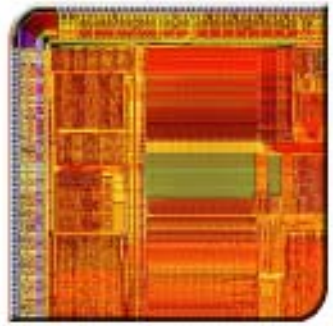
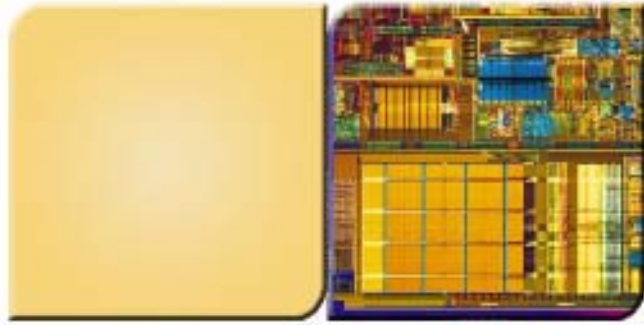
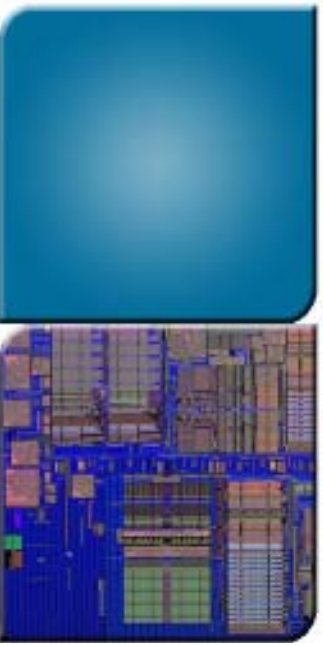




Noise-Direct: A Technique for Power Supply Noise Aware Floorplanning Using Microarchitecture Profiling

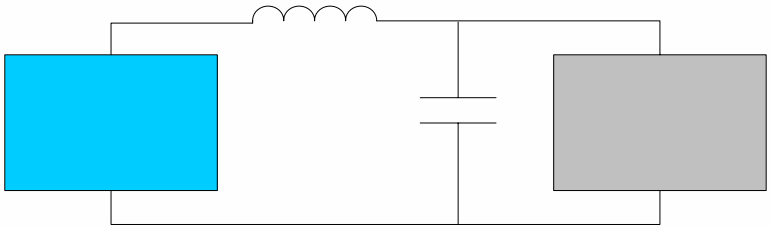


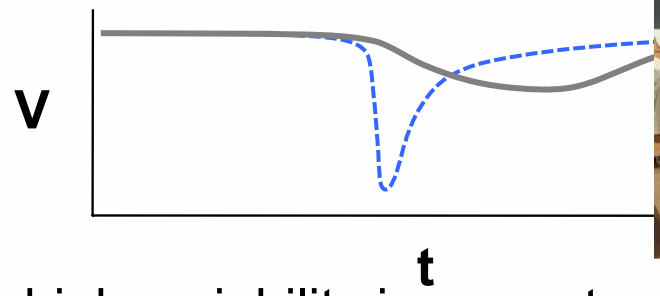
Fayez Mohamood*
Michael Healy
Sung Kyu Lim
Hsien-Hsin "Sean" Lee

School of Electrical and Computer Engineering
Georgia Institute of Technology
AMD, Inc*

GSRC C2S2 MARCO Georgia Tech MARS

Inductive Noise





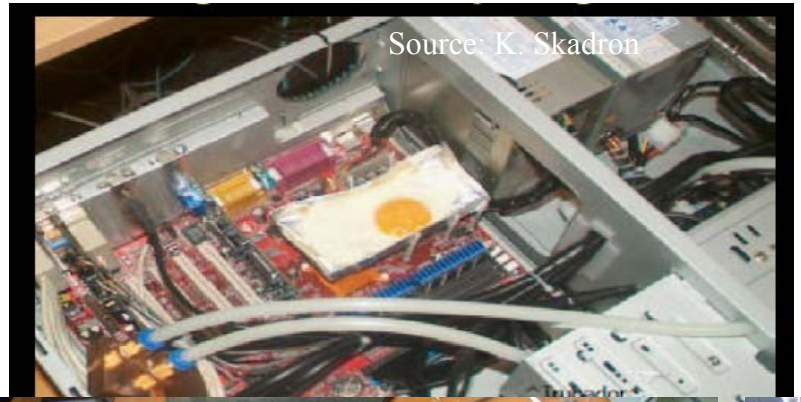
- Power supply noise caused due to high variability in current per unit time
  - $\Delta V = L(di/dt)$
- Reliability Issue that needs to be guaranteed
  - Typically done through a multi-stage decap placement (motherboard/package/on-die)
- Can be addressed by an over-designed power network, however
  - Leads to high use of multi-stage decap
  - More metal for power grid, leaving less for signals
- Chip is designed to account for a program that can induce the worst-case power supply noise

Voltage Regulator

Georgia Tech MARS

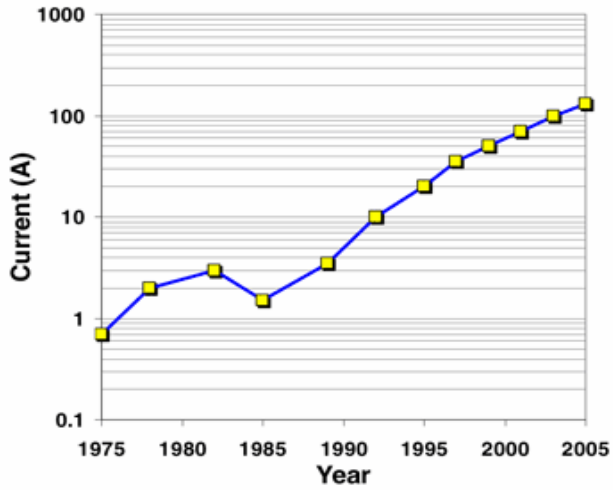
Why Now

- More active devices on chip
  - Higher power consumption



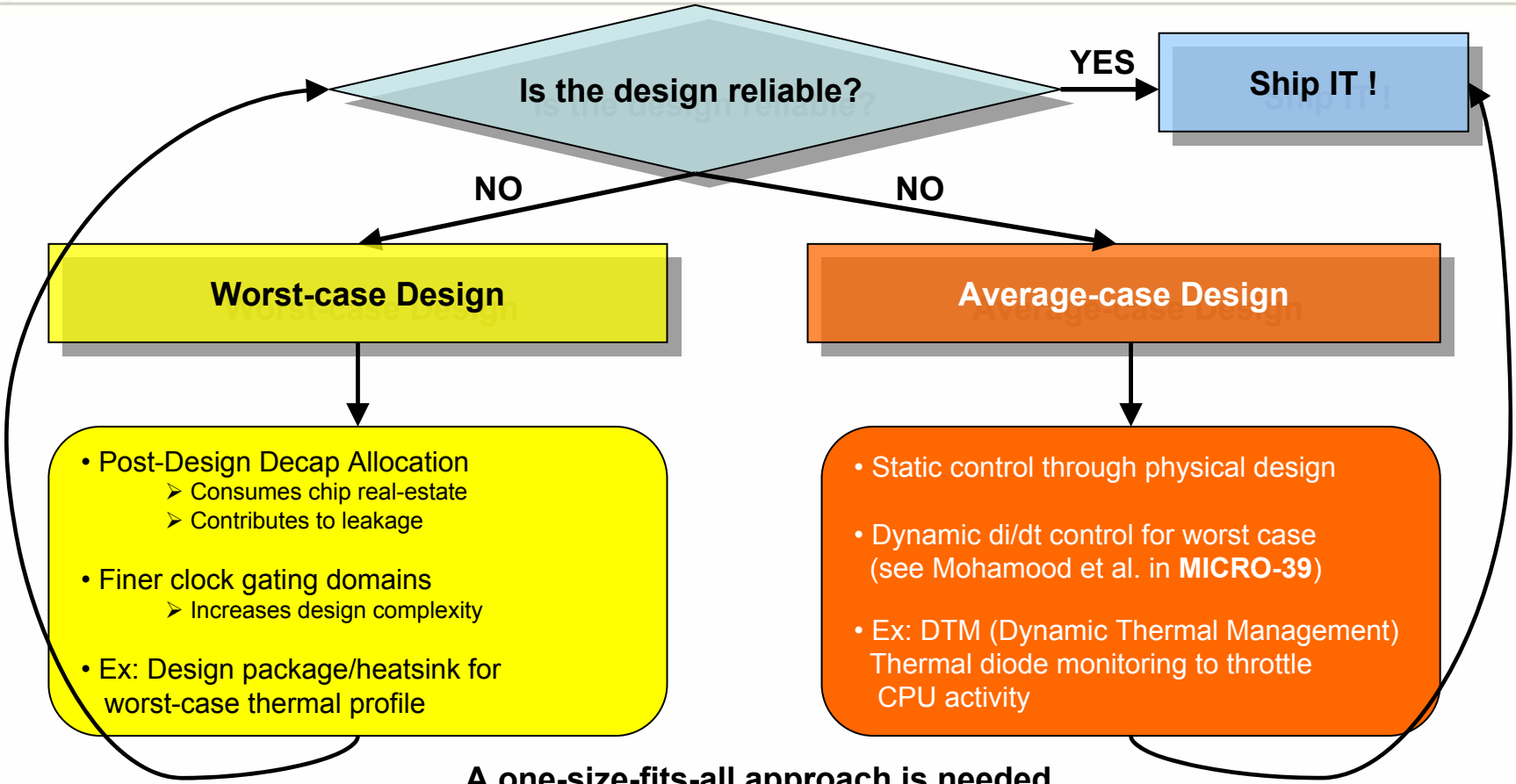
Why Now?

- More active devices on chip
  - Higher power consumption
- Exponential increase in current consumption
  - Intel reports 225% increase per unit area per generation
- Device size miniaturization leads to lower operating voltages
  - Lower noise margins
- Aggressive power saving techniques
  - Clock-gating
- Multi-core trend can exacerbate di/dt issues



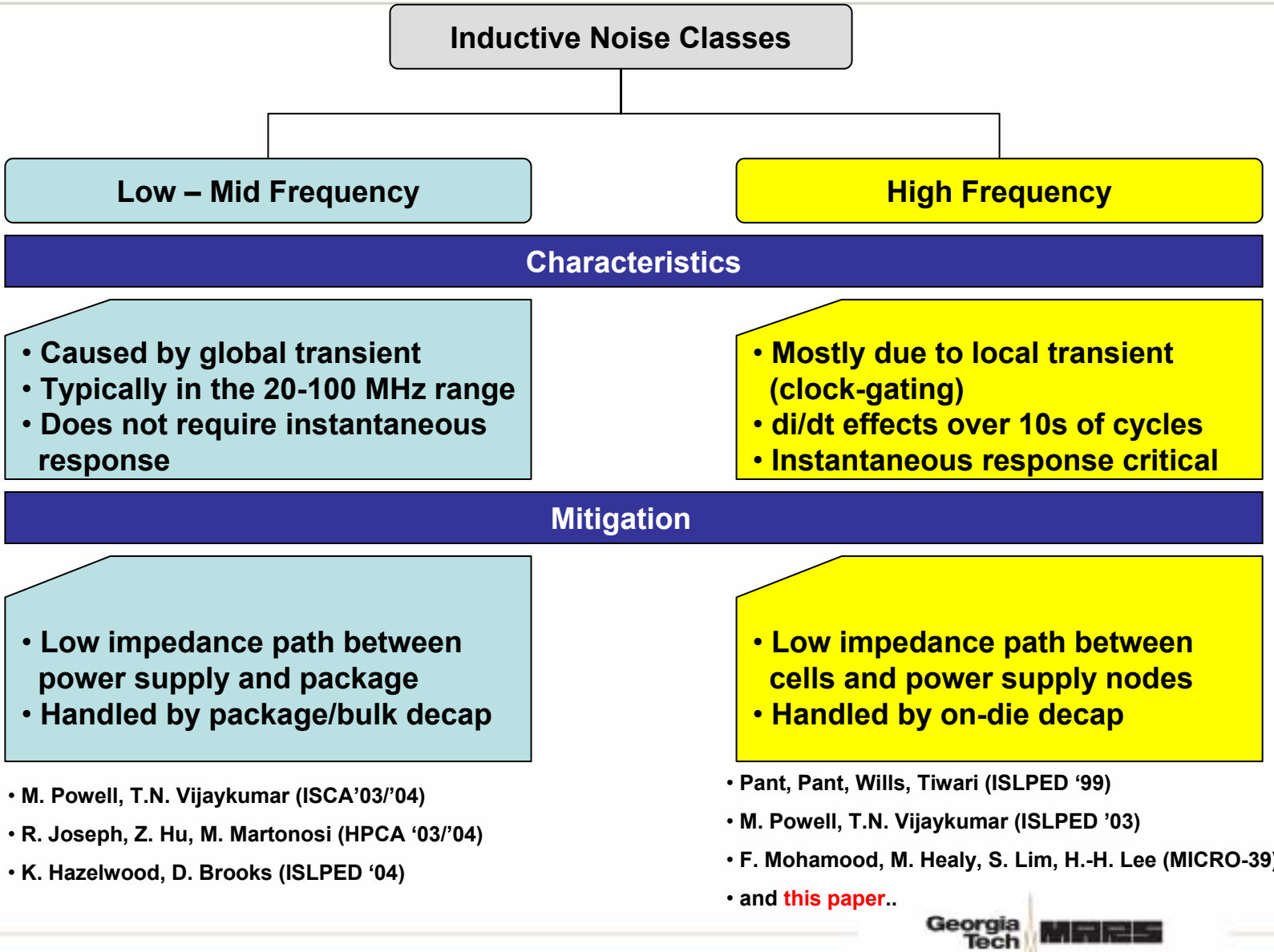
Source: Intel Technology Journal
Volume 09, Issue 04 Nov 9, 2005

Worst-case Design Inefficiency



A one-size-fits-all approach is needed

Inductive Noise Taxonomy



di/dt from Microarchitectural Perspective

- Noise characteristics reflect program behavior
  - Static characteristics
    - Functional Unit Usage
    - Location of modules relative to power pin
  - Dynamic characteristics like cache misses
  - E.g. power virus

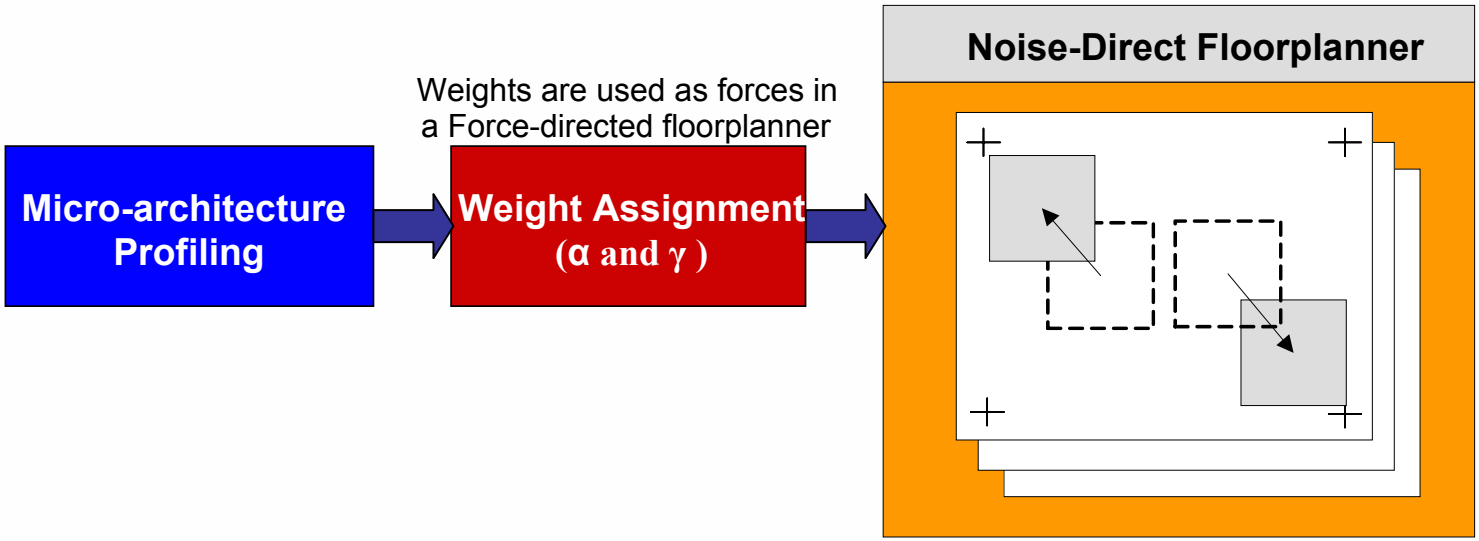
- Can floorplanning can exploit the above characteristics?
  - Use microarchitectural information to identify “problematic” modules
  - Optimize the floorplan based on benchmark profile information



Exploiting Floorplanning for di/dt

- High frequency di/dt is a function of the chip floorplan
- Factors affecting noise at a module:
  - Frequency and intensity of switching activity
  - Distance between each μarch module and power-pins
  - Proximity to a simultaneously switching module
- Formulating the problem:
  - Quantify fine-grained microarchitectural activity
  - Employ a floorplanning algorithm that optimizes for di/dt
- Result is a floorplan that is inherently noise tolerant (for the average case)

Noise-Direct Design Methodology



- Profile microarchitectural module activity to quantify average-case behavior
- Quantifying metrics:
  - Self-Switching Weight (α)
  - Correlated-Switching Weight (γ)
- Optimized floorplan:
  - Direct modules with high α closer to power-pins
  - Direct module pairs with high γ away from each other

Self-Switching Weight

- Self-Switching Weight (**α**)
  - Relative likelihood of **a module** switching at a given time
  - Certain modules gated far more than others
  - For instance, the I$ is likely to be accessed all the time (except during fetch bottlenecks) → Low α

# of switching

$$\alpha_i = sw_i \bullet I_i$$

Intensity
(Current consumption)

Correlated Switching Weight

- Correlated-Switching Weight (γ)
  - Relative likelihood of **a module pair** switching **simultaneously** at a given time
  - Microarchitecture dependent metric
  - For instance, a VIPT cache would result in an I$ and I-TLB that are accessed in parallel → High γ

Xi,j : correlated switching for i

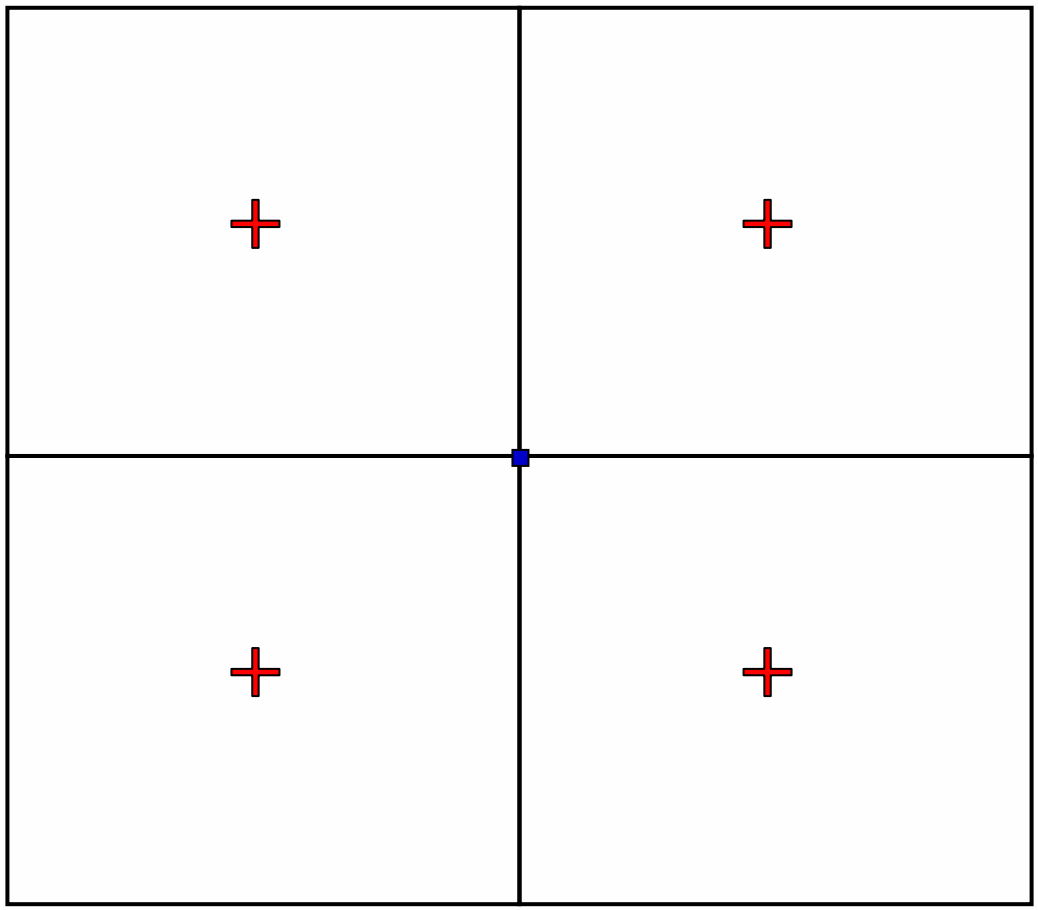
$$\gamma_{i,j} = \frac{1}{2}\left(\frac{X_{i,j}}{sw_i} + \frac{X_{j,i}}{sw_j}\right) \bullet \frac{1}{2}(I_i + I_j)$$

Average correlated Intensity

Self- and Correlated-Switching Activity

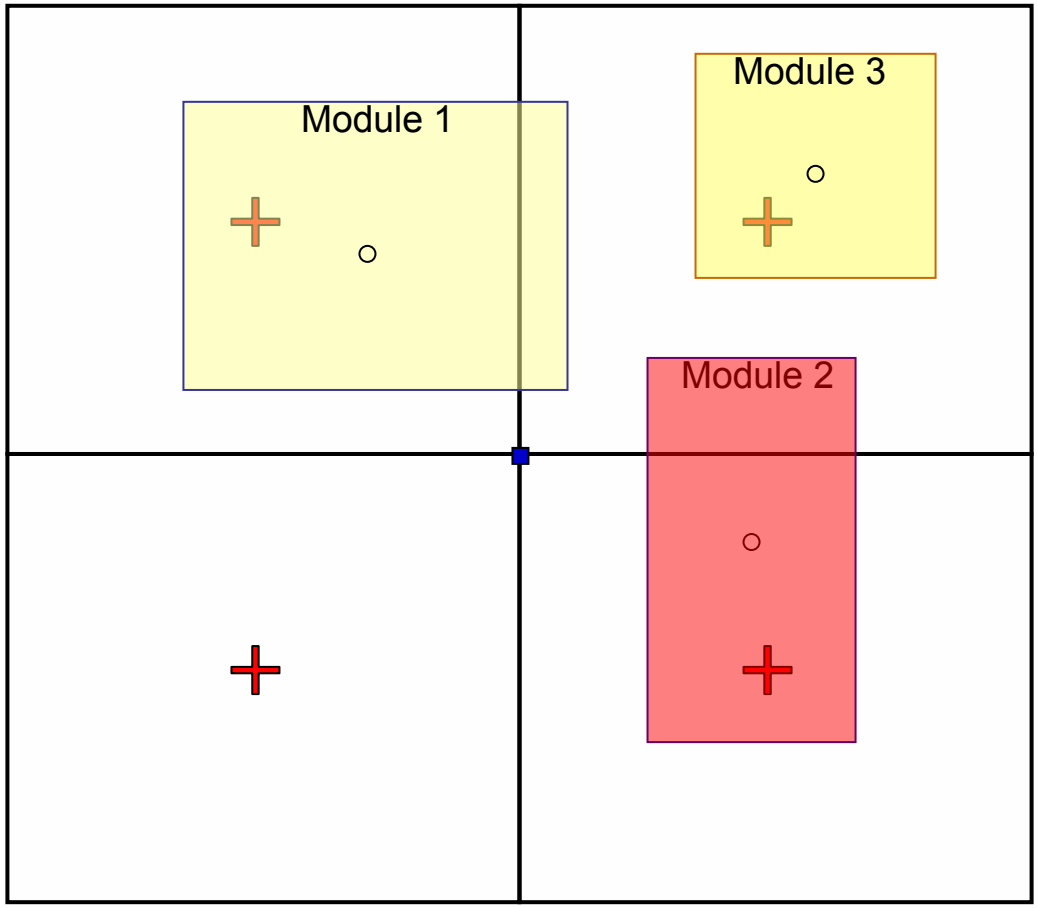
	LSQ	RUU	BTB	L2$	IRF	L1D$	ALU0	ALU1	ALU2	ALU3	ALU4	ALU5	L1I$	Bpred	DTLB	ITLB	FALU0	FALU1	Freg
LSQ	*28*	0	20	13	20	2	10	10	10	10	10	10	11	20	0	11	10	10	12
RUU		*26*	8	4	13	2	0	0	0	0	0	0	5	8	2	5	0	0	5
BTB			*18*	7	29	17	13	13	13	13	13	13	37	100	17	37	13	13	13
L2$				*16*	14	28	12	12	12	12	12	12	21	7	26	21	4	4	7
IRF					*10*	17	7	7	7	7	7	7	23	29	17	23	8	8	24
L1D$						*7*	6	6	6	6	6	6	11	17	93	11	5	5	6
ALU0							*3*	100	100	100	100	100	15	13	6	15	66	66	4
ALU1								*3*	100	100	100	100	15	13	6	15	66	66	4
ALU2									*3*	100	100	100	15	13	6	15	66	66	4
ALU3										*3*	100	100	15	13	6	15	66	66	4
ALU4											*3*	100	15	13	6	15	66	66	4
ALU5												*3*	15	13	6	15	66	66	4
L1I$													*3*	37	12	100	11	11	5
Bpred														*3*	17	37	13	13	13
DTLB															*2*	12	5	5	6
ITLB																*1*	11	11	5
FALU0																	*1*	100	5
FALU1																		*1*	5
Freg																			*0*

Force-Directed Floorplanning



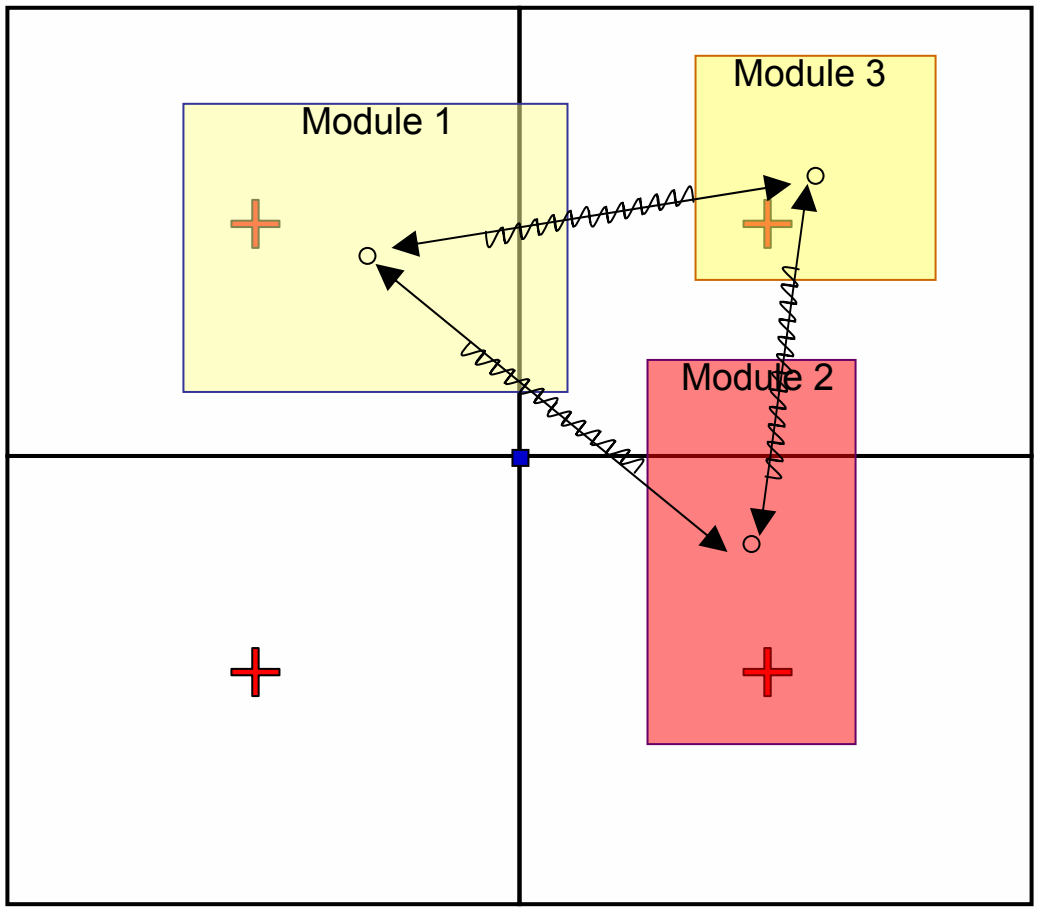
Power Pin

Force-Directed Floorplanning



Power Pin

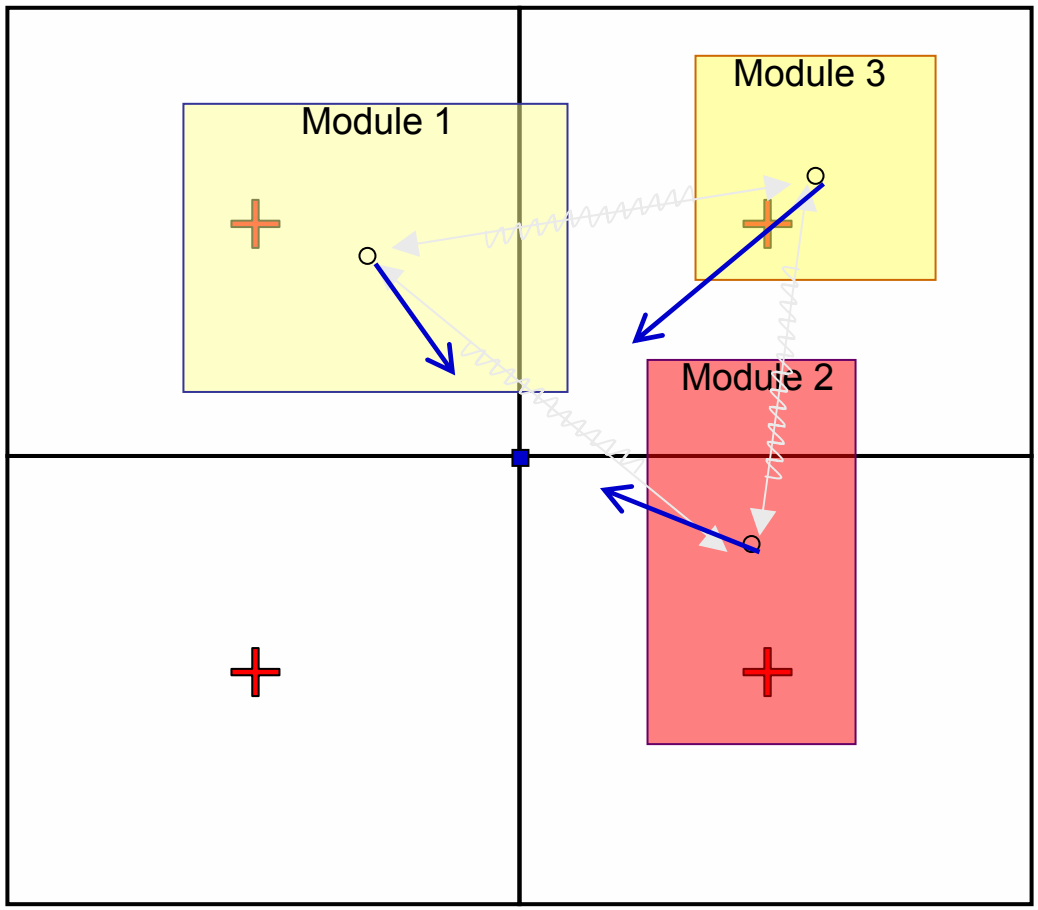
Force-Directed Floorplanning



Power Pin

Net Force

Force-Directed Floorplanning

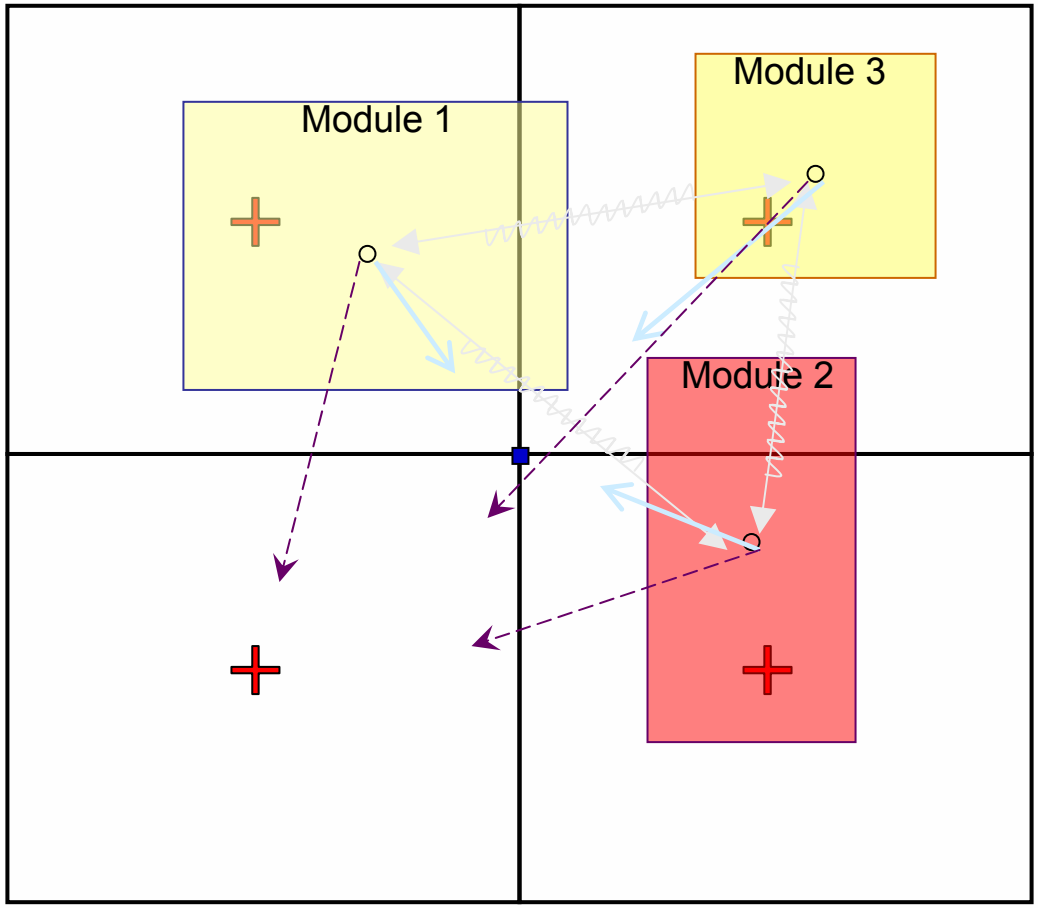


Power Pin

Net Force

Center Force

Force-Directed Floorplanning



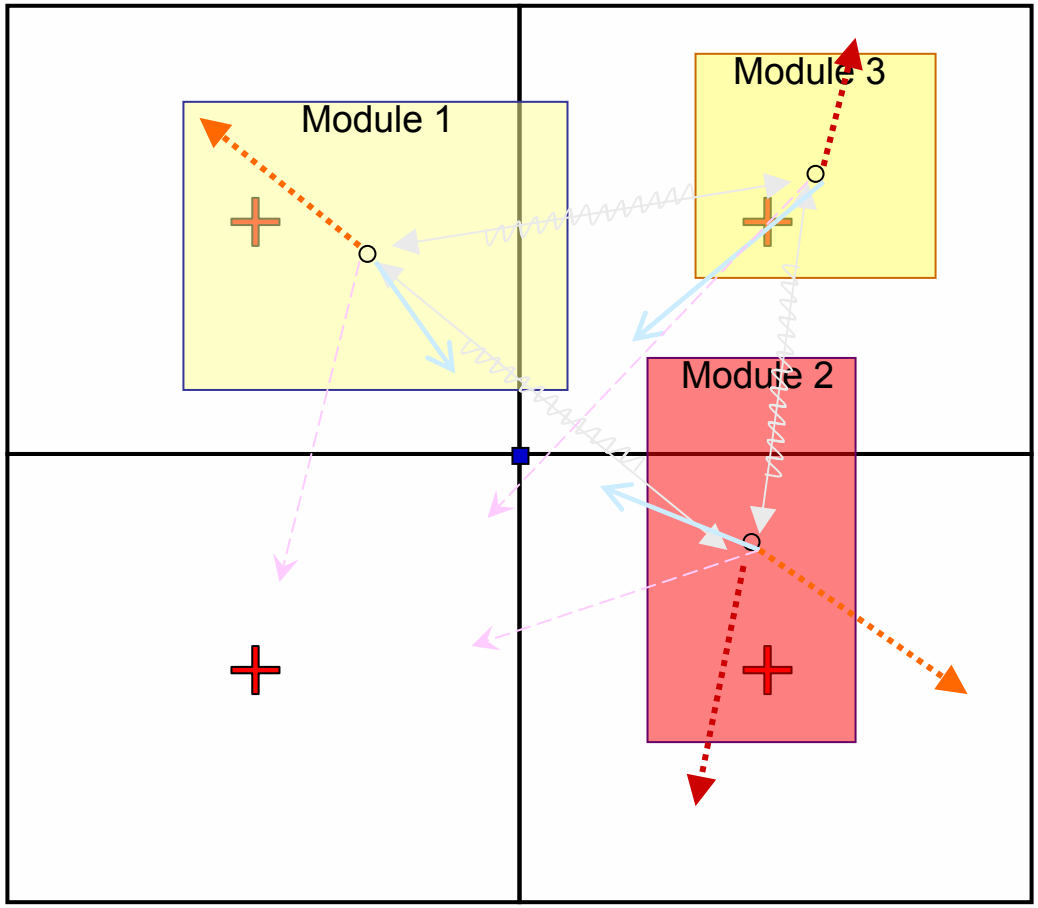
Power Pin

Net Force

Center Force

Density Force

Force-Directed Floorplanning



Power Pin

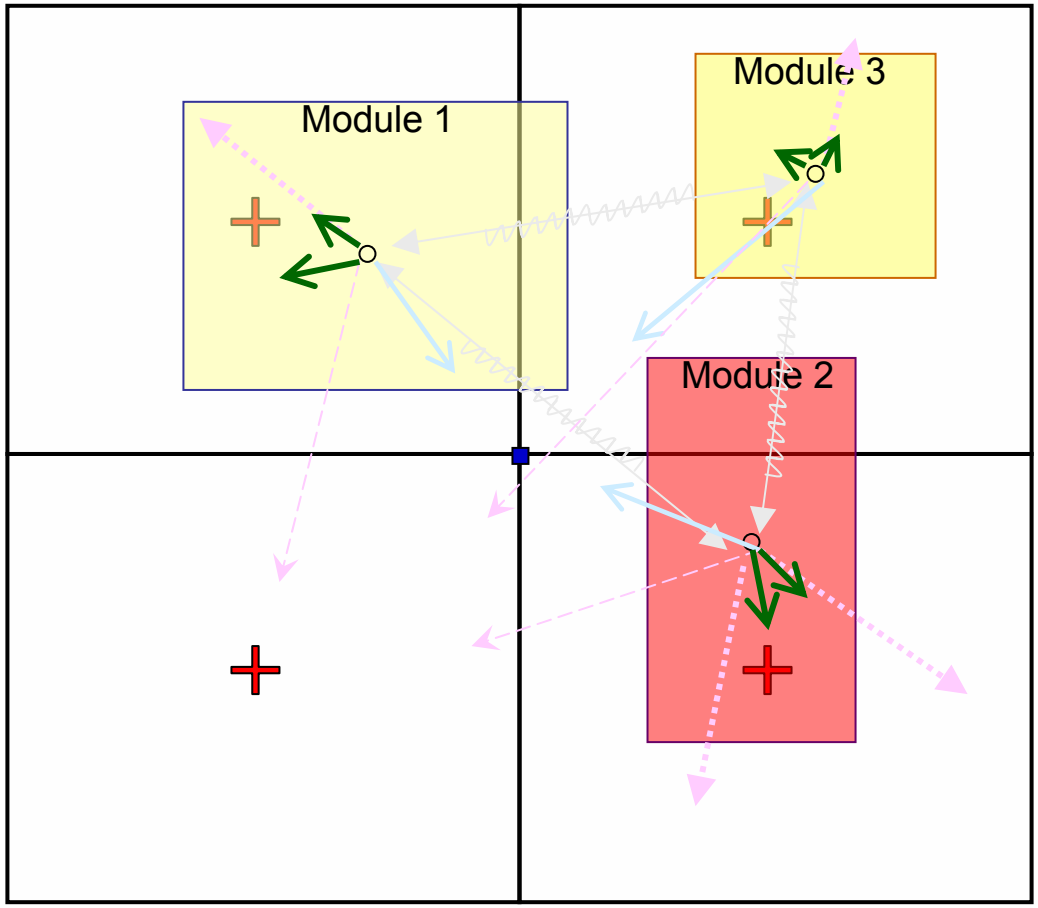
Net Force

Center Force

Density Force

Correlation Force (γ)

Force-Directed Floorplanning



Power Pin

Net Force

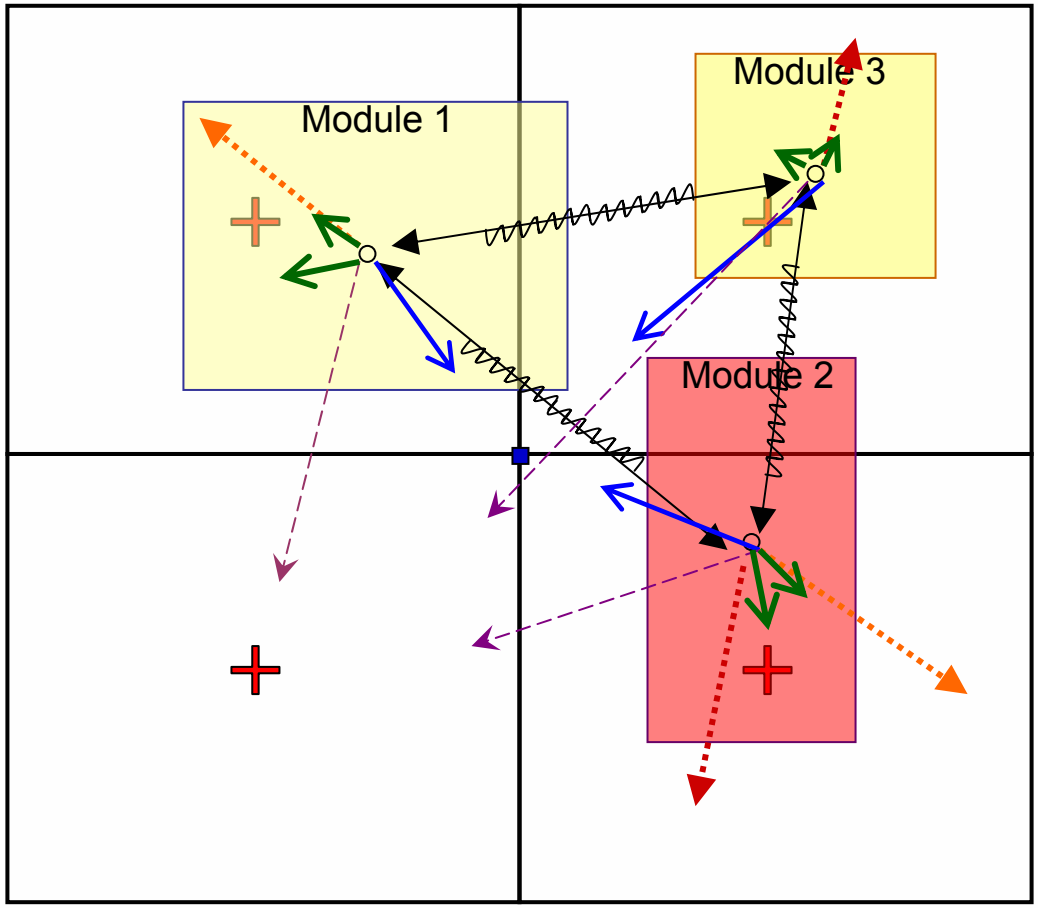
Center Force

Density Force

Correlation Force (γ)

Pin Force (α) x, y directions

Force-Directed Floorplanning



Power Pin

Net Force

Center Force

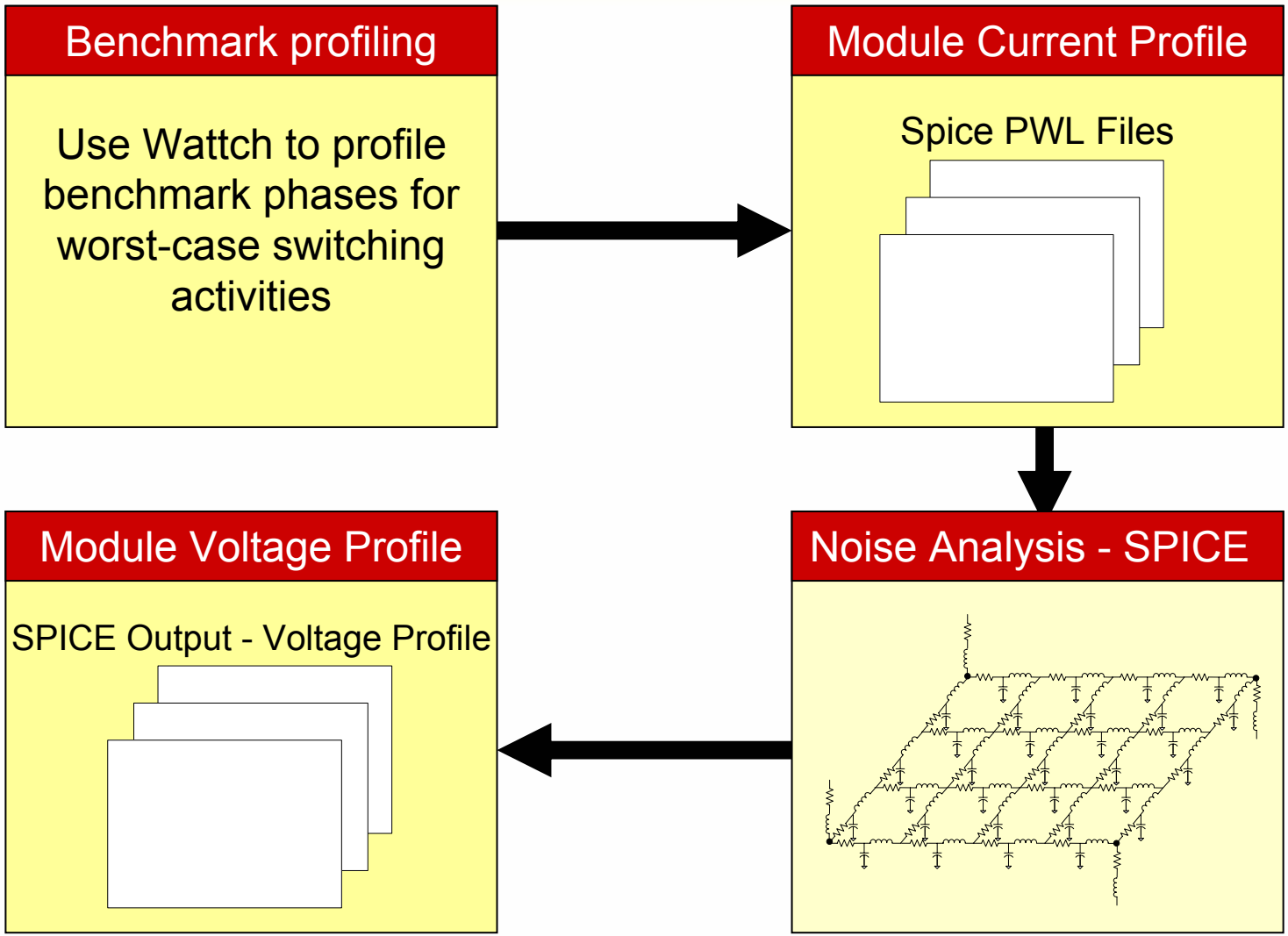
Density Force

Correlation Force (γ)

Pin Force (α) x, y directions

$$F_{tot} = \lambda \cdot F_{net} + \theta \cdot F_{cen} + \kappa \cdot F_{den} + \mu \cdot F_{cor} + \rho \cdot F_{pin}$$

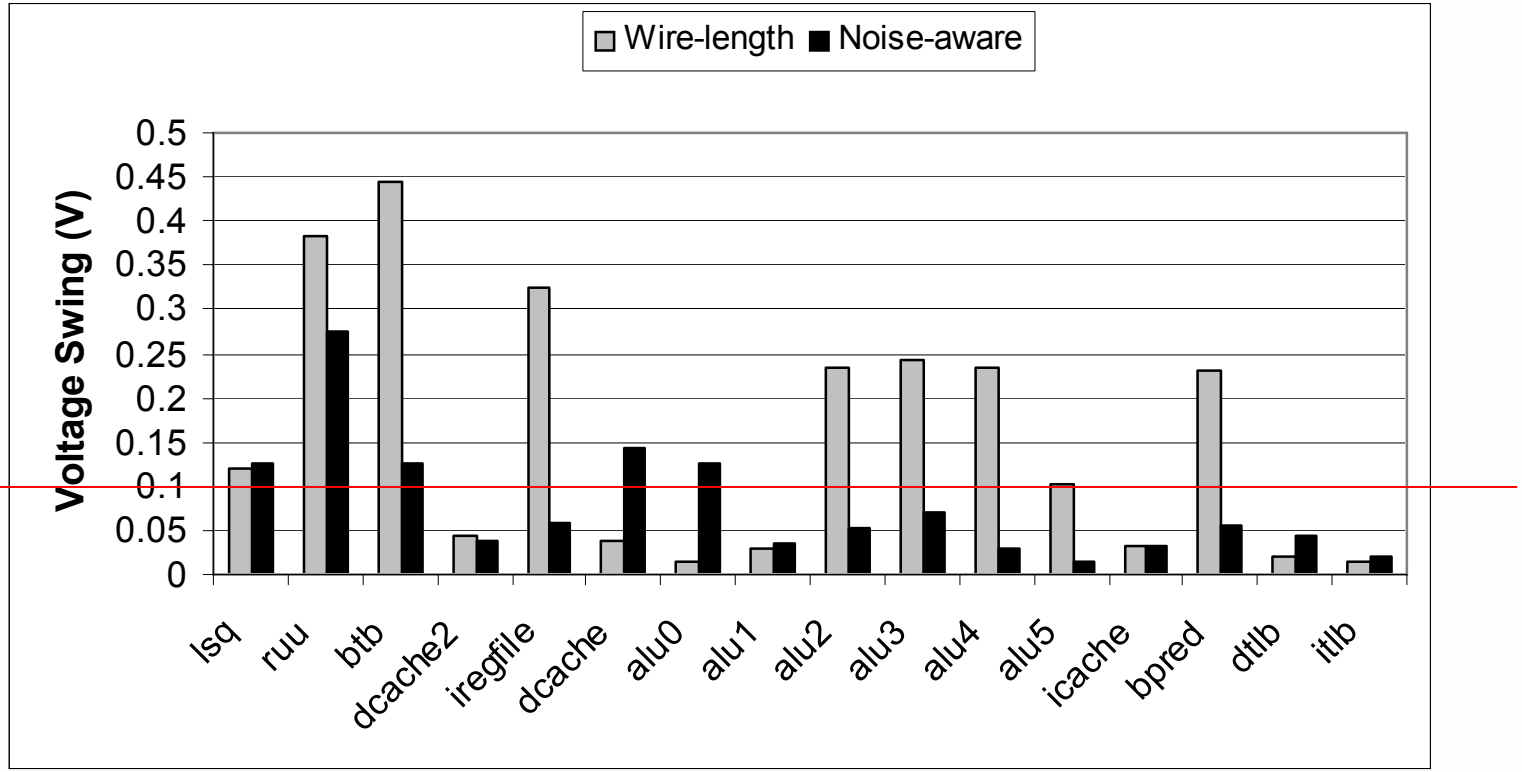
Noise (ΔV) Analysis Method



Simulated Processor Model

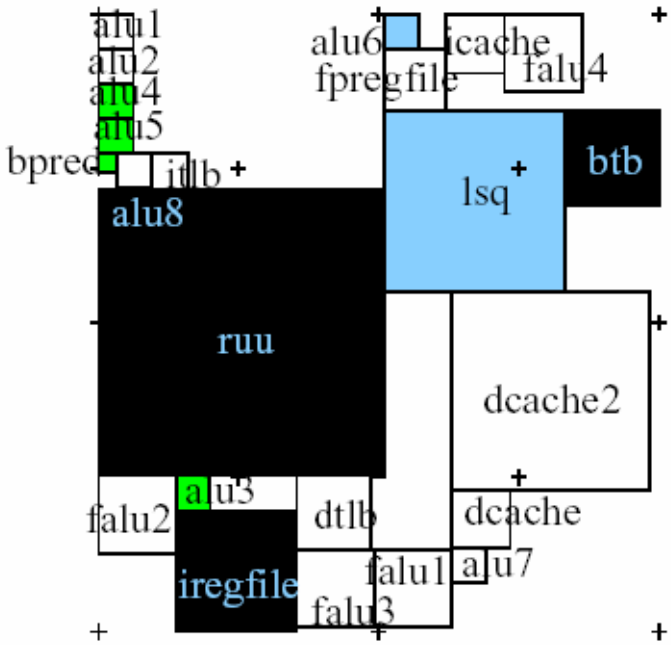
Parameters	Values
Fetch/Decode Width	8-wide
Issue/Commit Width	8-wide
Branch Predictor	Combining 16K-Entry Metatable Bimodal: 16K Entries 2-Level: 14 bit BHR, 16K entry PHT
BTB	4-way, 4096 sets
L1 I$ & D$	16KB 4-Way 64B Line
I-TLB & D-TLB	128 Entries
L2 Cache	256KB, 8-way, 64B Line
L1/L2 Latency	1 cycle/6 cycles
Main Memory Latency	500 cycles
LSQ Size	64 entries
RUU Size	256 entries

Power Supply Noise

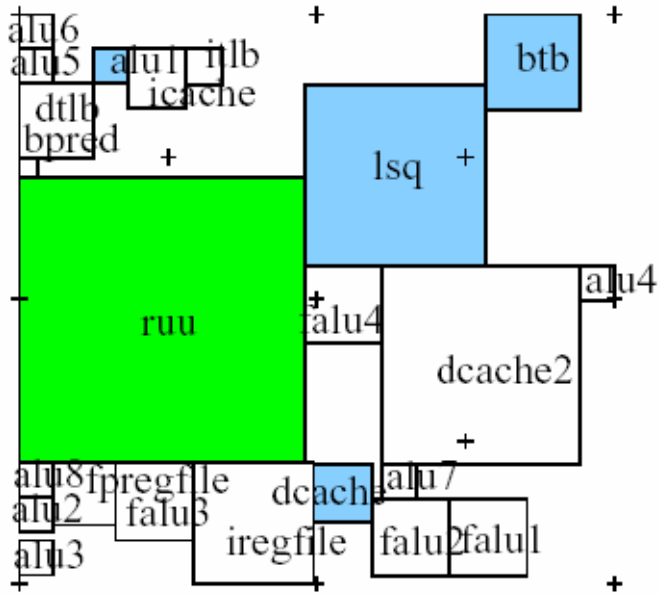


- Most worst-case voltage swings are pushed below margin
- For exceptions, most are still below the threshold (10%), and the remaining are marginal
- Outliers due to
  - Other ALUs (other than alu0) have higher correlation (γ)
  - Dcache does not have high correlation (γ) with others

Noise Tolerance of Microarch Modules



(a) Wire-length Floorplan



(b) Noise-aware Floorplan

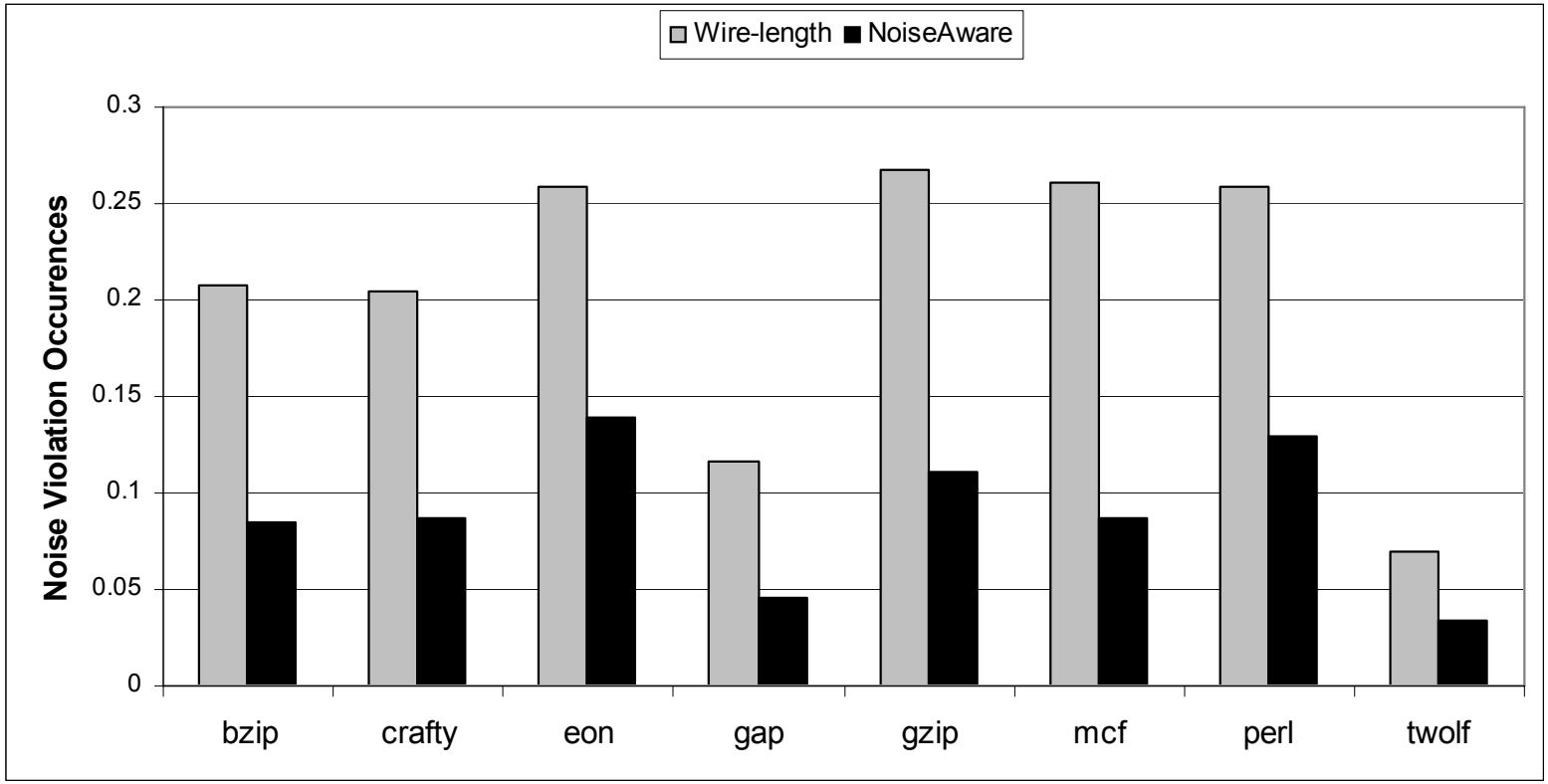
Noise > 30 %

Noise 20-30 %

Noise > 10-20 %

Below Noise Margin

Noise Violation Frequency



- Noise margin violations are reduced by more than half
- Illustrates the potential for better performance in presence of a dynamic di/dt control mechanism

Dealing with Worst-Case

- Even with Noise-Direct, worst-case must be guaranteed

- We advocate: Noise Direct + Dynamic di/dt control
  - Details in our paper in *MICRO-39, 2006*
  - Use decay counters for each module
  - Control simultaneous gating
    - Based on a queue-based controller in each power domain
    - Throttle gating when threshold is exceeded
  - Other synergistic approaches
    - Pre-emptive ALU gating
    - Progressive gating for large modules
    - Based on a queue-based controller in each power domain
    - Throttle gating when threshold is exceeded



Conclusion

- Traditional design methodologies continue to be inefficient
- Inductive noise no longer a design afterthought
- Decaps consume chip real-estate, and contribute to leakage, eroding benefits from clock-gating
- Our research proposes
  - Cooperative *physical design* and *microarchitecture* techniques
  - Noise-Direct: Floorplanning for the average-case
  - Guarantee worst case through dynamic di/dt control

Thank you

Home of the 1996 Olympic Village

Georgia Tech

http://arch.ece.gatech.edu
http://www.3D.gatech.edu

GSRC C2S2 MARCO

Georgia Tech MARS