Gate Sizing and Vth Assignment for Asynchronous Circuits Using Lagrangian Relaxation

Gang Wu, Ankur Sharma and Chris Chu

Iowa State University

Introduction

- Asynchronous Gate Selection Problem:
  - Selecting gates of different sizes and Vth from a standard cell library to minimize leakage power and performance

Previous Works (1/2)

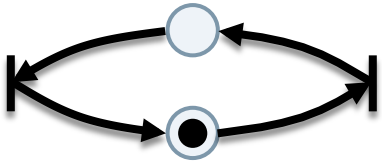
- Gate selection algorithm for synchronous circuits:
  - Convex programming, sensitivity based algorithms, dynamic programming based algorithms, Lagrangian relaxation (LR) based algorithms
- Most of the leading gate selection algorithms are applying the LR based approach:
  - G. Flach et al., **Effective Method for Simultaneous Gate Sizing and Vth Assignment Using Lagrangian Relaxation**, TCAD 2014
  - M. M. Ozdal et al., **Algorithms for Gate Sizing and Device Parameter Selection for High-performance Designs**, TCAD 2012

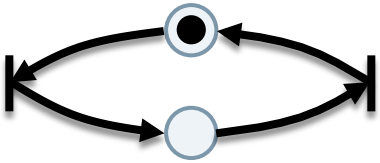
Previous Works (2/2)

- Asynchronous gate selection can be more critical
  - Higher gate counts
- Most works try to leverage synchronous EDA tools:
  - P. A. Beerel et al., **Proteus: An ASIC Flow for GHz Asynchronous Designs,** Design Test of Computers 2011
  - Y. Thonnart et al., **A Pseudo-synchronous Implementation Flow for WCHB QDI Asynchronous Circuits,** ASYNC 2012
- Gate selection algorithm specific for asynchronous circuits:
  - B. Ghavami et al., **Low Power Asynchronous Circuit Back- End Design Flow,** Microelectronics Journal 2011

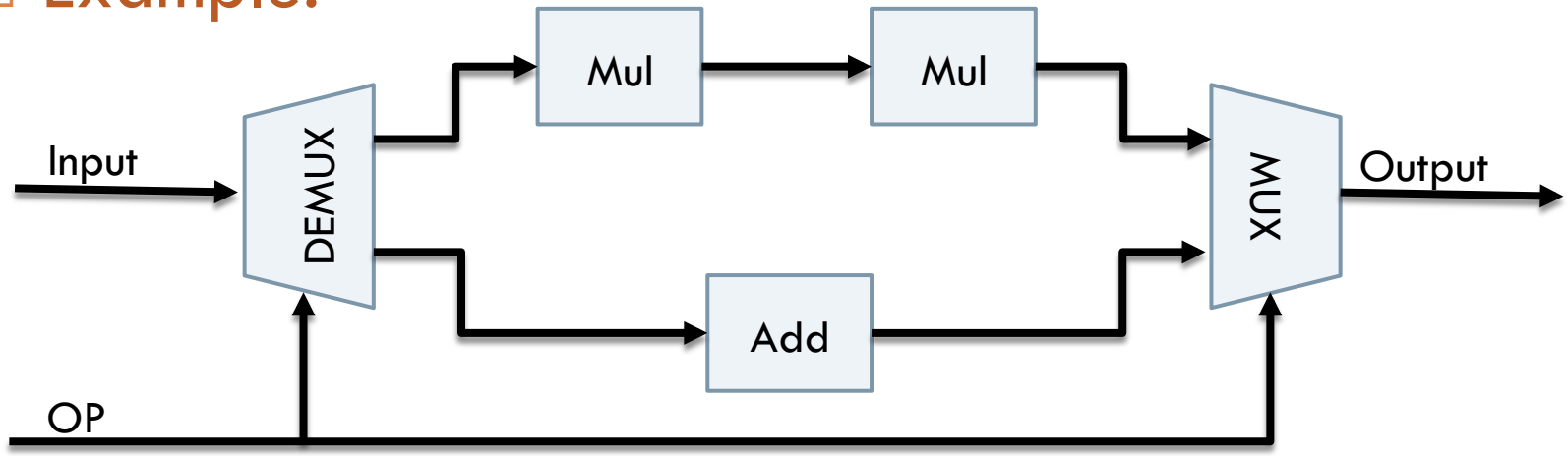
Backgrounds (1/2)

- Full Buffer Channel Net Model
  - Each cell is modeled using a transition
  - Channel between cells is modeled with a pair of places

Full Channel: 

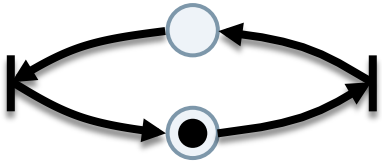
Empty Channel: 

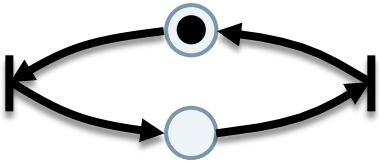
- Example:



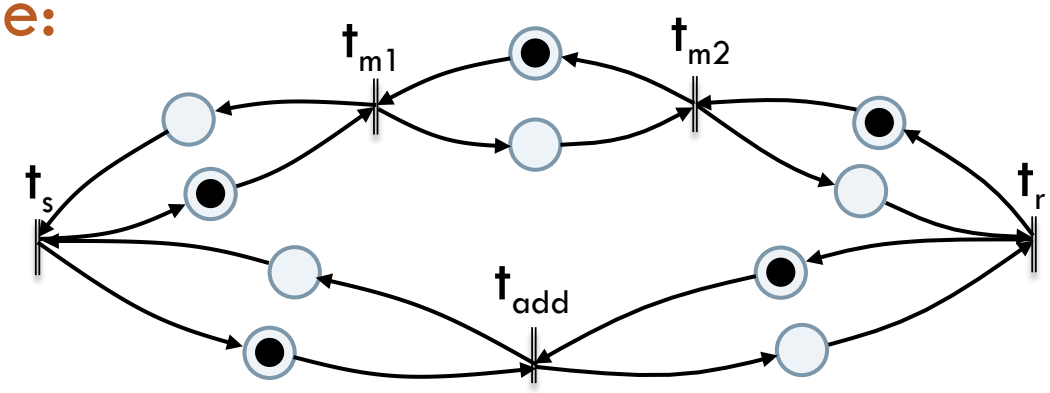
Backgrounds (1/2)

- Full Buffer Channel Net Model
  - Each cell is modeled using a transition
  - Channel between cells is modeled with a pair of places

Full Channel: 

Empty Channel: 

- Example:



Backgrounds (2/2)

- Performance Analysis
  - A linear program approach [J. Magott IPL '84]

$$\begin{aligned} &\text{Minimize} \quad \tau \\ &\text{Subject to} \quad a_i + D_{ij} - m_{ij}\tau \leq a_j \quad \forall(i,j) \end{aligned}$$

- Timing Constraints
  - Minimum and maximum bounded delay
  - Relative timing constraints

Problem Formulation

- Minimizing both total leakage power consumption and cycle time subject to timing constraints:

$$\begin{aligned}
&\text{Minimize} && \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + \alpha\tau/\tau_0 \\
&\text{Subject to} && a_i + D_{ij} - m_{ij}\tau \le a_j && \forall\, p(i,j) \in P && (1)\\
& && L_{ij} \le a_j - a_i \le U_{ij} && \forall\, p(i,j) \in P_b && (2)\\
& && |(a_i - a_k) - (a_j - a_k)| \le I_{ij} && \forall\, p(i,j) \in P_{rt} && (3)
\end{aligned}$$

- Transform all constraints into the same form:

$$(a_i + L_{ij} \le a_j) \wedge (a_j - U_{ij} \le a_i)$$
$$(a_j - I_{ij} \le a_i) \wedge (a_i - I_{ij} \le a_j)$$

Problem Formulation

- Minimizing both total leakage power consumption and cycle time subject to timing constraints:

$$\begin{aligned}
&\text{Minimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + \alpha\tau/\tau_0 \\
&\text{Subject to} \quad a_i + D_{ij} - m_{ij}\tau \le a_j \qquad \forall\, p(i,j) \in P \qquad (1) \\
&\qquad\qquad L_{ij} \le a_j - a_i \le U_{ij} \qquad \forall\, p(i,j) \in P_b \qquad (2) \\
&\qquad\qquad |(a_i - a_k) - (a_j - a_k)| \le I_{ij} \;\forall\, p(i,j) \in P_{rt} \qquad (3)
\end{aligned}$$

- A more concise representation of the primal problem:

$$\begin{aligned}
\mathcal{PP}: \quad &\text{Minimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + \alpha\tau/\tau_0 \\
&\text{Subject to} \quad a_i + \hat{D}_{ij} - \hat{m}_{ij}\tau \le a_j \quad \forall (i,j)
\end{aligned}$$

Outline – Gate Selection Algorithm

- Primal Problem(PP)
  - Minimizing both total leakage and cycle time subject to timing constraints
- Lagrangian Relaxation Subproblem (LRS)
  - Obtained by relaxing constraints of PP
  - Provide a lower bound of PP
- Simplified LRS (LRS*)
  - Obtained by applying KKT condition to LRS
  - Solved using greedy sizing technique
- Lagrangian Dual Problem (LDP)
  - Iteratively solve LRS* to improve the lower bound

Outline – Gate Selection Algorithm

- Primal Problem(PP)
  - Minimizing both total leakage and cycle time subject to timing constraints
- **Lagrangian Relaxation Subproblem (LRS)**
  - **Obtained by relaxing constraints of PP**
  - **Provide a lower bound of PP**
- Simplified LRS (LRS*)
  - Obtained by applying KKT condition to LRS
  - Solved using greedy sizing technique
- Lagrangian Dual Problem (LDP)
  - Iteratively solve LRS* to improve the lower bound

Lagrangian Relaxation Subproblem

- Relax all the constraints into objective function

$$\mathcal{PP}: \quad \text{Minimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + \alpha\tau/\tau_0$$
$$\text{Subject to} \quad a_i + \hat{D}_{ij} - \hat{m}_{ij}\tau \leq a_j \quad \forall(i,j)$$

$$\mathcal{LRS}: \quad \text{Mimimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + \alpha\tau/\tau_0 + \sum_{\forall(i,j)} \lambda_{ij}(a_i + \hat{D}_{ij} - \hat{m}_{ij}\tau - a_j)$$

- Solving LRS will provide a lower bound to the optimal solution of PP

Outline – Gate Selection Algorithm

- Primal Problem(PP)
  - Minimizing both total leakage and cycle time subject to timing constraints
- Lagrangian Relaxation Subproblem (LRS)
  - Obtained by relaxing constraints of PP
  - Provide a lower bound of PP
- **Simplified LRS (LRS*)**
  - **Obtained by applying KKT condition to LRS**
  - **Solved using greedy sizing technique**
- Lagrangian Dual Problem (LDP)
  - Iteratively solve LRS* to improve the lower bound

Simplified LRS (1/2)

- Rearrange LRS

$$\mathcal{LRS}: \quad \text{Mimimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + \alpha\tau/\tau_0 + \sum_{\forall(i,j)} \lambda_{ij}(a_i + \hat{D}_{ij} - \hat{m}_{ij}\tau - a_j)$$

  - Group all the coefficients associated with τ, a_k

$$\text{Mimimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + (\alpha - \sum_{\forall(i,j)} \lambda_{ij}\hat{m}_{ij})\tau/\tau_0 + \sum_{k\in T}(\sum_{\forall(k,j)} \lambda_{kj} - \sum_{\forall(i,k)} \lambda_{ik})a_k + \sum_{\forall(i,j)} \lambda_{ij}\hat{D}_{ij}$$

Simplified LRS (2/2)

- Rearranged LRS:

$$\text{Mimimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + (\alpha - \sum_{\forall(i,j)} \lambda_{ij}\hat{m}_{ij})\tau/\tau_0$$

$$+ \sum_{k\in T}(\sum_{\forall(k,j)} \lambda_{kj} - \sum_{\forall(i,k)} \lambda_{ik})a_k$$

$$+ \sum_{\forall(i,j)} \lambda_{ij}\hat{D}_{ij}$$

- Use KKT condition to simplify LRS:

KKT Stationarity Condition:

$$\frac{\partial \mathcal{L}}{\partial a_i} = 0, \quad \frac{\partial \mathcal{L}}{\partial \tau} = 0$$

$$\mathcal{KKT}: \qquad \alpha = \sum_{\forall(i,j)} \lambda_{ij}\hat{m}_{ij}$$

$$\sum_{\forall(k,j)} \lambda_{kj} = \sum_{\forall(i,k)} \lambda_{ik} \quad \forall\, k \in T$$

Simplified LRS (2/2)

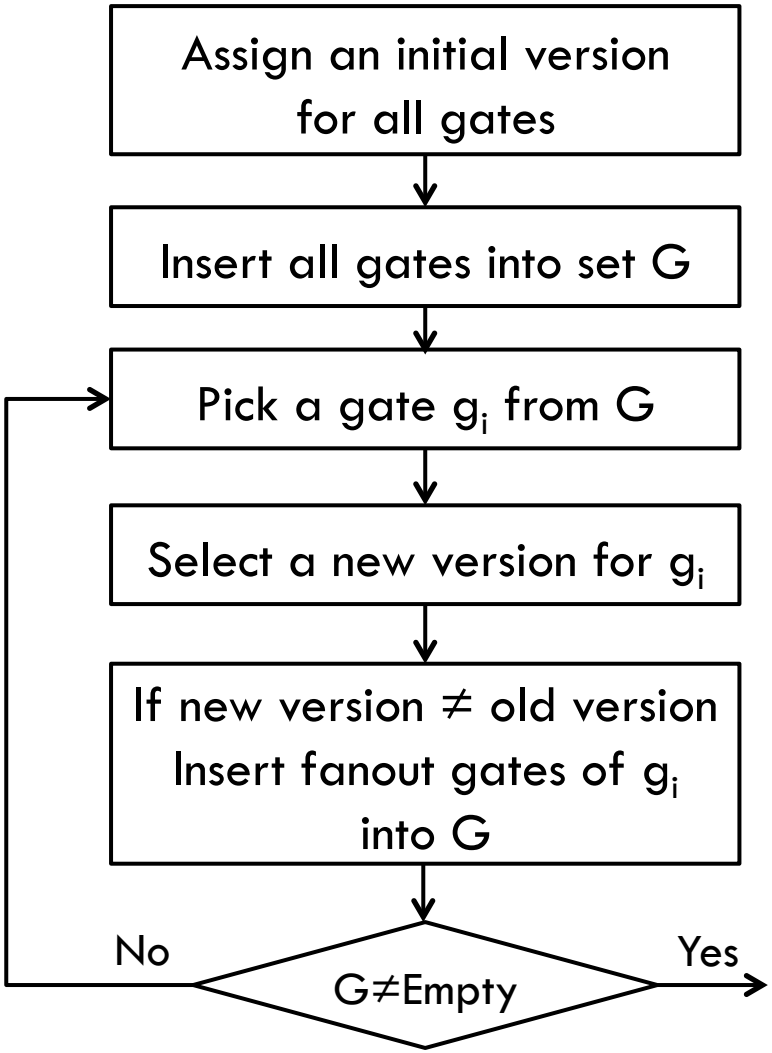
- Rearranged LRS:

$$\text{Mimimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + (\alpha - \sum_{\forall(i,j)} \lambda_{ij}\hat{m}_{ij})\tau/\tau_0$$
$$+ \sum_{k\in T}(\sum_{\forall(k,j)} \lambda_{kj} - \sum_{\forall(i,k)} \lambda_{ik})a_k$$
$$+ \sum_{\forall(i,j)} \lambda_{ij}\hat{D}_{ij}$$

- Use KKT condition to simplify LRS:

$$\mathcal{LRS}^* : \quad \text{Minimize} \quad \text{leakage}(\boldsymbol{g})/\text{leakage}(\boldsymbol{g_0}) + \sum_{\forall(i,j)} \lambda_{ij}\hat{D}_{ij}$$

Solve Simplified LRS



- Select a new version for g_i
  - Iterate through all possible options of current gate
  - Calculate the cost for all timing arcs:

g_i, fanin(g_i), fanout(g_i), side(g_i)

$$\text{Cost}(g_i) = \text{leakage}(g_i) + \sum_{(u,v)\in Arc_i} \lambda_{uv}\hat{D}_{uv}$$

  - Pick option with lowest cost

Outline – Gate Selection Algorithm

- Primal Problem(PP)
  - Minimizing both total leakage and cycle time subject to timing constraints
- Lagrangian Relaxation Subproblem (LRS)
  - Obtained by relaxing constraints of PP
  - Provide a lower bound of PP
- Simplified LRS (LRS*)
  - Obtained by applying KKT condition to LRS
  - Solved using greedy sizing technique
- **Lagrangian Dual Problem (LDP)**
  - **Iteratively solve LRS* to improve the lower bound**

Lagrangian Dual Problem

- Find the specific λ to get the tightest lower bound:

$$\mathcal{LDP}: \quad \text{Maximize} \quad \mathcal{LRS}$$
$$\text{Subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}$$

- Use the equivalent yet simpler problem LRS*

$$\text{Maximize} \quad \mathcal{LRS}^*$$
$$\text{Subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0}, \ \boldsymbol{\lambda} \in \mathcal{KKT}$$

Solve Lagrangian Dual Problem (1/2)

- We apply a direction finding approach* instead of the subgradiant optimization approach
  - Easier to redistribute λ
  - Better convergence
- An improving feasible direction is found by solving:

$$\mathcal{DF}: \quad \text{Maximize} \quad \sum_{\forall(i,j)} \Delta\lambda_{ij}\hat{D}_{ij}$$

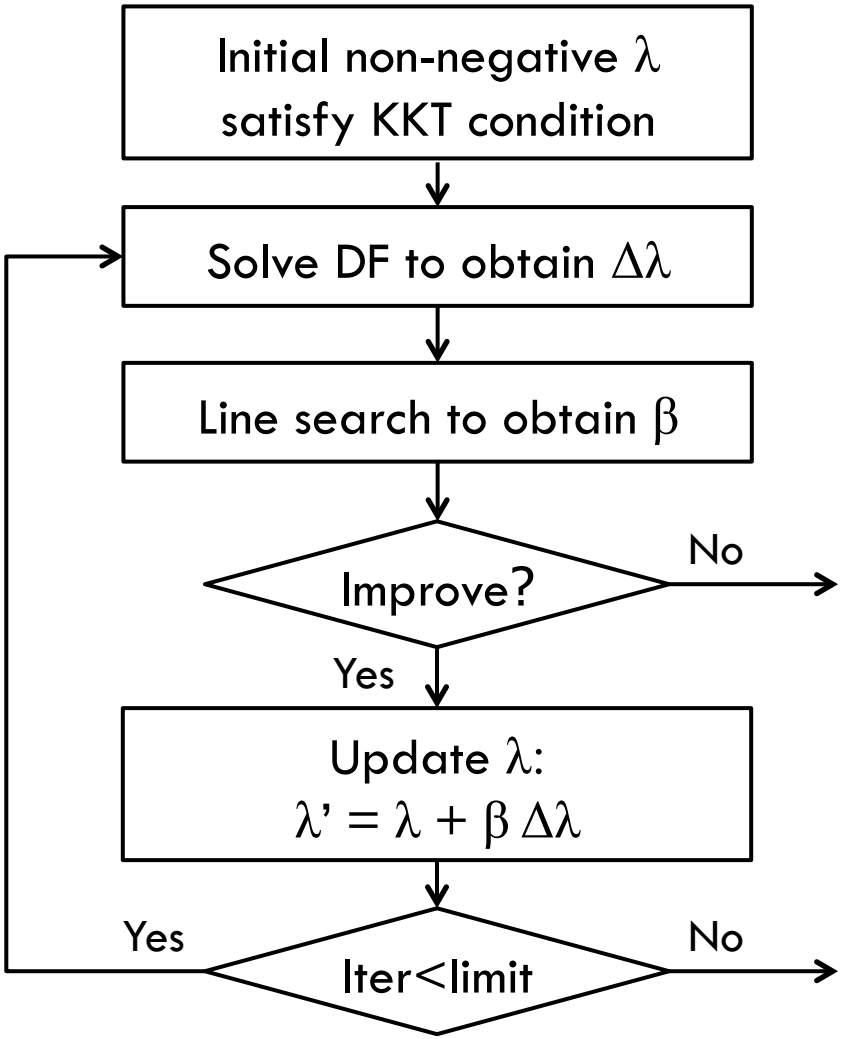
$$\text{Subject to} \quad \boldsymbol{\lambda} \geq \mathbf{0},\ \boldsymbol{\lambda} \in \mathcal{KKT}$$

$$\max(-u, -\lambda_{ij}) \leq \Delta\lambda_{ij} \leq u$$

- A step size β is found by line search
- At each iteration, improve obj: $q(\boldsymbol{\lambda} + \beta\Delta\boldsymbol{\lambda}) > q(\boldsymbol{\lambda})$

* J. Wang et al., **Gate Sizing by Lagrangian Relaxation Revisited**, TCAD 2009

Solve Lagrangian Dual Problem (2/2)



STA for Asynchronous Circuits (1/2)

- In order to use library based timing model, we need to update the slew value
  - Difficult than synchronous circuits due to its internal loops

Algorithm 4 Iterative Slew Update

Ensure: A tight upper bound of the output slew for all the gates;

1: Initialize the output slew to 0 for all the gates;
2: Insert all the gates into a set $\mathcal{G}$;
3: **while** $\mathcal{G} \neq \emptyset$ **do**
4: Pick one gate g_i from $\mathcal{G}$. Let its current output slew be s_{old};
5: Compute new output slew s_{new} of g_i based on its input slew;
6: **if** $s_{new} > s_{old}$ **then**
7: Update the output slew of g_i to s_{new};
8: Insert all gates $\notin \mathcal{G}$ and directly driven by g_i into $\mathcal{G}$;
9: **end if**
10: Remove g_i from set $\mathcal{G}$;
11: **end while**

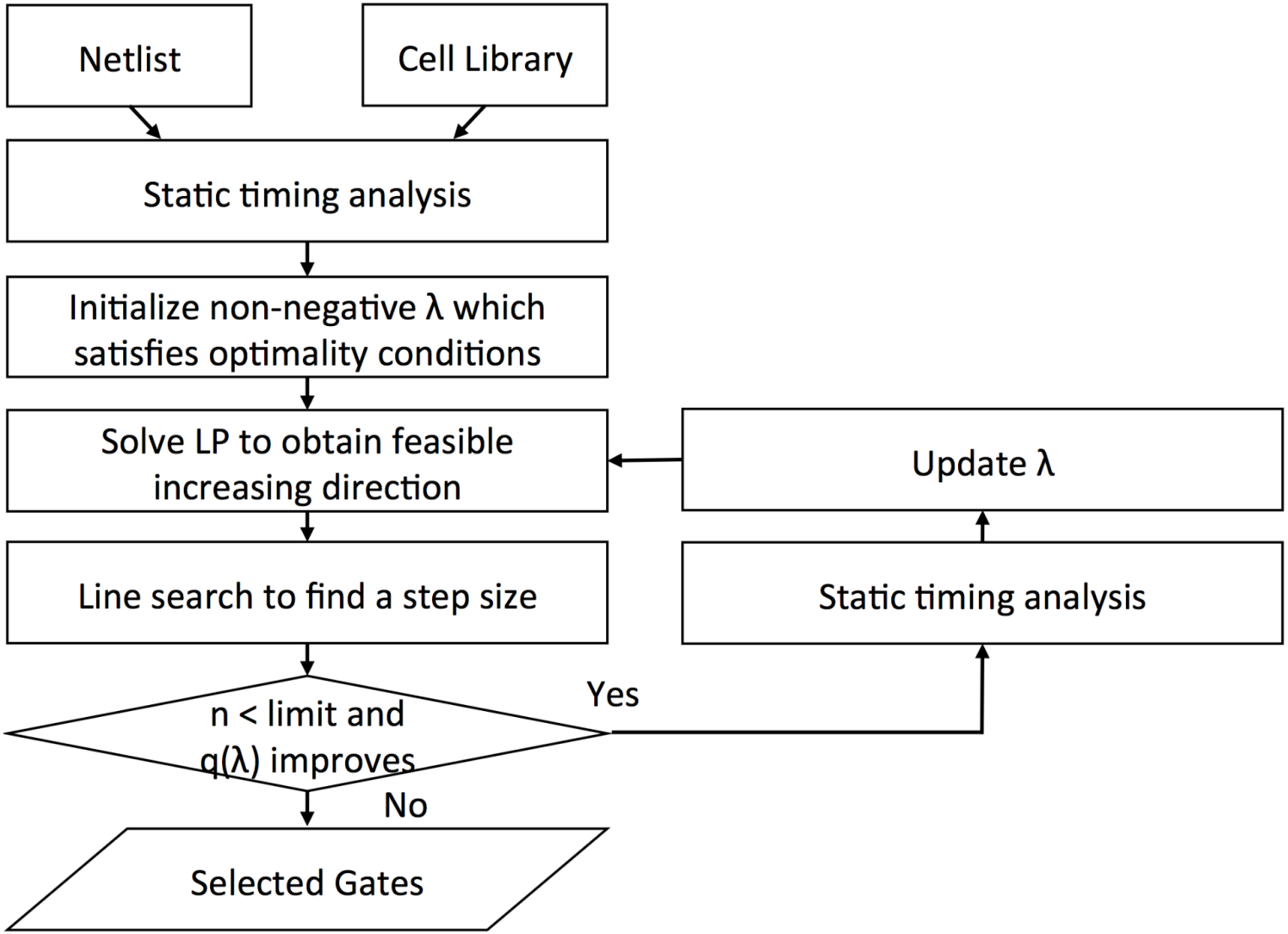
STA for Asynchronous Circuits (2/2)

- The proposed algorithm always converge and provide a tight upper bound of the output slew value
  - Proved using induction technique
- Update delay by look up table interpolation
- Obtain cycle time by solving LP

$$\text{Minimize} \quad \tau$$

$$\text{Subject to} \quad a_i + D_{ij} - m_{ij}\tau \leq a_j \quad \forall (i,j)$$

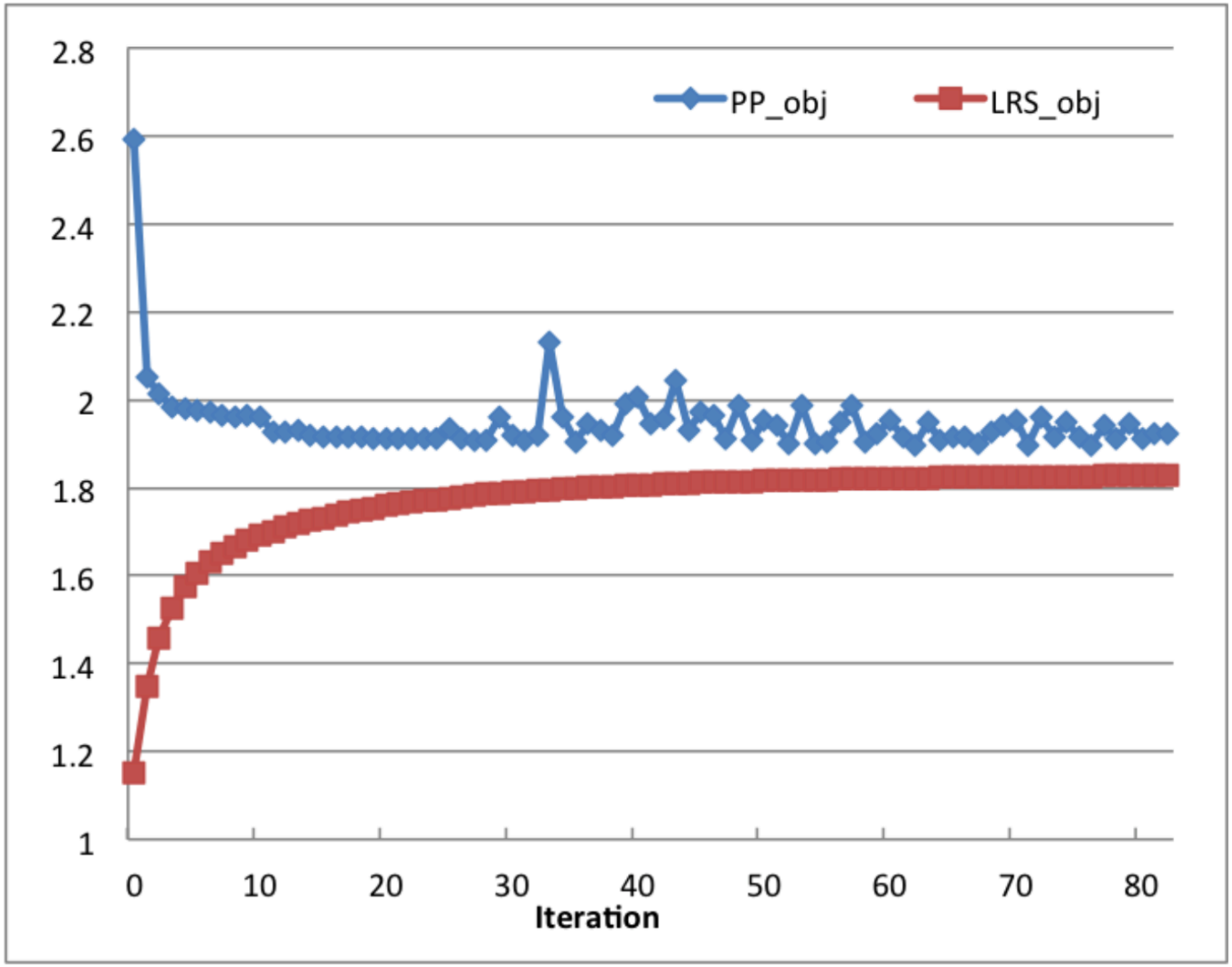
Asynchronous Gate Selection Flow



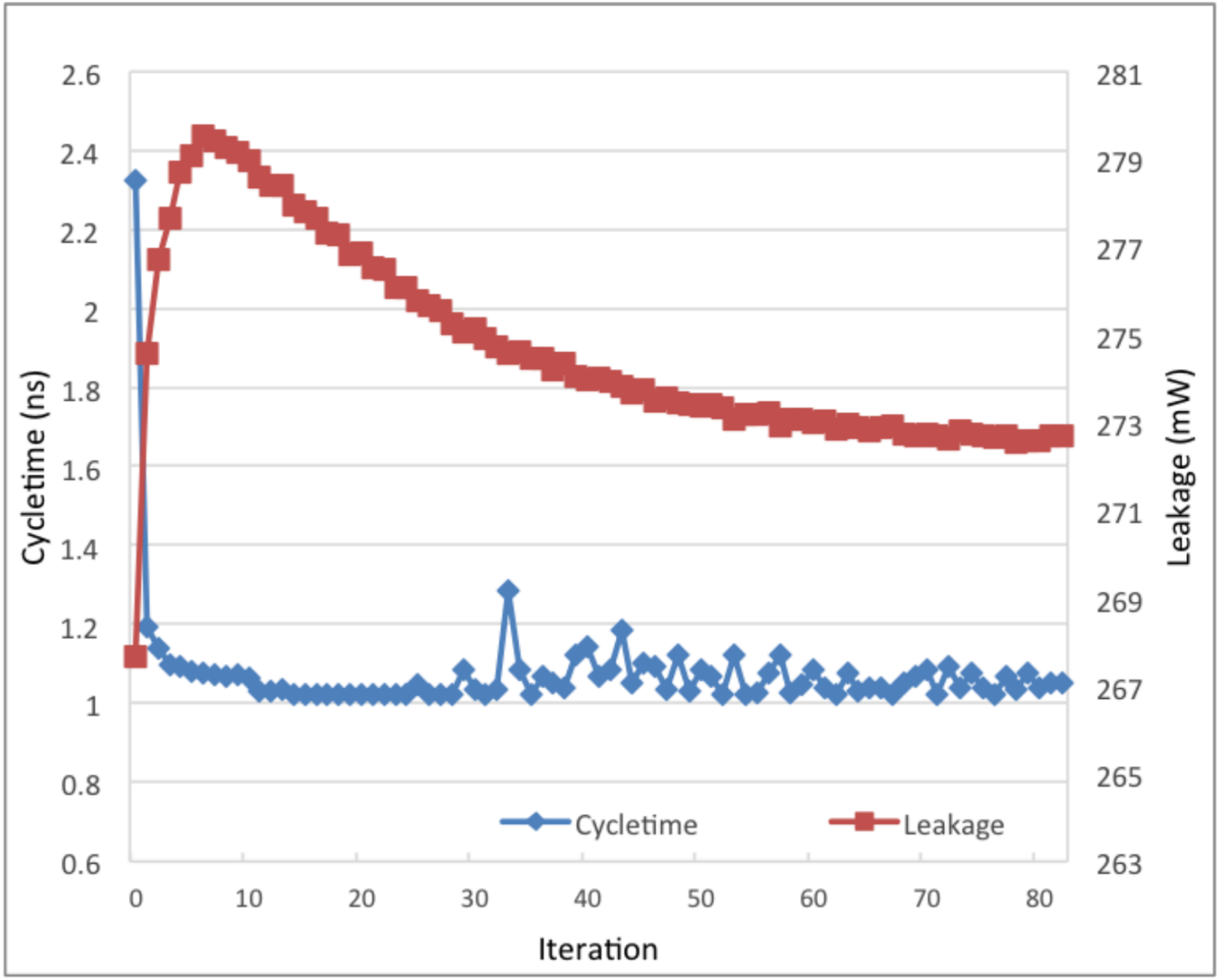
Experiments Setup

- Two standard cell libraries:
  - Proteus standard cell library (single-Vth)
  - Extended library (multi-Vth)
- Two sets of benchmarks:
  - Asynchronous designs transformed from ISCAS89 benchmarks
  - Specific asynchronous designs
  - All the designs are generated using Proteus flow based on the PCHB pipeline template
- No parameter tuning:
  - $\alpha = 2$, *iteration limit* = 50 for all the benchmarks

The Convergence Sequence of s38417



Cycle time/Leakage power of s38417



Comparison on Transformed ISCAS89 Benchmarks

Design	Total Itr.	Cycle Time (ns)				Leakage (uW)			
		Init	Proteus	Ours		Init	Proteus	Ours	
				Single-Vt	Multi-Vt			Single-Vt	Multi-Vt
…	…	…	…	…	…	…	…	…	…
as1196	20	0.78	0.45	0.37	0.39	21147.30	26729.40	22739.80	6596.47
as1488	31	1.96	0.92	0.77	0.81	21038.70	27106.10	22678.30	6611.33
as1238	18	0.77	0.44	0.36	0.39	21815.70	26416.30	23438.60	6483.97
as1423	44	1.94	1.04	0.95	0.95	18684.50	22470.90	19514.60	5901.81
as5378	42	1.19	0.57	0.54	0.55	40032.20	52317.60	42443.10	12548.60
as9234	39	1.69	0.82	0.71	0.70	32586.60	40344.70	33821.10	9963.48
as13207	44	1.68	0.83	0.68	0.70	86949.50	99875.60	90722.80	26414.00
as15850	50	2.44	1.29	0.99	0.96	105889.00	123374.00	107923.00	31622.60
as38417	50	2.32	2.04	1.02	1.01	267671.00	267062.00	273556.00	80887.90
		2.192	**1.213**	1.000	1.024	0.963	**1.095**	1.000	0.295

Comparison on Asynchronous Benchmarks

Design	Total Itr.	Cycle Time (ns)				Leakage (uW)			
		Init	Proteus	Ours		Init	Proteus	Ours	
				Single-Vt	Multi-Vt			Single-Vt	Multi-Vt
ALU8	38	0.68	0.40	0.35	0.38	14849.70	20252.20	15668.10	4223.92
ALU16	16	0.84	0.78	0.39	0.40	41103.60	41017.20	43402.50	12194.50
ALU32	17	1.26	1.18	0.45	0.50	118062.00	118125.00	123546.00	34077.00
ALU64	19	1.38	1.25	0.55	0.59	493119.00	494252.00	520688.00	144837.00
ACC16	27	0.99	0.65	0.50	0.55	7198.16	8565.35	7945.34	2216.16
ACC32	50	1.30	0.73	0.67	0.67	16291.60	22786.80	17217.10	5046.62
ACC64	37	1.13	0.59	0.56	0.57	48672.90	63712.10	51894.60	15904.20
ACC128	50	1.35	0.89	0.69	0.69	125107.00	144956.00	131821.00	39325.50
GCD	35	1.77	1.59	0.82	0.84	7017.06	6967.30	7359.21	2139.78
Fetching U.	40	1.77	0.80	0.74	0.76	75562.70	106374.00	80706.60	24465.00
		2.180	**1.550**	1.000	1.041	0.947	**1.027**	1.000	0.284

Conclusions

- Proposed an effective gate selection algorithm specific for asynchronous circuits
  - Number of constraints is polynomial in circuit size
  - No explicit cycle constraints
  - Simplified LRS is solved effectively using a greedy approach
- Proposed an effective slew update approach for the STA of asynchronous circuits
  - Guarantees convergence and provide a tight bound
- Achieved promising experimental results

Questions?