Graph Partition based Path Selection for Testing of Small Delay Defects

Zijian He, Tao Lv, Hua-wei Li, Xiao-wei Li

Key Laboratory of Computer System & Architecture
Institute of Computing Technology, CAS

Outline

- Introduction
- Motivation
- Critical path graph
- Graph partition based path selection
  - Cut-node partition
  - Independent Path Set Partition
- Experimental results
- Conclusions

Introduction

- Continuous shrinking of device feature size increases the following effects:
  - Process Variation
  - Power Noise
  - Crosstalk
- Path Correlations
  - Share common gates and wires
  - Spatial correlations
- Purpose: find a path set *S*,

$$probability(circuit\_delay > clk \mid \forall p \in S, delay_p < clk) \approx 0$$

Motivation

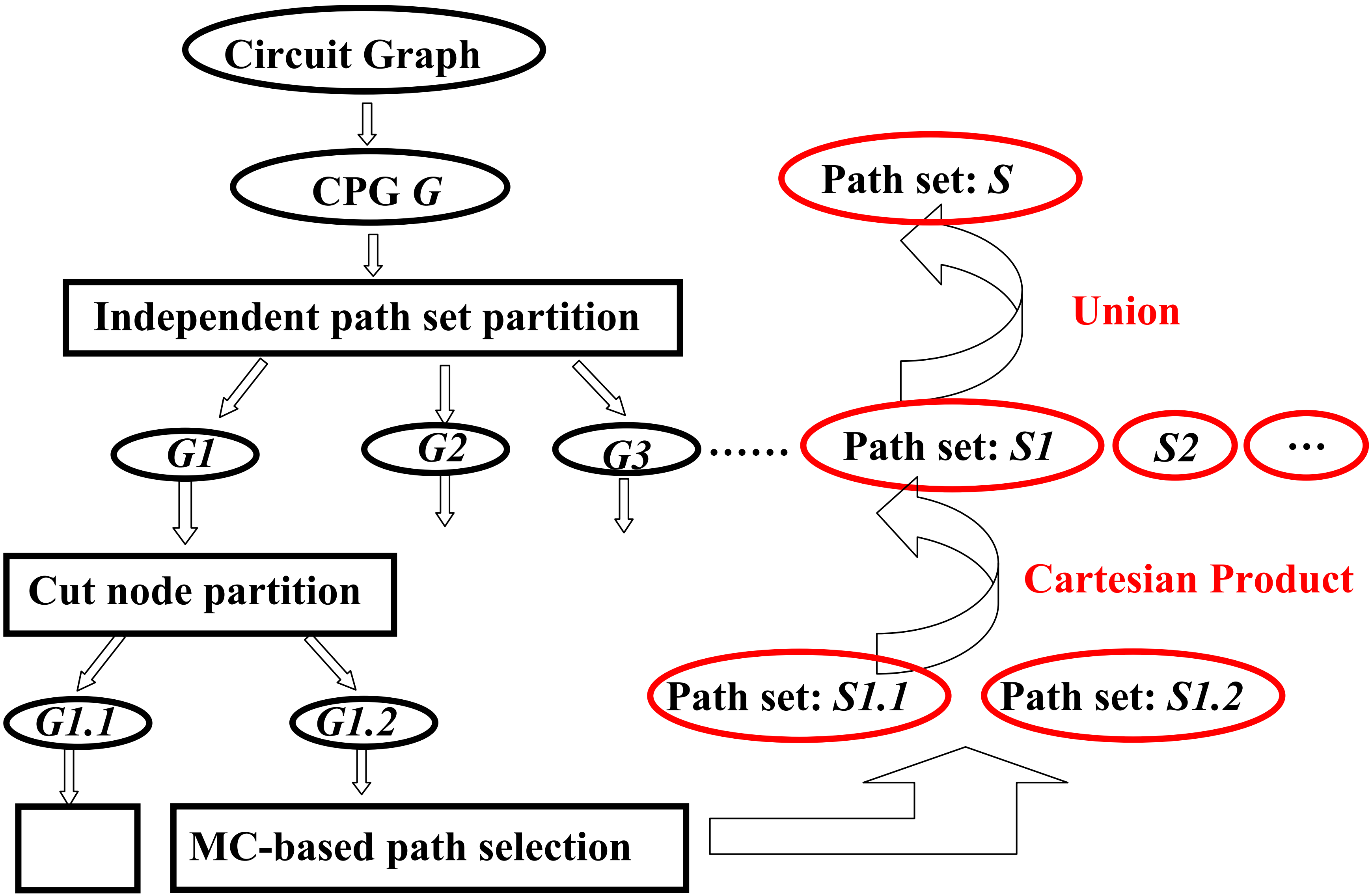
- Monte Carlo simulation based path selection
  - A candidate critical path set is generated first
  - Circuit sample are generated for path selection

- Computing time is propositional to the number of candidate critical paths
  - Very inefficient for some circuits with large number of candidate critical paths

Our contribution

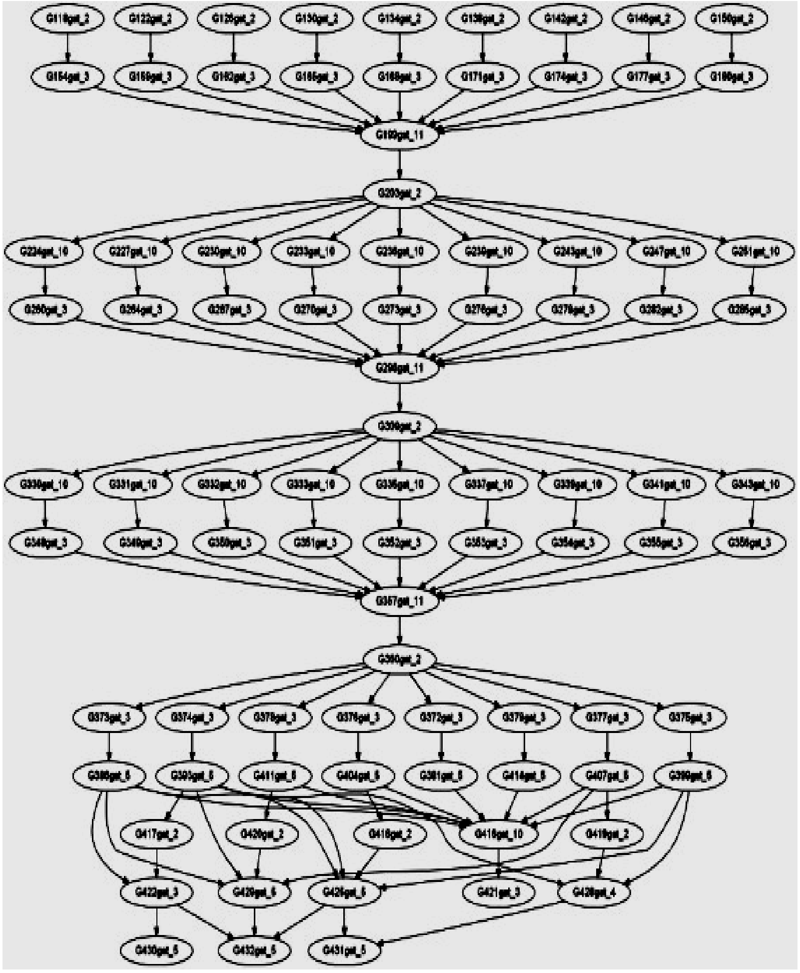
- Graph Partition based path selection
  - CPG: Critical Path Graph
    - sub graph containing all candidate critical paths
  - Two graph partition approaches for CPG
    - Cut-node partition
    - Independent path set partition
  - Two operations for test path set generation
    - Cartesian product operation
    - Union operation
- Square root level of computing complexity

Overflow of our approach



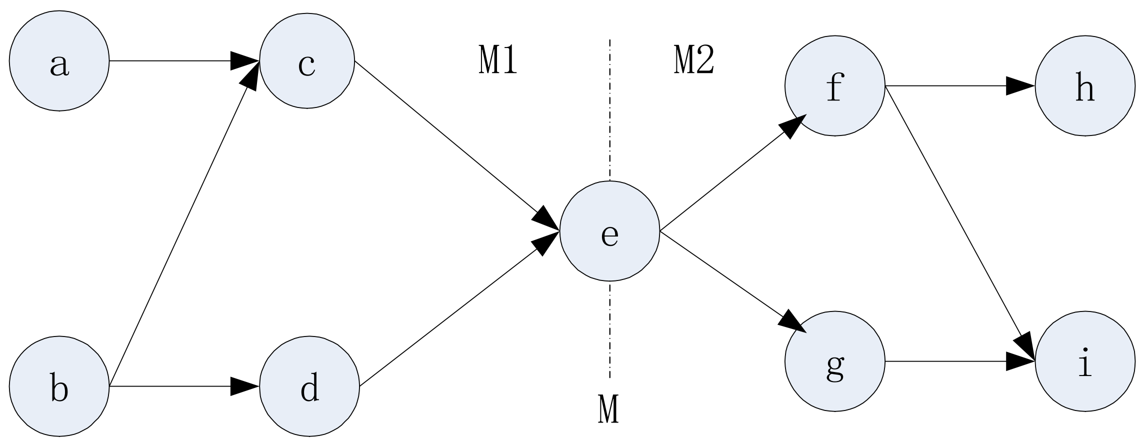
Critical Path Graph

- A path is defined critical if it has certain probability longer than reference clock *T*
- *CPG (Critical path graph)* contains all critical paths



CPG of *C432* from ISCAS85

Cut Node Partition based path selection

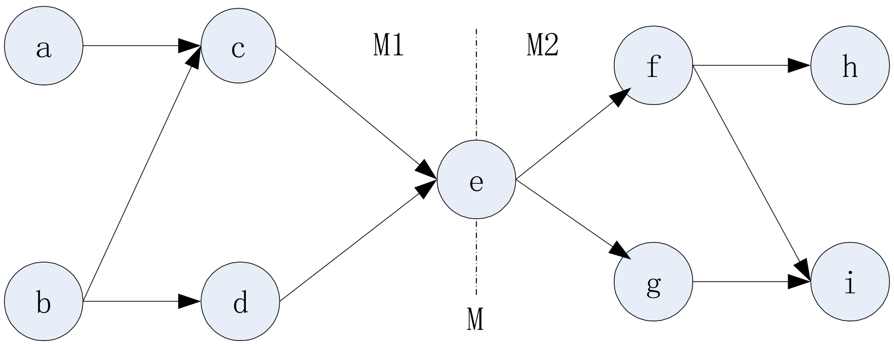


S1: A path set of *M1* that can capture the worst delay situation of *M1* with a high probability,

S2: A path set of *M2* that can capture the worst delay situation of *M2* with a high probability,

*S=S1*S2*: A path set of *M* that can capture the worst delay situation of *M* with a high probability,

Theoretical analysis



- Definition:

$$CL(S_1) = Min\{prob(d_{P1} + d_{P2} < clk \mid d_{S1} + d_{P2} < clk) \mid P_1 \in R_{M1}, P_2 \in R_{M2}\} \quad (1)$$

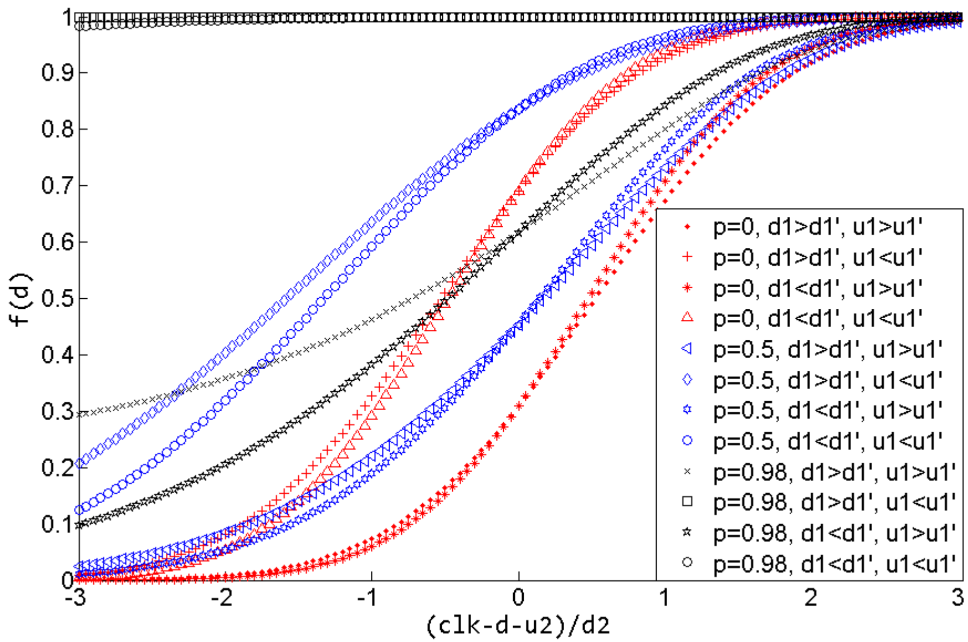
$$CL(S_2) = Min\{prob(d_{P1} + d_{P2} < clk \mid d_{P1} + d_{S2} < clk) \mid P_1 \in R_{M1}, P_2 \in R_{M2}\} \quad (2)$$

$$CL(S) = Min\{prob(d_P < clk \mid d_S < clk) \mid P \in R_M\} \quad (3)$$

- *CL*(*S1*) and *CL*(*S2*) give the confidence level how the character path set *S1* and *S2* can capture the delay character of *M1* and *M2*.
- *CL*(*S*) gives an confidence level how the character path set *S* can capture the delay character of *M*.
- Theroem1: $CL(S) \geq CL(S_1) * CL(S_2)$

An interesting phenomena

- *P1* ∈ *M1*, *P1'* ∈ *M1*,
- *P* ∈ *M2*, *d* represent the delay of *P*
- $f(d) = prob(d_{P1} + d < clk \mid d_{P1'} + d < clk)$
- *f(d)* is almost monotonically decreases with *d*.



Useful results

$$CL(S_1) = Min\{prob(d_{P1} + d_{P2} < clk \mid d_{S1} + d_{P2} < clk) \mid P_1 \in R_{M1}, P_2 \in R_{M2}\}$$

$$f(d) = prob(d_{P1} + d < clk \mid d_{P1'} + d < clk) \mid P_1 \in R_{M1}, P_1' \in R_{M1}, P \in R_{M2})$$

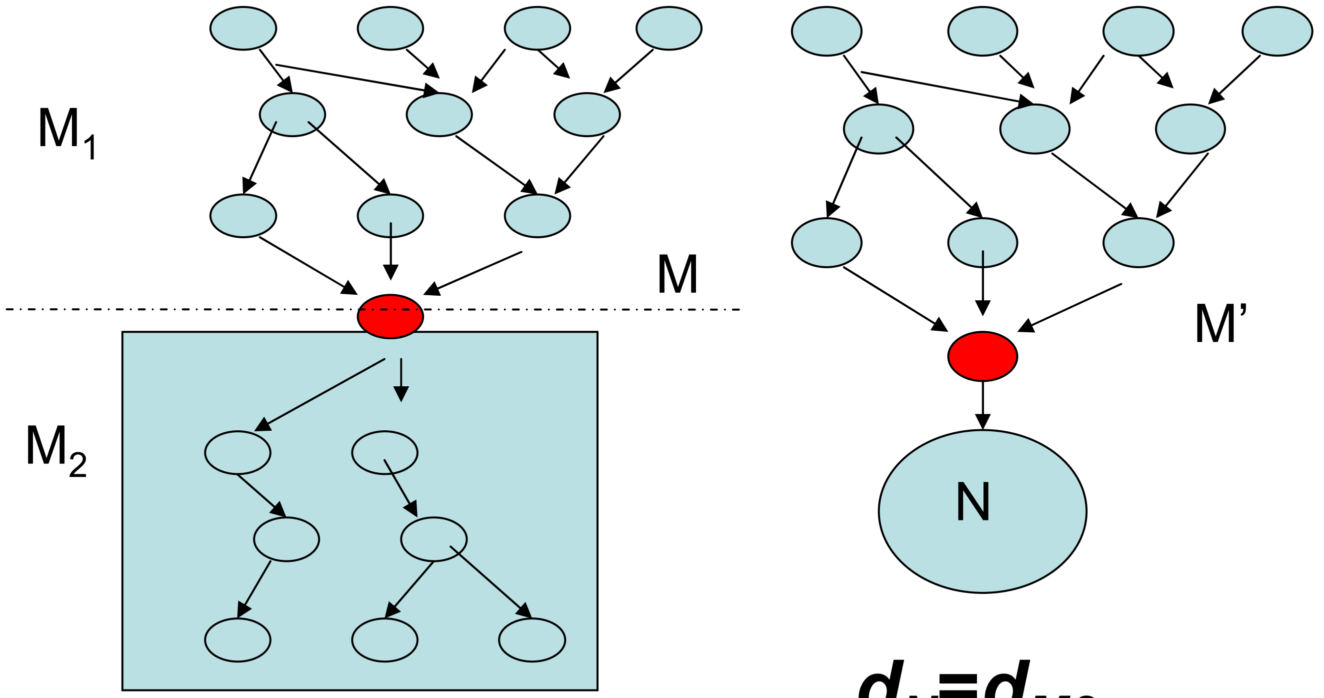
- *If P1' =S1, then CL(S1) = Min(f(dP2).*
- *Min*(*f*(*dP2*)=*f*(*dM2*),
  - because *f*(*d*) is a monotonically decreasing function,
  - where dM2 is the maximum delay of *M2.*
- Hence

$$CL(S_1) = Min\{prob(d_{P1} + d_{M2} < clk \mid d_{S1} + d_{M2} < clk) \mid P_1 \in R_{M1}\}$$

$$CL(S_2) = Min\{prob(d_{M1} + d_{P2} < clk \mid d_{M1} + d_{S2} < clk) \mid P_2 \in R_{M2}\}$$

Abstraction

$$CL(S_1) = Min\{prob(d_{P1} + d_{M2} < clk \mid d_{S1} + d_{M2} < clk) \mid P_1 \in R_{M1}\}$$



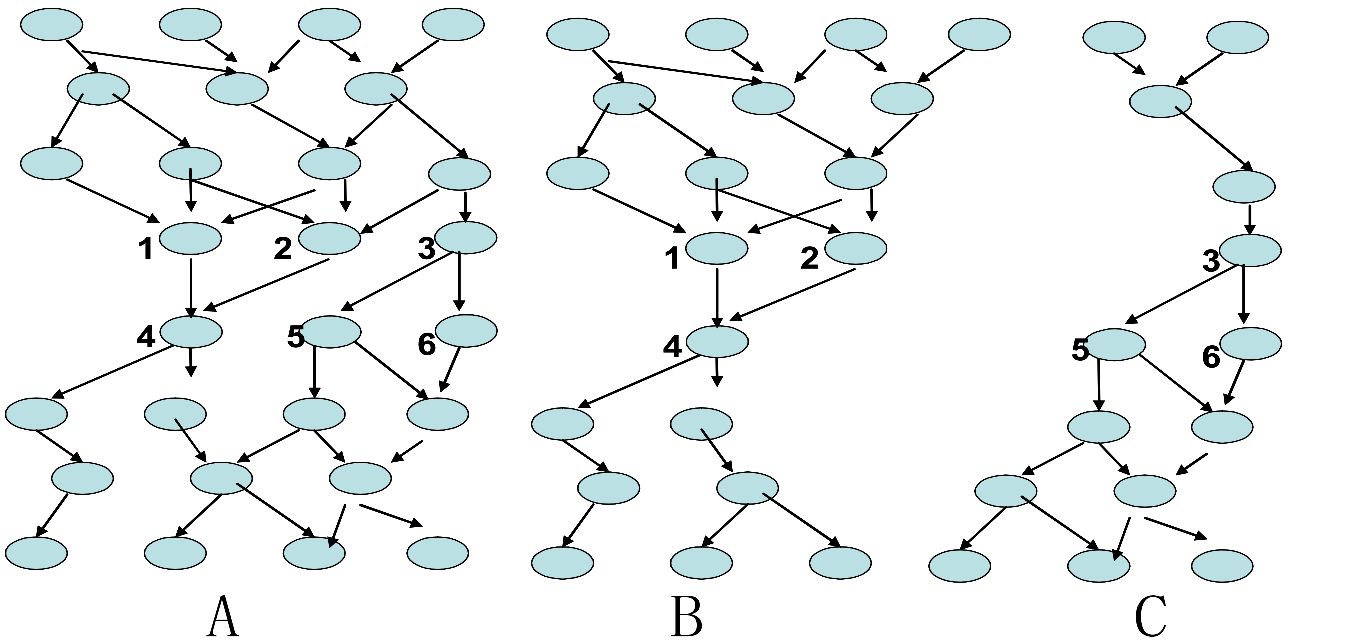
$d_N = d_{M2}$

Computation complexity analysis

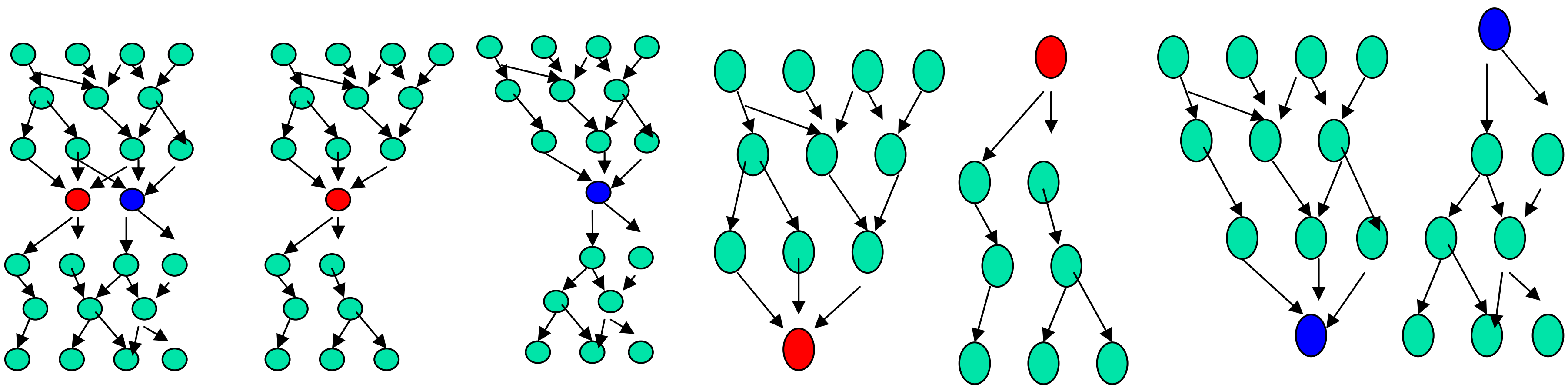
- Graph *G* is partition into *G1* and *G2*
  - *M*, *M1*, *M2*: path number of *G*, *G1* and *G2*
  - *N*, *N1*, *N2*: the number of test paths of *G*, *G1* and *G2*
- $||M||=||M1||*||M2||$.
- $||N||=||N1||*||N2||$.
- The computing time is propositional to $||M||$ and $||N||$
  - Square root level computing complexity

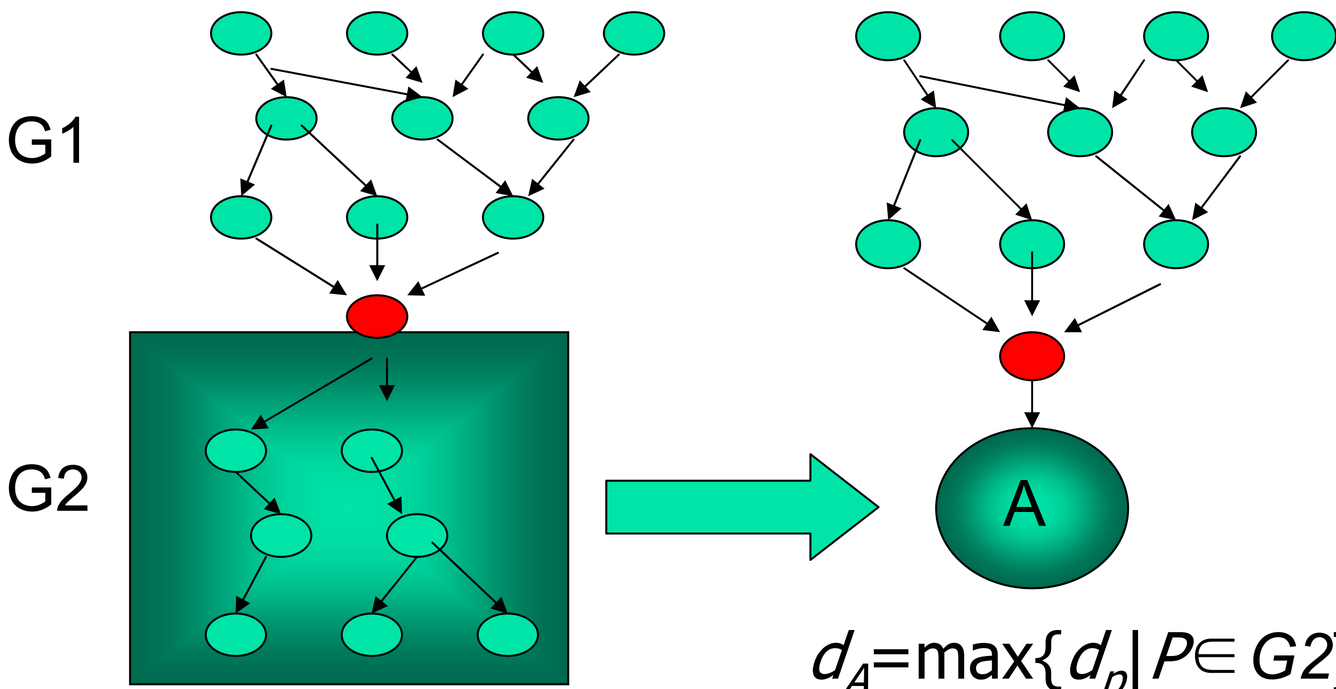
Independent Path Set Partition based path selection

- *CNS* (cut-node set): a node set satisfying the condition that any path of *G* passes through one and only one node in it.
- The original graph *G* is partitioned into several sub-graphs *G1, G2*, … ,each contains a node in *CNS*
- The test path set of *G* is the union of the test path set of *G1, G2, …*



Example





$d_A = \max\{ d_p | P \in G2 \}$

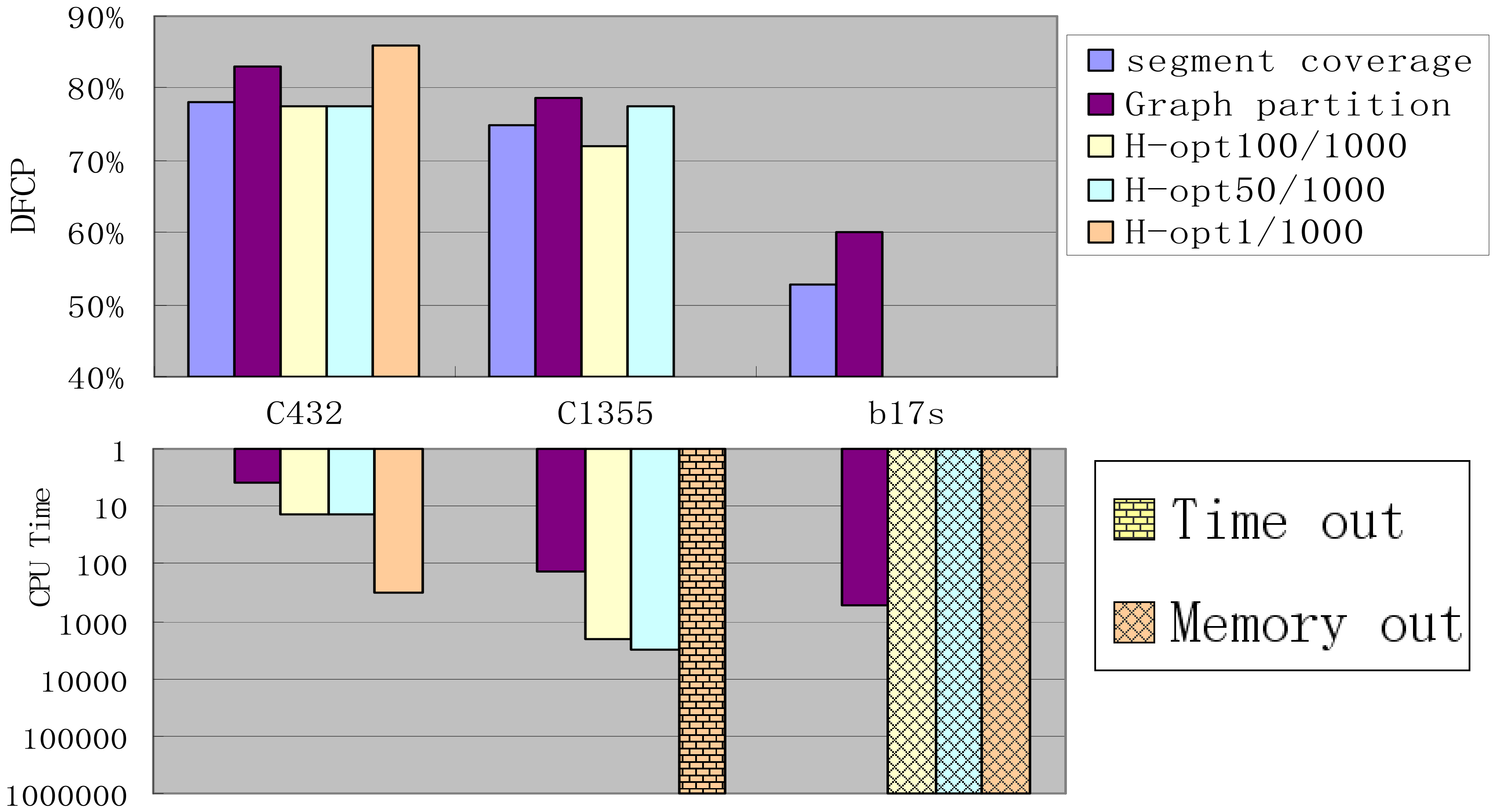
Experimental results(1/2)

- Path selection methods:
  - Segment Coverage
  - H-opt 1/1000, 50/1000, 100/1000*
  - Graph Partition based
- Partial benches

Bench Name	Critical Node Number	Path Number
C432	89	24,786
C1355	426	638,976
b17s	744	133,374,144

* :Critical path selection for delay fault testing based upon a statistical timing model. *IEEE Trans. CAD*

Experimental results(2/2)



C432: 50 paths selected
C1355: 500 paths selected
b17s: 500 paths selected

Conclusions & Future work

- Graph-partition base path selection algorithm
  - Cut-node partition based path selection
  - Independent path set partition based path selection
- Advantage
  - Square root level computing complexity
  - Almost the same DFCP compared to MC-based method
- Limitation
  - Can not guaranty the testability of selected paths
- Future work
  - Redundant paths selection for testability

Thank You ! ☺