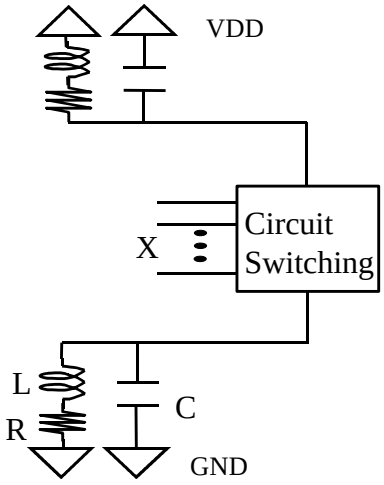
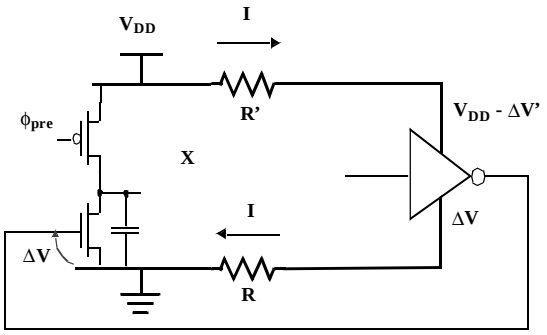
Power Supply Networks: Analysis and Synthesis

What is Power Supply Noise?

- Problem: Degraded voltage level at the delivery point of the power/ground grid causes performance and/or functional failure
  - Lower supply voltage slows the circuit down
  - Lower supply voltage can inhibit switching and loss of state
  - Voltage fluctuation causes noise injection in the circuit



Logic Failure due to IR drop



Trends

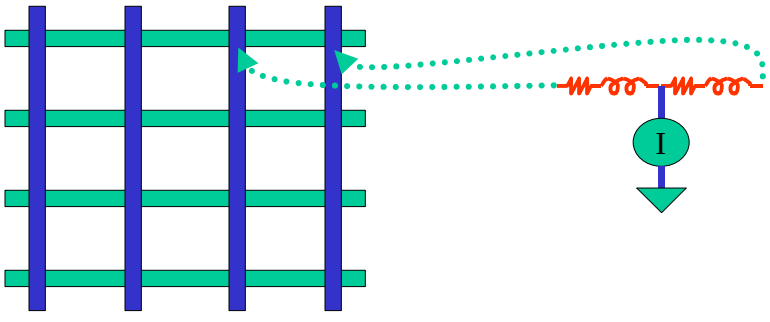
- Process shrink: increased current density
- Lower supply voltage: decreased voltage margin
- Increased frequency: rate of change of current increases
- Increased complexity: large die size increases the routing length of power supply
- New packaging methods: new bonding mathods (flip-chip bump) improves the drops

Issues in PSN Analysis

- On-chip resistance (R) and inductance (L) for P/G network
- Worst case noise does not correspond to average current, or peak current
- Small things add up
  - Each gate draws a small current pulse when switching
- Switching events and their spatio-temporal correlation
  - Find the simulation trace that creates a switching pattern in the design resulting in the worst case voltage drop at the specific location in the grid
- Conservative: Approach must err on the side of predicting too much voltage drop

Design Planning

- Chip planning will occur before a definite floorplan
  - Current is estimated based on chip area
  - Assume a equal distribution of power sources and power grid



Early Analysis

- Initial floorplan and global power grid are complete
- Global power grid is extracted with R's, L's, and C's
- Each block is modeled as a single current source based on an estimated DC-value or on the gate level implementation

Late Analysis

- Both global and local power grids are extracted
- Current sources are modeled at the transistor or gate level

Simulation Method

- Decouple simulation of interconnect from the circuit
- Characterize the switching current of a gate/transistor
- Sampling frequency allows for run-time/accuracy trade-off
- Use a switch-level or gate-level simulator to generate switching events
- Iteration allows for reduced conservatism

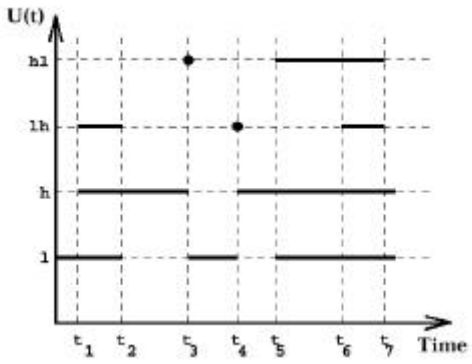
Issues of Simulation Method

- Strengths:
  - Accuracy of model
  - Simple integration with existing tools
- Weaknesses:
  - Simulation speed is not adequate for full chip microprocessor designs
  - Confidence of covering the worst case event with a test vector is not known
  - Large test vectors are needed, resulting in long run times

Static Approach

- Model the current for a block/gate for a single clock cycle
  - Use timing windows from timing analysis to model gate switching
  - Apply gate switching current for the entire duration of the window
  - Sum current of each gate to obtain a block current for early analysis

Improved Window Generation



AND	l	h	hl	lh
l	l	l	l	l
h	l	h	hl	lh
hl	l	hl	hl	l
lh	l	lh	l	lh

OR	l	h	hl	lh
l	l	h	hl	lh
h	h	h	h	h
hl	hl	h	hl	h
lh	lh	h	h	lh

NOT	
l	h
h	l
hl	lh
lh	hl

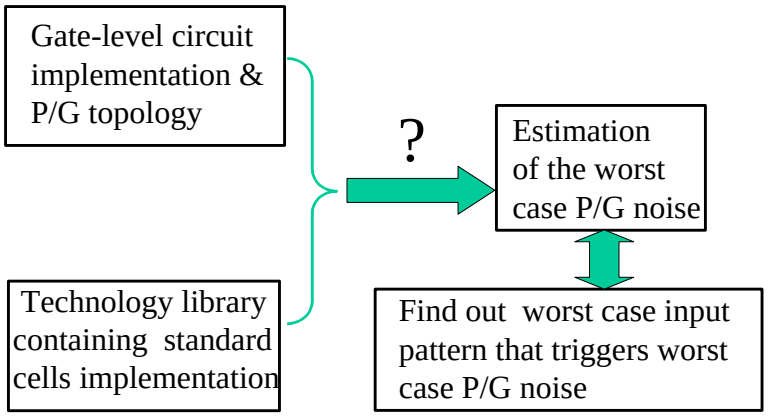
Issues in Static Analysis

- Strengths:
  - Very short run times
  - Conservative formulation
- Weaknesses:
  - Topological correlation between switching is lost
  - Switching current is applied over the entire window

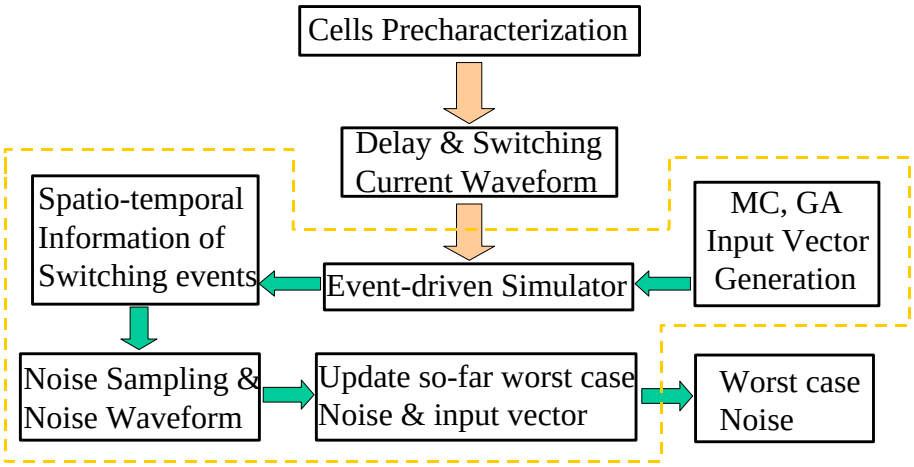
Statistical Approach

- Based on a user specified confidence level, calculate the worst case current as a function of time, using:
  - Switching intervals of the nodes in the circuit
  - Switching probabilities of each node
  - Gate current characterizations

Problem Formulation



Proposed Methodology



Input Vector Generation

- Monte Carlo is used to generate input vectors according to prescribed signal probability and activity.
- A set of so-far worst case input vectors is selected to form an initial gene pool
- Genetic algorithm is employed to generate the new generations of input vectors
- Worst case noise & corresponding input vectors are the goals

Pre-characterization of Standard Cells

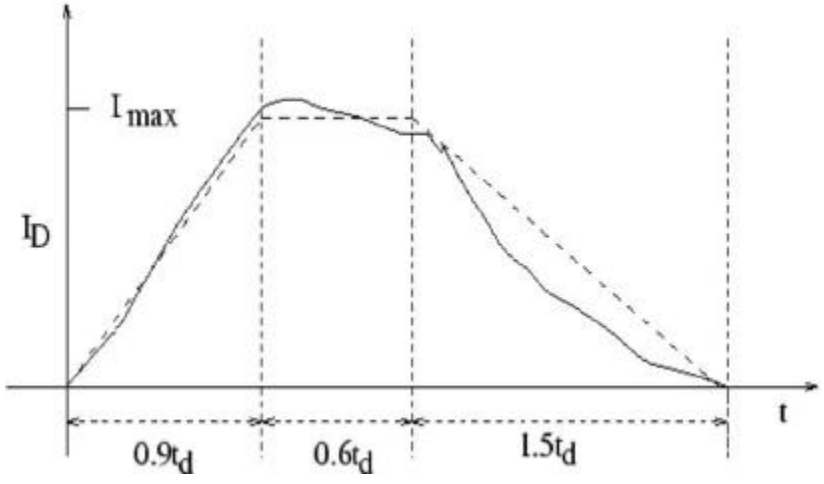
- Technology and design parameters available
- Standard cells are pre-characterized with SPICE to obtain drive capability and delay information
- A delay look-up table is used for timing analysis
- Current waveforms are approximated as trapezoids based on the delay and drive capability of switching gates

Delay Model--Lookup Table

- A delay lookup table is tabulated for each standard gate based on SPICE simulation data
- Delay depends on capacitive load and input slope
- Linear interpolation is used if necessary

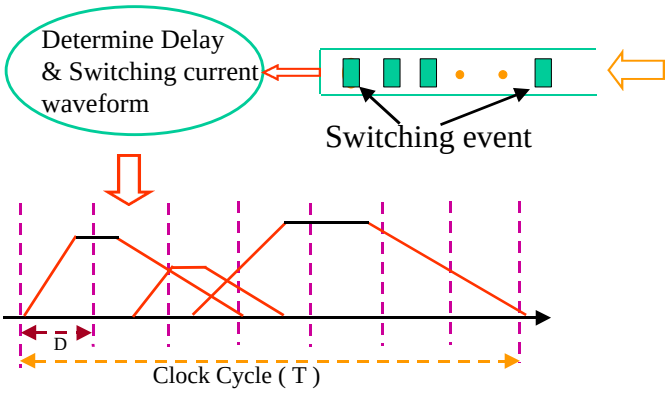
Input slope	**Capacitive load**	**Delay**	**Output slope**
τ_s	C_L	t_d	τ_o
(ps)	(fF)	(ps)	(ps)
40	20	45	58
60	80	198	250
...	...	...	...

Approximate Switching Current Waveforms with Trapezoid



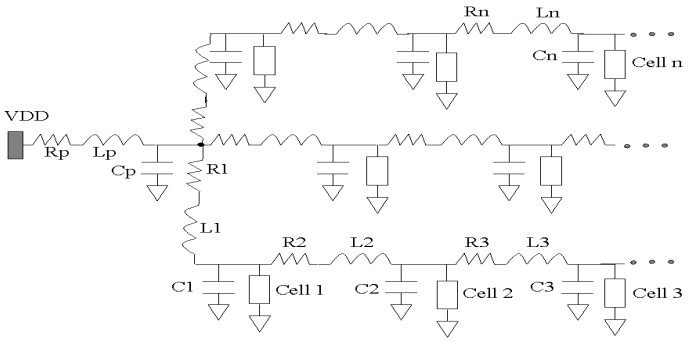
Switching Current Waveforms & Timing Information

- Switching Event Queue (Event-driven Simulator)

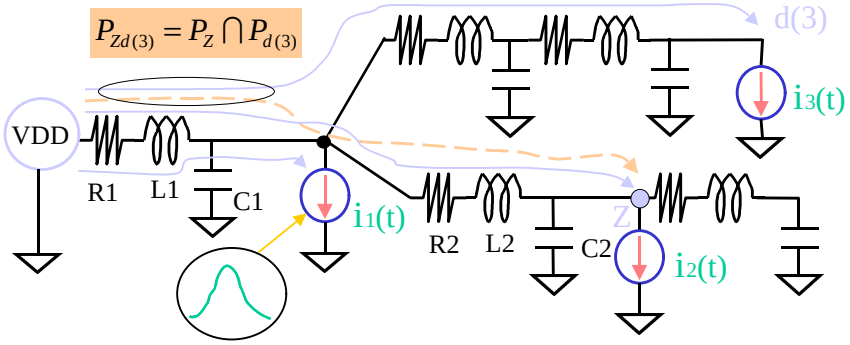


Modeling P/G Network

- P/G network is modeled as pseudo-distributed RLC network (of tree topology)



Noise Calculation



$$V_{dd} - V_Z = [i_1 R_1 + L_1 \frac{di_1}{dt}] + [i_3 R_1 + L_1 \frac{di_3}{dt}] + [(R_1 + R_2) i_2 + (L_1 + L_2) \frac{di_2}{dt}]$$

Noise Feedback & Data Post-processing

- Noise bounce on P/G reduces the effective power supply, therefore, lowers the drive current and prevents the noise bounce from going worse
- Estimated data need to post-processed
- Assume triode region operation, noise feedback is given as follows:

$$\left| V_{noise}^{est} \right| = b \, V_{dd}$$

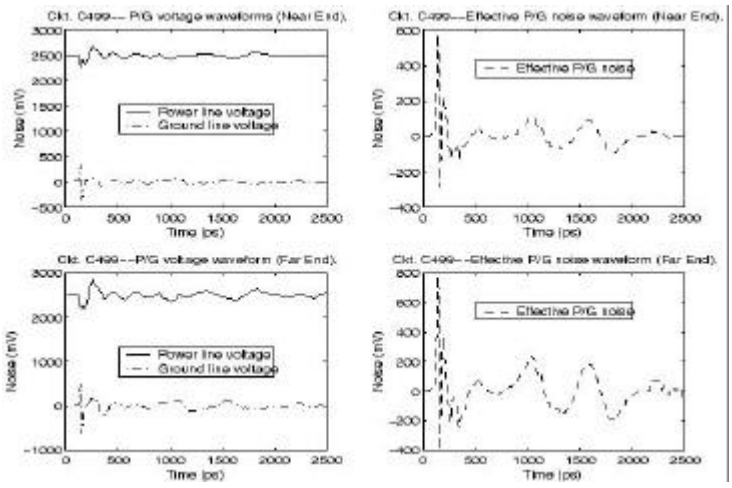
$$V_{noise}^{act} = d \, V_{dd}$$

$$(1 - d)^2 \, b = d$$

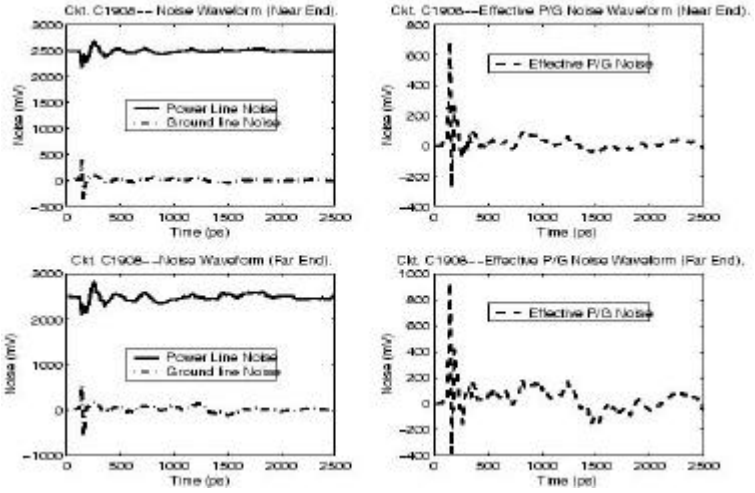
Experimental Results

Circuit	PI's No.	Gate No.	Peak Noise (Near End)	Peak Noise (Far End)	CPU Time (per input pattern)
			(mV)	(mV)	(s)
C17	5	6	35.4	39.4	0.0007
C432	36	160	372.8	394.7	0.0314
C499	41	202	573.5	780.0	0.0412
C880	60	357	612.2	698.3	0.0473
C1355	41	514	575.3	785.7	0.0779
C1908	33	880	568.3	739.6	0.1056
C2670	233	1161	701.9	814.7	0.0954
C3540	50	1667	716.0	774.7	0.3476
C5315	178	2290	1050.3	1102.0	0.4038
C6288	32	2416	676.4	1059.7	3.9042
C7552	207	3466	1079.6	1122.8	0.6397

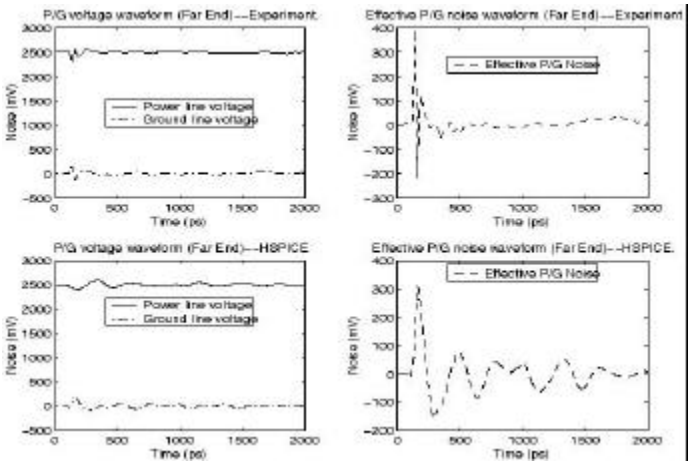
Experimental Results



Experimental Results



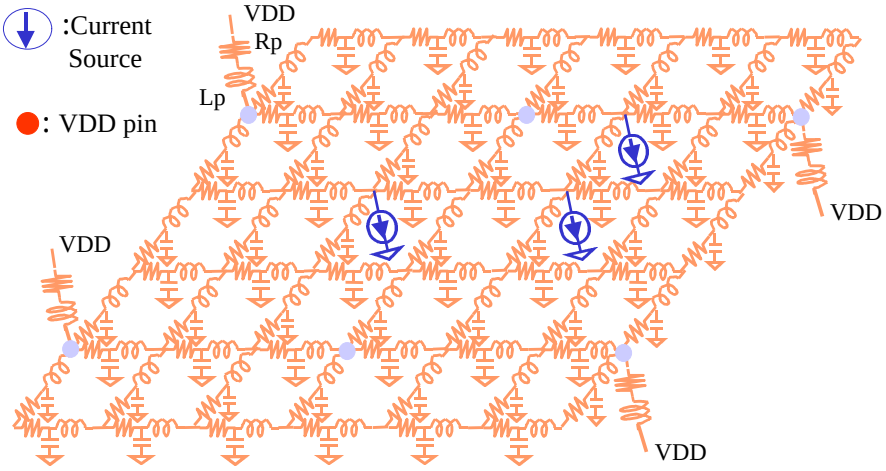
Experimental Results (compared with SPICE)



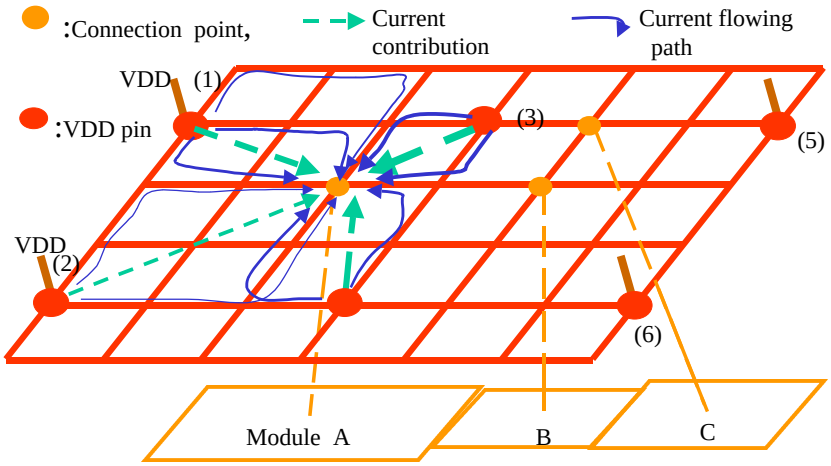
Voltage Drop Correction

- Given a floorplan with switching activities information available for each module:
  - Determine how much decap is required by each module to keep the supply noise below a specified upper limit
  - Allocate white-space to each module to meet its decap budget
- Related issue
  - Determine worst case power supply noise for each module in the floorplan
  - Allocate the existing white space in the floorplan

Power Supply Network—RLC Mesh



Current Distribution in Power Supply Mesh



Current Distribution in Power Supply Network

- Distribute switching current for each module in the power supply mesh
- Observation: Currents tend to flow along the least-impedance paths
- Approximation: Consider only those paths with minimal impedance --shortest, second shortest, …

$$I_1 + I_2 + \cdots + I_n = I$$

$$Z_1 I_1 = Z_2 I_2 = \cdots = Z_n I_n$$

$$I_j = \frac{Y_j}{\sum_{i=1}^{n} Y_i} I, \qquad j = 1, 2, \ldots n$$

Decoupling Capacitance Budget

- Decap budget for each module can be determined based on its noise level
- Initial budget can be estimated as follows:

$$\text{Charge}: \quad Q^{(k)} = \int_0^t I^{(k)}(t)dt$$

$$\text{Noise ratio}: \quad q = \max(1, \frac{V_{noise}^{(k)}}{V_{noise}^{(\lim)}})$$

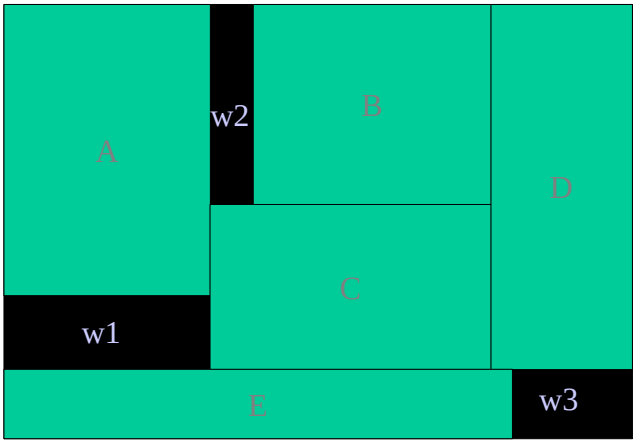
$$\text{Decap}: \quad C^{(k)} = (1 - \frac{1}{q}) Q^{(k)} / V_{noise}^{(\lim)}, \quad k = 1,2,\cdots M$$

- Iterations are performed if necessary until noise at each module in the floorplan is kept under certain limit

Allocation of Decoupling Capacitance

- Decap needs to be placed in the vicinity of each target module
- Decap requires WS to manufacture on
  - Use MOS capacitors
- Decap allocation is reduced to WS allocation
- Two-phase approach:
  - Allocate the existing WS in the floorplan
  - Insert additional WS into the floorplan if required

Allocation of Existing White Space



Allocation of Existing WS-- Linear Programming (LP) Approach

- Objective: Maximize the utilization of available WS
- Existing WS can be allocated to neighboring modules using LP
- Notation:

  S: *sum of allocated WS*

  S_k: *area of* WS_k

  $S^{(j)}$: *decap budget of* mod_j

  $x_k^{(j)}$: *ws allocated to* mod_j *from* WS_k

  N_k: *neighbors set of* WS_k

- LP Approach:

$$\text{maximize} \quad S = \sum_{k=1}^{H} \sum_{j \in N_k} x_k^{(j)},$$

$$s.t. \quad \sum_{j \in N_k} x_k^{(j)} \le S_k, \quad k=1,2,\cdots,H$$

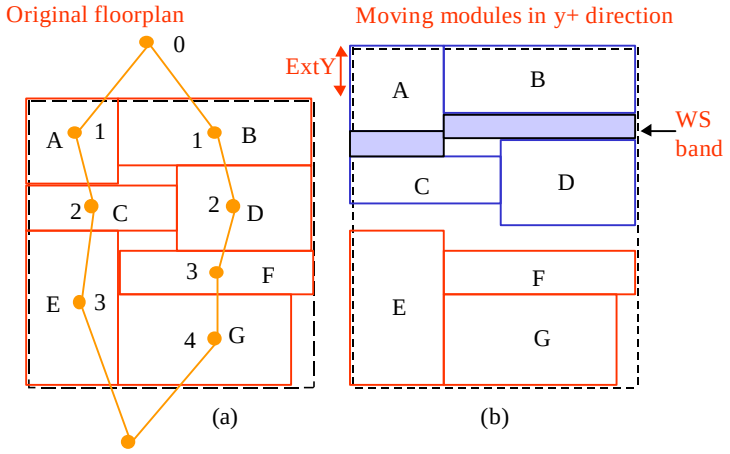
$$\sum_{k=1}^{k=H} x_k^{(j)} \le S^{(j)}, \quad j=1,2,\cdots,M$$

$$x_k^{(j)} \ge 0, \quad \forall j, \forall k$$

Insert Additional WS into Floorplan If Necessary

- Update decap budget for each module after existing WS has been allocated
- If additional WS if required, insert WS into floorplan by extending it horizontally and vertically
- Two-phase procedure:
  - insert WS band between rows based the decap budgets of the modules in the row
  - insert WS band between columns based on the decap budgets of the modules in the column

Moving Modules to Insert WS



Experimental Results
Comparison of Decap Budgets
(Ours vs "Conventional Solution")

Circuit	decap budget (nF) (our method)	decap budget (nF) ("greedy solution")	Percentage (%)
apte	27.73	32.64	85.04
xerox	8.00	13.50	59.30
hp	3.45	6.18	55.80
ami33	0	0.80	0.00
ami49	10.28	24.80	41.50
playout	42.91	61.67	69.6

Experimental Results for MCNC Benchmark Circuits

Circuit	Modules	Existing WS (μm^2) (%)	decap Budget (nF)	Inacc. WS (μm^2) (%)	Added WS (μm^2) (%)	Est. Peak Noise (V) before	Est. Peak Noise (V) after
apte	9	751652 (1.6)	27.73	0 (0)	4794329 (10.3)	1.95	0.24
xerox	10	1071740 (5.5)	8.00	0 (0)	528892 (2.7)	0.94	0.20
hp	11	695016 (7.8)	3.45	306076 (3.5)	300824 (3.4)	1.09	0.23
ami33	33	244728 (21.3)	0	N/A	0	0.16	0.16
ami49	49	2484496 (7.0)	10.28	891672 (2.5)	463615 (1.3)	1.45	0.25
playout	62	5837072 (6.6)	42.91	792110 (0.9)	3537392 (4.0)	1.23	0.24

Floorplan of playout Before/After WS Insertion