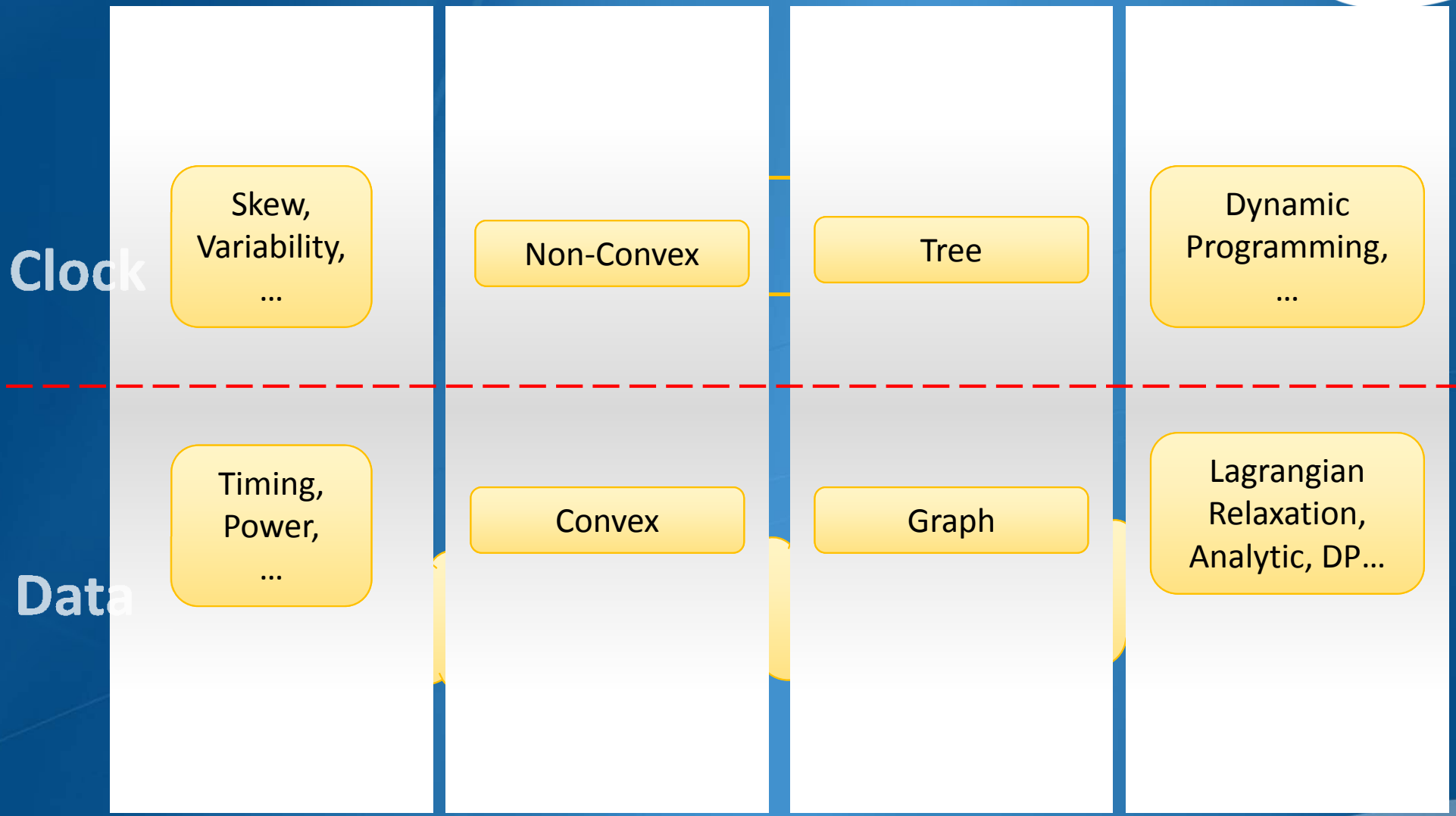
Simultaneous Clock and Data Gate Sizing with Common Global Objective

Gregory Shklover
Ben Emanuel
Intel Corporation

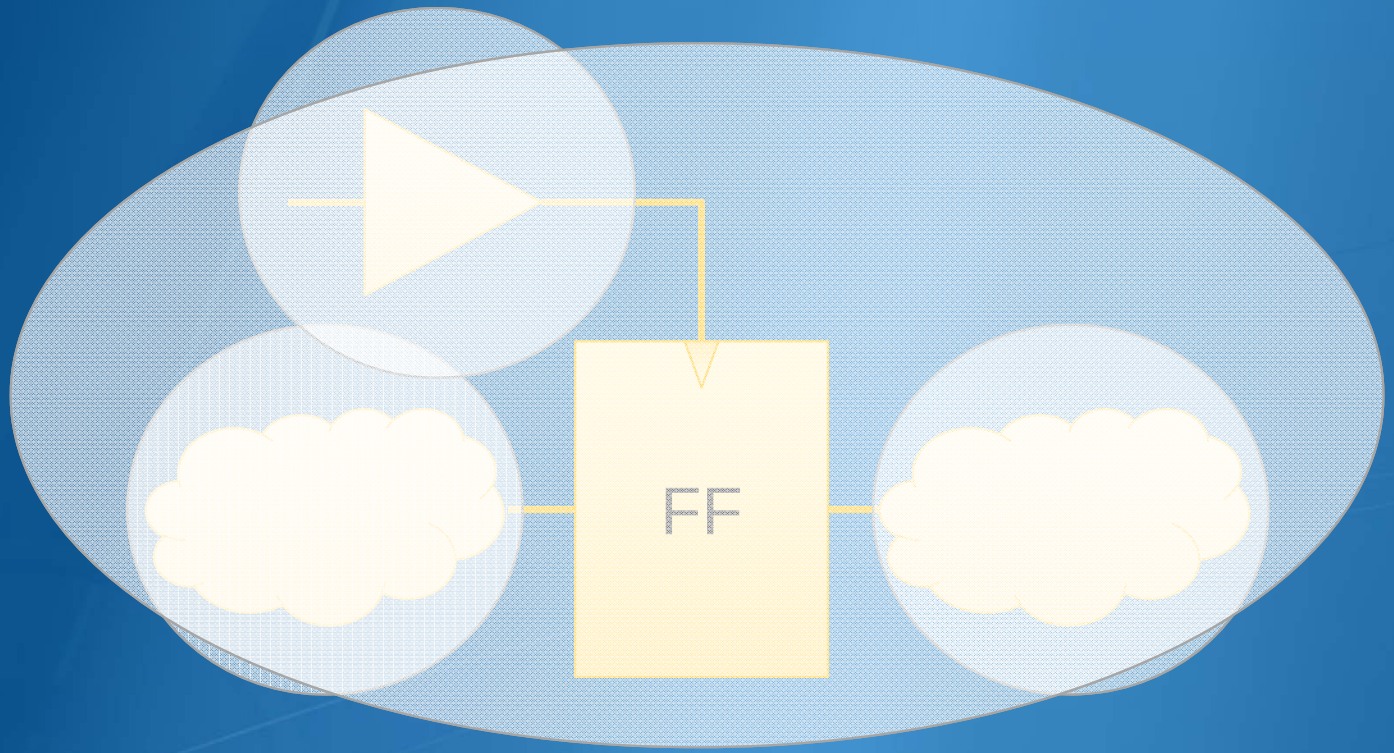
Outline

- Motivation
- Data Gate Sizing by Lagrangian Relaxation (LR)
- Clock & Data Gate Sizing Algorithm
- Experimental Results

Motivation


Clock
Skew, Variability, …
Non-Convex
Tree
Dynamic Programming, …
Data
Timing, Power, …
Convex
Graph
Lagrangian Relaxation, Analytic, DP…

Motivation (cont.)



Balance power and timing in both clock and data for better global solution.

Lagrangian Relaxation

- Proposed by C. Chen *et al*
- Exploits the nature of timing constraints to reduce complexity
- Efficient, suitable for industrial design flows (standard library with Vt/sizing).

Gate Sizing by LR

$$Objective = \sum_{n \in Seq \cap Out} \max(0, -slack_n) + \alpha \times \sum_{c \in Cells} power_c$$

$Subject\ to$:

$\forall i \in Inputs: a_i = const_i$

$\forall u \to v: a_u + delay_{u \to v} \leq a_v$

$\forall s \in Seq: slack_s \leq cycle + a_{s,clk} - a_{s,d} - setup_s$

$\forall o \in Outputs: slack_o \leq cycle - a_o - const_o$

Setup constraints

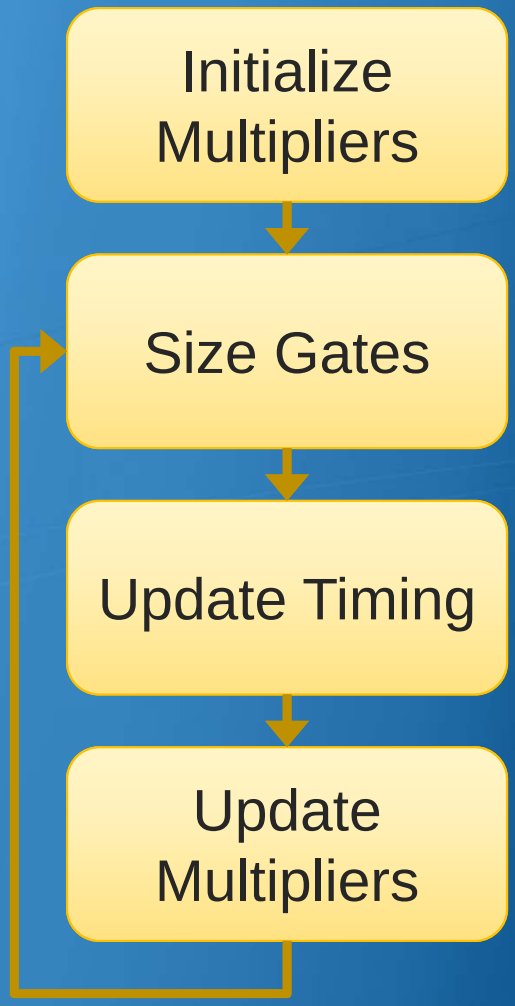
Gate Sizing by LR

Lagra ers (λ)
+ KKT- fication

$$Objective_\lambda = \sum_{u \to v \in E} \lambda_{u,v} \times delay_{u \to v} +$$

$$+ \sum_{s \in Seq} \lambda_{s,d} \times setup_s +$$

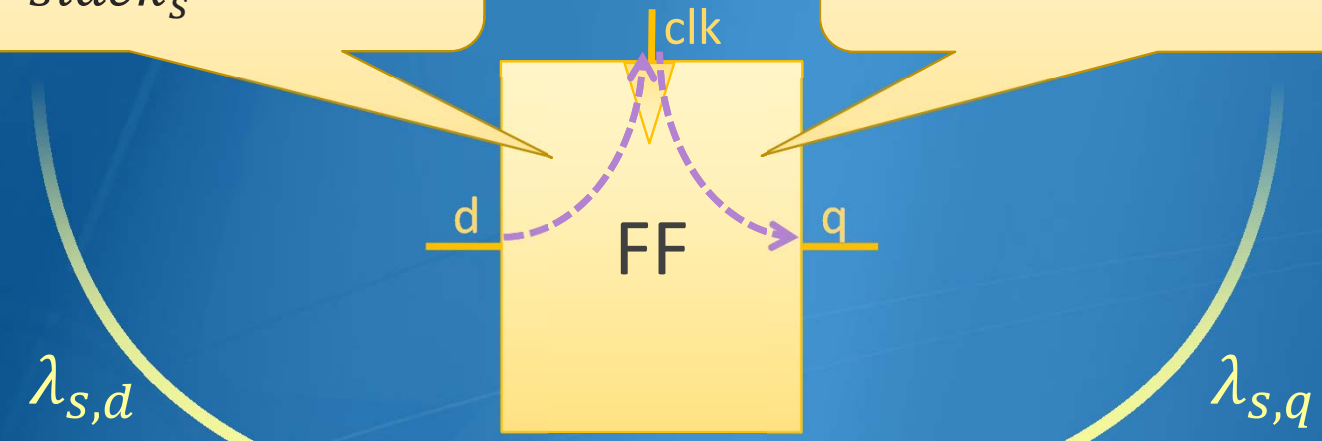
$$+ \alpha \times \sum_{c \in Cells} power_c$$



Clock Formulations

$$a_{s,d} + setup_s - a_{s,clk} - cycle \leq -slack_s$$

$$a_{s,clk} + delay_{clk \to q} \leq a_{s,q}$$



$\lambda_{s,d}$

$\lambda_{s,q}$

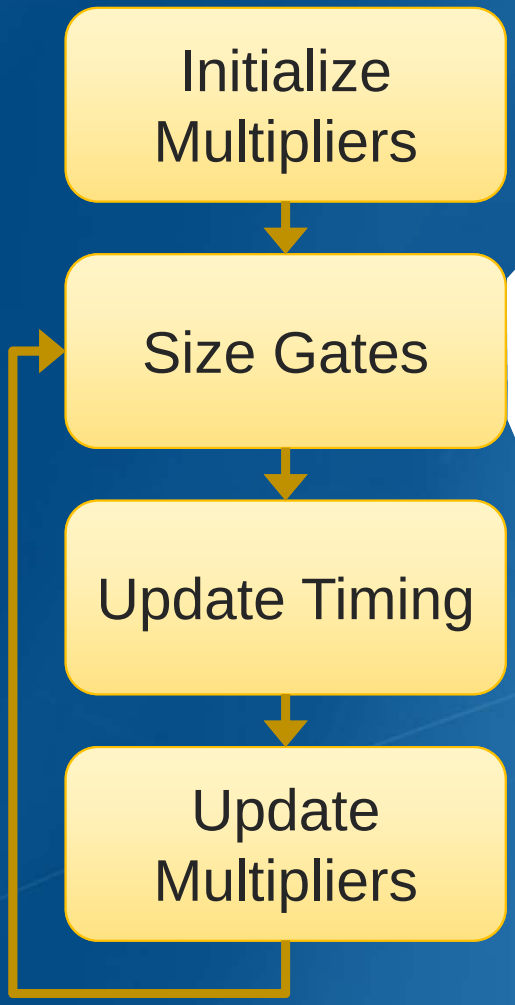
$$Objective_\lambda = \cdots + \sum_{s \in Seq} (\lambda_{s,q} - \lambda_{s,d}) \times a_{s,clk} + \alpha \times \sum_{c \in Clock} power_c$$

Clock Formulations (cont.)

$$a_{s,clk} = delay_{root \to \cdots \to v} + delay_{v \to w} + \dots$$

$$Objective_{\lambda} = \cdots + \sum_{g \in Clock} delay_g \times \sum_{s \in Leaves_g} (\lambda_{s,q} - \lambda_{s,d})$$

Clock Gate Sizing Algorithm



Dynamic Programming (DP) Algorithm

- Originates from buffered tree construction by Van Ginneken
- Systematically explores solution space by building partial solutions bottom-up

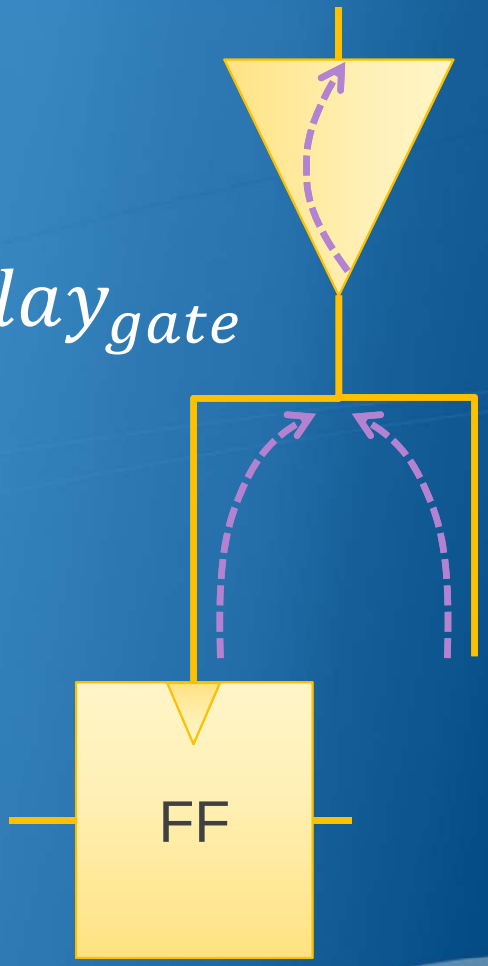
DP Algorithm

$$Objective_\lambda = \cdots + \sum_{c \in Clock} power_c + \sum_{g \in Clock} delay_g \times \Lambda_g$$

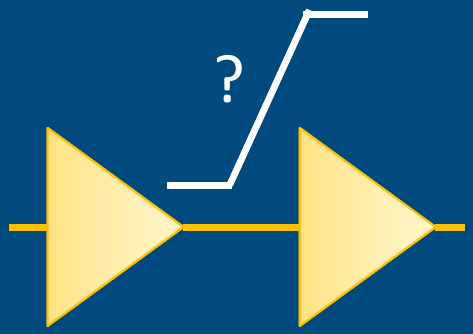
- Set of solutions per tree node n
  - $solution_n \stackrel{\text{def}}{=} \langle cap_n, obj_n \rangle$
- Pruning criterion
  - $cap_a = cap_b \wedge obj_a \geq obj_b$
  - (differs from minimal delay objectives)

DP Algorithm

- Gate sizing:
  - $cap_{new} = c_{gate}$
  - $obj_{new} = obj + \alpha power_{gate} + \Lambda delay_{gate}$
- Solution merge:
  - $cap_{new} = cap_l + cap_r$
  - $obj_{new} = obj_l + obj_r$
- Leaf nodes:
  - $S_n = \{\langle c_n, 0 \rangle\}$

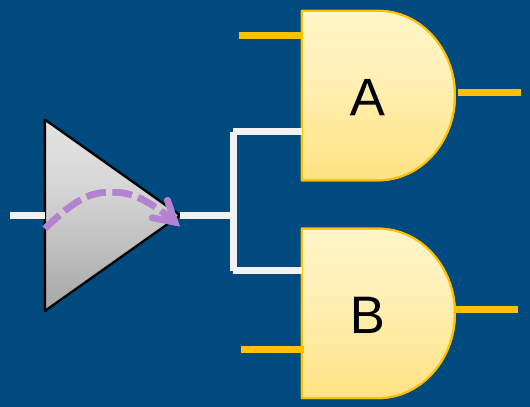


Additional Considerations



Input slews

Approximation
+
convergence

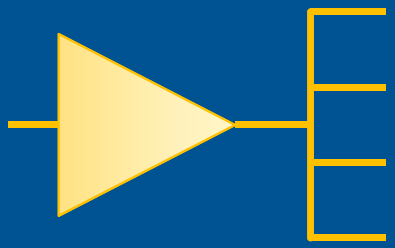


Side-load effects

Approximation
+
convergence

Additional Considerations

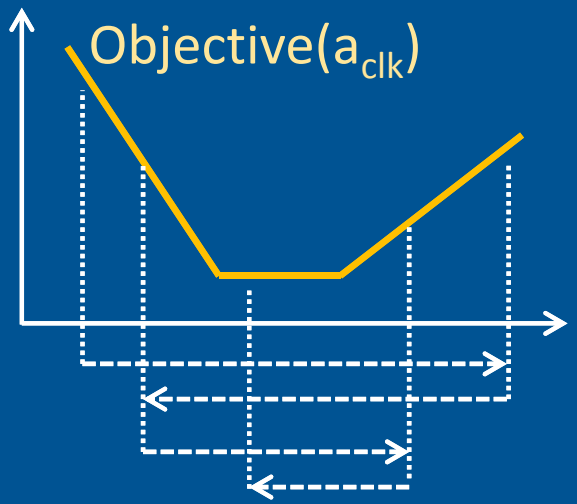
$N \times N \times \ldots$



Exponential number of solutions

k-Sampling

$O(max(k,L)kN)$



Convergence

"Cooling"

$|delay_g^i - delay_g^{i+1}| \leq R^{i+1}$

Experimental Results

- Reference: Separate optimization
  - Data sizing for given clock schedule
  - Timing-preserving clock sizing

VS

- Test: Simultaneous clock & data sizing
  - Same objective as above, but clock and data sized simultaneously

Experimental Results

Block	Total Slack		Leakage		ClkDPwr		Total Power	
	ref	new	ref	new	ref	new	ref	new
block1	-0.038	-0.044	2.26	2.10	2.07	1.77	4.33	3.87
block2	-0.051	-0.015	1.80	1.77	1.38	1.36	3.19	3.14
block3	-2.387	-1.902	6.59	6.22	5.51	5.18	12.10	11.40
block9	-1.075	-0	6.49	6.24		0.89	7.44	7.12
block14	-0.168	-0.072	2.57	2.51	1.78	1.70	4.35	4.20
block15	-0.062	-0.063	3.10	3.02	2.33	1.97	5.43	4.99
Total	-6.02	-4.03	60.88	58.03	33.32	31.52	94.20	89.55

Block	Total Slack		Leakage		ClkDPwr		Total Power	
	ref	new	ref	new	ref	new	ref	new
Total	-6.02	-4.03	60.88	58.03	33.32	31.52	94.20	89.55

Useful skew: better timing, lower gate leakage

Natively balances clock power vs timing

Summary

- Extend traditional gate sizing to simultaneous clock & data optimization
- Benefits of global optimization
  - Balances between useful skew, clock power and data power
- Future directions:
  - Extend optimization objective
  - Topological changes

Acknowledgements

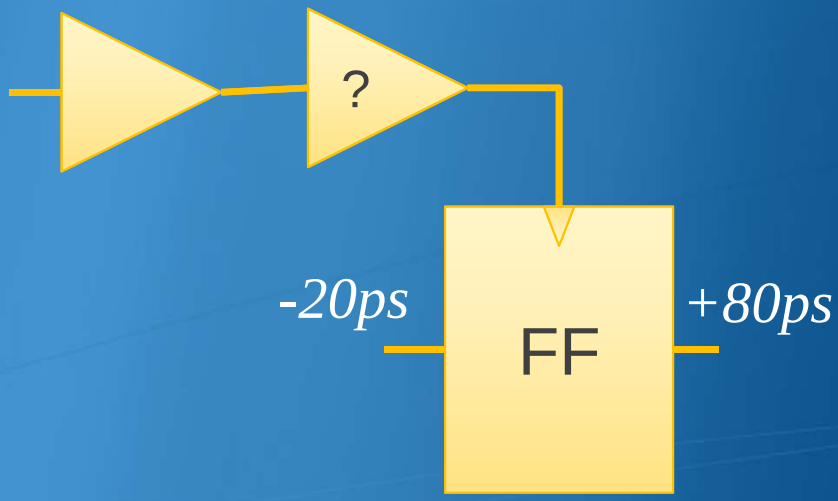
- Prof. C. Chen for participating in discussion and reviews
- Yoram Aloni and Lior Nissim for supporting this effort

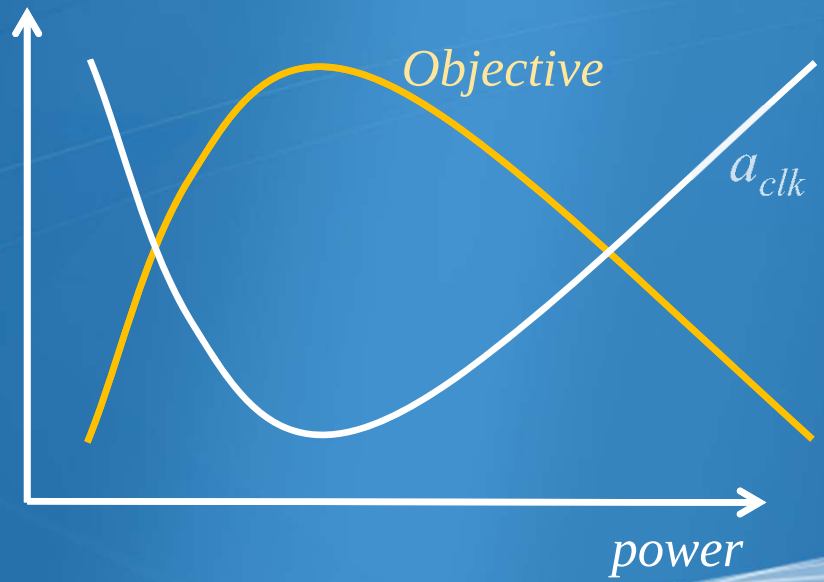
Questions?

Extras

Convexity

$$Objective = \alpha \times power - slack$$

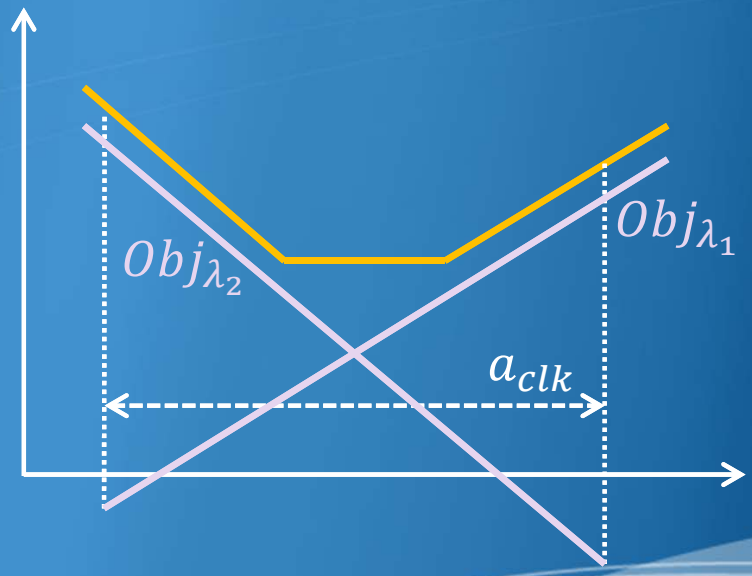




Convergence

Block	Total Slack	
	cooling off	cooling on
block1	-0.023	-0.023
block2	-0.019	-0.019
block3	-2.649	-1.885
block4	-0.036	-0.013
block5	-0.166	-0.160
block6	-0.153	-0.064
block7	-0.126	-0.118
block8	-0.224	-0.211
block9	-0.693	-0.535
block10	-0.185	-0.083
block11	-0.662	-0.553
block12	-0.102	-0.118
block13	-0.073	-0.032
block14	-0.055	-0.053
block15	-0.130	-0.052
Total	-5.29	-3.92

Convergence control eliminates overshoot while optimizing piecewise linear objective.



Backup