Fast Dual-V_{dd} Buffering Based on Interconnect Prediction and Sampling

Yu Hu King Ho Tam Tom Tong Jing Lei He

Electrical Engineering Department
University of California at Los Angeles

System Level Interconnect Prediction (SLIP), 2007

Outline

Motivation

1. Aggressive buffering increases interconnect power
   - 35% cells are buffers at 65nm technology [Saxena, TCAD'04]
2. Previous work for singal-V_{dd} buffering
   - Power-optimal single-V_{dd} buffer insertion [Lillis, JSSC'96]
   - Delay-optimal buffered tree generation [Cong, DAC'00; Alpert, TCAD'02]
3. Previous work for dual-V_{dd} buffering
   - Power-optimal dual-V_{dd} buffer insertion and buffered tree construction [Tam, DAC'05]
4. Problem remains
   - Power aware buffering suffers from the expensive computational cost
   - Dual-V_{dd} buffers dramatically increase computational complexity

Motivation

1. Aggressive buffering increases interconnect power
   - 35% cells are buffers at 65nm technology [Saxena, TCAD'04]
2. Previous work for singal-V_{dd} buffering
   - Power-optimal single-V_{dd} buffer insertion [Lillis, JSSC'96]
   - Delay-optimal buffered tree generation [Cong, DAC'00; Alpert, TCAD'02]
3. Previous work for dual-V_{dd} buffering
   - Power-optimal dual-V_{dd} buffer insertion and buffered tree construction [Tam, DAC'05]
4. Problem remains
   - Power aware buffering suffers from the expensive computational cost
   - Dual-V_{dd} buffers dramatically increase computational complexity

Motivation

1. Aggressive buffering increases interconnect power
   - 35% cells are buffers at 65nm technology [Saxena, TCAD'04]
2. Previous work for singal-V_{dd} buffering
   - Power-optimal single-V_{dd} buffer insertion [Lillis, JSSC'96]
   - Delay-optimal buffered tree generation [Cong, DAC'00; Alpert, TCAD'02]
3. Previous work for dual-V_{dd} buffering
   - Power-optimal dual-V_{dd} buffer insertion and buffered tree construction [Tam, DAC'05]
4. Problem remains
   - Power aware buffering suffers from the expensive computational cost
   - Dual-V_{dd} buffers dramatically increase computational complexity

Motivation

1. Aggressive buffering increases interconnect power
   - 35% cells are buffers at 65nm technology [Saxena, TCAD'04]
2. Previous work for singal-V_{dd} buffering
   - Power-optimal single-V_{dd} buffer insertion [Lillis, JSSC'96]
   - Delay-optimal buffered tree generation [Cong, DAC'00; Alpert, TCAD'02]
3. Previous work for dual-V_{dd} buffering
   - Power-optimal dual-V_{dd} buffer insertion and buffered tree construction [Tam, DAC'05]
4. Problem remains
   - Power aware buffering suffers from the expensive computational cost
   - Dual-V_{dd} buffers dramatically increase computational complexity

Major Constributions

1. Focus on speedup for power-aware dual-V_{dd} buffer insertion and buffered tree construction
2. Propose three speedup techniques for power optimized dual-V_{dd} buffer insertion based on interconnect prediction and sampling
   - Pre-buffer Slack Pruning (PSP) extended from the one presented in [Shi, DAC'03]
   - Predictive Min-delay Pruning (PMP)
   - 3D sampling
   - Runtime grows linearly *w.r.t.* the tree-size and we achieve 50x speedup compared with [Tam, DAC'05]
3. Incorporate the fast buffer insertion and grid reduction in power optimized buffered tree construction algorithm

Major Constributions

1. Focus on speedup for power-aware dual-V_{dd} buffer insertion and buffered tree construction
2. Propose three speedup techniques for power optimized dual-V_{dd} buffer insertion based on interconnect prediction and sampling
   - Pre-buffer Slack Pruning (PSP) extended from the one presented in [Shi, DAC'03]
   - Predictive Min-delay Pruning (PMP)
   - 3D sampling
   - Runtime grows linearly *w.r.t.* the tree-size and we achieve 50x speedup compared with [Tam, DAC'05]
3. Incorporate the fast buffer insertion and grid reduction in power optimized buffered tree construction algorithm

Major Constributions

1. Focus on speedup for power-aware dual-V_{dd} buffer insertion and buffered tree construction
2. Propose three speedup techniques for power optimized dual-V_{dd} buffer insertion based on interconnect prediction and sampling
   - Pre-buffer Slack Pruning (PSP) extended from the one presented in [Shi, DAC'03]
   - Predictive Min-delay Pruning (PMP)
   - 3D sampling
   - Runtime grows linearly *w.r.t.* the tree-size and we achieve 50x speedup compared with [Tam, DAC'05]
3. Incorporate the fast buffer insertion and grid reduction in power optimized buffered tree construction algorithm

Delay, Slew and Power Modeling

1. Elmore Delay Model
   - The delay of wire with length l is

$$d(l) = (\frac{1}{2} \cdot c_w \cdot l + c_{load}) \cdot r_w \cdot l \quad (1)$$

   - The delay of a buffer d_{buf} is

$$d_{buf} = d_{int} + r_o \cdot c_{load} \quad (2)$$

   - Bakoglus slew metric (Elmore delay times $ln9$)
2. Power Model
   - Wire power dissipation

$$E_w = 0.5 \cdot c_w \cdot l \cdot V_{dd}^2 \quad (3)$$

   - Lumped buffer dynamic/short-circuit power
   - Can be easily extended to leakage power

Dual-V_{dd} Buffering

Intuitions for power aware dual-V_{dd} buffering

1. High V_{dd} buffers drive critical interconnect edges for timing optimization
2. Low V_{dd} buffers drive non-critical interconnect edges for power
3. Achieves power saving since power is proportional to $\alpha \cdot V_{dd}^2$
4. Suffer no loss of delay optimality

Constraints in dual-V_{dd} buffering

1. Disallowing low V_{dd} drives high V_{dd}
   - Not affect optimality [Tam, DAC'05]
   - Exclude power and delay overhead of level converters

Problem Formulation

Dual-V_{dd} buffer insertion and sizing (dBIS)

1. Given
   - An interconnect fanout tree with source n_{src}, sink nodes n_s and Steiner points n_p
   - Buffer stations n_b
2. Find
   - A buffer size and V_{dd} level assignment
3. Such that (objective)
   - The RAT q_n^{src} at the source n_{src} is met
   - The power consumed by the interconnect tree is minimized
   - Slew rate at every input of the buffers and the sinks n_s is upper bounded by the slew rate bound $\bar{s}$

Problem Formulation

Dual V_{dd} Buffered Tree Construction (dTree)

1. Given
   - An un-routed net with source n_{src} and sink nodes n_s
   - Buffer stations n_b
2. Find
   - A routing for the buffered tree
   - A buffer size and V_{dd} level assignment
3. Such that (objective)
   - The RAT q_n^{src} at the source n_{src} is met
   - The power consumed by the interconnect tree is minimized
   - Slew rate at every input of the buffers and the sinks n_s is upper bounded by the slew rate bound $\bar{s}$

Baseline Algorithm Flow

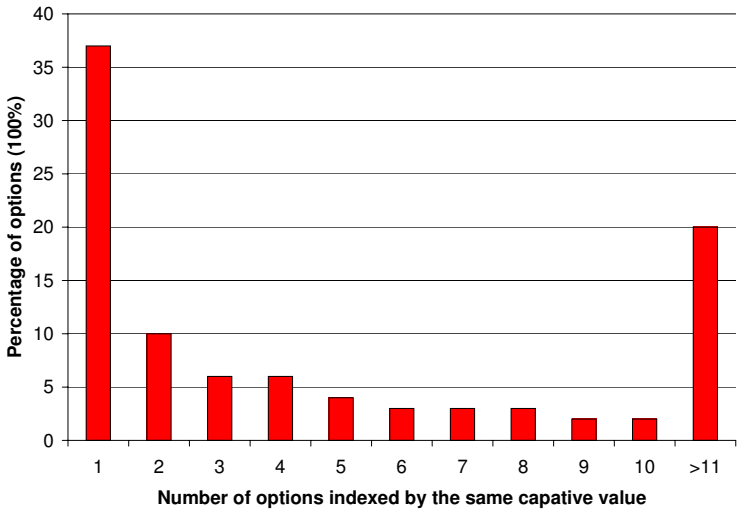
1. Based on [Lillis, JSSC'96]
2. Dynamic programming with partial solution (option) pruning
3. Options must now record downstream V_{dd} levels for buffering to prevent $V_{dd}L \Rightarrow V_{dd}H$, which removes unnecessary search in solution space
4. Still quite slow for large nets

Definition (**Domination**)

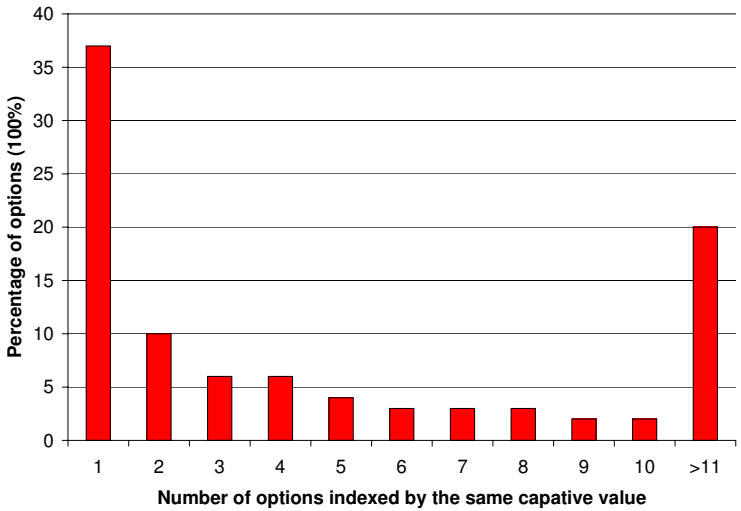
In node n, option $\Phi_1 = (rat_1, cap_1, pwr_1, \theta)$ dominates $\Phi_2 = (rat_2, cap_2, pwr_2, \theta)$, if $rat_1 \geq rat_2$, $cap_1 \leq cap_2$, and $pwr_1 \leq pwr_2$

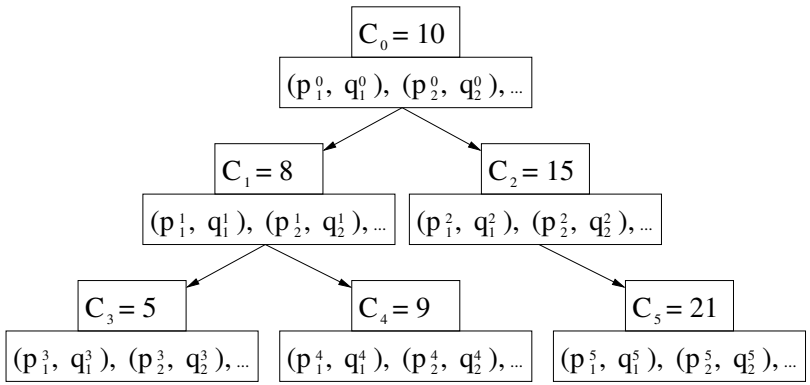
1. Options are indexed by capacitive values
2. Few options left after power-delay sampling [Tam, DAC'05] under the same capacitive index
3. A linear list is used to store options under the same capacitive value
4. Capacitive values are organized by a binary search tree

1. Options are indexed by capacitive values
2. Few options left after power-delay sampling [Tam, DAC'05] under the same capacitive index
3. A linear list is used to store options under the same capacitive value
4. Capacitive values are organized by a binary search tree



1. Options are indexed by capacitive values
2. Few options left after power-delay sampling [Tam, DAC'05] under the same capacitive index
3. A linear list is used to store options under the same capacitive value
4. Capacitive values are organized by a binary search tree





Motivation for 3D Sampling

1. Power-delay sampling has shown effectiveness [Tam, DAC'05]
2. The size of the third dimenstion (capacitive values) increases significantly for large testcases

node#	sink#	> 100	> 50
515	299	14%	62%
784	499	17%	64%
1054	699	28%	65%
1188	799	33%	71%

3. Sampling on all dimensions in power-delay-capacitance solution space is necessary

Motivation for 3D Sampling

1. Power-delay sampling has shown effectiveness [Tam, DAC'05]
2. The size of the third dimenstion (capacitive values) increases significantly for large testcases

node#	sink#	> 100	> 50
515	299	14%	62%
784	499	17%	64%
1054	699	28%	65%
1188	799	33%	71%

3. Sampling on all dimensions in power-delay-capacitance solution space is necessary

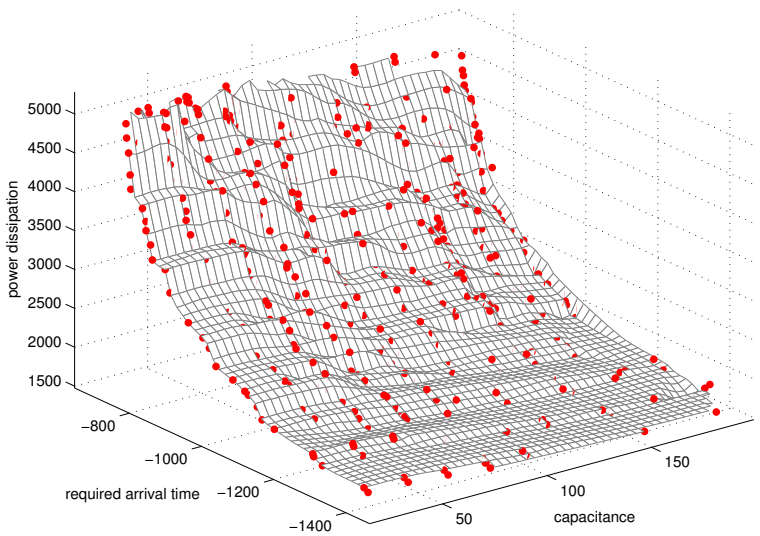
Motivation for 3D Sampling

1. Power-delay sampling has shown effectiveness [Tam, DAC'05]
2. The size of the third dimenstion (capacitive values) increases significantly for large testcases

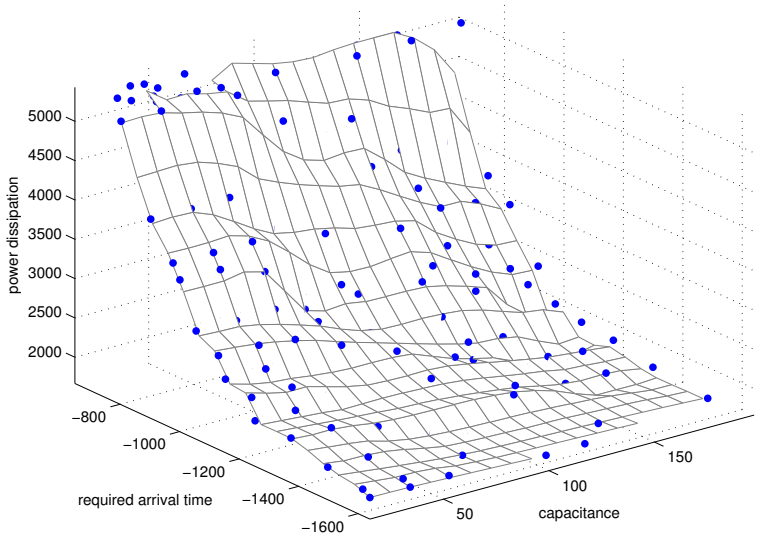
node#	sink#	> 100	> 50
515	299	14%	62%
784	499	17%	64%
1054	699	28%	65%
1188	799	33%	71%

3. Sampling on all dimensions in power-delay-capacitance solution space is necessary

The idea of 3D sampling is to pick only a certain number of options among all options uniformly over the power-delay-capacitance space for upstream propagation



(a) 2D sampling



(b) 3D sampling

Pre-buffer Slack Pruning (PSP) [Shi, ASPDAC'05]

Suppose R_{min} is the minimal resistance in the buffer library. For two non-redundant options $\Phi_1 = (rat_1, cap_1, pwr_1, \theta_1)$ and $\Phi_2 = (rat_2, cap_2, pwr_2, \theta_2)$, where $rat_1 < rat_2$ and $cap_1 < cap_2$, then Φ_2 is pruned, if $(rat_2 - rat_1)/(cap_2 - cap_1) \geq R_{min}$

Extension for Dual-V_{dd} PSP

1. Choose a proper high / low V_{dd} buffer resistance R_H / R_L for PSP to avoid overly aggressive pruning
2. If $\theta = true$, $R_{min} = R_H$. Otherwise, $R_{min} = R_L$.

1. Assume a continuous number of buffers and buffer sizes
2. The optimum unit length delay, $delay_{opt}$, is given by [Bakoglou, book, 1990]

$$delay_{opt} = 2\sqrt{r_s c_o rc}(1 + \sqrt{\frac{1}{2}(1 + \frac{c_p}{c_o})}) \quad (4)$$

3. (4) calculates the lower bound of delay from any node to the source

Predictive min-delay pruning (PMP)

A newly generated option $\Phi = (rat, cap, pwr, \theta)$ is pruned if $rat - delay_{opt} \cdot dis(v) < RAT_0$, where RAT_0 is the target RAT at the source, $dis(v)$ is the distance of the node-to-source path

Experimental Settings

1. Extract technical parameters by BSIM4 and SPICE under 65nm

Table: Settings for the 65nm global interconnect.

Settings	Values
Interconnect	$r_w = 0.186\Omega/\mu m$, $c_w = 0.0519fF/\mu m$
$V_{dd}H$ Buffer (min size)	$c_{in} = 0.47fF$, $V_{dd}^H = 1.2V$ $r_o^H = 4.7k\Omega$, $d_b^H = 72ps$, $E_b^H = 84fJ$
$V_{dd}L$ Buffer (min size)	$c_{in} = 0.47fF$, $V_{dd}^L = 0.9V$ $r_o^L = 5.4k\Omega$, $d_b^L = 98ps$, $E_b^L = 34fJ$
Level converter (min size)	$c_{in} = 0.47fF$, $E_{LC} = 5.7fJ$ $d_{LC} = 220ps$

2. Randomly generate and route 10 nets by GeoSteiner

Experimental Settings

1. Extract technical parameters by BSIM4 and SPICE under 65nm

Table: Settings for the 65nm global interconnect.

Settings	Values
Interconnect	$r_w = 0.186\Omega/\mu m$, $c_w = 0.0519fF/\mu m$
$V_{dd}H$ Buffer (min size)	$c_{in} = 0.47fF$, $V_{dd}^H = 1.2V$ $r_o^H = 4.7k\Omega$, $d_b^H = 72ps$, $E_b^H = 84fJ$
$V_{dd}L$ Buffer (min size)	$c_{in} = 0.47fF$, $V_{dd}^L = 0.9V$ $r_o^L = 5.4k\Omega$, $d_b^L = 98ps$, $E_b^L = 34fJ$
Level converter (min size)	$c_{in} = 0.47fF$, $E_{LC} = 5.7fJ$ $d_{LC} = 220ps$

2. Randomly generate and route 10 nets by GeoSteiner

	runtime (s)					delay				power			
net	DVB	sam	pmp+sam	psp+sam	all	sam	pmp+sam	psp+sam	all	sam	pmp+sam	psp+sam	all
s1	36	15	2	3	1	1.01	1.01	1.01	1.01	1.06	1.00	1.06	0.99
s2	62	19	4	5	2	1.01	1.00	1.01	1.00	0.98	1.01	0.97	1.00
s3	96	36	10	7	4	1.02	1.00	1.01	1.00	0.94	0.98	0.96	0.98
s4	264	50	16	14	6	1.02	1.00	1.01	1.00	1.00	0.99	1.01	1.04
s5	640	71	46	32	20	1.05	1.00	1.03	1.01	0.99	0.99	0.95	0.98
s6	987	101	77	42	34	1.06	1.01	1.03	1.01	1.04	1.00	1.05	1.01
s7	2232	209	135	80	59	1.08	1.00	1.06	0.99	0.98	0.95	1.00	0.99
s8	3427	309	219	127	89	1.08	1.00	1.07	1.00	0.99	1.00	0.95	0.97
s9	5625	327	256	133	95	1.08	1.01	1.08	1.01	1.04	1.03	1.02	1.03
ave	1485	128	85	49	34	1.06	1.00	1.04	1.00	1.00	0.99	1.00	1.00
	1	$\frac{1}{10}$	$\frac{1}{15}$	$\frac{1}{30}$	$\frac{1}{50}$								

1. Sampling grid size is 20x20x20 for 3D sampling and 20x20 for DVB
2. Accompanied with substantial speedup, 3D sampling brings significant error for large testcases
3. Combining PSP/PMP with 3D sampling improves runtime and solution quality
4. PSP / PMP prunes many redundant options and 3D sampling can always select option samples from a good candidate pool

	runtime (s)					delay				power			
net	DVB	sam	pmp+sam	psp+sam	all	sam	pmp+sam	psp+sam	all	sam	pmp+sam	psp+sam	all
s1	36	15	2	3	1	1.01	1.01	1.01	1.01	1.06	1.00	1.06	0.99
s2	62	19	4	5	2	1.01	1.00	1.01	1.00	0.98	1.01	0.97	1.00
s3	96	36	10	7	4	1.02	1.00	1.01	1.00	0.94	0.98	0.96	0.98
s4	264	50	16	14	6	1.02	1.00	1.01	1.00	1.00	0.99	1.01	1.04
s5	640	71	46	32	20	1.05	1.00	1.03	1.01	0.99	0.99	0.95	0.98
s6	987	101	77	42	34	1.06	1.01	1.03	1.01	1.04	1.00	1.05	1.01
s7	2232	209	135	80	59	1.08	1.00	1.06	0.99	0.98	0.95	1.00	0.99
s8	3427	309	219	127	89	1.08	1.00	1.07	1.00	0.99	1.00	0.95	0.97
s9	5625	327	256	133	95	1.08	1.01	1.08	1.01	1.04	1.03	1.02	1.03
ave	1485	128	85	49	34	1.06	1.00	1.04	1.00	1.00	0.99	1.00	1.00
	1	$\frac{1}{10}$	$\frac{1}{15}$	$\frac{1}{30}$	$\frac{1}{50}$								

1. Sampling grid size is 20x20x20 for 3D sampling and 20x20 for DVB
2. Accompanied with substantial speedup, 3D sampling brings significant error for large testcases
3. Combining PSP/PMP with 3D sampling improves runtime and solution quality
4. PSP / PMP prunes many redundant options and 3D sampling can always select option samples from a good candidate pool

	runtime (s)					delay				power			
net	DVB	sam	pmp+sam	psp+sam	all	sam	pmp+sam	psp+sam	all	sam	pmp+sam	psp+sam	all
s1	36	15	2	3	1	1.01	1.01	1.01	1.01	1.06	1.00	1.06	0.99
s2	62	19	4	5	2	1.01	1.00	1.01	1.00	0.98	1.01	0.97	1.00
s3	96	36	10	7	4	1.02	1.00	1.01	1.00	0.94	0.98	0.96	0.98
s4	264	50	16	14	6	1.02	1.00	1.01	1.00	1.00	0.99	1.01	1.04
s5	640	71	46	32	20	1.05	1.00	1.03	1.01	0.99	0.99	0.95	0.98
s6	987	101	77	42	34	1.06	1.01	1.03	1.01	1.04	1.00	1.05	1.01
s7	2232	209	135	80	59	1.08	1.00	1.06	0.99	0.98	0.95	1.00	0.99
s8	3427	309	219	127	89	1.08	1.00	1.07	1.00	0.99	1.00	0.95	0.97
s9	5625	327	256	133	95	1.08	1.01	1.08	1.01	1.04	1.03	1.02	1.03
ave	1485	128	85	49	34	1.06	1.00	1.04	1.00	1.00	0.99	1.00	1.00
	1	$\frac{1}{10}$	$\frac{1}{15}$	$\frac{1}{30}$	$\frac{1}{50}$								

1. Sampling grid size is 20x20x20 for 3D sampling and 20x20 for DVB
2. Accompanied with substantial speedup, 3D sampling brings significant error for large testcases
3. Combining PSP/PMP with 3D sampling improves runtime and solution quality
4. PSP / PMP prunes many redundant options and 3D sampling can always select option samples from a good candidate pool

	runtime (s)					delay				power			
net	DVB	sam	pmp+sam	psp+sam	all	sam	pmp+sam	psp+sam	all	sam	pmp+sam	psp+sam	all
s1	36	15	2	3	1	1.01	1.01	1.01	1.01	1.06	1.00	1.06	0.99
s2	62	19	4	5	2	1.01	1.00	1.01	1.00	0.98	1.01	0.97	1.00
s3	96	36	10	7	4	1.02	1.00	1.01	1.00	0.94	0.98	0.96	0.98
s4	264	50	16	14	6	1.02	1.00	1.01	1.00	1.00	0.99	1.01	1.04
s5	640	71	46	32	20	1.05	1.00	1.03	1.01	0.99	0.99	0.95	0.98
s6	987	101	77	42	34	1.06	1.01	1.03	1.01	1.04	1.00	1.05	1.01
s7	2232	209	135	80	59	1.08	1.00	1.06	0.99	0.98	0.95	1.00	0.99
s8	3427	309	219	127	89	1.08	1.00	1.07	1.00	0.99	1.00	0.95	0.97
s9	5625	327	256	133	95	1.08	1.01	1.08	1.01	1.04	1.03	1.02	1.03
ave	1485	128	85	49	34	1.06	1.00	1.04	1.00	1.00	0.99	1.00	1.00
	1	$\frac{1}{10}$	$\frac{1}{15}$	$\frac{1}{30}$	$\frac{1}{50}$								

1. Sampling grid size is 20x20x20 for 3D sampling and 20x20 for DVB
2. Accompanied with substantial speedup, 3D sampling brings significant error for large testcases
3. Combining PSP/PMP with 3D sampling improves runtime and solution quality
4. PSP / PMP prunes many redundant options and 3D sampling can always select option samples from a good candidate pool

Baseline Algorithm

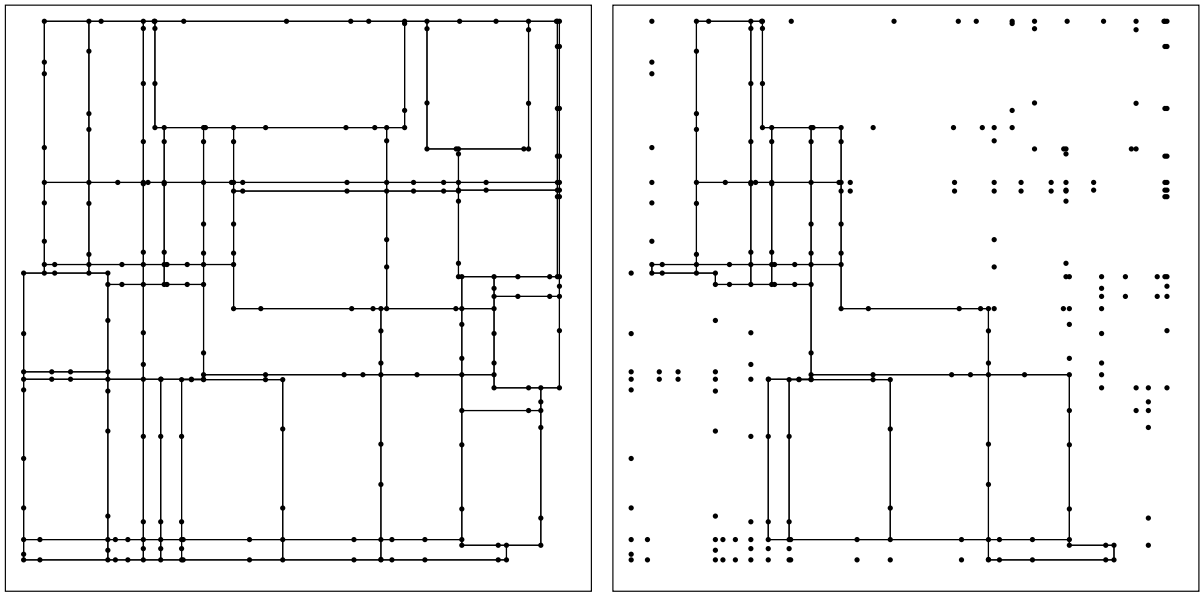
1. Extend the delay optimization buffered tree construction algorithm [Cong, DAC'00]
   - Build Hanan Graph w / buffer insertion nodes according to locations of buffer stations
   - Path search on the grid by option propagation
2. Option growth is exponential
3. Power and dual-V_{dd} buffers further accelerate option growth

Definition (**Domination**)

In node n, option $\Phi_1 = (\mathcal{S}_1, \mathcal{R}_1, rat_1, cap_1, pwr_1, \theta_1)$ dominates $\Phi_2 = (\mathcal{S}_2, \mathcal{R}_2, rat_2, cap_2, pwr_2, \theta_2)$, if $\mathcal{S}_1 \supseteq \mathcal{S}_2$, $rat_1 \geq rat_2$, $cap_1 \leq cap_2$, and $pwr_1 \leq pwr_2$

Routing Grid Reduction

1. Option growth is exponential *w.r.t.* routing grid size
2. Restrict the progation direction always towards the source



(c) Before reduction (d) After reduction

Experimental Results

1. Our Fast sTree / dTree algorithm runs over 100x faster than S-Tree/D-Tree [Tam, DAC'05] within 1% power overhead
2. Be able to solve the buffered tree construction problem on 10 sinks with 400+ buffer stations

test cases				runtime(s)				RAT*(ps)		power(fJ)			
name	n#	s#	nl#	S-Tree	sTree	D-Tree	dTree	S-Tree	sTree	S-Tree	sTree	D-Tree	dTree
grid.2	97	2	36	0	0	0	0	-223	-224	1492	1492	1430	1430
grid.3	165	3	142	19	1	102	5	-604	-608	3908	3456	3907	3456
grid.4	137	4	82	44	2	297	8	-582	-583	3426	3426	3131	3131
grid.5	261	5	162	2849	8	5088	37	-532	-533	4445	4355	3979	3989
grid.6	235	6	143	5200	25	13745	115	397	-399	4919	4718	4860	3718
grid.10	426	10	267	-	2346	-	3605	-	-625	-	7338	-	5915
				1	$< \frac{1}{100}$	1	$< \frac{1}{100}$	1	$> 99\%$	1	$< 101\%$	1	$< 101\%$

Conclusions

- Presented efficient algorithms to dual-V_{dd} buffering for power optimization
- Studied three pruning techniques including PSP, PMP and 3D sampling
- Proposed grid reduction for buffered tree construction speedup
- Experimental results show that we obtain over 50x and 100x speedup compared with the most efficient existing algorithms [Tam, DAC'05] for dual-V_{dd} buffer insertion and buffered tree construction, respectively

Future Work

1. Further improve the efficiency of buffered tree construction by adapting hierarchical tree generation algorithm
2. Slack allocation for more power reduction
   - Chip level FPGA dual-V_{dd} assignment [Lin, DAC'05]
     - Fix buffer location, assign V_{dd} levels
     - Consider multiple critical path
     - Solve as a linear programming problem
   - More freedom of ASIC buffering introduces more challenge to algorithms