Supported by
DARPA

Warped Gates: Gating Aware Scheduling and Power Gating for GPGPUs

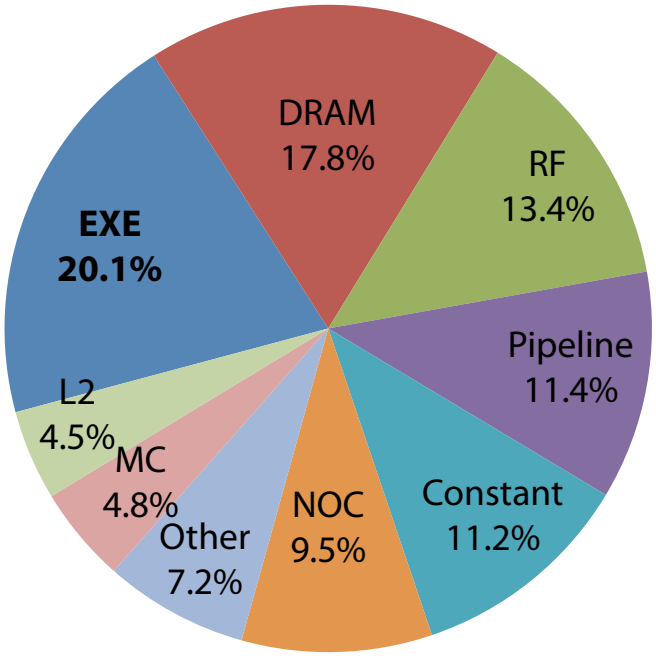
*Mohammad Abdel-Majeed** **Daniel Wong***
Murali Annavaram

Ming Hsieh Department of Electrical Engineering
University of Southern California

* Equal Contribution

Problem Overview

- Execution unit accounts for majority of energy consumption in GPGPU, even more than Mem and Reg!
- Leakage energy is becoming a greater concern with technology scaling

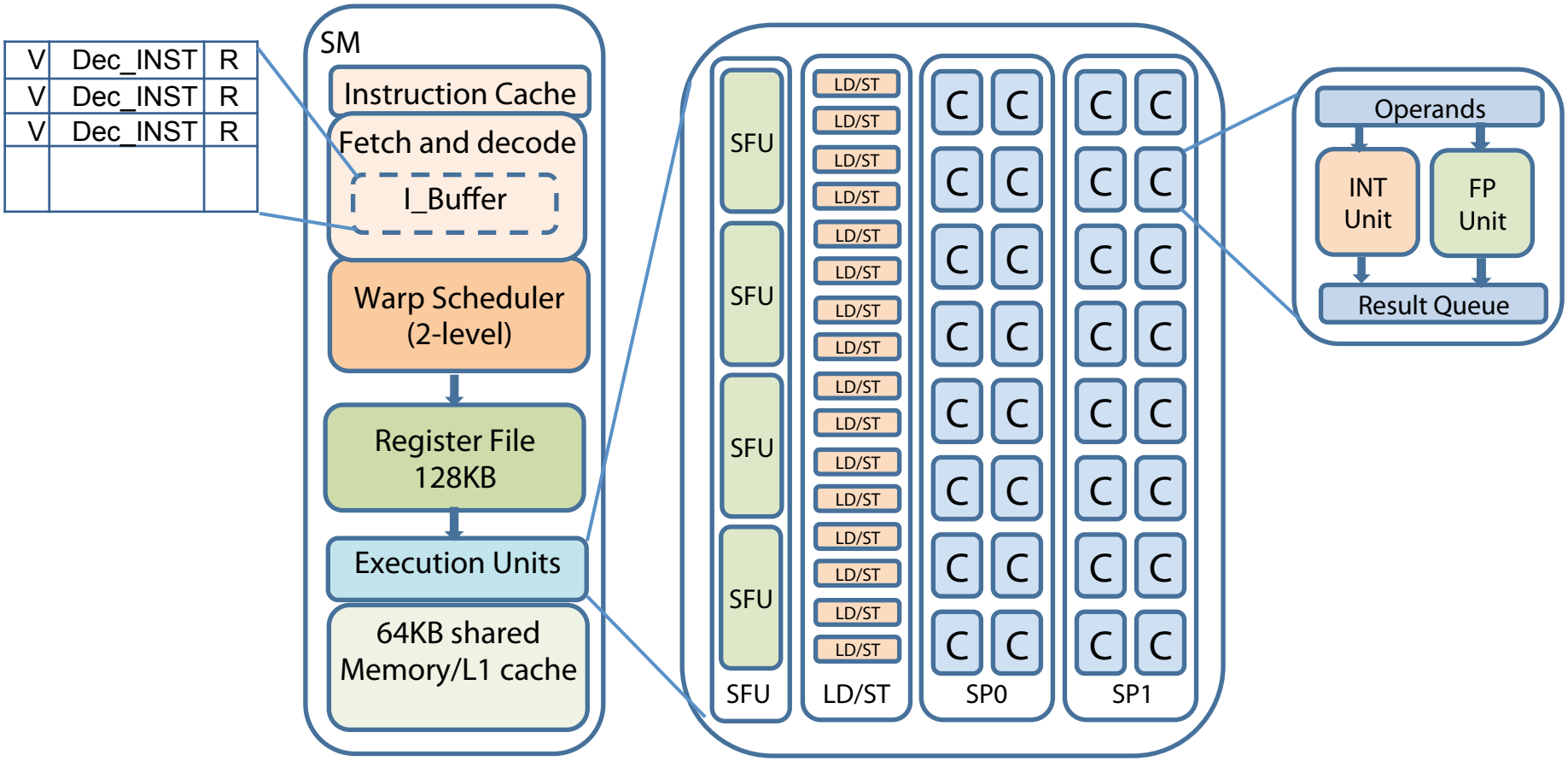


Component Energy Breakdown for GTX480[1]

Traditional microprocessor power gating techniques are ineffective in GPGPUs

[1] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N. S. Kim, T. M. Aamodt, and V. J. Reddi, "GPUWattch: enabling energy optimizations in GPGPUs," presented at the ISCA '13: Proceedings of the 40th Annual International Symposium on Computer Architecture, 2013.

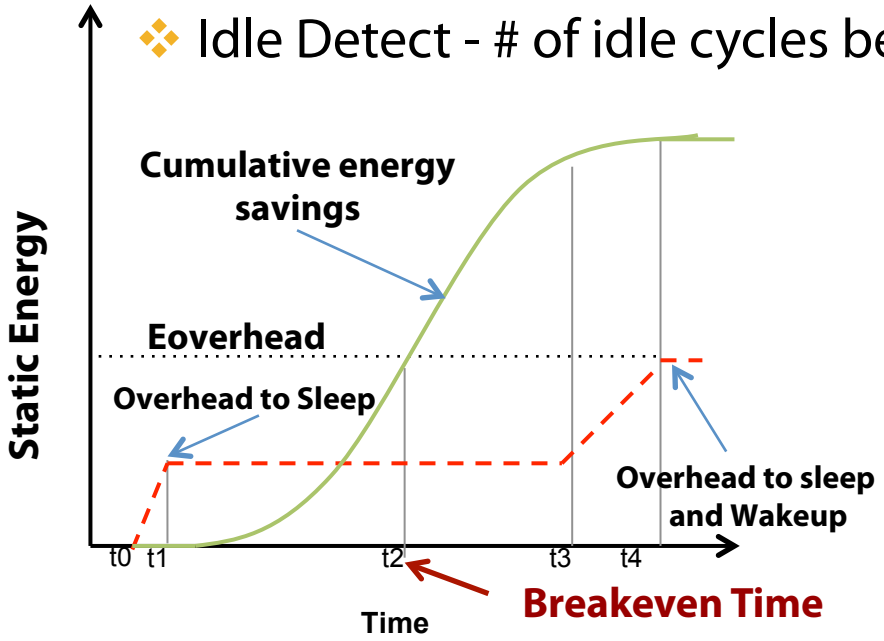
GPGPU Overview (GTX480)

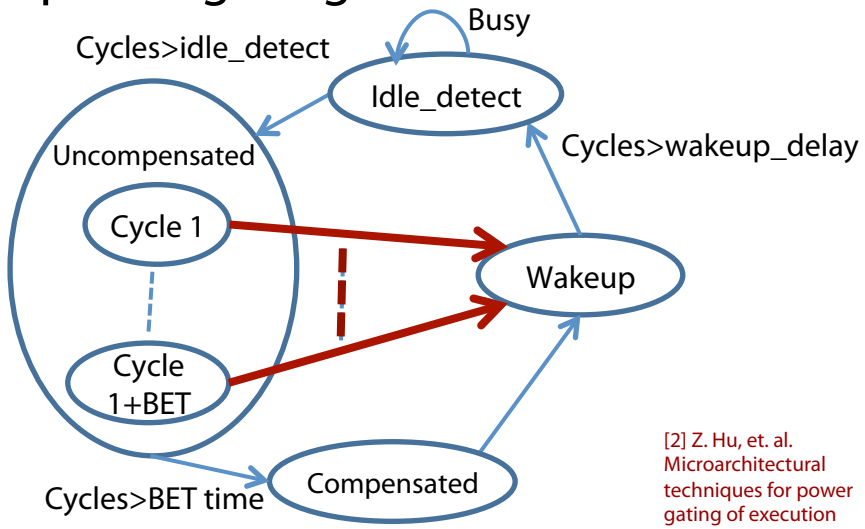


- SP accounts for 98% of Execution Unit Leakage Energy
- Execution units account for 68% of total on chip area

Power Gating Overview

- Cuts off leakage current that flows through a circuit block
- Power gate at SP granularity
- Important Parameters:
  - Wakeup Delay – Time to return to Vdd (3 cycles)
  - Breakeven Time – # of consecutive power gated cycles required to compensate PG energy overhead (9-24 cycles)
  - Idle Detect - # of idle cycles before power gating[2]



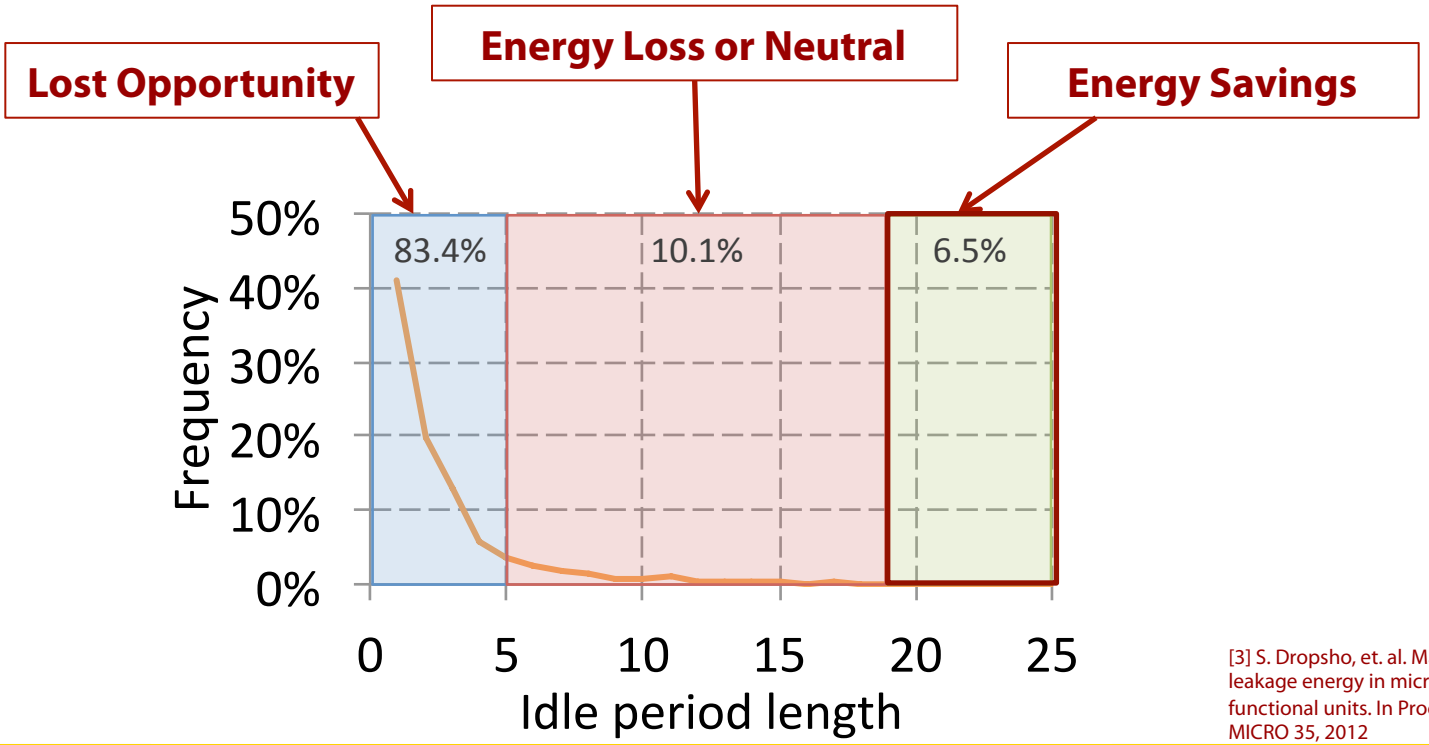


[2] Z. Hu, et. al. Microarchitectural techniques for power gating of execution units. In ISLPED '04.

Power Gating Challenges in GPGPUs

Power Gating Challenges in GPGPUs

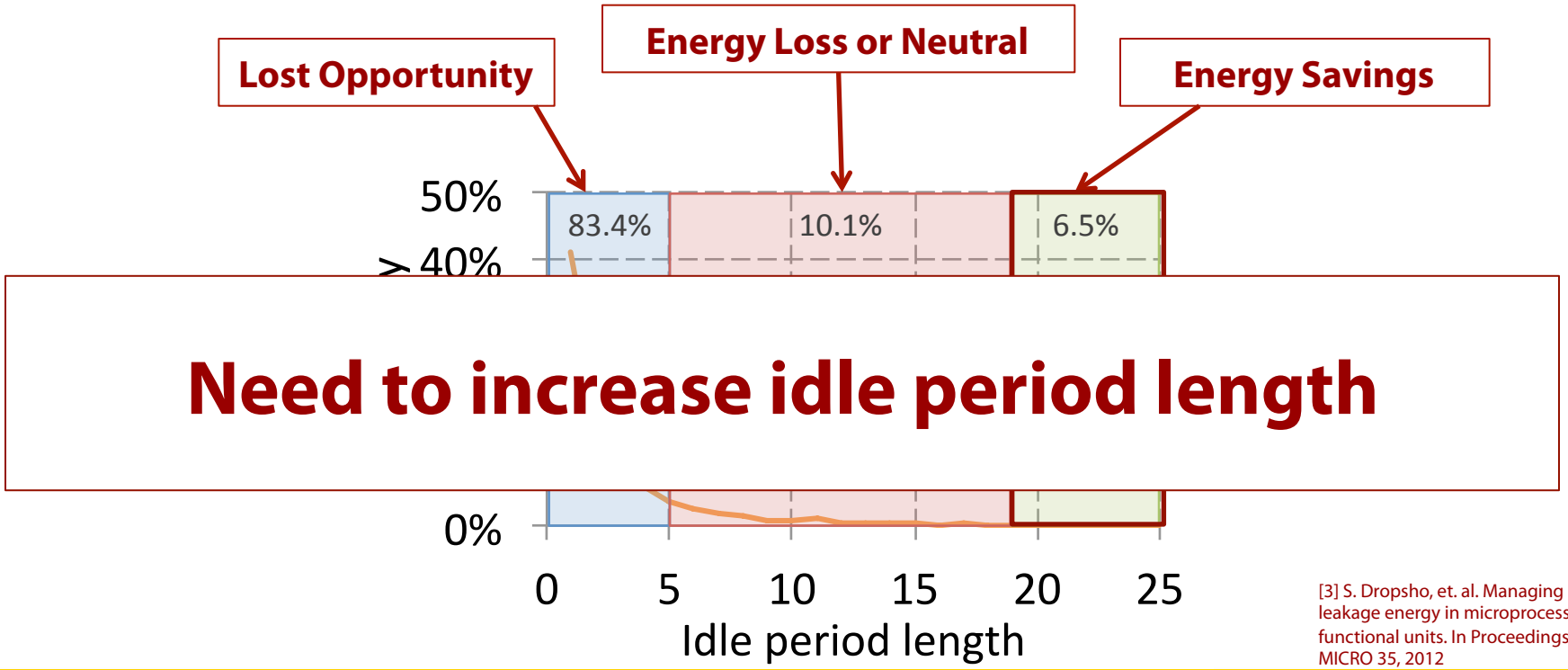
- Traditional microprocessors experience idle periods many 10s of cycles long[3]
- Int. Unit Idle period length distribution for `hotspot`
  - Assume 5 idle detect, 14 BET



[3] S. Dropsho, et. al. Managing static leakage energy in microprocessor functional units. In Proceedings of the MICRO 35, 2012

Power Gating Challenges in GPGPUs

- Traditional microprocessors experience idle periods many 10s of cycles long[3]
- Int. Unit Idle period length distribution for `hotspot`
  - Assume 5 idle detect, 14 BET



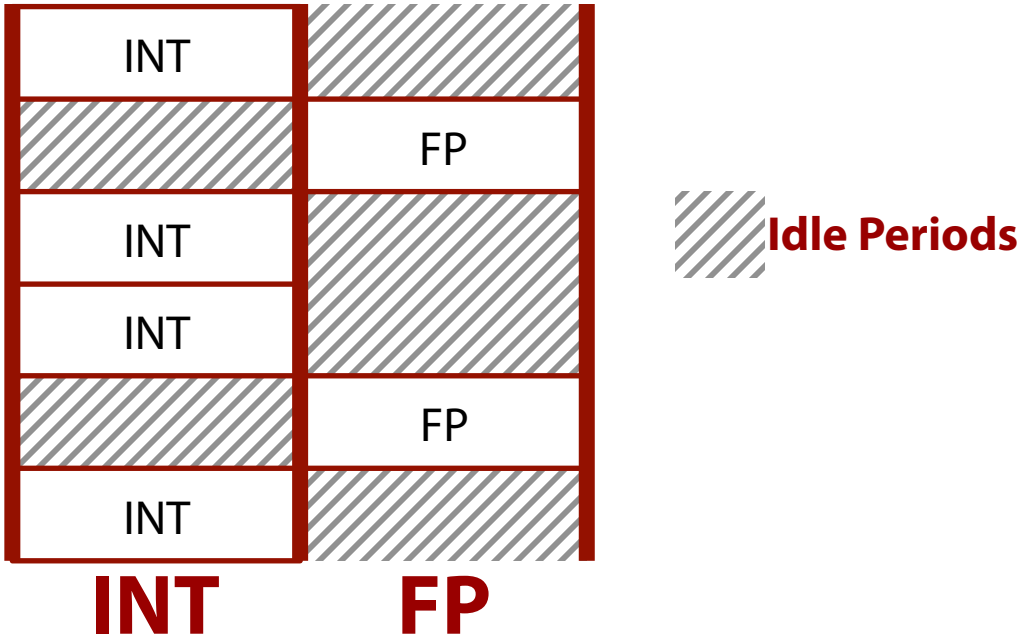
Need to increase idle period length

[3] S. Dropsho, et. al. Managing static leakage energy in microprocessor functional units. In Proceedings of the MICRO 35, 2012

Warp Scheduler Effect on Power Gating

Need to coalesce warp issues by resource type

- Idle periods interrupted by instructions that are greedily scheduled



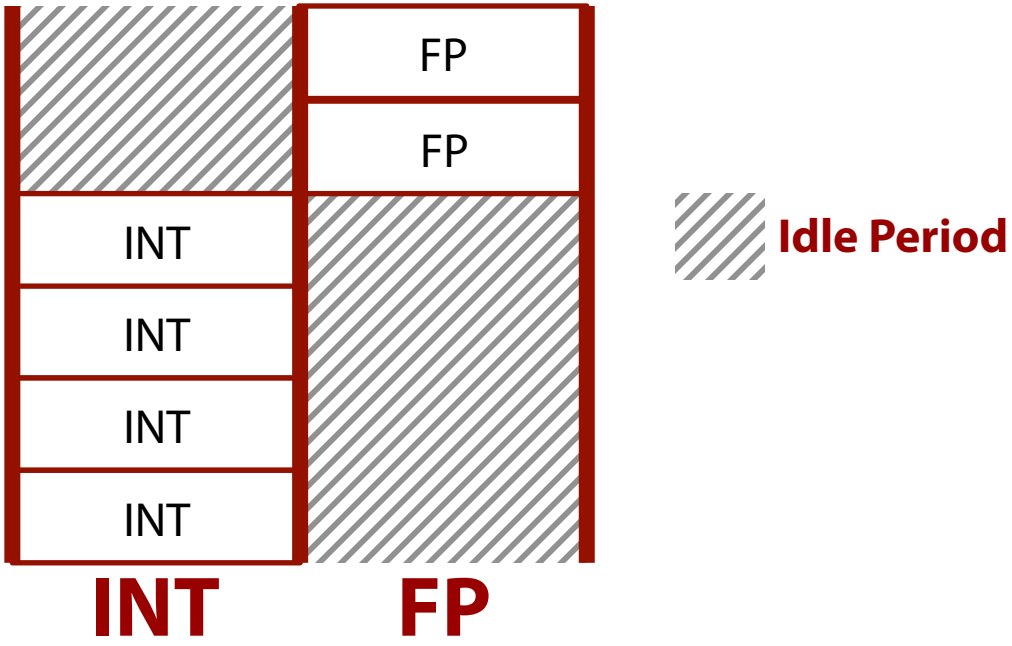
GATES:

Gating Aware Two-level Scheduler

- Issue warps based on execution unit resource type

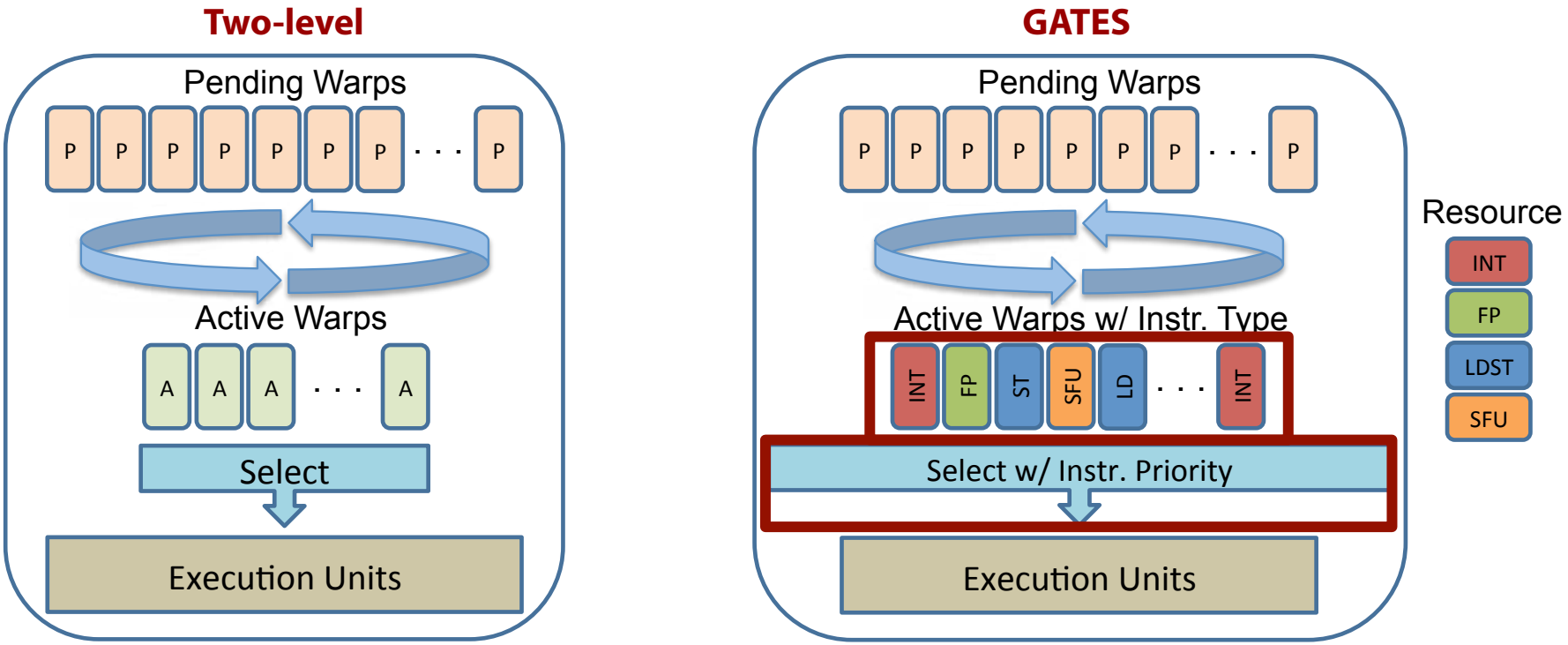
Gating Aware Two-level Scheduler (GATES)

- Idle periods are coalesced



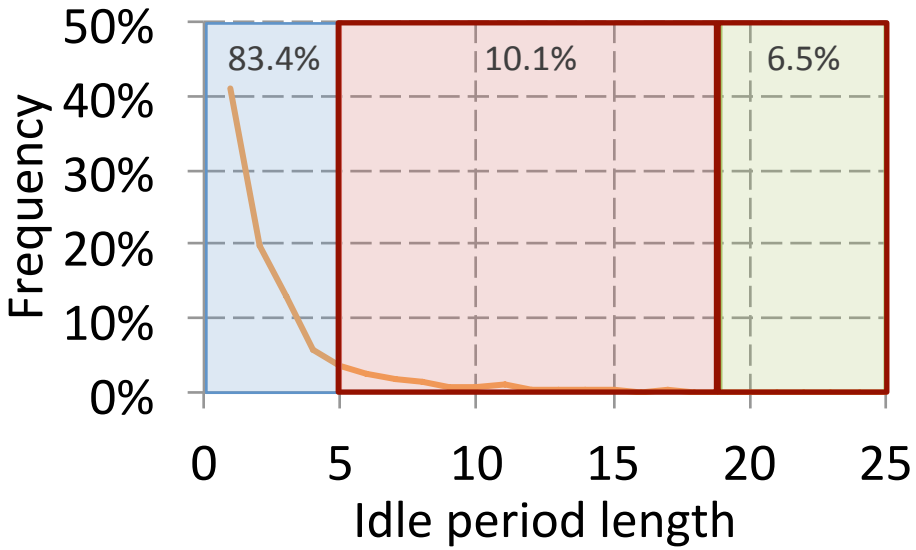
Gating Aware Two-level Scheduler (GATES)

- Per instruction type active warps subset
- Instruction Issue Priority
- Dynamic priority switching
  - Switch highest priority when it out of ready warps

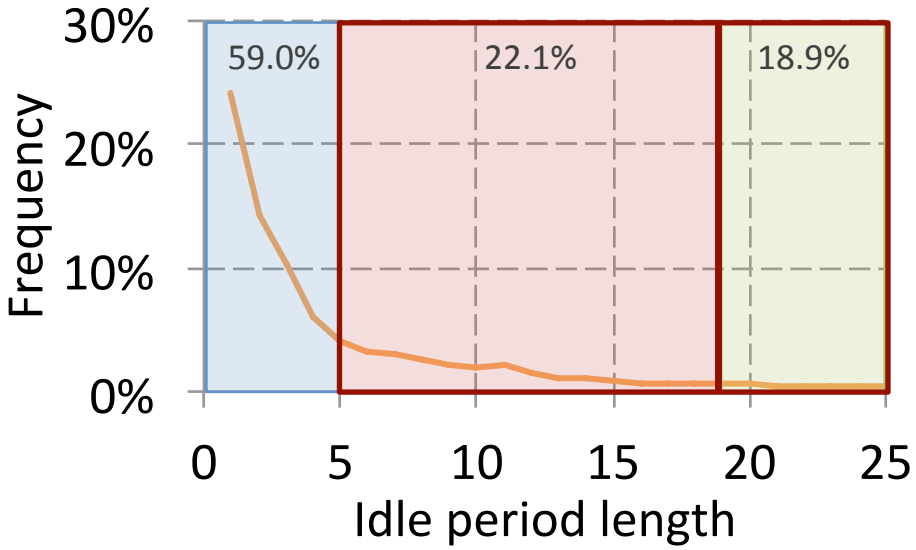


Effect of GATES on Idle Period Length

Need to further stretch idle periods



Two-level



GATES

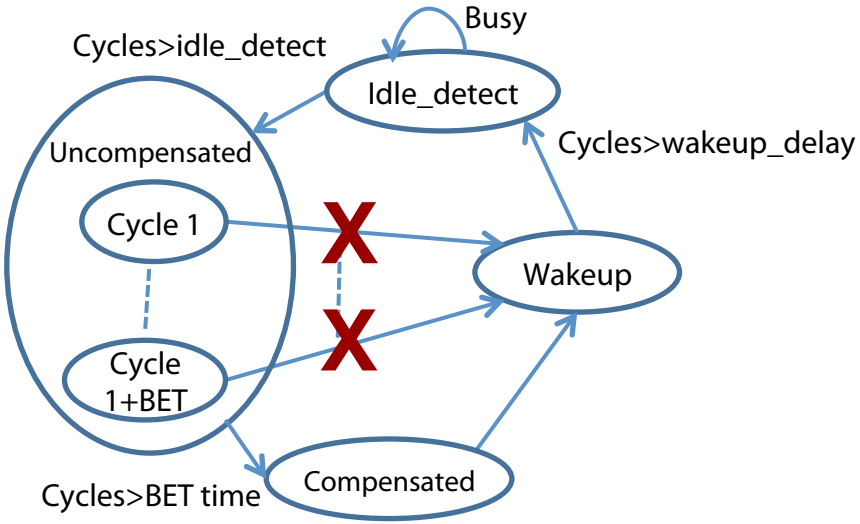
- ~3x increase in positive power gating events
- ~2x increase in negative power gating events

Blackout Power Gating

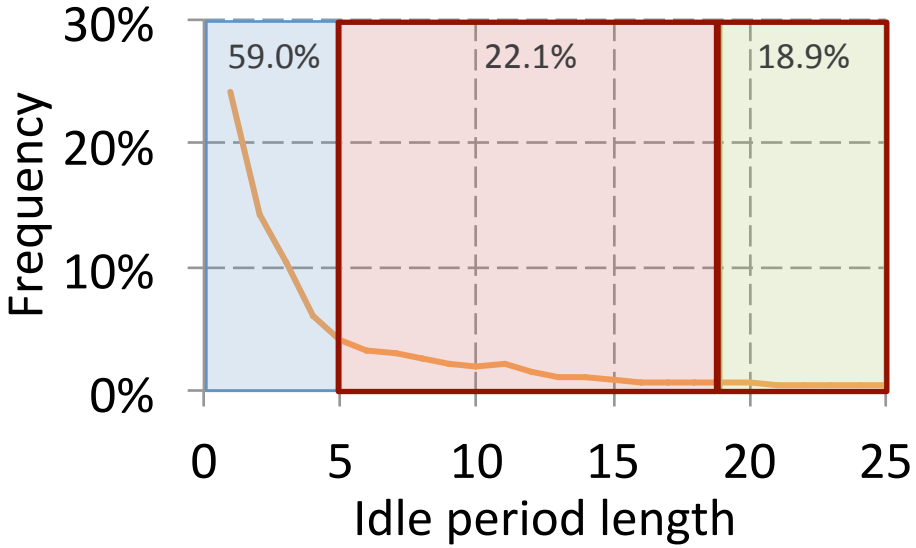
- Forced idleness of execution units to meet BET

Blackout Power Gating

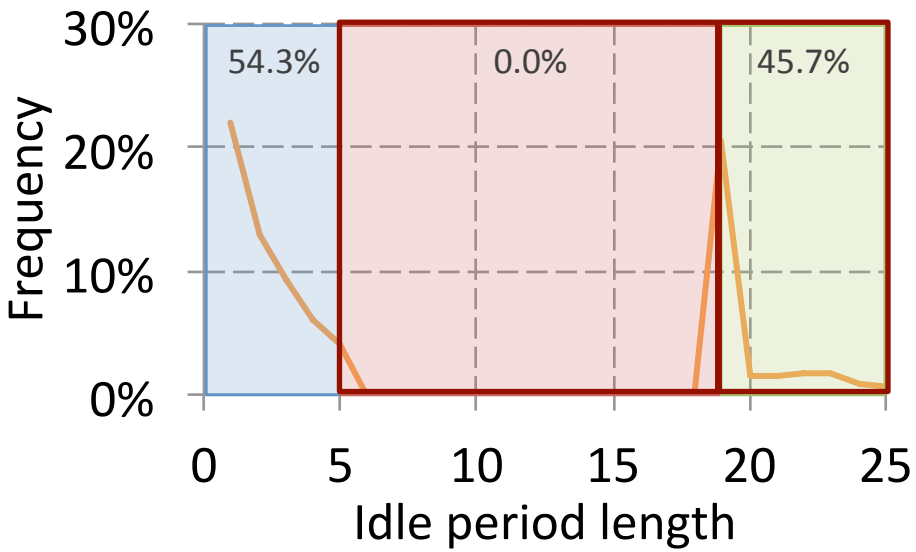
- Force idleness until break even time has passed
  - Even when there are pending instructions
- Would this not cause performance loss?
  - No, because of GPGPU-specific large heterogeneity of execution units and good mix of instruction types



Blackout Power Gating



GATES

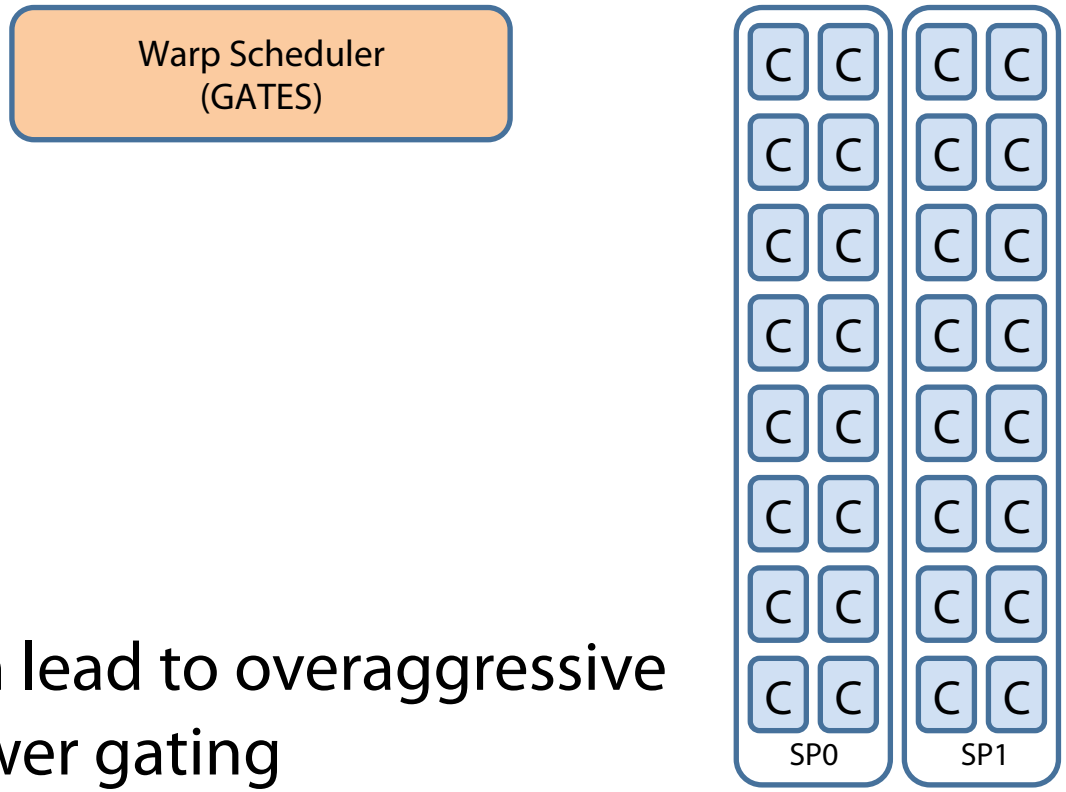


GATES + Blackout

- ~2.4x increase in positive PG events over GATES (GATES ~3x w.r.t. baseline)

Blackout Policies

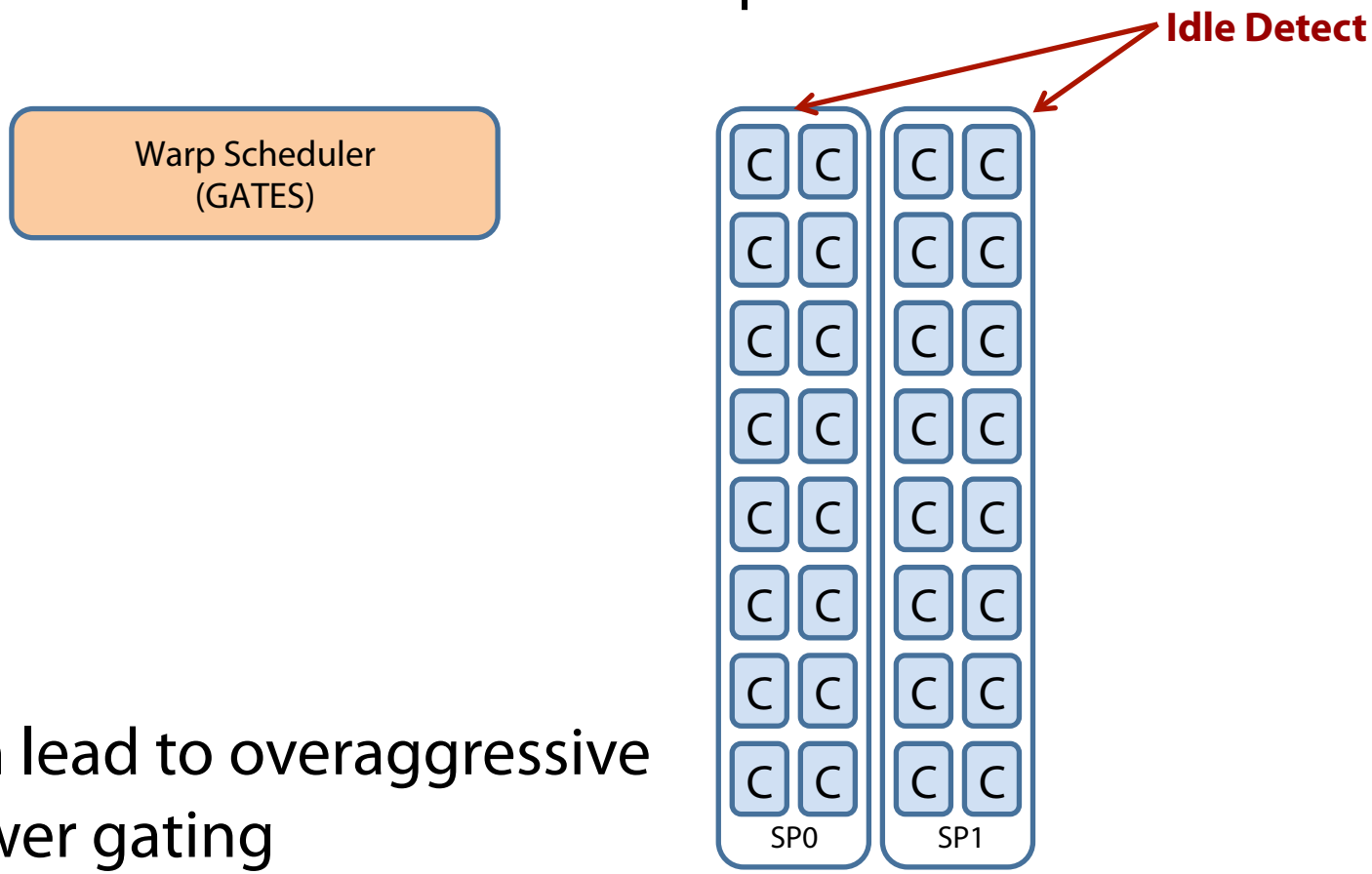
- Naïve Blackout
  - GATES and Blackout is independent



- Can lead to overaggressive power gating

Blackout Policies

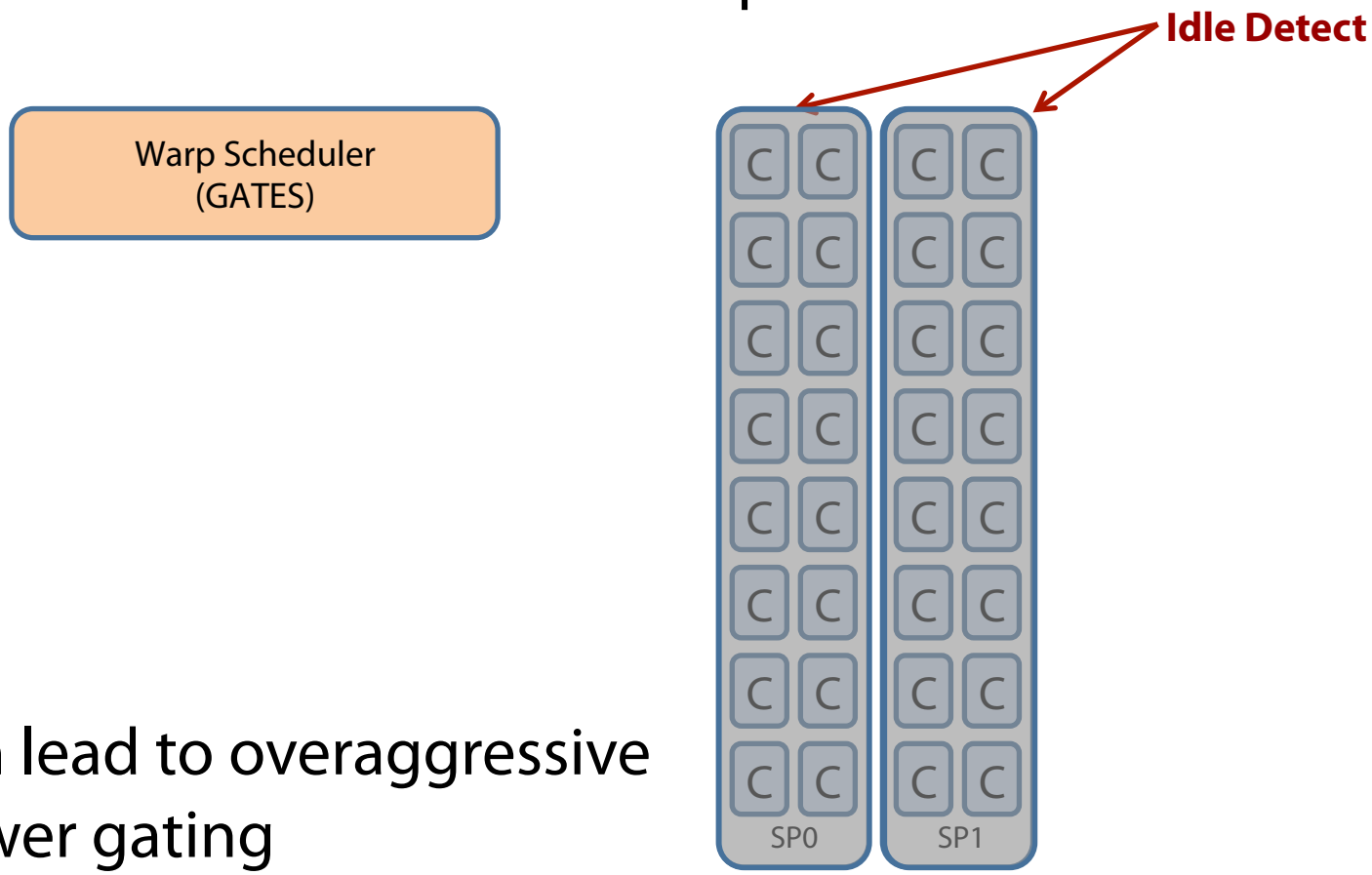
- Naïve Blackout
  - GATES and Blackout is independent



- Can lead to overaggressive power gating

Blackout Policies

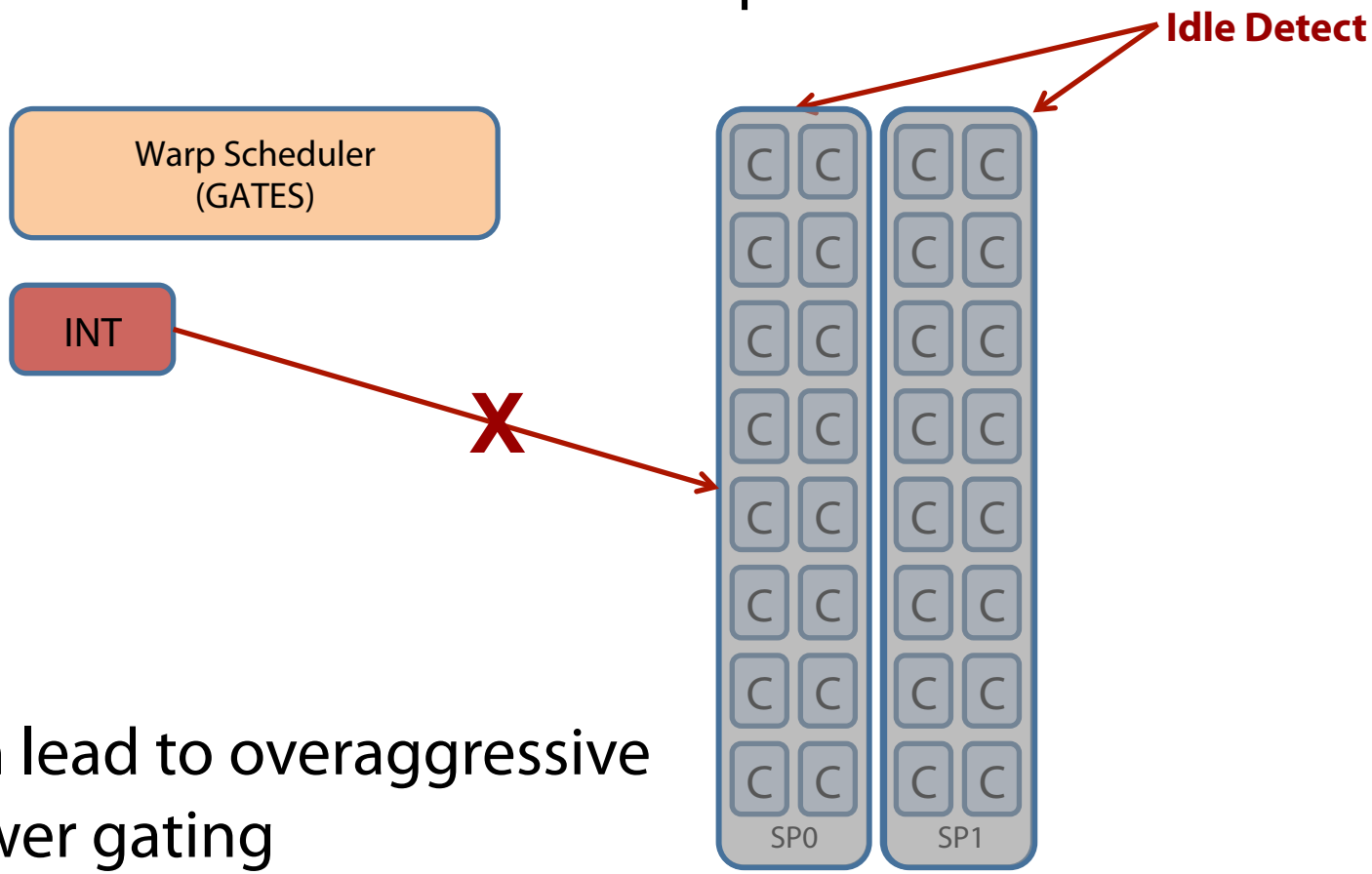
- Naïve Blackout
  - GATES and Blackout is independent



- Can lead to overaggressive power gating

Blackout Policies

- Naïve Blackout
  - GATES and Blackout is independent



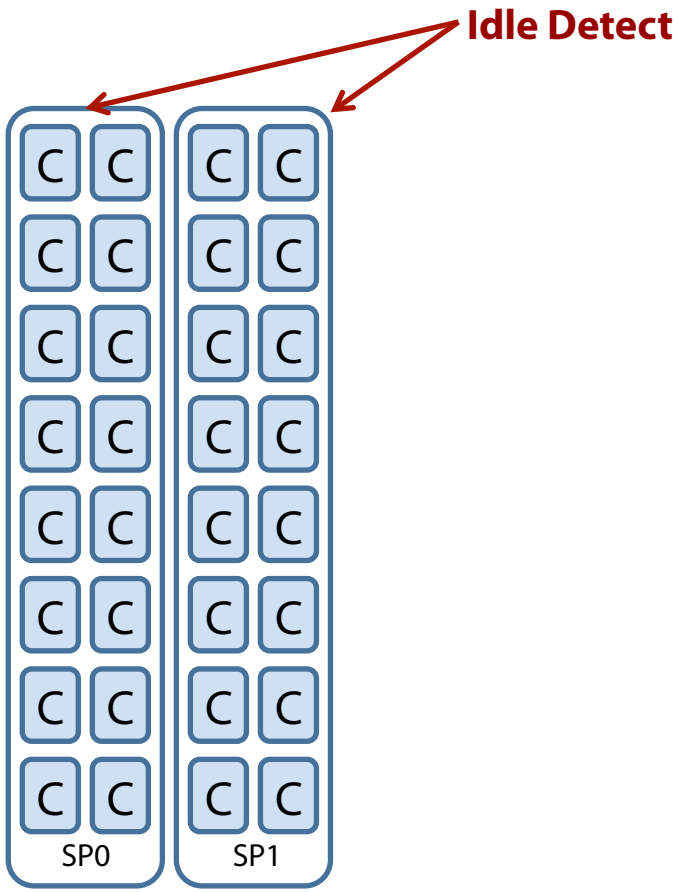
- Can lead to overaggressive power gating

Blackout Policies

Coordinated Blackout

Warp Scheduler
(GATES)

Dynamic priority switching
is Blackout aware

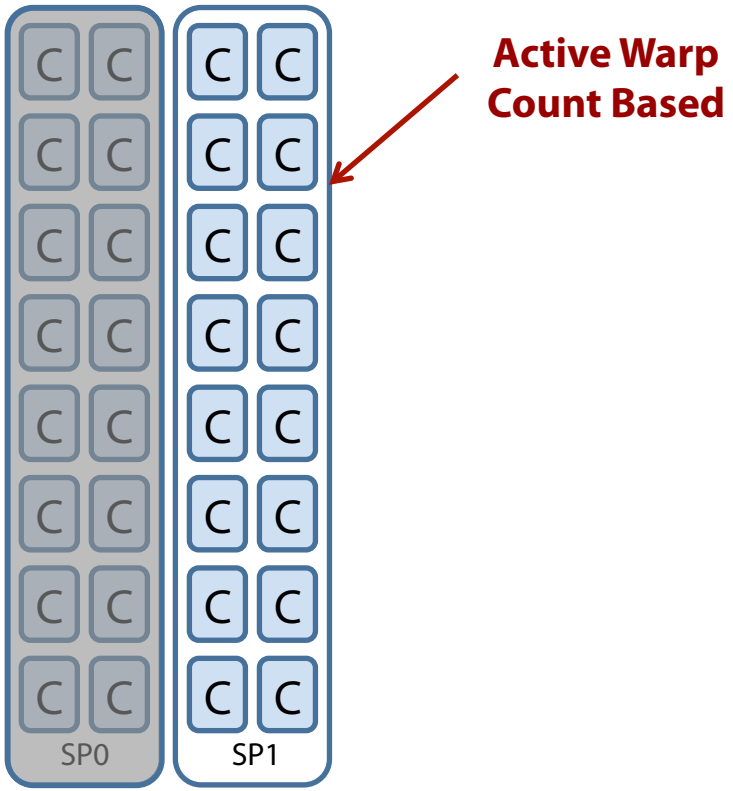


Blackout Policies

- Coordinated Blackout

Warp Scheduler
(GATES)

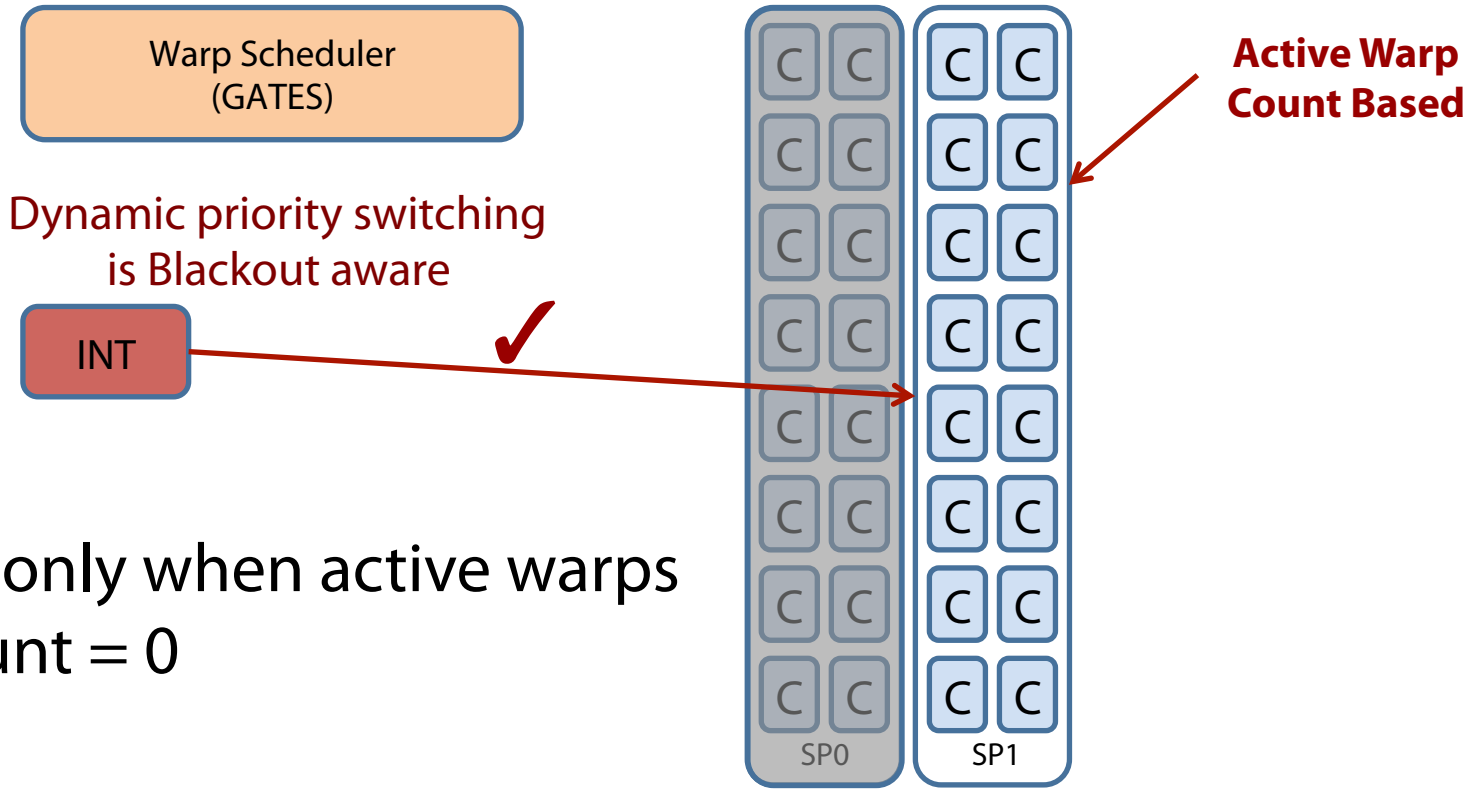
Dynamic priority switching
is Blackout aware



- PG only when active warps count = 0

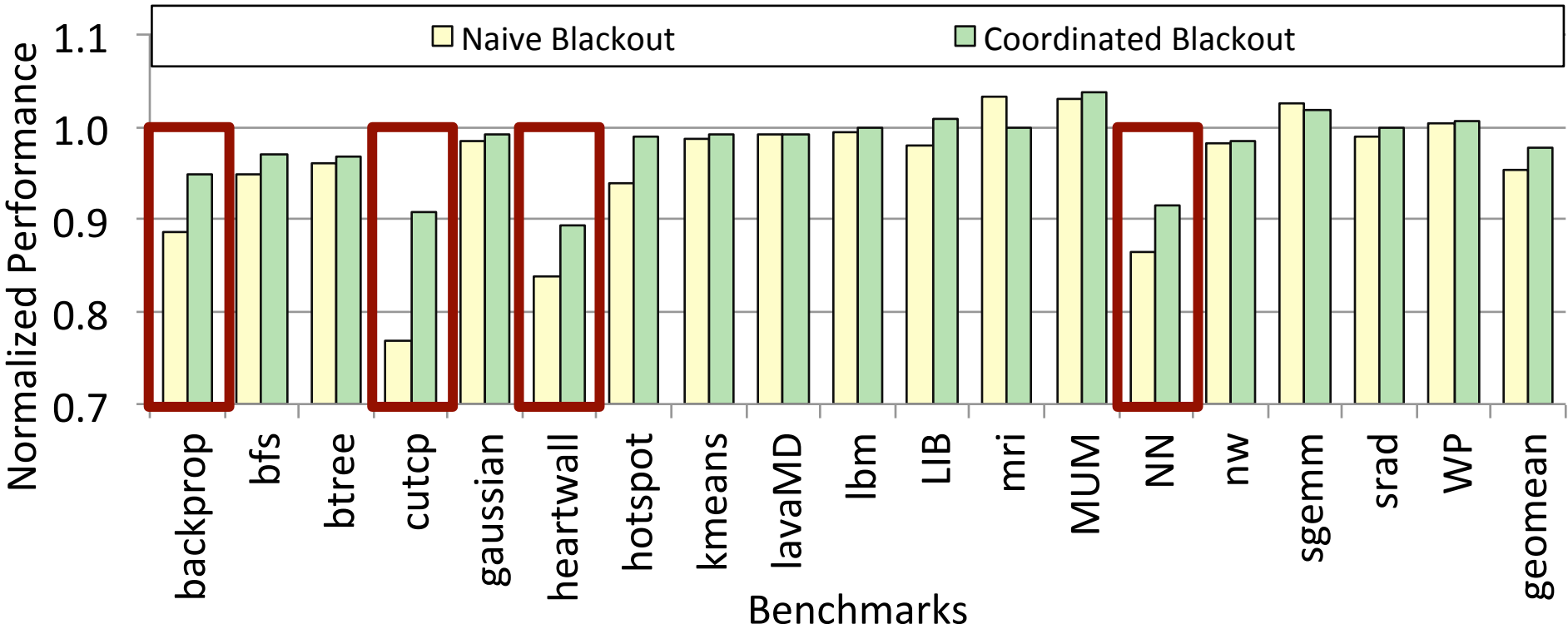
Blackout Policies

- Coordinated Blackout



- PG only when active warps count = 0

Impact of Blackout



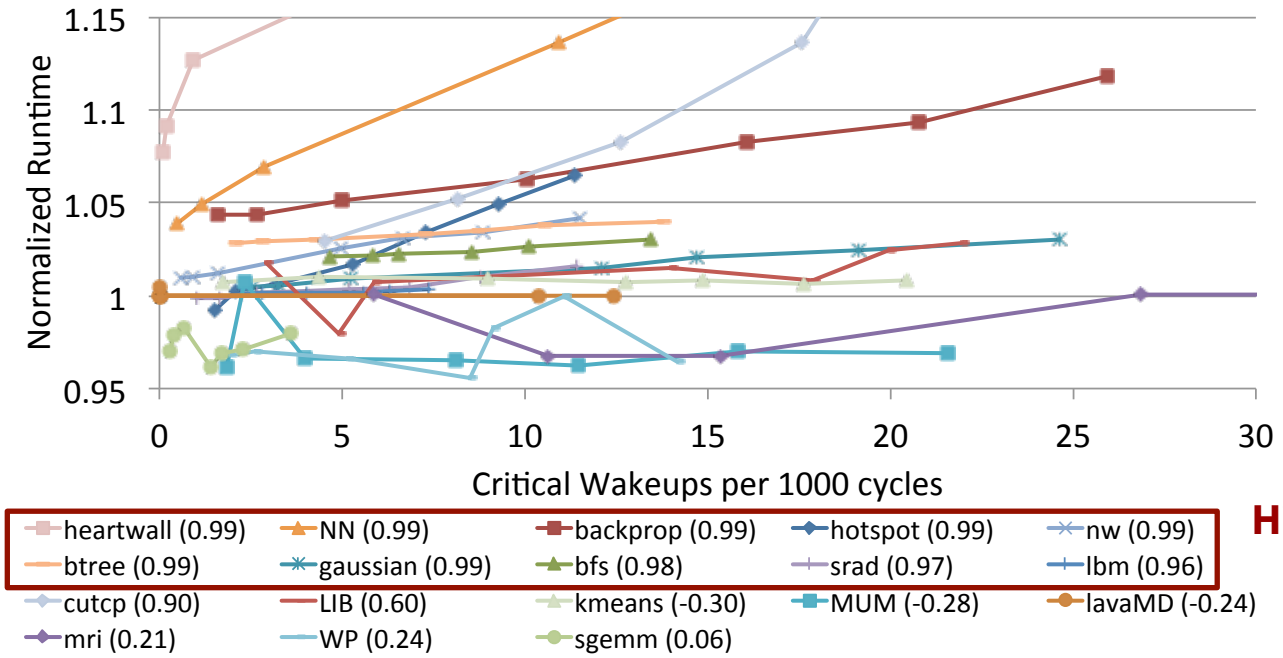
- Some benchmarks still show poor performance
  - Not enough active warps to hide forced idleness
- Goal is as close to 0% overhead as possible

Adaptive Idle Detect

Reducing Worst Case Blackout Impact

Adaptive Idle Detect

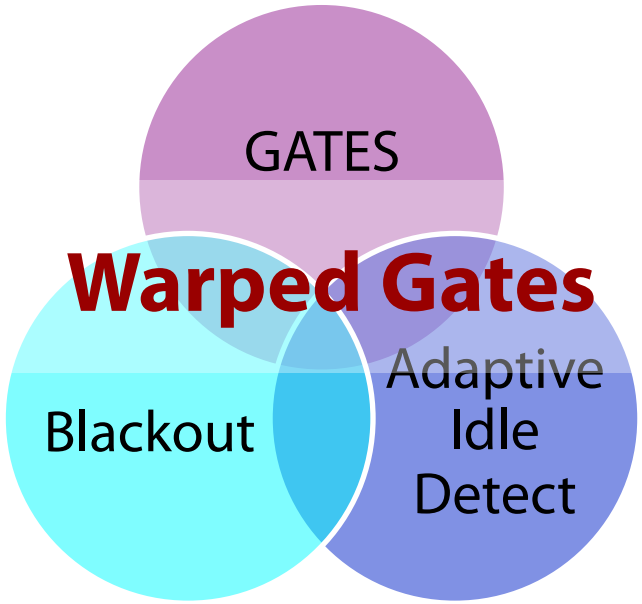
- Dynamically change idle detect to avoid aggressive PG
- Infer performance loss due to Blackout
  - "Critical Wakeup" – Wakeup that occur the moment blackout period ends



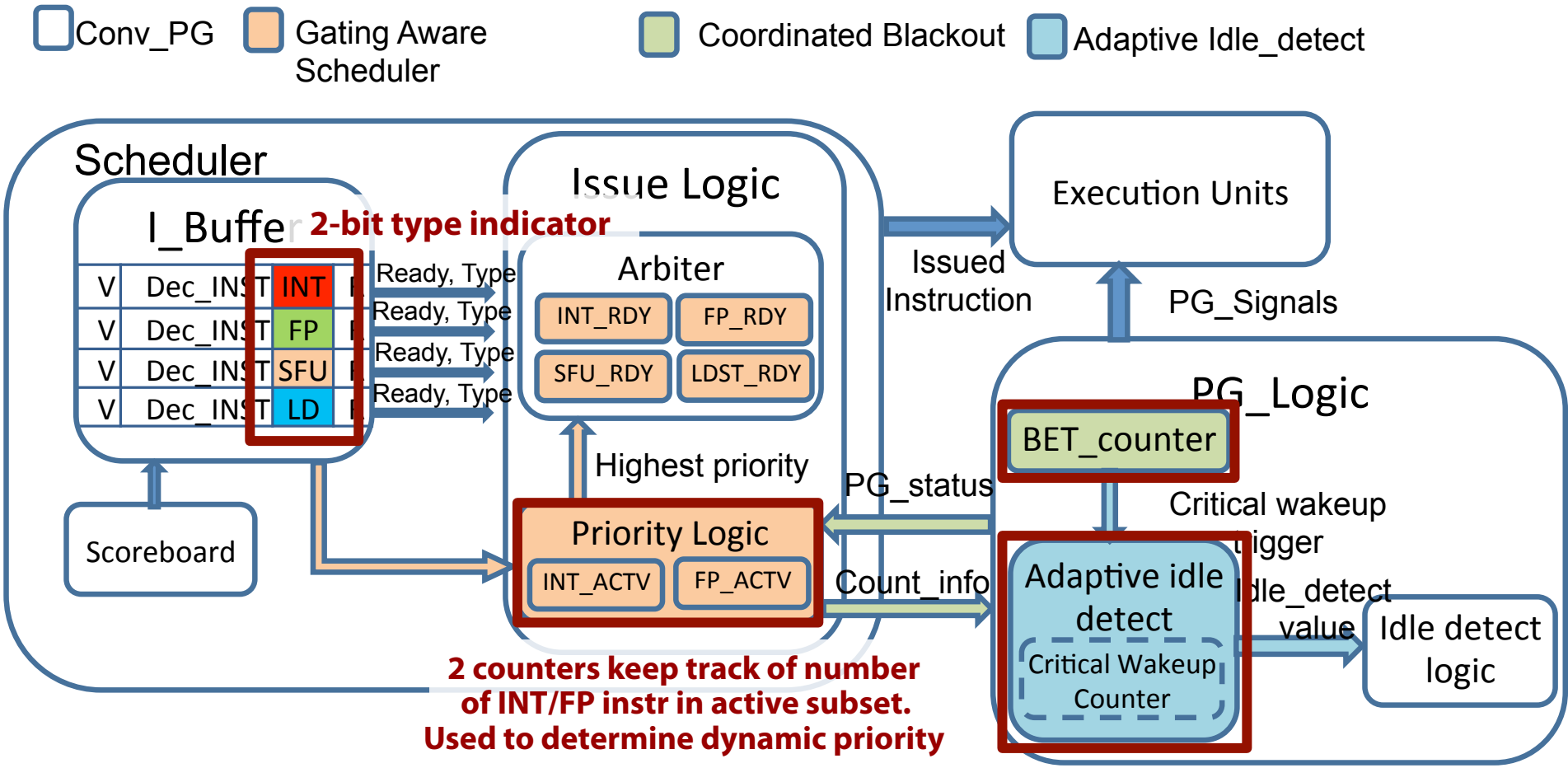
High Correlation vs Runtime

Adaptive Idle Detect

- Independent idle detect values for INT and FP pipelines
- Break execution time into epoch (1000 cycles)
- If critical wakeup > threshold, idleDetect++
- Conservatively decrement idleDetect every 4 epochs
- Bound idle detect between 5 – 10 cycles



Architectural Support

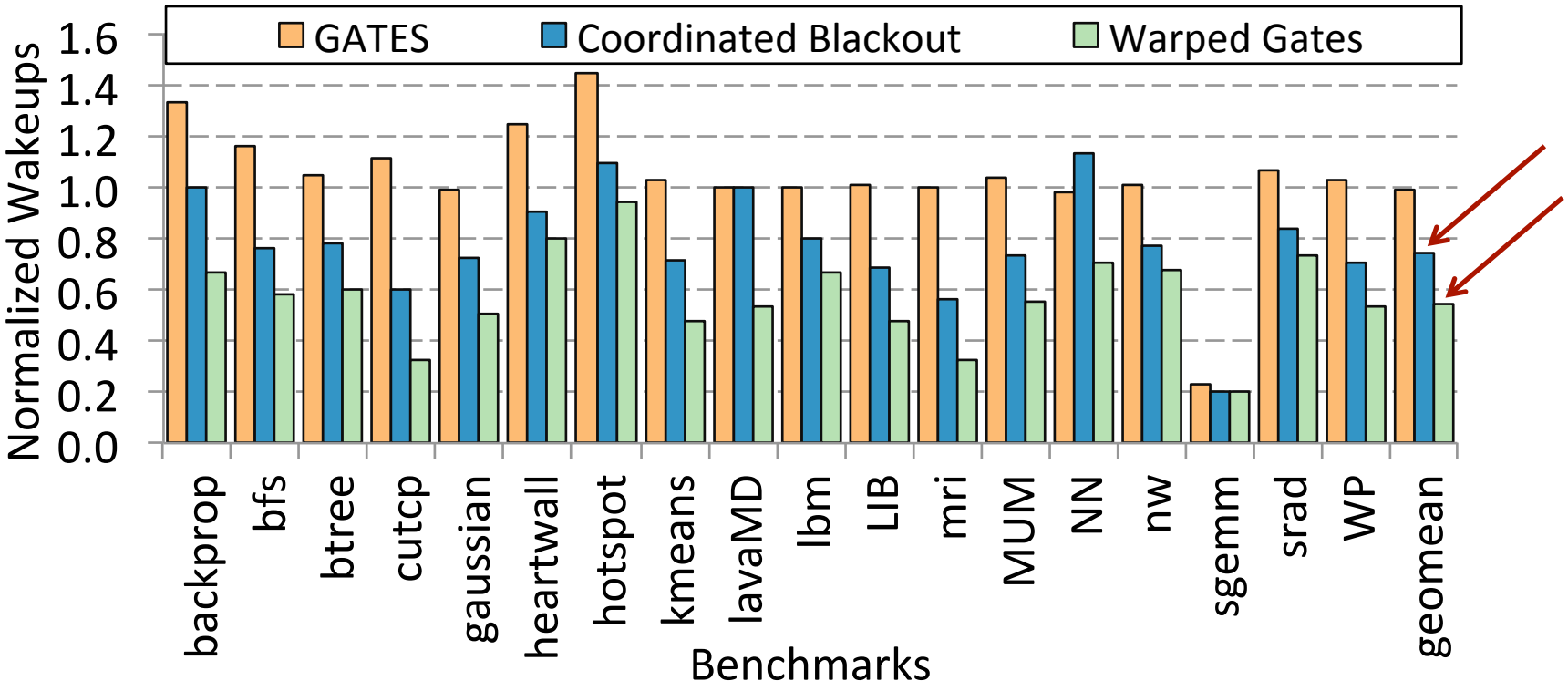


Evaluation

Evaluation Methodology

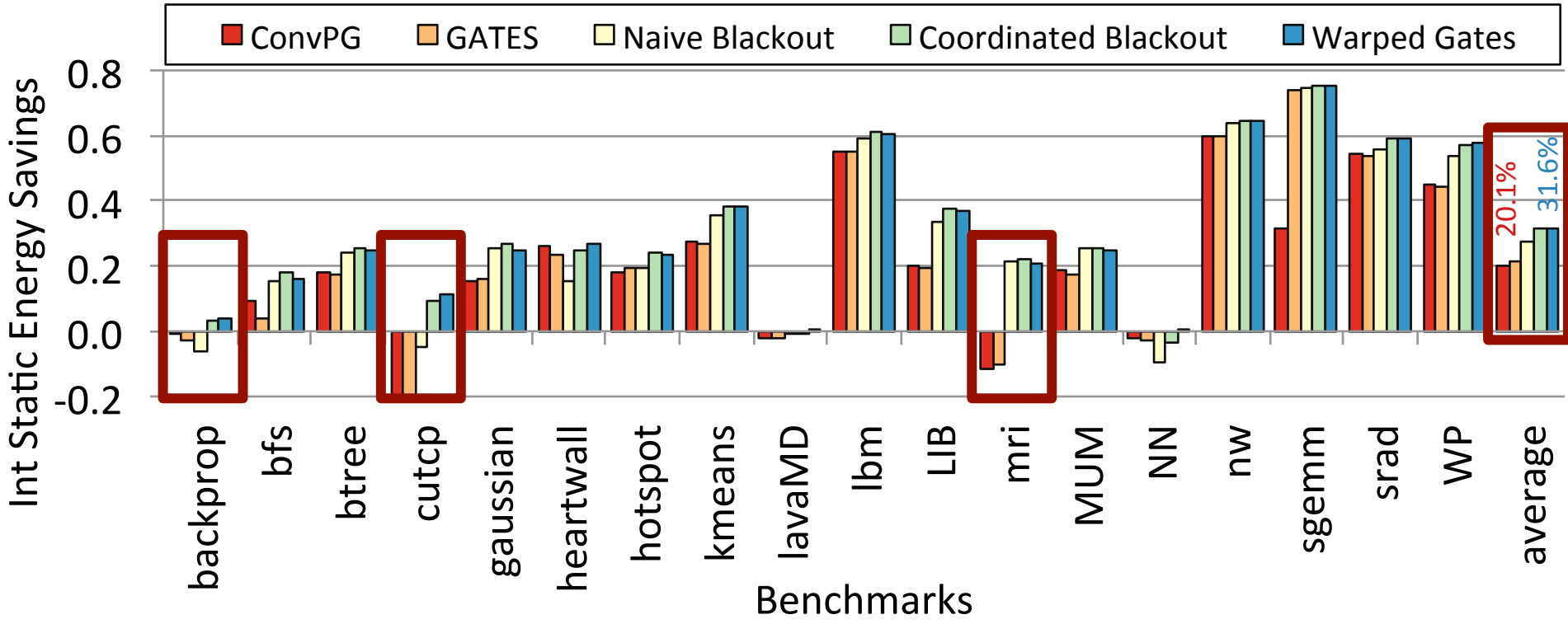
- GPGPU-Sim v3.0.2
  - Nvidia GTX480
- GPUWattch and McPAT for Energy and Area estimation
- 18 Benchmarks from ISPASS, Rodinia, Parboil
- Power Gating parameters
  - Wakeup delay – 3 cycles
  - Breakeven time – 14 cycles
  - Idle detect – 5 cycles

Power Gating Wakeups / Overhead



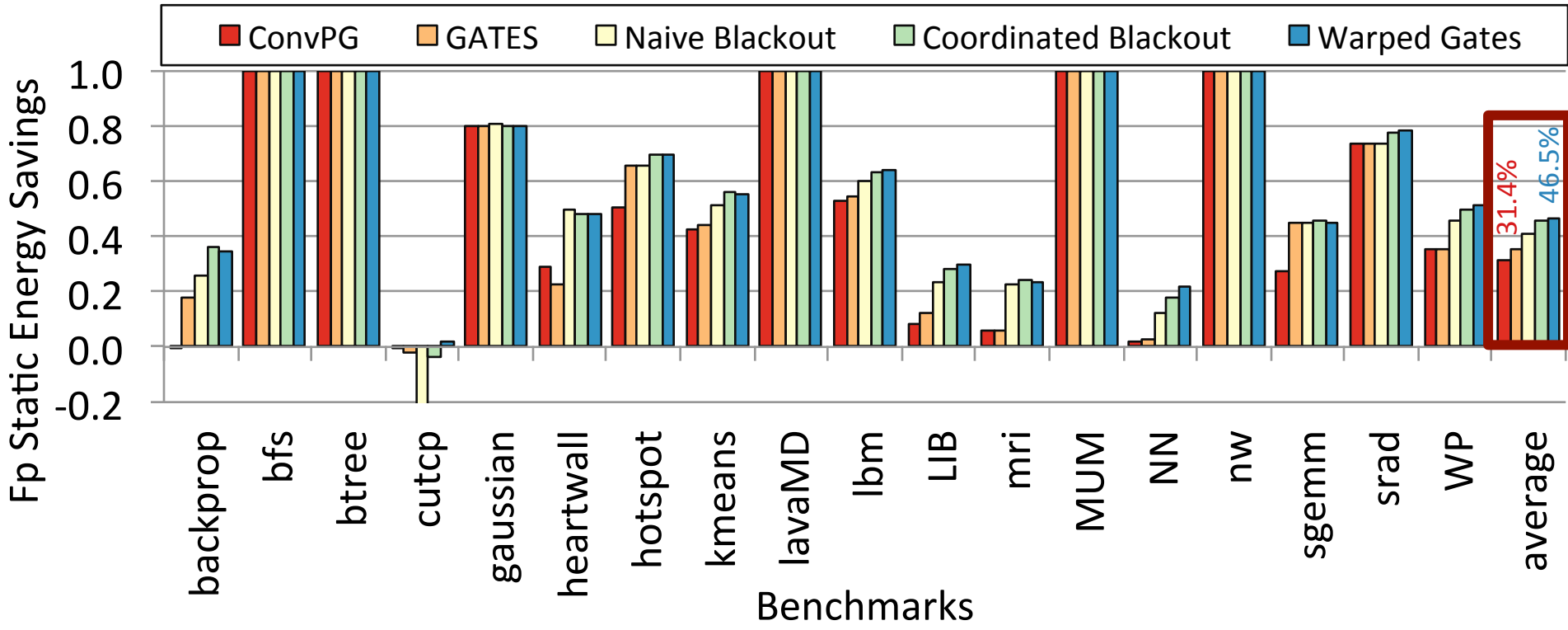
- Coalescing idle periods – fewer, but longer, idle periods
- Blackout reduces PG overhead by 26%
- Warped Gates reduces PG overhead by 46%

Integer Unit Static Energy Savings



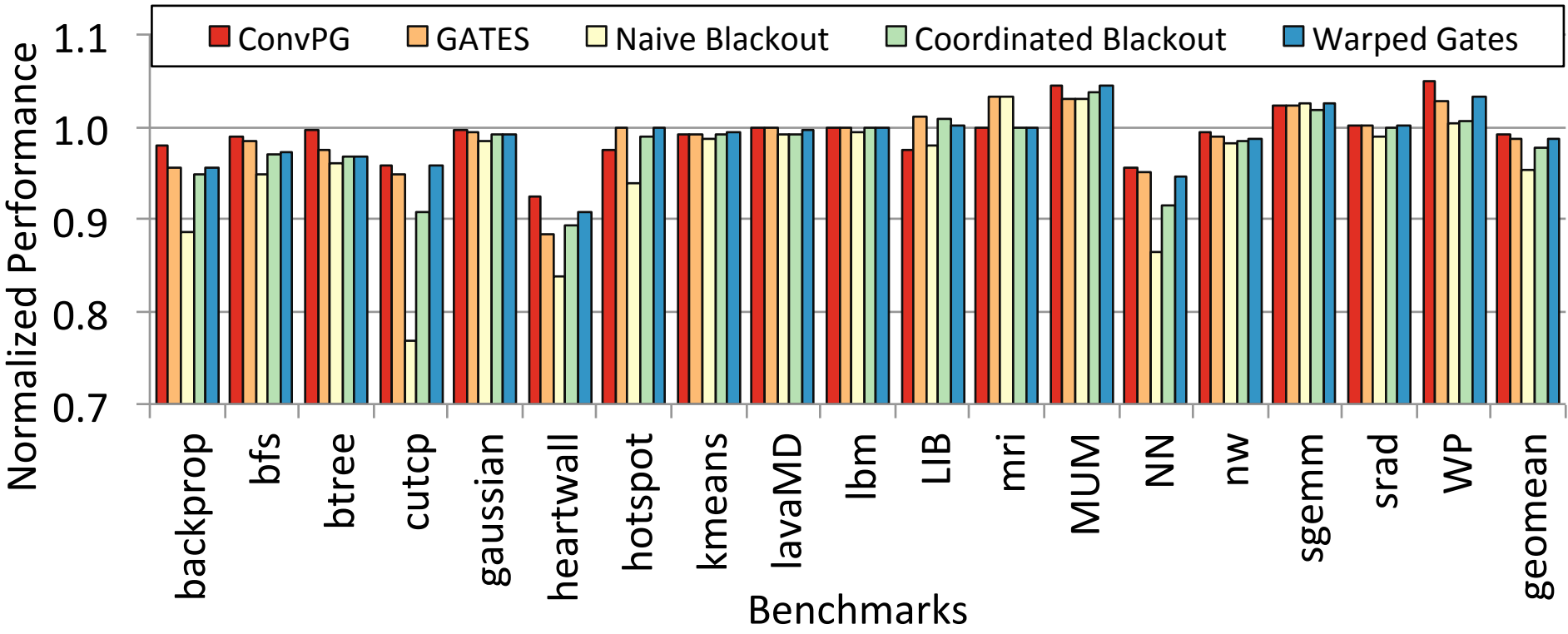
- Blackout/Warped Gates is able to save energy when ConvPG cannot
- Warped Gates saves ~1.5x static energy w.r.t. ConvPG

FP Unit Static Energy Savings



- Warped Gates save ~1.5x static energy w.r.t. ConvPG
  - (Ignores Integer only benchmarks)

Performance Impact



- Naïve Blackout has high overhead due to aggressive PG
- Both ConvPG and Warped Gates has ~1% overhead

Conclusion

- Execution units – largest energy usage in GPGPUs
- Static energy becoming increasingly important
- Traditional microprocessor power gating techniques ineffective in GPGPUs due to short idle periods
- GATES – Scheduler level technique to increase idle periods by coalescing instruction type issues
- Blackout – Forced idleness of execution unit to avoid negative power gating events
- Adaptive Idle Detect – Limit performance impact
- Warped Gates able to save 1.5x more static power than traditional microprocessor techniques, with negligible performance loss

Thank you!

Questions?