Low Power Discrete Voltage Assignment Under Clock Skew Scheduling

Li Li[1] Jian Sun[2] Yinghai Lu[1] Hai Zhou[1] Xuan Zeng[2]

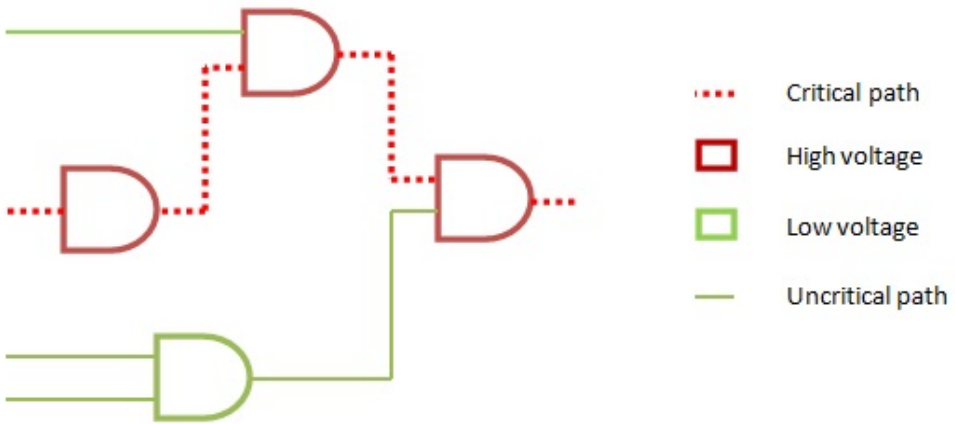
[1]Department of EECS
Northwestern University

[2]Department of Microelectronics
Fudan University

January 27, 2011

Power Optimization via Multiple Supply Voltage Assignment

- Each gate has multiple choices of supply voltage
  - Tradeoff between power and performance



- High supply voltage is assigned to timing critical cells to guarantee performance
- Low supply voltage is assigned to other cells to save power.

Discrete Voltage Assignment Problem

- The problem of discrete voltage assignment has been studied in various context, some of the previous works include:

Publications	Approach	Solution Type
Lee, ICCAD'06	Dynamic Programming	Optimal
Lee, ICCAD'07	ILP	Optimal
Qian, ISPD'09	Branch and Bound	Optimal
Ma, ICCAD'08	Network Flow & Flooring	Heuristic
Feng, ICCAD'09	LP & Dual Binary Search	ϵ-approximation

- All the existing works only consider combinational circuits.

Clock skew scheduling can "steal" time from uncritical paths and further reduce power!

Discrete Voltage Assignment Problem

- The problem of discrete voltage assignment has been studied in various context, some of the previous works include:

Publications	Approach	Solution Type
Lee, ICCAD'06	Dynamic Programming	Optimal
Lee, ICCAD'07	ILP	Optimal
Qian, ISPD'09	Branch and Bound	Optimal
Ma, ICCAD'08	Network Flow & Flooring	Heuristic
Feng, ICCAD'09	LP & Dual Binary Search	ϵ-approximation

- All the existing works only consider combinational circuits.

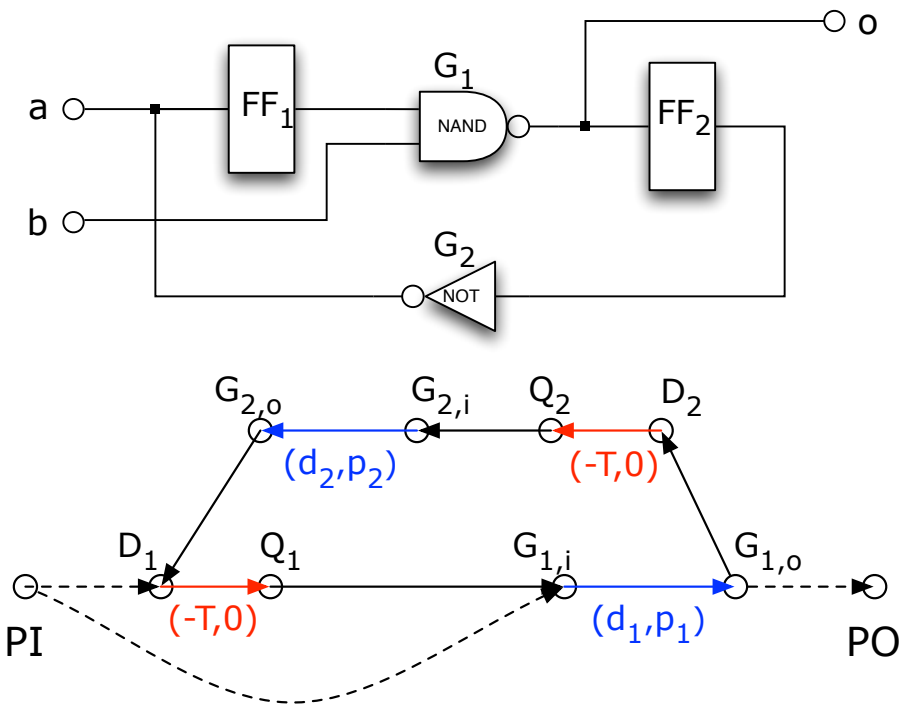
Clock skew scheduling can "steal" time from uncritical paths and further reduce power!

Outline

Outline

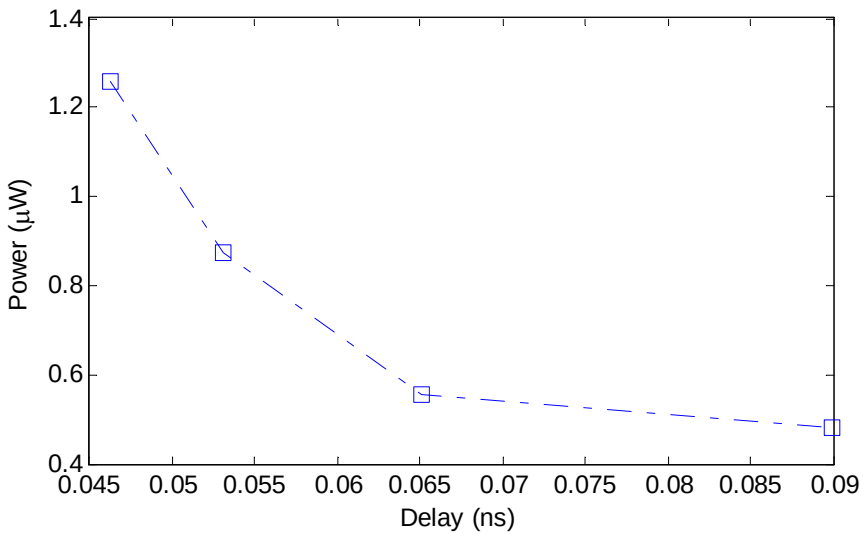
Constraint Graph

- Given a sequential circuit, a directed graph $G = (V, E)$ can be constructed as follows. $E = E_{gate} \cup E_{FF} \cup E_{net}$.
  - $(i, j) \in E_{gate} \cup E_{FF}$ represents a gate or FF. i is input, j is output.
  - $(i, j) \in E_{net}$ represents interconnect between gates and FFs.
  - Two dummy node: PI represents all primary inputs, PO represents all primary outputs.



Delay-Power Curve

- Each gate has several supply voltage candidates with fixed *(delay, power)* pair.
- $\forall (i,j) \in E_{gate}$, $\overline{P}_{ij}(d_{ij}^k) = p_{ij}^k$ for $1 \leqslant k \leqslant k_{ij}$, where k_{ij} is the number of possible voltage options of each gate, $\overline{P}_{ij}$ represents the function that maps delay to power on (i,j).
- $\overline{P}_{ij}$ is typically a convex function.



Timing Constraints and Transformation

- d_{ij}: gate delay or net delay on edge (i, j).
- t_i: arrival time on node i.
- $\forall (i, j) \in E_{gate} \cup E_{net}$

$$t_i + d_{ij} \leqslant t_j$$

- s_{ij}: clock skew on flip-flop (i, j).
- s_{max}: the maximum acceptable skew.
- $\forall (i, j) \in E_{FF}$

$$t_j \geqslant s_{ij} \quad \wedge \quad t_i \leqslant s_{ij} + T \quad \wedge \quad 0 \leqslant s_{ij} \leqslant s_{max}$$

which is equal to

$$t_j \geqslant t_i - T \quad \wedge \quad t_j \geqslant 0 \quad \wedge \quad t_i \leqslant T + s_{max}$$

Formulation

- The timing constraint on each edge $(i,j) \in E$ can be uniformly stated as $t_i + d_{ij} \leqslant t_j$, where d_{ij} is defined as

$$d_{ij} = \begin{cases} d_{ij}, & (i,j) \in E_{gate} \cup E_{net} \\ -T, & (i,j) \in E_{FF} \end{cases}$$

- Discrete Voltage Assignment Problem under Clock Skew Scheduling:

$$\begin{aligned} Min \quad & \sum_{(i,j)\in E_{gate}} \overline{P}_{ij}(d_{ij}) \\ s.t. \quad & t_i \geqslant 0, && i = PI && (1a) \\ & t_i \leqslant T, && i = PO && (1b) \\ & t_i \leqslant t_j - d_{ij}, && \forall (i,j) \in E && (1c) \\ & t_j \geqslant 0, && \forall (i,j) \in E_{FF} && (1d) \\ & t_i \leqslant T + s_{max}, && \forall (i,j) \in E_{FF} && (1e) \\ & d_{ij} \in \{d_{ij}^1, \cdots, d_{ij}^{k_{ij}}\}, && \forall (i,j) \in E_{gate} && (1f) \end{aligned}$$

Hardness Review

- Well known as NP-hard.
- Though Feng et al. proposed an ϵ-approximation to this problem, it turns out to be flawed.

- Discrete Voltage Assignment is a special application of Discrete Time-Cost Tradeoff problem, which is a classic project scheduling problem in operation science. It has been proved to be
  - Strongly NP-hard
  - No ϵ-approximation could exist unless $P = NP$.
  - No constant-approximation was found.

Hardness Review

- Well known as NP-hard.
- Though Feng et al. proposed an ϵ-approximation to this problem, it turns out to be flawed.

- Discrete Voltage Assignment is a special application of Discrete Time-Cost Tradeoff problem, which is a classic project scheduling problem in operation science. It has been proved to be
  - Strongly NP-hard
  - No ϵ-approximation could exist unless $P = NP$.
  - No constant-approximation was found.

Outline

Continuous Relaxation

- Linear piece-wise *delay-power curve*.
- A new power-delay function P_{ij} without bounds:

$$P_{ij}(d_{ij}) = \begin{cases} \overline{P}_{ij}(d_{ij}^{k_{ij}}), & d_{ij} \geqslant d_{ij}^{k_{ij}} \\ \overline{P}_{ij}(d_{ij}^{q}) - b_{ij}^{q}(d_{ij} - d_{ij}^{q}), & d_{ij}^{q} \leqslant d_{ij} \leqslant d_{ij}^{q+1} \\ \quad \vdots & \\ \overline{P}_{ij}(d_{ij}^{1}) - M(d_{ij} - d_{ij}^{1}), & d_{ij} \leqslant d_{ij}^{1} \end{cases}$$

$b_{ij}^{q} = -\frac{P_{ij}(d_{ij}^{q}) - P_{ij}(d_{ij}^{q-1})}{d_{ij}^{q} - d_{ij}^{q-1}}$ is the absolute slope of power-delay function between d_{ij}^{q} and d_{ij}^{q-1}. M is a sufficiently large number which can be viewed as a penalty factor for violating the bound constraints.

Convex Cost Flow Dual Problem

- Define a bound constraint function

$$B_i(t_i) = \begin{cases} M \times (t_i - u_i), & t_i > u_i \\ 0, & 0 \leq t_i \leq T \\ -M \times (t_i - l_i), & t_i < l_i \end{cases}$$

  where l_i and u_i are the lower and upper bounds of t_i

- After putting arrival time constraints into objective function, our problem becomes a convex-cost flow dual problem

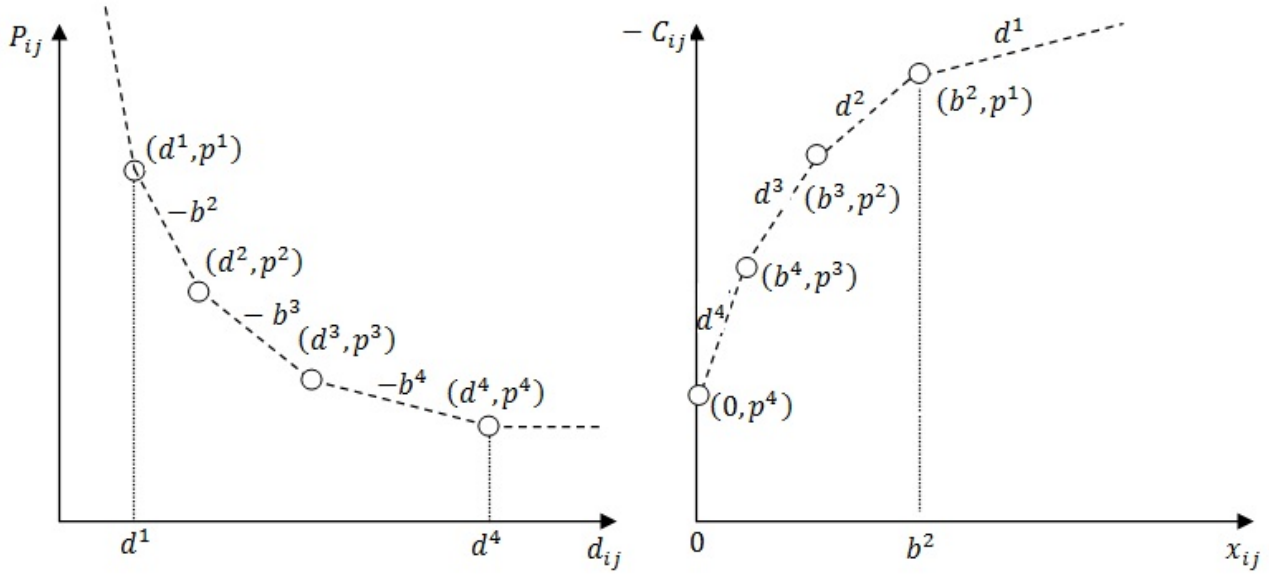
$$\begin{aligned} Min \quad & \sum_{(i,j)\in E} P_{ij}(d_{ij}) + \sum_{i \in V} B_i(t_i) \\ s.t. \quad & t_i \leqslant t_j - d_{ij}, \qquad \forall (i,j) \in E \end{aligned}$$

Convex-cost Flow Problem

- The dual of voltage assignment problem

$$
\begin{aligned}
Min \quad & \sum_{(i,j)\in E} C_{ij}(x_{ij}) \\
s.t. \quad & \sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 0 \qquad \forall i \in V \\
& 0 \leqslant x_{ij} \leqslant M \qquad\qquad \forall (i,j) \in E
\end{aligned}
$$

- Cost function C_{ij} can be obtained from P_{ij}.
- Dual relation: $d_{ij} \leftrightarrow b_{ij} \quad (x_{ij})$



ϵ-Residual Graph

- A potential $\pi(i)$ is attached to each node i.
- The excess of a node i is defined as

$$X(i) = \sum_{(j,i)\in E} x_{ji} - \sum_{(i,j)\in E} x_{ij}$$

- The residual capacity $q(i,j)$ of each edge $(i,j) \in E$ is defined as

$$q(i,j) = -\mathcal{P}_{ij}^{-}(\lfloor \pi(j) - \pi(i) \rfloor) - x_{ij}$$

for forward edge and

$$q(j,i) = \mathcal{P}_{ij}^{+}(\lfloor \pi(j) - \pi(i) \rfloor) + x_{ij}$$

for backward edge, where $\mathcal{P}_{ij}^{-}$ and $\mathcal{P}_{ij}^{+}$ are the left and right slope of power function P_{ij}.

ϵ-Residual Graph

- The residual cost of each edge $(i, j) \in E$ is defined as

$$c_{ij}^{\pi} = \mathcal{C}_{ij}^{+}(x_{ij}) - \pi(i) + \pi(j)$$

  for forward edge and

$$c_{ji}^{\pi} = -\mathcal{C}_{ij}^{-}(x_{ij}) - \pi(j) + \pi(i)$$

  for backward edge, where $\mathcal{C}_{ij}^{-}$ and $\mathcal{C}_{ij}^{+}$ are the left slope and right slope of cost function C_{ij}.
- A pseudoflow is said to be ϵ-optimal for $\epsilon > 0$ if $\exists \pi$ satisfies that $c_{ij}^{\pi} \geqslant -\epsilon \ \forall (i, j) \in E$.
- An edge (i, j) is admissible if $X(i) > 0 \wedge q(i, j) > 0 \wedge$ $-\epsilon \leqslant c_{ij}^{\pi} < 0$

Convex Cost-Scaling Algorithm

- The algorithm gradually transforms the ϵ-optimal solution into $\epsilon/2$-optimal solution by pushing as much as possible flow through admissible edges, until$\epsilon < 1/|V|$.
- `Initialize` $\pi \leftarrow 0$ $x \leftarrow 0$ $\epsilon \leftarrow \max_{(i,j)\in E} \left| b_{ij}^{k_{ij}} \right|$

```
while ε ≥ 1/|V|
    for each admissible edge (i,j) ∈ E(x)
        Push q(i,j) flow through (i,j)
    end for
    while there is a node i with excess flow
        if there is an admissible edge (i,j)
            push min(X(i), q(i,j)) flow through (i,j)
        else
            π(i) ← π(i) + ε/2
        end if
    end while
end while
```

Continuous Solution

- The optimal flow x^* and node potential π^* returned by convex-cost scaling algorithm may be non-integer.
- Integer π can be obtained by computing the shortest distance $sp(i)$ from PI to other nodes in the residual graph and set $\pi(i) = -sp(i)$.
- The optimal solution can be then obtained by assigning $t_i = \pi(i)$ and $d_{ij} = t_j - t_i$ for each $(i, j) \in E_{gate}$.
- Clock skew s_{ij} for FF embedded in edge (i, j) is feasible with any value in $[t_i - T, t_j]$.

Outline

Continuous $\longrightarrow$ Discrete

- Deterministic Flooring?
  - Can not guarantee good performance, may waste lots of power.
- Random Rounding?
  - May violate timing constraints.

We develop a mincut based heuristic to reuse the "leftover".

Continuous $\longrightarrow$ Discrete

- Deterministic Flooring?
  - Can not guarantee good performance, may waste lots of power.
- Random Rounding?
  - May violate timing constraints.

We develop a mincut based heuristic to reuse the "leftover".

Continuous $\longrightarrow$ Discrete

- Deterministic Flooring?
  - Can not guarantee good performance, may waste lots of power.
- Random Rounding?
  - May violate timing constraints.

We develop a mincut based heuristic to reuse the "leftover".

Continuous $\longrightarrow$ Discrete

- Deterministic Flooring?
  - Can not guarantee good performance, may waste lots of power.
- Random Rounding?
  - May violate timing constraints.

We develop a mincut based heuristic to reuse the "leftover".

Continuous $\longrightarrow$ Discrete

- Deterministic Flooring?
  - Can not guarantee good performance, may waste lots of power.
- Random Rounding?
  - May violate timing constraints.

We develop a mincut based heuristic to reuse the "leftover".

Continuous $\longrightarrow$ Discrete

- Deterministic Flooring?
  - Can not guarantee good performance, may waste lots of power.
- Random Rounding?
  - May violate timing constraints.

We develop a mincut based heuristic to reuse the "leftover".

Continuous $\longrightarrow$ Discrete

- Deterministic Flooring?
  - Can not guarantee good performance, may waste lots of power.
- Random Rounding?
  - May violate timing constraints.

We develop a mincut based heuristic to reuse the "leftover".

Reduced Problem with Known Skew

- With optimal clock skew in continuous solution, the sequential circuit can be reduced to combinational circuit.
- The reduced problem

$$
\begin{aligned}
Min \quad & \sum_{(i,j)\in E_{gate}} P(d_{ij}) \\
s.t. \quad & t_i + d_{ij} \leqslant t_j, && \forall (i,j) \in E^{'} && (2a) \\
& d_{ij} \in \{d_{ij}^1, \cdots, d_{ij}^{k_{ij}}\}, && \forall (i,j) \in E_{gate} && (2b)
\end{aligned}
$$

- The new graph $G^{'} = (V, E^{'})$ is slightly different.
  - All edges $(i,j) \in E_{FF}$ are canceled and two new edges (PI, j) and (i, PO) are introduced with $d(PI, j) = s_{ij}^*$ and $d(i, PO) = -s_{ij}^*$, where s_{ij}^* is the optimal clock skew.
  - Edge (PO, PI) with $d(PO, PI) = -T$ is introduced to eliminate the arrival time constraints.

Karush-Kuhn-Tucker condition

- The KKT condition of reduced problem

$$
\begin{aligned}
t_i + d_{ij} &\leqslant t_j && \forall (i,j) \in E' \\
-\mathcal{P}_{ij}^{+}(d_{ij}) \leqslant x_{ij} &\leqslant -\mathcal{P}_{ij}^{-}(d_{ij}) && \forall (i,j) \in E' \\
\sum_{(i,j)\in E'} x_{ij} - \sum_{(j,i)\in E'} x_{ji} &= 0 && \forall i \in V' \\
x_{ij} &\geqslant 0 && \forall (i,j) \in E' \\
x_{ij}(t_j - t_i - d_{ij}) &= 0 && \forall (i,j) \in E'
\end{aligned}
$$

- Observations: If initially set all $x_{ij} = 0$, $d_{ij} = d_{ij}^{k_{ij}}$, $t_{PI} = 0$ and $t_i = MaxDelay(PI, i)$, the ONLY violated constraint is timing constraint on edge (PO, PI)!
- Intuition: Maintain constraints while reducing delays.
- Push along longest paths (critical network) on which each edge $(i,j) \quad s.t. \quad t_j = t_i + d_{ij}$!
- *Monotonic Grow*: Once in critical network, always in.

Karush-Kuhn-Tucker condition

- The KKT condition of reduced problem

$$
\begin{aligned}
t_i + d_{ij} &\leqslant t_j && \forall (i,j) \in E' \\
-\mathcal{P}_{ij}^{+}(d_{ij}) \leqslant x_{ij} &\leqslant -\mathcal{P}_{ij}^{-}(d_{ij}) && \forall (i,j) \in E' \\
\sum_{(i,j)\in E'} x_{ij} - \sum_{(j,i)\in E'} x_{ji} &= 0 && \forall i \in V' \\
x_{ij} &\geqslant 0 && \forall (i,j) \in E' \\
x_{ij}(t_j - t_i - d_{ij}) &= 0 && \forall (i,j) \in E'
\end{aligned}
$$

- Observations: If initially set all $x_{ij} = 0$, $d_{ij} = d_{ij}^{k_{ij}}$, $t_{PI} = 0$ and $t_i = MaxDelay(PI, i)$, the ONLY violated constraint is timing constraint on edge (PO, PI)!
- Intuition: Maintain constraints while reducing delays.
- Push along longest paths (critical network) on which each edge $(i,j) \quad s.t. \quad t_j = t_i + d_{ij}$!
- *Monotonic Grow*: Once in critical network, always in.

Mincut-based Optimal Algorithm for Combinational Circuits

Initialize $d_{ij} \leftarrow d_{ij}^{k_{ij}}$ $x_{ij} \leftarrow 0$ $\forall (i,j) \in E'$
Initialize $c_{ij} \leftarrow -\mathcal{P}_{ij}^{-}(d_{ij})$ $c_{ji} = 0$ $\forall (i,j) \in E'$
while $MaxDelay(PI, PO) > T$
    Identify critical network G^0
    Solve maximum flow in the residual graph of G^0
    Select a mincut M and maximum possible δd
    for each $(i,j) \in M$
        if (i,j) is a forward edge
            $d_{ij} = d_{ij} - \delta d$
        else $d_{ij} = d_{ij} + \delta d$
        end if
    end for
    $c_{ij} \leftarrow -\mathcal{P}_{ij}^{-}(d_{ij}) - x_{ij}$ $\forall (i,j) \in G^0$
    $c_{ji} \leftarrow \mathcal{P}_{ij}^{+}(d_{ij}) + x_{ij}$ $\forall (i,j) \in G^0$
end while

Mincut-based Heuristic

Initialize $q_{ij} \leftarrow m_{ij}$ $x_{ij} \leftarrow 0$ $\forall (i,j) \in E'$
Initialize $c_{ij} \leftarrow -\mathcal{P}_{ij}^{-}(d_{ij}^{q_{ij}})$ $\forall (i,j) \in E'$
while $MaxDelay(PI, PO) > T$
    Identify approximate critical network G^0
    Solve maximum flow in the residual graph of G^0
    Select a mincut M
    $q_{ij} \leftarrow q_{ij} - 1$ $\forall forward\ \ edge\ \ (i,j) \in M$
    $c_{ij} \leftarrow -\mathcal{P}_{ij}^{-}(d_{ij}^{q_{ij}}) - x_{ij}$ $\forall (i,j) \in E'$
end while

- Main changes from continuous version
  - Switch from discrete level to level.
  - Initial delay is the floor of continuous solution instead of max.
  - Ignore backward edges.
  - Expand the range of critical network.

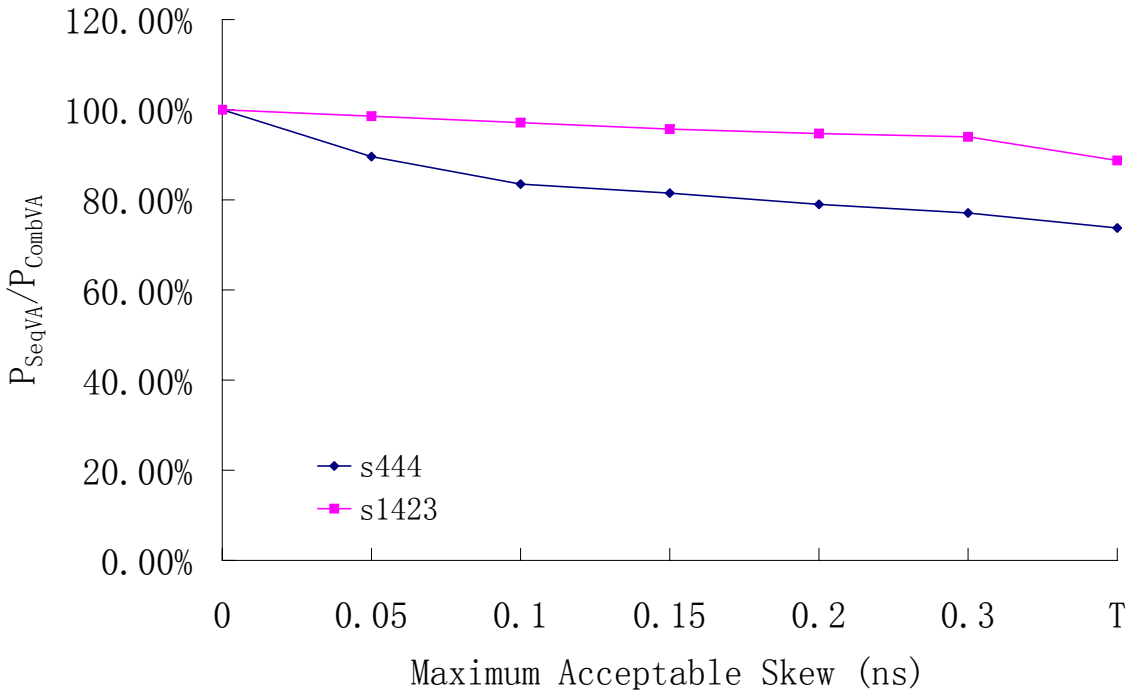
Outline

Experiment Setup

- Linux workstation with 3.0GHz CPU and 2.0GB memory.
- Implemented in C++.
- Tested on ISCAS89 benchmarks.
- The delay-power curve of each cell is simulated by HSPICE_U-2003.03-SP1 with supply voltage set as 0.8V, 1.0V, 1.2V and 1.4V respectively.
- Clock period T is set to 1.1X the minimum period and maximum allowed clock skew s_{max} is set to T
- Results compared with Ma et al. ICCAD'09 Network flow based approach.

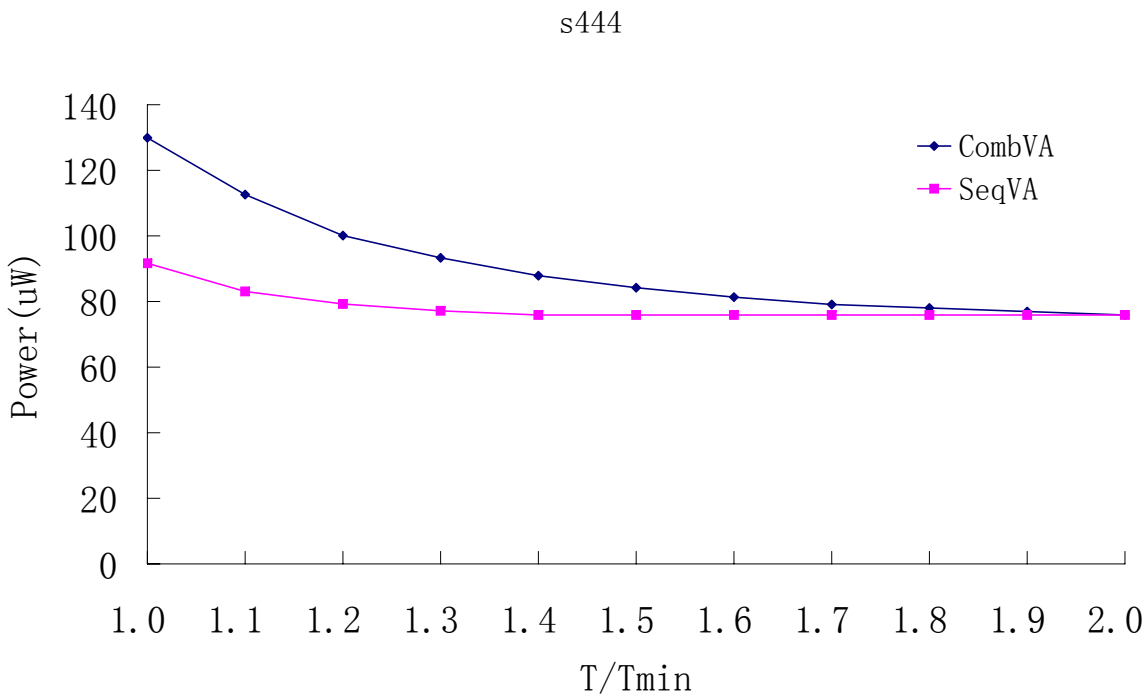
Power Ratio with Maximum Allowed Skew

- As the maximum allowed clock skew increases, more power can be saved through clock skew scheduling.



Power under Different Timing Constraints

- The impact of clock skew scheduling is more significant under tight timing constraints. The power consumption of SeqVA is very close to the lower limit even when the timing constraints are very tight.



- 9.2% additional power saving on average.
- Less running time.
- Achieve power consumption only 1.77% larger than theoretical lower bound.

Thank you!