CLOCK TREE OPTIMIZATION FOR ELECTROMAGNETIC COMPATIBILITY (EMC)

Xuchu Hu, Matthew R. Guthaus
University of California Santa Cruz

http://vlsida.soe.ucsc.edu

Outline

- Background
- Problem formulation
- Van Ginneken's dynamic programming
- Our algorithm
- Experimental results
- Conclusion

Outline

- **Background**
- Problem formulation
- Van Ginneken’s dynamic programming
- Our algorithm
- Experimental results
- Conclusion

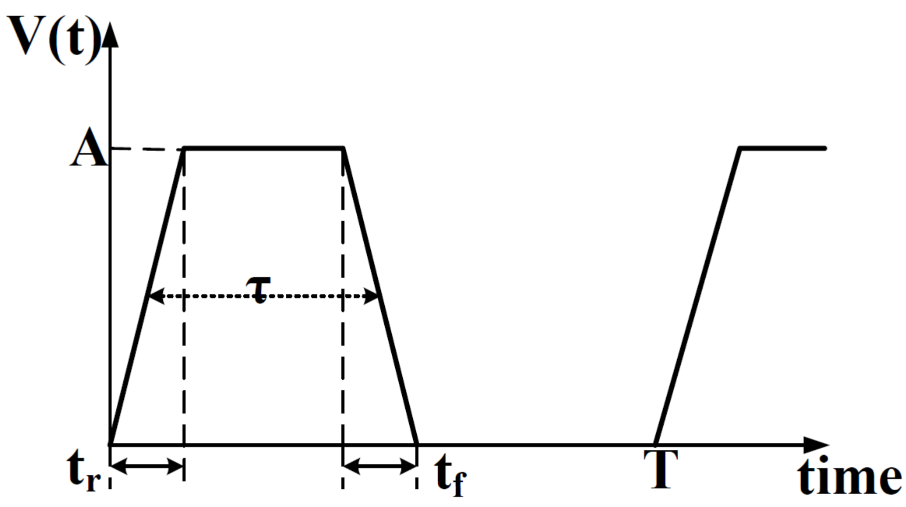
Electromagnetic Interference (EMI)

- On-chip Electromagnetic Interference (EMI)
  - Signal integrity, reflections
  - Simultaneous switching noise (SSN)
- Previous work
  - Decoupling capacitances
  - Random jitter insertion, spread spectrum
  - Clock design
    - Polarity of buffers
    - Skew
    - Reduce buffer sizes

} Manual optimization
NOT CTS

Spectral Analysis of Clock Signal

- Clock in time domain
  - Period T, frequency f, amplitude A
  - Rising/falling time, t_r/t_f
  - duty cycle $D = \tau/T$



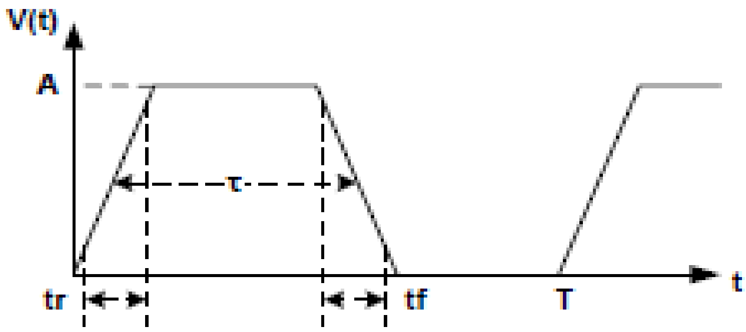
Spectral Analysis of Clock Signal

- Clock in frequency domain
  - Fourier analysis
  - Periodic function → sum of a series of sine and cosine functions

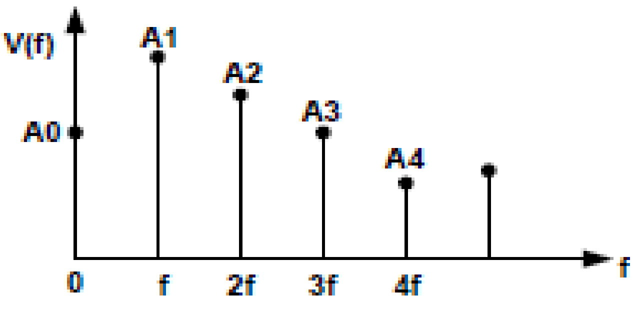
$$V(t) = A_0 + A_1\cos(\omega_0 t + \phi_1) + A_2\cos(\omega_0 t + \phi_2) + A_3\cos(\omega_0 t + \phi_3) + \ldots$$

$$A_0 = \frac{1}{T}\int_0^T v(t)dt = A\frac{\tau}{T}$$

$$|A_n| = 2A\frac{\tau}{T}\left|\frac{\sin(n\pi\tau/T)}{n\pi\tau/T}\right|\left|\frac{\sin(n\pi t_r/T)}{n\pi t_r/T}\right|$$



Clock signal in time domain



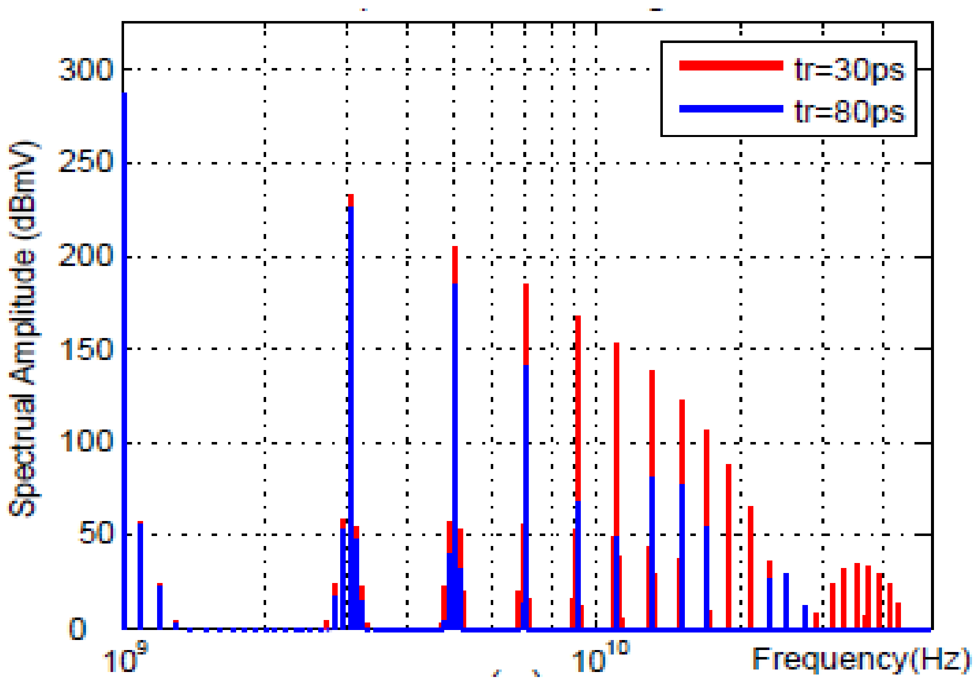
Clock signal in frequency domain

Spectral Analysis of Clock Signal

- Majority of spectrum power lies below f_{max}

$$f_{\max} \approx 1/t_r$$

- High-frequency spectrum: ***rising/fall time, t_r/t_f***



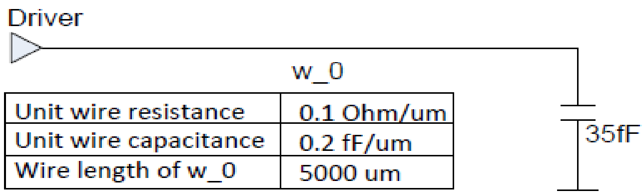
Spectrum of clock signal

t_r=30ps f_{max}=31.8G

t_r=80ps f_{max}=11.9G

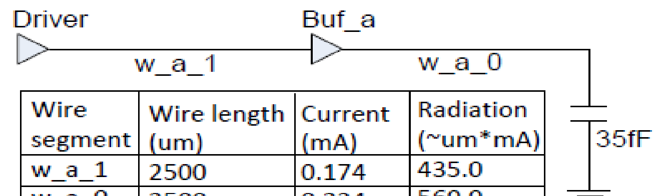
Radiation Emission

- Radiation emission
  - Radiation power ∝ frequency × length × current
  - Frequency → fixed
  - Total wire length → fixed
- Different buffer locations → different radiation



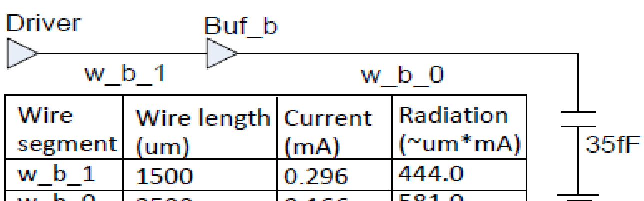
Unit wire resistance	0.1 Ohm/um
Unit wire capacitance	0.2 fF/um
Wire length of w_0	5000 um

(a)



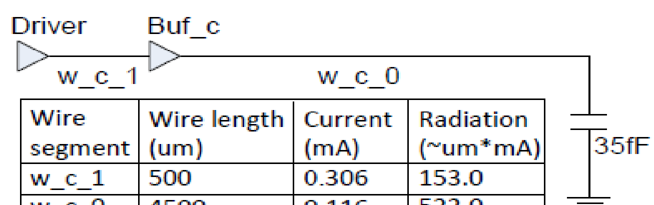
Wire segment	Wire length (um)	Current (mA)	Radiation (~um*mA)
w_a_1	2500	0.174	435.0
w_a_0	2500	0.224	560.0
Total			995.0

(b)



Wire segment	Wire length (um)	Current (mA)	Radiation (~um*mA)
w_b_1	1500	0.296	444.0
w_b_0	3500	0.166	581.0
Total			1025.0

(c)



Wire segment	Wire length (um)	Current (mA)	Radiation (~um*mA)
w_c_1	500	0.306	153.0
w_c_0	4500	0.116	522.0
Total			675.0

(d)

Outline

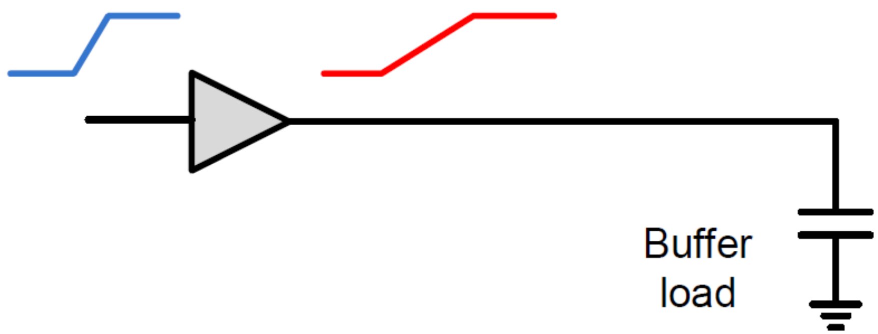
- Background
- **Problem formulation**
- Van Ginneken’s dynamic programming
- Our algorithm
- Experimental results
- Conclusion

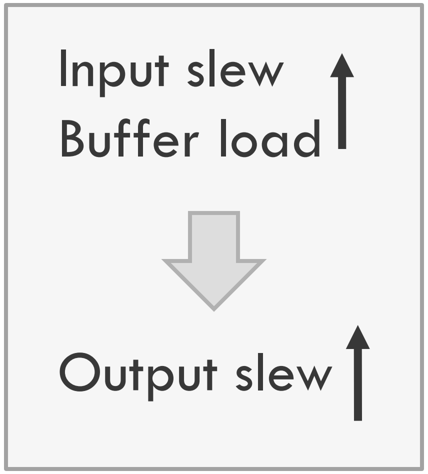
Problem Formulation

- Given:
  - A buffered clock tree T
  - Buffer library $L=\{s_1,s_2,s_3,\ldots,s_n\}$
  - Maximum slew rate constraint S_{max}: noise immunity
  - Minimum slew rate constraint S_{min}: EMC
- Output:
  - Minimize skew, power (traditional metrics in CTS)
  - Location p_i and size s_i of each buffer $b_i \in B$
  - Reduce high-frequency spectrum contents

Slew Rates Constraints

- Problem: control slew rate in a feasible range
  - Buffer output slew rate: input slew, buffer load





- Solution: control the buffer load in a feasible range

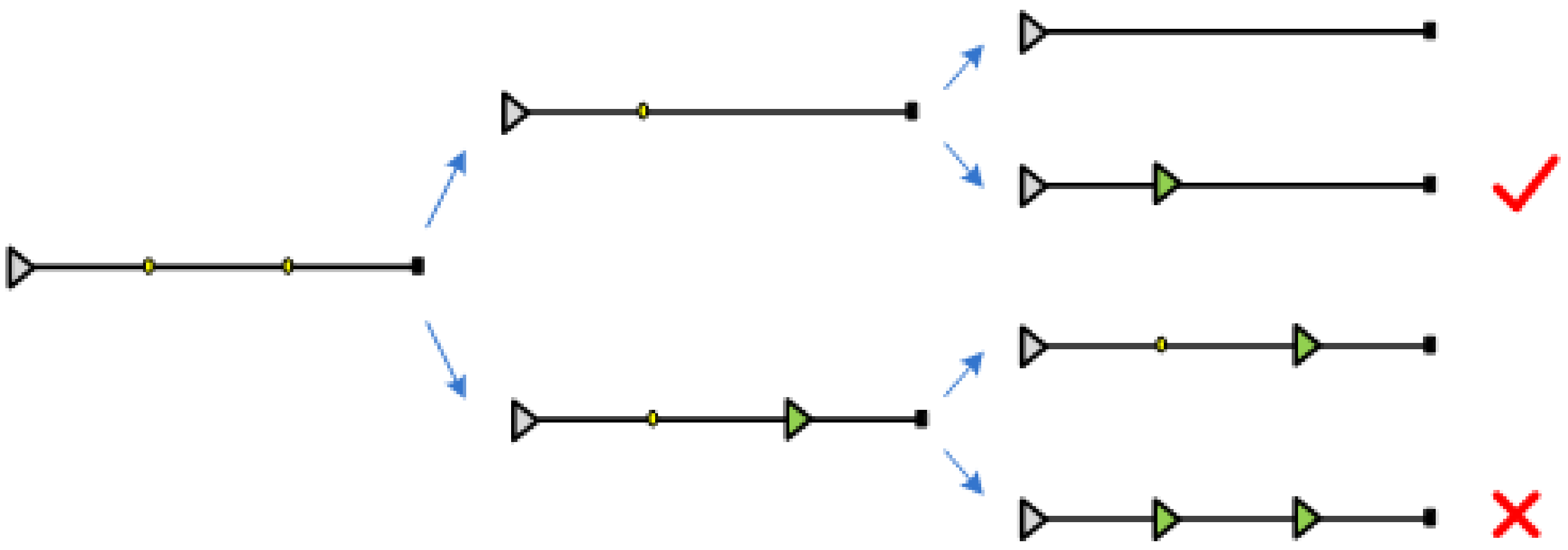
$S_{max} \sim S_{min} \Leftrightarrow C^H \sim C^L$

Outline

- Background
- Problem formulation
- **Van Ginneken’s dynamic programming**
- Our algorithm
- Experimental results
- Conclusion

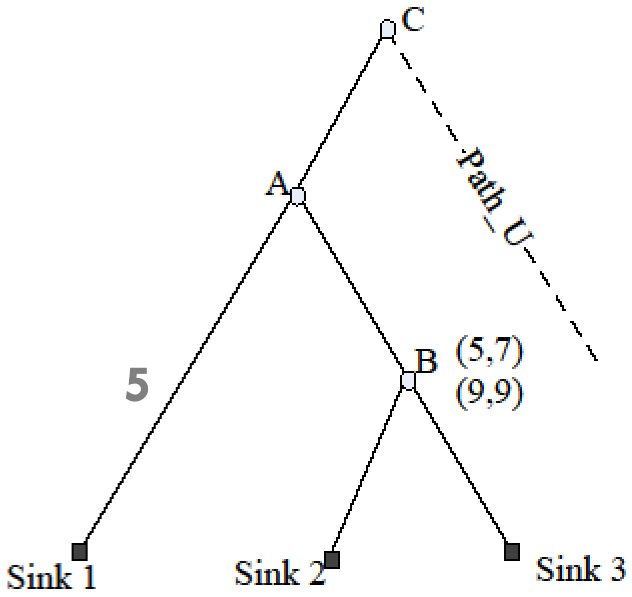
Van Ginneken's Dynamic Programming

- Goal: maximize slack/RAT, minimize delay
- Method:
  - Generate all possible solution
  - Deleted dominated solutions/Pruning
    - worse down stream load cap & worse RAT/delay



Dynamic Programming in CTS

- Clock tree
  - Skew: difference between each pair of sinks
- Pruning
  - Base on delay
  - Base on skew



Outline

- Background
- Problem formulation
- Van Ginneken’s dynamic programming
- **Our algorithm**
- Experimental results
- Conclusion

Top Level Algorithm

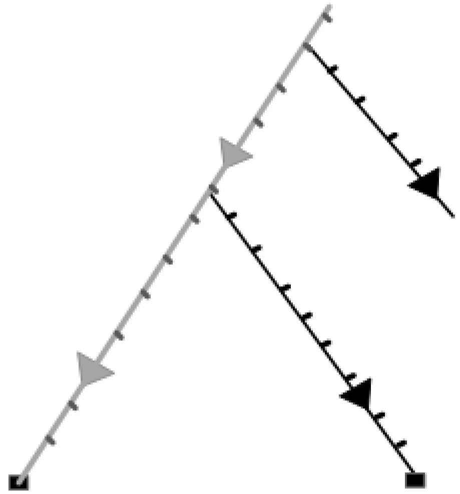
- Dynamic programming
  - ~~Optimize whole tree~~
  - Optimize the critical path
  - Reduces complexity
- All buffers on non-critical path are fixed
- Timing of non-critical path are fixed
- Non-critical path as reference when do pruning

Top Level Algorithm

- Procedure
  - Step 1: segment edges
  - Step 2: relocate and size the critical path (DP)
  - Step 3: update timing
  - Step 4: repeat step2~4 until no improvement

Improvement:

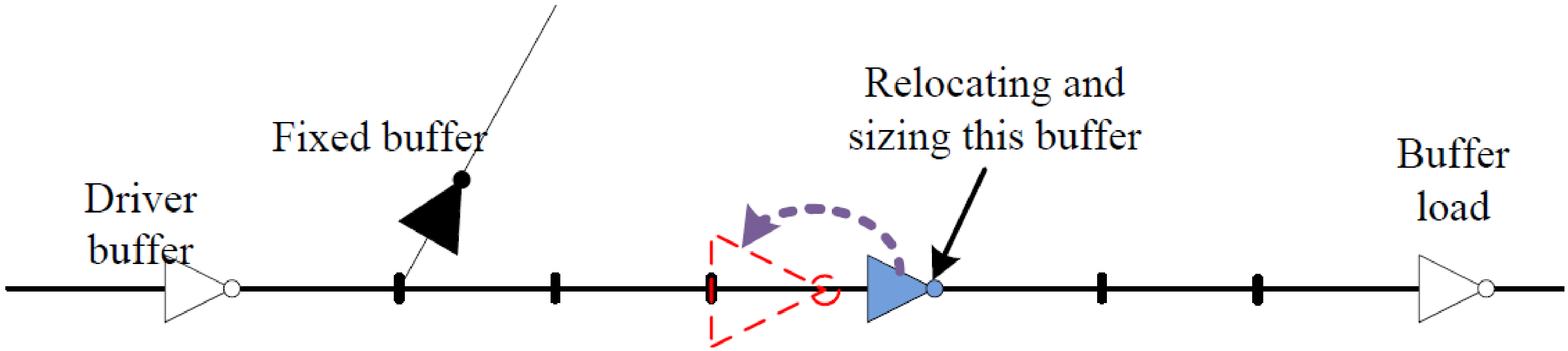
skew, power, radiation, slew rate



Relocating and Sizing the Critical Path

Relocate and size buffers on critical from sink to root

- For each buffer b_i
    - Generate all possible position and size combinations (p_i,s_i)
    - Check the capacitance constraints of each combination
    - If satisfy, save this solution
- End for

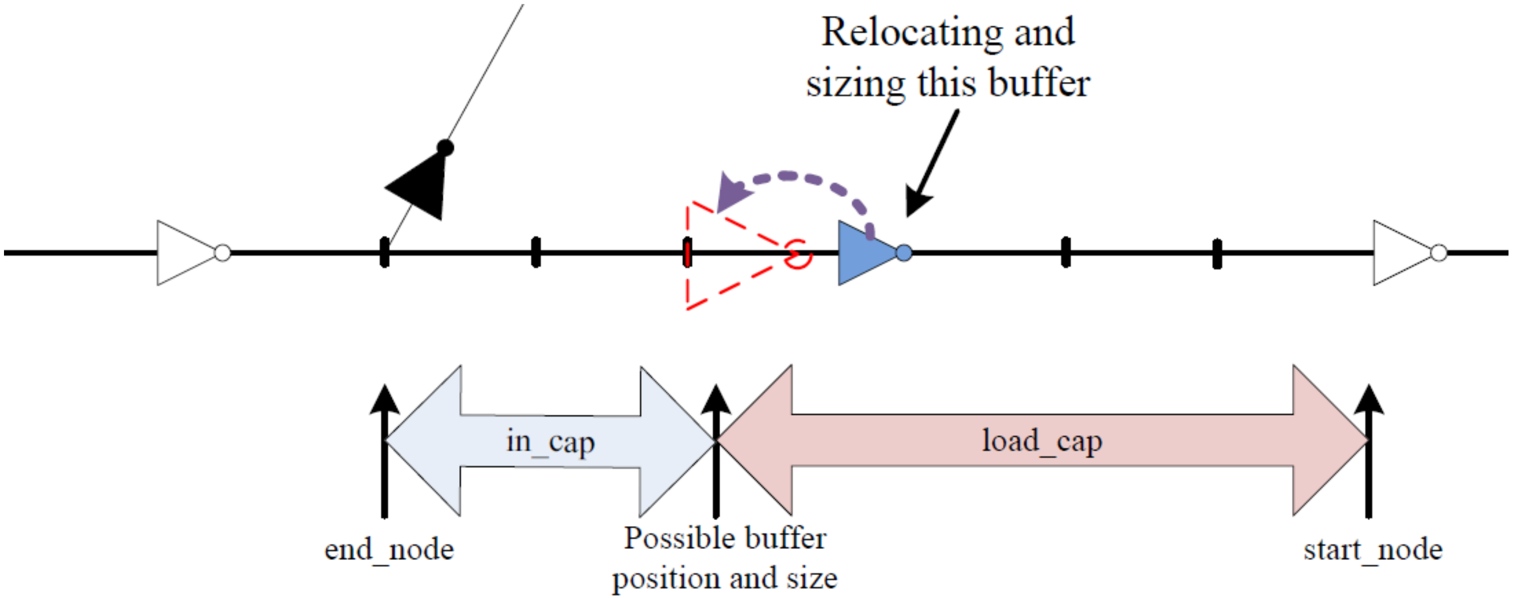


Capacitance Constraints

- Slew rate constraints
- Capacitance constraints

$$C_{l_k}^{L} \le out\_cap \le C_{c_k}^{H}$$

$$in\_cap \le C^{H}$$



Interval Solution Pruning

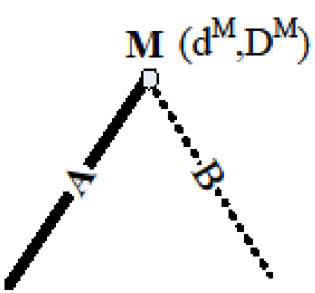
- Only do pruning at merging nodes
- Based on minimum/maximum delay
- Delays in original tree: (d^M, D^M)
- Classify the solutions into two categories:
  - $d = d^M$: A1,A2,A3
  - $d < d^M$: B1,B2

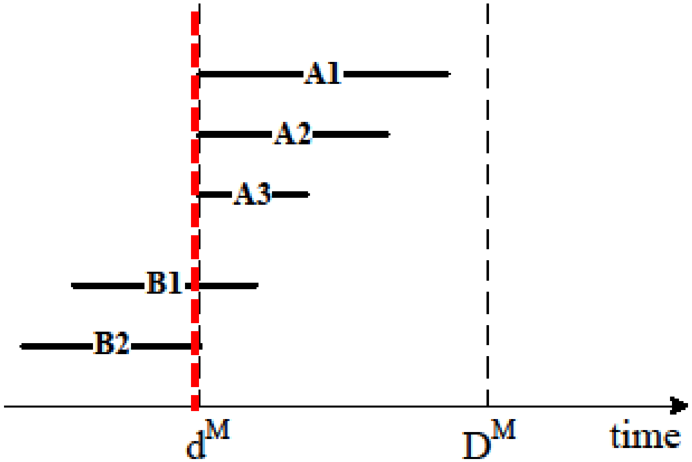
Optimizing a maximum path

Path A: maximum delay path

Path B: fixed path

M : merging node





Outline

- Background
- Problem formulation
- Van Ginneken’s dynamic programming
- Our algorithm
- **Experimental results**
- Conclusion

Experimental Setup

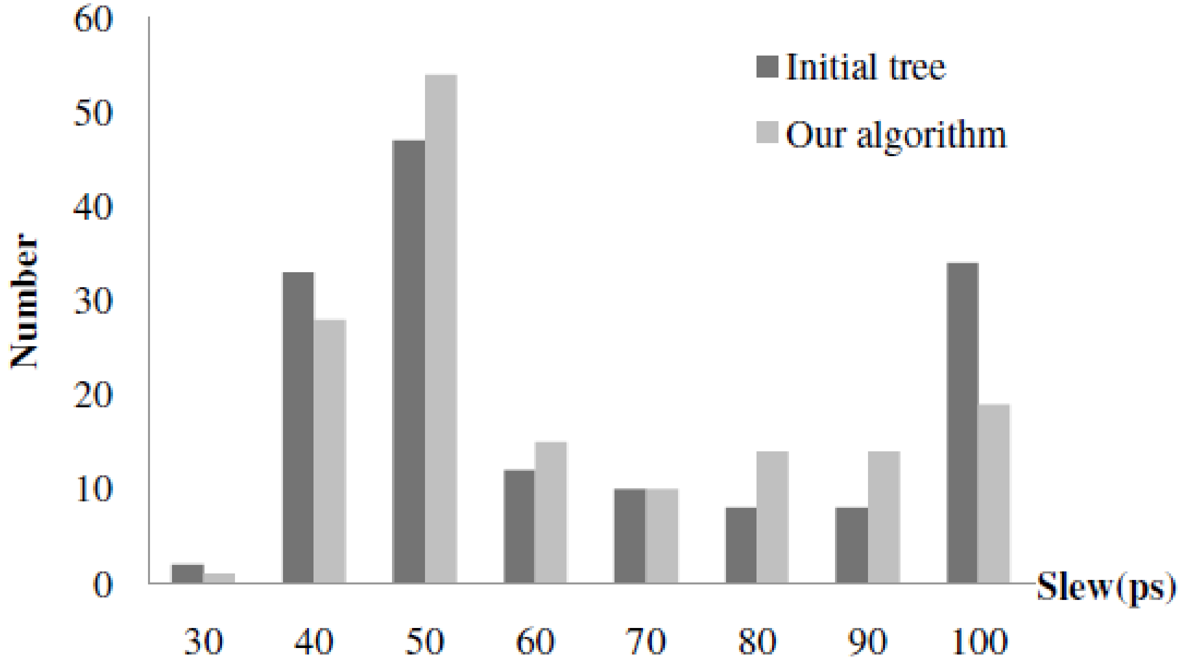
- 45nm technology
- ISCAS benchmarks
- Clock frequency: 1GHz
- Minimum slew: S_{min} = 50ps
- Maximum slew: S_{max} = 100ps
- Four different buffers in buffer library L

Experimental Results

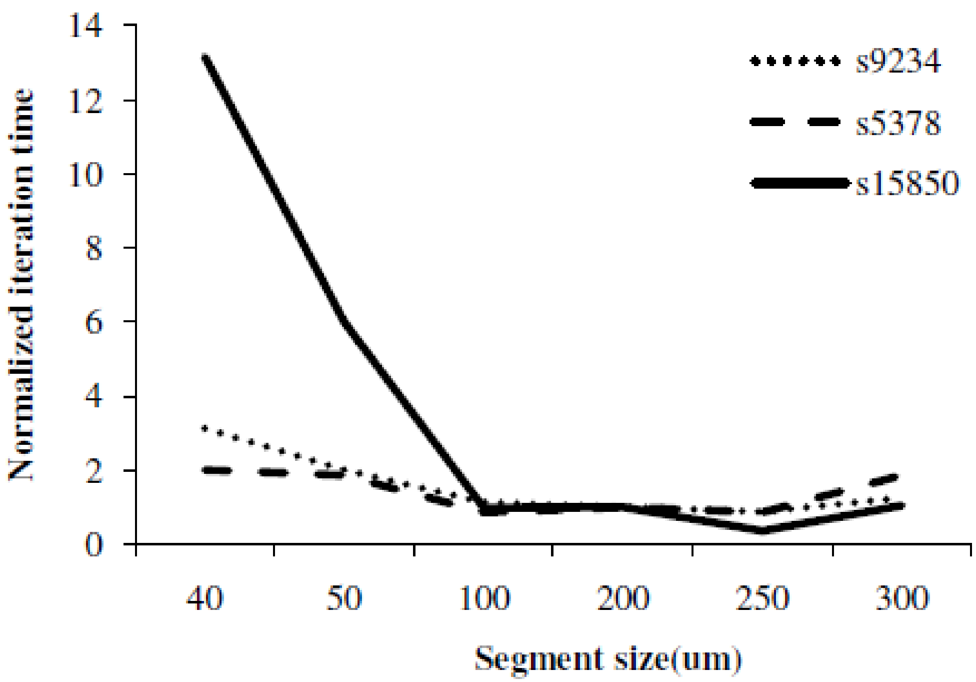
Benchmark		Initial tree				Our method					
Name	Sink	Power	Radiation	Skew	Slew $< 50ps$ (%)	Power	Radiation	Skew	Slew $< 50ps$ (%)	CPU (s)	Iteration (#)
s5378	179	1.0	1.00	1.00	44	1.02	0.99	0.57	33	0.7	10
s9234	211	1.0	1.00	1.00	43	1.01	0.95	0.49	31	1.5	10
s13207	638	1.0	1.00	1.00	50	1.00	0.88	0.74	30	9.9	9
s15850	534	1.0	1.00	1.00	38	1.00	0.88	0.78	28	5.9	4
s35932	1728	1.0	1.00	1.00	38	1.00	0.94	0.63	32	203.2	19
s38584	1426	1.0	1.00	1.00	64	1.00	0.98	0.81	48	90.7	8
Average	786	1.0	1.00	1.00	46	1.01	0.94	0.67	34	52.0	10

- Power: 1% ↑
- Skew: 33% ↓
- Radiation: 6%↓
- Slew <50ps: 46% →34%
- Run time: 52s

Slew distribution



Segment size



Conclusions

- Comprehensive analysis of EMI in clock tree
- A solution for EMI reduction
- Consider both maximum and minimum buffer slew rate
- An incremental dynamic programming in CTS

THANKS !