Combined Use of Rising and Falling Edge Triggered Clocks for Peak Current Reduction in IP-Based SoC Designs

Shi-Yi Huang

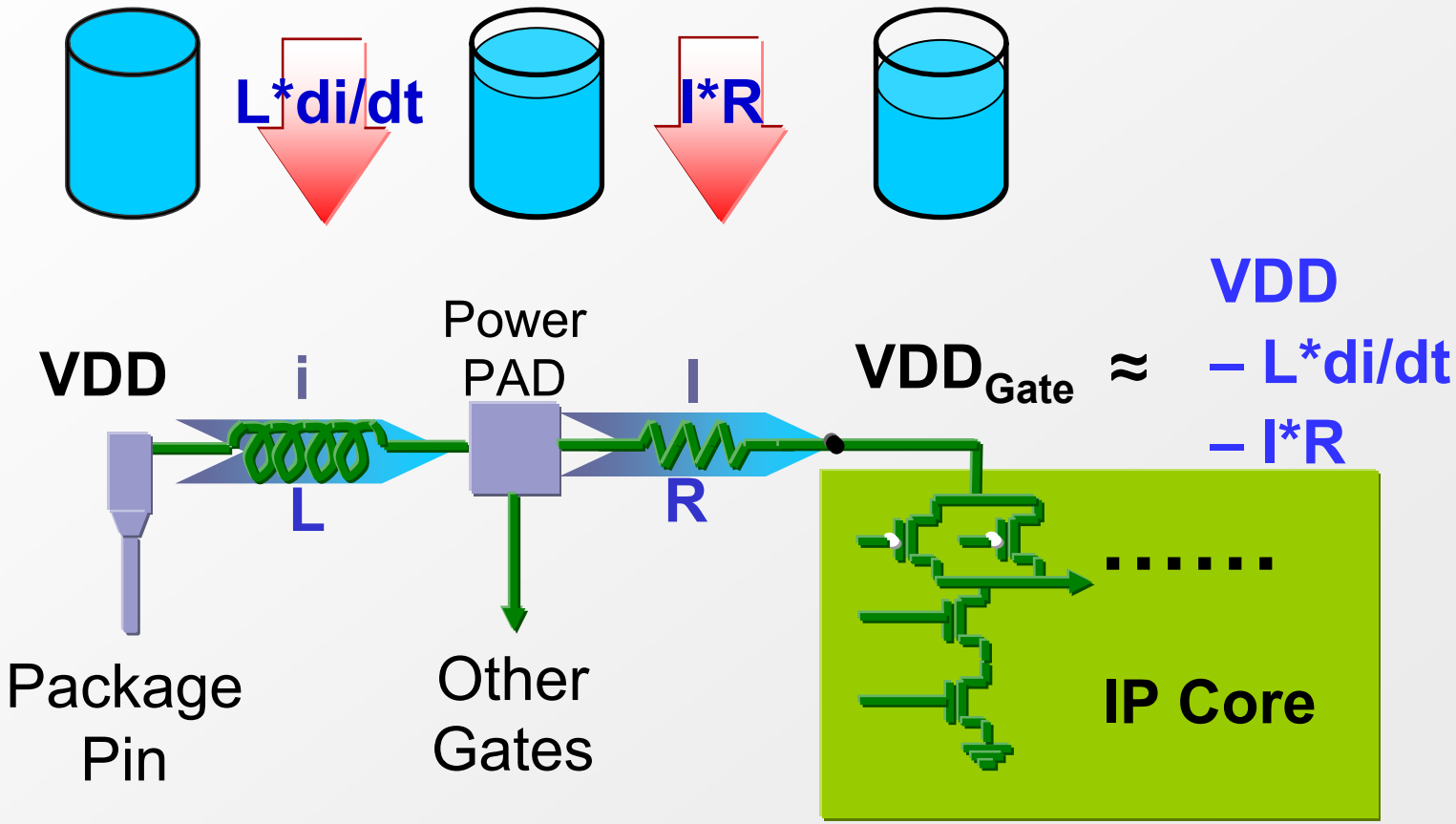
Department of Electronic Engineering
National Changhua University of Education
Taiwan

Outline

- Introduction
- Practicability
- Problem Definition
- Algorithms
- Experiments
- Conclusions

Introduction

- **IR Drop (Voltage Drop)**



Introduction

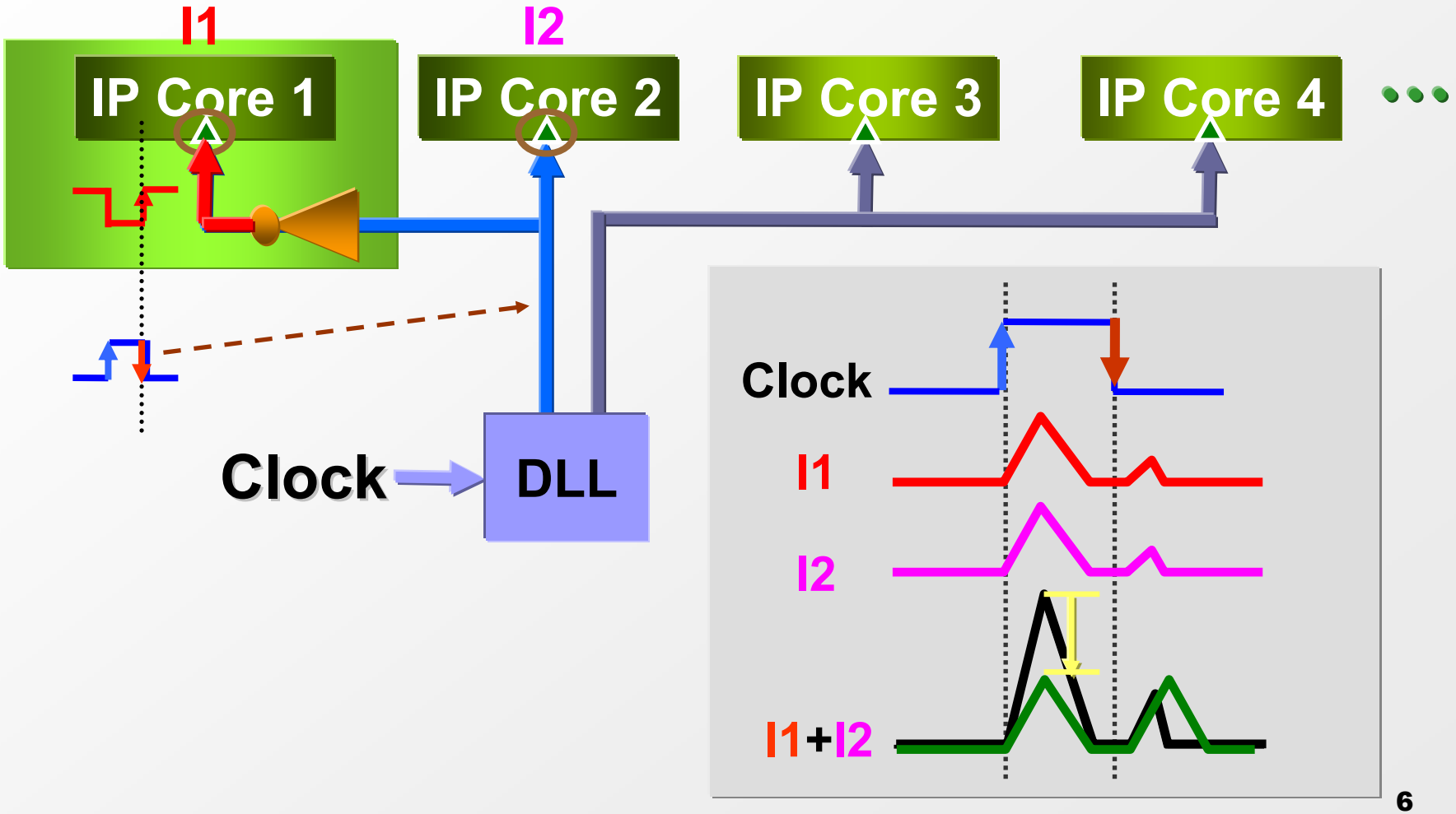
- **IR Drop (Voltage Drop)**
  - Influence the **Performance** of a Circuit
  - Even Cause **Logic Errors**
- **Straightforward Solutions for IR Drop**
  - Reduce **R** (Resistance)
  - Reduce **I** (Current)
    - **Especially** Peak Current
    - Our work reduces the peak current

Introduction

- **Related Works for Peak Current Reduction**
  - Physical Design
    - Clock Skew Scheduling
    - Opposite Phase Clock Scheme
    - Etc.
  - High Level Synthesis
    - Scheduling and Allocation
  - System Level
    - Memory Partition
    - Etc.

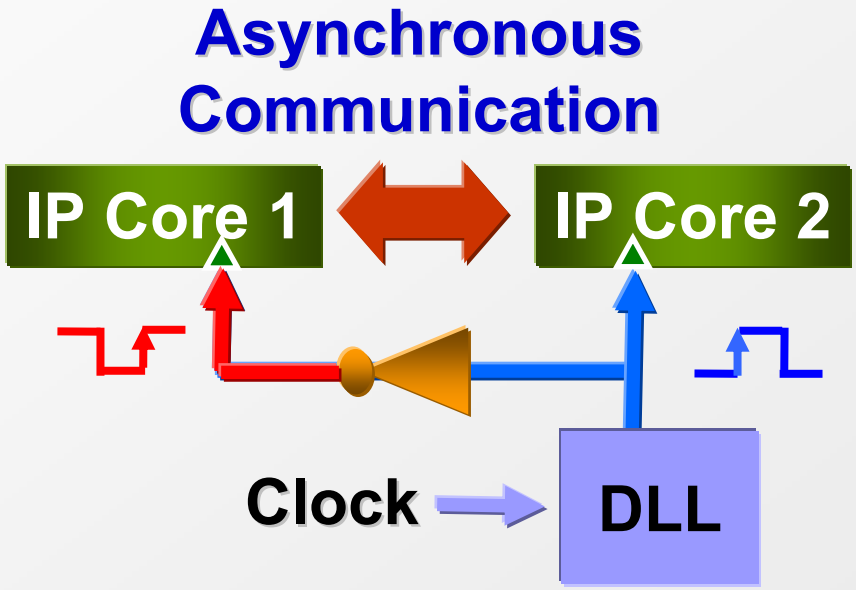
Introduction

- **Our Work**



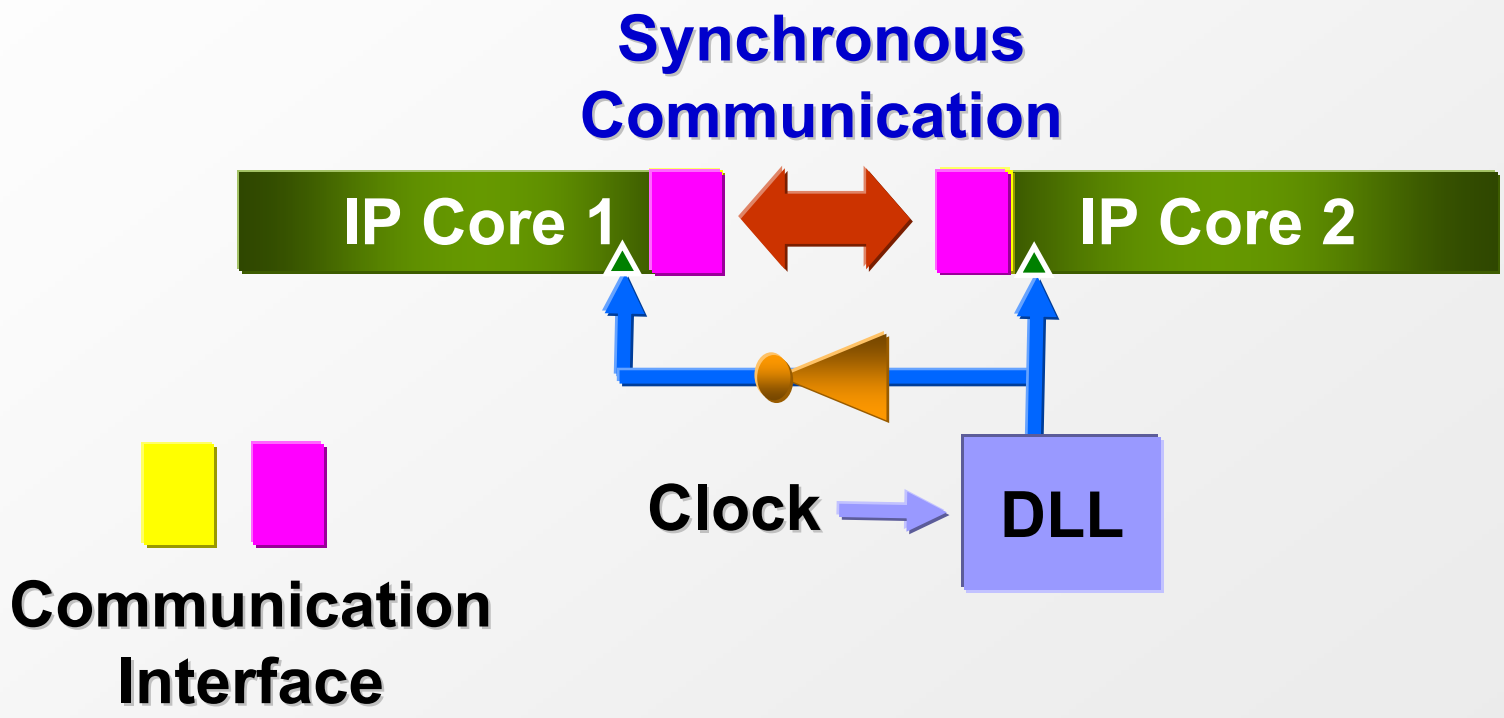
Practicability

- **Asynchronous Communication**
  - Network-on-Chip (NoC)



Practicability

- **Configurable IP Cores**

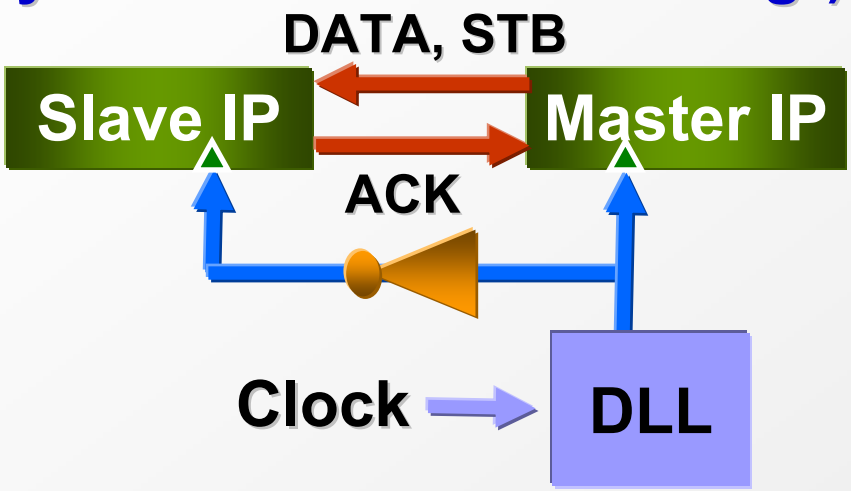


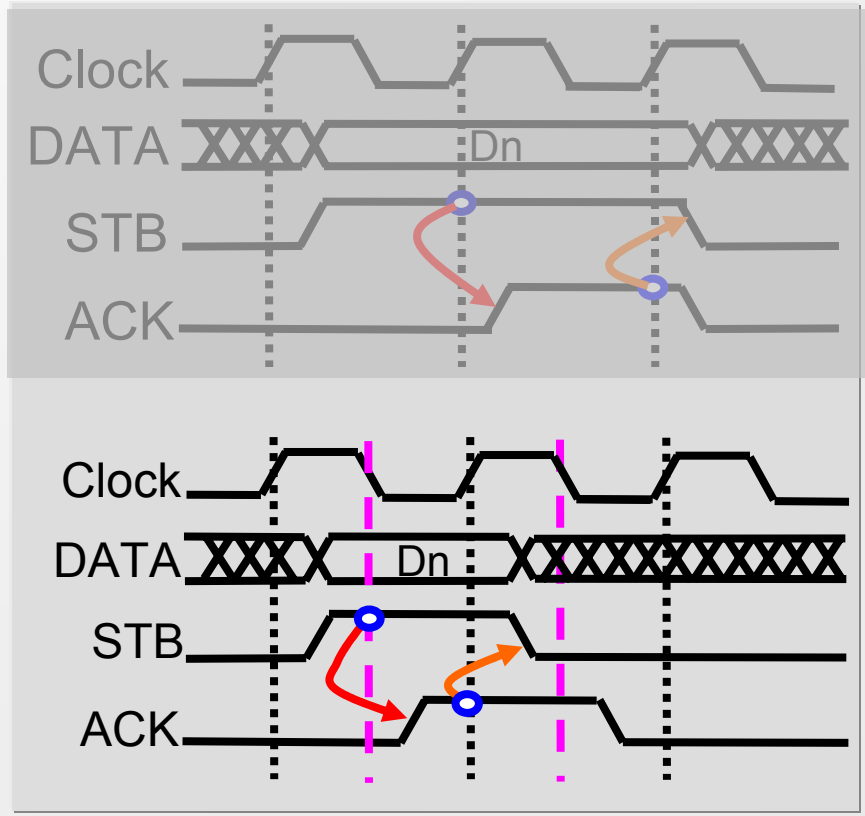
Practicability

- **Robust IP Cores**

No Timing/Function Violation!

WISHBONE Bus
(Synchronous Handshaking*)

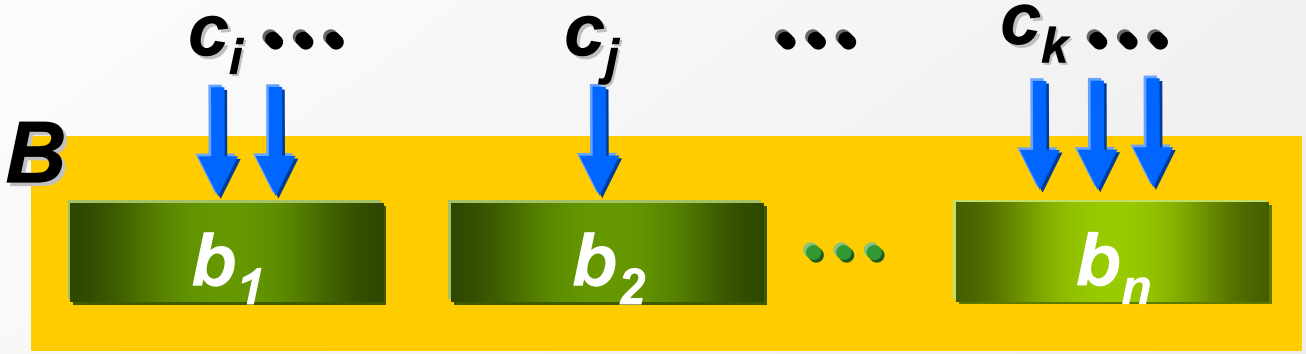


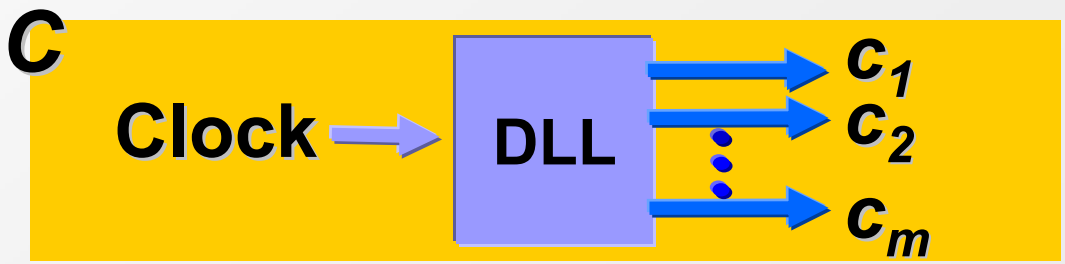


*** WISHBONE Bus,**
Virtual Component Interface,
AMBA, etc.

Problem Definition

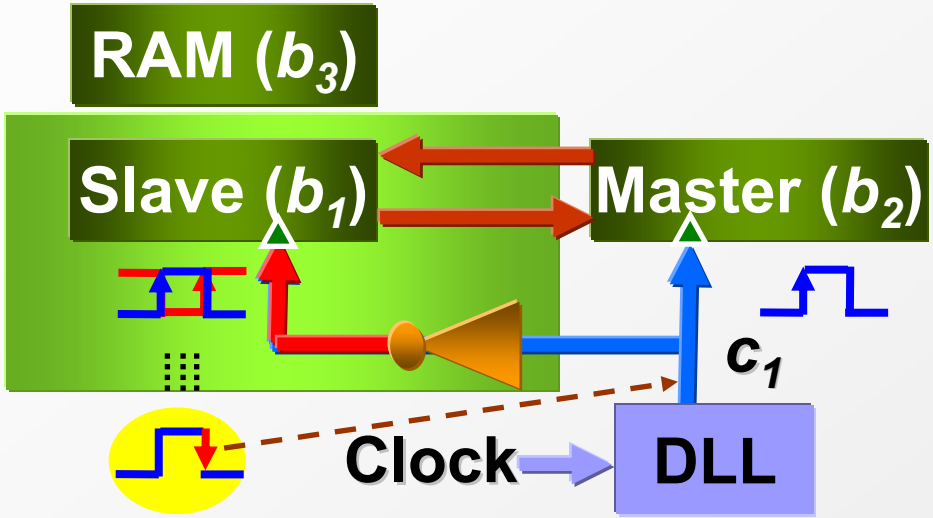
- ***System S***
  - *n Blocks* $B = \{b_1, b_2, \ldots, b_n\}$
  - *m Synchronous* Clock Domains $C = \{c_1, c_2, \ldots, c_m\}$





Problem Definition

- **Functions**
  - Edge-Type-**A**ssignment (ETA) Function f_a
  - Edge-Type-**C**onstraint (ETC) Function f_c



The triggering edge of b_1's clock for c_1 must be fixed to be positive ↔ $f_c(b_1, c_1) = 1$

$f_c(b_1, c_1) = 0$ ↔ *The triggering edge of b_2's clock for c_1 is unconstrained*

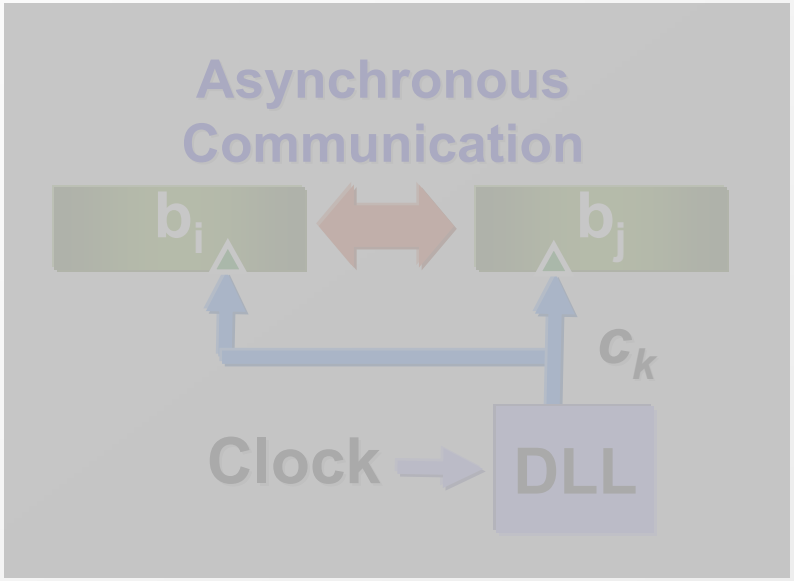
$f_a(b_1,c_1)$ = **-1** (**negative** edge triggered)

$f_a(b_2,c_1) = 1$ (*positive* edge triggered)

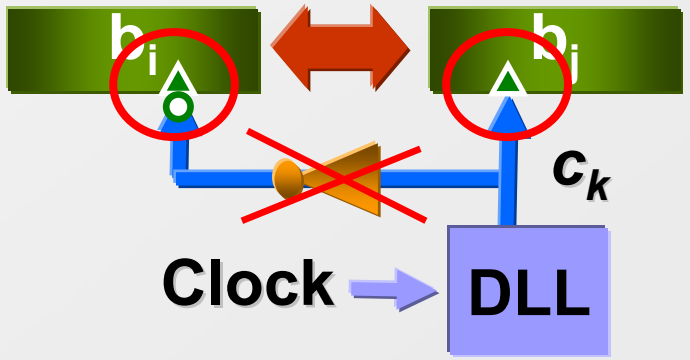
$f_a(b_3,c_1) = 0$ (*without* the clock of c_1)

Problem Definition

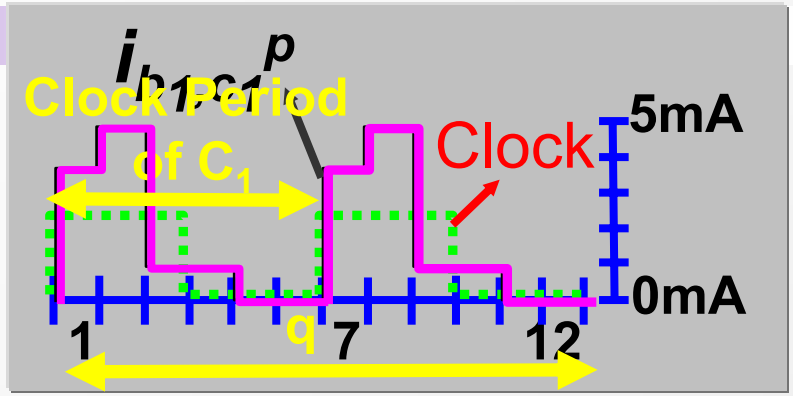
- **Functions**
  - ✶ Edge-Type-**R**elationship (ETR) Function f_r
    - $f_r(b_i, b_j, c_k) = 0$
    - $f_r(b_i, b_j, c_k) = 1$
    - $f_r(b_i, b_j, c_k) = -1$



Critical Timing/Performance Condition, Communication Constraint, etc.



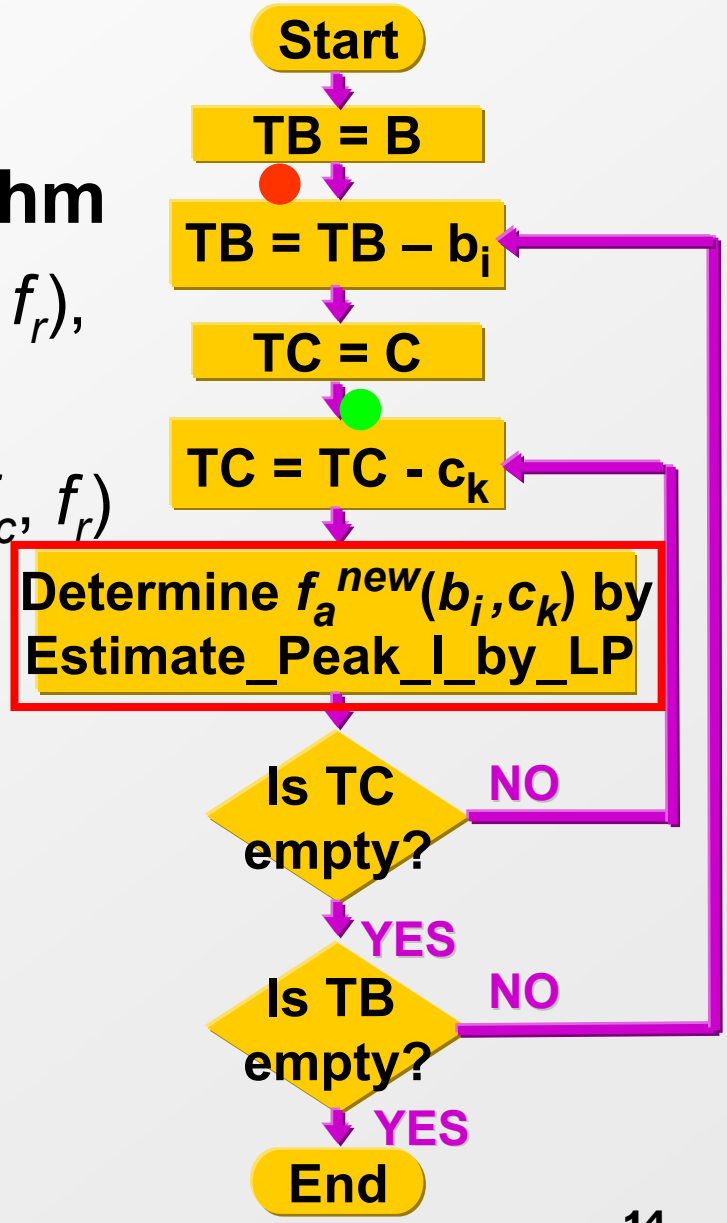
Problem Definition



- **Problem**
  - A given a system $S^{orig} = (B, C, f_a^{orig}, f_c, f_r)$
  - Finding a new ETA Function f_a^{new} such that the ***peak current*** of $S^{new} = (B, C, f_a^{new}, f_c, f_r)$ is **as small as possible**
- $i_{bh,cl}^{p}$, $i_{bh,cl}^{n}$: Peak Current Vector (PCV) of b_h for c_l
  - $i_{bh,cl}^{p} = [v_{1,h,l}^{p}, v_{2,h,l}^{p}, \ldots, v_{q,h,l}^{p}]$
    - **e.g.** $i_{b1,c1}^{p} = [4, 5, 1, \ldots, 0]$
  - **Dummy clock period** ***q*** is the ***least common multiple*** **of the periods of all clock domains**

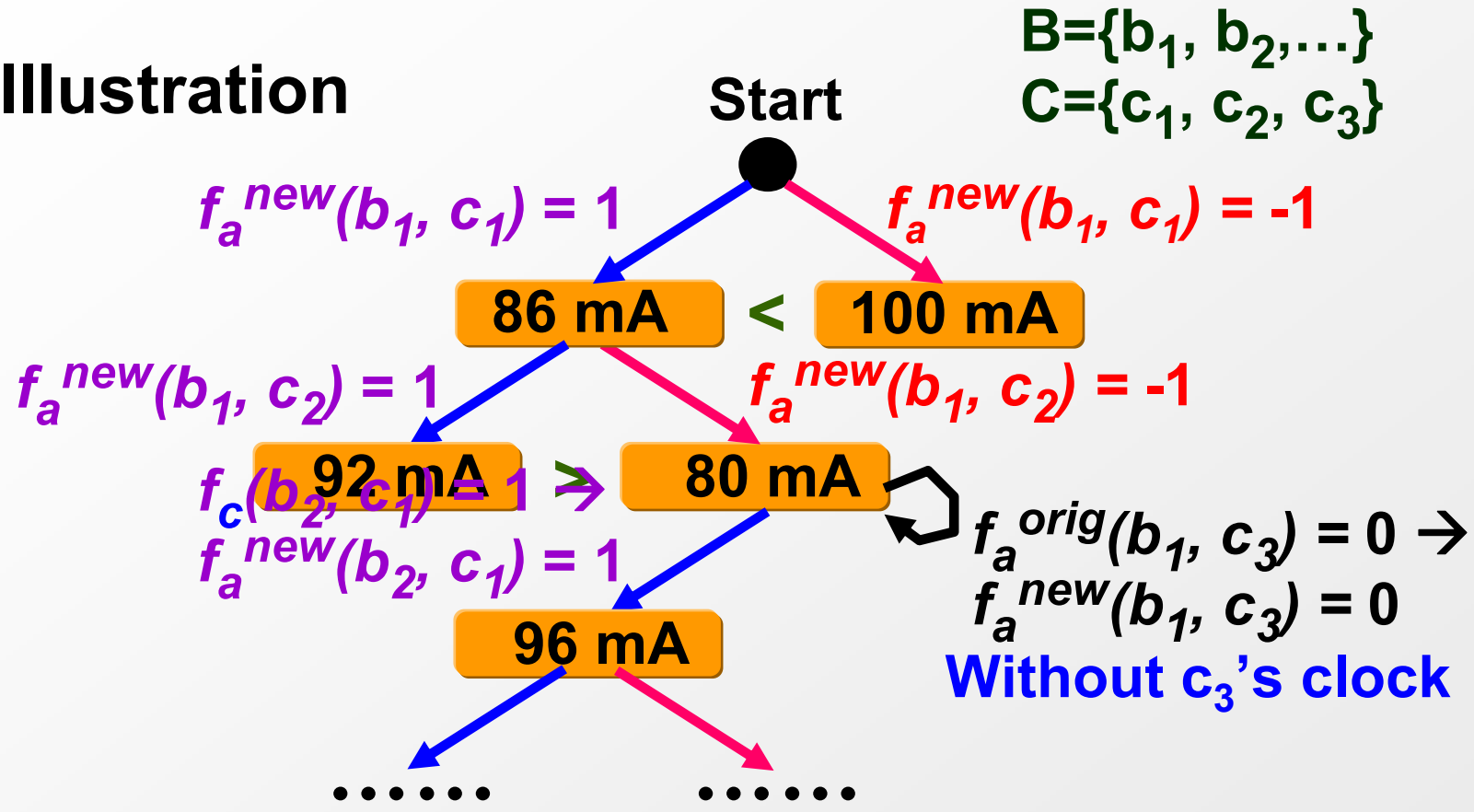
Algorithms

- **Heuristic LP-Based Algorithm**
  - Inputs: $S^{orig} = (B, C, f_a^{orig}, f_c, f_r)$, B's PCV Data
  - Output : $S^{new} = (B, C, f_a^{new}, f_c, f_r)$
  - Estimate_Peak__l_by_LP
    - **Linear Programming**
    - Consider ETA, ETC, ETR Functions



Algorithms

- **Illustration**



: Estimate_Peak_I_by_LP(B, C, f_a^{new}, f_c, f_r, I)

Algorithms

$p_{i,k}$ = 1 (0) if the clock of b_i for c_k is (isn't) positive edge triggered

$n_{i,k}$ = 1 (0) if the clock of b_i for c_k is (isn't) negative edge triggered

- **ILP Model Generator**

$f_a^{orig}(b_i, c_j)$

	c_1	c_2
b_1	1	1
b_2	1	0
b_3	1	0

$f_c(b_i, c_j)$

	c_1	c_2
b_1	1	1
b_2	0	0
b_3	0	0

$f_r(b_i, b_j, c_k)$

		c_1	c_2
b_1	b_2	1	–
b_1	b_3	0	–
b_2	b_3	0	–

Minimize *Peak_I*(S)

Subject to:

:

$p_{1,1} + n_{1,1} = 1$ $p_{2,1} + n_{2,1} = 1$ $p_{3,1} + n_{3,1} = 1$

$p_{1,2} + n_{1,2} = 1$ $p_{2,2} + n_{2,2} = 0$ $p_{3,2} + n_{3,2} = 0$

$p_{1,1} = 1$ $p_{1,2} = 1$

$p_{1,1} + n_{2,1} = 1$ $n_{1,1} + p_{2,1} = 1$

$Peak\_I(S) \geqq 4p_{1,1}+4p_{2,1}+4p_{3,1}+3p_{1,2}+n_{1,1}+n_{2,1}+n_{3,1}$

: $+ n_{1,2}$

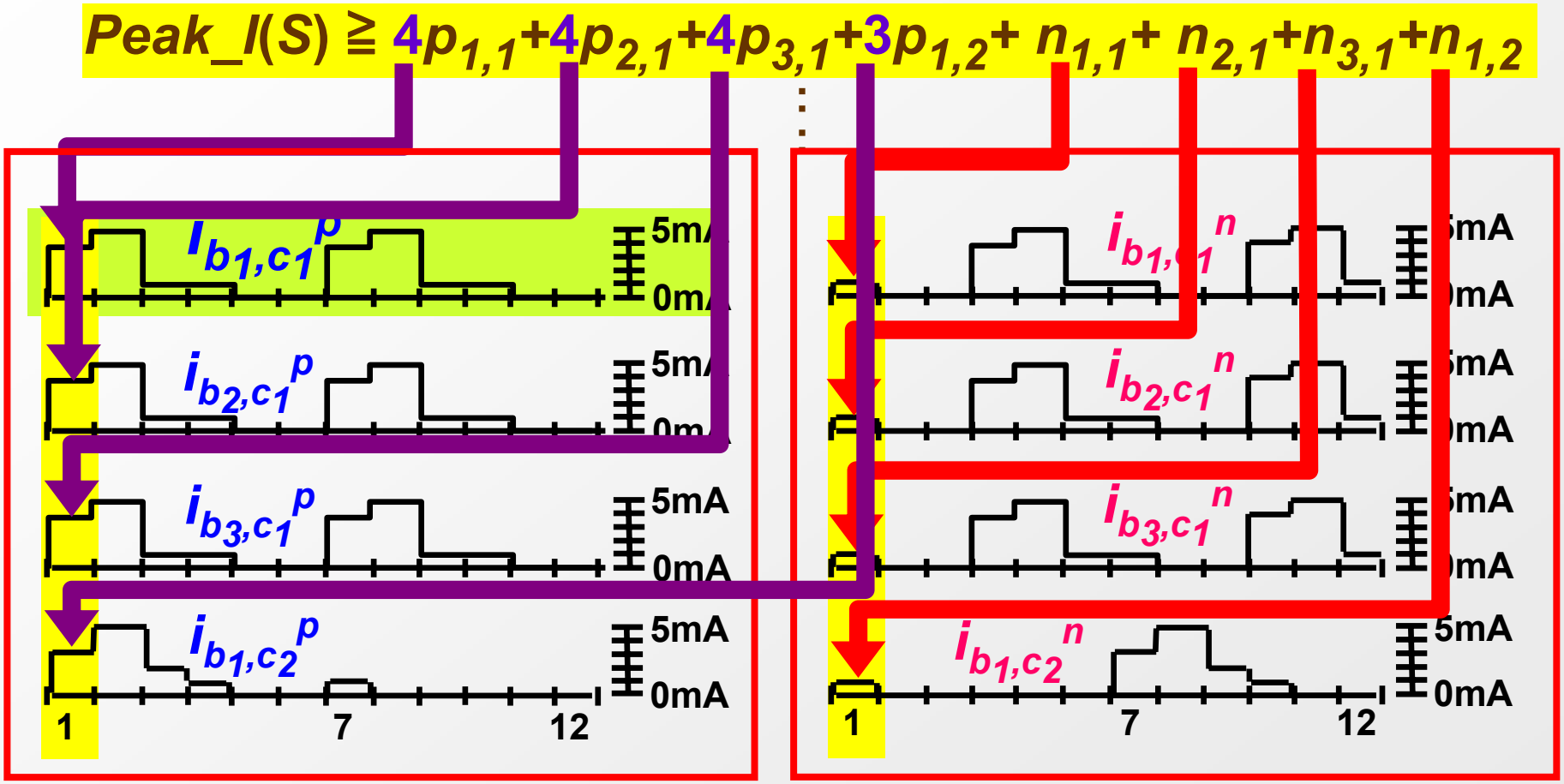
:

$Peak\_I(S) \geqq 5n_{1,1}+5n_{2,1}+5n_{3,1}$

$Peak\_I(S) \geqq n_{1,1}+n_{2,1}+n_{3,1}$

Algorithms

- **ILP Model Generator**



Experiments

- **SLPCR (System-Level Peak Current Reducer)**
  - C Program
- **Peak Current Vector Data of an IP (Block)**
  - Commercial Tool
  - Our In-house Software
  - Etc.
- **LP/ILP Solver**
  - GLPK (GNU Linear Programming Kit)

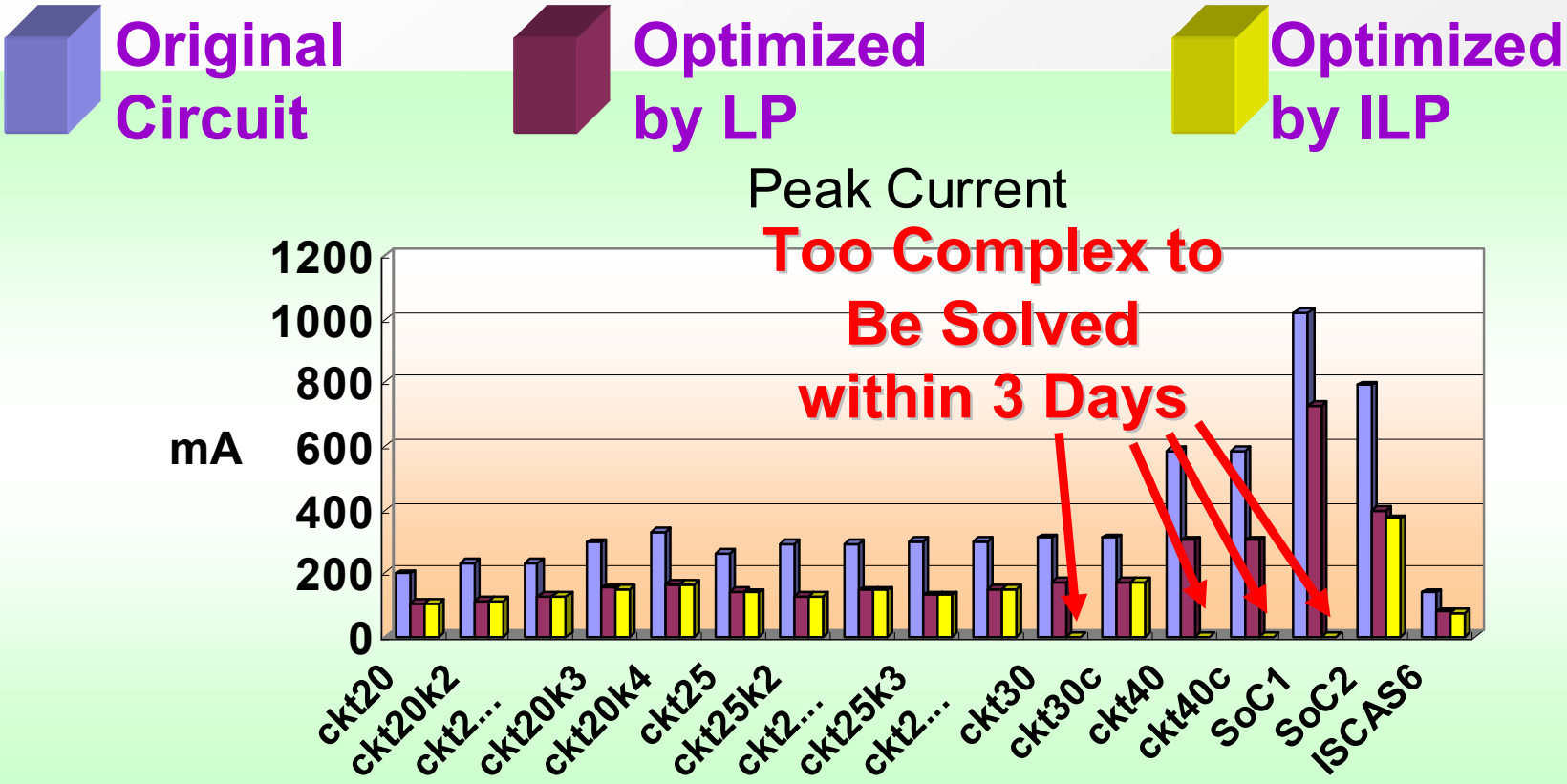
Experiments

- **Benchmark Circuits**
  - 14 Dummy Circuits
  - 2 SoC Designs
  - 1 Gate Level Design
  - The clocks of most blocks of the bench-mark circuits are initialized as positive edge triggered

Bench-marks	#B	#C	#CP	#Ct
ckt20	20	1	20	0
ckt20k4	20	4	33	0
:	:	:	:	:
ckt40c	40	1	40	17
SoC1	56	1	56	4
SoC2	34	4	55	18
ISCAS6	6	1	6	0

Experiments

- **Peak Current Result Reported by SLPCR**



Experiments

- **CPU Times**

Benchmark	LP (sec)	ILP (sec)
ckt20	0.25	0.02
ckt20k2	0.40	745.23
ckt20k2c	0.39	24.46
ckt20k3	0.59	623.60
ckt20k4	0.72	2043.67
ckt25	0.39	33565.01
ckt25k2	0.55	4903.34
ckt25k2c	0.56	242.60
ckt25k3	0.66	6143.18
ckt25k3c	0.64	25.99
ckt30	0.58	–
ckt30c	0.54	40.91
ckt40	1.11	–
ckt40c	1.09	–
SoC1	2.68	–
SoC2	0.87	0.18
ISCAS6	0.04	0.01

Conclusions

- **We propose an approach and some algorithms to reduce the peak current of an IP-based SoC design**
  - Using a clock scheme of mixed positive and negative triggering edges
  - The LP-based algorithm can effectively decide the triggering-edge of each block clock
- **We also propose an ILP technique for finding an optimum solution**
- **Experimental results show that our technique can reduce the peak current up to 56.3%**

Thank You!

Q&A