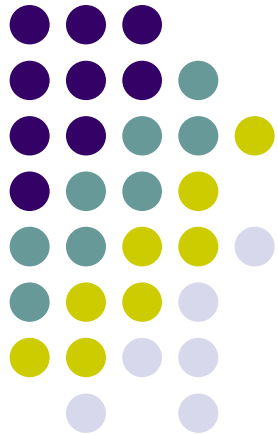
8B-4

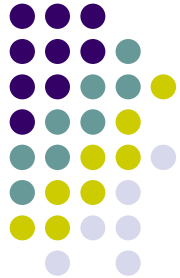
Voltage Island Generation under Performance Requirement for SoC Designs

Wai-Kei Mak and Jr-Wei Chen



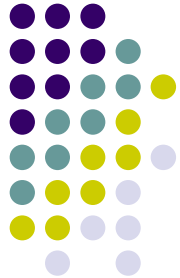
National Tsing Hua University

Outline



- Motivation
- Previous works
- Voltage Assignment with given floorplan
  - ILP Formulation
- Multi-Supply Voltage Floorplanning
- Experimental results
- Conclusion

Introduction



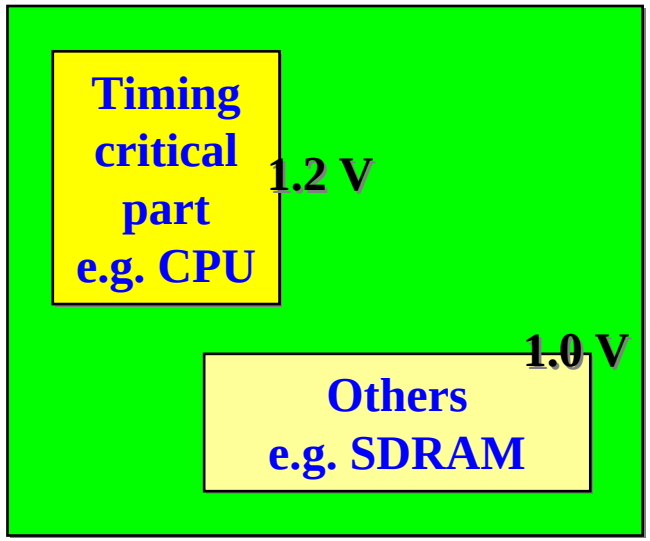
- How to reduce power of a System-on-a-Chip (SoC) design consisting of a mixture of cores?

- Dynamic power consumption depends of the square of the operating voltage.

$$P_d = f \cdot k \cdot C \cdot V^2$$

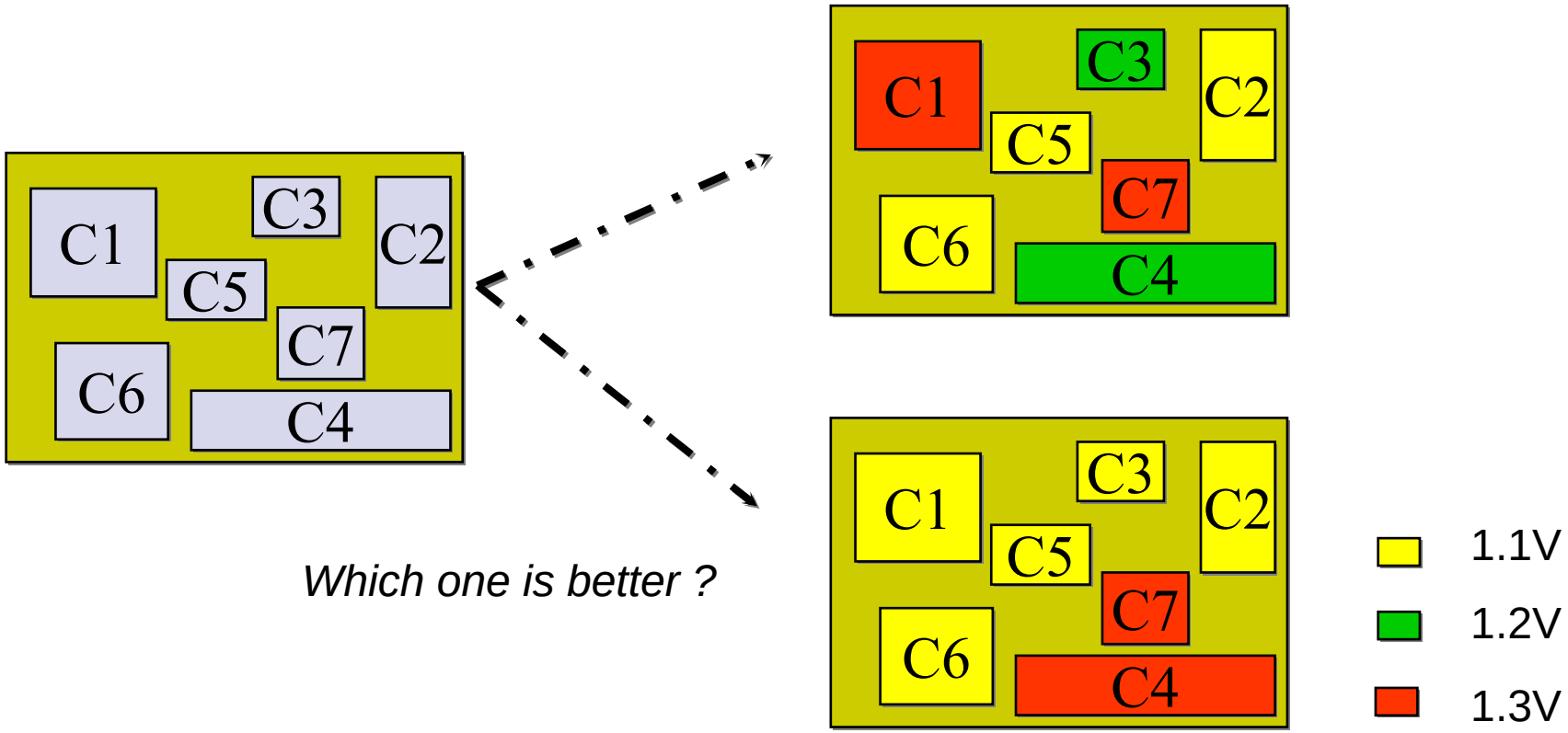
Mixed Voltage SoC Design

- Assign higher voltages to performance-critical components.
- Assign lower voltages to less timing-critical components to save power without sacrificing system performance.



Voltage Island Generation

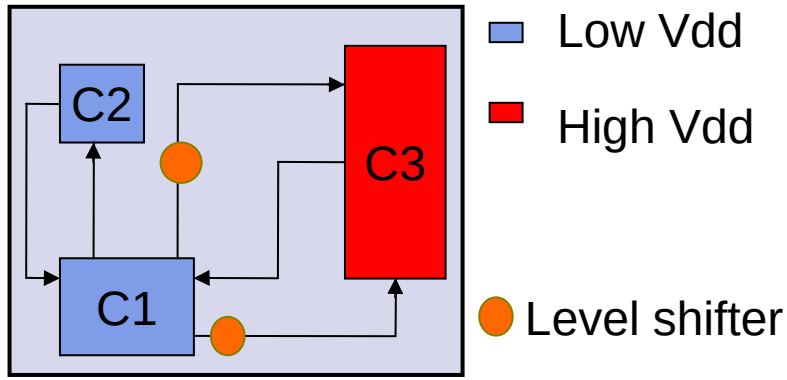
- A voltage island is a contagious region driven by the same voltage level.



Which one is better ?

Multi-Supply Voltage Floorplanning

- Construct a floorplan with voltage islands.
- To optimize
  - Power
  - Chip area
  - Wirelength
- Overhead Consideration
  - Power grid complexity (cost, noise)
  - Level shifter insertion (area, delay, power)



Previous Works

- "Architecting Voltage Island In Core-based System-on-a-Chip Design", ISLPED'04
  - Unconstrained optimization using simulated annealing
  - Minimize (*core power* + *#voltage islands* + *level shifter area*)
  - Obtained power saving of 14% to 28%
  - Chip area and wirelength were not reported
- "Temperature-Aware Voltage Islands Architecting in System-on-Chip Design", ICCD'05
  - Unconstrained optimization using simulated annealing and genetic algorithm
  - Minimize (*core power* + *chip area* + *wirelength*)
  - Chip area was significantly worsened

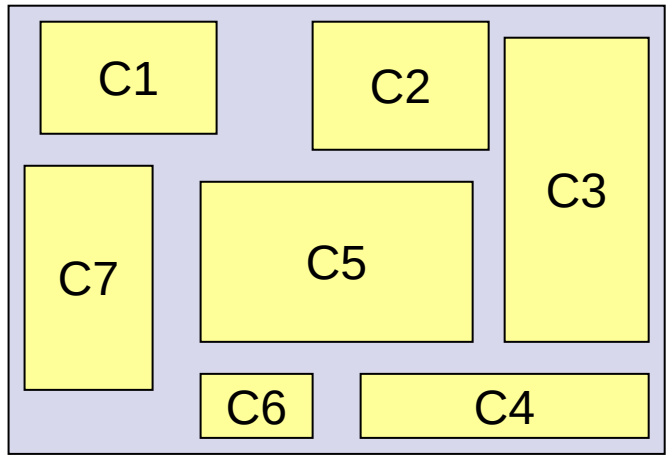
Our Contributions

- We combined a traditional simulated annealing based floorplanner with a voltage level assignment (VLA) algorithm.
- Unlike previous works, we optimize total power consumption and power network complexity w/o sacrificing wirelength and chip area.
- Integer programming-based approach which can handle many different constraints.
- 2 formulations: one is exact, the other is much faster and very close to optimal.
- Obtained 17-53% power savings compared to using a single voltage.

How to form Voltage Islands?

Given

1. Each core's acceptable voltages and corresponding power consumptions.
2. Floorplan of the cores.
3. The netlist.



	C1		C2	…
Voltage (V)	1.1	1.3	…	…
Power (mW)	10	15	…	…

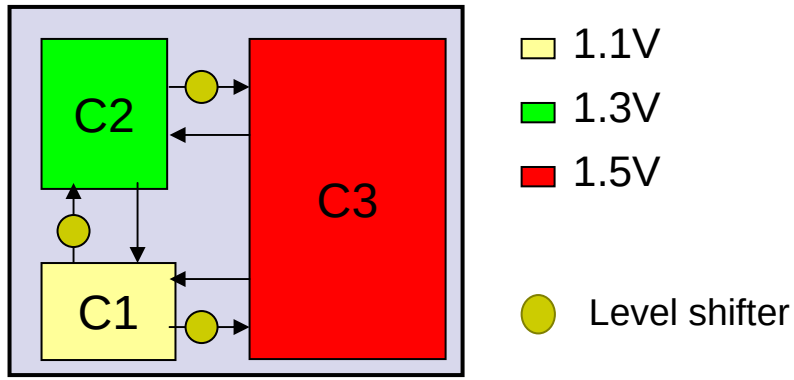
- Assign each core a "proper" voltage to get a good solution which is determined by the overall power consumption, level shifter area, and power grid complexity cost.

Voltage Assignment

- Minimize $\text{cost} = P_{total} + \beta \cdot F$

  P_{total} denotes total power consumption
  F denotes the fragmentation cost.

- P_{total} = *total power consumption*
  $= P_{cores} + P_{shifters}$



- *F = fragmentation cost*
  = # adjacent block pairs given different voltages

Binary Decision Variables

- Define voltage assignment variables b_{ip}

$$\begin{cases} b_{ip} = 1 & \text{if core } i \text{ is assigned voltage level } p \\ b_{ip} = 0 & \text{otherwise} \end{cases}$$

- Must assign a voltage to each core, so

$$\sum_{p=1}^{L_i} b_{ip} = 1$$

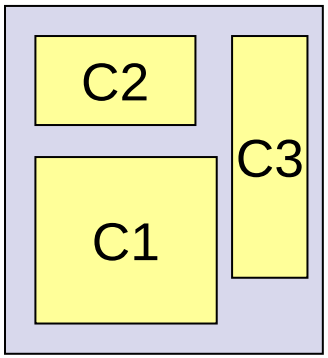
- e.g. C2 $\begin{cases} 1.1\text{V} \\ 1.3\text{V} \\ 1.5\text{V} \end{cases}$ ⇨ $b_{21} + b_{22} + b_{23} = 1$

Power Consumption (1)

- Power consumption of all cores

$$P_{cores} = \sum_{i=1}^{m} \sum_{p=1}^{L_i} (P_{ip} \cdot b_{ip})$$

- e.g.



C1		C2			C3		
1.1	1.3	1.1	1.3	1.5	1.3	1.5	V
10	15	23	33	43	28	38	mW

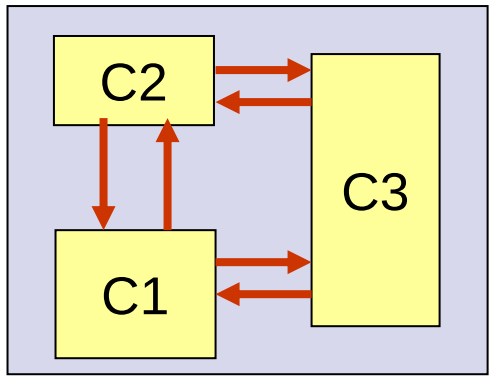
➡ $P_{cores} = \underbrace{(10b_{11}+15b_{12})}_{C1} + \underbrace{(23b_{21}+33b_{22}+43b_{23})}_{C2} + \underbrace{(28b_{31}+38b_{32})}_{C3}$

Power Consumption (2)

- Level shifter power consumption

$$P_{shifters} = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{p=1}^{L_i} \sum_{q=1}^{L_j} (P_{ip,jq} \cdot b_{ip} \cdot b_{jq})$$

  - Constant $P_{ip,jq}$ depends on # wires and switching activity on each wire between cores *i* and *j*
  - Level shifter not needed from high to low voltage, so $P_{ip,jq} = 0$ if $V_{ip} > V_{jq}$

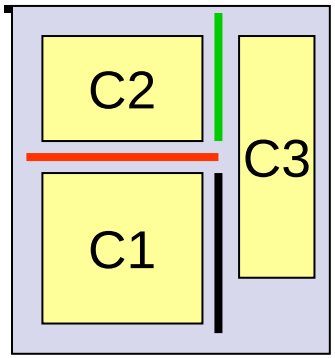


Fragmentation Cost

- Incur fragmentation cost if adjacent cores operate at different voltages

$$F = \sum_{i} \sum_{\substack{i<j \wedge \\ j \in N_i}} \sum_{V_{ip} \neq V_{jq}} (b_{ip} \cdot b_{jq})$$

- e.g.



C1		C2			C3		
1.1	1.3	1.1	1.3	1.5	1.3	1.5	V

$$F = (b_{11}b_{22}+b_{11}b_{23}+b_{12}b_{21}+b_{12}b_{23}) + (b_{11}b_{31}+b_{11}b_{32}+b_{12}b_{32}) + (b_{21}b_{31}+b_{21}b_{32}+b_{22}b_{32}+b_{23}b_{31})$$

Linearize(1)

- Previous equations for $P_{shifters}$ and F are non-linear.

$$P_{shifters} = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{p=1}^{L_i}\sum_{q=1}^{L_j}(P_{ip,jq} \cdot b_{ip} \cdot b_{jq}) \qquad F = \sum_{i}\sum_{\substack{i<j \wedge \\ j\in N_i}}\sum_{V_{ip}\neq V_{jq}}(b_{ip} \cdot b_{jq})$$

- How to get around that?

  By introducing new binary variables $b'_{ip,jq}$ s.t.

$$b'_{ip,jq} \geq b_{ip} + b_{jq} - 1$$

Then, we've

$$\begin{cases} b'_{ip,jq} = 1 & \text{if } b_{ip}=1 \ \& \ b_{jq}=1 \\ b'_{ip,jq} = 0 & \text{otherwise} \end{cases}$$

Linearize(2)

- New equations for $P_{shifters}$ and F

$$P_{shifters} = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{p=1}^{L_i} \sum_{q=1}^{L_j} (P_{ip,jq} \cdot b'_{ip,jq})$$

$$F = \sum_{i=1} \sum_{\substack{i<j \wedge \\ j \in N_i}} \sum_{V_{ip} \neq V_{jq}} b'_{ip,jq}$$

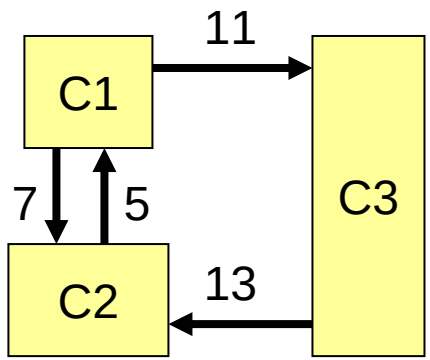
Other Useful Constraints (1)

- Upper bound the total level shifter area allowed

$$\sum_{i}\sum_{j}\sum_{\substack{i\neq j\wedge \\ V_{ip}<V_{jq}}} (S_{ij}\cdot b'_{ip,jq}) \leq A$$

(S_{ij} denotes #signals from core *i* to core *j*)

- e.g.



C1		C2		C3		
1.1	1.3	1.3	1.5	1.3	1.5	V

➡ $(7b'_{11,21}+7b'_{11,22}+7b'_{12,22}) + (11b'_{11,31}+11b'_{11,32}+11b'_{12,32}) + (13b'_{31,22}) \leq A$

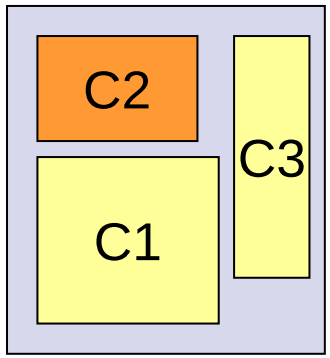
Other Useful Constraints (2)

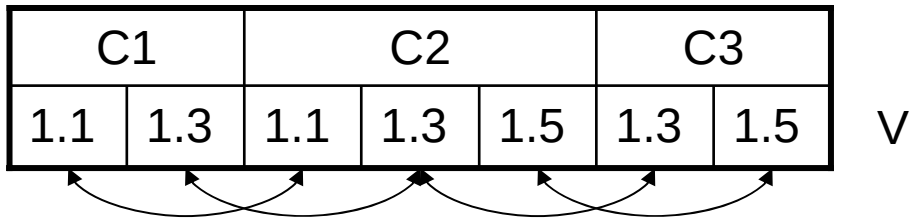
- Adjacency constraint
  - enforce every core must have same voltage as some adjacent cores

$$\sum_{j \in N_i} \sum_{V_{ip}=V_{jq}} b'_{ip,jq} \geq U_i$$

$$\begin{cases} U_i = 1 & \text{if } 2 \leq \#\text{ neighbors} \leq 3 \\ U_i = 2 & \text{if } 4 \leq \#\text{ neighbors} \leq 7 \\ U_i = 3 & \text{if } 7 \leq \#\text{ neighbors} \end{cases}$$

- e.g.





For C2, we've

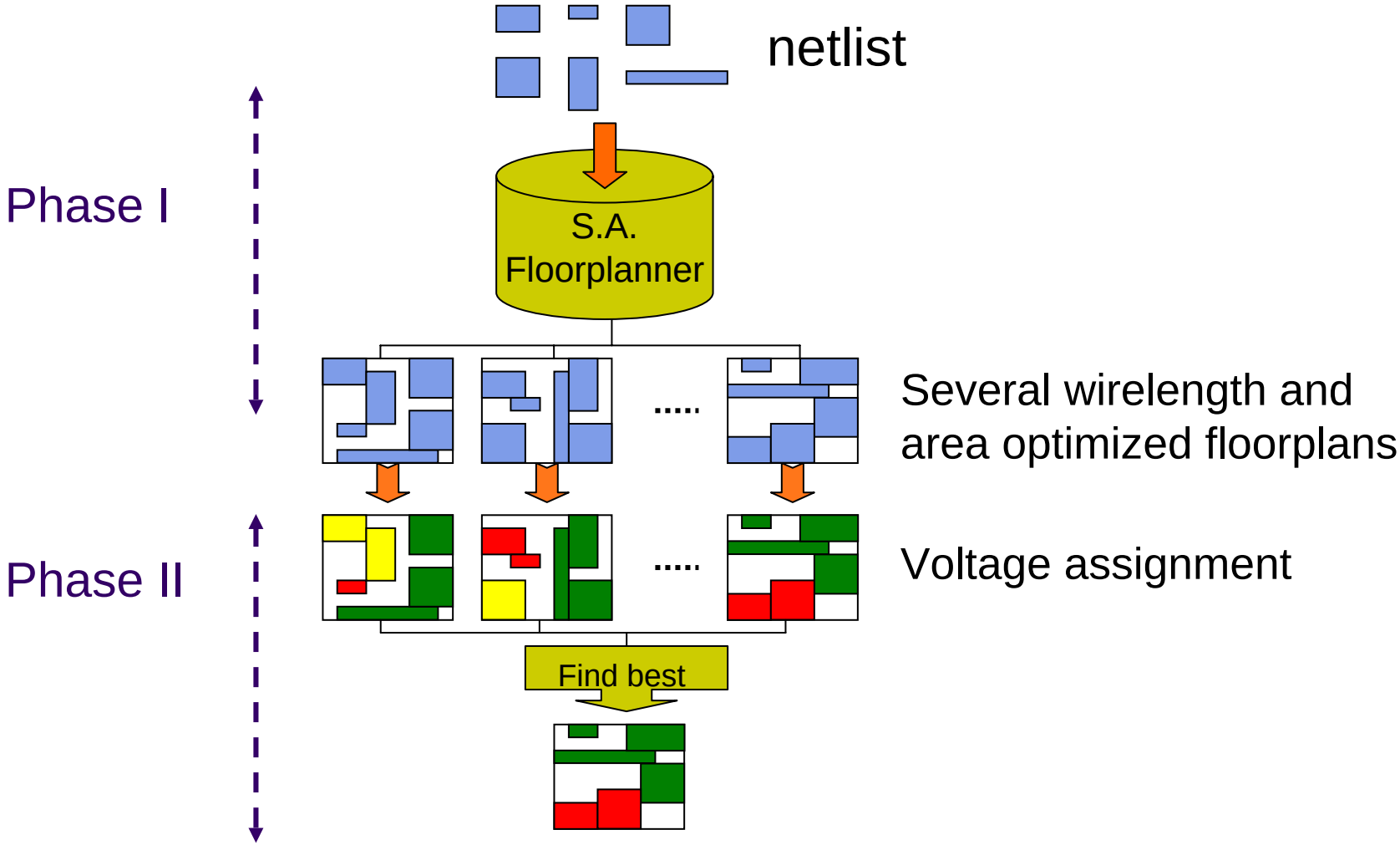
$b'_{21,11}+b'_{22,12}+b'_{22,31}+b'_{23,32} \geq 1$

Other Useful Constraints (3)

- Upper bound the total power consumption

$$P_{total} \leq P_{bound}$$

Multi-Supply Voltage Floorplanning



Experimental results

- Implemented our algorithm in C++.
- AMD Opteron 2.1GHz and 4GB memory.
- ILPs solved by lp_solve 5.5.0.2.
- We randomly generate the switching activities of the nets such that 80% of them are below 0.5 and 20% are above 0.5.

- Voltage level distribution { 1.1V, 1.3V, 1.5V, 1.8V }.
- Assume power consumption of a block is proportional to its area and the square of its supply voltage.

Experimental results

- Benchmark spec.

Benchmark	# blocks	Benchmark	# blocks
apte	9	2xerox	20
xerox	10	2hp	55
hp	11	ami75	75
ami33	33	ami99	99
ami49	49	ami200	200
		ami300	300

- For each case, we generated 5 top floorplans in terms of area and wirelength.

Comparing 5 floorplan candidates

Benchmark	Power and fragmentation cost		
	Best	Worst	Difference(%)
apte	1127.09	1374.72	18.1
xerox	2198.39	2290.13	4.00
hp	2248.18	2352.93	4.45
ami33	5262.14	5443.02	3.32
ami49	8426.63	8485.99	0.68
2xerox	3709.56	4018.2	7.68
2hp	4476.32	4634.13	3.40
ami75	13188.1	13420.8	2.72
ami99	16667	16696.8	1.73
ami200	36835.8	37490.9	1.78
ami300	49445.35	50861.38	2.78

Quality and Runtime Comparison of 2 Formulations

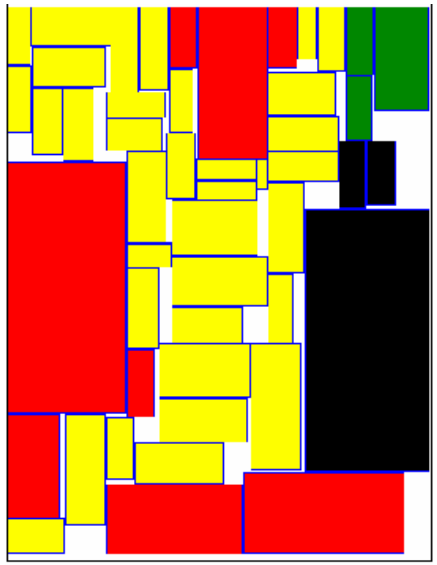
Benchmark	Exact		Fast		P_{total} diff. (%)	T diff. (X)
	P_{total}	Run time(s)	P_{total}	Run time(s)		
apte	1113.75	0.079	1113.75	0.012	0	6.58
xerox	2065.99	0.117	2065.99	0.031	0	3.77
hp	2198.18	0.02	2198.18	0.015	0	1.33
ami33	4937.14	10.585	4937.14	2.1359	0	4.96
ami49	7789.56	13.52	7854.78	2.011	0.83	6.72
2xerox	3559.56	3.476	3559.56	0.343	0	10.134
2hp	4201.32	0.566	4201.32	0.172	0	3.29
ami75	12040.9	195.003	12238.1	12.134	1.61	16.07
ami99	15596.8	147.16	15596.8	37.571	0	3.91
ami200	N/A	>3600	31112.6	49.359	N/A	>72.94
ami300	N/A	>3600	47640.0	125.258	N/A	>28.74

Power Saving

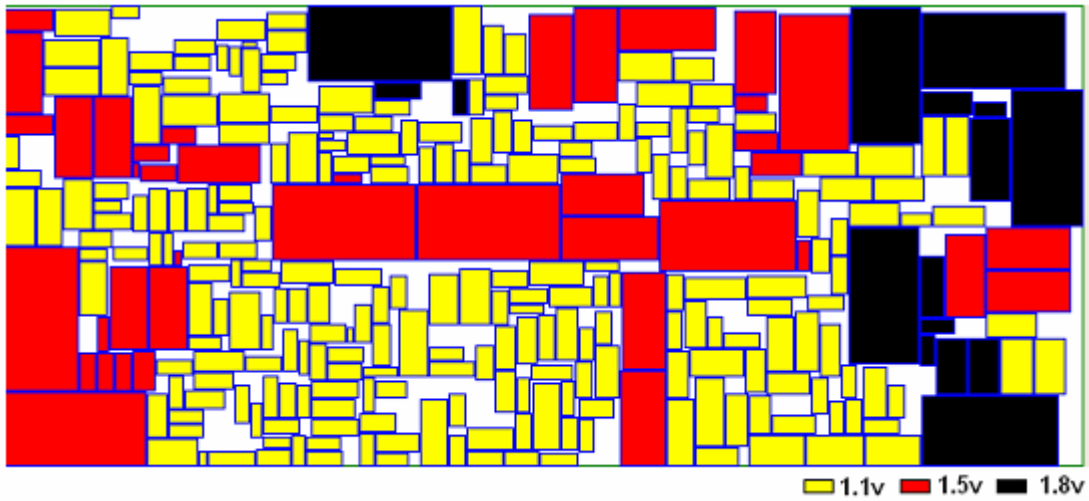
Benchmark	Power w/o VLA	Power (VLA)		Power saving (%)	Run time (s)	
		P_{cores}	$P_{shifters}$		Phase I	Phase II
apte	2409.92	1093.95	19.8	53.78	5.3435	0.1390
xerox	2677.69	1658.94	407.05	22.85	6.8626	0.2173
hp	2945.45	2119.88	78.3	25.37	122.84	0.1564
ami33	8836.36	4567.74	369.4	44.12	76.493	12.903
ami49	13120.7	7800.03	54.75	41.13	80.736	9.7319
2xerox	5355.37	3055.36	504.2	33.53	73.036	1.7174
2hp	5090.91	3909.27	292.05	17.47	25.832	1.0357
ami75	20082.6	12026.2	211.9	39.06	251.83	64.736
ami99	26509.1	15432	164.85	41.16	494.52	189.90
ami200	53553.7	31247.8	502.11	41.90	3606.9	244.73
ami300	80330.6	47211	429.3	40.69	7023.5	357.14
average	***---***	***---***	***---***	***36.46***	***---***	***---***

Sample Floorplanning Results

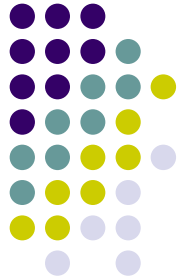
ami49



ami300



Conclusions



- In this paper, we proposed a new effective floorplanning approach targeting multiple supply voltage core-based SoC designs.

- It can effectively optimize the total power consumption of cores and level shifters, control the level conversion area overhead and the power network complexity without compromising the wirelength and chip area.

Thank You.