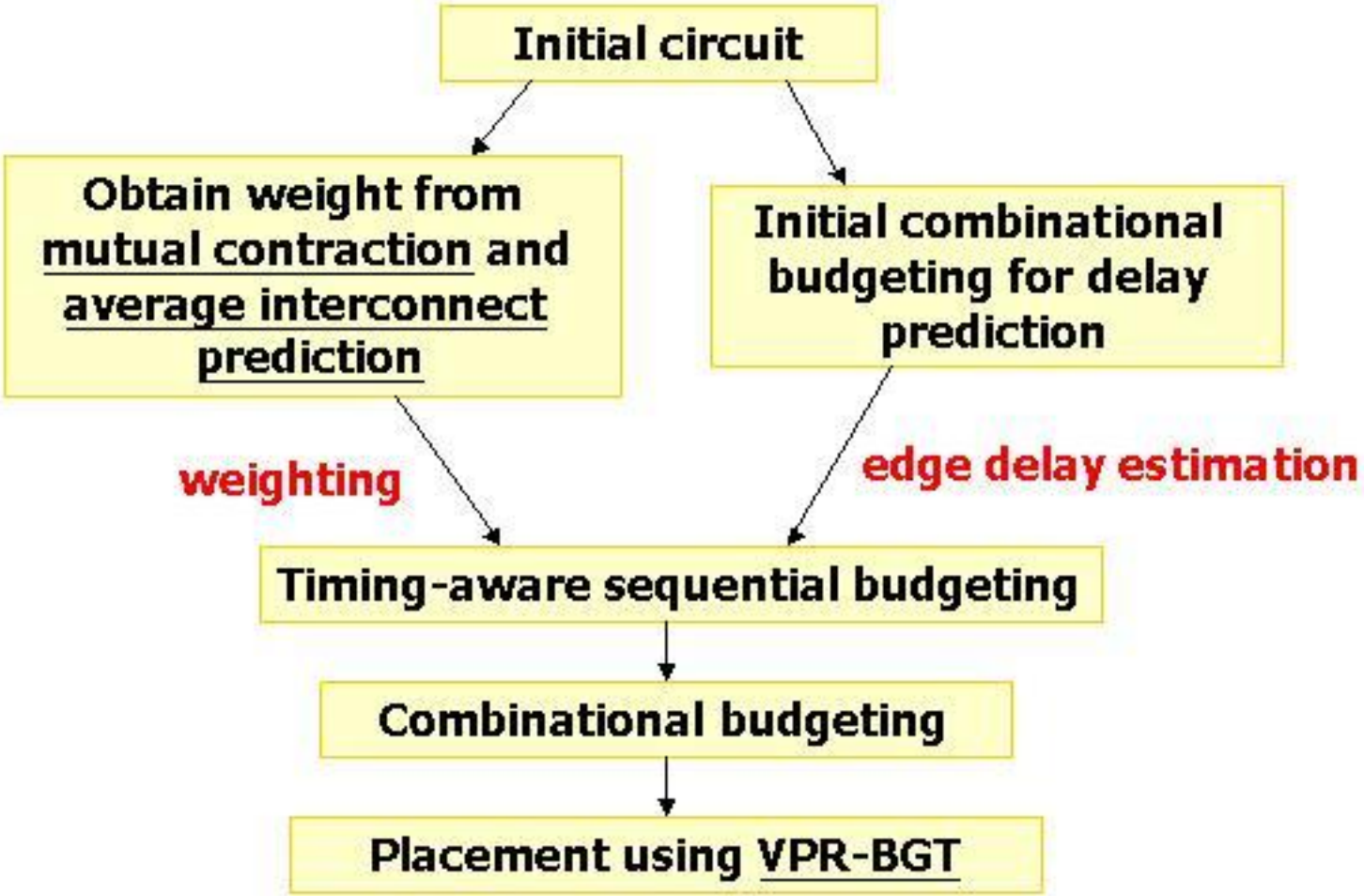

- Less dependent on previous placement




- Use interconnect prediction to generate weighting for each net
  - Net predicted to be longer => increase its budget

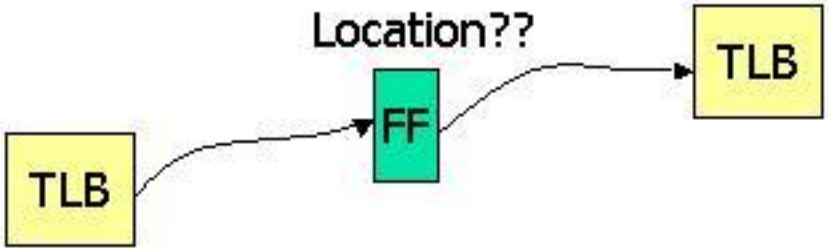
- Using combinational budgeting to predict interconnect delay

# Interconnect prediction with sequential budgeting



# Modified VPR Placer

- VPR – a simulated-annealing based FPGA placer
- VPR-FF – Decouple FFs and TLBs in placement.




- VPR-BGT – Budgeting-aware Placer
  - Penalize edges whose delay are larger than their budgets

# Experimental Results

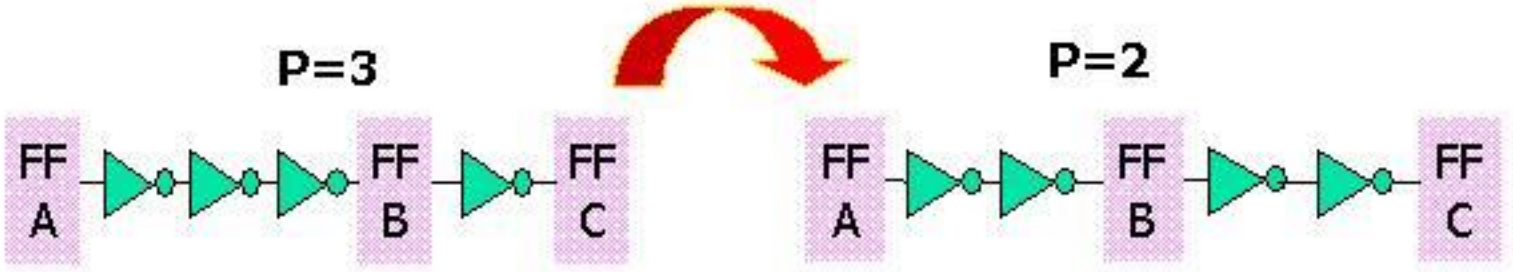
| *Circuit* | *VPR-FF* | *New Placement Flow* | *Uniform Weighting* | *Retiming with VPR-FF* |
|---|---|---|---|---|
| | P (ns) | P (ns) | P (ns) | P (ns) |
| bigkey | 7.75 | 5.68 | 5.60 | 7.54 |
| elliptic | 16.57 | 15.58 | 15.42 | 16.58 |
| s298 | 19.88 | 19.93 | 20.08 | 20.6 |
| s38417 | 12.70 | 11.28 | 11.39 | 12.70 |
| diffeq | 13.29 | 12.68 | 12.57 | 13.03 |
| tseng | 12.15 | 11.41 | 12.67 | 11.77 |
| dsip | 8.47 | 5.62 | 5.20 | 7.09 |
| clma | 24.96 | 22.38 | 24.67 | 24.95 |
| | 1 | 87.71% | 89.39% | 97.44% |

# Conclusion

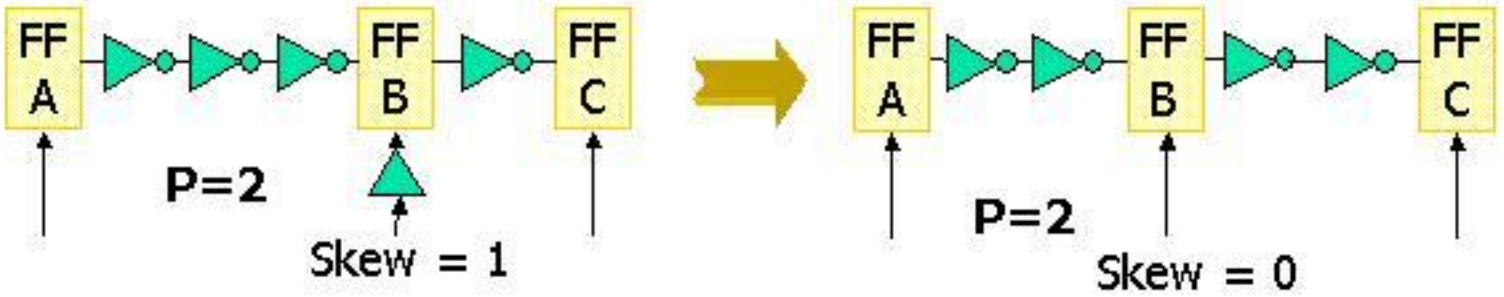
- Using delay budgeting to predict interconnect delay
- Incorporate interconnect prediction in sequential budgeting
- Apply in FPGA placement flow

# Retiming (extend the solution space for budgeting)

- Retiming: min-area, max-speed




- Skew-retiming equivalence
  - Use clock skew assignment to move FFs
    - Positive skew => FF move backward.
    - Negative skew => FF move forward.



# Sequential Delay Budgeting with application to FPGAs

Chao-Yang Yeh

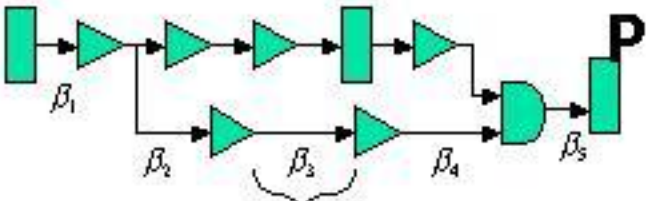
Malgorzata Marek-Sadowska

# Motivation

- Placement of circuit elements has a major impact on wire length/delay
- Timing-driven placement
  - Path-based algorithm
    - Analyze path delay explicitly during placement
  - Net-based algorithm
    - Net weight
    - 

# Motivation (cont)

- Net Delay budgeting:
  - Done for combinational blocks
  - Spread appropriate delay budget along the path



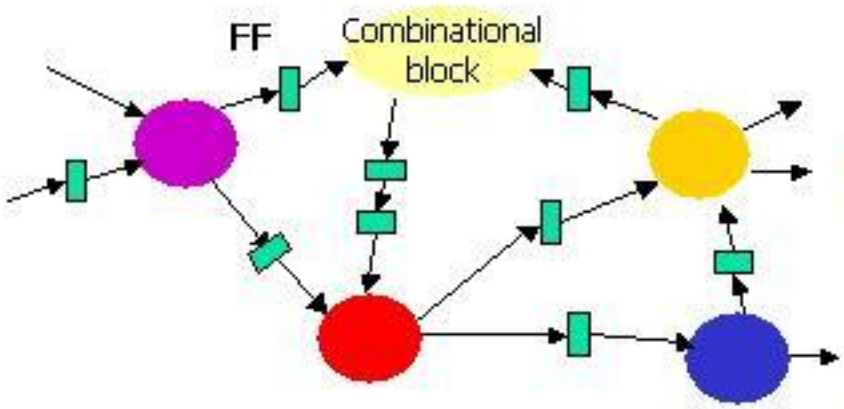

Assign delay upper bound for each net


- New Idea (Sequential budgeting):
  - Extend budgeting to handle sequential circuit better by incorporating [redacted] with it

# Delay Budgeting

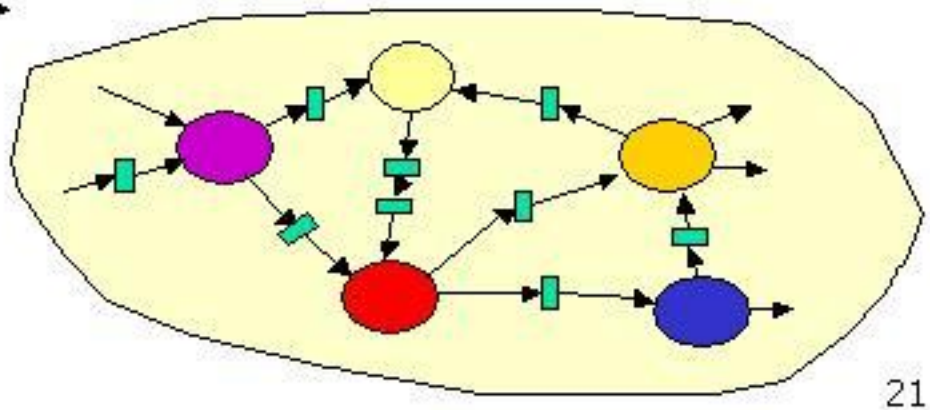
Previous work:

- Combinational budgeting
- Treat FFs like POs, PIs



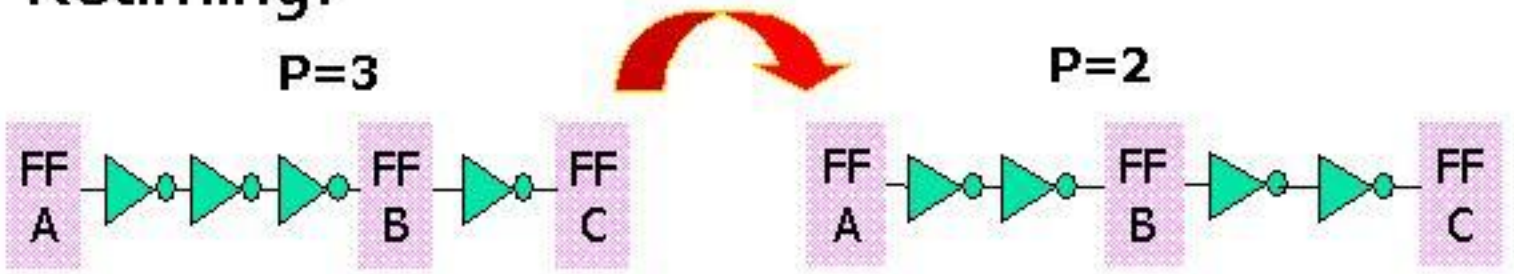

New idea: Sequential Budgeting

- Solve the circuit as a whole by combining budgeting with retiming
- Include clock period constraints



# Retiming (extend the solution space for budgeting)

- Retiming:

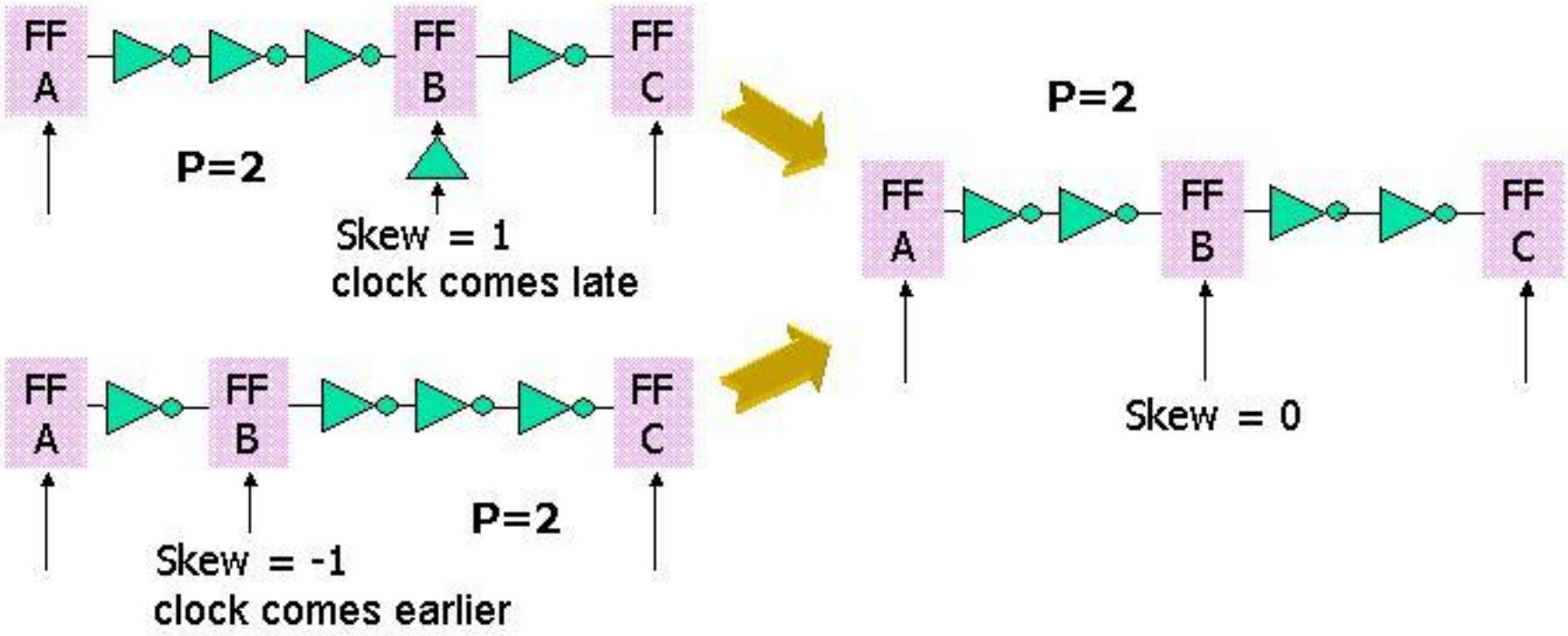



- 2 main optimization goals in retiming

1. Min-area retiming:
   - The cost function is the total # of FFs.
2. Min-period retiming:
   - Skew-based retiming:
     - Based on the analogy between clock skew and re-positioning FFs
     1) [redacted] to minimize clock period.
     2) [redacted] relation to realize skews.

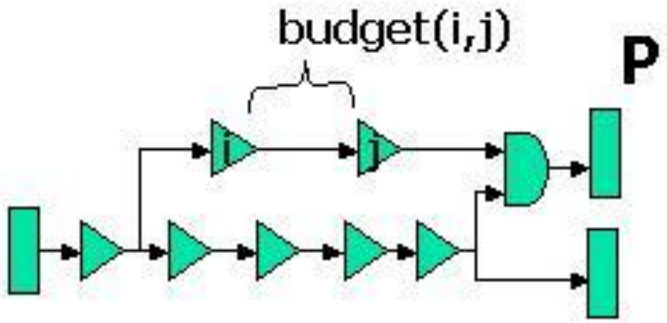
# Skew-retiming equivalence

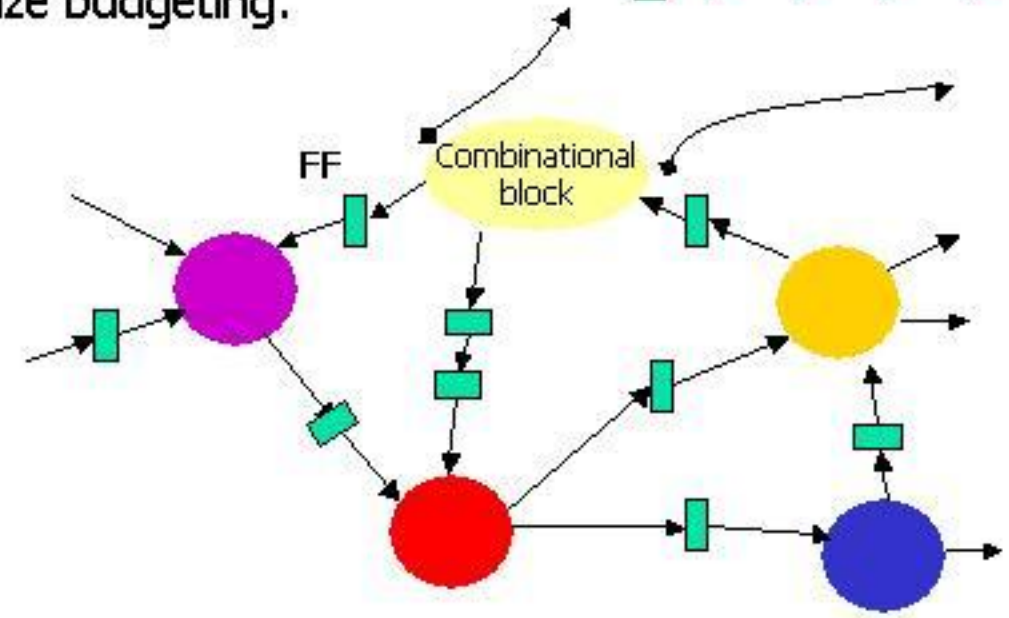
- Realize skews, after clock skew optimization
  - Positive skew => move backward
  - Negative skew => move forward



# Combinational Budgeting formulation

- Combinational budgeting
  - Transform Path constraints to edge constraints
  - Assign fan-in arrival time for each gate.
  - Optimize budgeting.



# Combinational Budgeting formulation (cont)

- Convex Delay Budgeting Problem (CDB):

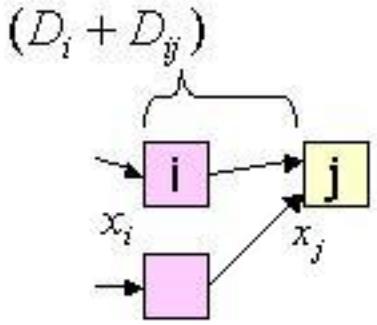
maximize:

Convex cost function

$$\sum_{e_{ij} \in E} C_{ij}[budget(i,j)]$$

subject to:

$$(x_j - x_i - D_i)$$

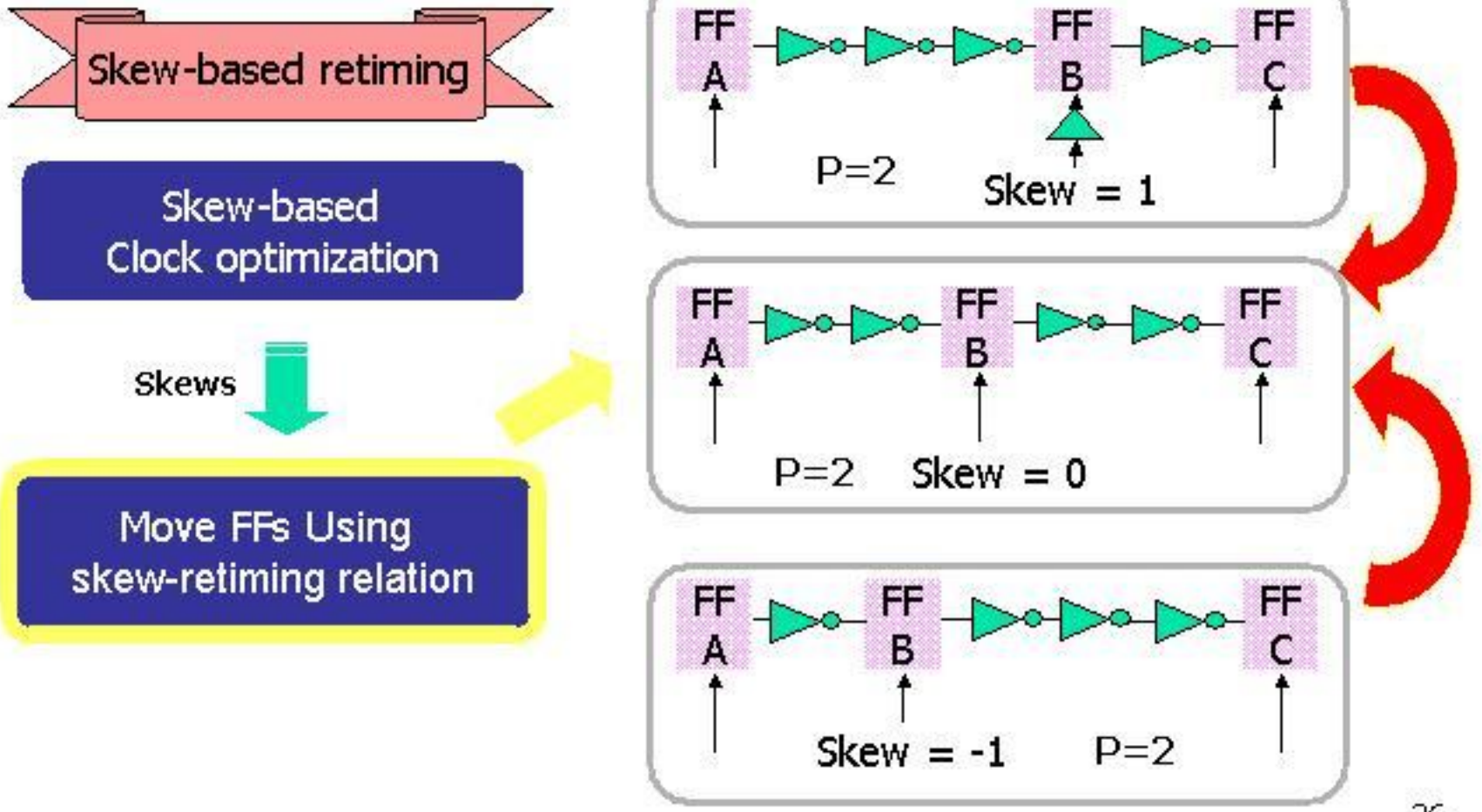



$$x_i + (D_i + D_{ij}) \le x_j \qquad \forall e_{ij} \in E$$

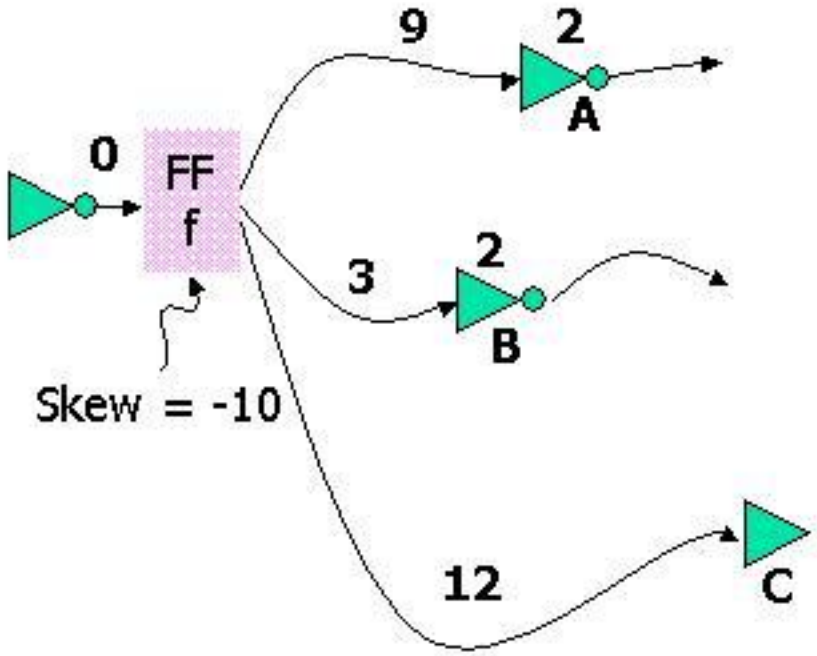
$$x_k \le P, \forall k \in PO; x_k = 0, \forall k \in PI$$

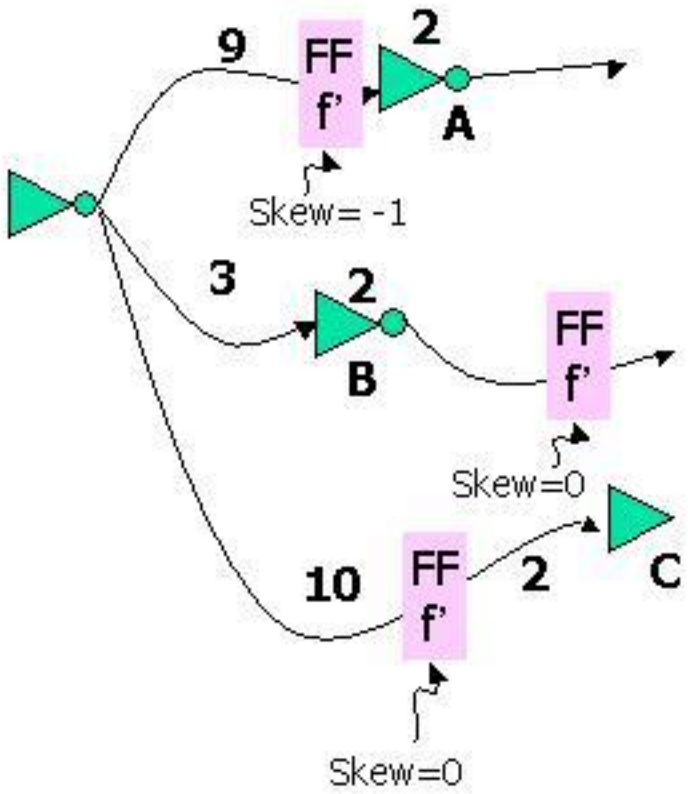
**NOTE: Doesn't consider FFs**

# Skew-based retiming



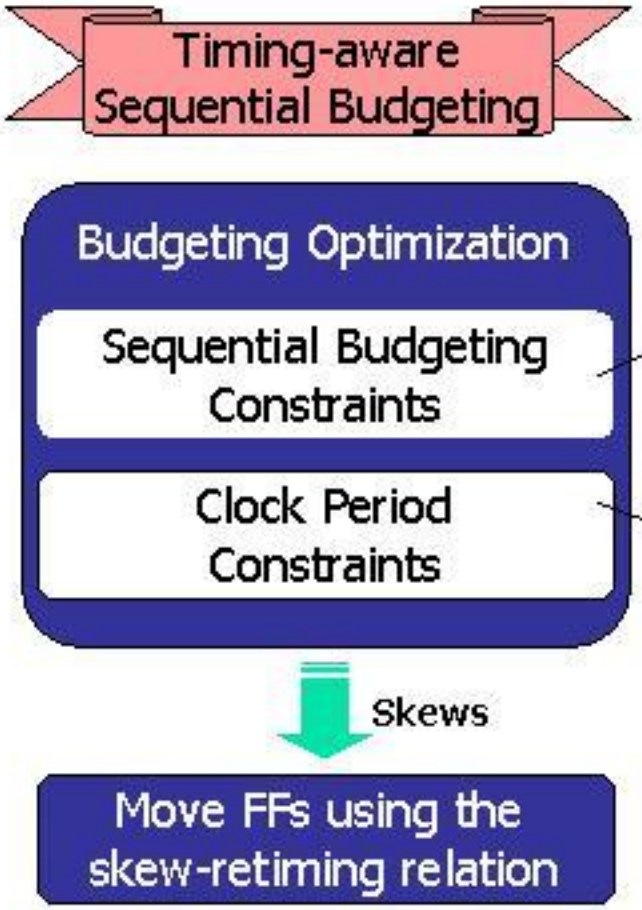
# Moving FFs according to skews (multi-fanout)




After FF movement, assign skew to 0.

# Timing-aware Sequential Budgeting




Sequential Budgeting constraints:

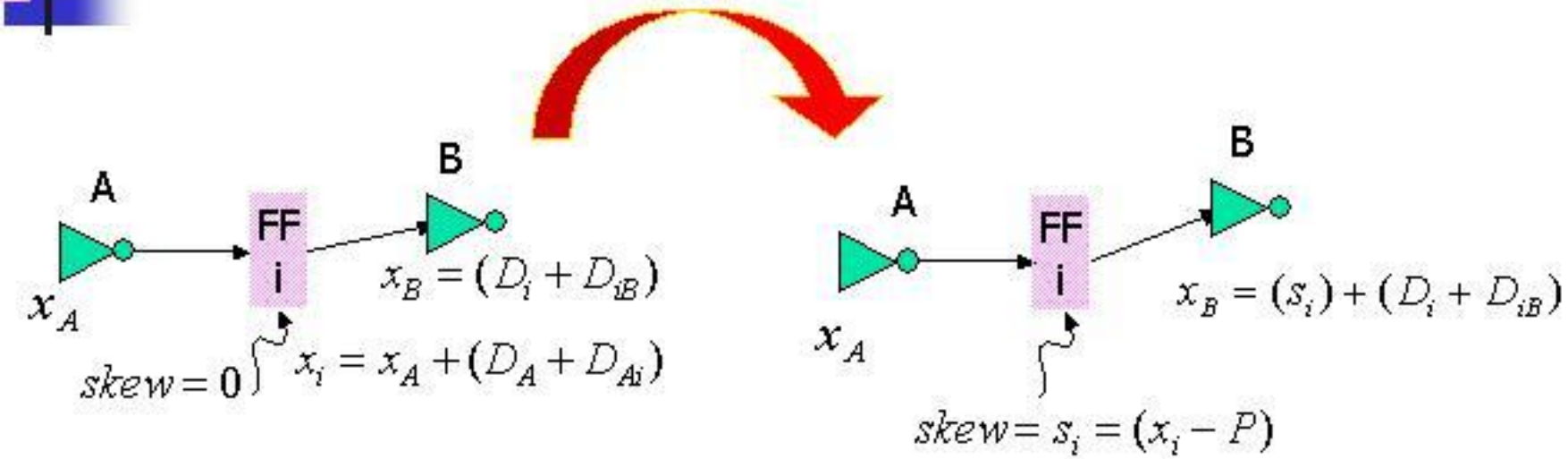
- Transform Path constraints to edge constraints
- Assign fan-in arrival time to each gate
- Add FFs into consideration

Clock period constraints:

- Budgeting doesn't violate timing
- Give larger budget for paths predicted long

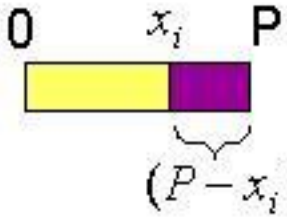
**Find the best budgeting that meet timing requirement**

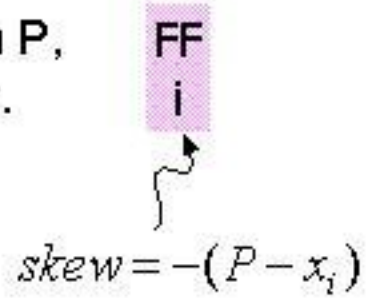
# Adding FFs in Combinational budgeting formulation




- Consider 2 cases for edge (i,j):
  - i is a FF – fan-in arrival time $(x_i - P)$
  - i is not a FF – fan-in arrival time $(x_i)$

If $x_i$ is smaller than P, clock comes earlier.



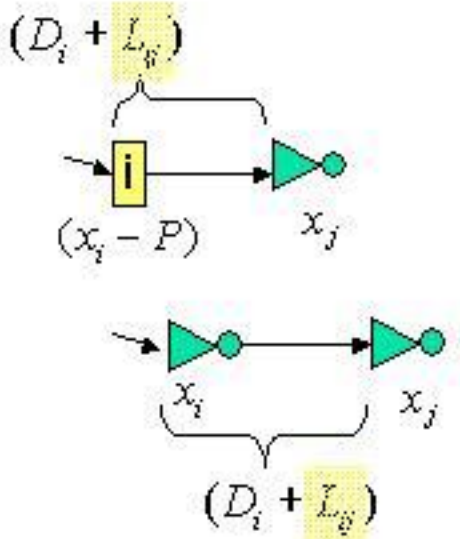



# Sequential Budgeting constraints

- Sequential Circuit Convex Delay Budgeting Problem (S-CDB):

  minimize: $\sum_{e_{ij} \in E} C_{ij}\left[budget(i,j)\right]$

  subject to:




$$(x_i - P) + (D_i + L_{ij}) \le x_j \qquad \forall e_{ij} \in E, i \in FF$$

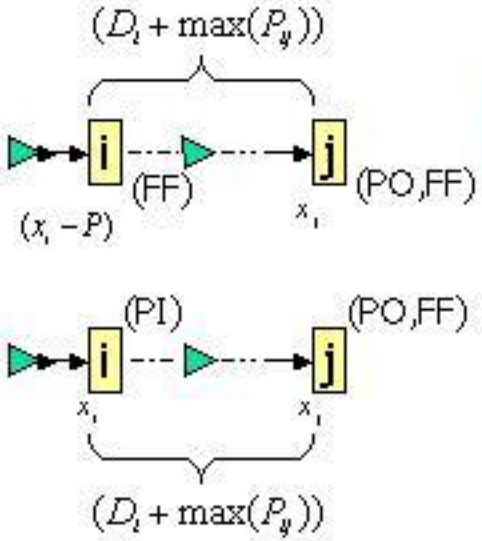
$$x_i + (D_i + L_{ij}) \le x_j \qquad \forall e_{ij} \in E, i \notin FF$$

$$x_k \le P, \forall k \in PO, x_k = 0, \forall k \in PI$$

# Clock Period constraints

- Make sure Budgeting doesn't violate timing
- Give larger budget for paths predicted long

Constraints:




$$(x_i - P) + (D_i + \max(P_{ij})) \le x_j \qquad \forall P_{ij} \in PS, i \in FF$$

Doesn't enumerate all paths

$$x_i + (D_i + \max(P_{ij})) \le x_j \qquad \forall P_{ij} \in PS, i \notin FF$$

$$x_k \le P, \forall k \in PO; x_k = 0, \forall k \in PI$$

# Timing-aware sequential budgeting formulation

- **T-SBGT**:

minimize:

$$\sum_{e_{ij} \in E} C_{ij}[budget(i,j)] \rightarrow \sum_{e_{ij} \in E} D_{ij} \cdot budget(i,j)$$

subject to:

skew of FF i

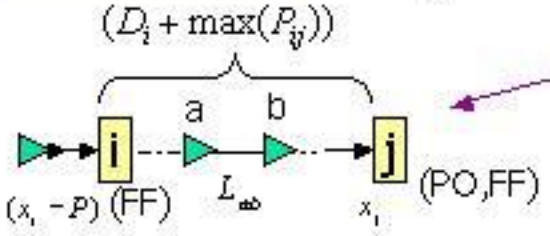
Clock period constraints

$$\begin{cases} (x_i - P) + (D_i + \max(P_{ij})) \le x_j & \forall P_{ij} \in PS, i \in FF \\ x_i + (D_i + \max(P_{ij})) \le x_j & \forall P_{ij} \in PS, i \notin FF \end{cases}$$

Budgeting constraints

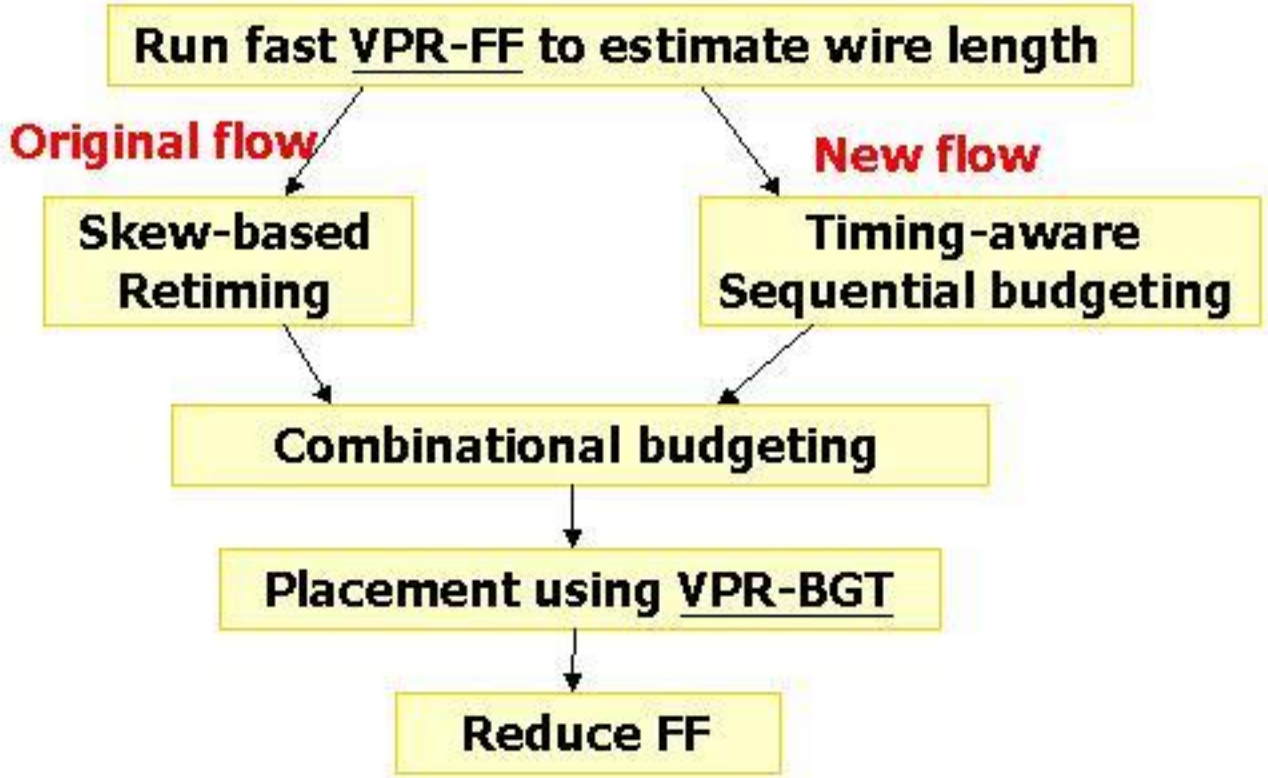
$$\begin{cases} (x_i - P) + (D_i + L_{ij}) \le x_j & \forall e_{ij} \in E, i \in FF \\ x_i + (D_i + L_{ij}) \le x_j & \forall e_{ij} \in E, i \notin FF \end{cases}$$

$$L\max(P_{ij}) \le \max(P_{ij}) \quad \forall P_{ij} \in PS$$



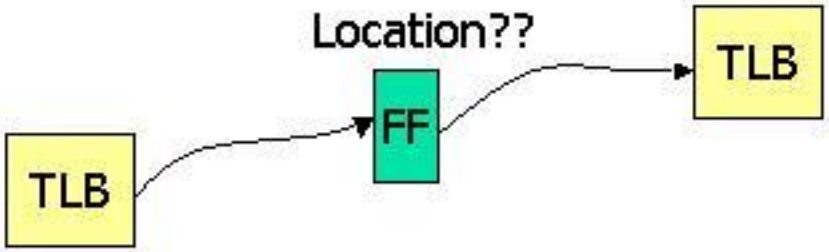

$$x_k \le P, \forall k \in PO, x_k = 0, \forall k \in PI$$

# Application in FPGA placement



# Modified VPR Placer

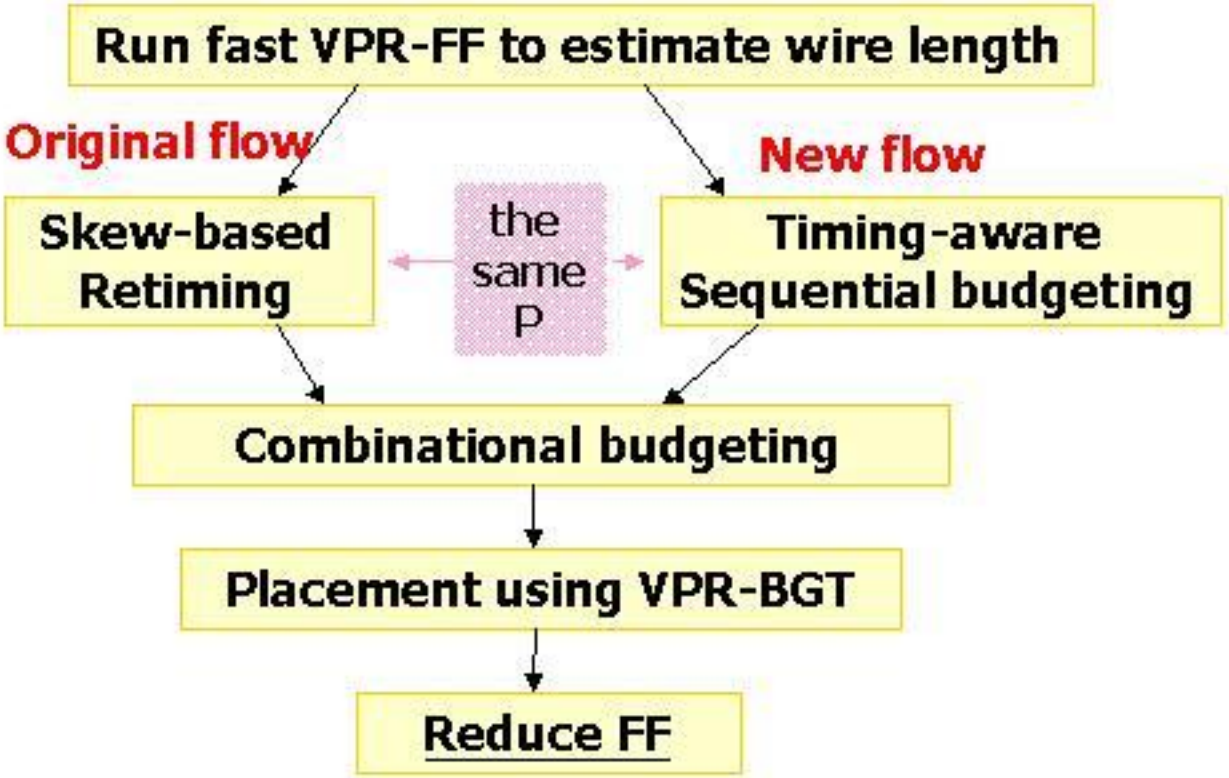
- VPR – a simulated-annealing based FPGA placer
- VPR-FF – Decouple FFs and TLBs in placement.



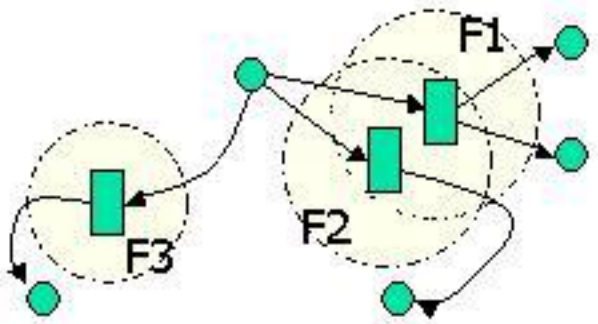

- VPR-BGT – Budgeting-aware Placer
  - Penalize edges whose delay are larger than their budgets

$$Bcost_{ij} = \begin{cases} 1000 \times (D_{ij} - B_{ij})^{1.5} & if\ (D_{ij} > B_{ij}) \\ 0.003 \times (D_{ij} - B_{ij}) & else \end{cases}$$

# Application in FPGA placement

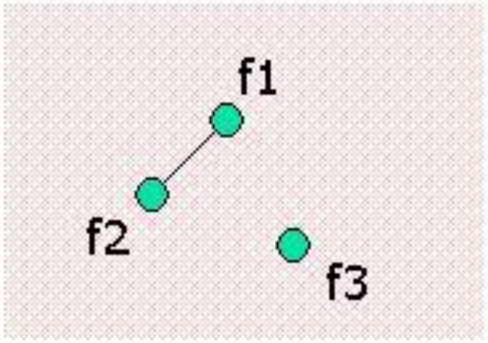


# Post-layout FF reduction (reduce WL, power)

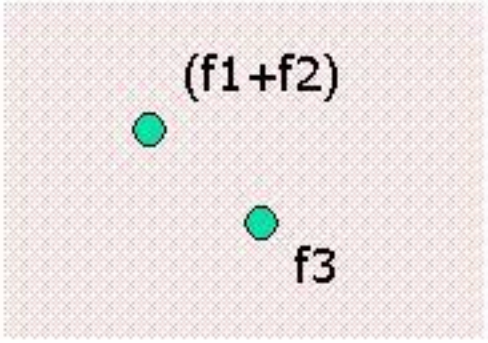



- Create an edge (i,j):
  - i and j are fan-outs of the same gate
  - The movable region of i and j intersect
- A Clique:
  - A group of nodes, every pair of them are connected by an edge.
- Maximum clique:
  - The clique with most nodes.

Iteration I:




Iteration II:



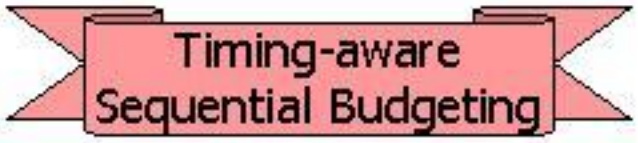

Find the max-clique until no edge in the graph

# Experimental Results

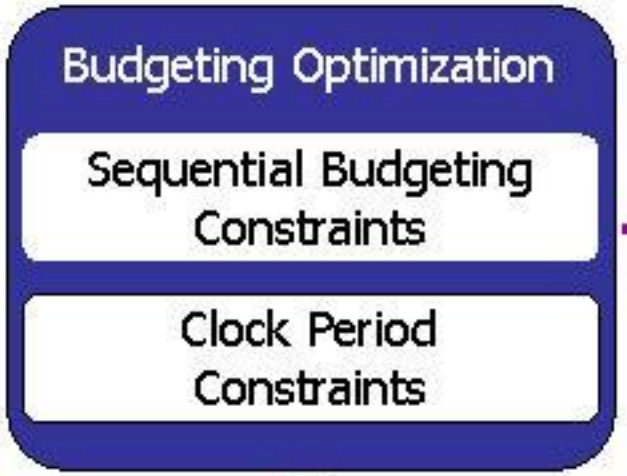
Percentage of nets violate budgeting

| # FFs after retiming | | Clock period | | Budget violation | | FF reduction (%) |
|---|---|---|---|---|---|---|
| New | Orig | New | Orig | New | Orig | New / Orig |
| 1 | 2.8 | 1 | 1.08 | 1 | 1.14 | 19 |

# Including interconnect prediction



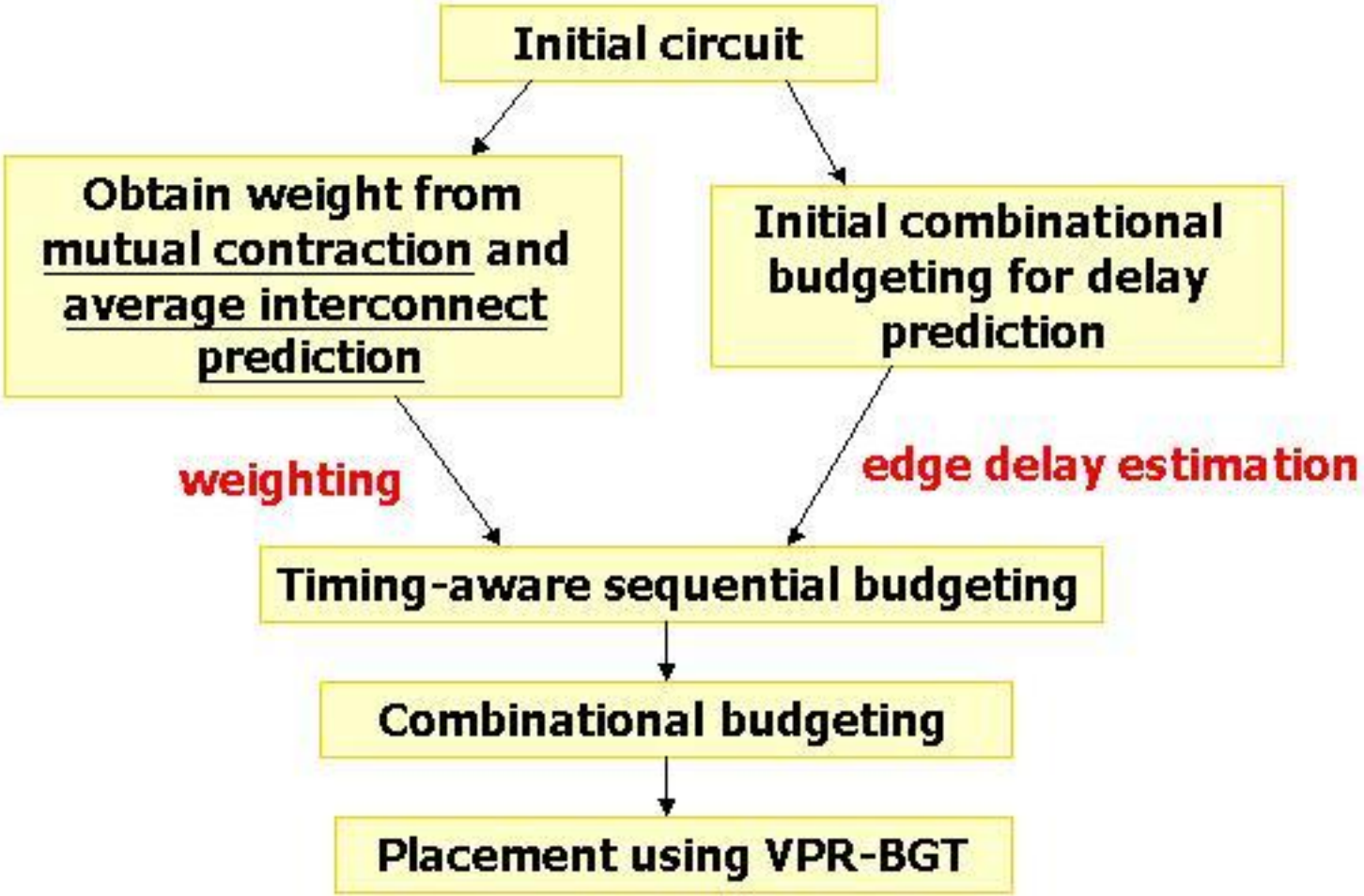

- Less dependent on previous placement




- Use interconnect prediction to generate weighting for each net
  - Net predicted to be longer => increase its budget

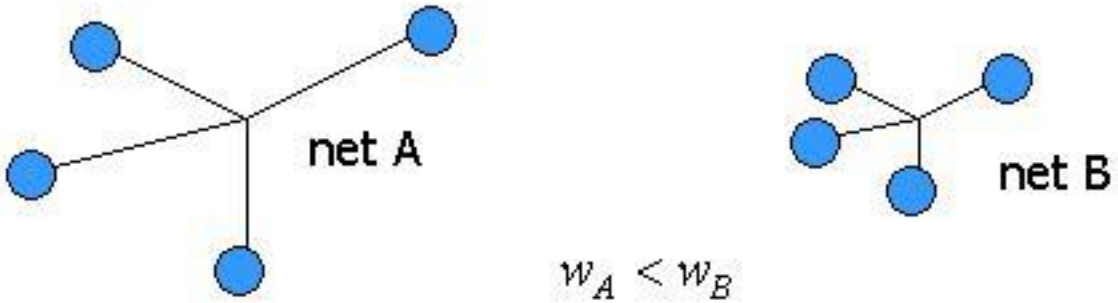
- Using combinational budgeting to predict interconnect delay

# Interconnect prediction with sequential budgeting



# Mutual contraction & Average Interconnect Lengths

- Mutual Contraction
  - Look at the neighborhood of a net and predict if this net will be long or short




$w_A < w_B$

- **Average Interconnect Lengths**
  - Predict average edge length for each partition level

$$L_k = \frac{4\left(\left(\frac{4\lambda}{3} - \frac{1}{3\lambda}\right) + 4\lambda\right)}{6}, \quad \lambda = 2^{H-k}$$

# Weighting for each net

- Assign weight for each net

$$\alpha_{ij} = \left( \frac{1 + e^{-3P_{NT}}}{2} \right)^{5/4} \times (2 - e^{-L_N})^5 \;, e_{ij} \in net\ N, N\ is\ at\ level\ k$$

Mutual Contraction

Average Interconnect length

- Budgeting cost function

$$\sum_{e_{ij} \in E} \alpha_{ij} \cdot \log[budget(i, j)]$$

# Experimental Results

| *Circuit* | *VPR-FF* | *New Placement Flow* | *Uniform Weighting* | *Retiming with VPR-FF* |
|---|---|---|---|---|
| | P (ns) | P (ns) | P (ns) | P (ns) |
| bigkey | 7.75 | 5.68 | 5.60 | 7.54 |
| elliptic | 16.57 | 15.58 | 15.42 | 16.58 |
| s298 | 19.88 | 19.93 | 20.08 | 20.6 |
| s38417 | 12.70 | 11.28 | 11.39 | 12.70 |
| diffeq | 13.29 | 12.68 | 12.57 | 13.03 |
| tseng | 12.15 | 11.41 | 12.67 | 11.77 |
| dsip | 8.47 | 5.62 | 5.20 | 7.09 |
| clma | 24.96 | 22.38 | 24.67 | 24.95 |
| | 1 | 90.3% | 92.9% | 98% |

Different from previous experiment, use prediction here

# Conclusion

- Derive the sequential budgeting algorithm that optimizes delay budgeting
- Apply it in FPGA placement
- Incorporate interconnect prediction in sequential budgeting

- Bkup slides

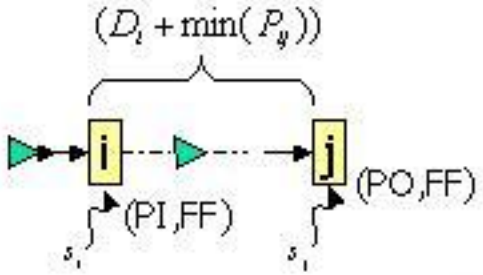
# Skew-based Retiming: clock optimization

- Skew-based Clock Optimization formulation (SCO):

  Minimize: $P$

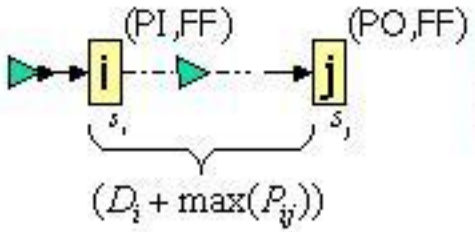
  Subject to:

Short path constraint




$$s_i + (D_i + \min(P_{ij})) - T_{hold} \geq s_j \quad \forall P_{ij} \in PS$$

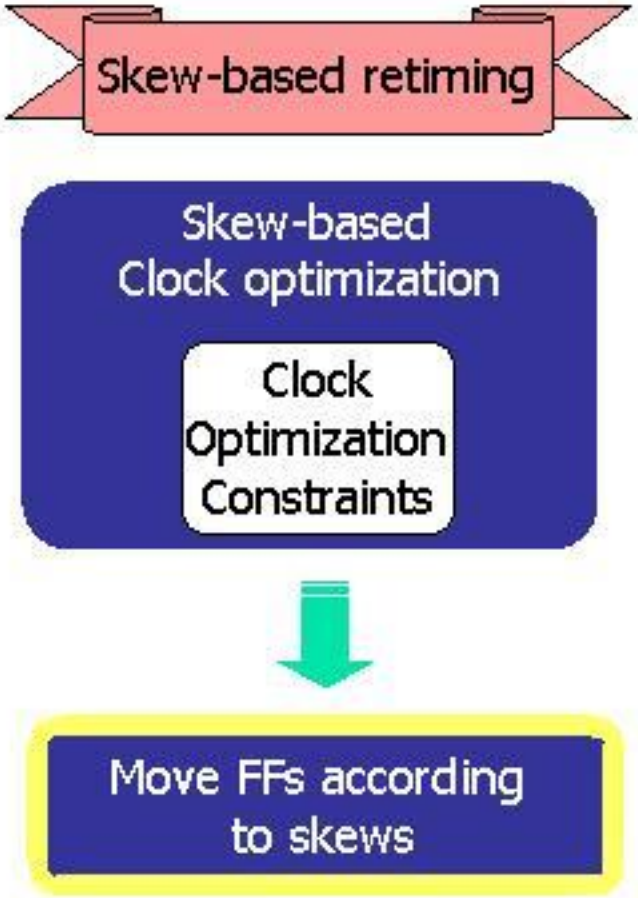
Long path constraint

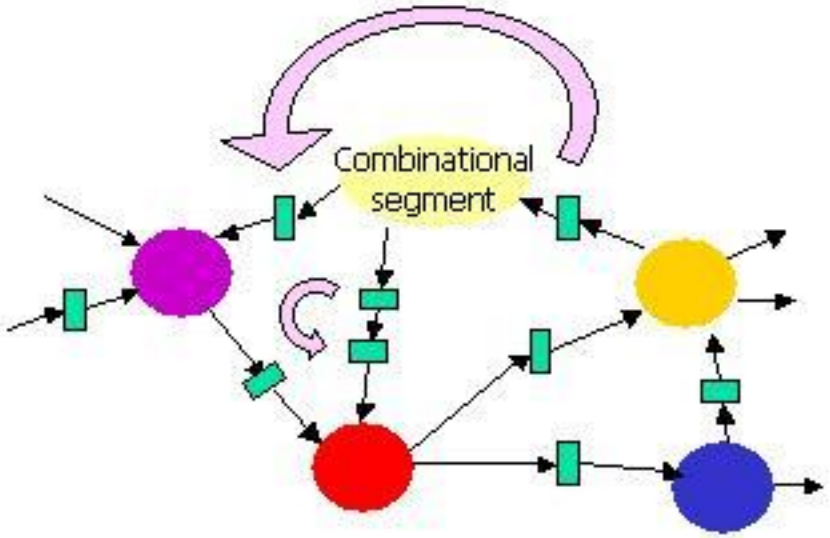



$$s_i + (D_i + \max(P_{ij})) + T_{setup} \leq s_j + P \quad \forall P_{ij} \in PS$$

$$s_k = 0, \forall k \in PO, PI$$

# Skew-based retiming




- Short path constraint
  - Prevent double clocking
- Long path constraint
  - Prevent violate timing

# Move FFs according to skews

- Skew-retiming equivalence