Switching Activity Driven Gate Sizing and Vth Assignment for Low Power Design

Yu-Hui Huang

Po-Yuan Chen

TingTing Hwang

National Tsing Hua University
Taiwan

Outline

- Introduction and Motivation
- Related Work
- Algorithm
- Experimental Result
- Conclusion

Introduction

- Power = α • Active Power +
  (1 - α) • Idle Power
- Active Power
  - Dynamic power
    - Gate sizing
  - Leakage power
    - Vth Re-assignment
- Idle Power
  - Leakage power
- Minimize total power

Motivation

- To enhance the performance of a circuit, we can size-up gates or replace the Vth of gates from high to low.
  - Size-up :

    increase dynamic power and small leakage power
  - Replace the Vth of cells from high to low :

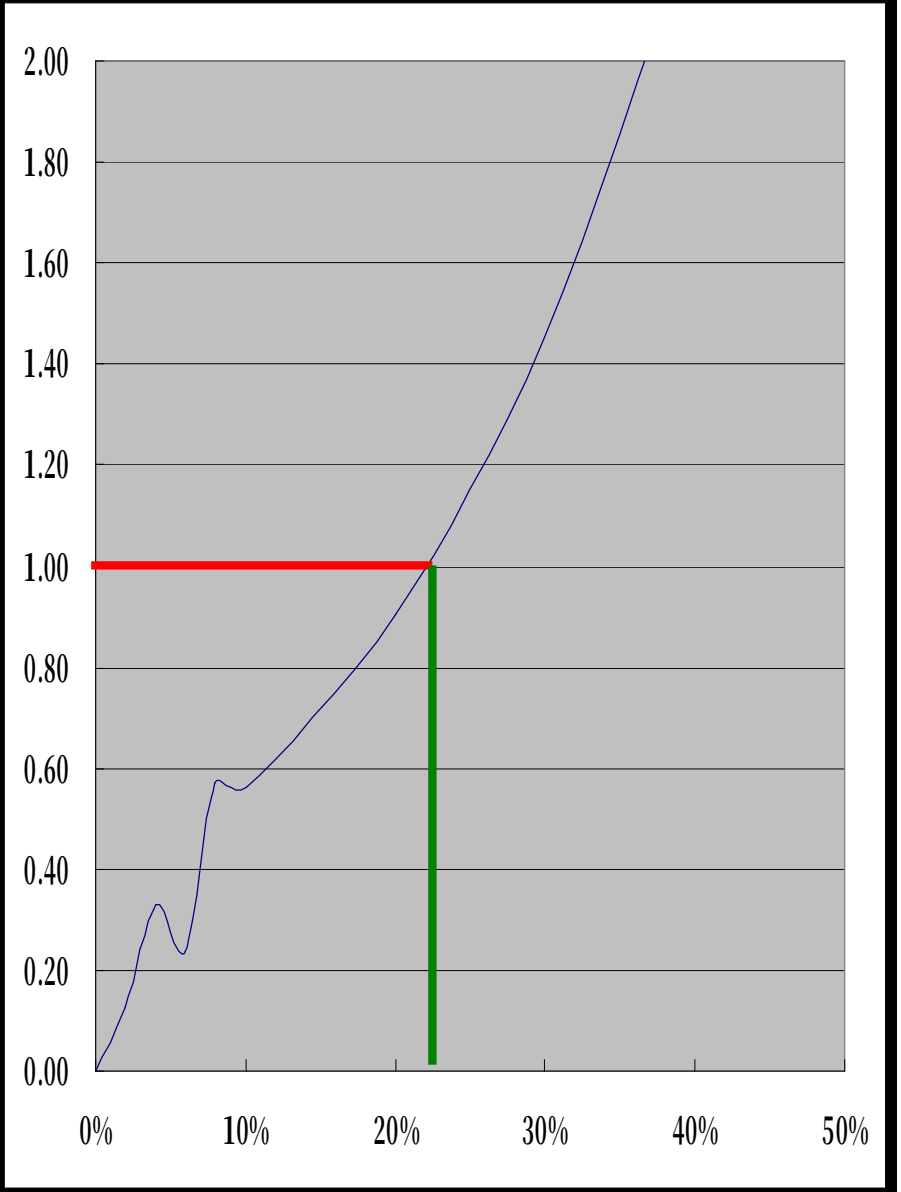
    Increase leakage power
- Which one is better?
  - Depends on the switching activity of a gate.

Motivation

	Inverter A	Inverter B
Vth	_	lower
Size	larger	_

- Inverter A and B have same delay and output loading
- Comparison function

$$\frac{dyn(A) - dyn(B)}{lea(B) - lea(A)}$$



Motivation

- Switching Activity
  - Gate Sizing
  - Vth re-assignment

Switching activity (α)	Ratio (%)						
	TOP	MAC	AVG	GCC	RSA	AES	Average
0 % < α <22%	71.0	48.9	70.9	55.3	84.5	60.8	65.3
22 % < α	29.0	51.1	29.1	44.7	15.5	39.2	34.7

Outline

- Introduction and Motivation
- Related Work
- Algorithm
- Experimental Results
- Conclusions

Related Work

- Previous work focused on minimizing power on non-critical path.
- We can minimize power both on critical path and non-critical path.
  - On critical path:

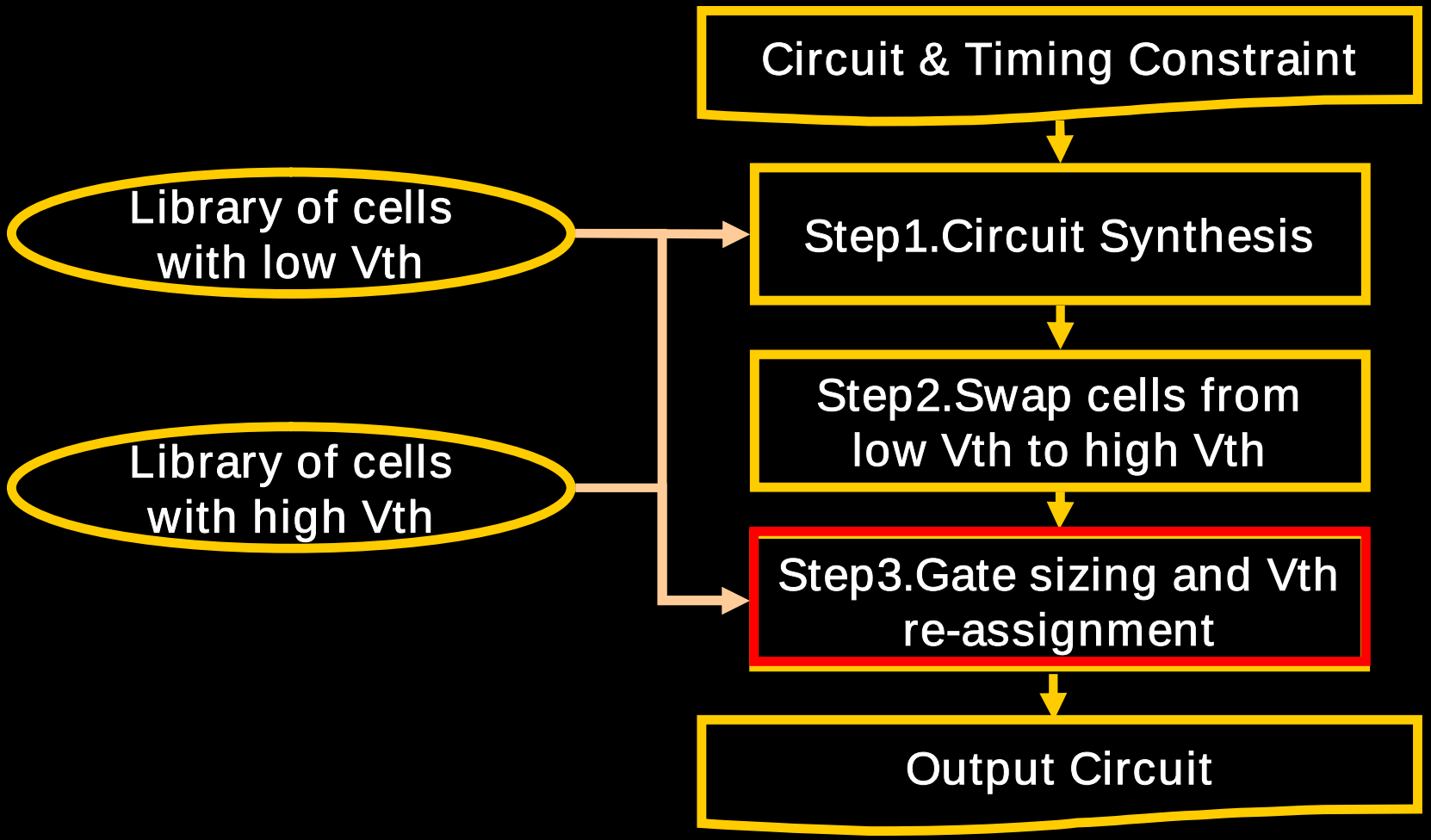
    We can re-assign Vth to high and up-size gates which has small switching activity.
  - On non-critical path:

    Slack can be used to down-size gate or assign Vth to high.

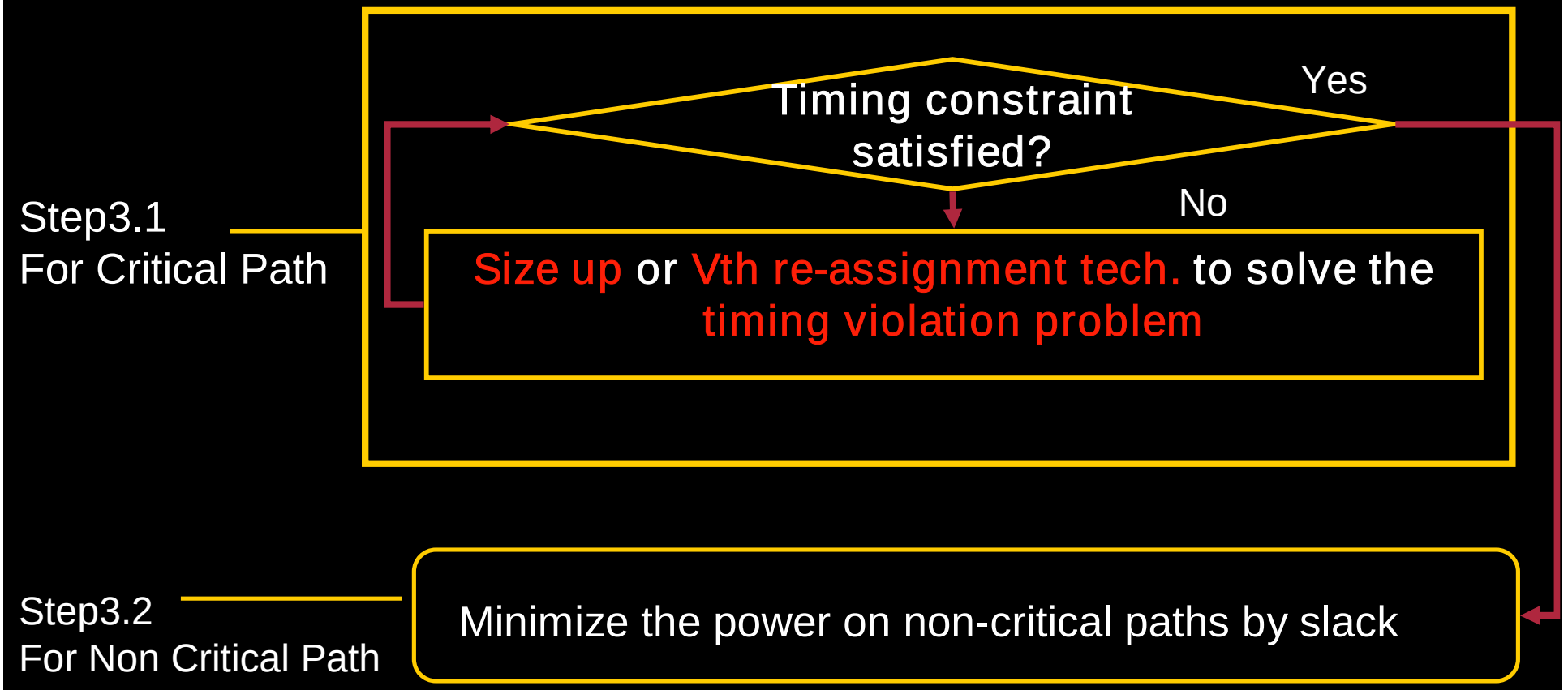
Outline

- Introduction and Motivation
- Related Work
- Algorithm
- Experimental Result
- Conclusion

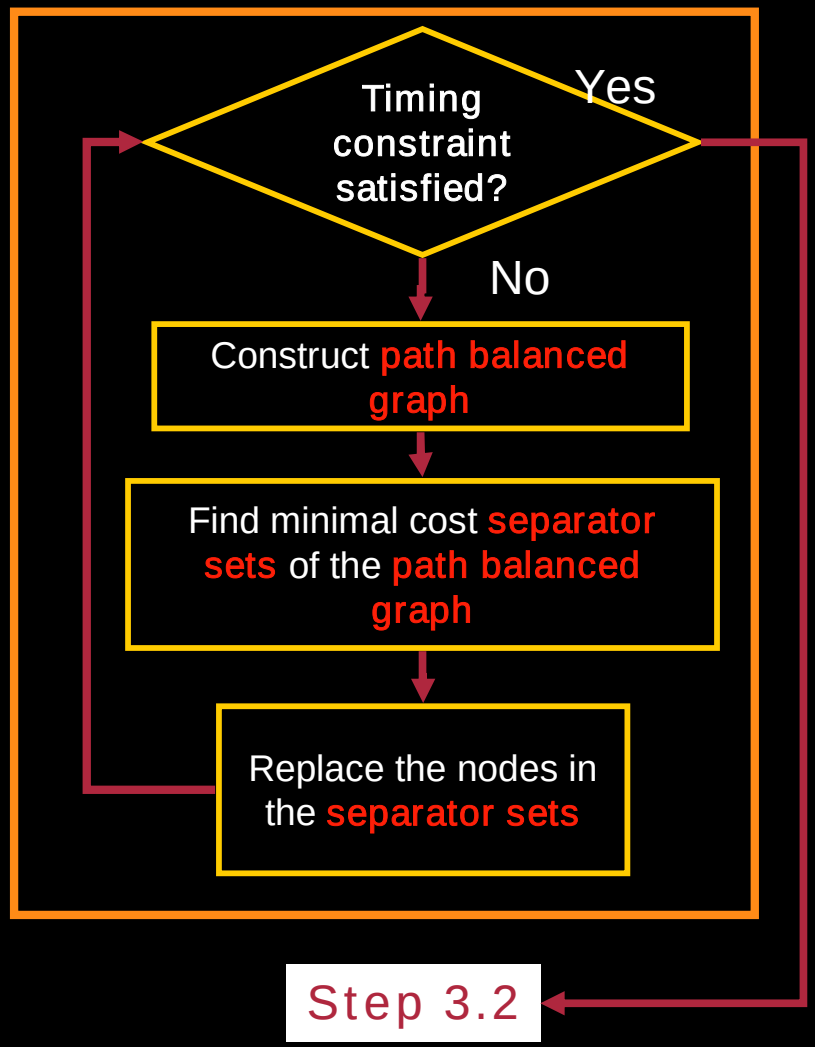
Algorithm –Design Flow



Algorithm –Step 3



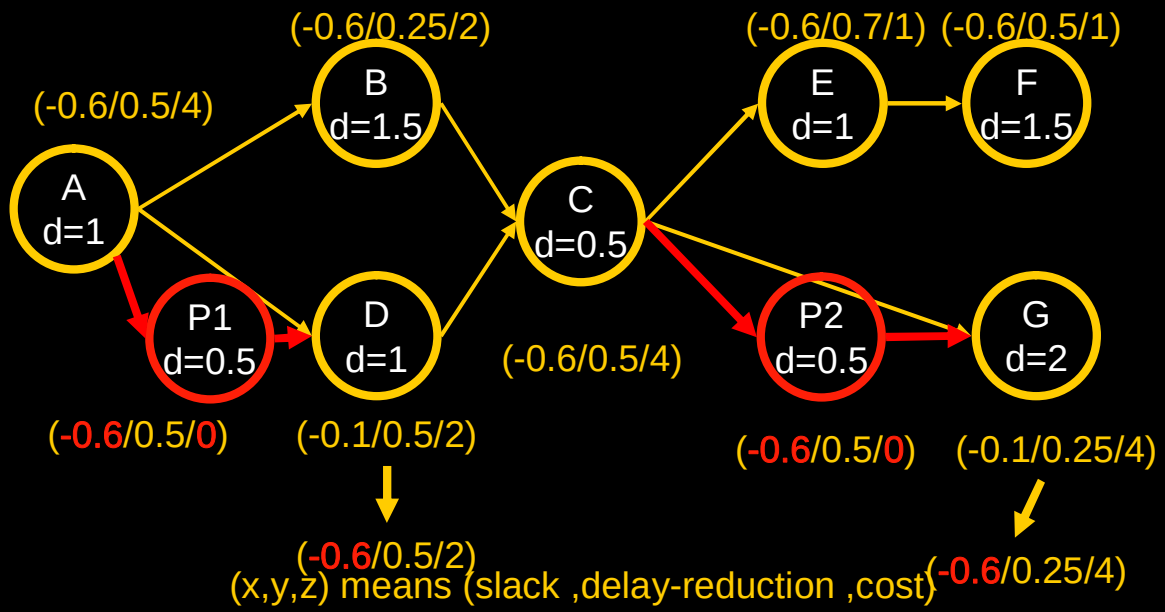
Algorithm for Critical Path (Step 3.1)



Step 3.1: Constructing Path Balanced Graph

- Path-Balanced Graph
  - Yutaka Tamiya, "Performance Optimization Using Separator Sets", ICCAD 1999

$$ds(e) = slack(head\_node(e)) - slack(tail\_node(e))$$



Step 3.1: Computing Cost

- Set cost of each node

$$\text{cost}(g) = \gamma * \text{penalty}(g) + \delta * \text{delay\_reduction}(g)$$

$$\text{penalty}(g) = \alpha * \text{p\_penalty}(g) + \beta * \text{a\_penalty}(g)$$

$$\text{p\_penalty}(g) = \text{per} * \left(\sum_{j \in \text{fanin}(g)} E(j) * C_{inc}(g) * V^2 + \text{leak}_{inc}(g) \right) + (1 - \text{per}) * \text{leak}_{inc}(g)$$

Active mode

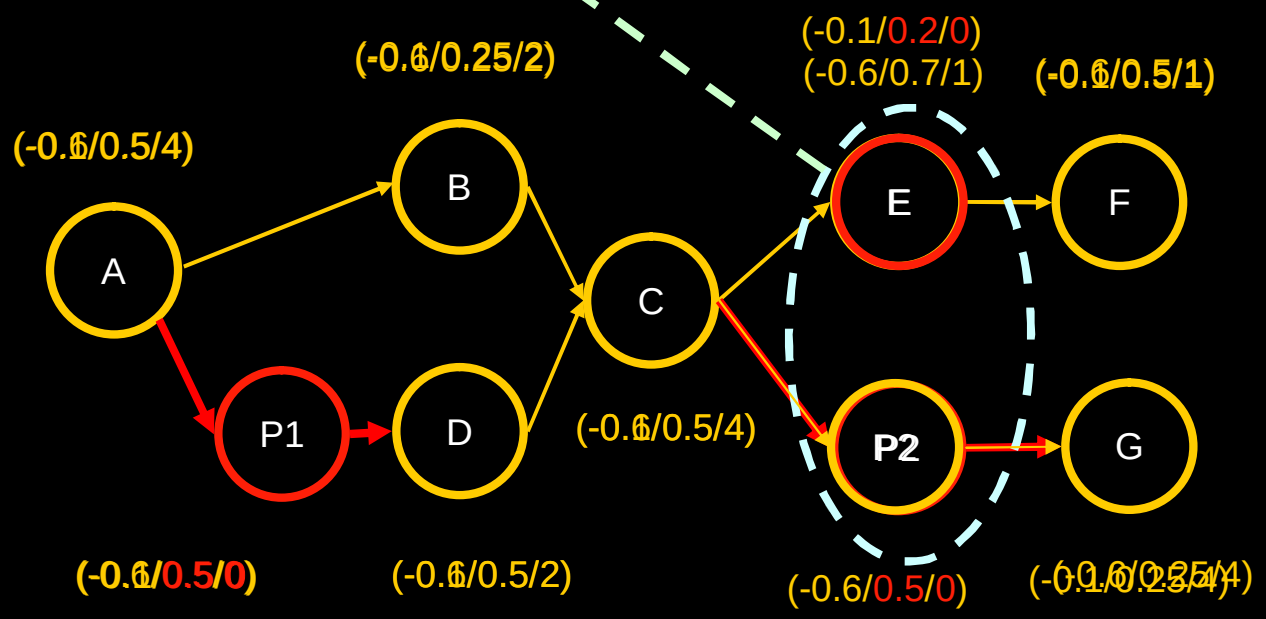
Idle mode

E(j) is the transition density of node j

Step 3.1: Finding Separator Set

- Find separator set of minimal cost in the graph

Delay improvement min{0.7,0.5} = 0.5

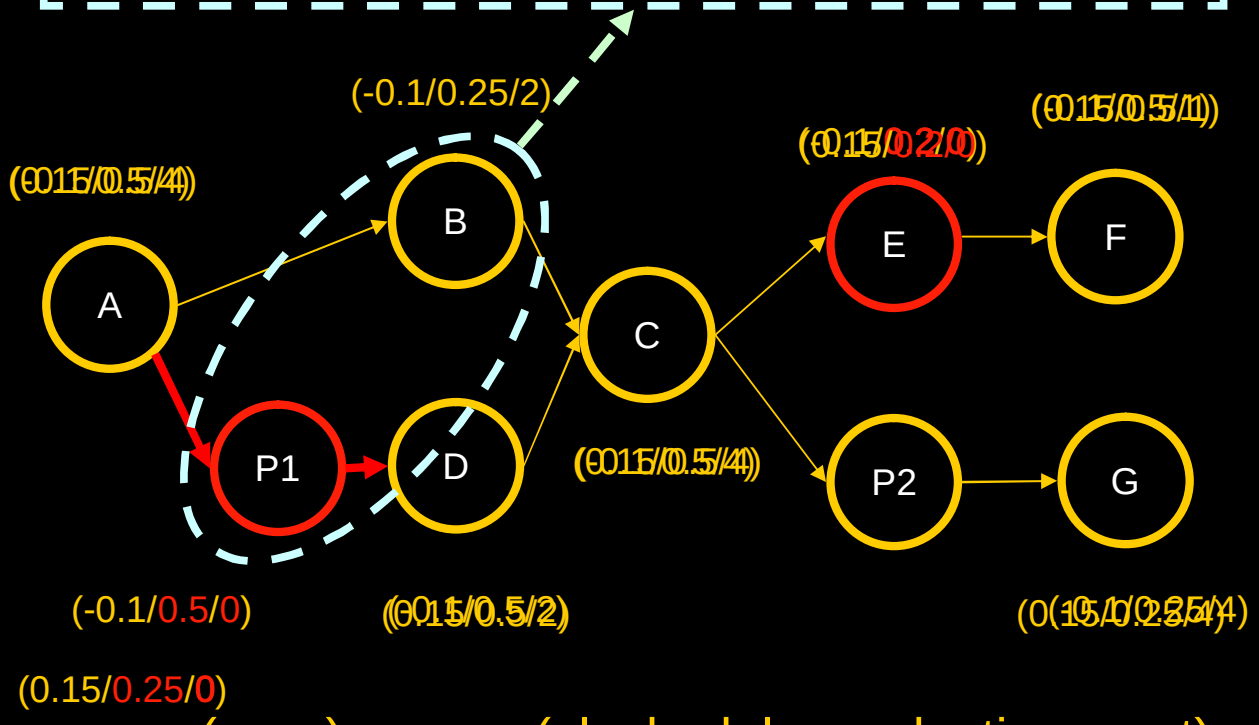


(x,y,z) means (slack ,delay-reduction ,cost)

Step 3.1: Finding Separator Set

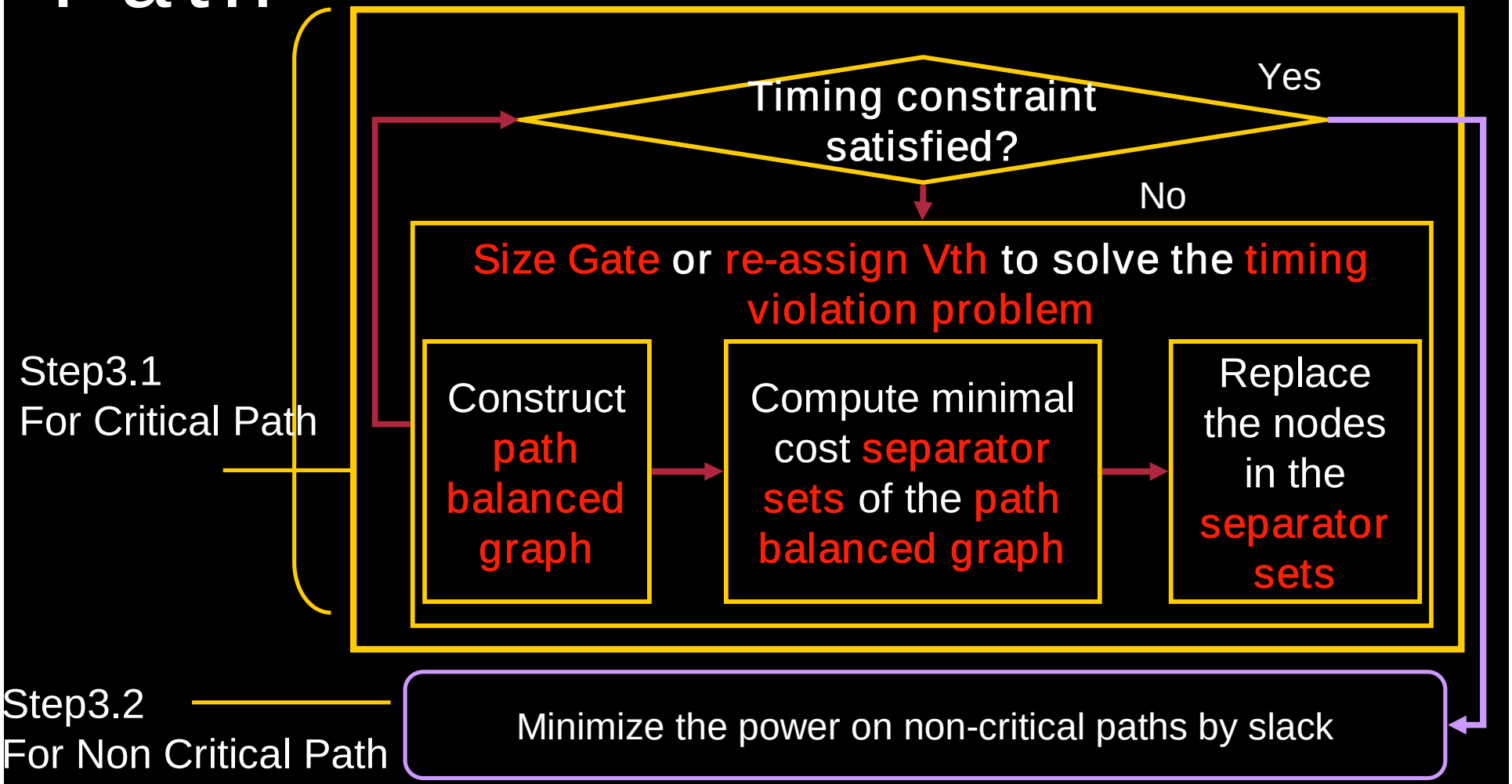
- Find separator set of minimal cost in the graph

Delay improvement min{0.5,0.25} = 0.25



(x,y,z) means (slack ,delay-reduction ,cost)

Algorithm for Non Critical Path



Algorithm for Non Critical Path

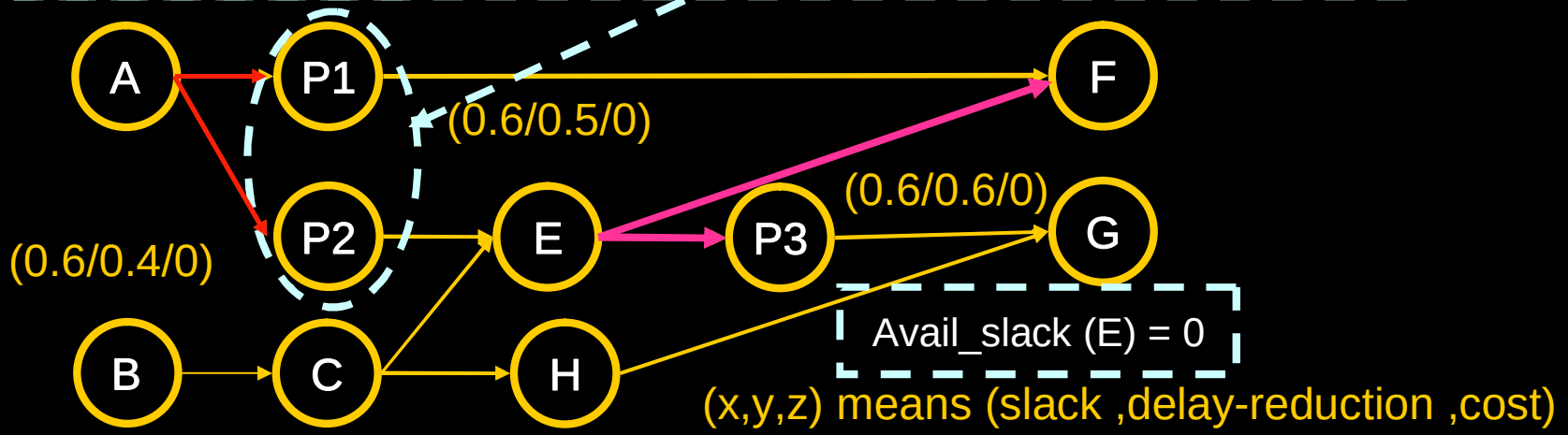
Compute available slack

Compute delay penalty caused by down-sizing or re- assigning Vth to high

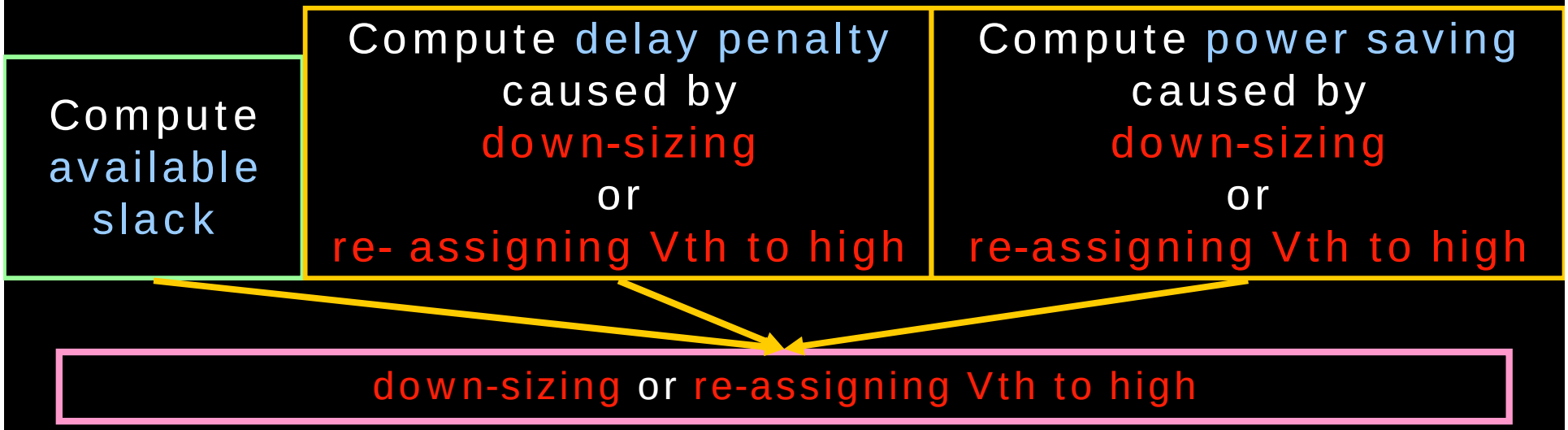
Compute power saving caused by down-sizing or re-assigning Vth to high

down-sizing or re-assigning high Vth

Avail_slack (A) = Minimal delay reduction of {P1,P2} = min{0.5,0.4} = 0.4



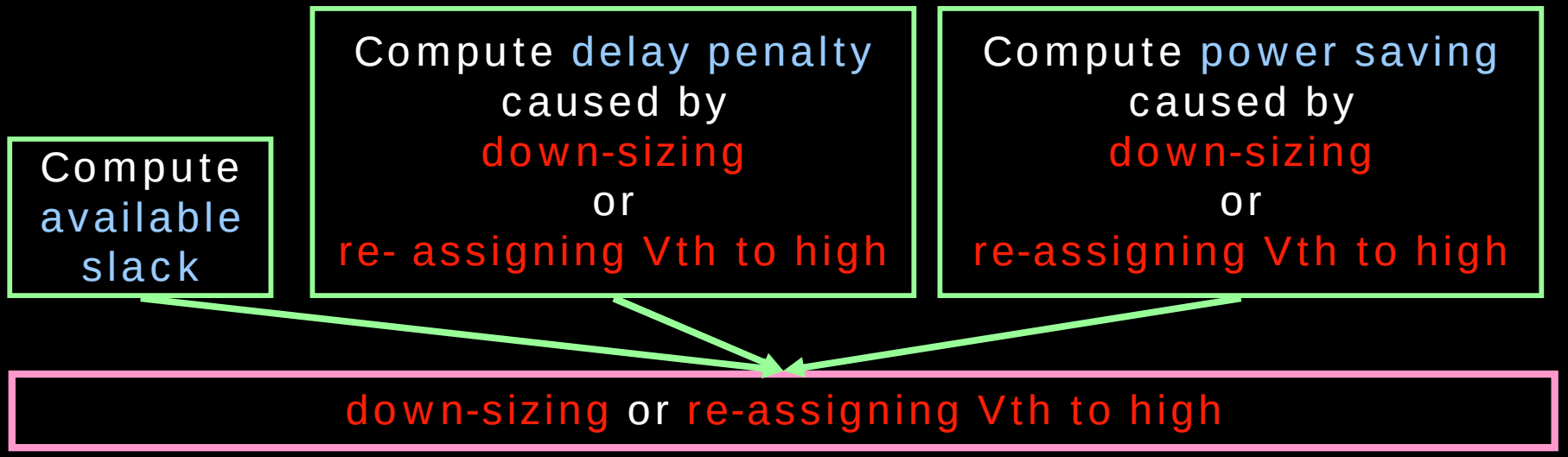
Step 3.2: Computing Penalty



$$delay\text{-}penalty(g) = Delay(new\_g) - Delay(g)$$

$$p\_saving(g) = p\_penalty(g) - p\_penalty(new\_g)$$

Step 3.2: Replacing Cell



- If only one delay penalty of two options is less than available slack ?
  - ➡ choose the available one
- If delay penalties of both options are less than available slack ?
  - ➡ depends on the larger power saving

Outline

- Introduction and Motivation
- Related Work
- Algorithm
- Experimental Result
- Conclusion

Experimental Result

- Benchmarks

Cir.	Cell Count	Characteristics
TOP	463	An Alarm Clock
MAC	2425	Multiplier and Accumulator
AVG	6361	Average Number Calculator
GCC	8204	Gravity Center Calculator
RSA	14815	Asymmetric Crypto-Processor
AES	16824	Advanced Encryption Core

Experimental Result

- TOOLS
  - *DesignCompiler*
  - *TSMC 0.13um* library
  - *PrimeTime*
  - *PrimePower*

Experimental Result

- Power saving

α is the fraction of active time

Circuit	α = 100%		α = 50%		α = 10%	
	P(mW)	*Red*	*P*(mW)	*Red*	*P*(mW)	*Red*
TOP	0.363	11.95%	0.179	14.24%	0.0371	19.12%
MAC	0.790	18.56%	0.397	21.09%	0.0837	35.97%
AVG	1.65	5.75%	0.835	8.79%	0.211	14.84%
GCC	0.753	6.48%	0.412	8.69%	0.142	15.46%
RSA	2.12	39.20%	1.08	41.40%	0.239	53.50%
AES	13.4	15.60%	6.70	16.99%	1.41	21.33%
Average		16.26%		18.53%		26.70%

Experimental Result

- Time Penalty

α is the fraction of active time

Circuit	Original T	α = 100%		α = 50%		α = 10%	
		T'	T penalty	T'	T penalty	T'	T penalty
TOP	1.43	1.37	-4.2%	1.39	-2.8%	1.40	-2.1%
MAC	3.30	3.33	0.8%	3.33	0.8%	3.33	0.8%
AVG	23.78	23.13	-2.7%	23.46	-1.3%	23.54	-1.0%
GCC	26.30	26.65	1.3%	26.73	1.6%	26.34	0.2%
RSA	10.00	10.08	0.8%	10.03	0.3%	10.10	1.0%
AES	2.29	2.21	-3.5%	2.27	-0.9%	2.27	-0.9%

Outline

- Introduction and Motivation
- Related Work
- Algorithm
- Experimental Result
- Conclusion

Conclusion

- Switching activity of a gate plays an important role in making decision to choose gate sizing or Vth assignment.
- Under the timing constraint, our circuit have 16% and 18% improvement as compared to the original circuits where the fraction of active time are 100% and 50%, respectively.

Thank you