Pulser Gating: A Clock Gating of Pulsed-Latch Circuits

Sangmin Kim, Inhak Han,
Seungwhun Paik, and Youngsoo Shin
Dept. of EE, KAIST

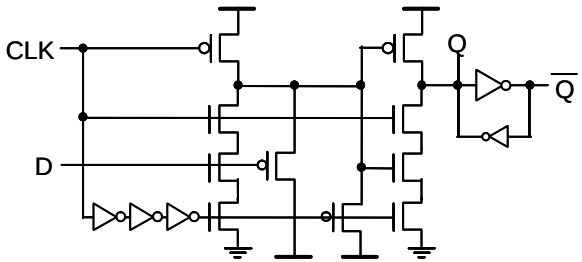
Contents

- Introduction
  - Pulsed-latch circuits
  - Clock gating synthesis
- Pulser gating synthesis
  - Problem definition
  - Synthesis algorithm
- Experiment results
- Summary

Pulsed-Latch Circuits

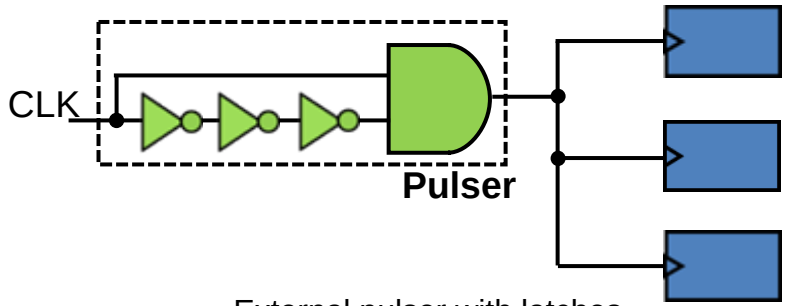
- Sequencing element
  - Edge triggered F/F: triggered at fixed instant, large overhead
  - Level-sensitive latch: triggered at varying instant, small overhead

- **Pulsed-latch**
  - Latch driven by a brief pulse (generated by pulse generator)
  - Combine the advantage of F/F and latch



Integrated pulser and latch [ISSCC96]

Robust design,
but many generators



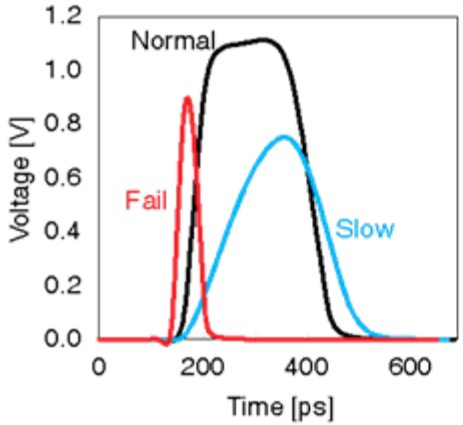
External pulser with latches

Shared pulsers,
but careful design

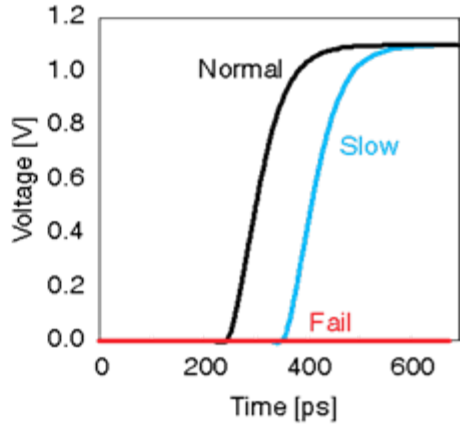
Pulsed-Latch Circuits

- Simple migration from F/F circuits to pulsed-latch circuits gives ~10% performance improvement and ~20% power saving

- **Design challenges**
  - Integrity of pulse shape (pulse can be easily degraded)
  - Inserting pulsers
  - Automatic placement
  - Increasing hold violations
  - **Applying clock gating**
  - Power network noise



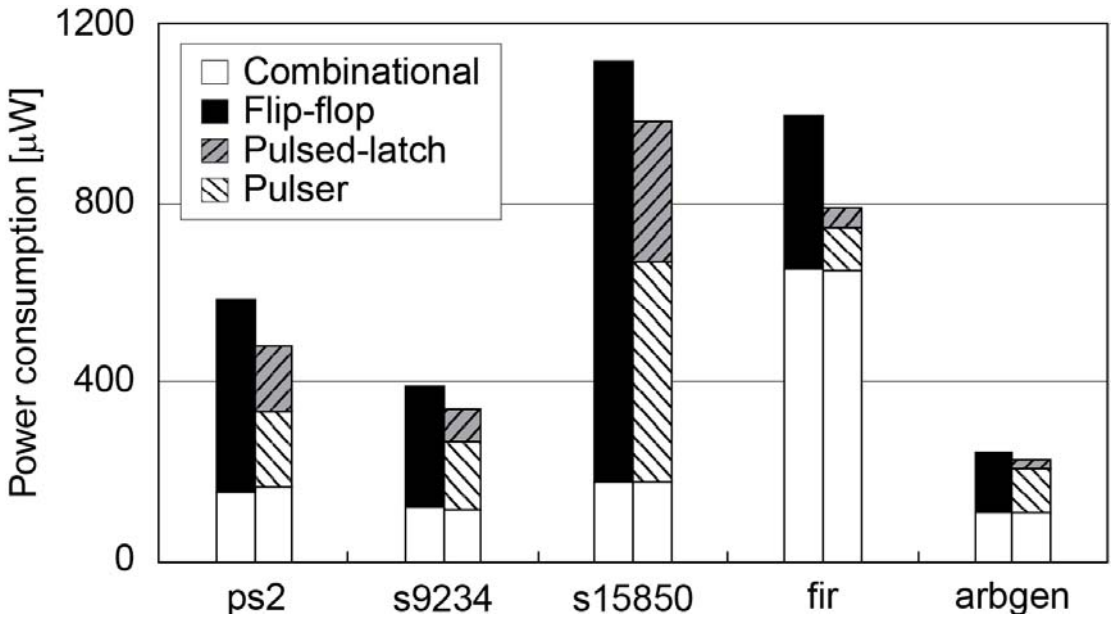
Latch input pulse



Latch output

Pulsed-Latch Circuits

- Power consumption of F/F vs. pulsed-latch circuits
  - Pulsed-latch has less power by **14%**
  - Pulser consumes 37% of power → gating is important



Clock Gating Synthesis - Review

0. Given a gate-level netlist

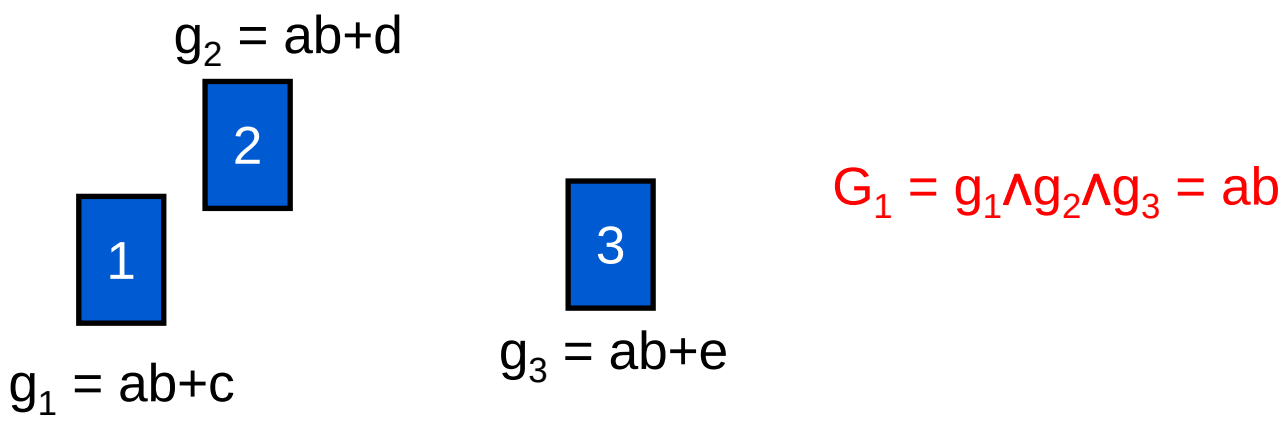
1. Derive a gating function of each latch i:

$g_i = \overline{\delta_i \oplus S_i}$ (δ_i: latch input, S_i: latch output)

–Gating probability: $P(g_i) = Prob(g_i = 1)$

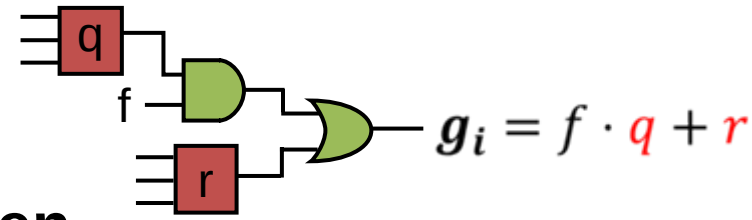
2. Merge gating functions that are "similar"

–To reduce extra logic to implement gating functions

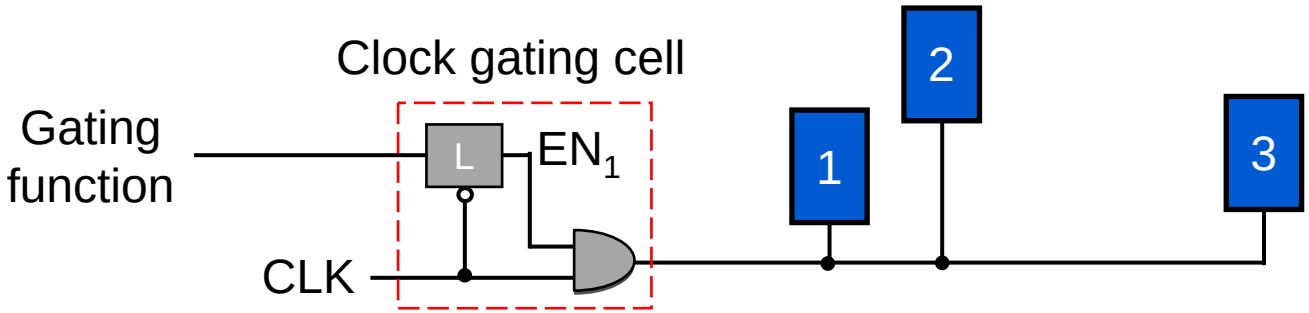


Clock Gating Synthesis - Review

3. **Simplify gating functions**
   - Division: use existing combinational logic (f), implement only quotient (q) and remainder (r)
   - Approximation: remove some terms (with low probability) from gating function



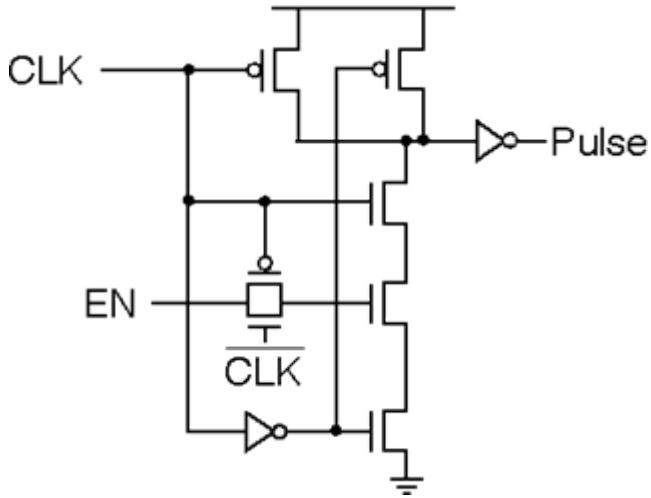
4. **Insert clock gating cell between gating function and latches**



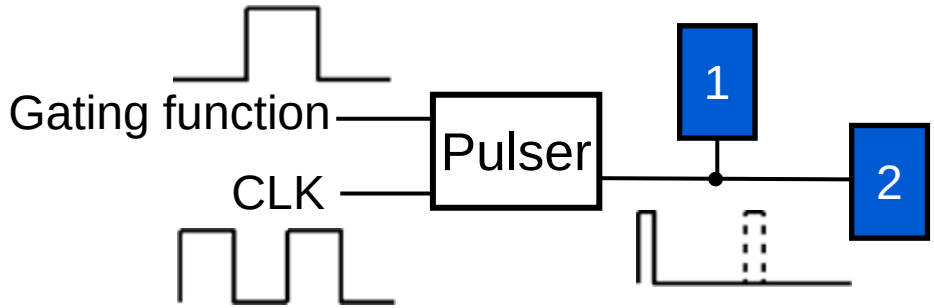
Pulser Gating

- **Pulser gating**: clock gating using pulser (pulse generator)
  - Clock gating can be conveniently integrated into a pulser
  - A pulser can drive a few nearby latches
  - A pulser gates/ungates its latches together

How to group latches becomes very important

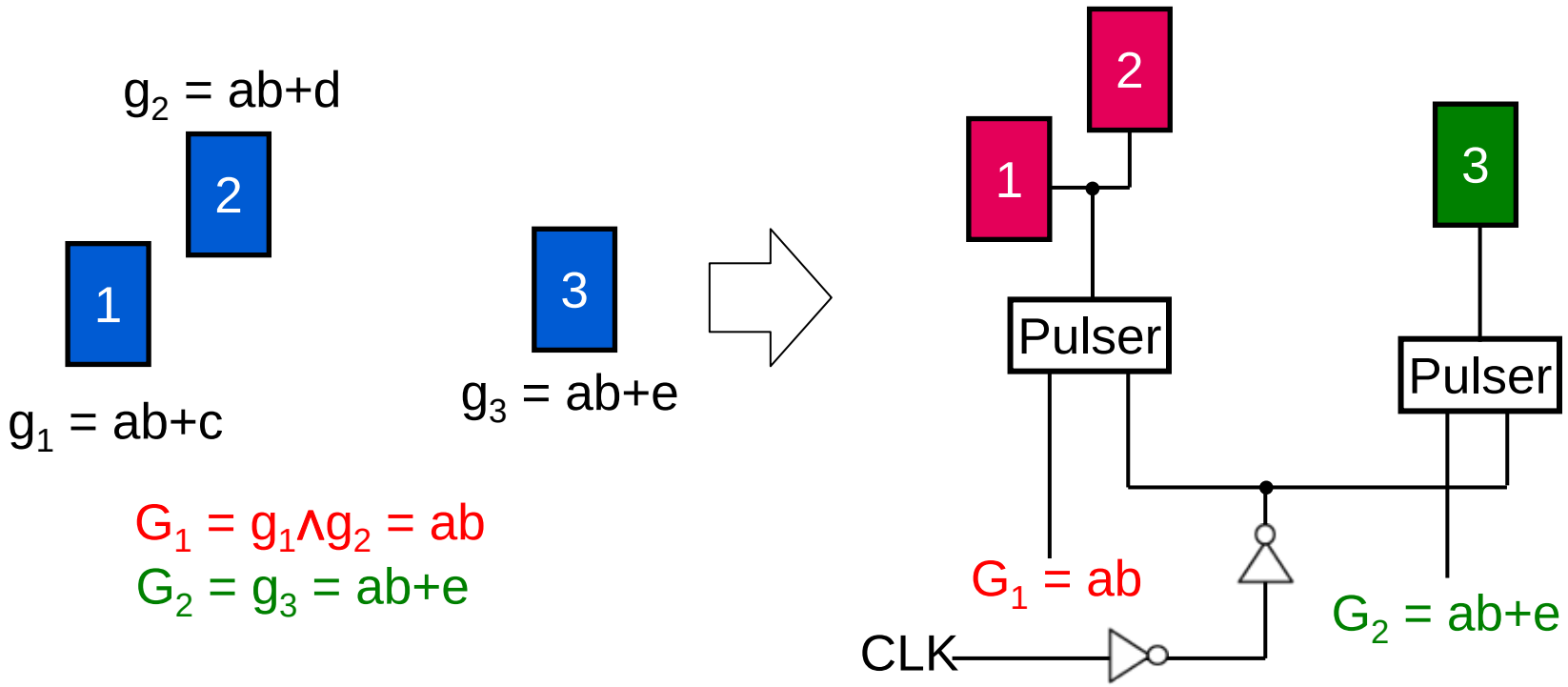


[Naffziger, JSSC02]



Pulser Gating Synthesis

- **Key problem**: merge gating functions (of latches) that are **functionally similar** & **physically close**



Pulser Gating Synthesis

- Problem definition
  - Given: **gating function** and **location** of each latch (after initial placement)
  - Create a set of clusters $\{\mathbb{C}_1, \mathbb{C}_2, \cdots, \mathbb{C}_i\}$; cluster $\leftrightarrow$ a set of latches and a pulser
  - **Objective**
    - Maximize average gating probability & minimize total literal count of gating functions
  - **Constraint**
    - Each pulser must drive load capacitance smaller than C_{max}

      $\boldsymbol{C_{load}(\mathbb{C}_i) \leq C_{max}}$

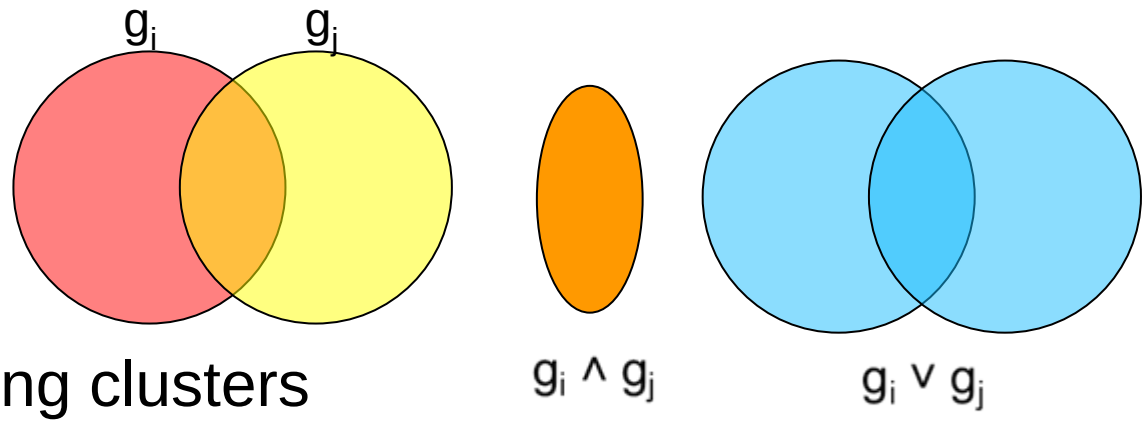
Pulser Gating Synthesis

1. Compute **similarity** of gating functions
2. Create a **similarity graph**
3. Perform **clustering**
4. **Assess** clusters
5. **Pulser insertion** for remaining latches

1: Compute Similarity of Gating Functions

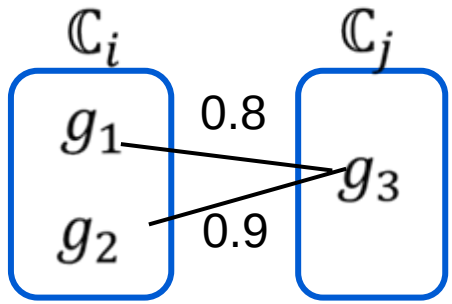
- Functional similarity of two gating functions g_i and g_j
  - Percentage of intersection

$$S(g_i, g_j) \triangleq \frac{P(g_i \wedge g_j)}{P(g_i \vee g_j)}$$



- Similarity of two gating clusters
  - **Average similarity** of gating functions in clusters

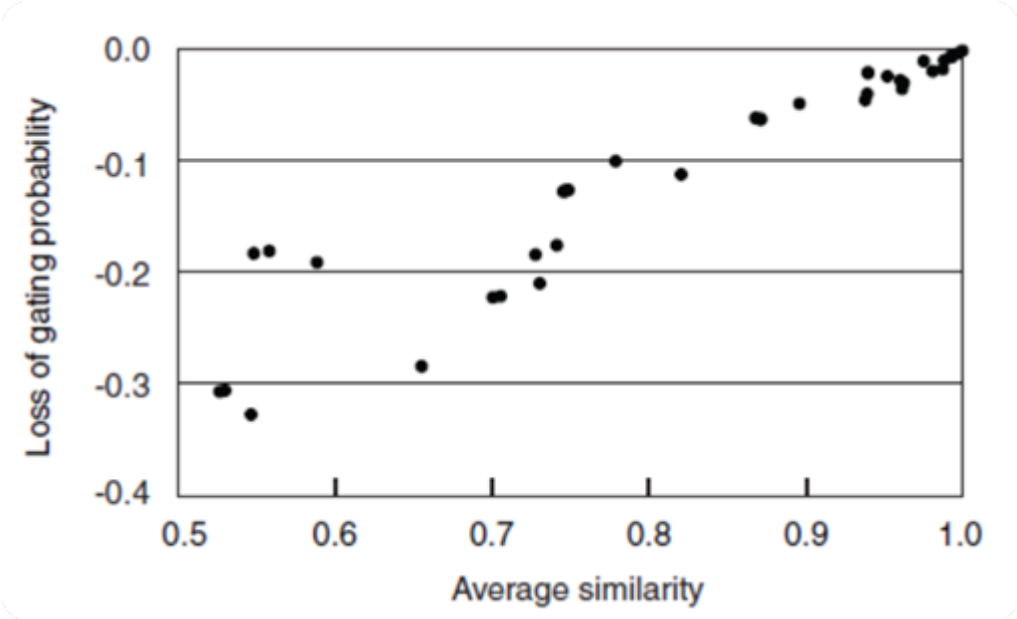
$$S(G_i, G_j) \triangleq \underset{x \in \mathbb{C}_i, y \in \mathbb{C}_j}{\text{AVG}}\, S(g_x, g_y)$$



$$S(G_i, G_j) = 0.85$$

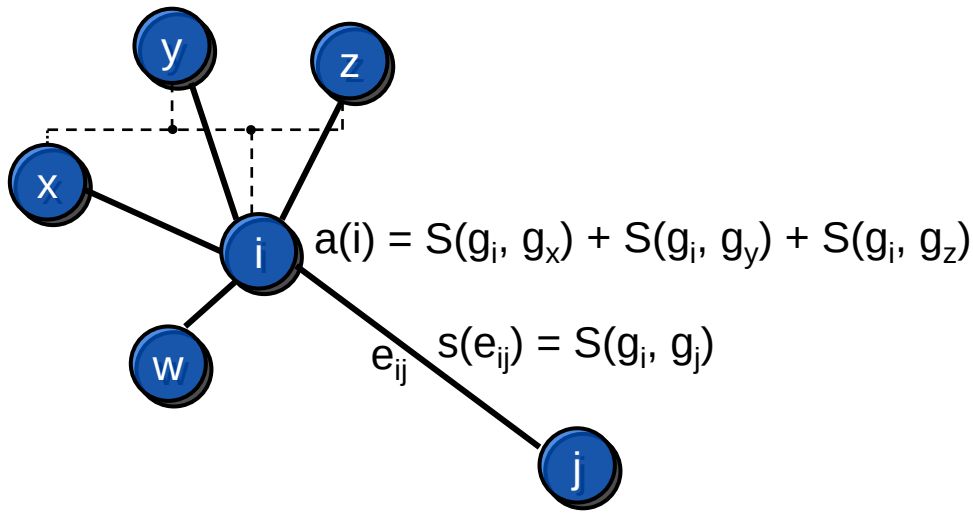
1: Compute Similarity of Gating Functions

- $\mathrm{AVG}\, S(g_x, g_y)$ vs. average loss of gating probability
  - High $\mathrm{AVG}\, S(g_x, g_y)$ should imply less loss of gating probability
  - Acceptable correlation shown through experiment



2: Create a Similarity Graph

- Similarity graph: G(V, E, a, s)
  - Vertex: a cluster of latches
  - Edge: if merging two vertices is potentially possible
  - Edge weight s: similarity between clusters
  - **Vertex weight a(i)**: sum of similarity at vertex i
    - Sort edges in descending order of similarity
    - Only add similarities one by one while $C_{load}(\mathbb{C}) \leq C_{max}$



3: Perform Clustering

- **Measure of merit**: value used to represent **quality of a cluster**
  - Consider both average gating probability and literal count of gating function

$$\mathcal{M} \triangleq \sum_i \left(|\mathbb{C}_i| \underset{\substack{x,y \in \mathbb{C}_i \\ x \neq y}}{\text{AVG}} S(g_x, g_y) - \alpha L(G_i) \right)$$

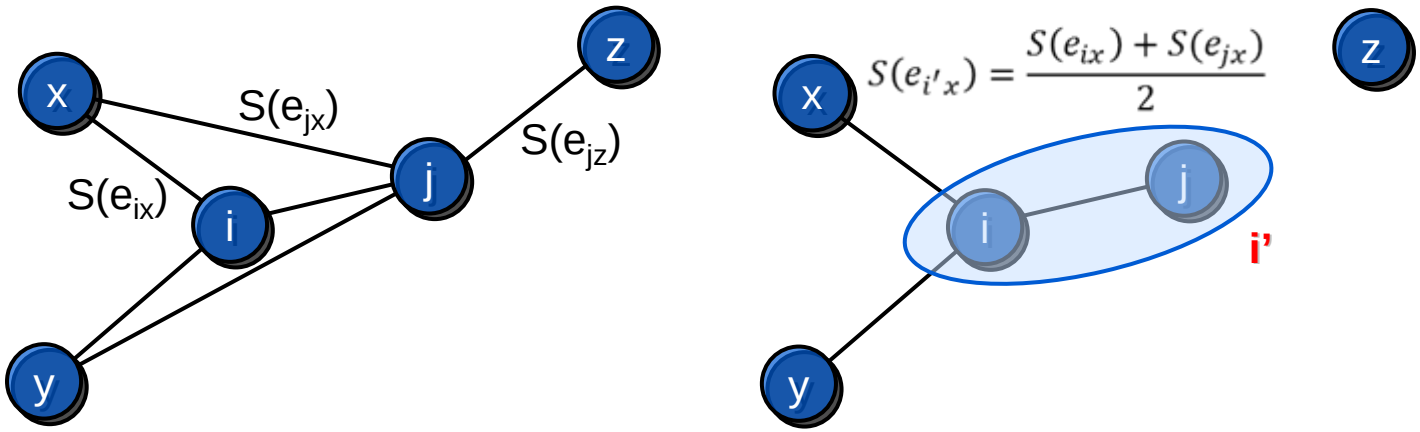
Size of cluster

Average similarity of gating functions in cluster

Literal count of gating function

3: Perform Clustering

- While E≠∅ in similarity graph
  - Select vertex i having **maximum vertex weight a(i)**
  - Select vertex j having maximum edge weight by choosing edge with **maximum similarity s(e_{ij})**
  - If measure of merit improves and $C_{load}(\mathbb{C}_{i'})$ is smaller than C_{max} , **merge i and j**
  - Else, remove edge



4: Assessment of Clusters

- Synthesize a clock gating function

- Net saving =
  (power saving – power consumption of extra gates)

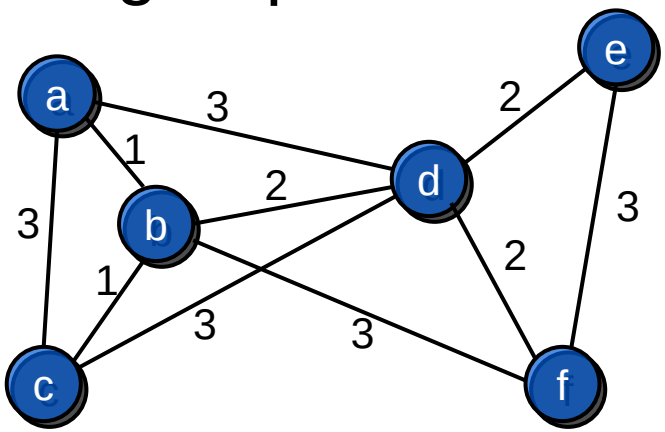
$P(G_i)\times$ (power consumption of pulser and latches)

- Pulser gating is performed **only on clusters which have positive net saving**

5: Pulser Insertion for Remaining Latches

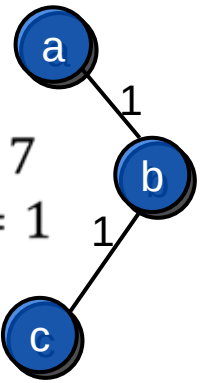
- Create graph G(V,E)
  - V: a set of latches that are not pulser gated
  - E: a set of edges (each edge is between latches that can share a pulser), **edge weight is Manhattan distance** between two latches

- Find groups of latches that **satisfy C_{max}**

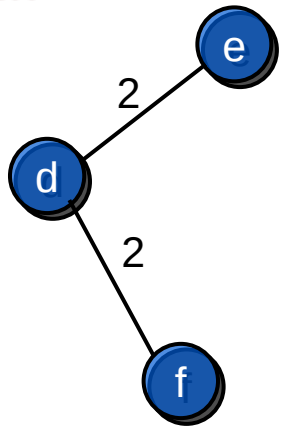


$C_{max} = 7$

$C_{latch} = 1$

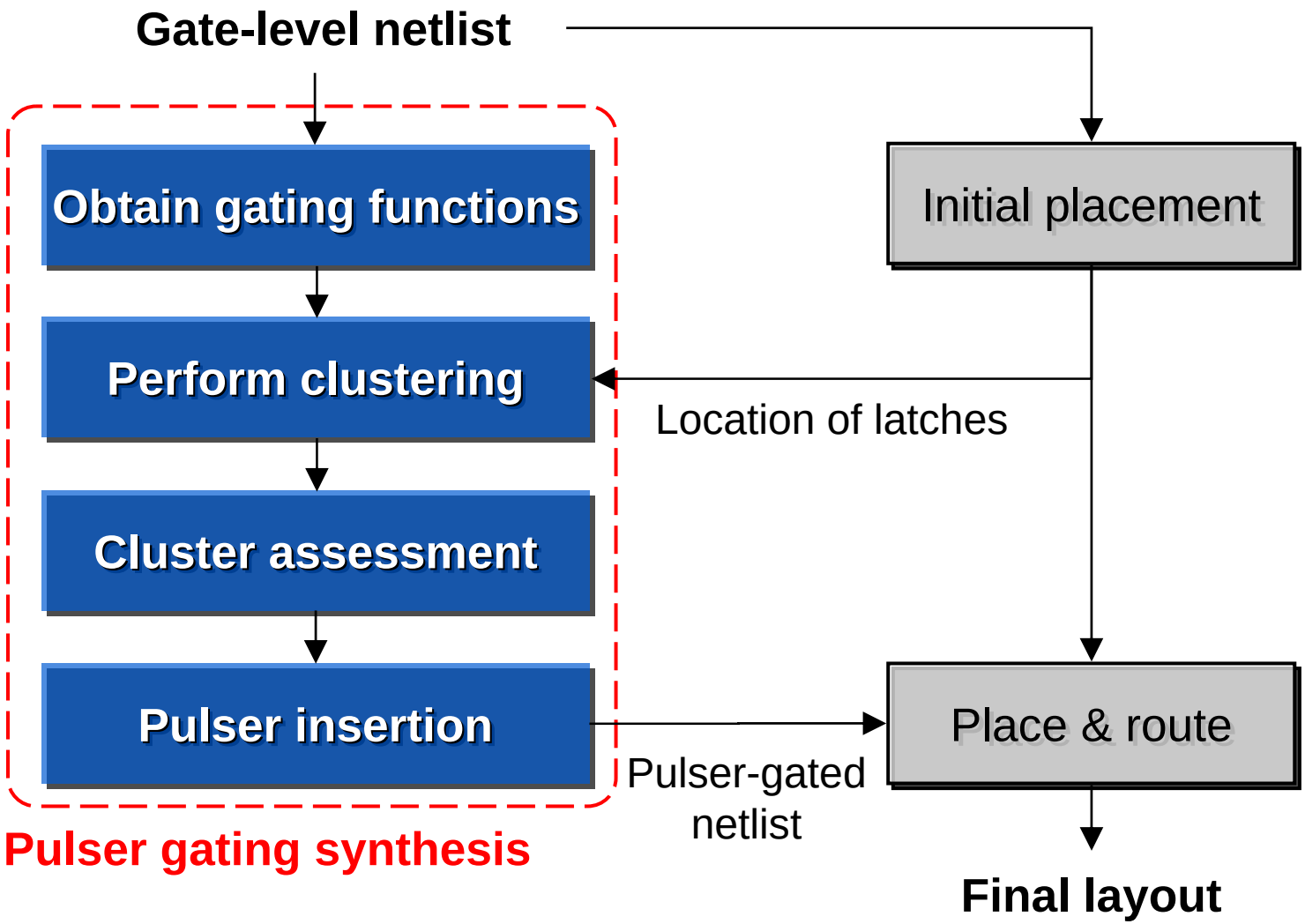


$C_L = 3 \cdot 1 + 2 = 5$



$C_L = 3 \cdot 1 + 4 = 7$

Pulser Gating Flow



Experiments

- Experimental setup
  - 45-nm commercial technology
  - Test circuits from ISCAS, ITC, and OpenCores
  - Pulser gating synthesis was implemented in SIS
- Results compared with pulsed-latch circuits without pulser gating

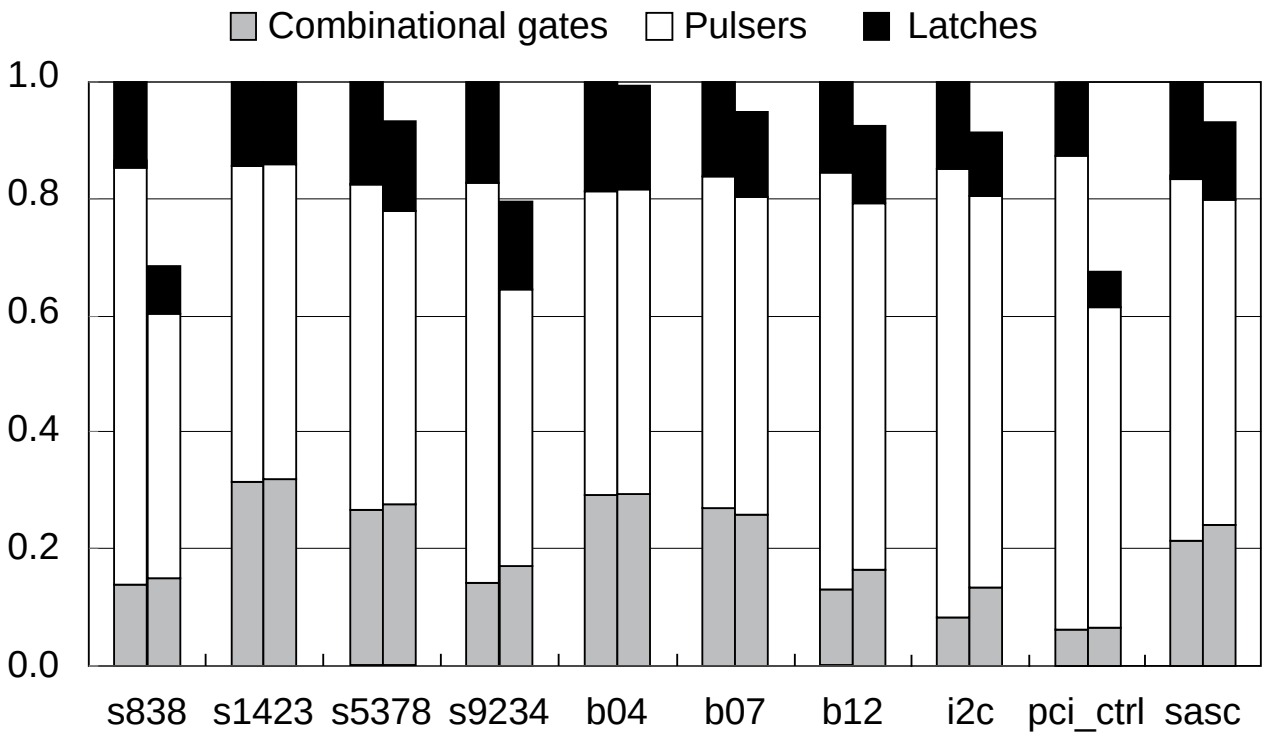
Experiments

- Synthesis results
  - # of pulsers increase to maintain high $P(G_i)$
  - **Area** increases by **11%** due to extra gates

Name	# Gates	# Latches	# Pulsers	*Pulser_Gating_Synthesis*				
				Δ Pulsers	# Extra gates	% Gated latches	AVG $P(G_i)$	Runtime (min.)
s838	351	32	6	2	0	65.6	0.63	0.3
s1423	1191	74	15	1	24	4.1	0.04	1.3
s5378	1781	160	30	5	417	23.8	0.22	16.6
s9234	1293	125	22	7	276	28.0	0.23	6.8
b04	833	66	13	1	17	7.6	0.07	2.3
b07	431	44	8	1	31	20.5	0.18	0.3
b12	1395	119	21	7	317	31.1	0.29	3.1
i2c	1125	128	22	6	475	39.8	0.30	4.9
pci_ctrl	879	60	12	3	90	51.7	0.51	1.3
sasc	1058	116	21	5	179	30.2	0.26	2.5

Experiments

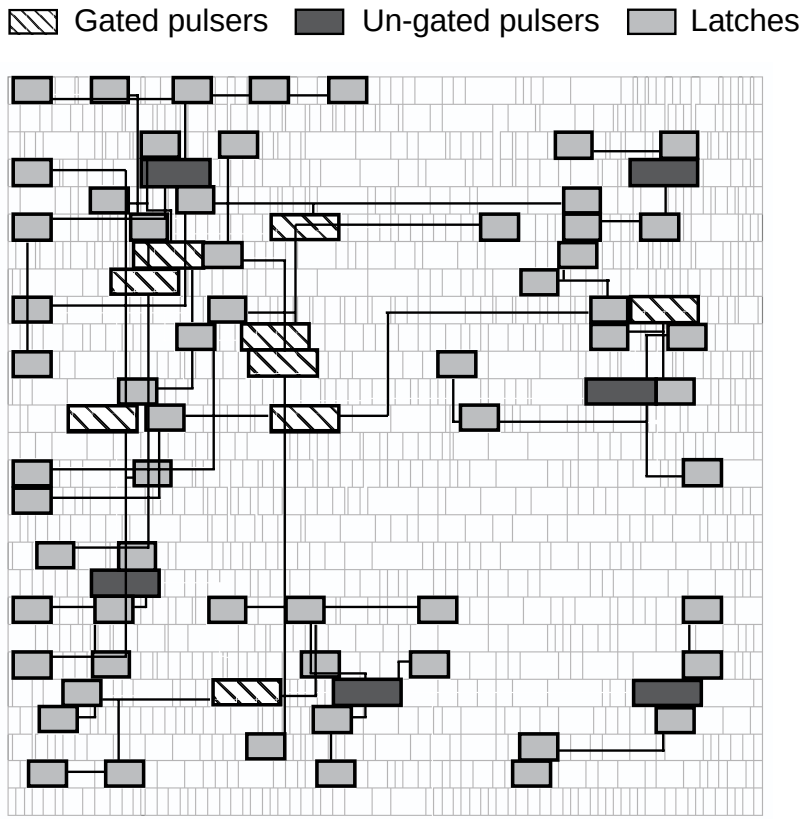
- Power saving of **12% on average**



Combinational: 13%
Pulsers: -20%
Latches: -15%

Experiments

- Test circuit **pci_ctrl**
  - 9 gated pulsers, 6 un-gated pulsers
  - Gated pulsers drive **smaller # of latches** (long distance between some latches with high similarity)



Summary

- Pulsed-latch
  - Retain the advantage of both flip-flop (simple timing model) and latch (fast and small)
  - External pulser: pulser and latches should be geographically close
- Pulser gating synthesis
  - When merging two gating functions, their “similarity” and “location of latches” have to be considered
- Experiments
  - Power savings of 12% compared to pulsed-latch circuits without gating