Activity and Register Placement Aware Gated Clock Network Design

Weixiang Shen, Yici Cai, Xianlong Hong

EDA Lab, CS Dept., Tsinghua University, China

Jiang Hu

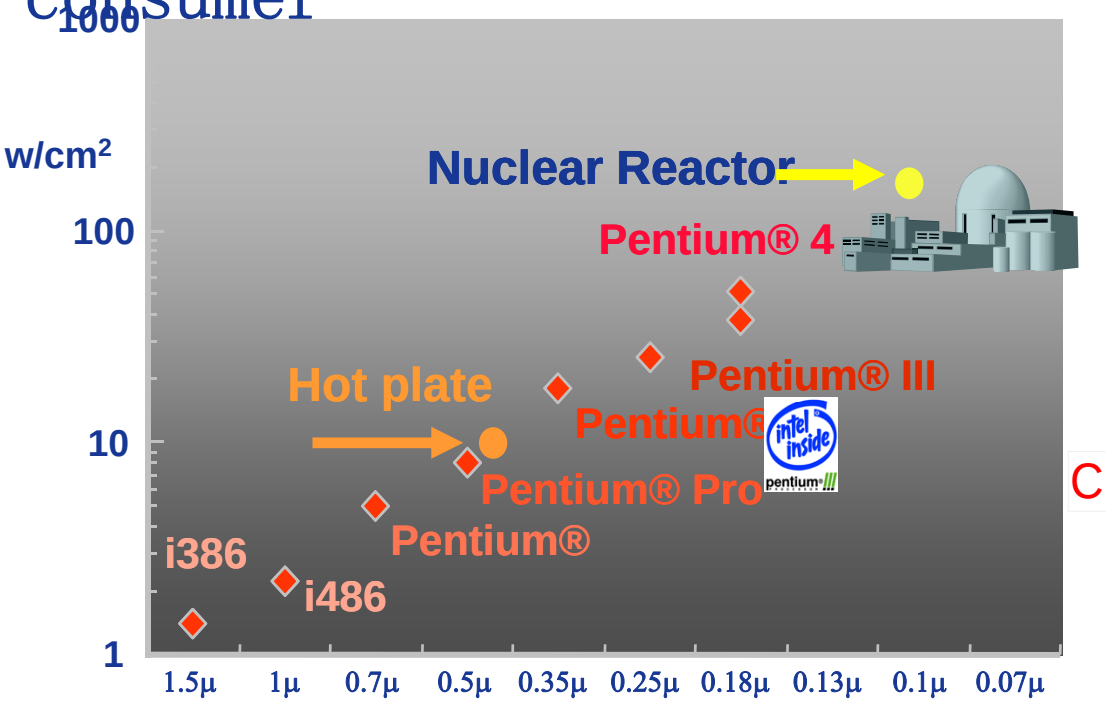
ECE Dept., A&M University, USA

Outline

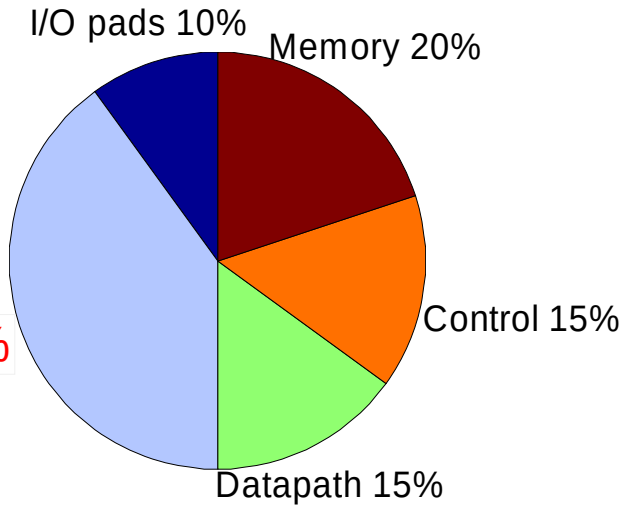
- Introduction
- Motivation
- Our Contributions
- Preliminaries
  - Top-down Min-Cut Placement
  - Power Model
- Our Algorithm
  - Initial Placement
  - Gated Clock Tree Construction
  - Incremental Placement
- Experimental Results
- Conclusions and Future Work

Introduction

- Power becomes a primary concern
- Clock network is the major power consumer



Clock 40%



Introduction

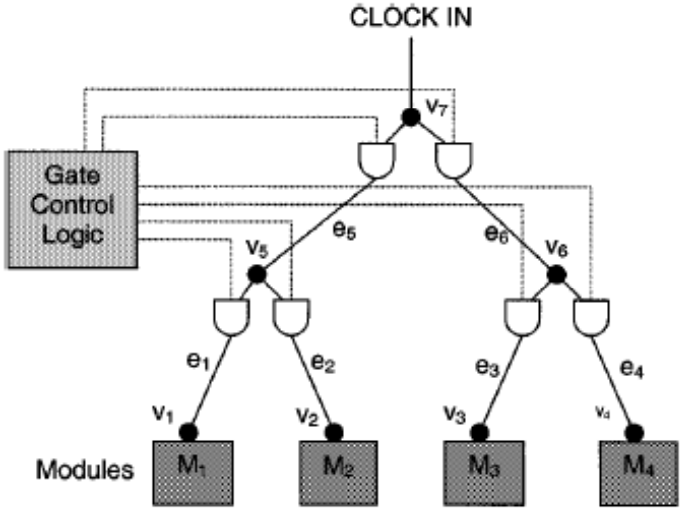
- Clock gating

  - Not all of the cells are active in a particular cycle

  - Shut down the idle cells could save the power

  - Clock gating is the most effective method to do this by

    controll fed into the idle cells



Introduction

- Previous work on clock gating
  - Activity-driven clock tree, TCAD'01
  - Zero-skew gated clock routing, ASPDAC'98, TCAD'01
  - Activity-sensitive clock tree, ISLPED'02
  - Power-aware clock tree planning, DAC'03, ISPD'04
  - Activity-aware register placement, ISVLSI'07
- The drawbacks of these methods
  - Focus on clock tree construction after placement or at RTL
  - Previous placers are not designed for gated clock tree

Outline

- Introduction
- Motivation
- Our Contributions
- Preliminaries
  - Top-down Min-Cut Placement
  - Power Model
- Our Algorithm
  - Initial Placement
  - Gated Clock Tree Construction
  - Incremental Placement
- Experimental Results
- Conclusions and Future Work

Motivation

- The parent's activity is OR-ing of its children
- Place the registers with similar activity patterns closely
  - decrease the clock tree wirelength and capacitance
  - decrease the parent's activity
  - shut off the clock signal more periods
- Activity rate is minimal != transition of control signal is optimal
- To place the gating logics other than removing overlap by ECO
- To control the clock skew

Motivation

- The relationship between activity and transition

reg1: 1010001 reg2: 1010101 reg3: 1100001

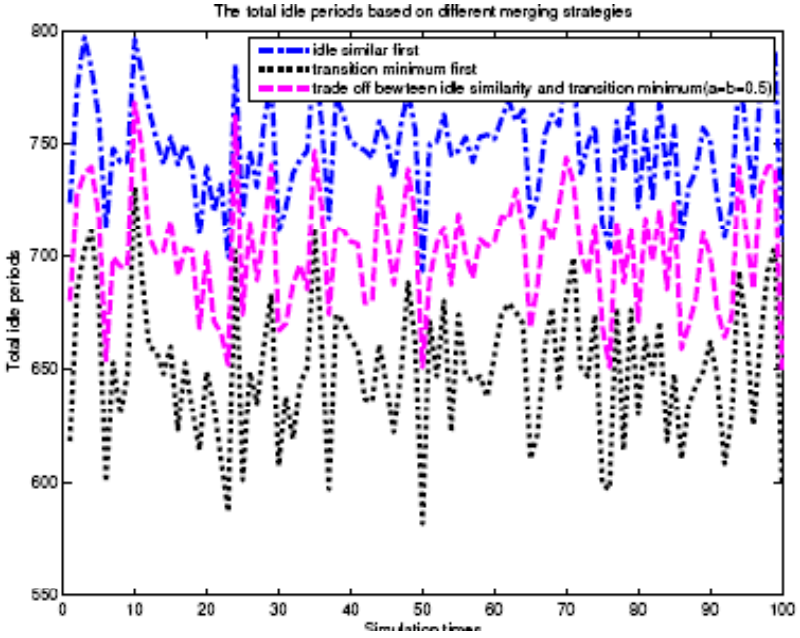
reg1-reg2: 1010101 idle: 3 transitions: 6

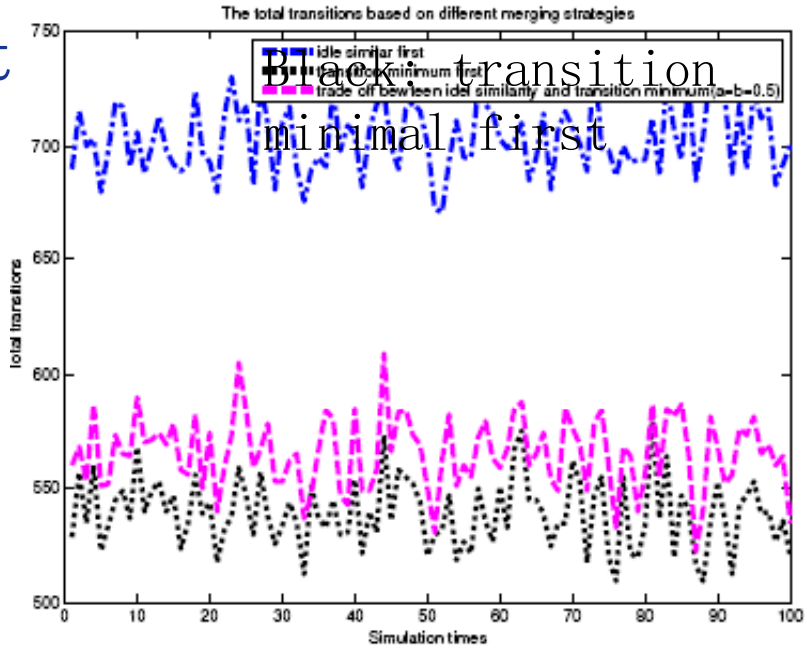
reg1-reg3: 1110001 idle: 3 transitions: 2

Blue: activity similar first

Pink: activity & transition

- Strat





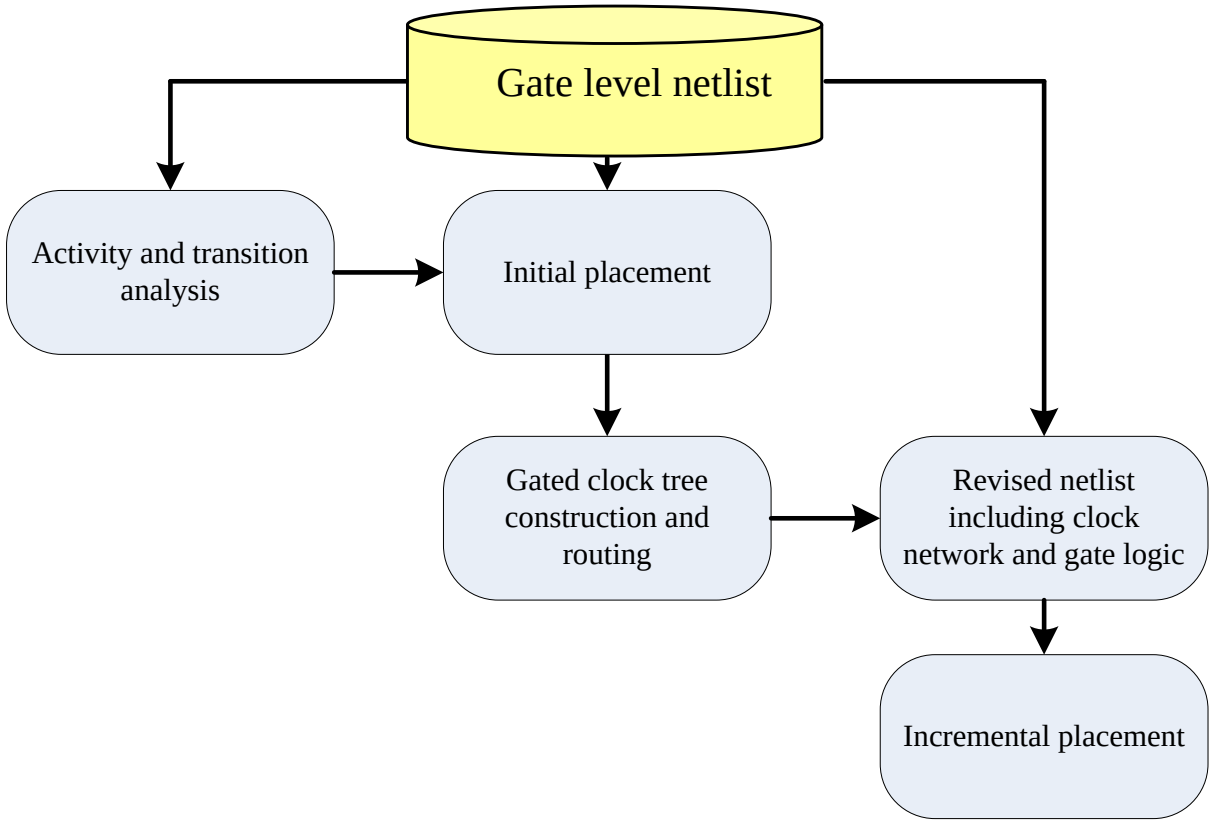
Black transition minimal first

Outline

- Introduction
- Motivation
- Our Contributions
- Preliminaries
  - Top-down Min-Cut Placement
  - Power Model
- Our Algorithm
  - Initial Placement
  - Gated Clock Tree Construction
  - Incremental Placement
- Experimental Results
- Conclusions and Future Work

Our Contributions

- Our design flow



Our Contributions

- Propose a more reasonable design flow, taking the gating logics into placement, avoiding the optimal positions of the gates are occupied by other cells

- In the initial placement, we consider both the logical and physical information. Besides considering the activity similarity of the registers, we also try to minimize the resultant transitions of the control signals

- Optimize the clock skew by implementing zero skew gated

Outline

- Introduction
- Motivation
- Our Contributions
- Preliminaries
  - Top-down Min-Cut Placement
  - Power Model
- Our Algorithm
  - Initial Placement
  - Gated Clock Tree Construction
  - Incremental Placement
- Experimental Results
- Conclusions and Future Work

Preliminaries

- Cut-based placement paradigm – Capo

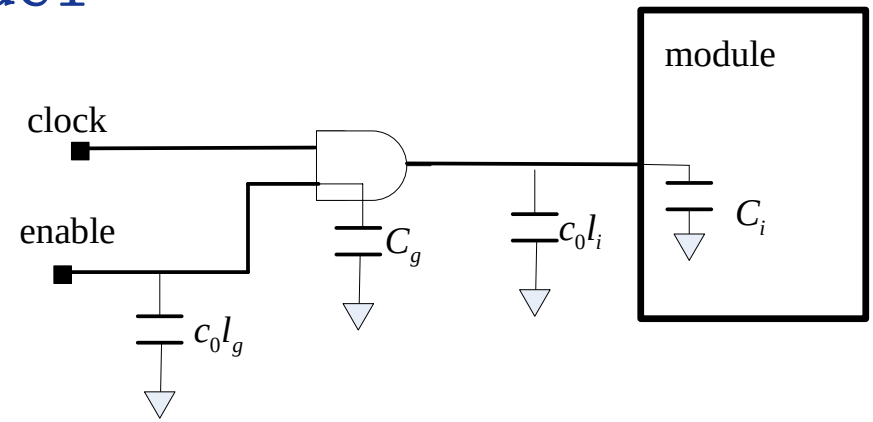
  – A min-cut placer recursively partitions each bin and its

  associated hyper-graph at current level, its objective is to

  minimize the total weighted net cuts

$$[(c_0 l_i + C_i) p(i) + 0.5 * (c_0 l_g + C_g) p_{tr}] f V_{dd}^2$$

- Power model



Outline

- Introduction
- Motivation
- Our Contributions
- Preliminaries
  - Top-down Min-Cut Placement
  - Power Model
- Our Algorithm
  - Initial Placement
  - Gated Clock Tree Construction
  - Incremental Placement
- Experimental Results
- Conclusions and Future Work

Initial Placement

- Objective:

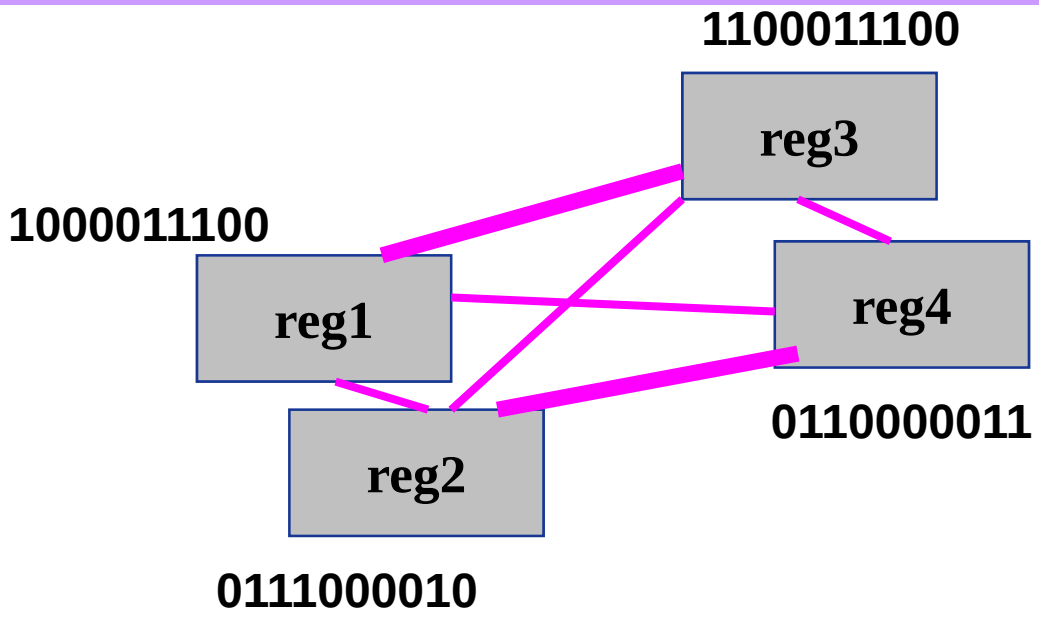
  - Clump the registers into a small area to decrease the clock wirelength

  - Pull the registers with the similar activity patterns closely to reduce its parent's activity and the transitions of the control signal

- Strategy:

  - Capo is a minimal cut-based placer, so we can add some pseudo edges between the registers we want to pull closely

Initial Placement



- The pseudo edge's weight is determined based on
  - w_l, with respect to the partition level
  - w_s, with respect to the similarity of the activity patterns
  - w_t, with respect to the transitions of the control

Initial Placement

- The weight of w_1

$$w_1(level)=-3.0*level/20.0+W \quad \text{if } level>T$$

- T: the partition level we start to add pseudo edge
- W: control the scope of w_1
- T is determined based on the circuit scale
- For the beginning level, w_1 should be assigned a relatively large

  value to avoid partitioning the registers with the similar activity

  patterns to different bins
- For the last levels, the area of the bin is relatively small

Initial Placement

- The weight of w_s

$$w_s(reg1, reg2) = \frac{(C_{reg1} + C_{reg2})}{C_{average}} * p(reg1, reg2)$$

$$p(reg1, reg2) = P(reg1.act(i) = 0, reg2.act(i) = 0) \quad (0 \le i \le act.size)$$

$$Average(bin) = \frac{2}{n*(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_s(regi, regj)$$

- If reg1 and reg2's activity patterns are similar, their parent's activity

  rate will reduce, and also they can share a common gate

- Add a pseudo edge between every pair of registers will affect the run

  time of Capo, and it is no need if w_s is small

- If the similarity between regi and regj is larger than C_0*Average(bin),

Initial Placement

- The weight of w_t

$$w_t(regi, regj) = \frac{trans(parAct(regi, regj))}{act.size() - 1}$$

- parAct(regi, regj) gets the parent's activity pattern of regi and regj

- trans(act) gets the transition of the control signal

e.g. reg1: 10110001 reg2: 00010010

➔ parent(reg1, reg2): 10110011

➔ transition probability of the control signal: 4/7

- The resultant weight w

$$w(edge(regi, regj)) = w_l(L) * (\alpha w_s(regi, regj) + \beta(1 - w_t(regi, regj)))$$

Gated Clock Tree Construction

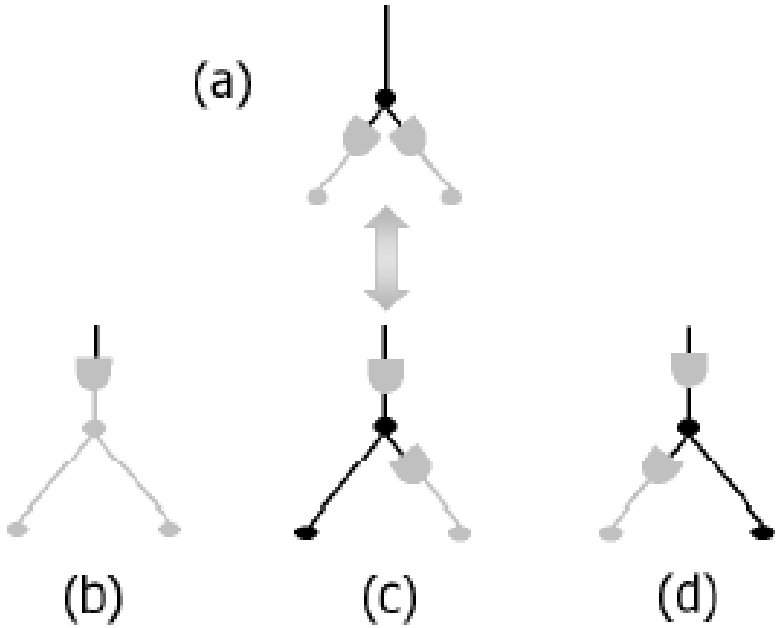
- Node merging

$$dis(i,j) = \alpha f(D(i,j)) + \beta g(L(i,j))$$

$$= \alpha[\frac{1}{\dim_{\max}}(|x_i - x_j| + |y_i - y_j|)] \rightarrow \text{Physical information}$$

$$+ \beta[1 - \frac{1}{C_{tot}}(C_i + C_j)p(i,j)] \Rightarrow \text{Activity formation}$$

- Gate moving



Gated Clock Tree Construction

Algorithm: Clock Tree Construction

Require: a set of sinks $\{n_i$

repeat

for each pair of n_i, n_j

compute $dis(n_i,$

end for

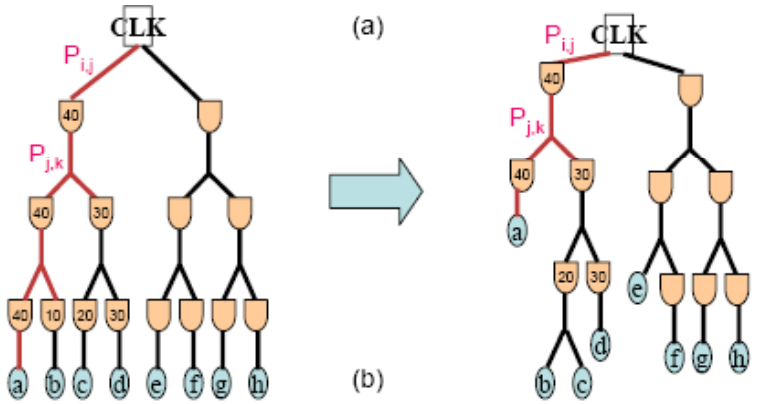
pick the pair n_i, n_j whose $dis(n_i, n_j)$ is minimum

merge these two nodes and generate parent n_k

remove node n_i, n_j

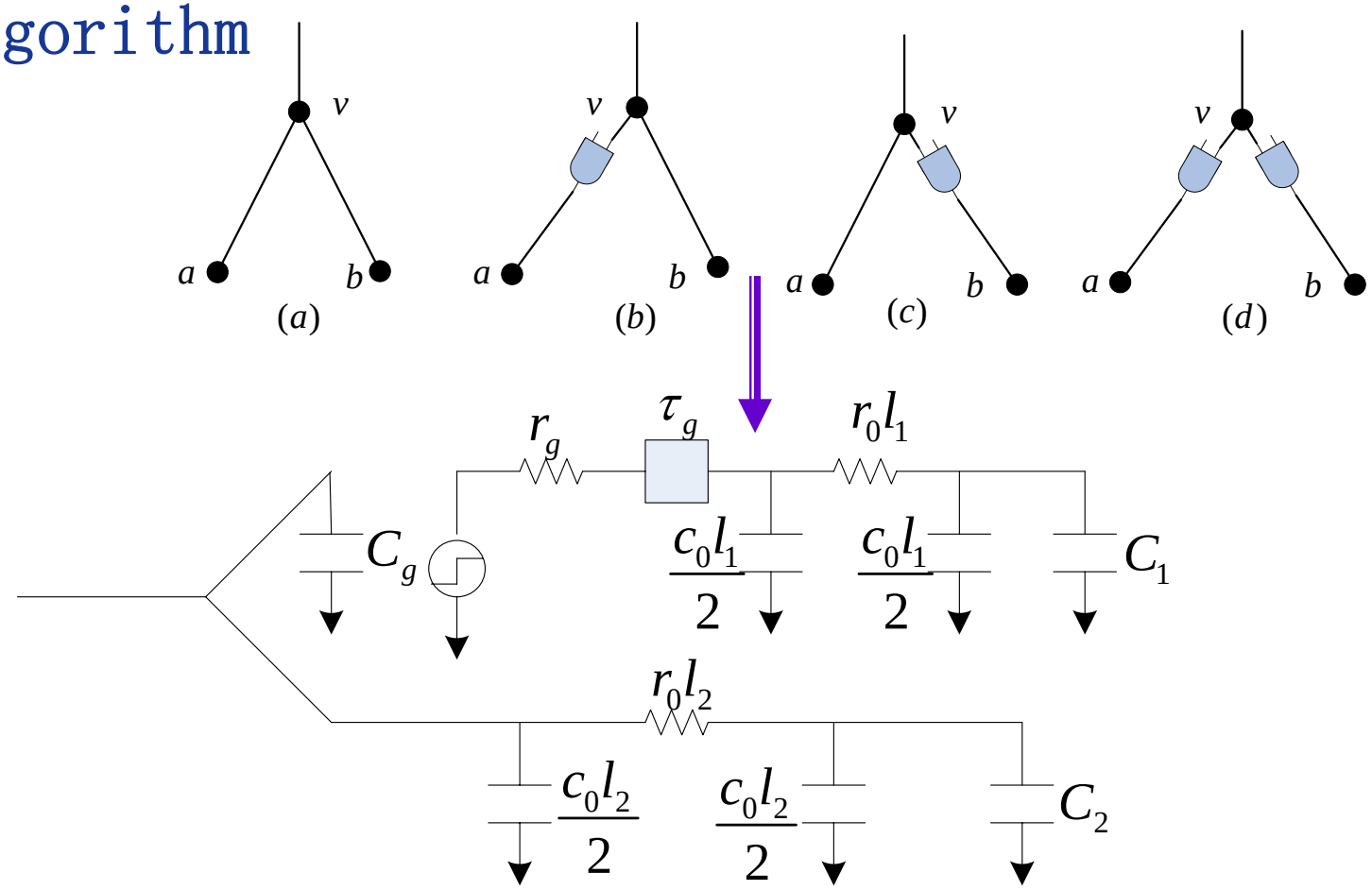
push n_k to nodes set

until only one node is left in nodes set



Gated Clock Tree Construction

- Zero skew clock routing based on the DME algorithm



Incremental Placement

- Integrate the gated clock network into previous database

  - change the +Placed attribute of the registers to +Fixed

  - add the gating logics into the database and fix them

- The key of our incremental placement

  - try to preserve the gated clock network we optimized before and

    remove the overlaps of the gating logics with the other cells

  - minimize the overheads of signal nets wirelength and power against

Outline

- Introduction
- Motivation
- Our Contributions
- Preliminaries
  - Top-down Min-Cut Placement
  - Power Model
- Our Algorithm
  - Initial Placement
  - Gated Clock Tree Construction
  - Incremental Placement
- Experimental Results
- Conclusions and Future Work

Experimental Results

- Comparison results of our algorithm with ISVLSI'07

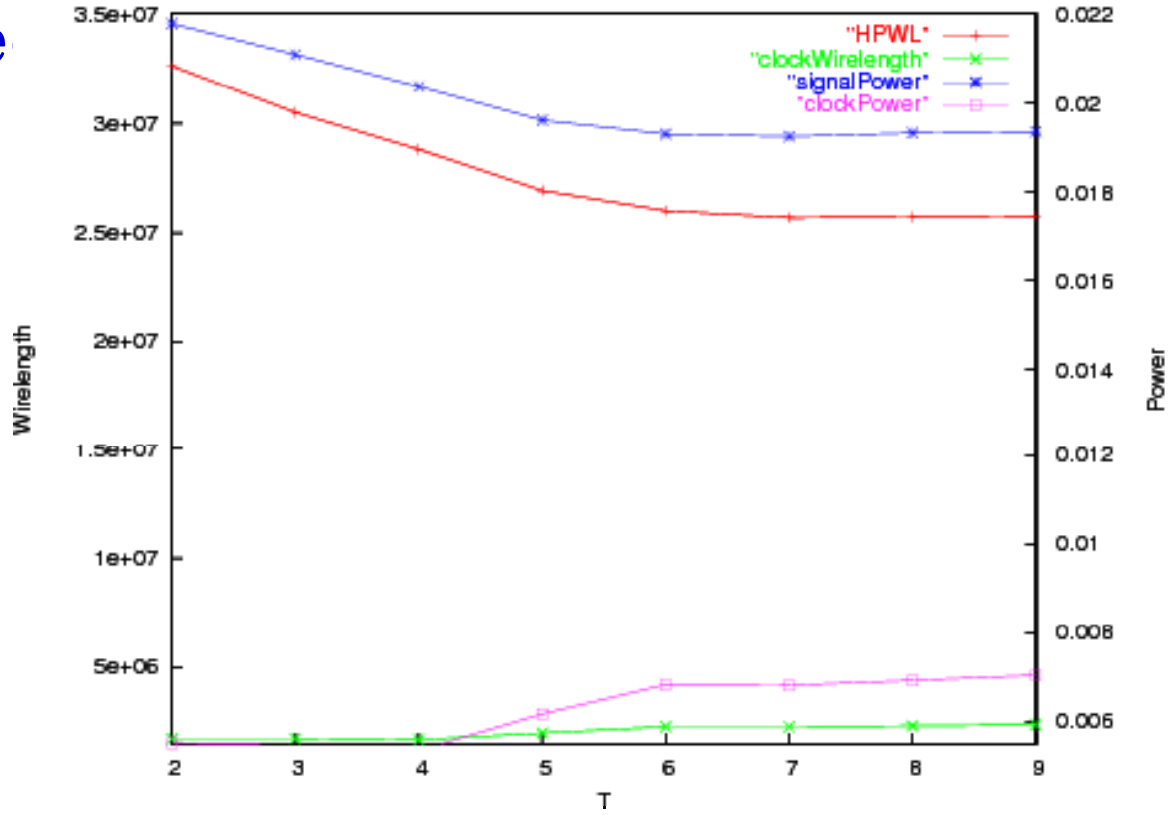
Circuit	Algorithm	Clock Wirelength	Clock Power	#Gate	Clock Skew (sec)
s1488	ref	11362.50	8.17E-5	2	1.62E-11
	our	9464.29	7.70E-5	1	<1.0E-12
s15850	ref	1.85E6	0.005795	38	2.02E-8
	our	1.68E6	0.005399	44	<1.0E-12
s35932	ref	5.61E6	0.007507	99	9.57E-8
	our	5. 23E6	0.007071	105	<1.0E-12
s38417	ref	5.48E6	0.027230	121	5.17E-8
	our	5.20E6	0.025969	103	<1.0E-12
s38584	ref	4.99E6	0.012993	103	8.09E-8
	our	4.69E6	0.012288	98	<1.0E-12
	Ave ratio	0.91	0.94	0.90	

Experimental Results

Circuit	Algorithm	HPWL	Signal Power	Slack (ns)
s1488	ref	1.90E6	0.003160	0.891
	our	1.93E6	0.003280	0.840
s15850	ref	2.53E6	0.018373	0.348
	our	2.66E6	0.019017	0.638
s35932	ref	4.81E6	0.014851	1.48
	our	5. 16E6	0.015409	3.59
s38417	ref	5.63E6	0.064018	-12.80
	our	5.98E6	0.066258	-9.10
s38584	ref	6.48E6	0.036635	-5.83
	our	6.82E6	0.037526	-2.48
	Average ratio	1.05	1.03	

Experimental Results

- Clock nets and signal nets trade off by pseudo e



T ↓ → clock nets wirelength and power ↓

HPWL and signal nets power ↑

Outline

- Introduction
- Motivation
- Our Contributions
- Preliminaries
  - Top-down Min-Cut Placement
  - Power Model
- Our Algorithm
  - Initial Placement
  - Gated Clock Tree Construction
  - Incremental Placement
- Experimental Results
- Conclusions and Future Work

Conclusions and Future Work

● Conclusions

- Activity pattern plays an important role in gated clock tree construction

- Activity aware register placement is effective for gated clock tree to

  reduce power

- Transition of the control signal could not be ignored

- Clock nets power and wirelength Vs. signal nets power and HPWL

● Future work

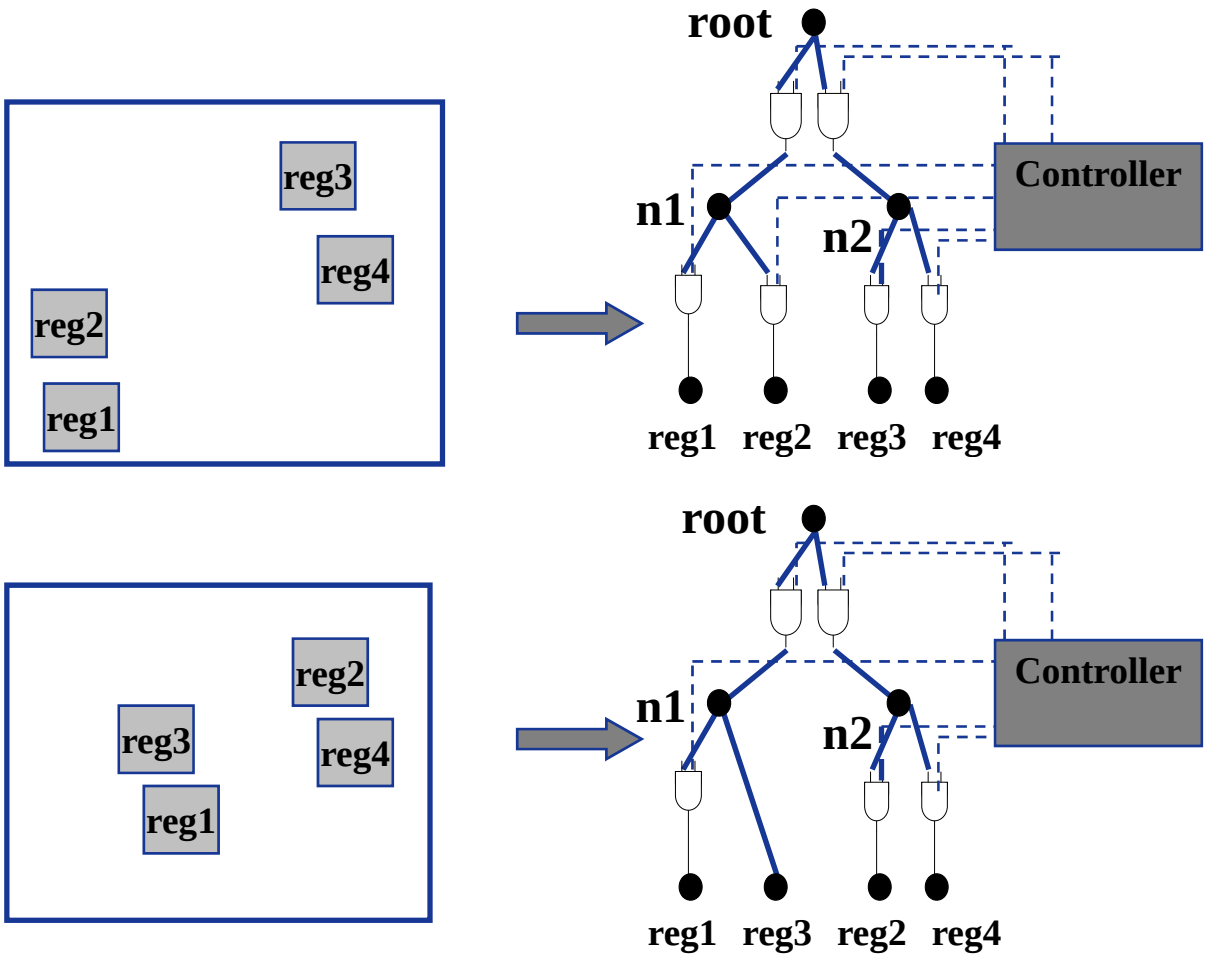
- Gating logic planning during placement

- Power Vs. performance (delay, skew, timing, et al.)

清華大學
Tsinghua University

Thank you !

E-mail: cwx04@mails.tsinghua.edu.cn



reg1: 1000011100
reg2: 0111000010
reg3: 1100011100
reg4: 0110000011

a) n1: 1111011110
n2: 1110011111

b) n1: 1100011100