Power-Driven Flip-Flop Merging and Relocation

Shao-Huan Wang
Yu-Yi Liang
Tien-Yu Kuo
Wai-Kei Mak
@National Tsing Hua University

Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions

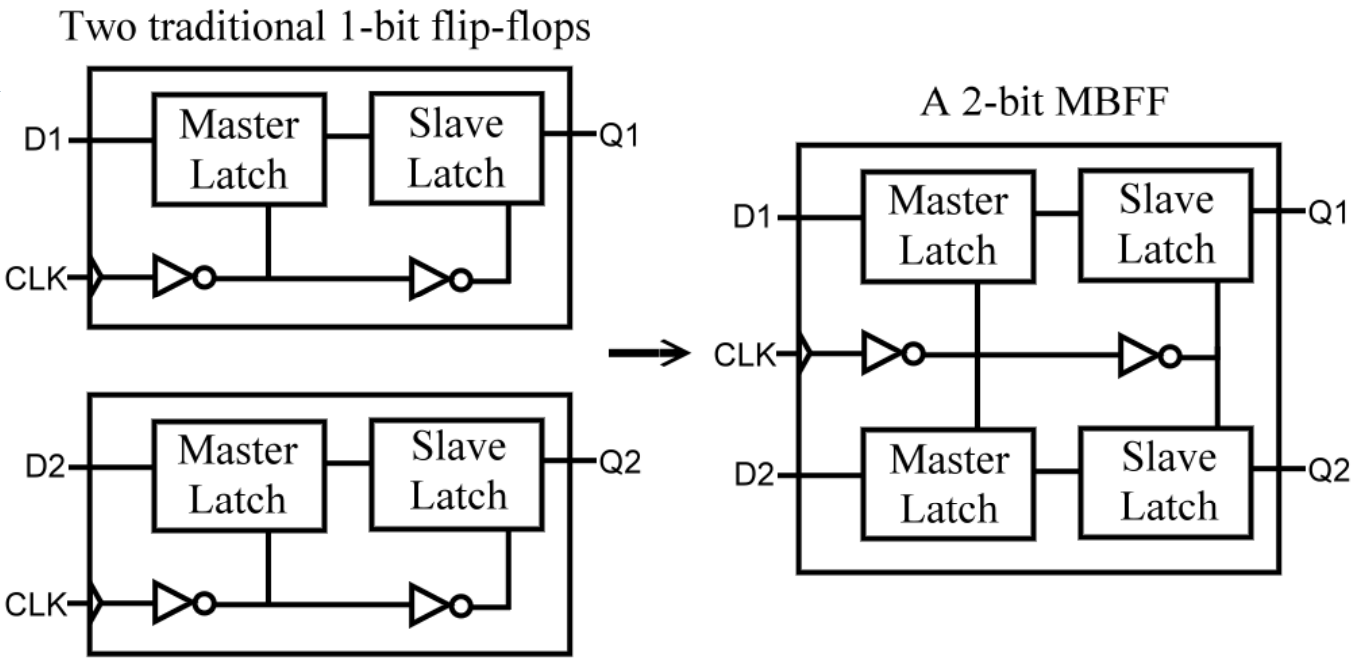
Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions

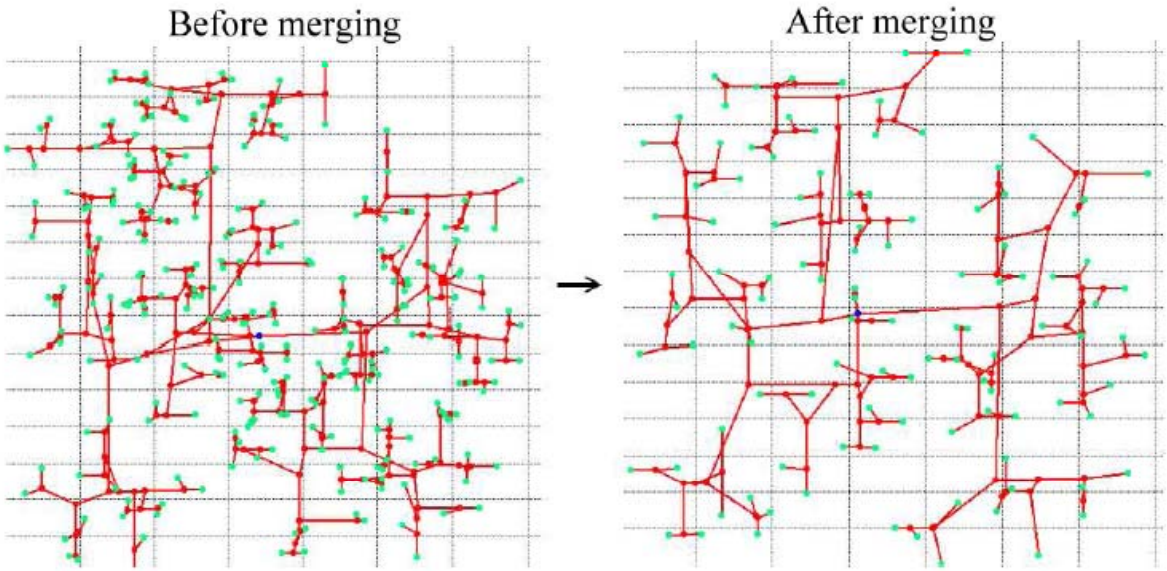
Flip-Flop Merging

- Merge several 1-bit Flip-Flops into a Multi-bit Flip-Flop (MBFF)
  - Eliminate some inverters and area
  - Reduce the # clock sinks

Flip-Flop Merging



Reduction of clock sinks

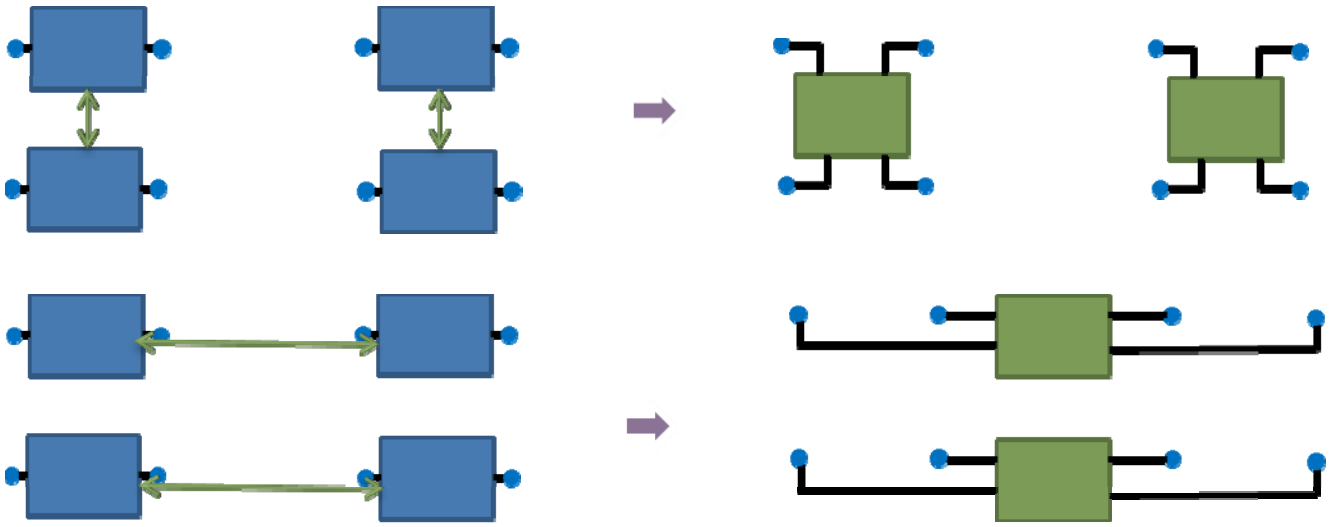


Related Work

- [15] *Post-placement power optimization with multi-bit flip-flops, I*CCAD'10
- The objective of [15] is to minimize the total FF Power
  - However, our objective function is to minimize the # clock sinks and switching power of signal nets

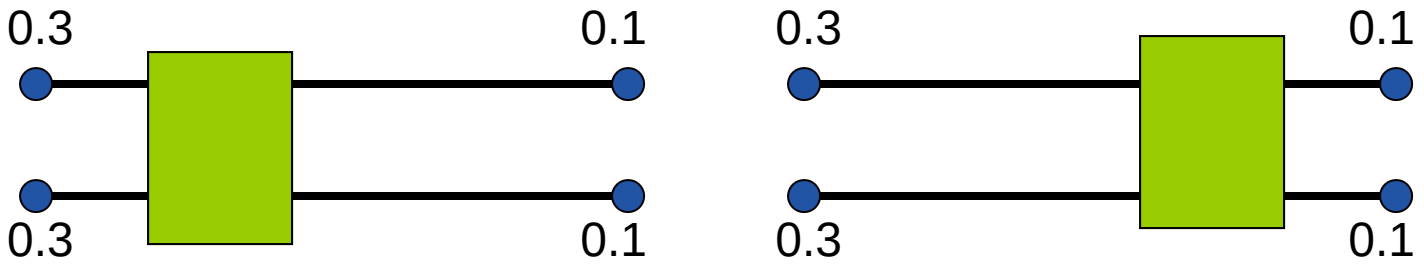
Wirelength of Signal Nets

- Different merging solutions will affect the wirelength and switching power of signal nets differently



Post-Placement Relocation

- After merging, we need to relocate these MBFFs
  - It will affect the total switching power of signal nets



Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions

Problem Formulation

- Inputs
  - A preplaced design and a MBFF Library
- Objectives
  - Minimize the # sinks in clock network
  - Minimize the switching power of signal nets

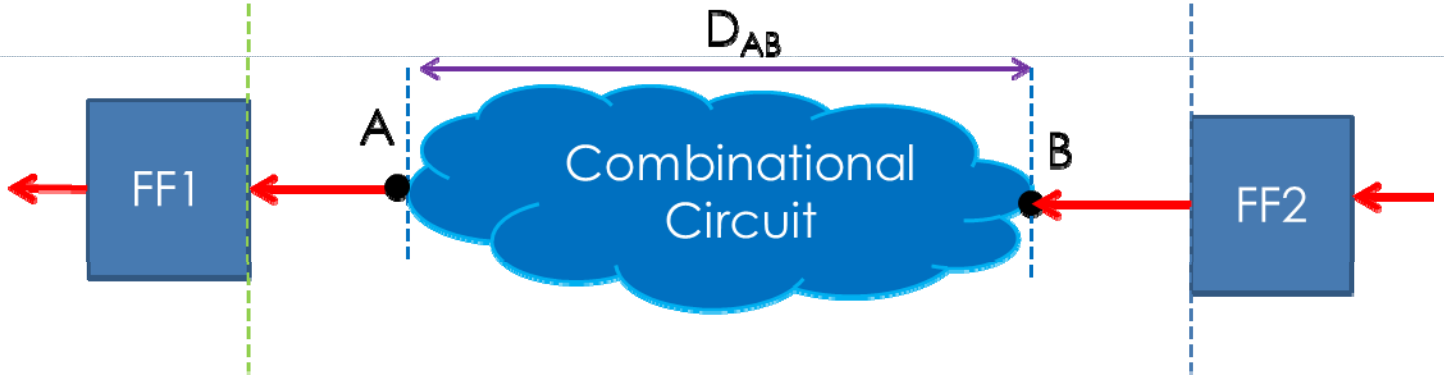
$$\omega \times \frac{\text{Final \#sinks}}{\text{Original \#sinks}} + (1-\omega) \times \frac{\sum_{net_i \in F} \alpha_i \times \text{Final } WL_i}{\sum_{net_i \in F} \alpha_i \times \text{Original } WL_i}$$

α_i is the switching rate of signal nets

Constraints

- Guarantee there is no timing violation
  - Feasible region of FFs
- Control the placement density
  - Maintain the quality of legalization
  - Consider routing congestion

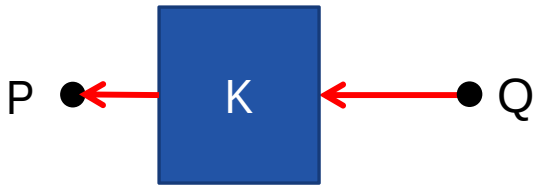
Feasible Region of a FF

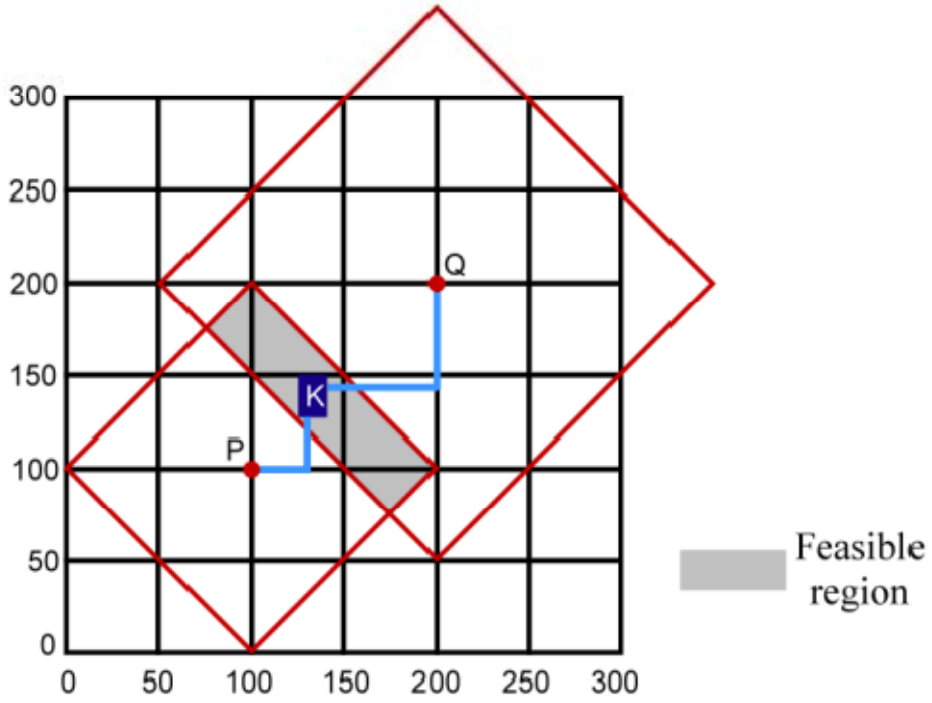


Slack = Maximum allowed delay - D_{AB}

$Slack_A = Slack_B = Slack / 2$

Feasible Region of a FF (cont.)



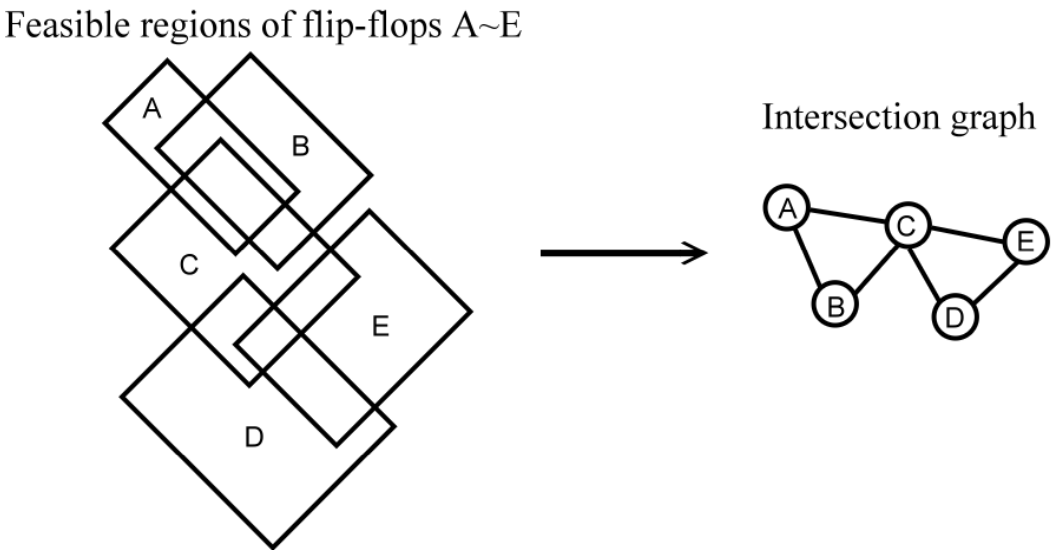


Outline

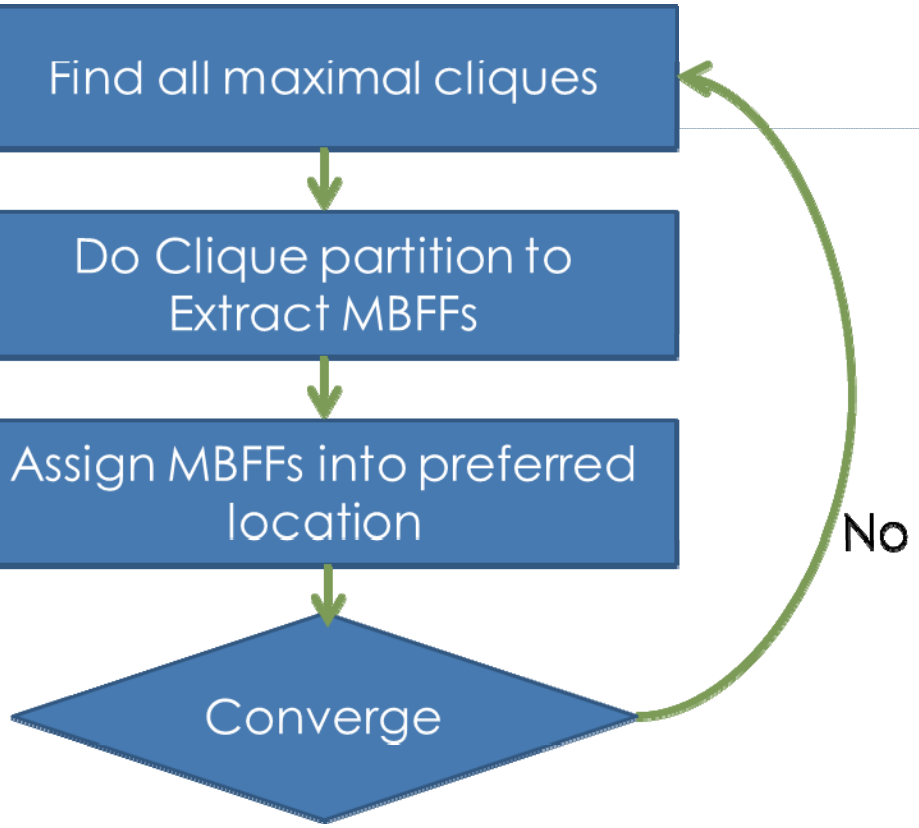
- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions

Intersection Graph

- Get the feasible regions of all FFs
- The intersection of feasible regions can be represented by an intersection graph



Design Flow


Find all maximal cliques
Do Clique partition to Extract MBFFs
Assign MBFFs into preferred location
Converge
No

Find all the Maximal Cliques

- Finding all the maximal cliques is NPC in general graph
  - However, it can be solved in polynomial time in the rectangle intersection graph
  - Solve by the sweep line algorithm

MBFF Extraction

- We want to extract the MBFFs by clique partitioning
  - Clique partitioning is a NP-Hard problem
- Different extraction strategies will affect
  - The number of clock sinks
  - The wirelength of signal nets

MBFF Extraction (cont.)

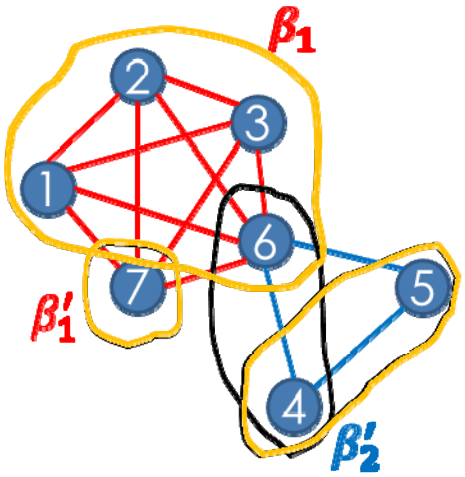
- Cost of creating MBFF β
  - D(β): the merging possibility of FFs in β
  - B(β): the # bits of β
  - Switching power of signal nets connected to β

$$cost(\beta) = \lambda \times \frac{\mathcal{D}(\beta)}{\mathcal{B}(\beta)} + (1-\lambda) \times \frac{\sum_{net_i \in Net(\beta)} \alpha_i \times \text{Estimated } WL_i}{\sum_{net_i \in Net(\beta)} \alpha_i \times \text{Original } WL_i}$$

α_i is the switching rate of signal nets

Example of Extraction Algorithm

- Assume we have 1/2/4-bit MBFF in library
- There are two maximal cliques
  - c_1 = {1,2,3,6,7}, c_2 = {4,5,6}
- Random sampling 1, 2 or 4 of FFs from c_1, c_2
  - β_1 = {1,2,3,6}, β_2 = {4,6}
- cost(β_1) < cost(β_2) => select β_1
  - Re-sampling β_1' = {7} from c_1
- cost(β_2) < cost(β_1')
  - FF6 already covered
  - Re-sampling β_2' = {4, 5} from c_2
- Final Extraction {β_1, β_2', β_1'}



MBFF Relocation

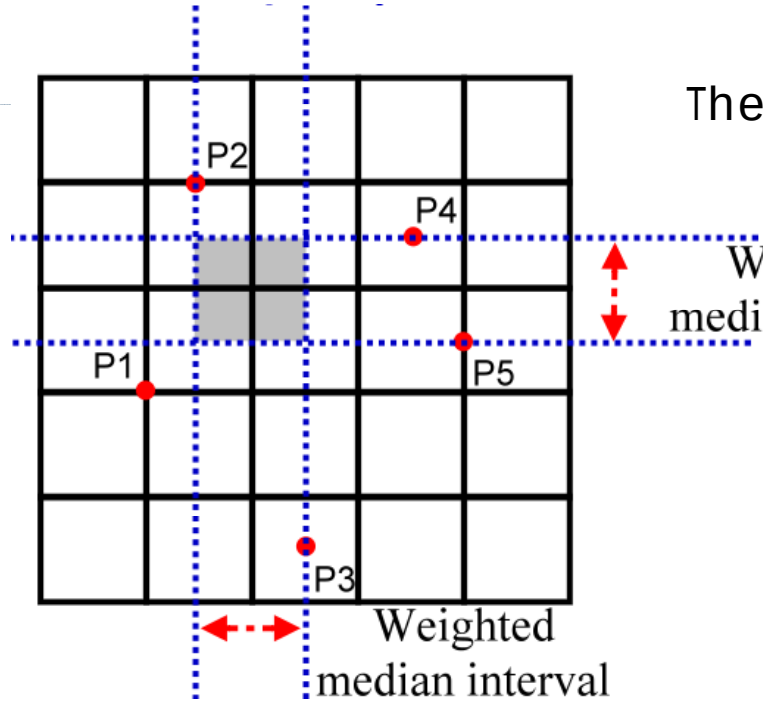
- For a MBFF β, we want to minimize the switching power of its signal nets

$$min \sum_{net_i \in Net(\beta)} \alpha_i \times WL_i$$

α_i is the switching rate of signal nets

- We can formulate it as a weighted median problem

MBFF Relocation (cont.)

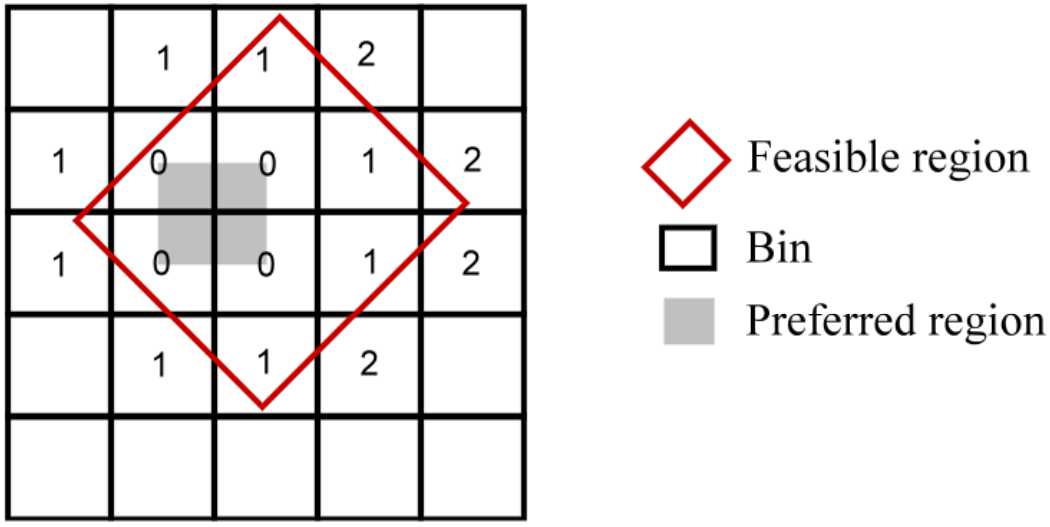


The weight of P1~P5 are 2:1:1:3:1

- Fanin/fanout of a MBFF
- Bin
- Preferred region

MBFF Relocation (cont.)

- Because of bin density constraints, some MBFFs cannot be placed in preferred region



Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions

Experimental Setup

- Implemented in C++
- Work on Linux with 2.13GHz CPU
- We have 9 test cases
  - r1~r5 from [22] *Exact Zero-Skew*
  - t0~t3 from 2010 CAD contest of Taiwan
  - Random generate switching rates 5%~15%

Experimental Results

- Reduction of clock sinks and wirelength of clock tree

Test cases	#FF			FF area reduction	Clock tree WL(nm)			Run time(s)
	original	final	reduction		original	final	reduction	
r1	267	101	62.17%	3.30%	1325183	930661	29.77%	1.53
r2	598	223	62.70%	3.38%	2621623	1781824	32.03%	5.91
r3	862	298	65.42%	3.47%	3357327	2184565	34.93%	6.98
r4	1903	592	68.89%	3.56%	6839628	4185940	38.79%	28.20
r5	3101	921	70.29%	3.63%	10145960	6002024	40.84%	51.62
t0	120	37	69.16%	3.58%	39637	22545	43.12%	0.03
t1	60000	15040	74.93%	3.75%	3981765	1955086	50.89%	1053.12
t2	5524	1525	72.39%	3.68%	985348	543020	44.89%	1.98
t3	953	246	74.18%	3.73%	201755	102271	49.39%	1.14
avg.			68.90%	3.56%			40.51%	

Experimental Results (cont.)

- Reduction of wirelength and estimated switching power of nets connected to FFs

Test cases	$\sum_{net_i \in F} WL_i$			$\sum_{net_i \in F} \alpha_i \times WL_i$		
	original	final	reduction	original	final	reduction
r1	1743703	1756925	-0.75%	179802	176664	1.74%
r2	3930879	3732569	5.04%	400928	370500	7.58%
r3	5672241	5191913	8.46%	574642	511426	11.00%
r4	12616681	12066921	4.35%	1266302	1187960	6.18%
r5	20528314	19012768	7.38%	2061324	1856472	9.93%
t0	83285	74365	10.71%	8755	7482	14.53%
t1	53624875	33077705	38.31%	5356145	3157927	41.04%
t2	3562985	2099595	41.07%	357151	204907	42.62%
t3	576710	448090	22.30%	58576	43931	25.00%
avg.			15.21%			17.74%

Comparison with [15]

- Our algorithm can be modified to target the objectives of [15]

Test cases	#FFs	[15]			Ours		
		FF Power Red.	HPWL Red.	Run time(s)	FF Power Red.	HPWL Red.	Run time(s)
c1	98	14.8%	8.7%	0.01	15.64%	8.2%	0.01
c2	423	16.9%	5.3%	0.04	17.52%	11.1%	0.05
c3	1692	17.1%	5.2%	0.10	17.41%	11.5%	0.22
c4	5129	16.8%	5.5%	0.28	17.07%	11.5%	0.72
c5	10575	17.1%	5.1%	0.60	17.29%	13.4%	1.89
c6	169200	17.2%	5.1%	78.92	17.52%	11.8%	36.12
avg.		16.65%	5.82%		17.03%	11.25%	

Conclusions

- We present a power-driven flip-flop merging and relocation approach to reduce the switching power consumption of the entire circuit

Q&A

- Thanks for your attention