Warped Gates: Gating Aware Scheduling and Power Gating for GPGPUs

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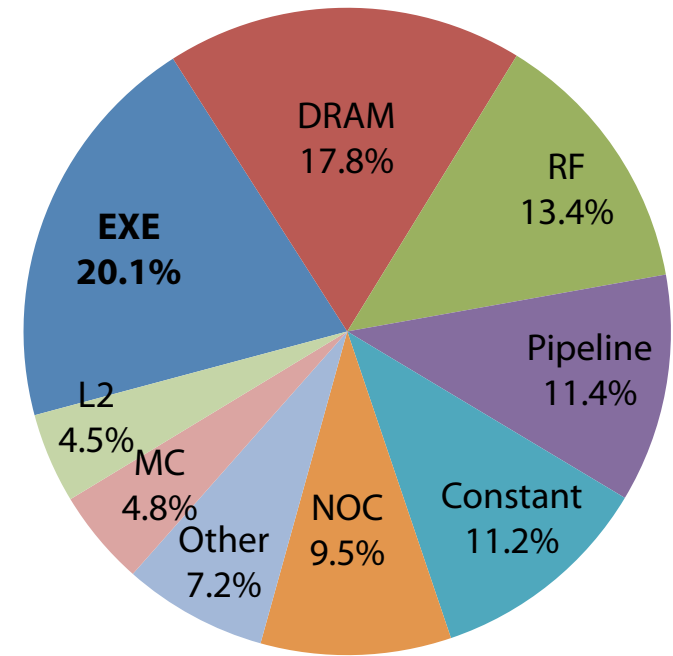*Murali Annavaram*

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]

EXE 20.1%
DRAM 17.8%
RF 13.4%
Pipeline 11.4%
Constant 11.2%
NOC 9.5%
Other 7.2%
MC 4.8%
L2 4.5%

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.

USC
School of Engineering
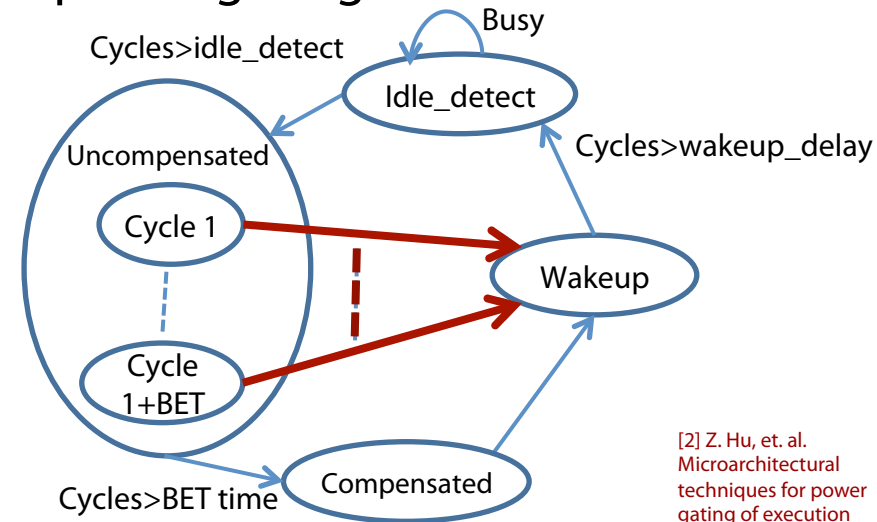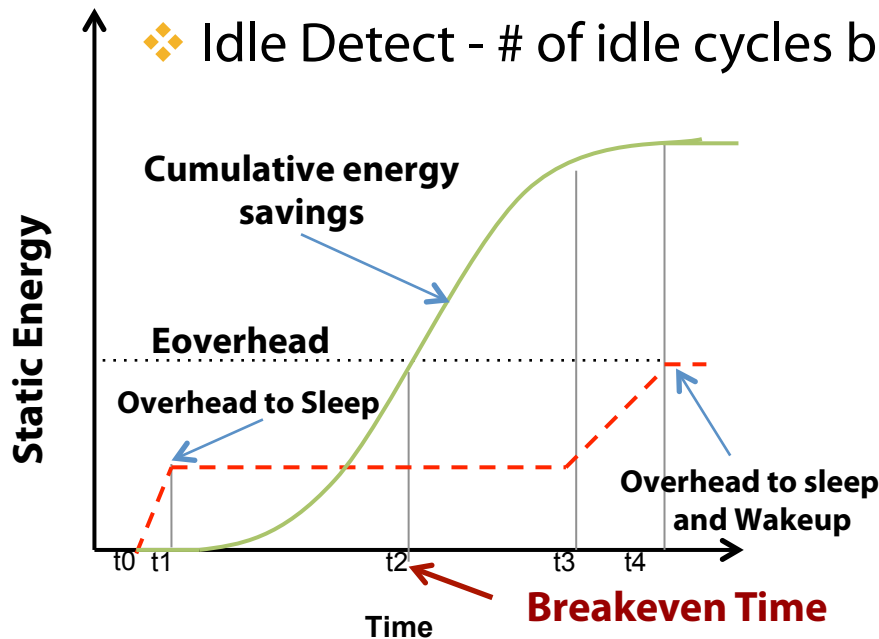
University of Southern California

GPGPU Overview (GTX480)
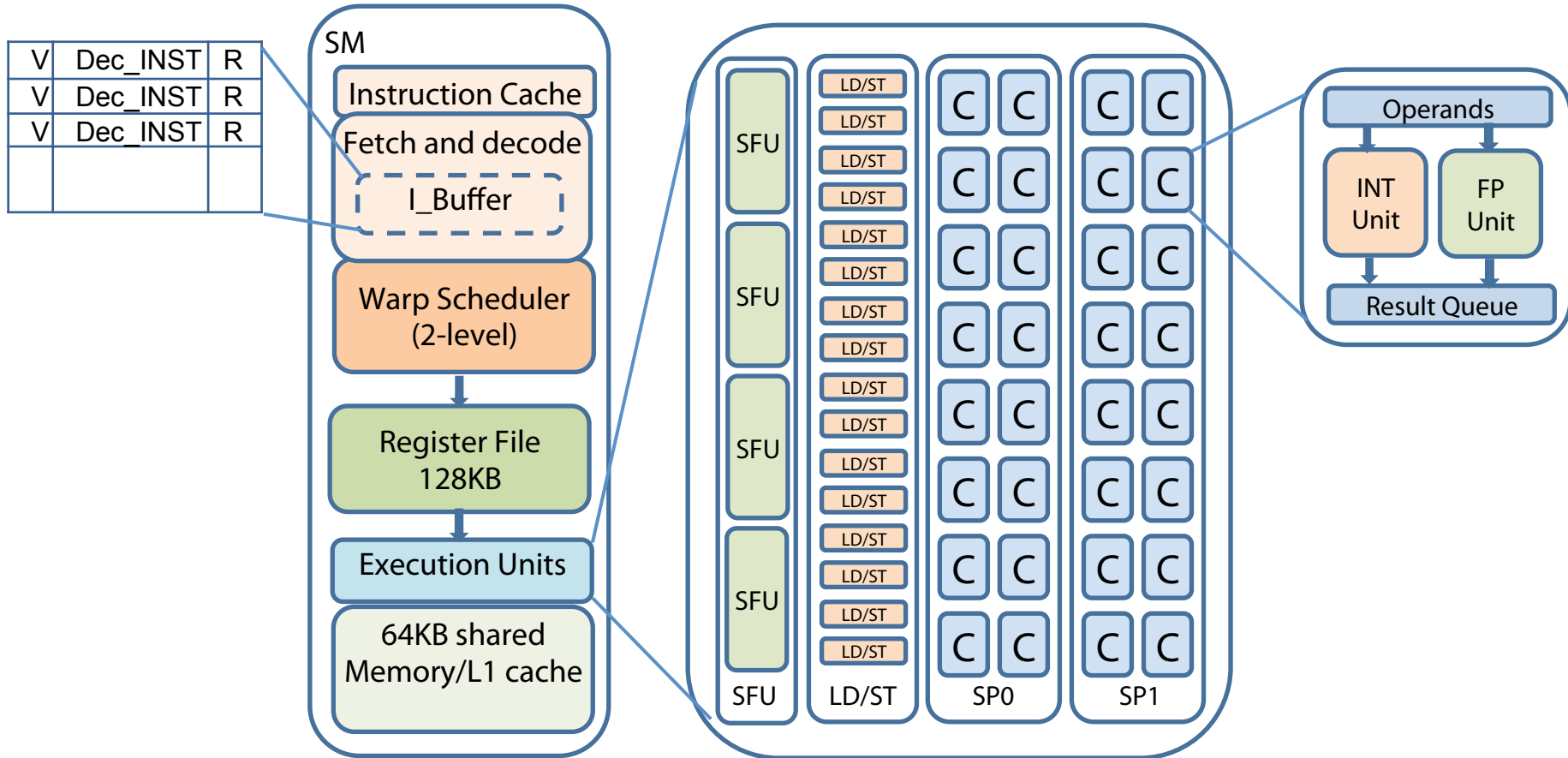
- SP accounts for 98% of Execution Unit Leakage Energy
- Execution units account for 68% of total on chip area

USC
School of Engineering

University of Southern California

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]

Cumulative energy savings

Static Energy

Eoverhead

Overhead to Sleep

Overhead to sleep and Wakeup

t0 t1   t2   t3 t4

Time

Breakeven Time

Busy

Cycles>idle_detect

Idle_detect

Uncompensated

Cycles>wakeup_delay

Cycle 1

Wakeup

Cycle 1+BET

Cycles>BET time

Compensated

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

USC

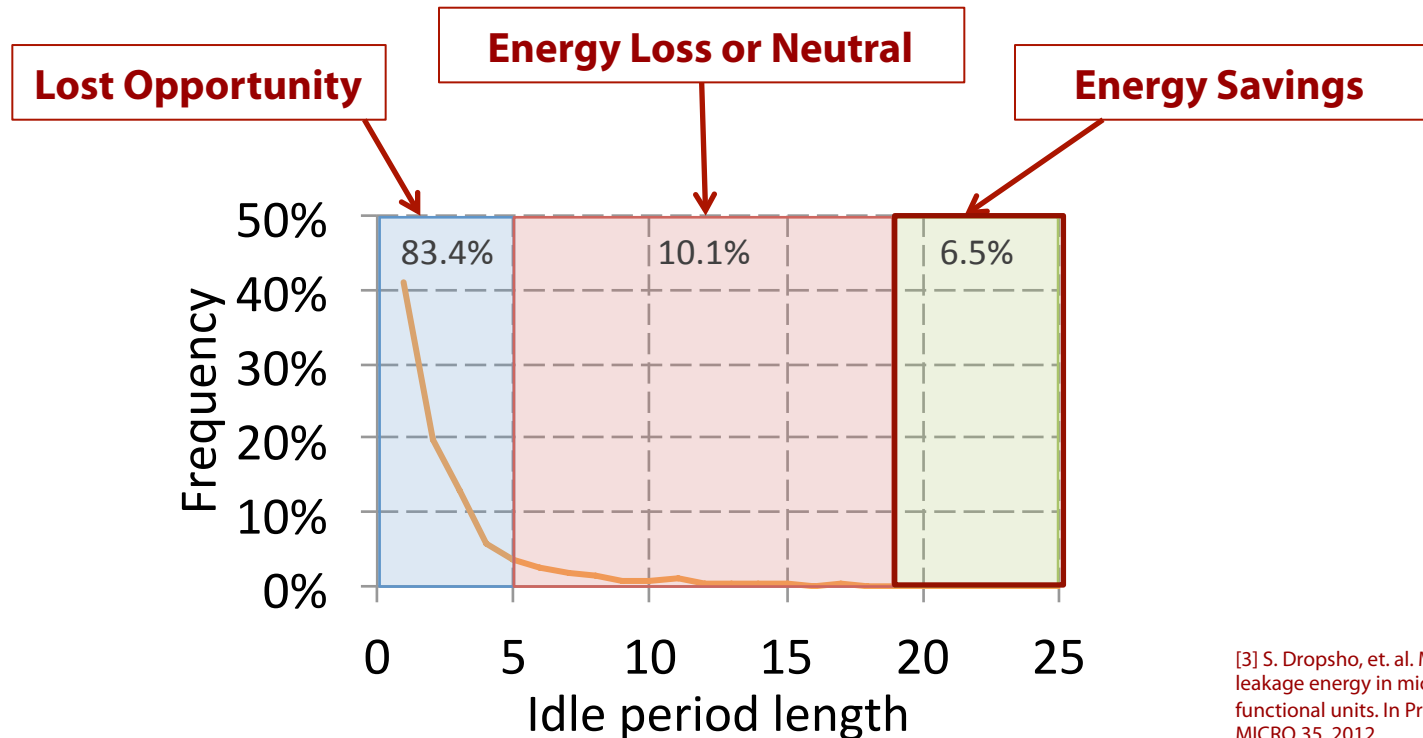School of Engineering

University of Southern California

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

Lost Opportunity

Energy Loss or Neutral

Energy Savings

83.4%   10.1%   6.5%

Frequency

50%
40%
30%
20%
10%
0%

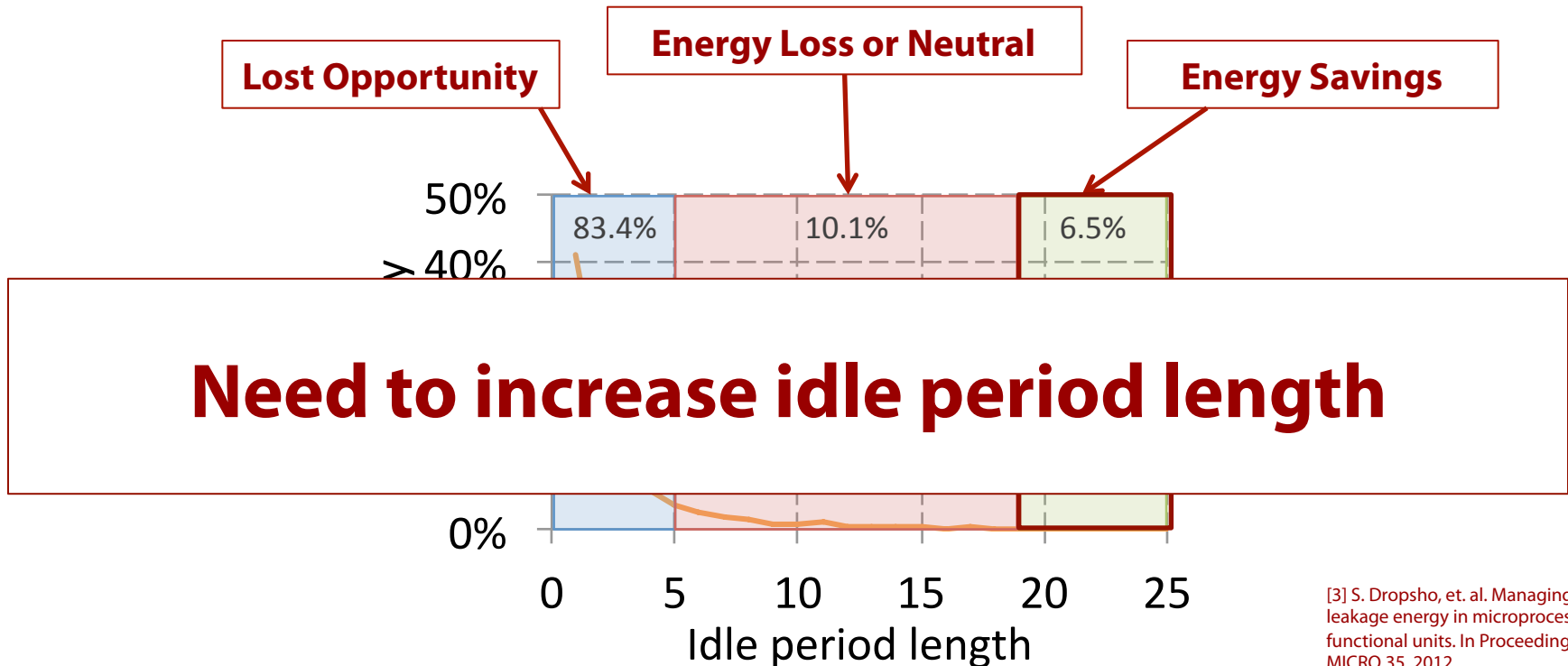0   5   10   15   20   25

Idle period length

[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

Lost Opportunity

Energy Loss or Neutral

Energy Savings

50%

> 40%

| 83.4% | 10.1% | 6.5% |

Need to increase idle period length

0%

0   5   10   15   20   25
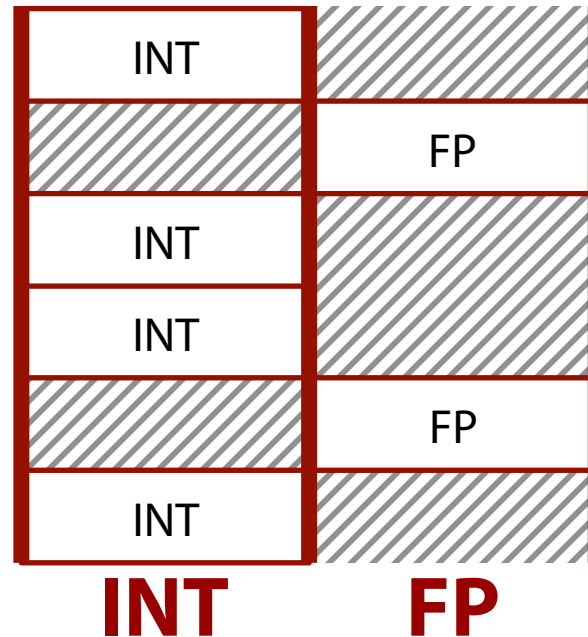
Idle period length

[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

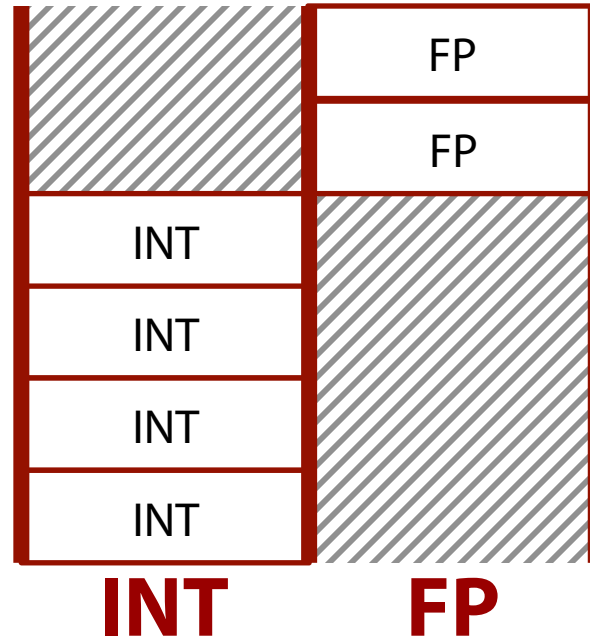
Idle Periods

GATES:

Gating Aware Two-level Scheduler

Issue warps based on execution unit resource type

Gating Aware Two-level Scheduler (GATES)

INT	INT	INT	INT	FP	FP

Ready Warps

| Idle periods are coalesced

INT   **FP**
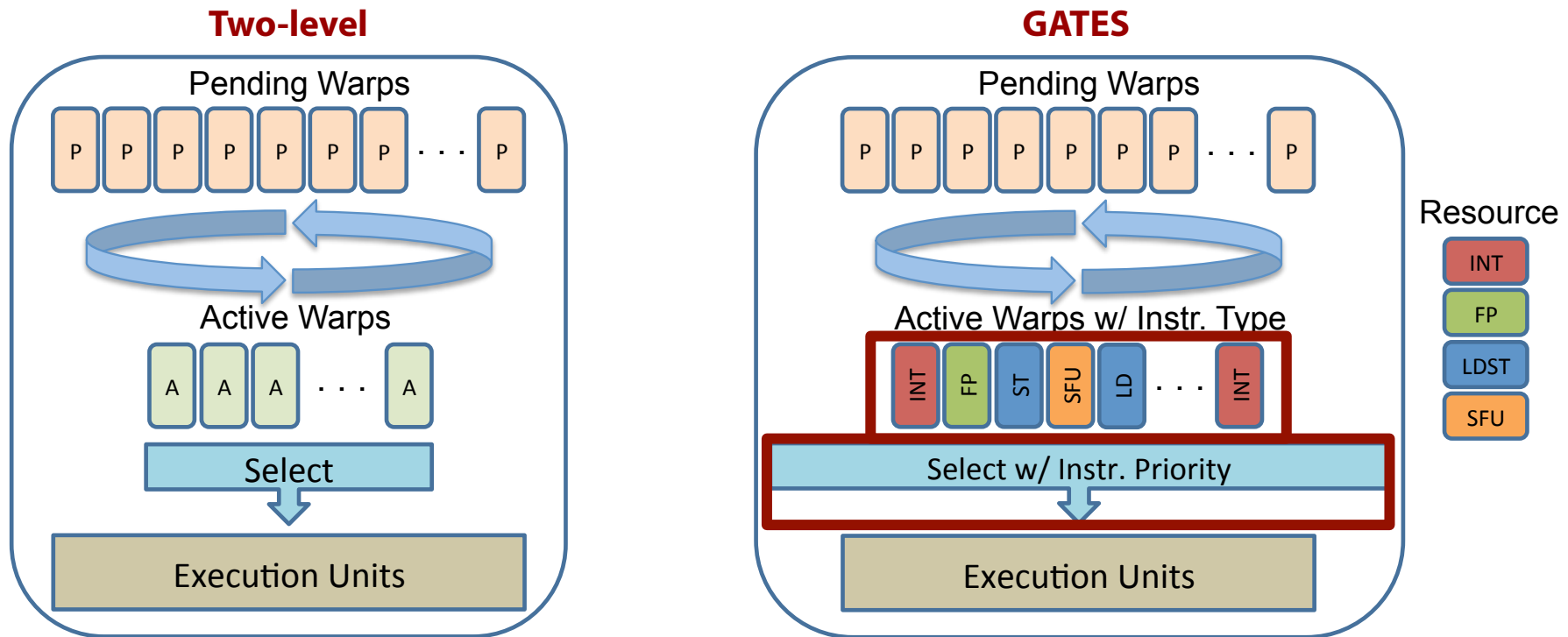
Idle Period

USC
School of Engineering

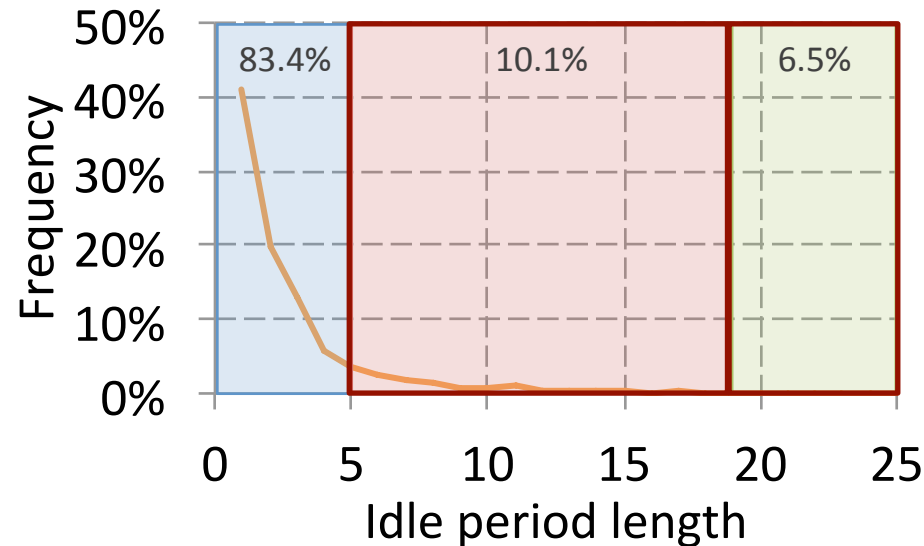University of Southern California

Gating Aware Two-level Scheduler (GATES)

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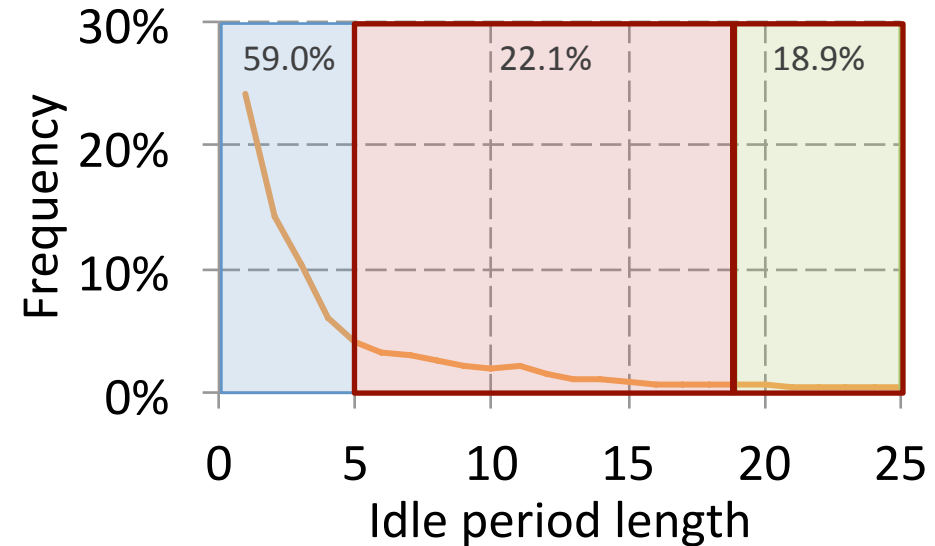 - ❖ Switch highest priority when it out of ready warps

Two-level

Pending Warps

P P P P P P P . . . P

Active Warps

A A A . . . A

Select

Execution Units

GATES

Pending Warps

P P P P P P P . . . P

Active Warps w/ Instr. Type

INT FP ST SFU LD . . . INT

Select w/ Instr. Priority

Execution Units

Resource

INT

FP

LDST

SFU

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

USC
School of Engineering

University of Southern California

Blackout Power Gating
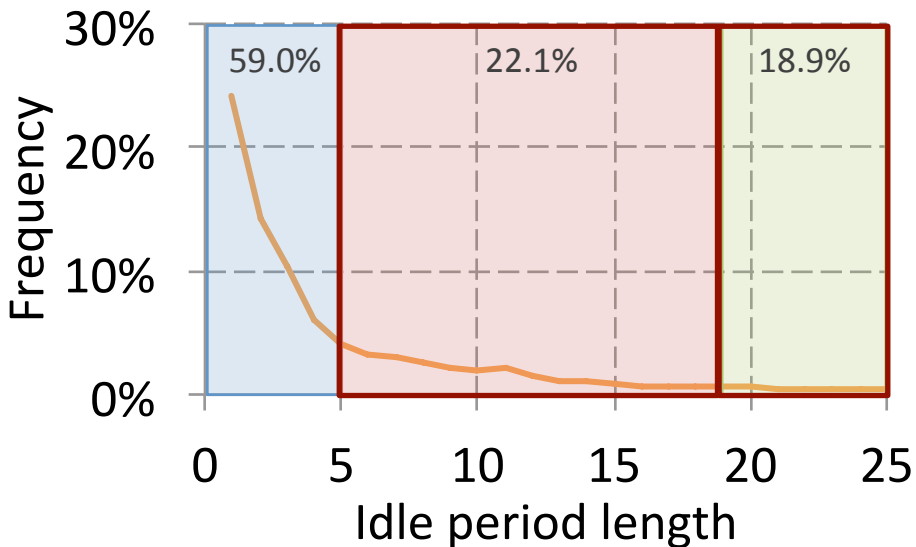
| Forced idleness of execution units to meet BET

Blackout Power Gating
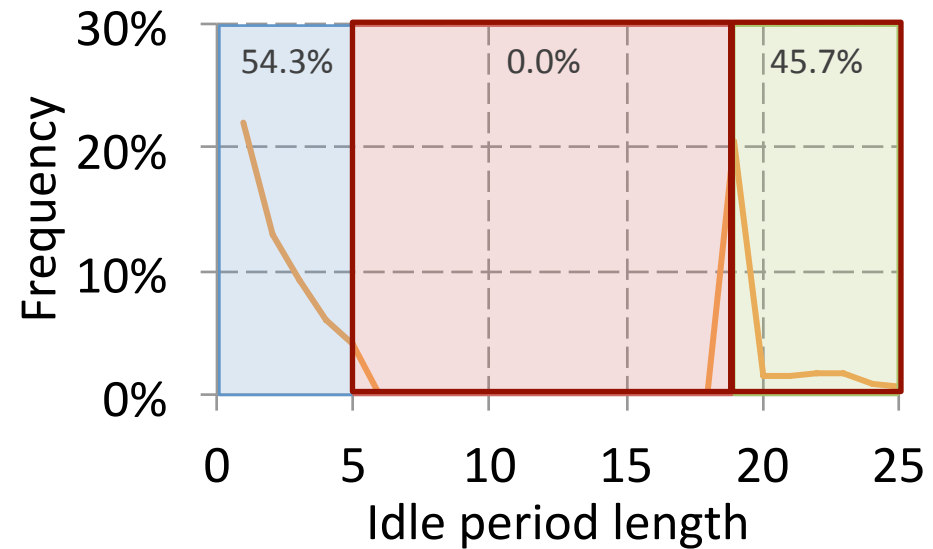
| Force idleness until break even time has passed

  ❖ 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
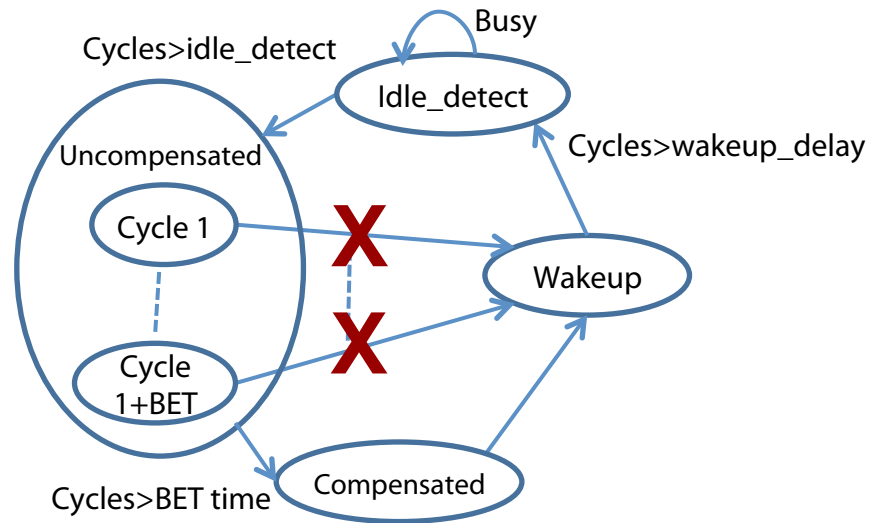
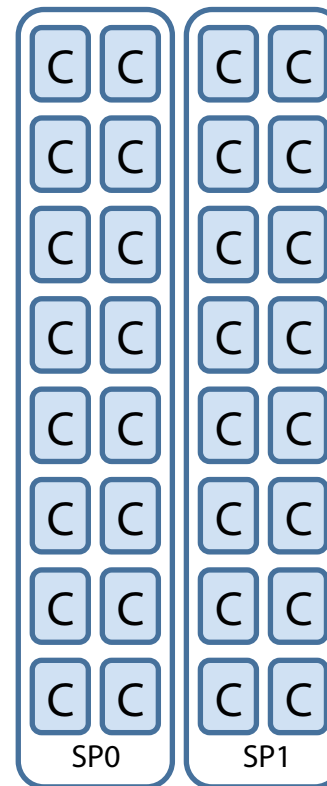# Blockout Power Gating

GATES
GATES + Blockout

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

USC
School of Engineering

University of Southern California

Blackout Policies

Naïve Blackout

❖ GATES and Blackout is independent

Warp Scheduler
(GATES)

C C	C C
C C	C C
C C	C C
C C	C C
C C	C C
C C	C C
C C	C C
C C	C C
SP0	SP1

Can lead to overaggressive power gating

USC
School of Engineering

University of Southern California

Blackout Policies

Naïve Blackout

❖ GATES and Blackout is independent

Idle Detect

Warp Scheduler
(GATES)

C	C	C	C
C	C	C	C
C	C	C	C
C	C	C	C
C	C	C	C
C	C	C	C
C	C	C	C
C	C	C	C
SP0		SP1	

Can lead to overaggressive power gating

Blackout Policies

Naïve Blackout

❖ GATES and Blackout is independent

Idle Detect
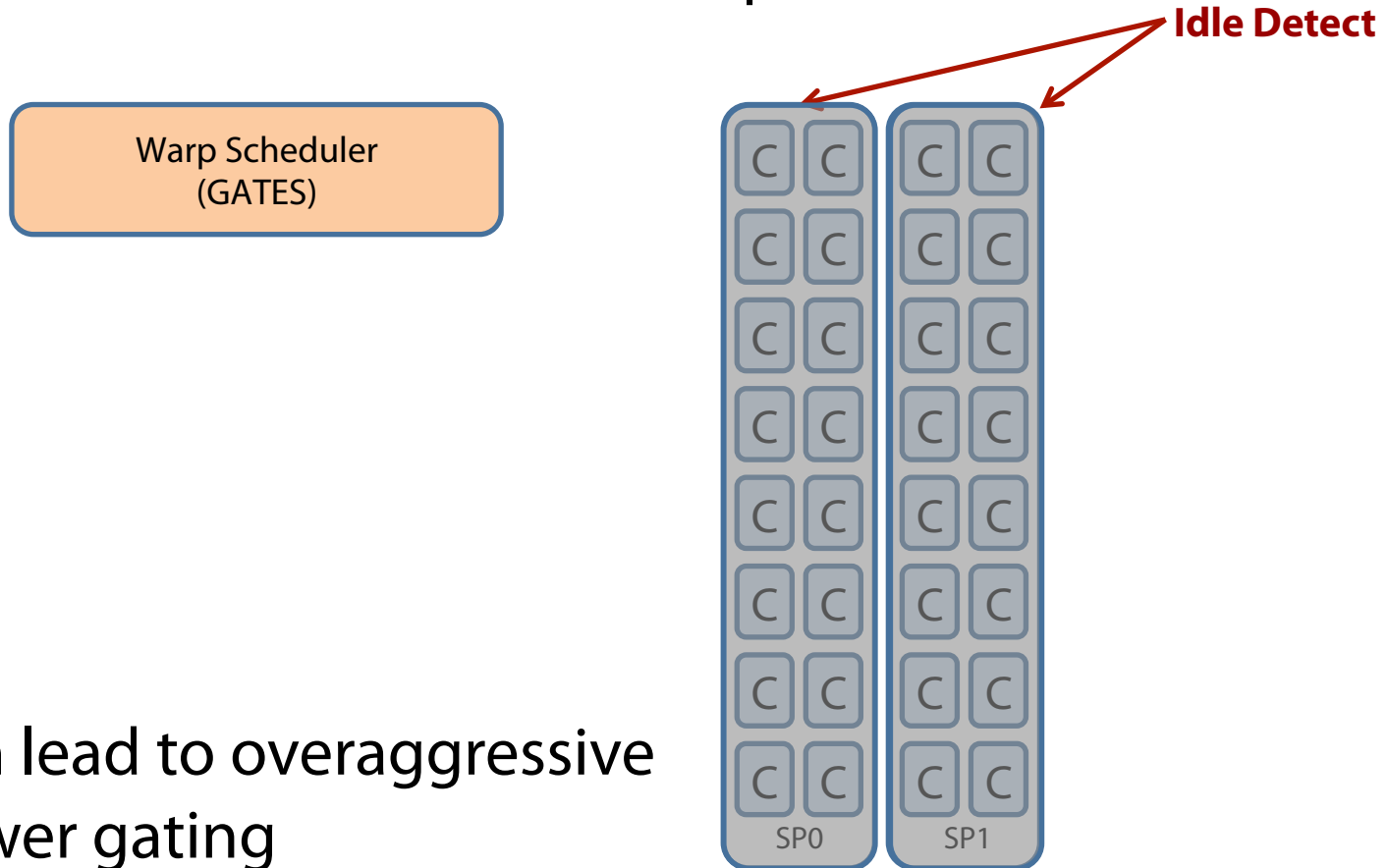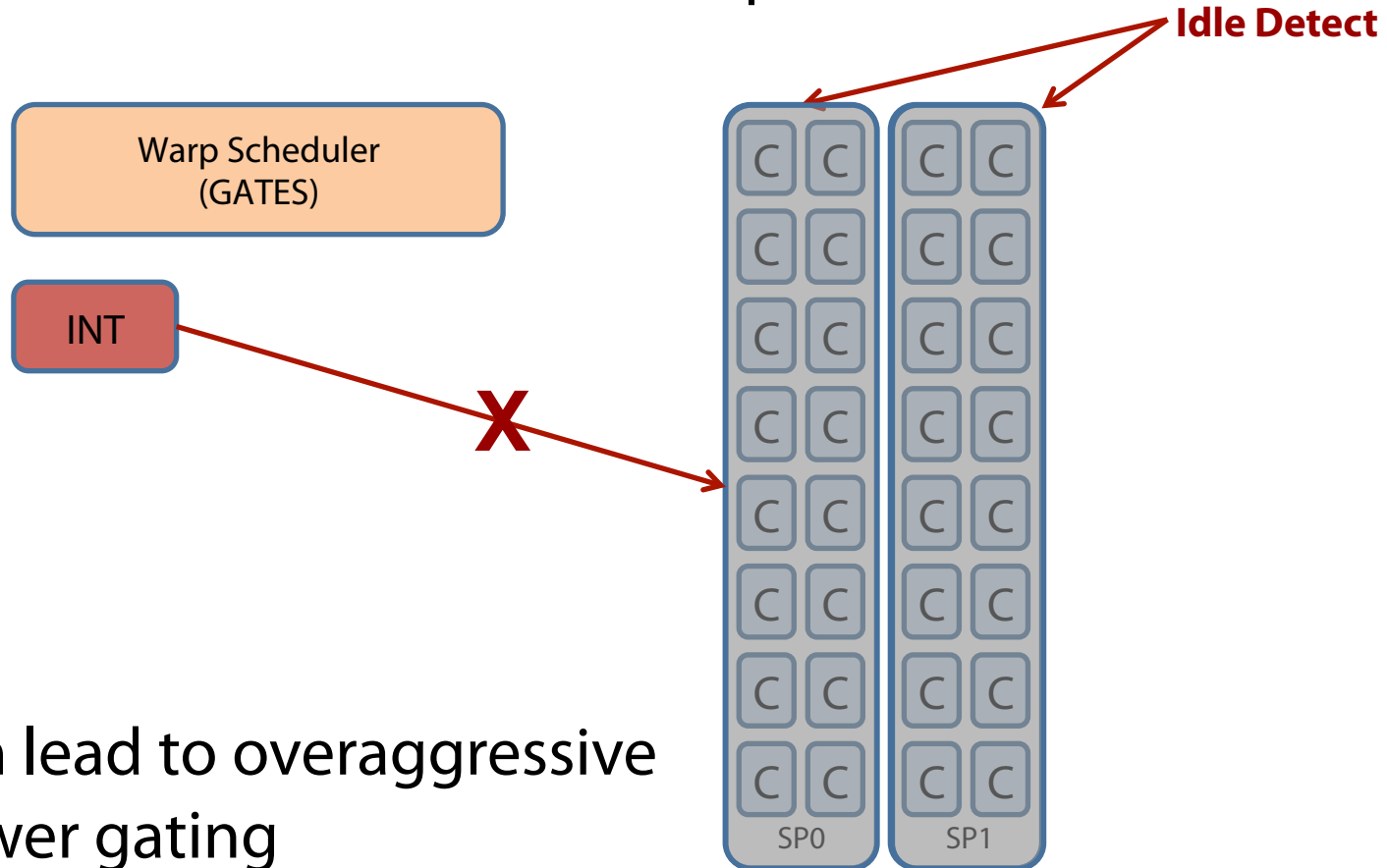
Warp Scheduler
(GATES)

C C   C C

C C   C C

C C   C C

C C   C C

C C   C C

C C   C C

C C   C C

C C   C C

SP0   SP1

Can lead to overaggressive power gating

USC
School of Engineering

University of Southern California

Blockout Policies

Naïve Blockout

❖ GATES and Blackout is independent

Idle Detect

Warp Scheduler
(GATES)

INT

SP0   SP1

Can lead to overaggressive power gating

Blackout Policies

Coordinated Blackout

Warp Scheduler (GATES)

Dynamic priority switching is Blackout aware

Idle Detect

C	C		C	C
C	C		C	C
C	C		C	C
C	C		C	C
C	C		C	C
C	C		C	C
C	C		C	C
C	C		C	C

SP0   SP1

Blackout Policies

| Coordinated Blackout

Warp Scheduler
(GATES)

Dynamic priority switching
is Blackout aware

**Active Warp
Count Based**

| PG only when active warps
count = 0

SP0   SP1

Blackout Policies

Coordinated Blackout

Warp Scheduler
(GATES)

Dynamic priority switching
is Blackout aware

INT ✓

Active Warp
Count Based

SP0

SP1

PG only when active warps
count = 0

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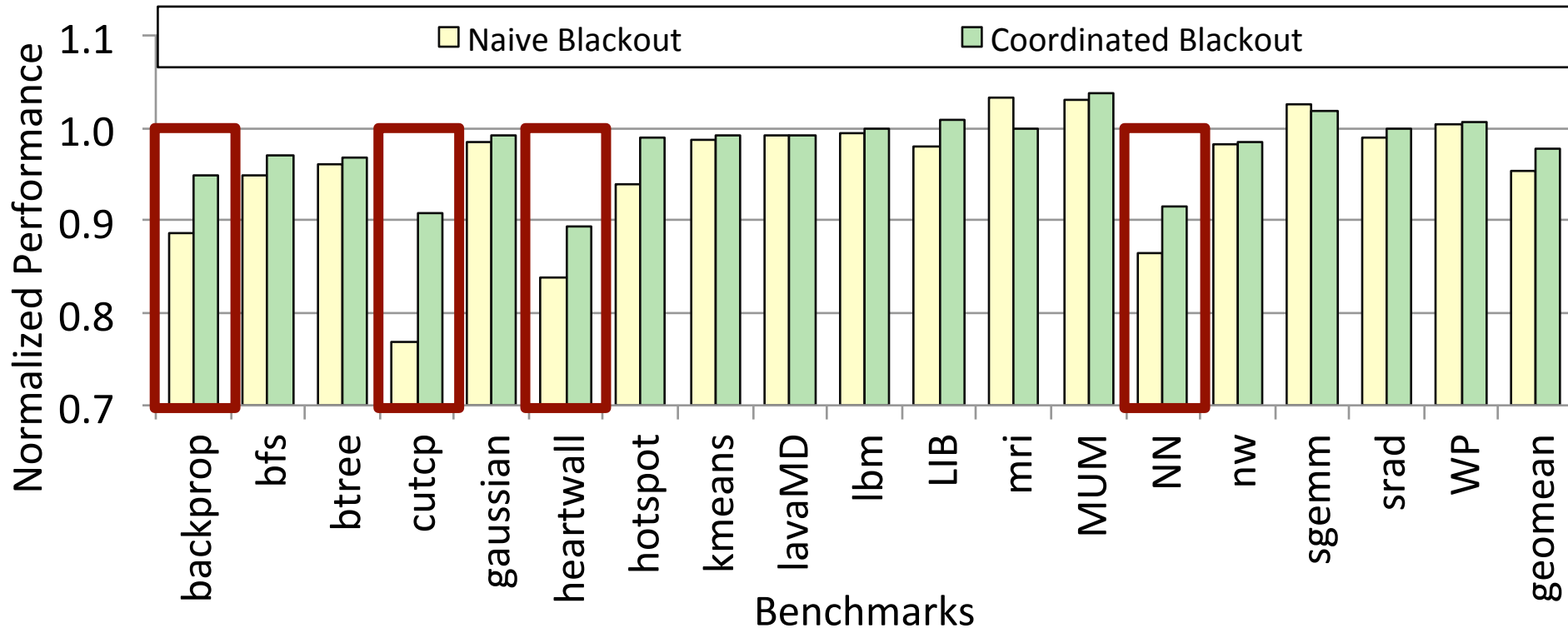
- Some benchmarks still show poor performance
  - ❖ Not enough active warps to hide forced idleness
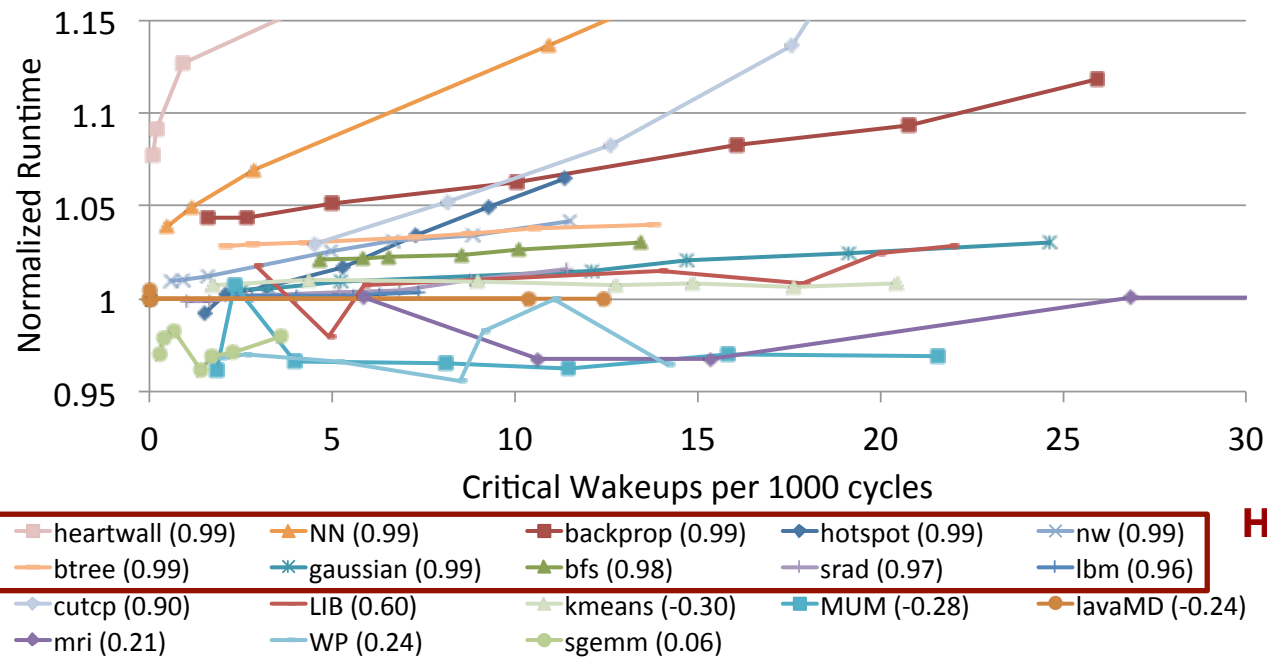- Goal is as close to 0% overhead as possible

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
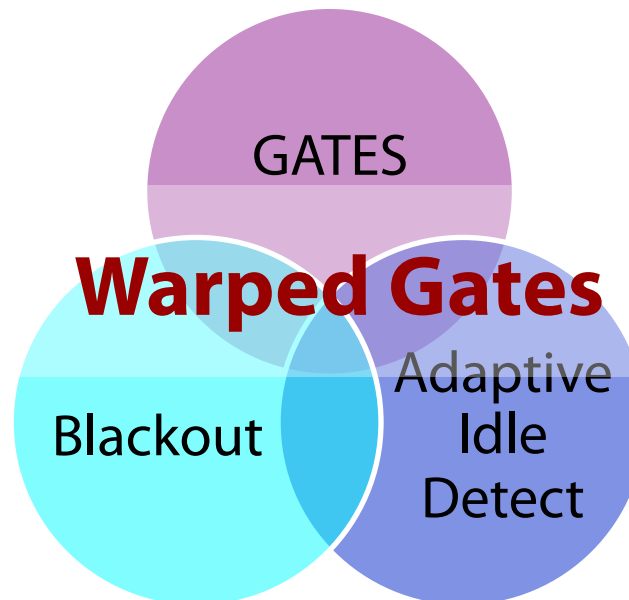

High Correlation vs Runtime

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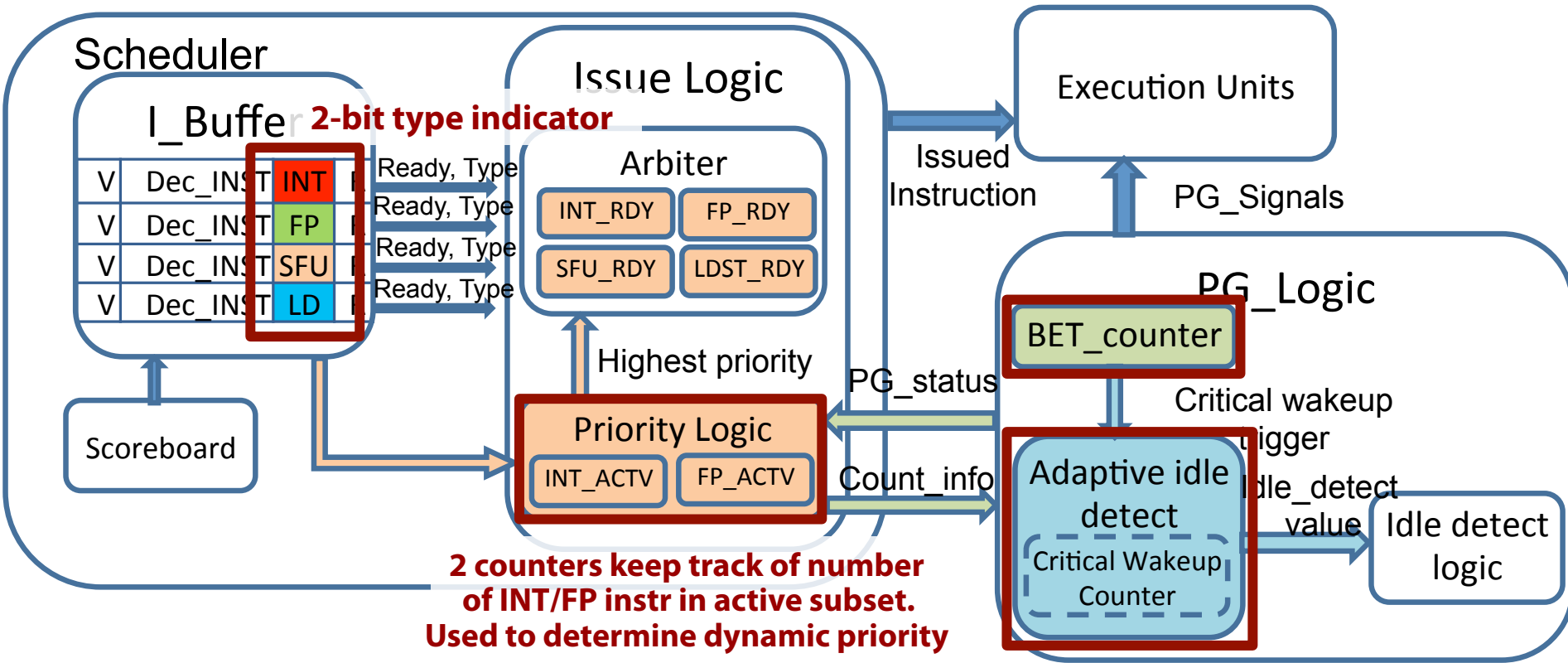
| Independent idle detect values for INT and FP pipelines
| Break execution time into epoch (1000 cycles)
| If critical wakeup > threshold, idleDetect++
| Conservatively decrement idleDetect every 4 epochs
| Bound idle detect between 5 – 10 cycles

GATES

Warped Gates

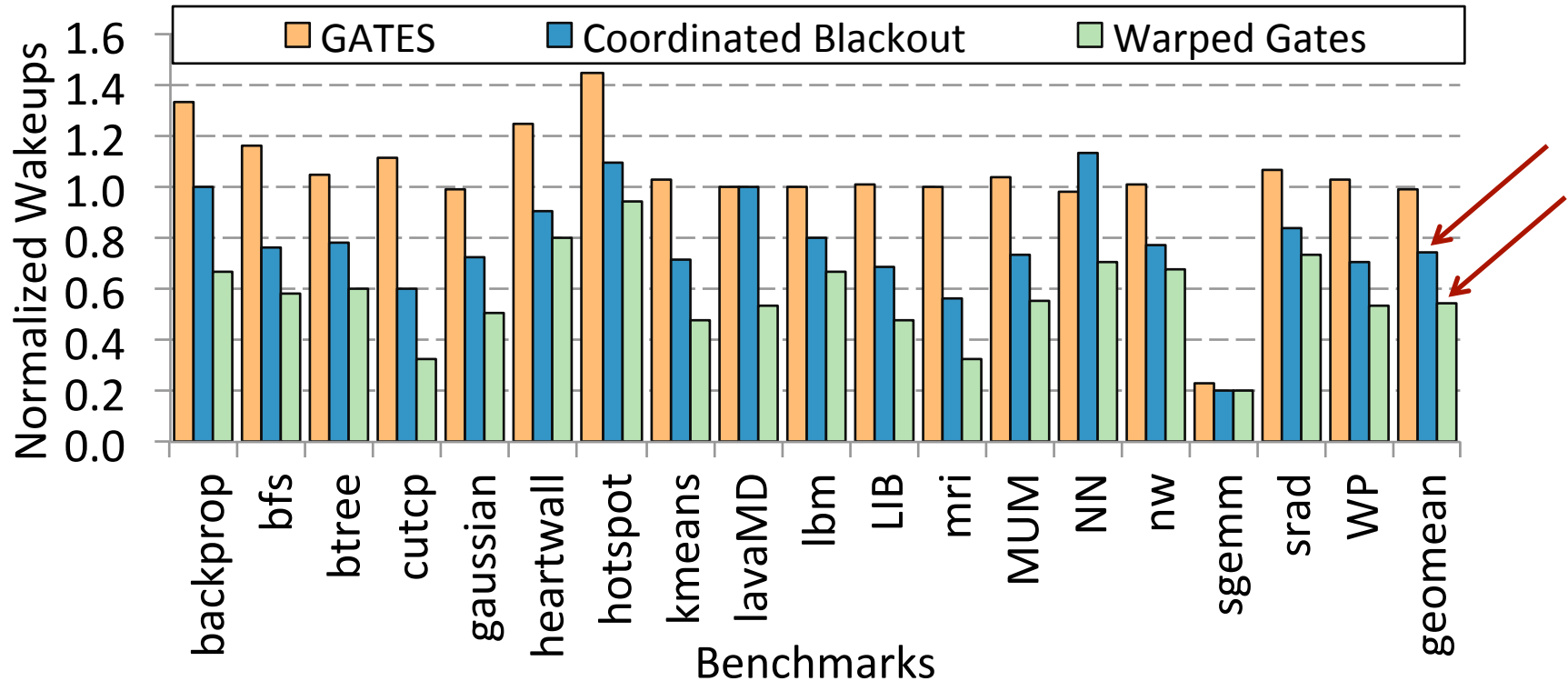Blackout

Adaptive Idle Detect

Architectural Support

Evaluation

Evaluation Methodology

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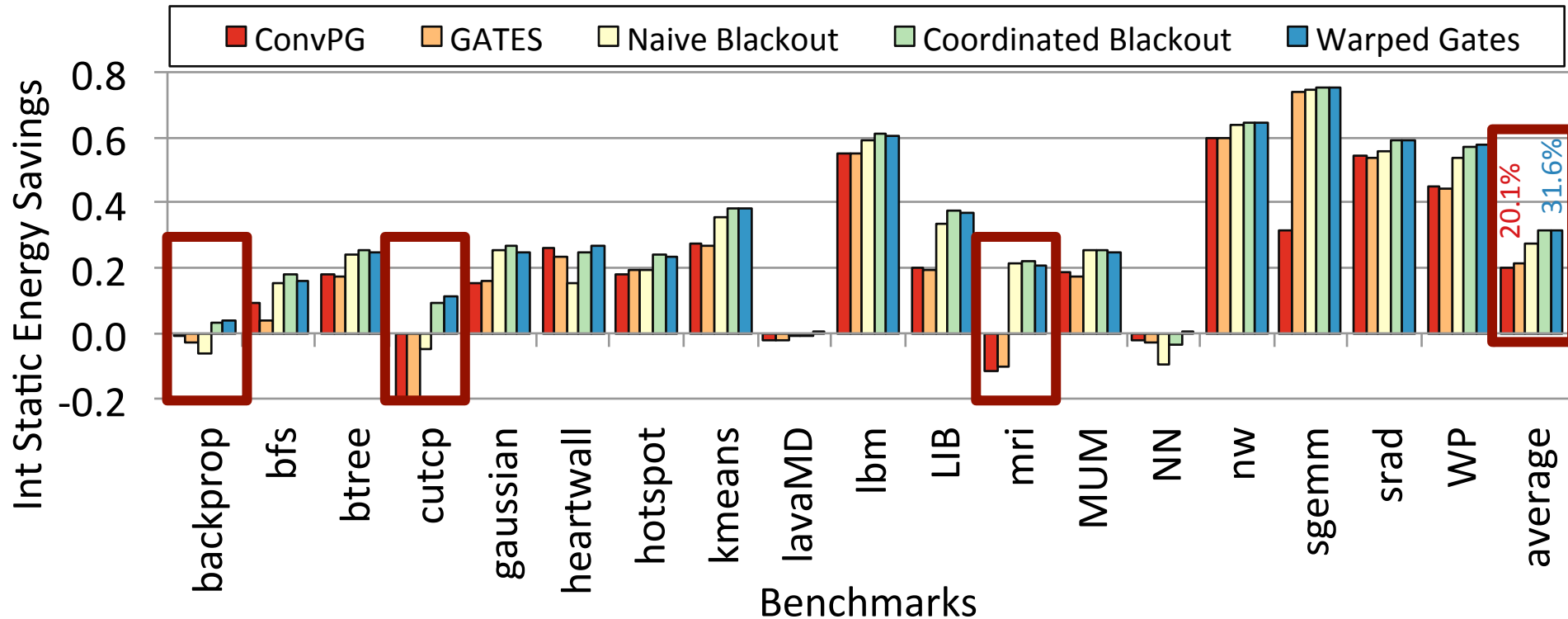- GPUWattch and McPAT for Energy and Area estimation
- 18 Benchmarks from ISPASS, Rodinia, Parboil
- Power Gating parameters
  - ❖ Wakeup delay – 3 cycles
  - ❖ Breakeven time – 14 cycles
  - ❖ Idle detect – 5 cycles

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

Legend: ConvPG, GATES, Naive Blackout, Coordinated Blackout, Warped Gates

Benchmarks: backprop, bfs, btree, cutcp, gaussian, heartwall, hotspot, kmeans, lavaMD, lbm, LIB, mri, MUM, NN, nw, sgemm, srad, WP, average

Average values highlighted: 20.1%, 31.6%

- 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

Legend: ■ ConvPG ■ GATES □ Naive Blackout ■ Coordinated Blackout ■ Warped Gates

Y-axis: Fp Static Energy Savings (1.0, 0.8, 0.6, 0.4, 0.2, 0.0, -0.2)

X-axis (Benchmarks): backprop, bfs, btree, cutcp, gaussian, heartwall, hotspot, kmeans, lavaMD, lbm, LIB, mri, MUM, NN, nw, sgemm, srad, WP, average

Callout values at average: 31.4%, 46.5%
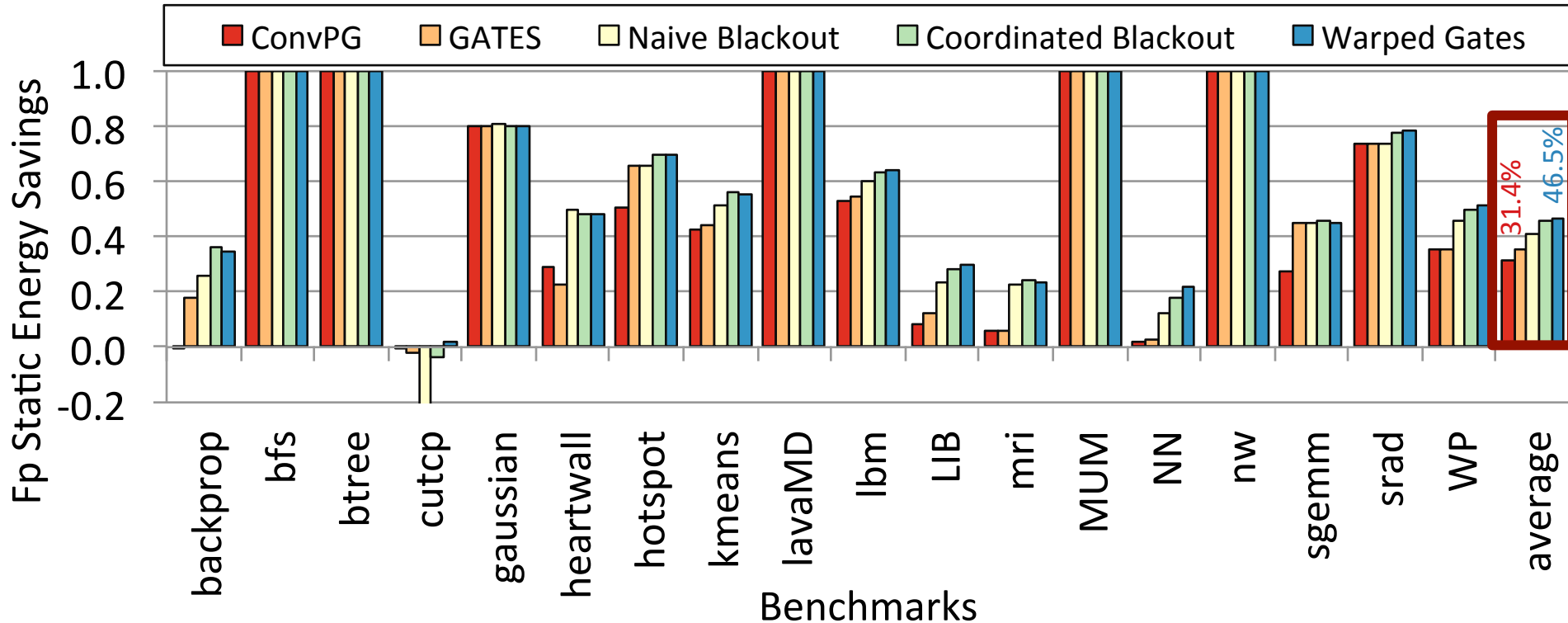
| Warped Gates save ~1.5x static energy w.r.t. ConvPG

❖ (Ignores Integer only benchmarks)

Performance Impact

Normalized Performance vs. Benchmarks

Legend: ConvPG, GATES, Naïve Blackout, Coordinated Blackout, Warped Gates

Benchmarks: backprop, bfs, btree, cutcp, gaussian, heartwall, hotspot, kmeans, lavaMD, lbm, LIB, mri, MUM, NN, nw, sgemm, srad, WP, geomean
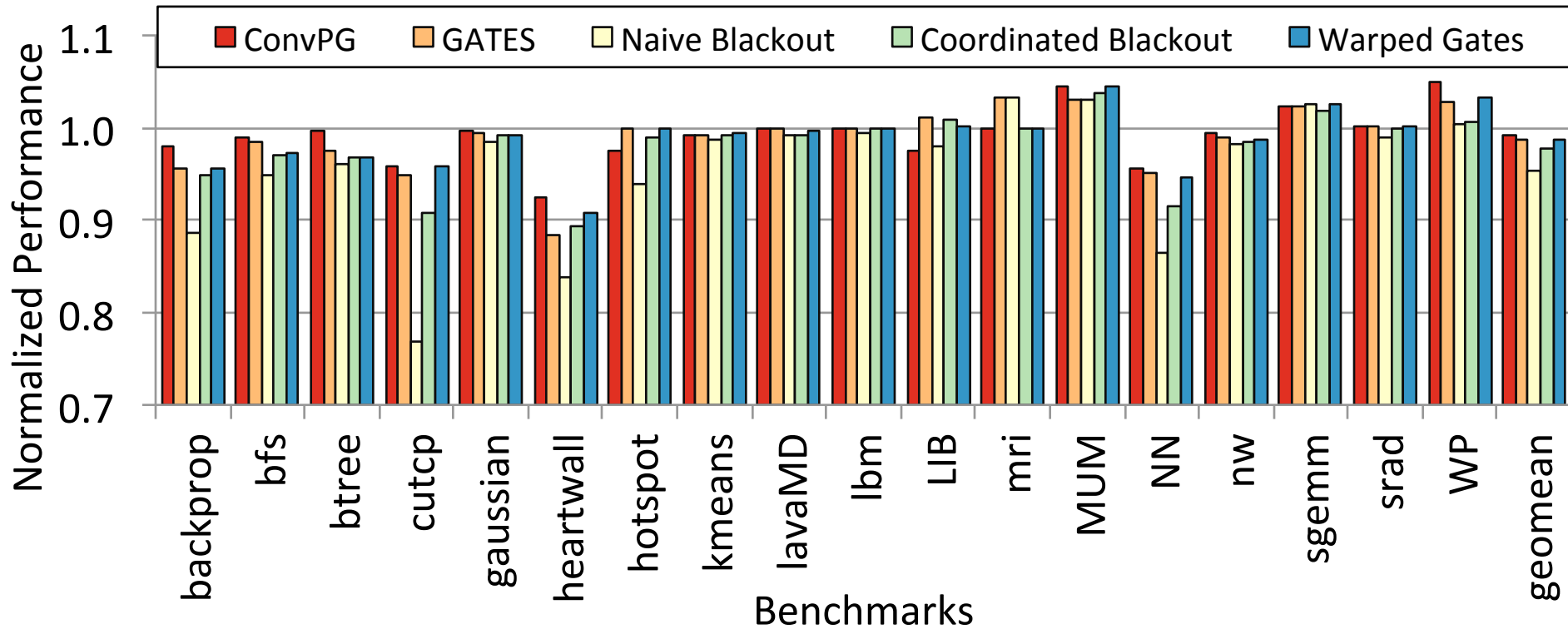
- Naïve Blackout has high overhead due to aggressive PG
- Both ConvPG and Warped Gates has ~1% overhead

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?