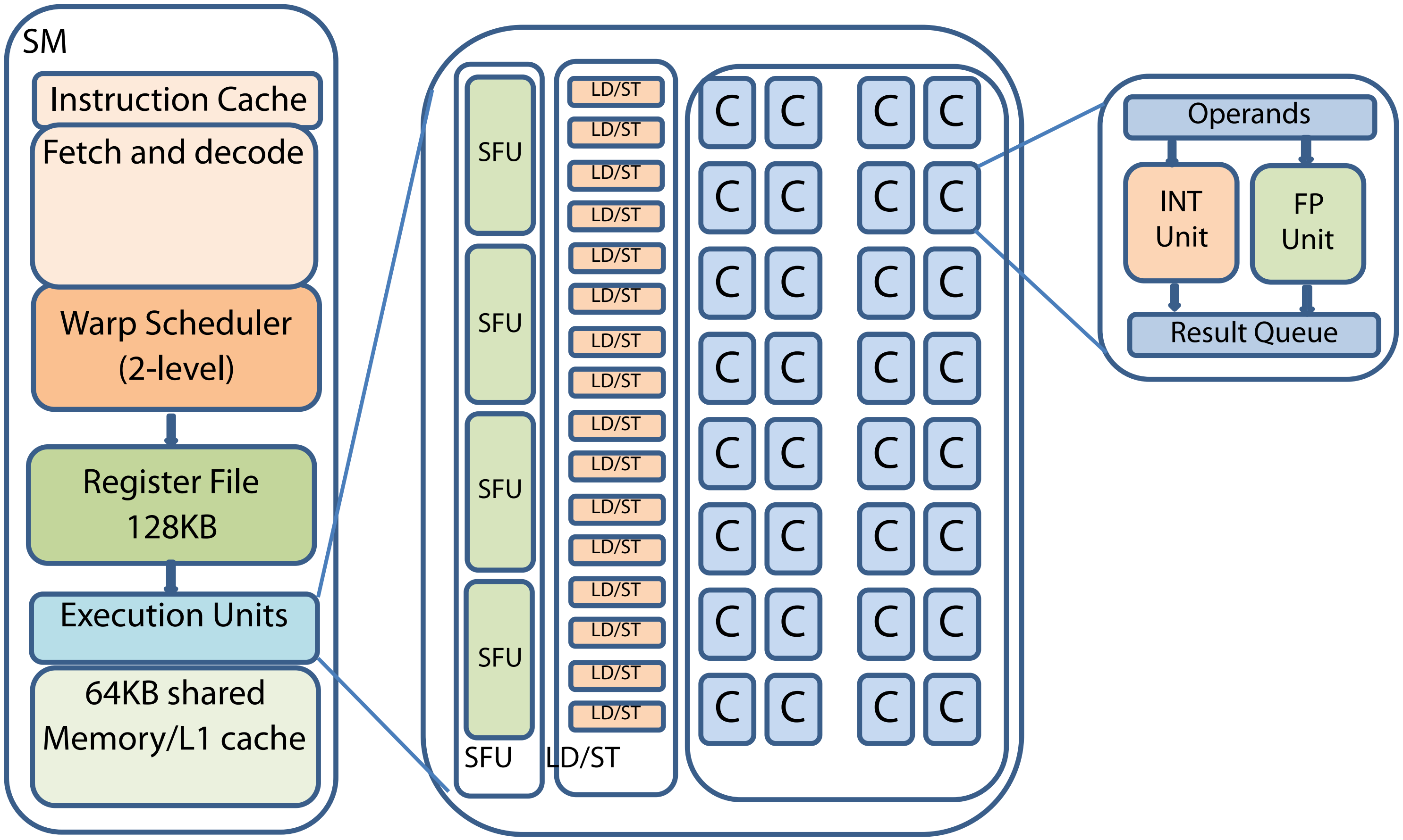
Origami: Folding Warps for Energy Efficient GPUs

Mohammad Abdel-Majeed, Daniel Wong†, Justin Huang‡ and Murali Annavaram**
** University of Southern California*
†University of California, Riverside
‡Stanford University

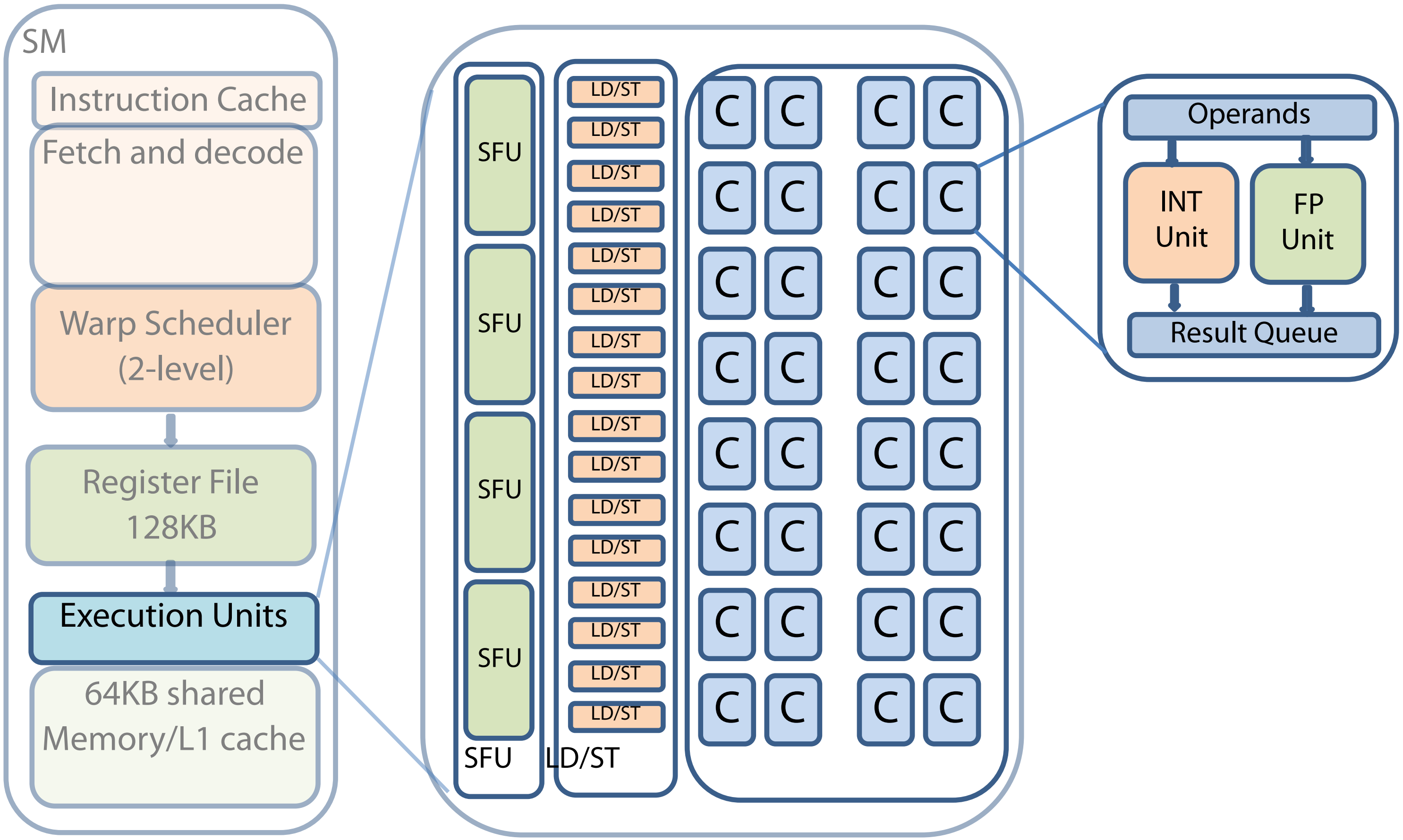
Outline

- GPU overview
- Motivation and related work
- Warp Folding
- Origami Scheduler
- Evaluation

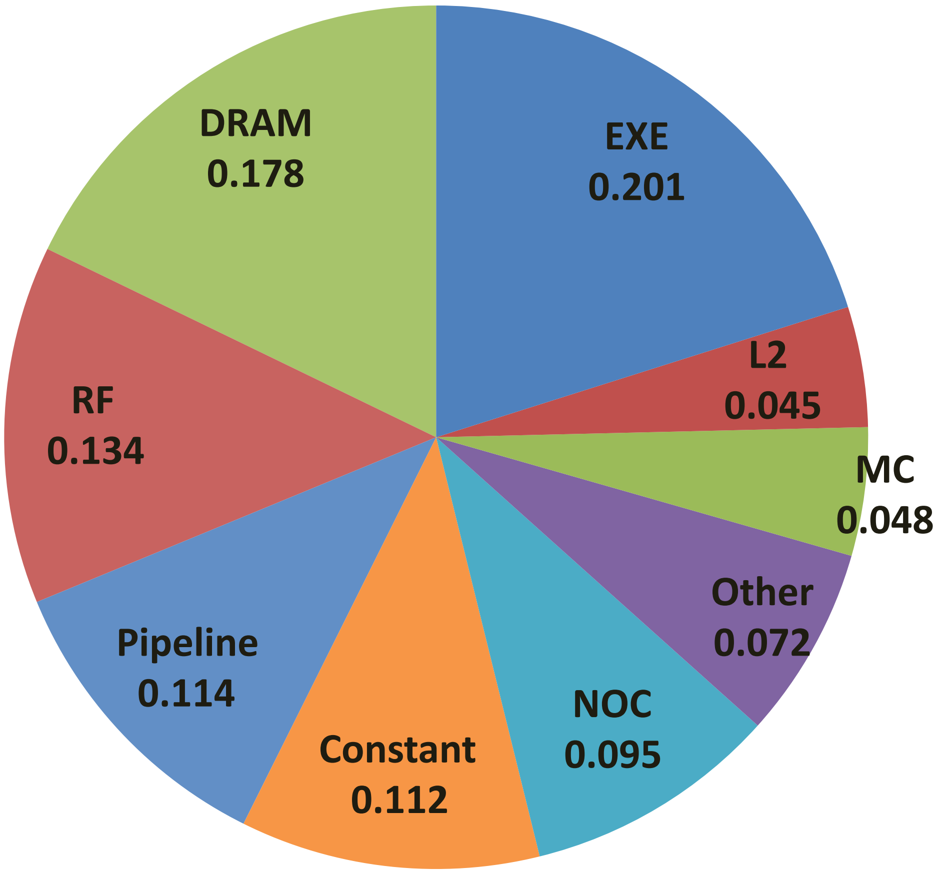
GPGPU Overview (GTX480)



GPGPU Overview (GTX480)

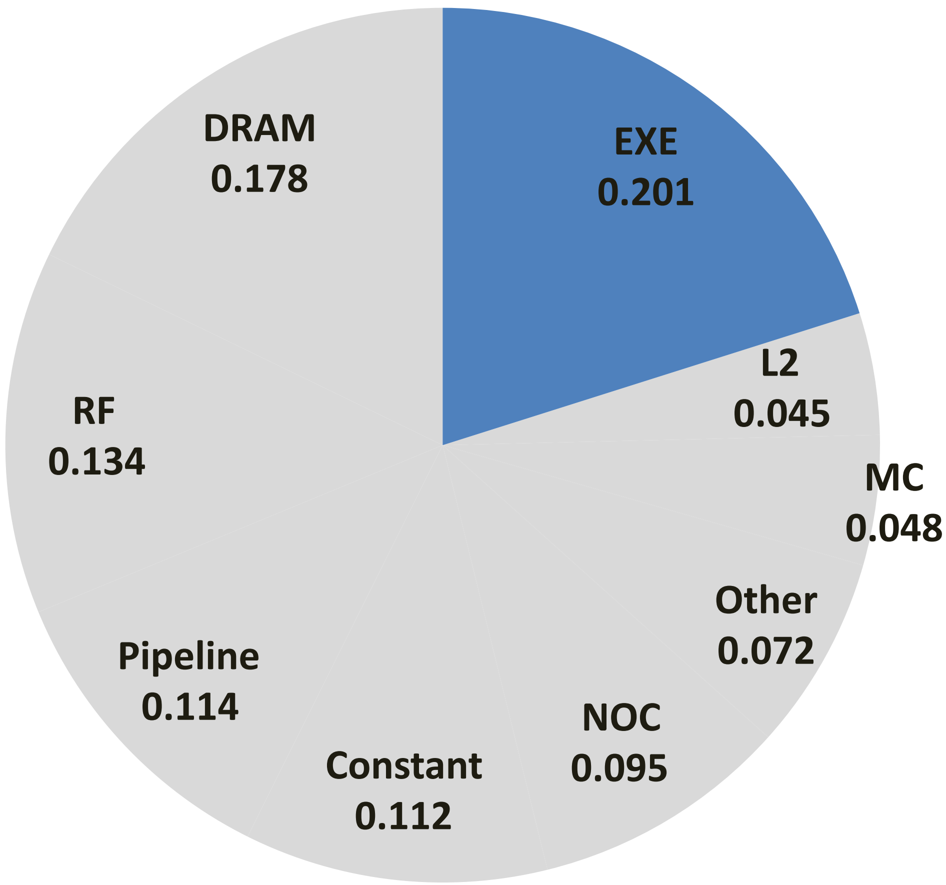


GPGPU Power Break-Down



GPUWattch, ISCA 2013

GPGPU Power Break-Down



EXE 20.1%

GPUWattch, ISCA 2013

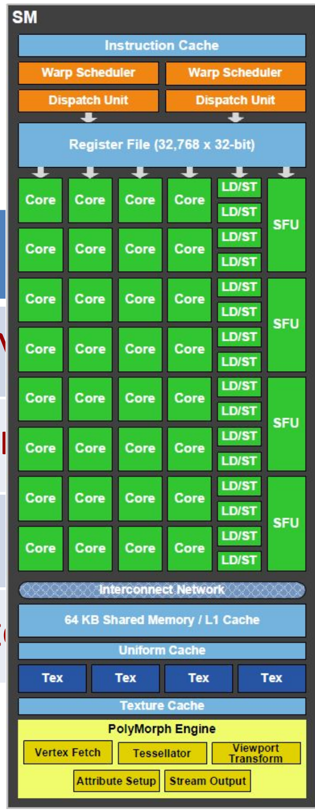
GPU Scaling Trend

GPU	Fermi GTX 480	Kepler GTX 680	Maxwell GTX 980
Cores (SMs)	16	8	16
Execution Units	512	1536	2048
RF size	128KB/SM	256KB/SM	256KB/SM
#transistors	3 billion	3.5 billion	5.2 billion

GPU Scaling Trend

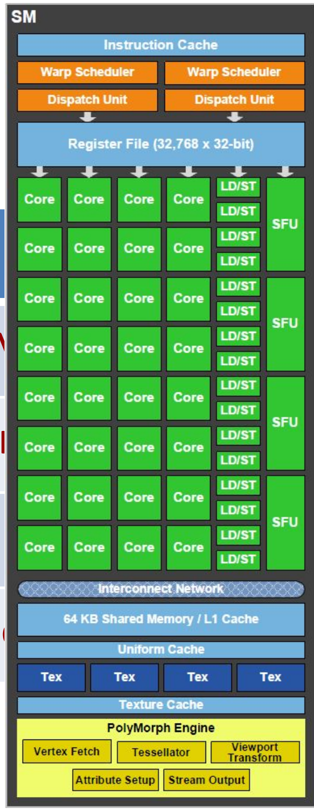
GPU	Fermi GTX 480	Kepler GTX 680	Maxwell GTX 980
Cores (SMs)	16	8	16
Execution Units	512	1536	2048
RF size	128KB/SM	256KB/SM	256KB/SM
#transistors	3 billion	3.5 billion	5.2 billion

GPU Scaling Trend



GPU	ermi GTX 480	Kepler GTX 680	Maxwell GTX 980
Cores (S	6	8	16
Executio	12	1536	2048
RF size	28KB/SM	256KB/SM	256KB/SM
#transist	billion	3.5 billion	5.2 billion

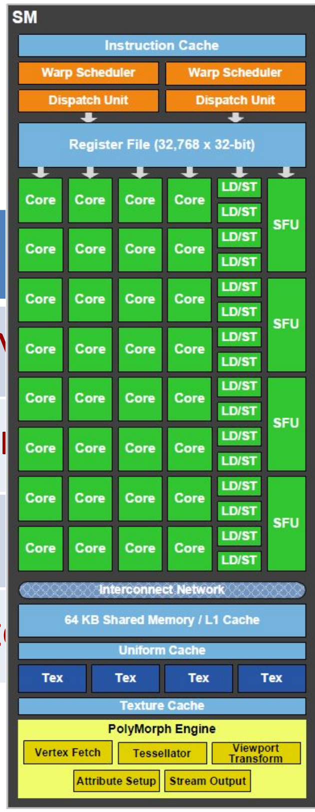
GPU Scaling Trend

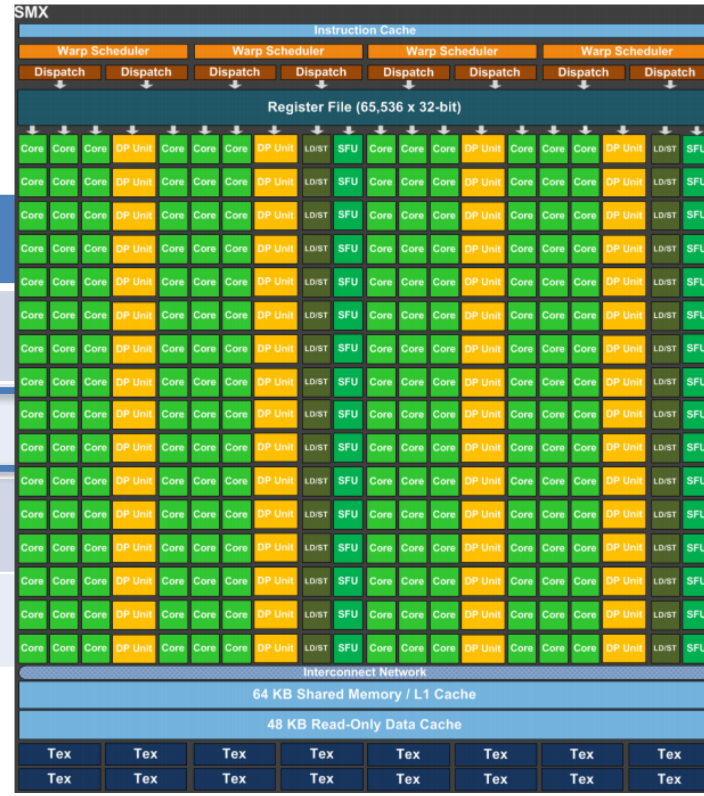


GPU	ermi GTX 480	Kepler GTX 680	Maxwell GTX 980
Cores (SN	6	8	16
Executio	12	1536	2048
RF size	28KB/SM	256KB/SM	256KB/SM
#transist	billion	3.5 billion	5.2 billion

GPU Scaling Trend

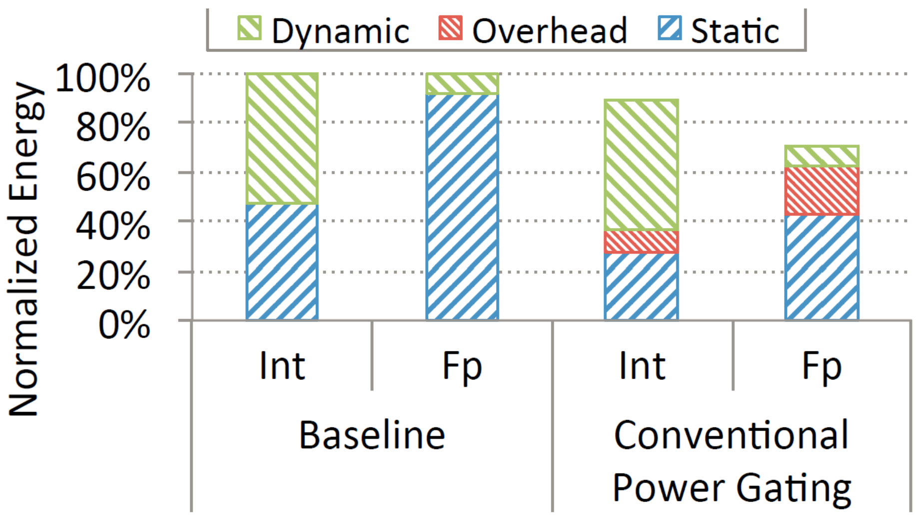
GPU	ermi GTX 480	TX 980
Cores (SM	6	
Executio	12	
RF size	28KB/SM	
#transist	billion	





Technology Scaling

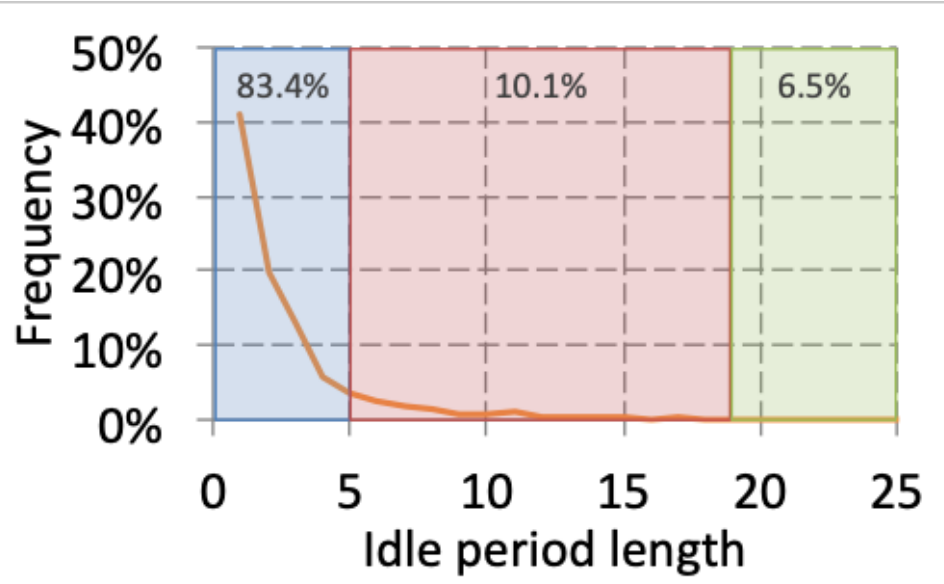
- As technology scales leakage power will increase
  - Accounts for 50% of the execution units power
- Power Gating can be used to reduce the leakage power
  - Need long idle periods to be effective



Warped Gates, MICRO 2012

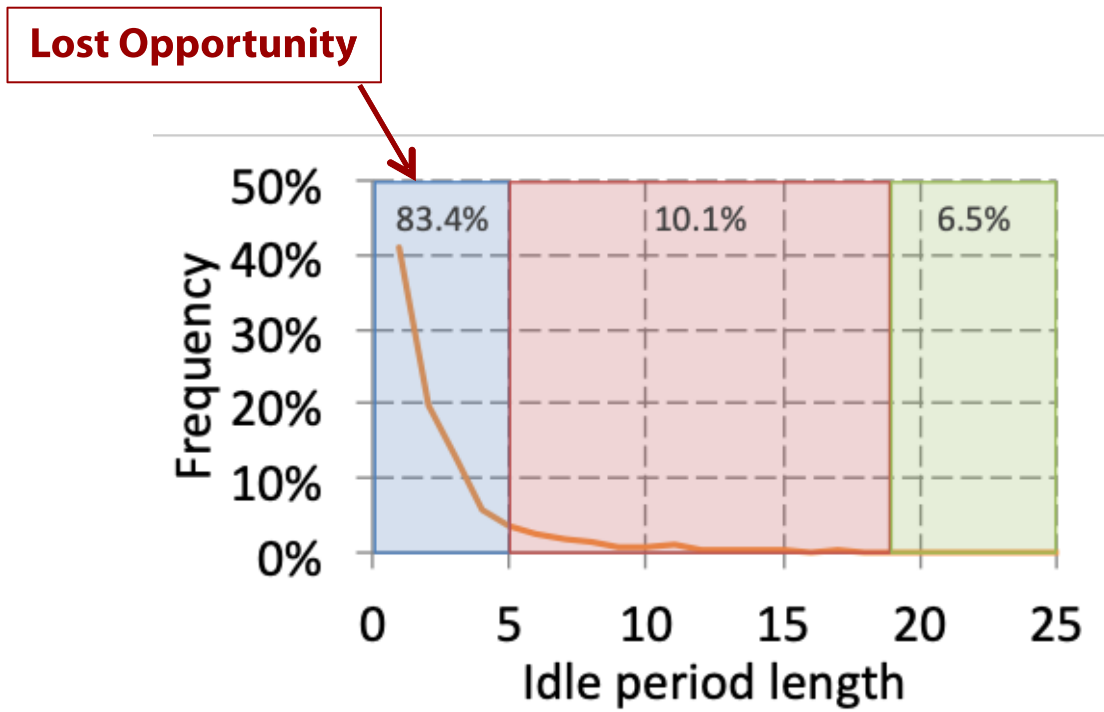
Power Gating Challenges in GPGPUs

- Int. Unit idle period length distribution for hotspot
  - Assume 5 idle detect, 14 BET



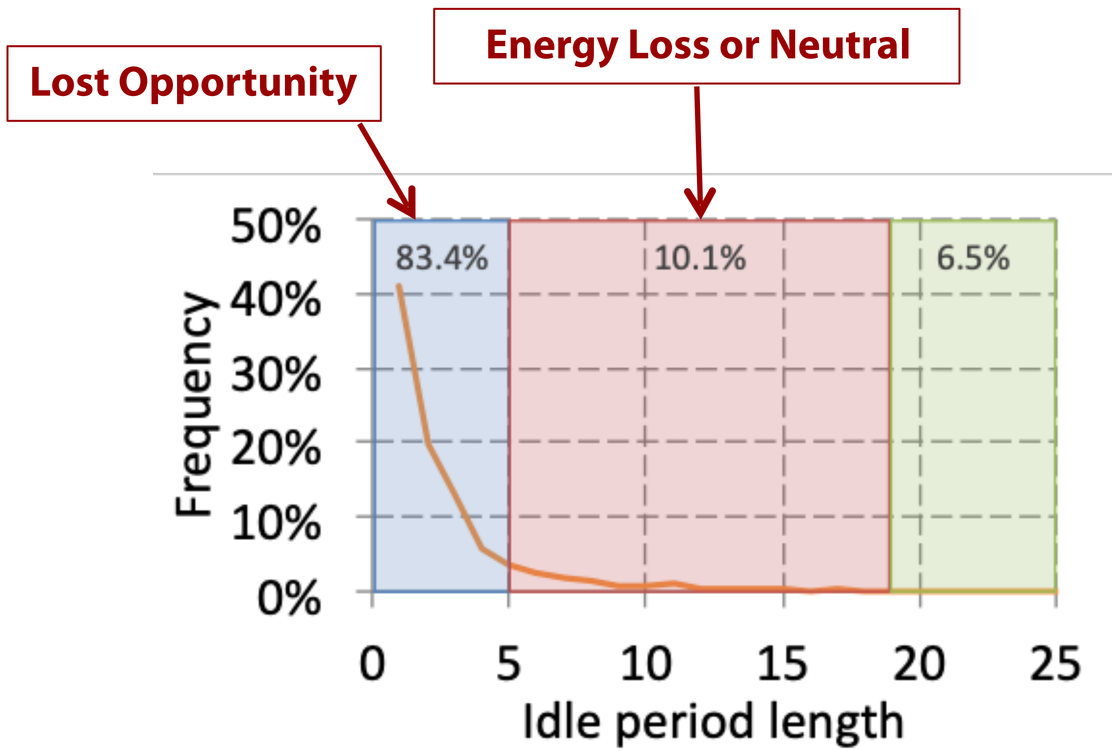
Power Gating Challenges in GPGPUs

- Int. Unit idle period length distribution for hotspot
  - Assume 5 idle detect, 14 BET



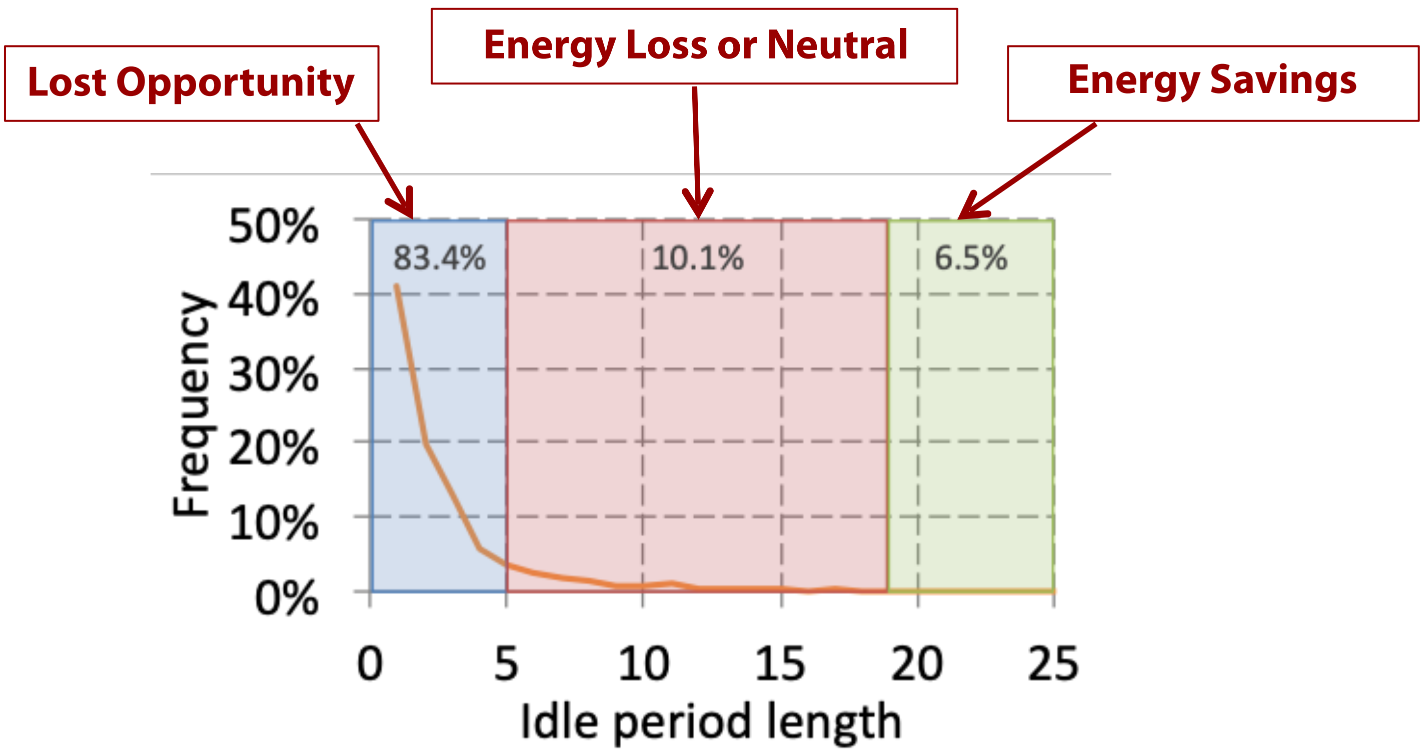
Power Gating Challenges in GPGPUs

- Int. Unit idle period length distribution for hotspot
  - Assume 5 idle detect, 14 BET



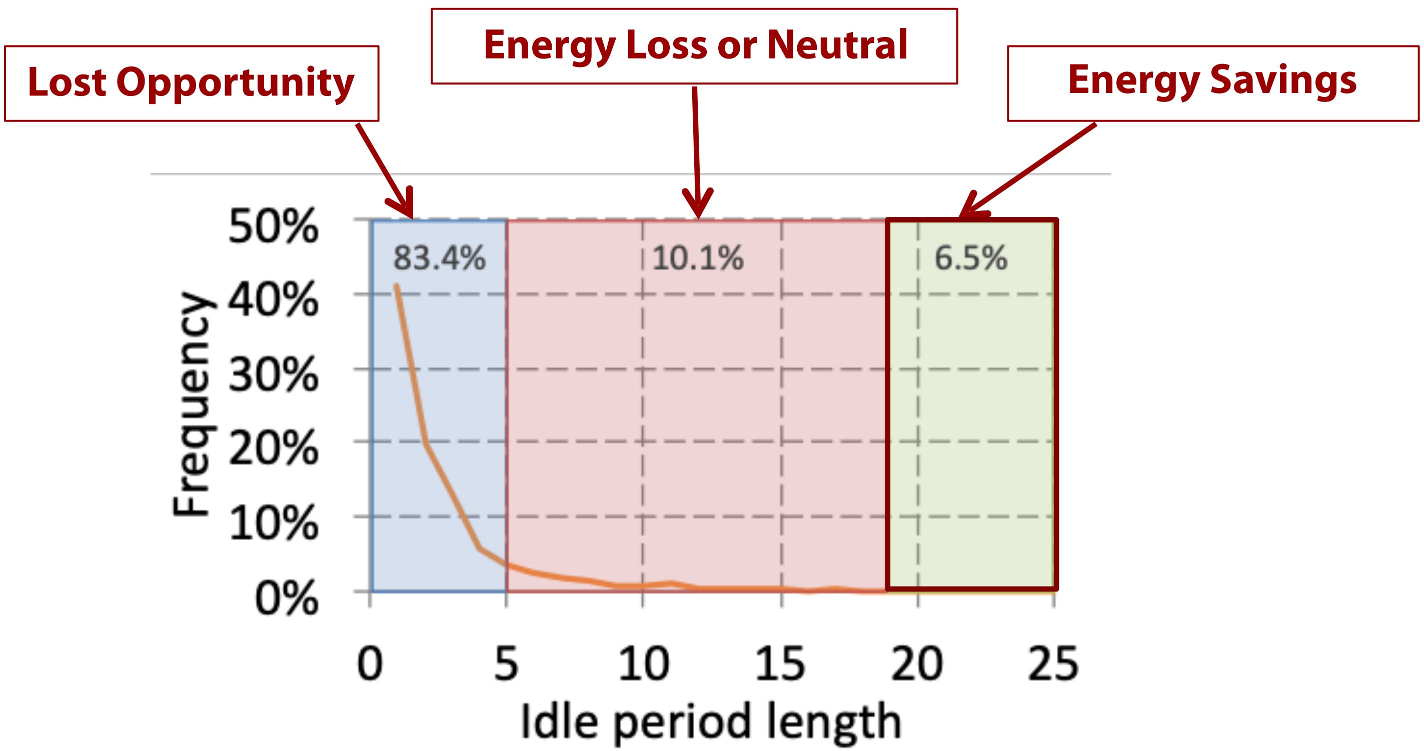
Power Gating Challenges in GPGPUs

- Int. Unit idle period length distribution for hotspot
  - Assume 5 idle detect, 14 BET



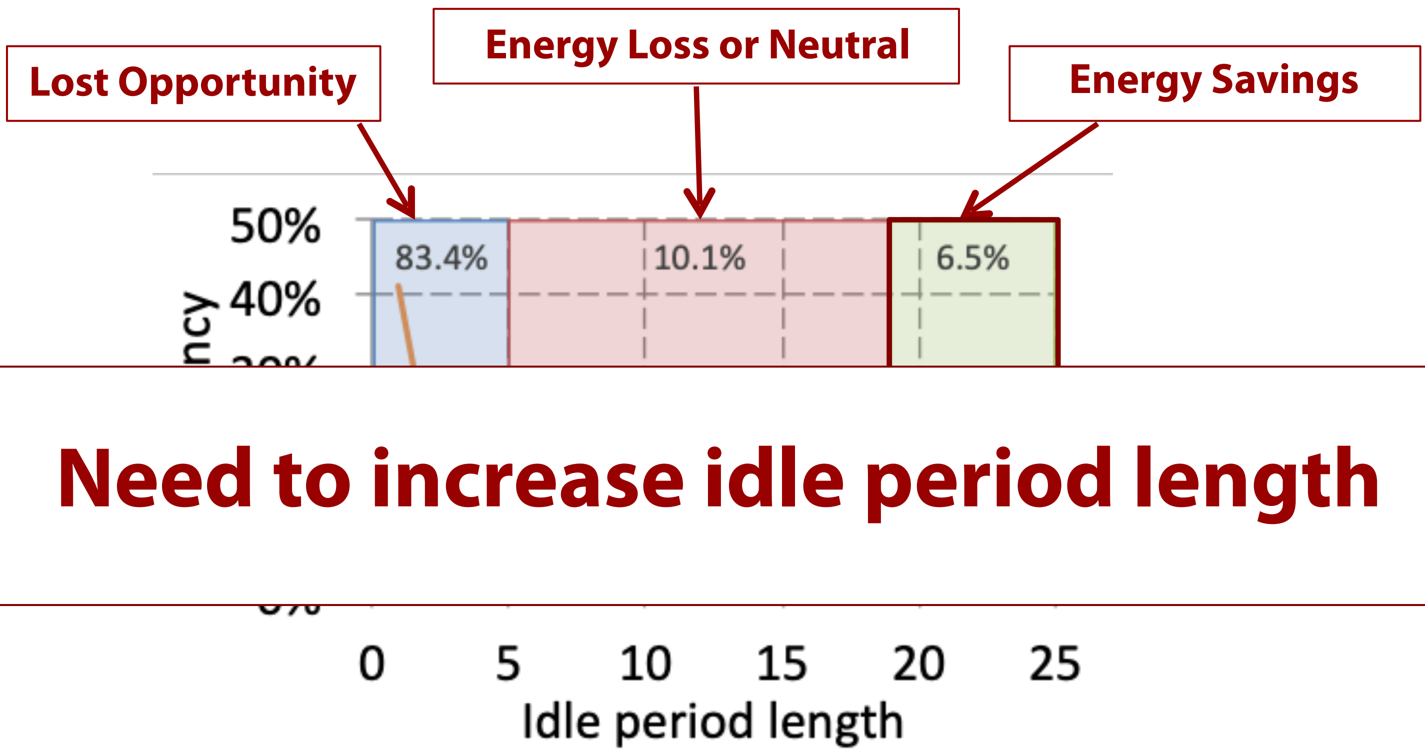
Power Gating Challenges in GPGPUs

- Int. Unit idle period length distribution for hotspot
  - Assume 5 idle detect, 14 BET



Power Gating Challenges in GPGPUs

- Int. Unit idle period length distribution for hotspot
  - Assume 5 idle detect, 14 BET



Need to increase idle period length

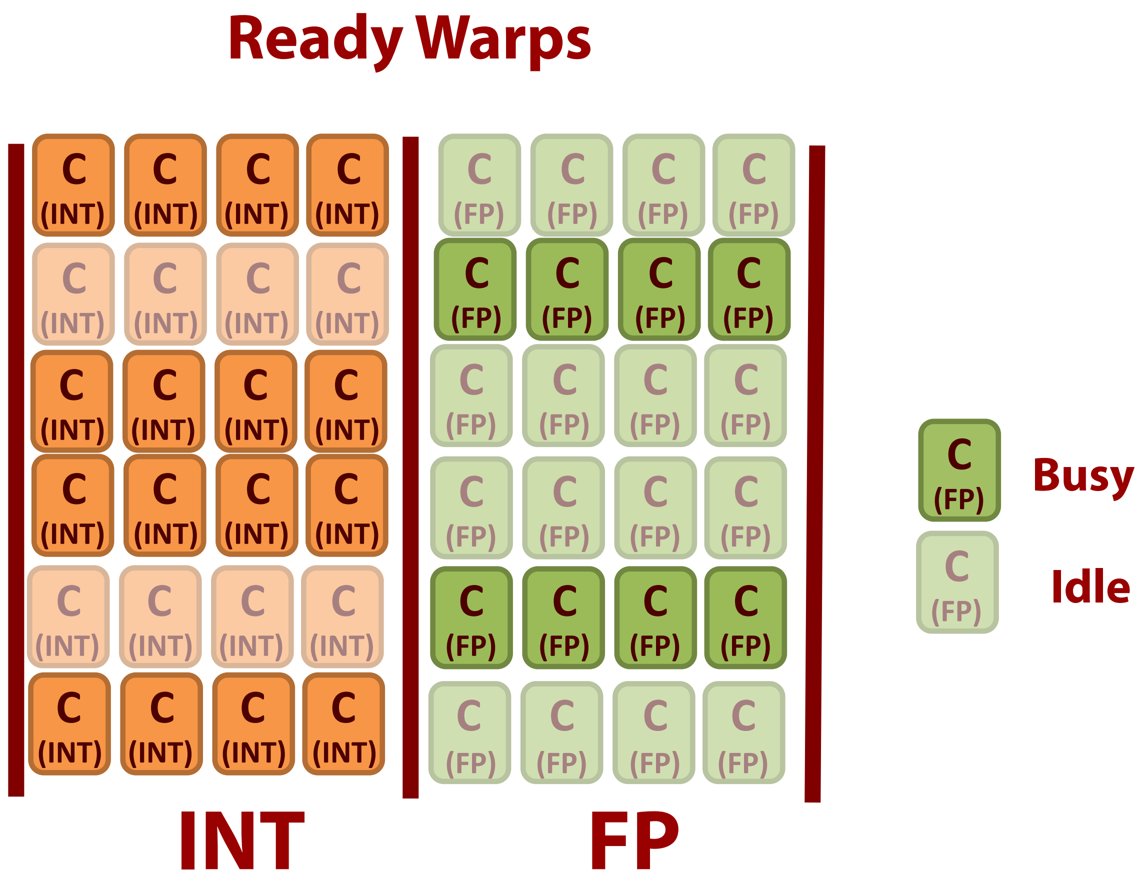
Warp Scheduler Effect on Power Gating

INT	FP	INT	INT	FP	INTO

Ready Warps

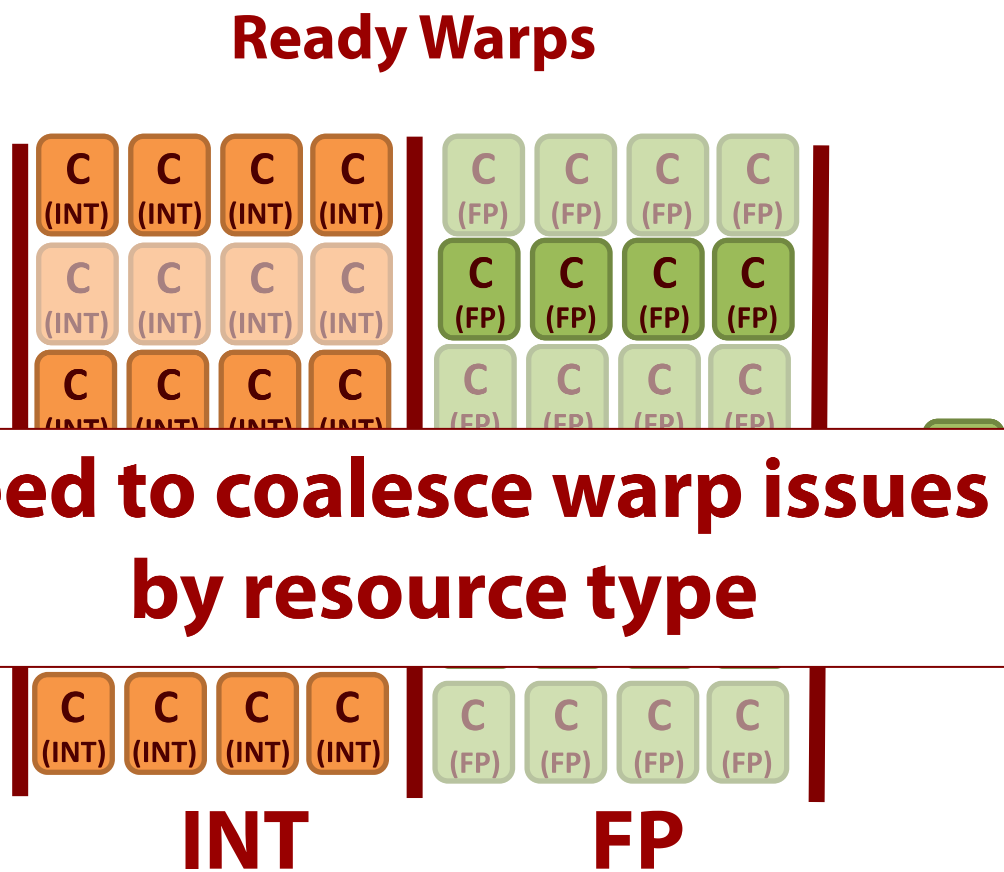
INT **FP**

Warp Scheduler Effect on Power Gating



Warp Scheduler Effect on Power Gating

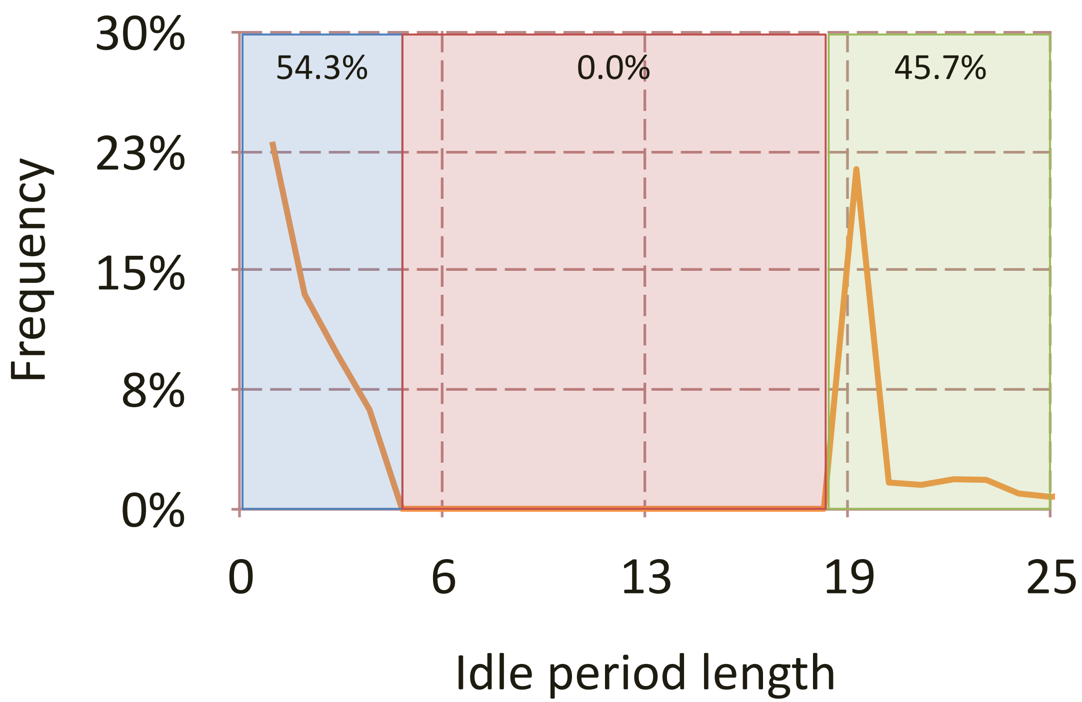
- Idle periods
  in
  by
  th
  scheduled



Need to coalesce warp issues by resource type

Related Work/Warped-Gates*

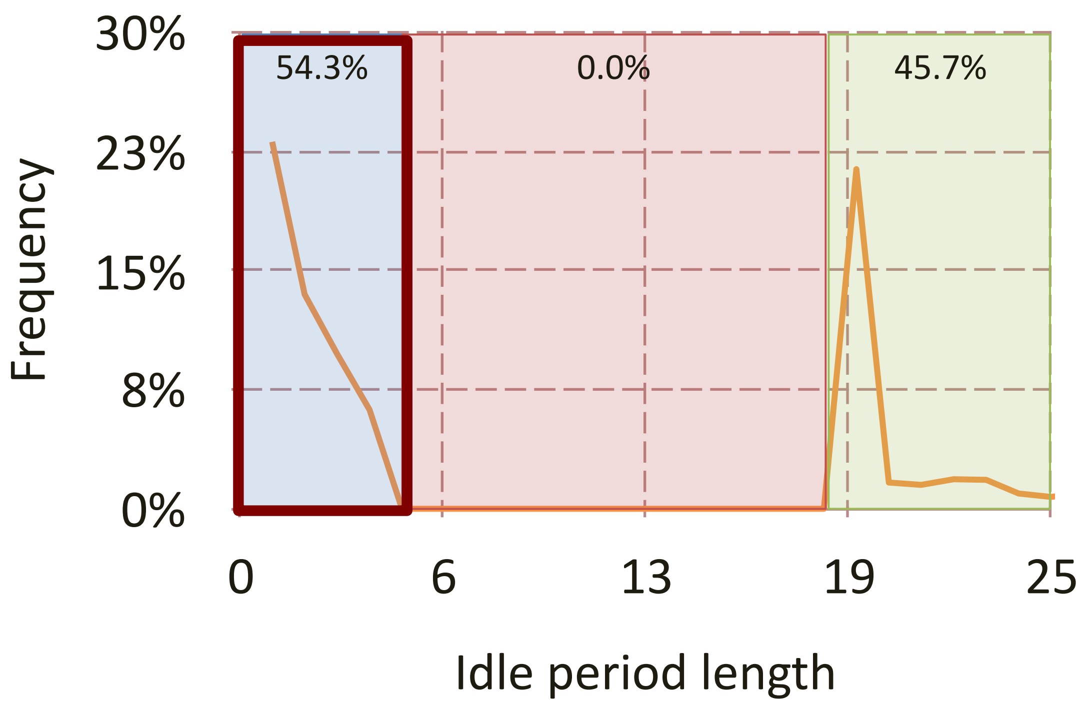
- Schedule instructions based on their type
- Force power gated units to stay in power gating state for at least the breakeven time



*Warped-Gates, MICRO 2013

Related Work/Warped-Gates*

- Schedule instructions based on their type
- Force power gated units to stay in power gating state for at least the breakeven time



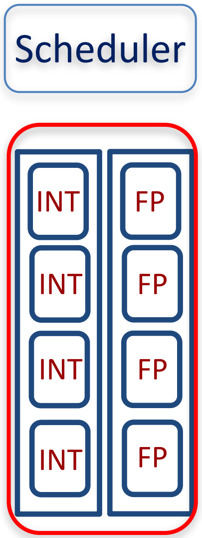
*Warped-Gates, MICRO 2013

Fine grain idleness

- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities

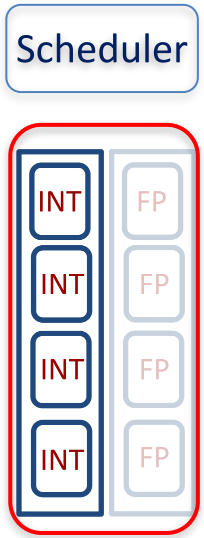
Fine grain idleness

- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities



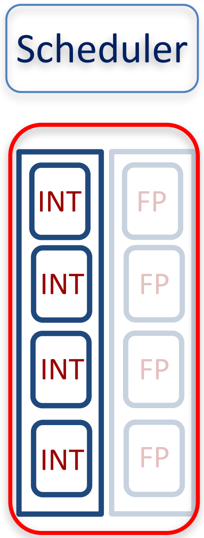
Fine grain idleness

- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities



Fine grain idleness

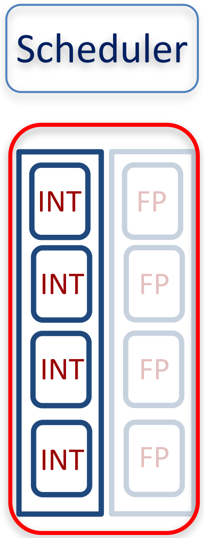
- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities



SP0

Fine grain idleness

- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities

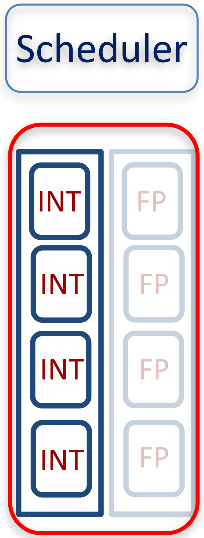


SP0

Cycle X 1111 1111

Fine grain idleness

- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities

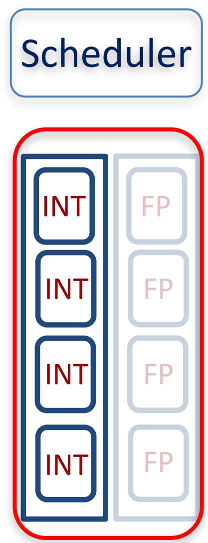


SP0

Cycle X	1111 1111
Cycle X+1	Bubble

Fine grain idleness

- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities

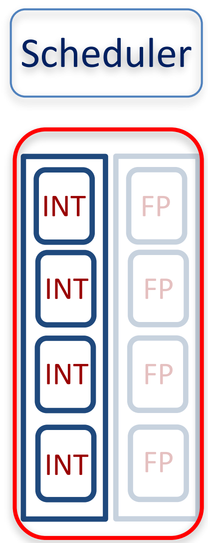


SP0

Cycle X	1111 1111
Cycle X+1	Bubble
Cycle X+2	1111 1111

Fine grain idleness

- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities

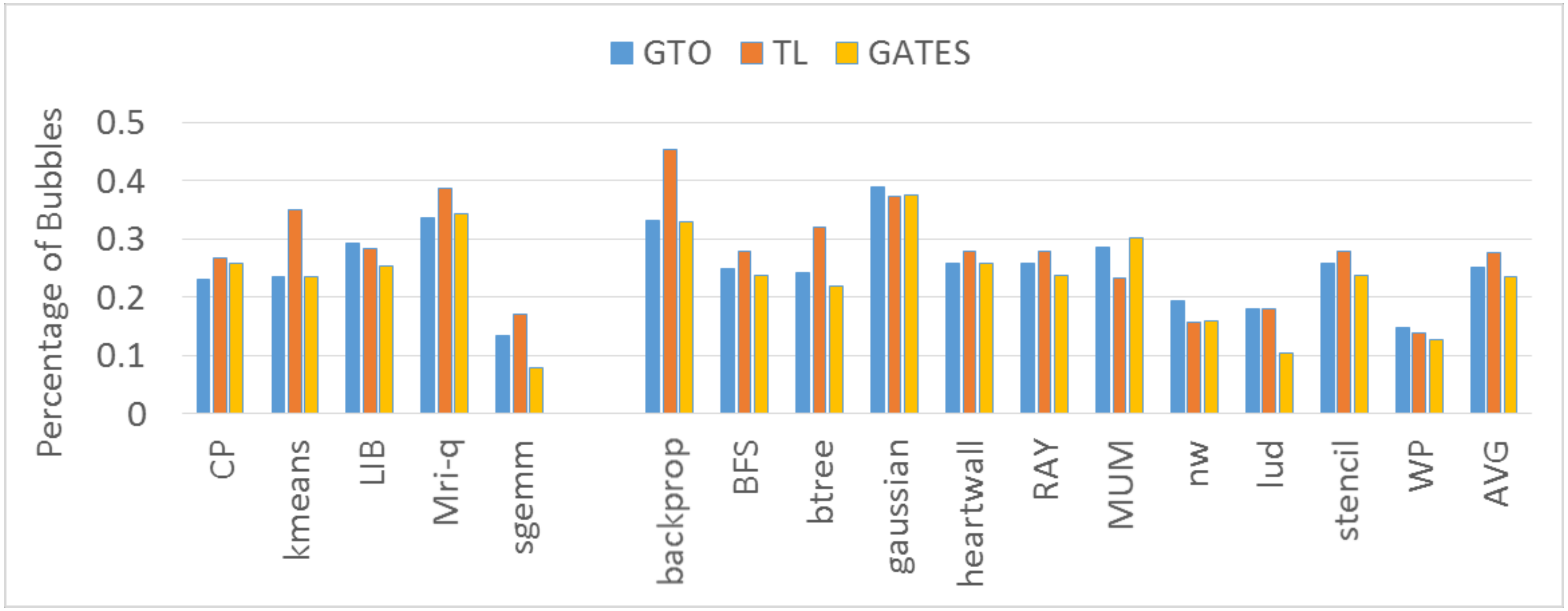


SP0

Cycle X	1111 1111
Cycle X+1	Bubble
Cycle X+2	1111 1111
Cycle X+3	Bubble

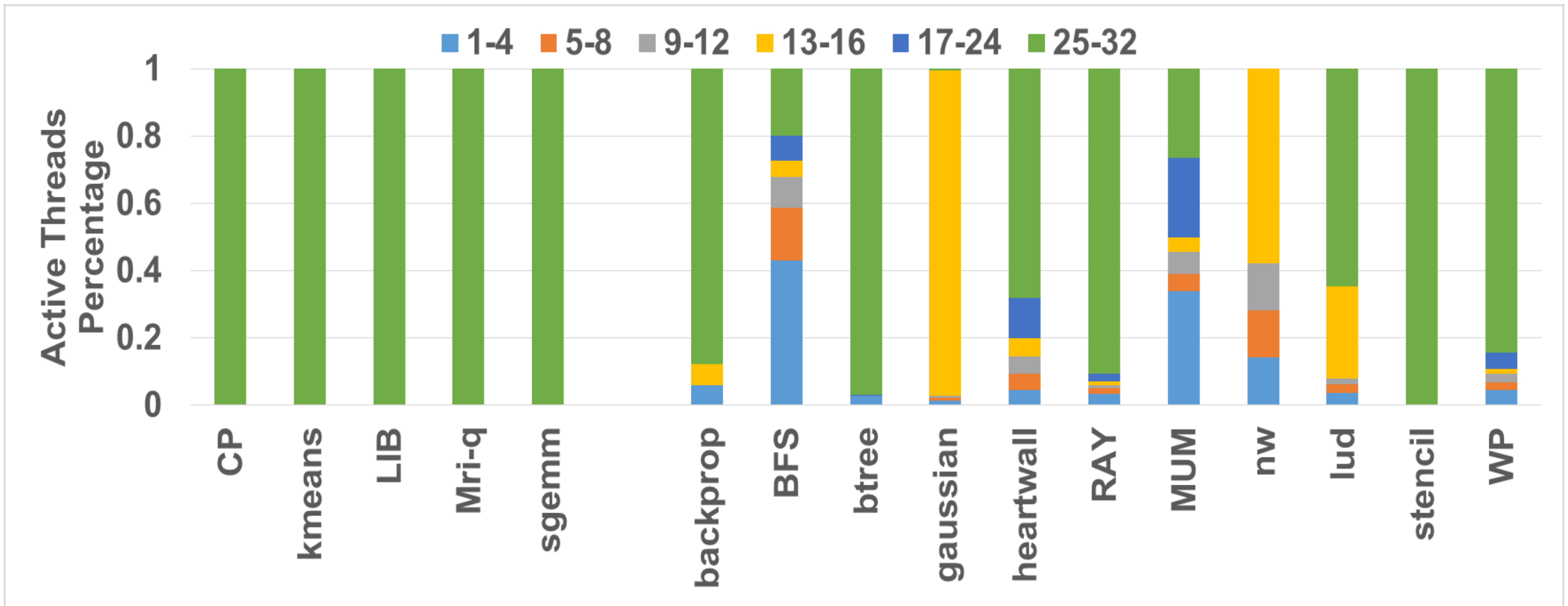
Fine grain idleness

- Temporal idleness
  - Infrequent issues to the same pipeline
  - Finely interspersed leading to limited power gating opportunities



Fine grain idleness

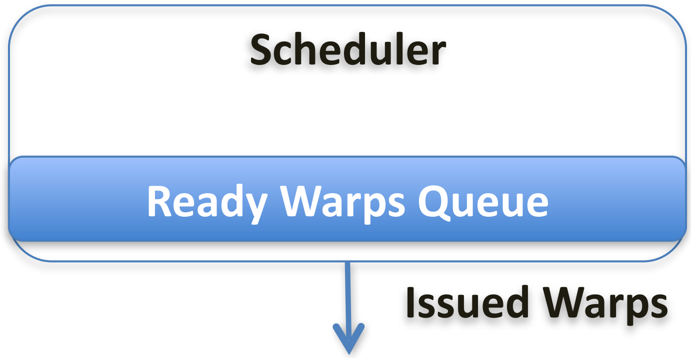
- Spatial Idleness
  - Lanes have different activity
    - Branch divergence
    - Insufficient parallelism



Warp Folding

➢ **Improve the power gating potential by coalescing the pipeline bubbles**

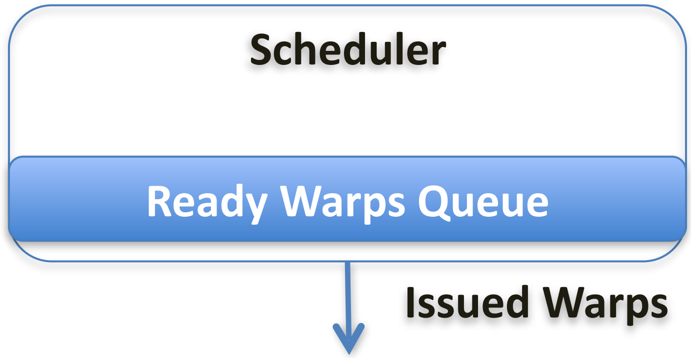
Warp Folding



Active Mask: 1 1 1 1 1 1 1 1

Active Mask: Bubble

Warp Folding



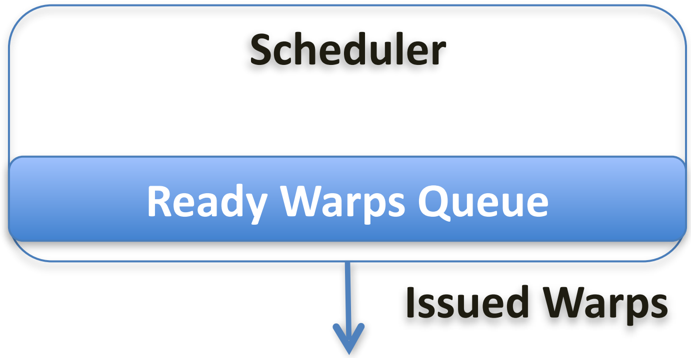
Active Mask: 1 1 1 1 1 1 1 1

Active Mask: Bubble

Sub_Warp0:

Sub_Warp1:

Warp Folding



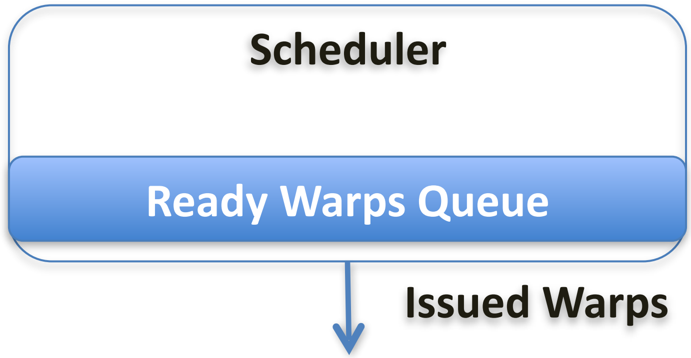
Active Mask: 1 1 1 1

Active Mask: Bubble

Sub_Warp0: 1 1 1 1

Sub_Warp1:

Warp Folding

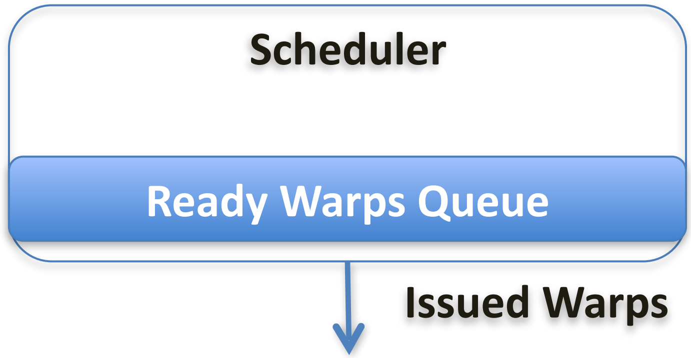


Active Mask:

Active Mask: Bubble

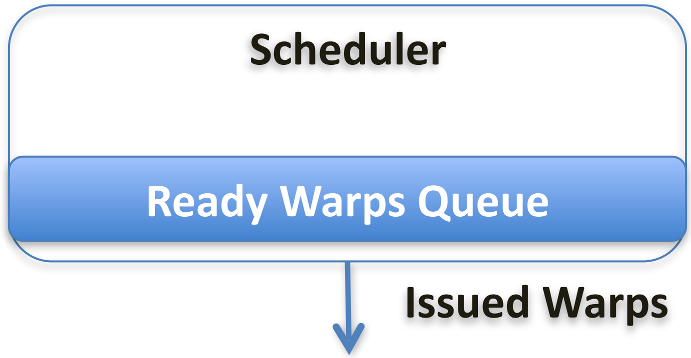
Sub_Warp0:	1	1	1	1				
Sub_Warp1:					1	1	1	1

Warp Folding



Sub_Warp0:	1 1 1 1	$\boxed{0\ 0\ 0\ 0}$
Sub_Warp1:	$\boxed{0\ 0\ 0\ 0}$	1 1 1 1

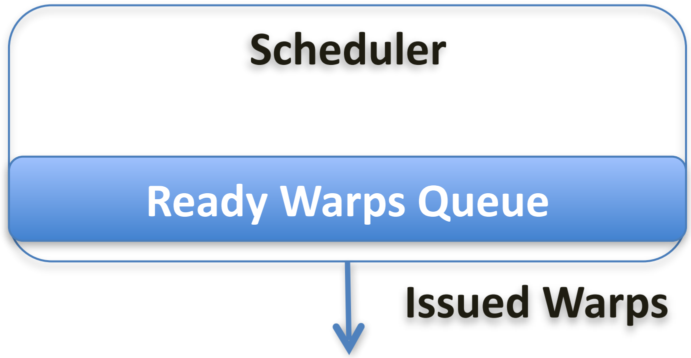
Warp Folding



Sub_Warp0:
Sub_Warp1: 0 0 0 0 1 1 1 1

Sub_Warp0: 1 1 1 1 0 0 0 0
Sub_Warp1:

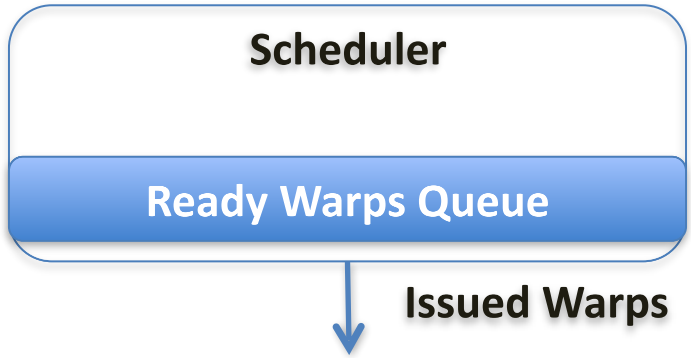
Warp Folding



Sub_Warp0:
Sub_Warp1: 0 0 0 0

Sub_Warp0: 1 1 1 1 0 0 0 0
Sub_Warp1: 1 1 1 1

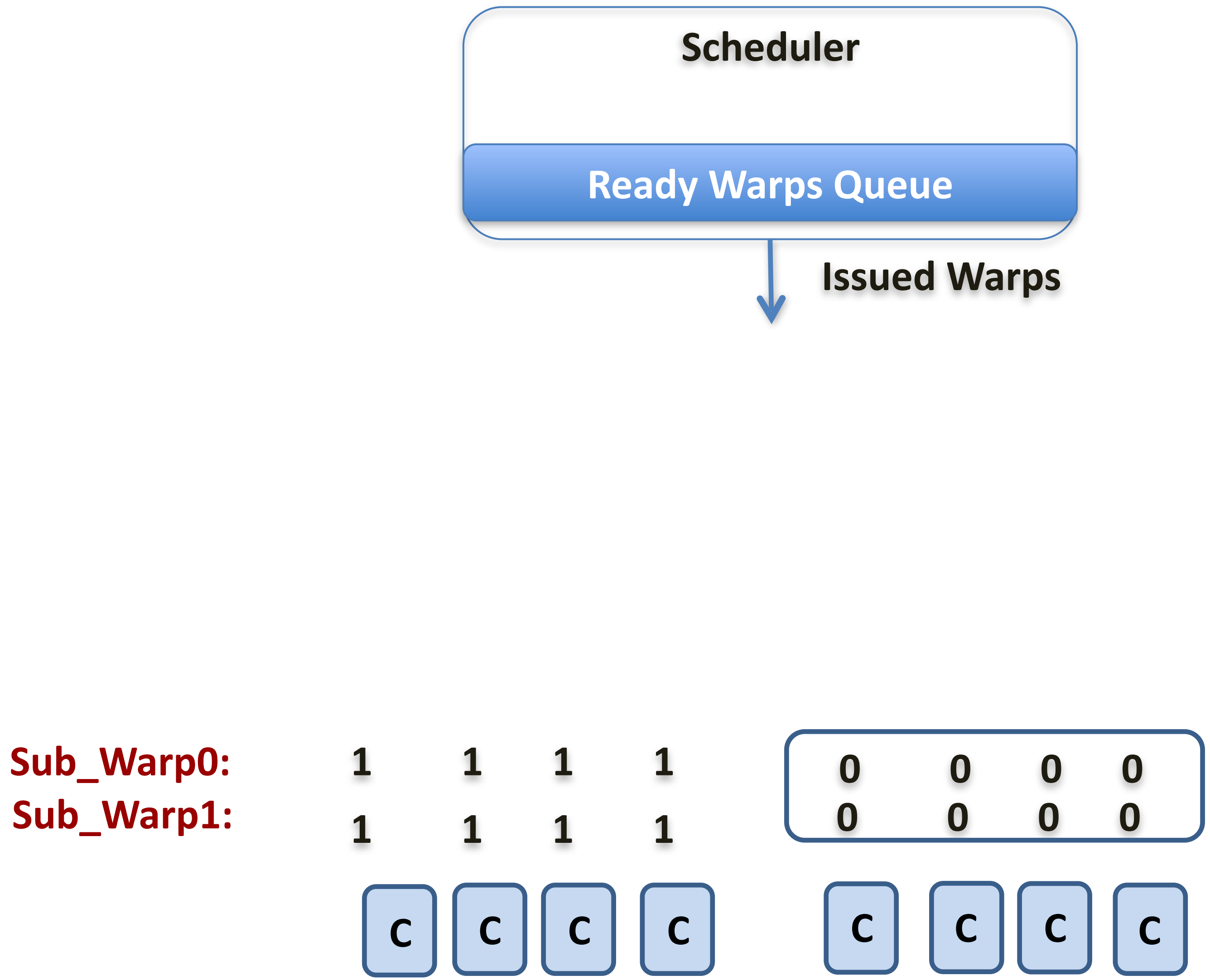
Warp Folding



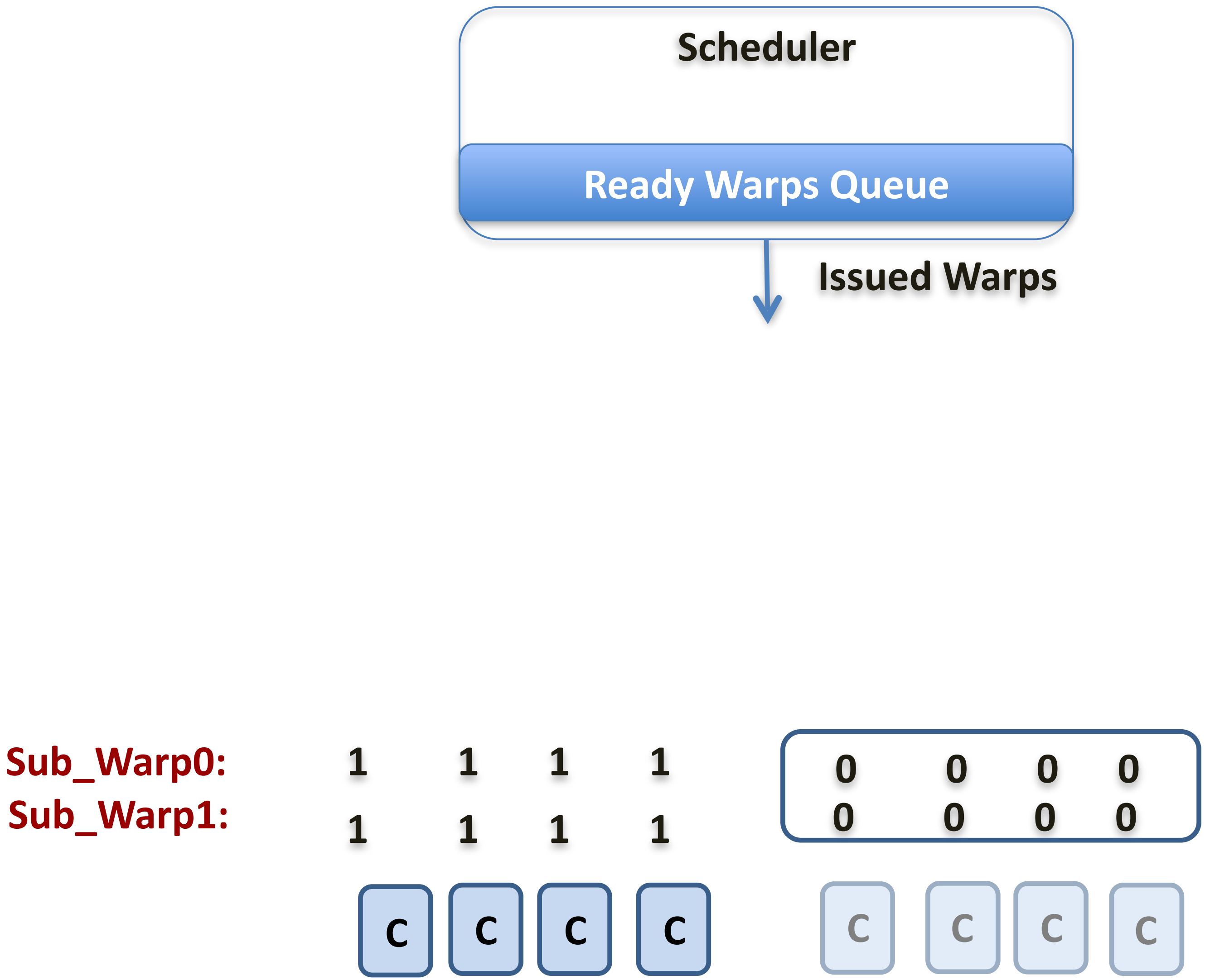
Sub_Warp0: 1 1 1 1 0 0 0 0

Sub_Warp1: 1 1 1 1 0 0 0 0

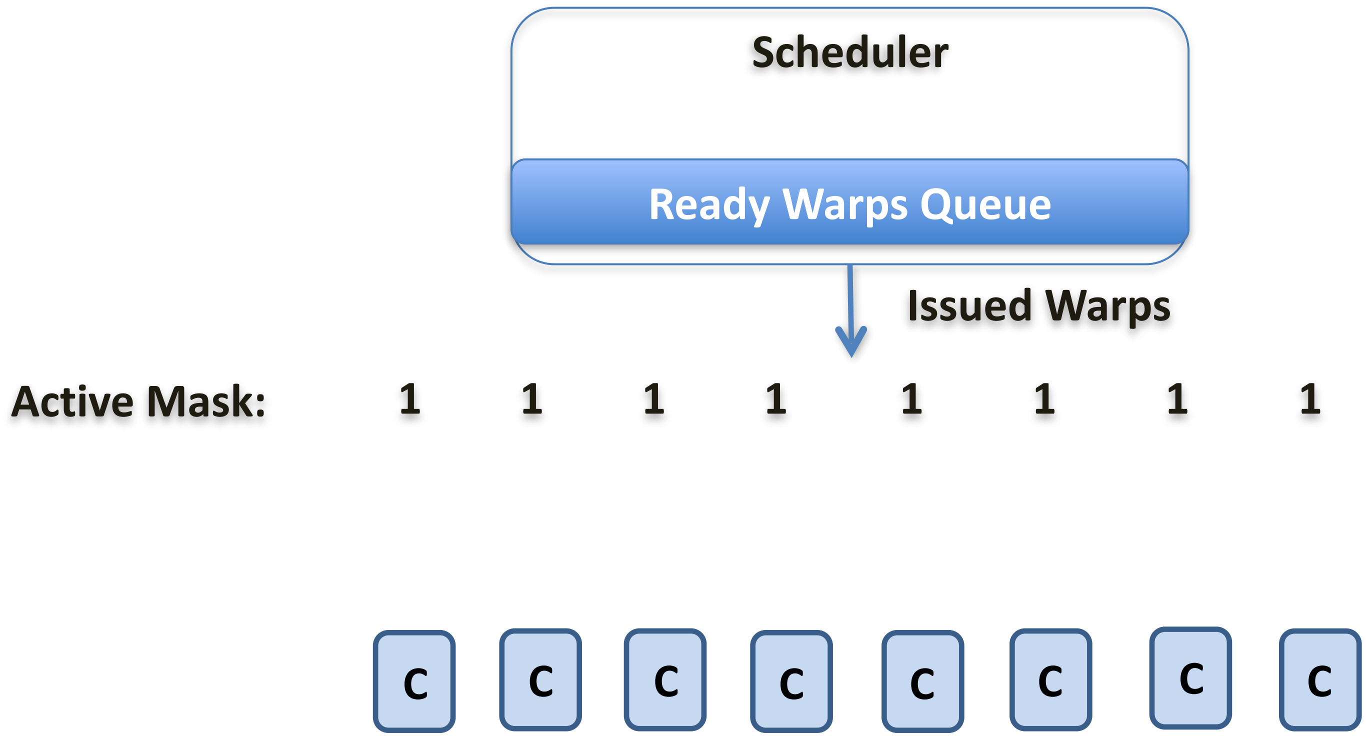
Warp Folding



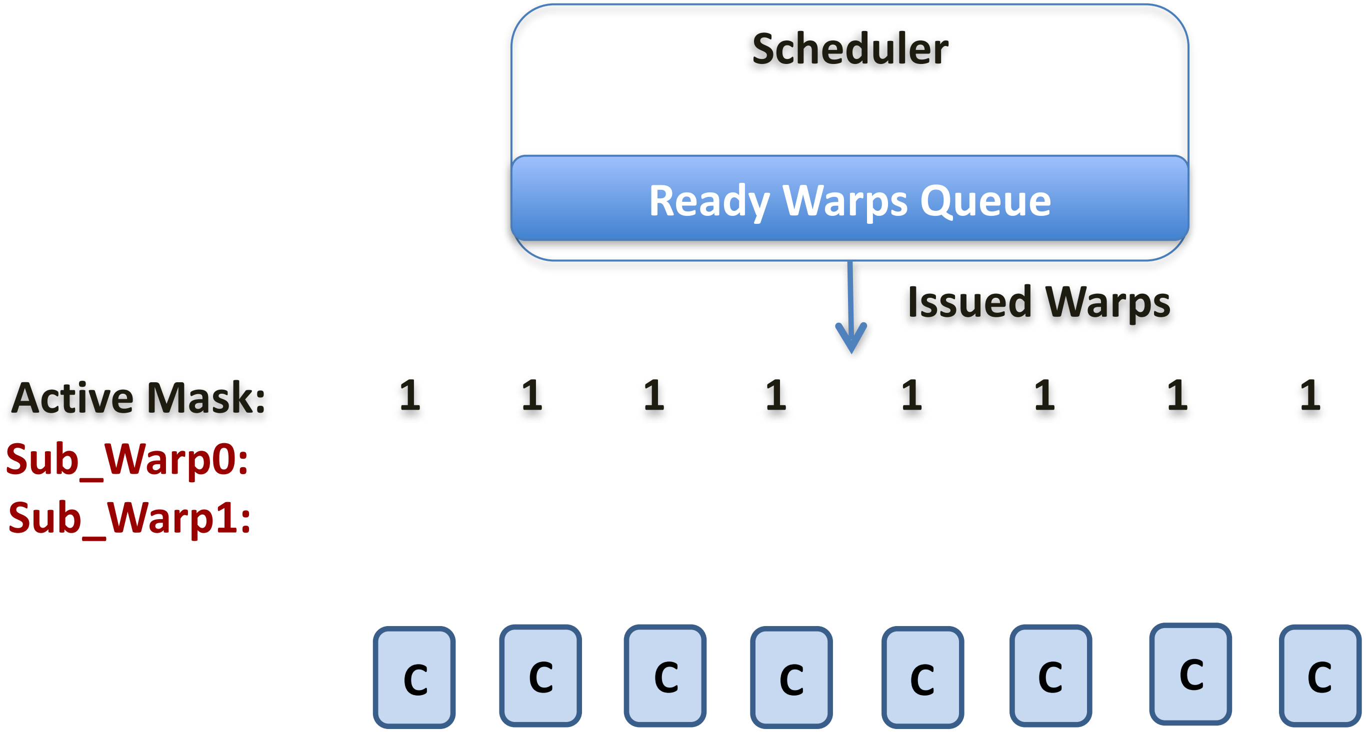
Warp Folding



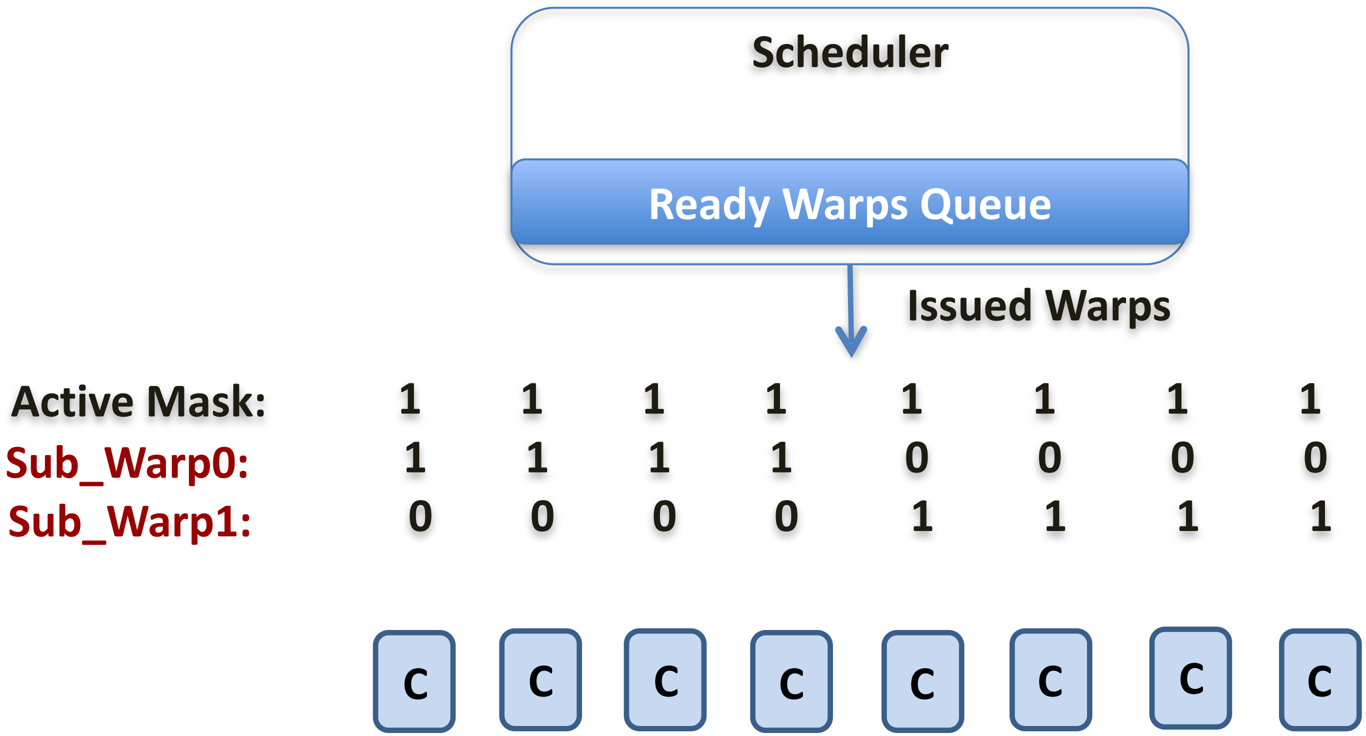
Folding Granularity



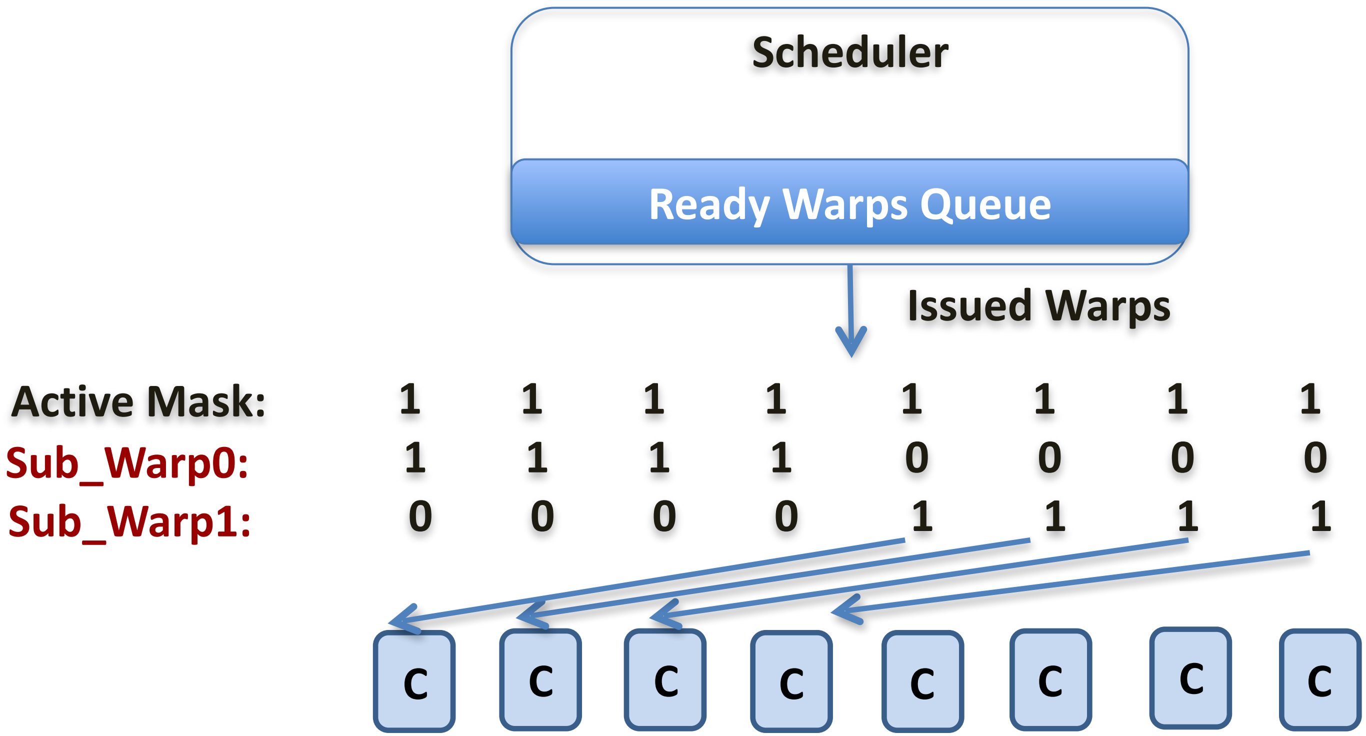
Folding Granularity



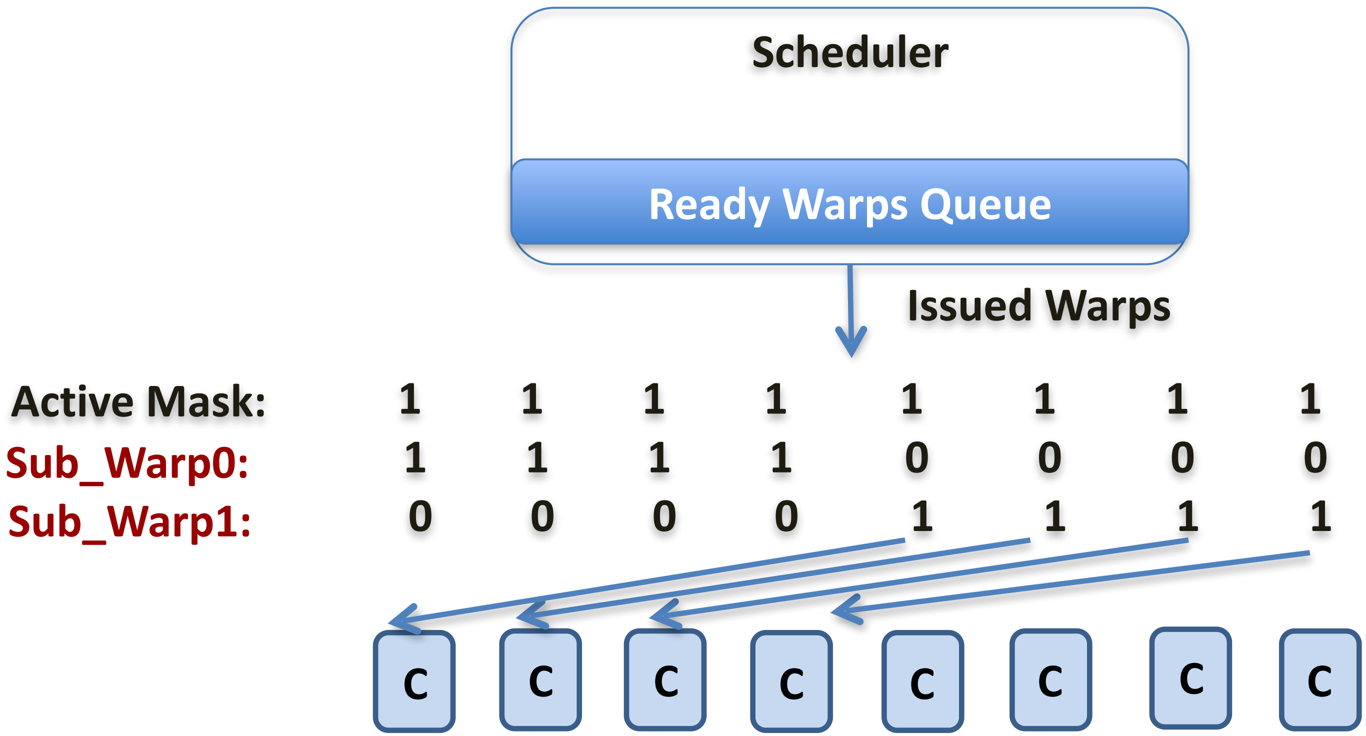
Folding Granularity



Folding Granularity

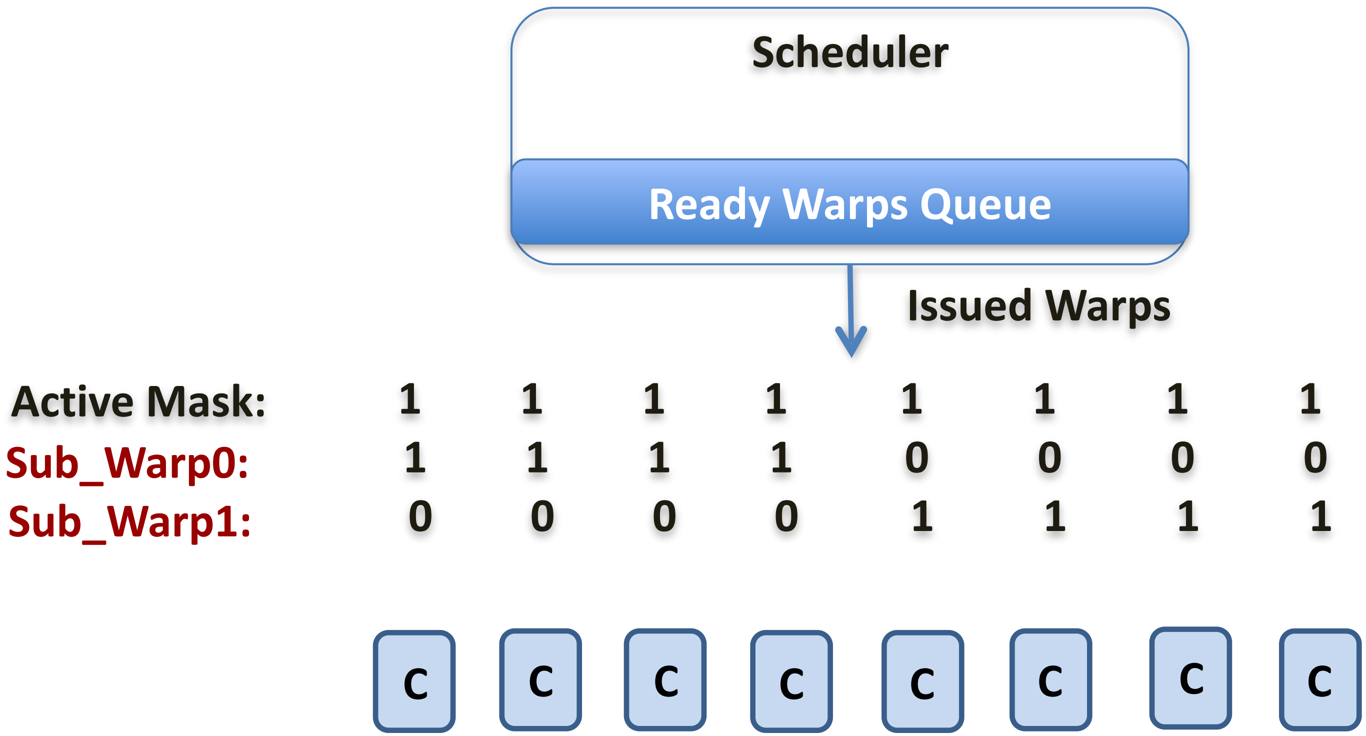


Folding Granularity

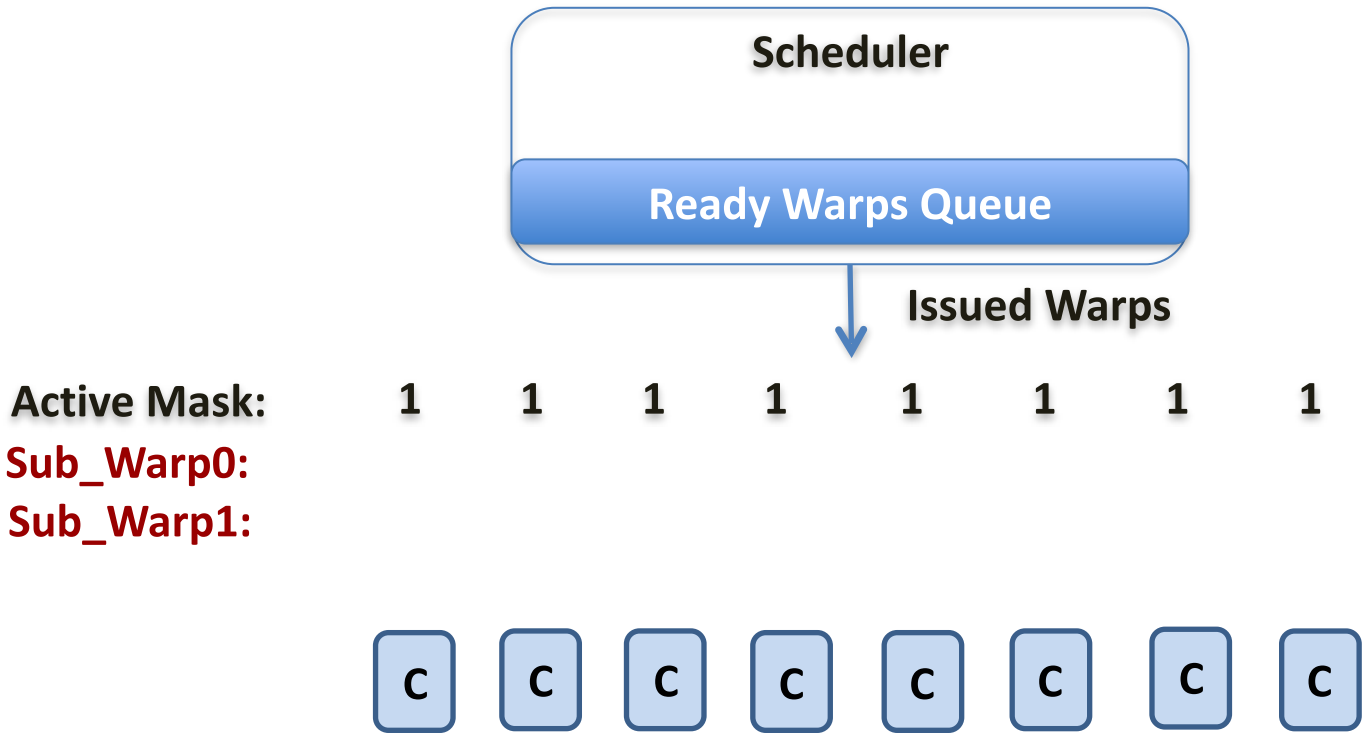


+ Simple
- High wiring overhead
- Delay

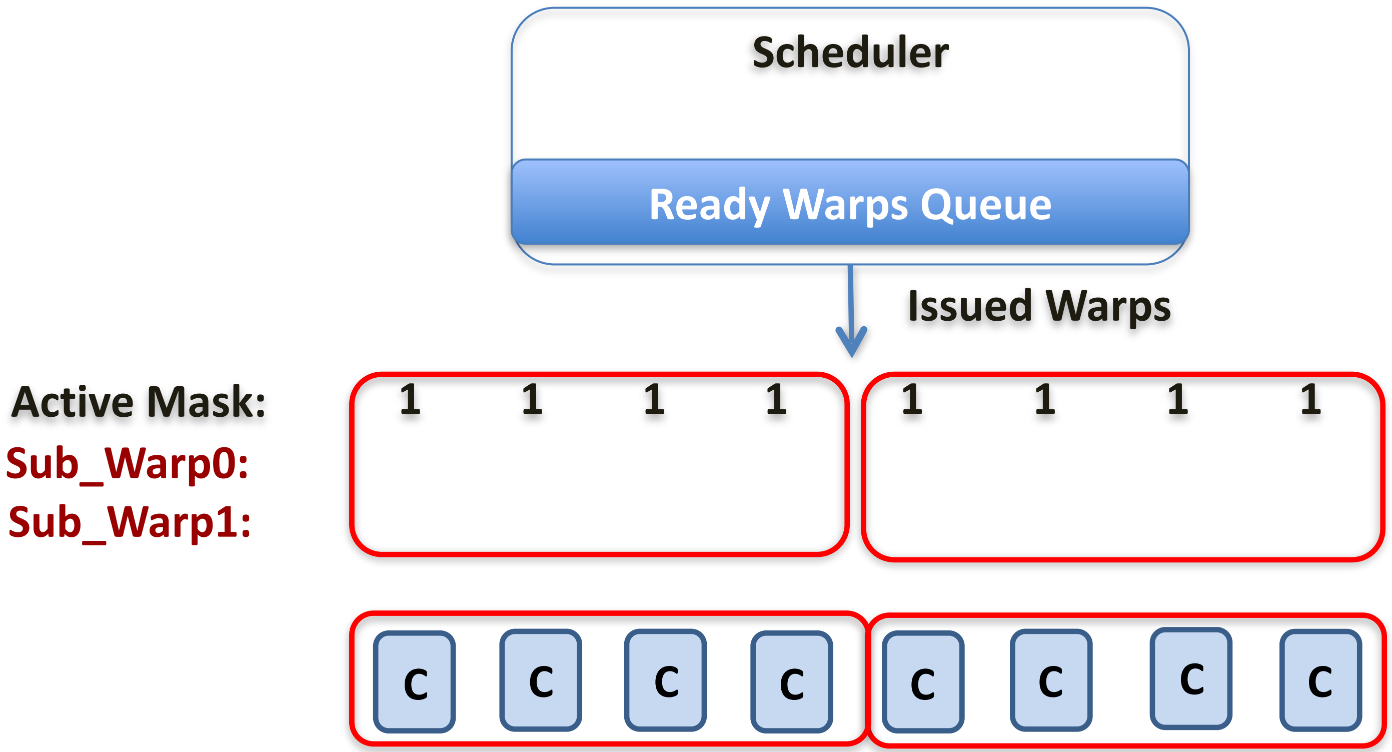
Folding Granularity



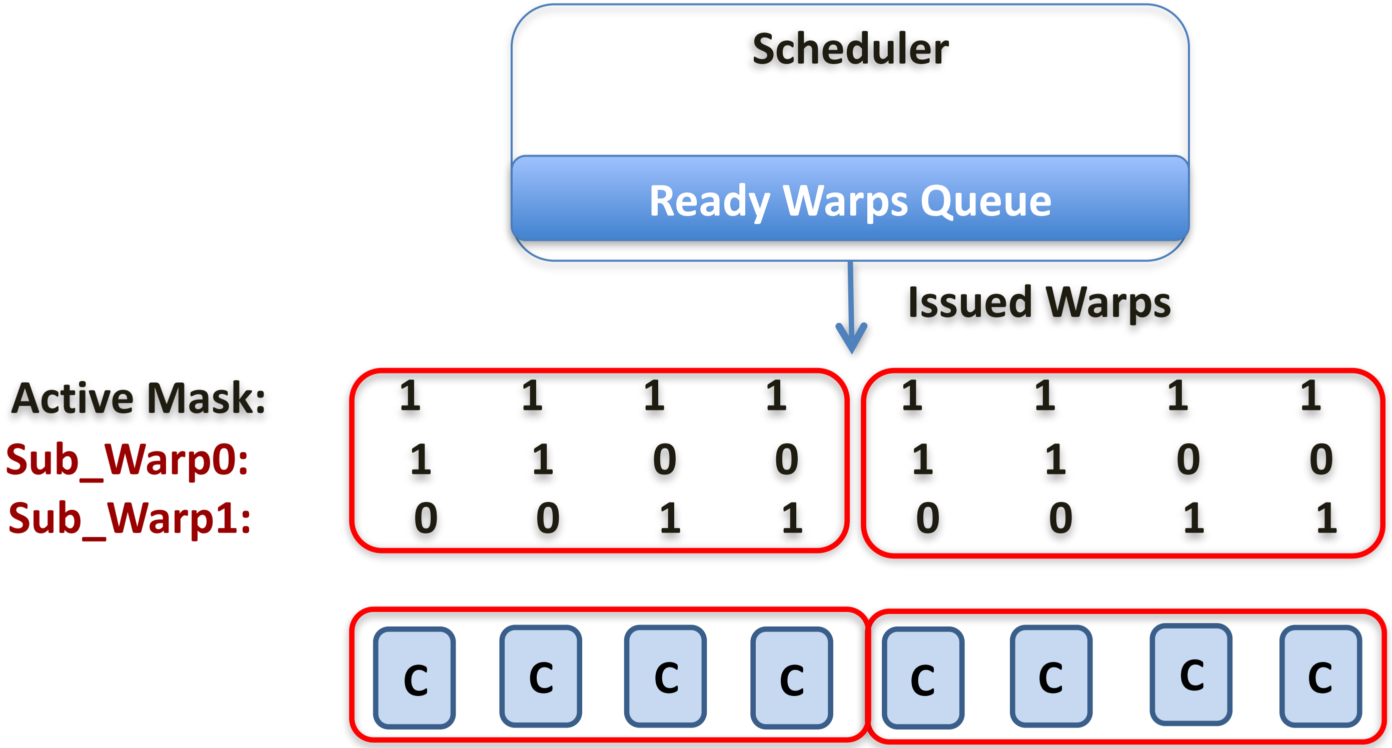
Folding Granularity



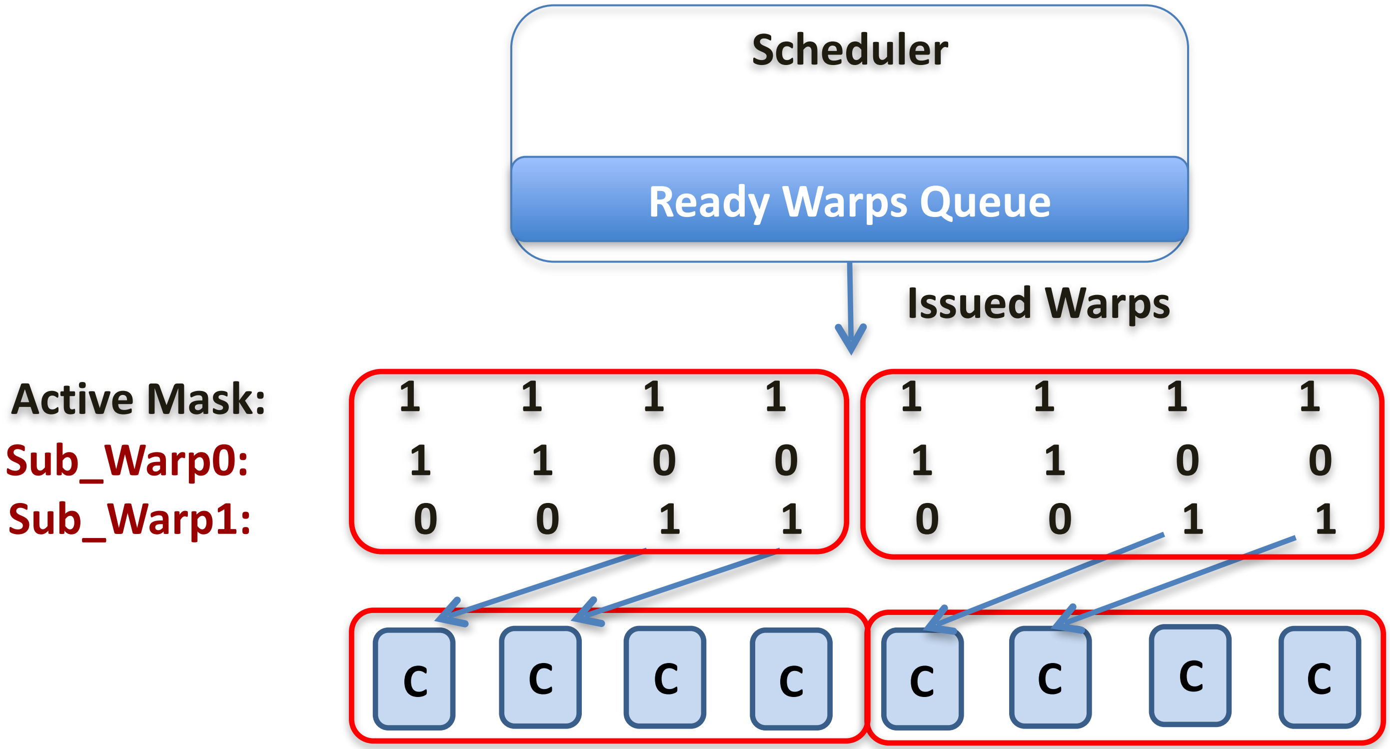
Folding Granularity



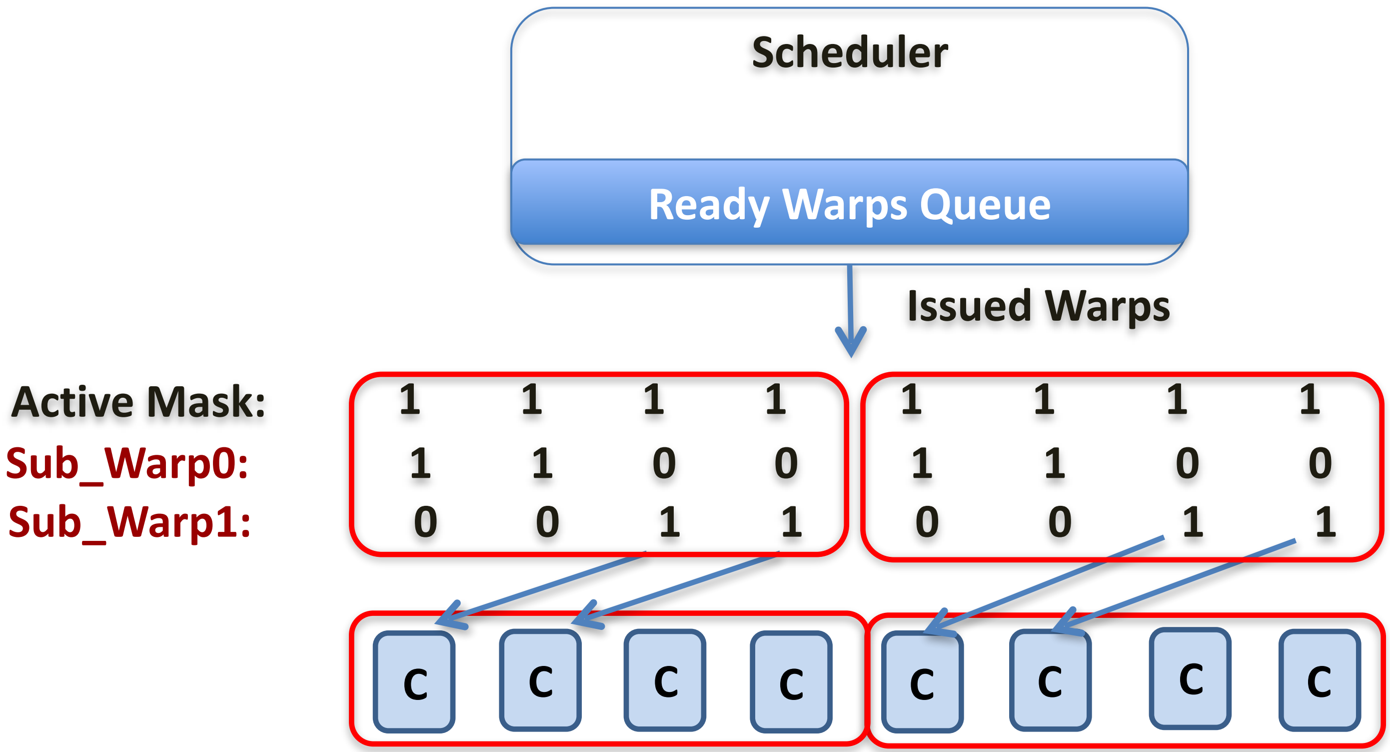
Folding Granularity



Folding Granularity

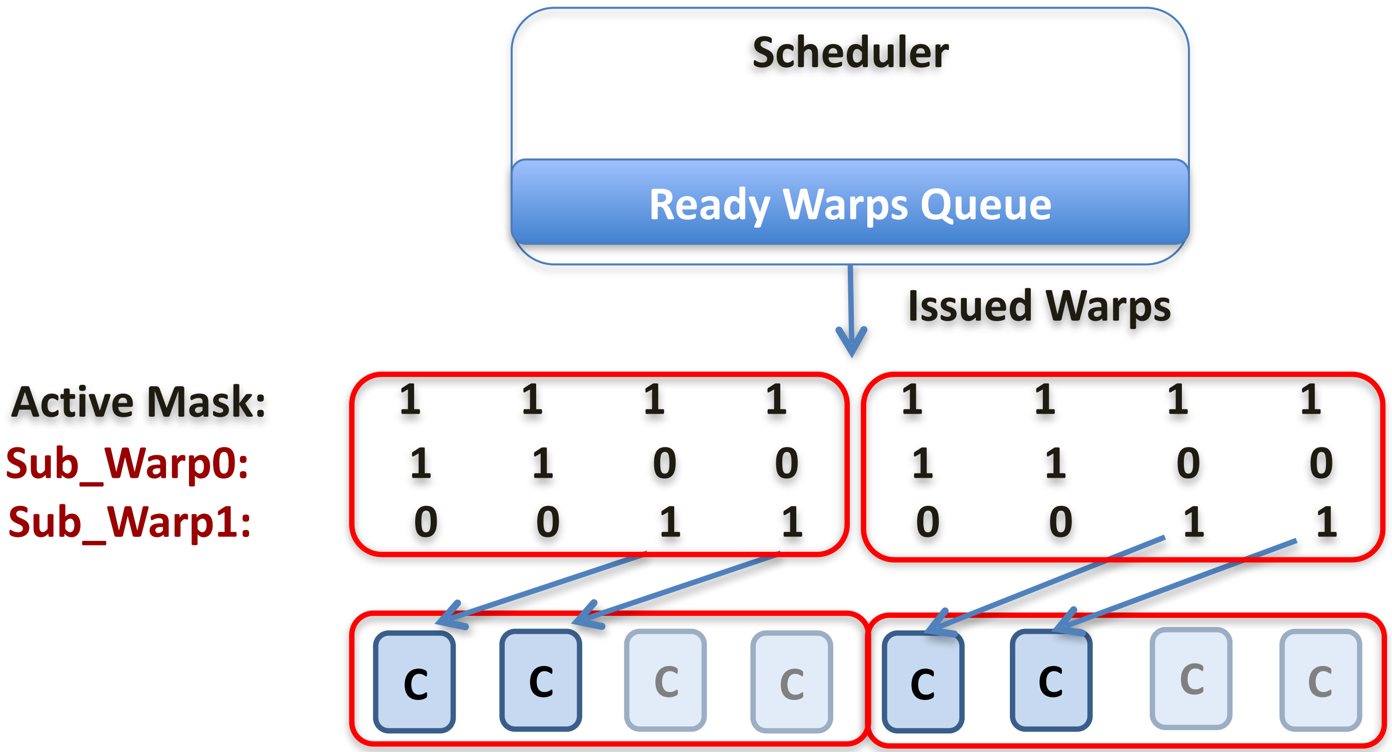


Folding Granularity



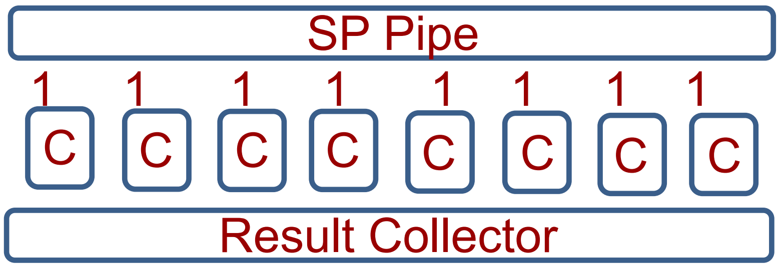
+ Simple
+ Low wiring overhead
+ Small delay
+Support for lane shuffling

Folding Granularity

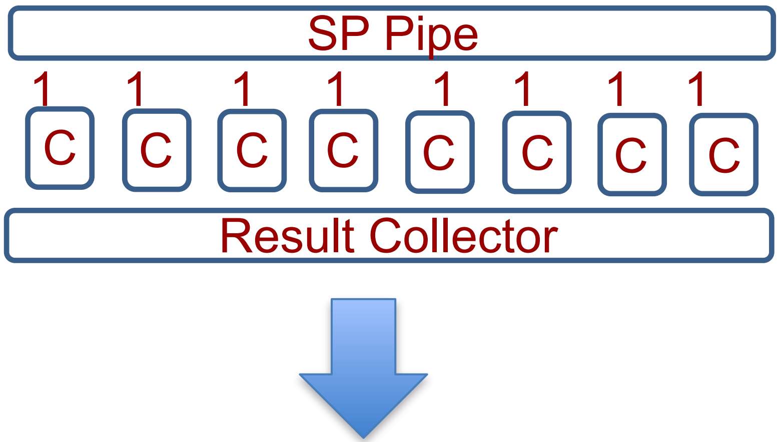


+ Simple
+ Low wiring overhead
+ Small delay
+Support for lane shuffling

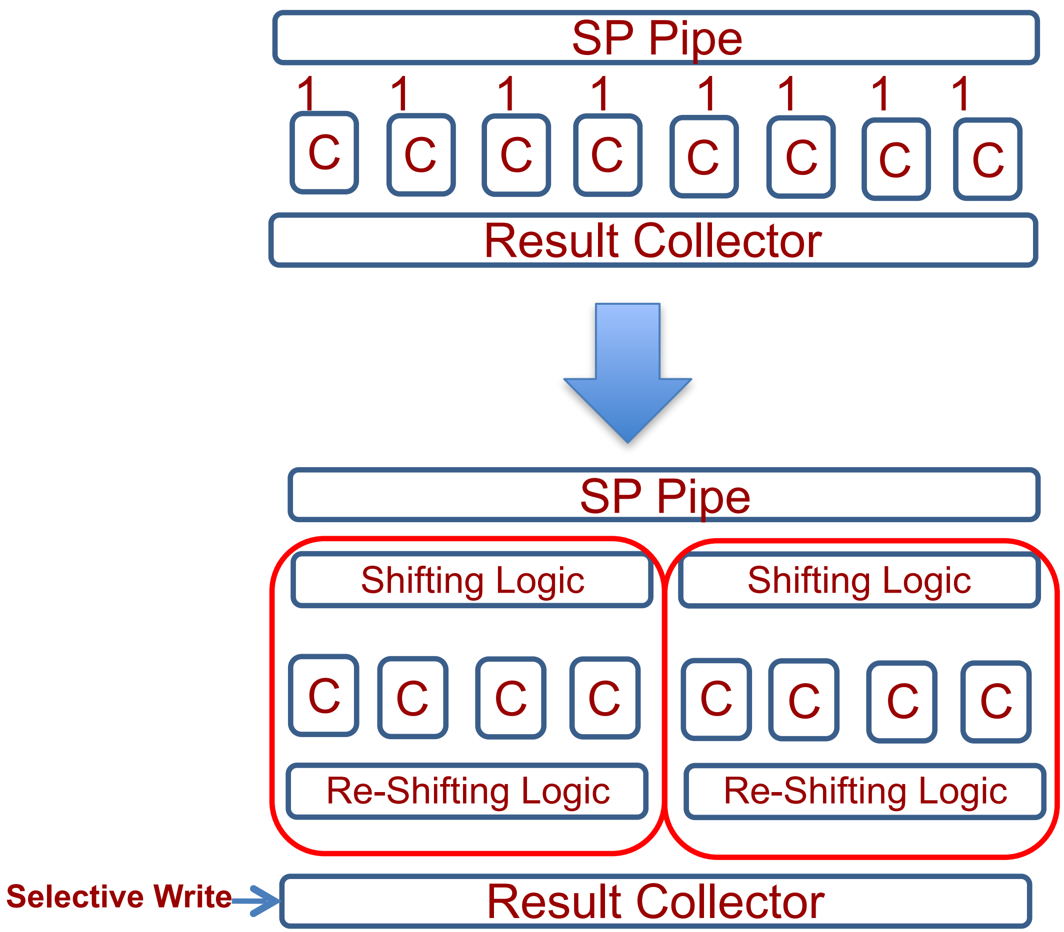
Warp Folding Pipeline



Warp Folding Pipeline



Warp Folding Pipeline



Example

Example

1111 1111

Example

1111 1111

1100 1100

Example

1111 1111

1100 1100

0011 0011

Example

1111 1111

1100 1100

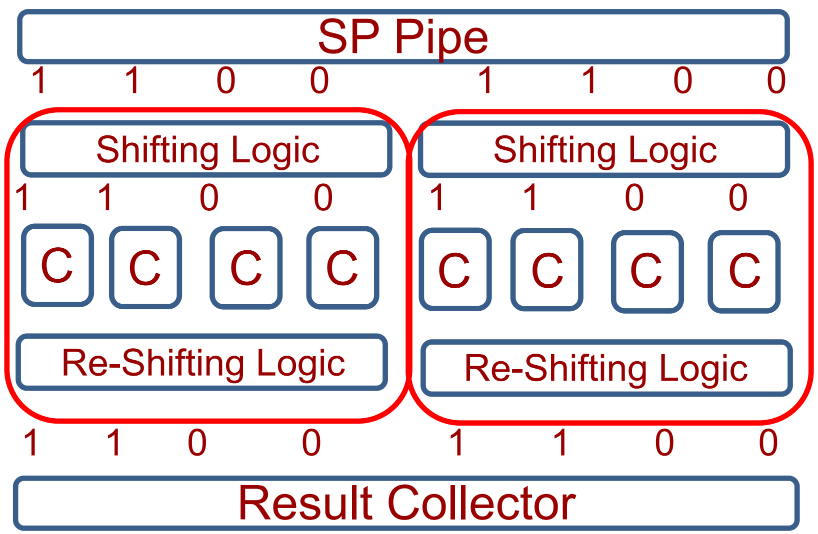
0011 0011

Example

1111 1111

1100 1100

0011 0011

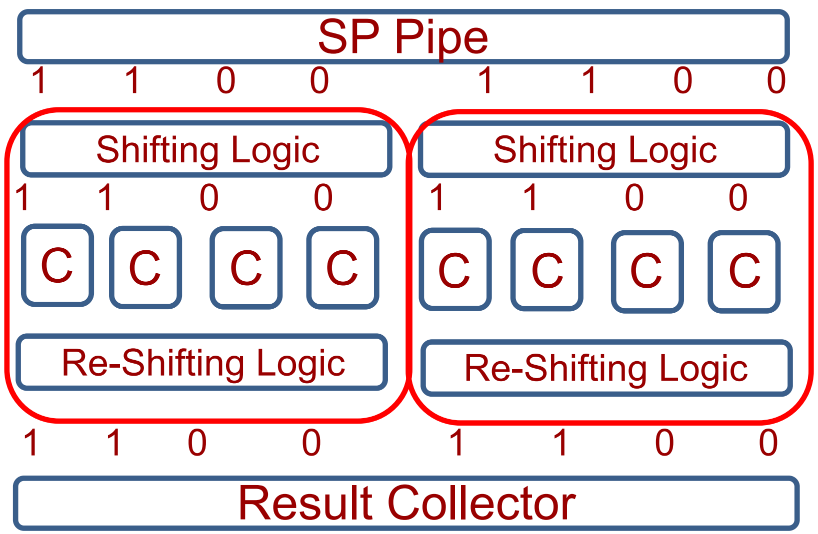


Example

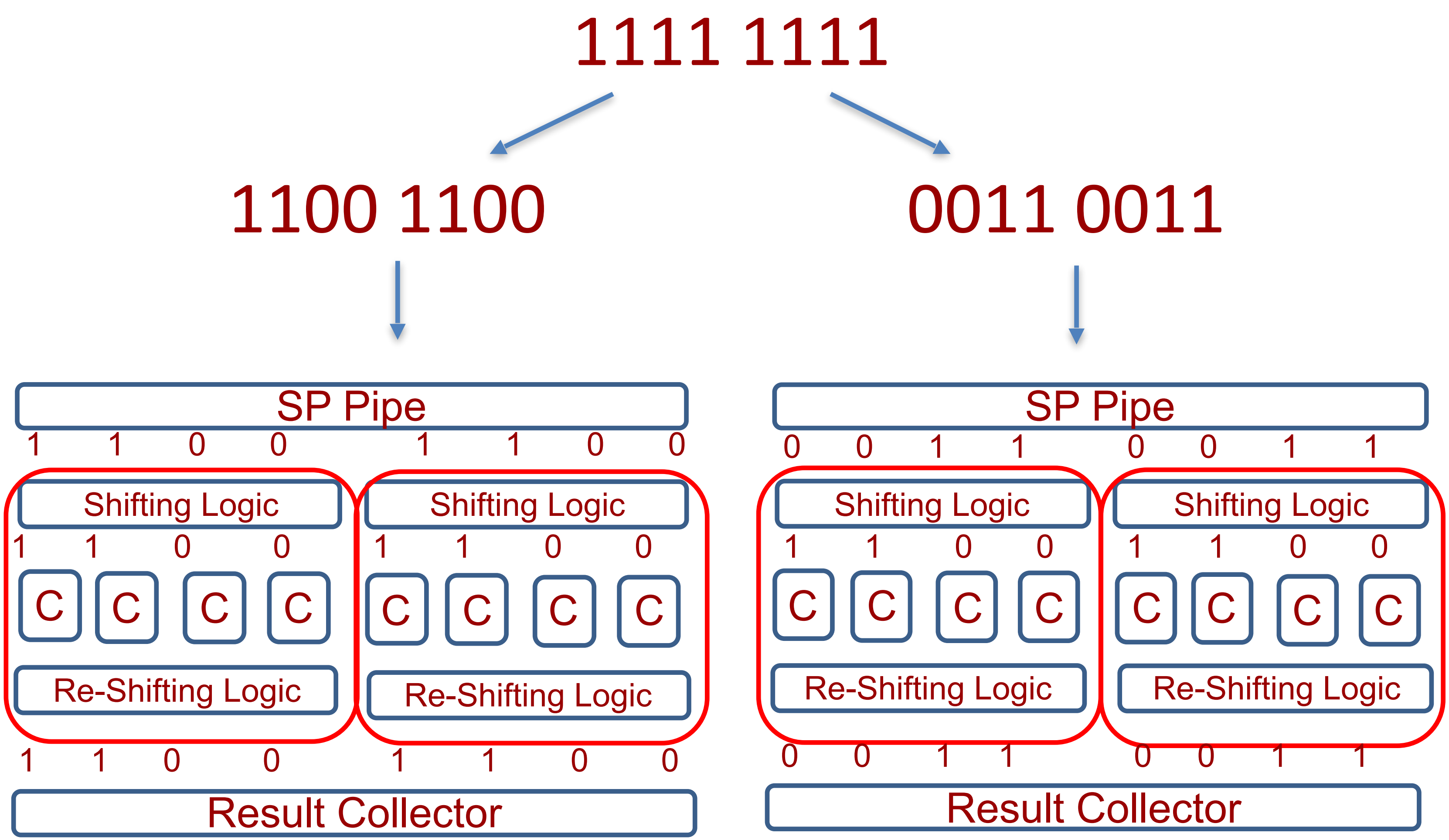
1111 1111

1100 1100

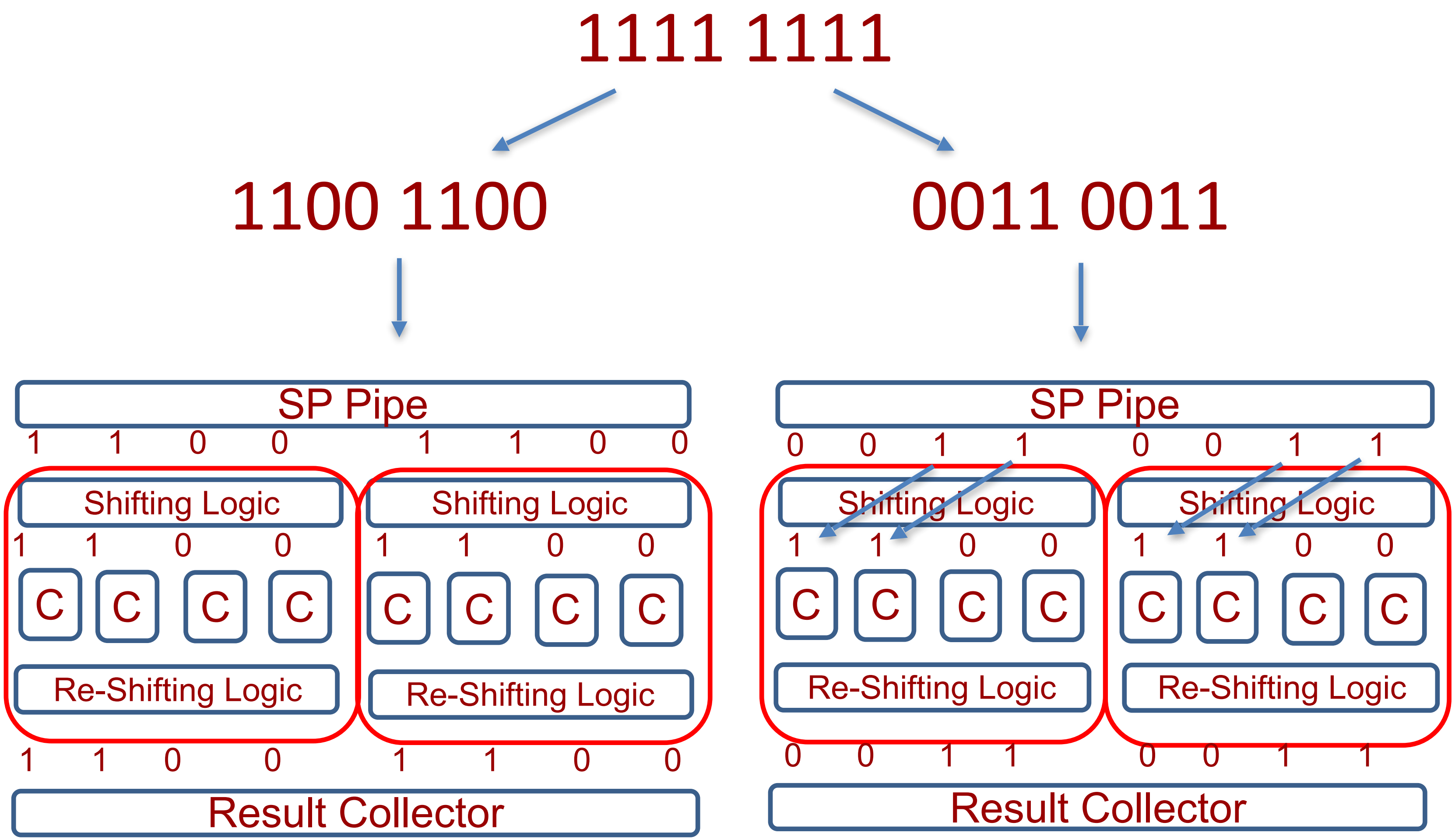
0011 0011



Example

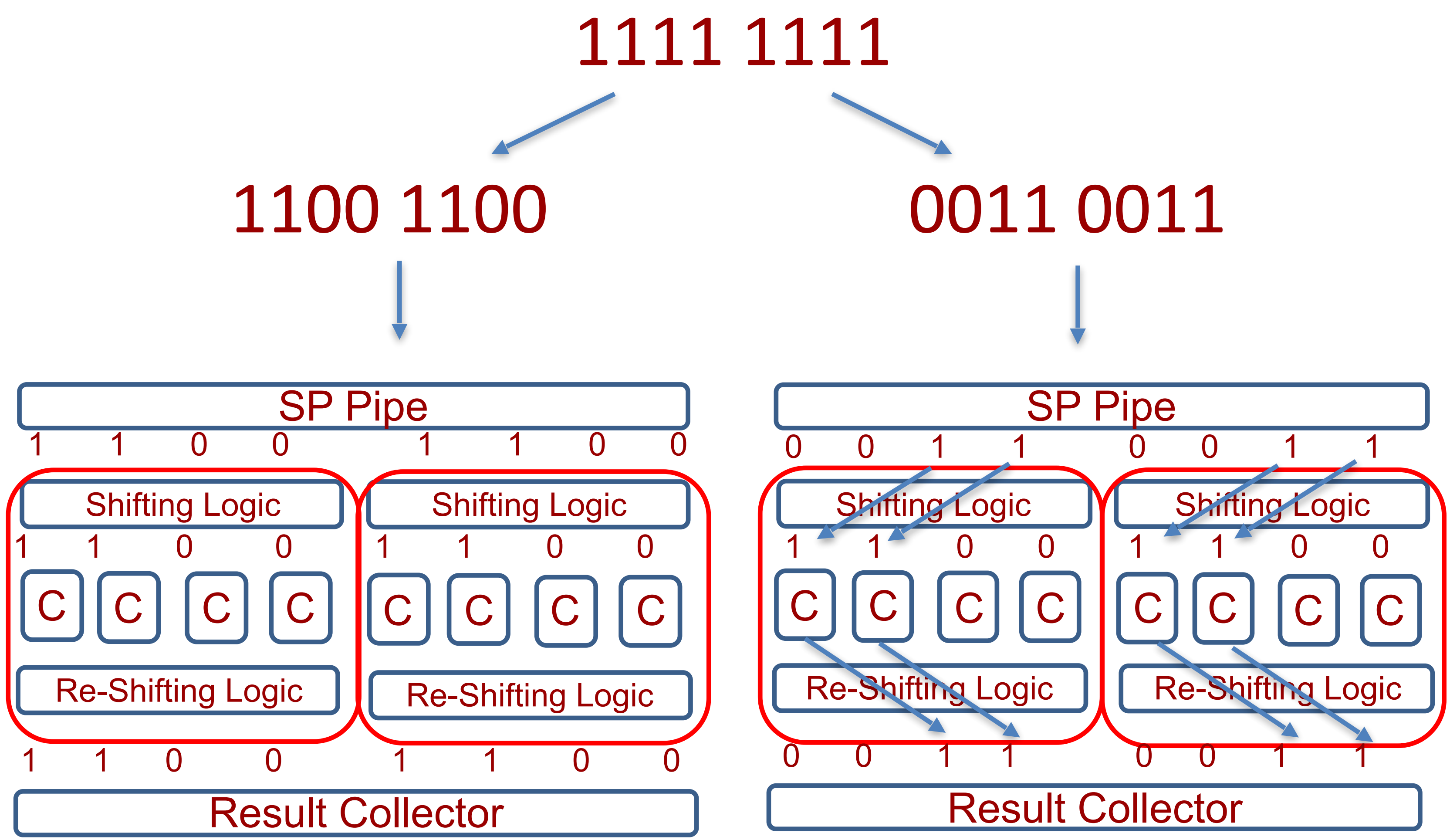

1111 1111
1100 1100
0011 0011
SP Pipe
1 1 0 0 1 1 0 0
Shifting Logic
1 1 0 0
C C C C
Re-Shifting Logic
Shifting Logic
1 1 0 0
C C C C
Re-Shifting Logic
1 1 0 0 1 1 0 0
Result Collector
SP Pipe
0 0 1 1 0 0 1 1
Shifting Logic
1 1 0 0
C C C C
Re-Shifting Logic
Shifting Logic
1 1 0 0
C C C C
Re-Shifting Logic
0 0 1 1 0 0 1 1
Result Collector

Example



1111 1111
1100 1100
0011 0011
SP Pipe
1 1 0 0 1 1 0 0
Shifting Logic
Shifting Logic
1 1 0 0 1 1 0 0
C C C C C C C C
Re-Shifting Logic
Re-Shifting Logic
1 1 0 0 1 1 0 0
Result Collector
SP Pipe
0 0 1 1 0 0 1 1
Shifting Logic
Shifting Logic
1 1 0 0 1 1 0 0
C C C C C C C C
Re-Shifting Logic
Re-Shifting Logic
0 0 1 1 0 0 1 1
Result Collector

Example



1111 1111
1100 1100
0011 0011
SP Pipe
1 1 0 0 1 1 0 0
Shifting Logic
Shifting Logic
1 1 0 0
1 1 0 0
C C C C
C C C C
Re-Shifting Logic
Re-Shifting Logic
1 1 0 0 1 1 0 0
Result Collector
SP Pipe
0 0 1 1 0 0 1 1
Shifting Logic
Shifting Logic
1 1 0 0
1 1 0 0
C C C C
C C C C
Re-Shifting Logic
Re-Shifting Logic
0 0 1 1 0 0 1 1
Result Collector

Origami scheduler

- ➢Improve the power gating potential by coalescing warps based on:
  - ➢Threads utilization
  - ➢Instruction type

Origami scheduler

- Group the threads based on their active mask
  - One group will have the active mask with less than 32 threads
  - The other group will have the active masks with 32 active threads

Origami scheduler

- Group the threads based on their active mask
  - One group will have the active mask with less than 32 threads
  - The other group will have the active masks with 32 active threads

Lane#:	0	1	2	3	4	5	6	7
Cycle x:	1	1	0	1	0	1	0	0
Cycle x+1:	0	1	1	1	0	1	0	0
Cycle x+2:	1	1	1	1	1	1	1	1
Cycle x+3:	0	0	1	1	0	1	0	1
Cycle x+4:	0	1	1	1	0	1	1	0
Cycle x+5:	1	1	1	1	1	1	1	1

Origami scheduler

- Group the threads based on their active mask
  - One group will have the active mask with less than 32 threads
  - The other group will have the active masks with 32 active threads

Lane#:	0	1	2	3	4	5	6	7	
Cycle x:	1	1	0	1	0	1	0	0	Less than 32 group
Cycle x+1:	0	1	1	1	0	1	0	0	
Cycle x+2:	0	0	1	1	0	1	0	1	
Cycle x+3:	0	1	1	1	0	1	1	0	
Cycle x+4:	1	1	1	1	1	1	1	1	Equal to 32 group
Cycle x+5:	1	1	1	1	1	1	1	1	

Origami scheduler

- Group the threads based on their active mask
  - One group will have the active mask with less than 32 threads
  - The other group will have the active masks with 32 active threads

Lane#:	0	1	2	3	4	5	6	7	
Cycle x:	1	1	0	1	0	1	0	0	Less than 32 group
Cycle x+1:	0	1	1	1	0	1	0	0	
Cycle x+2:	0	0	1	1	0	1	0	1	
Cycle x+3:	0	1	1	1	0	1	1	0	
Cycle x+4:	1	1	1	1	1	1	1	1	Equal to 32 group
Cycle x+5:	1	1	1	1	1	1	1	1	

Origami scheduler

- Group the threads based on their active mask
  - One group will have the active mask with less than 32 threads
  - The other group will have the active masks with 32 active threads

Lane#:	0	1	2	3	4	5	6	7
Cycle x:	1	1	0	1	0	1	0	0
Cycle x+1:	0	1	1	1	0	1	0	0
Cycle x+2:	0	0	1	1	0	1	0	1
Cycle x+3:	0	1	1	1	0	1	1	0
Cycle x+4:	1	1	1	1	1	1	1	1
Cycle x+5:	1	1	1	1	1	1	1	1

Less than 32 group

Equal to 32 group

Origami scheduler

- Group the threads based on their active mask
  - One group will have the active mask with less than 32 threads
  - The other group will have the active masks with 32

Active masks are not aligned!!!

Lane#:	0	1	2	3	4	5	6	7	
Cycle x:	1	1	0	1	0	1	0	0	
Cycle x+1:	0	1	1	1	0	1	0	0	Less than 32 group
Cycle x+2:	0	0	1	1	0	1	0	1	
Cycle x+3:	0	1	1	1	0	1	1	0	
Cycle x+4:	1	1	1	1	1	1	1	1	Equal to 32 group
Cycle x+5:	1	1	1	1	1	1	1	1	

Lane Shifting

- Shift the threads to the lower order SIMT lanes
  - Done at the cluster level to reduce overhead

Lane#:	0	1	2	3	4	5	6	7
Cycle x:	1	1	0	1	0	1	0	0
Cycle x+1:	0	1	1	1	0	1	0	0
Cycle x+2:	0	0	1	1	0	1	0	1
Cycle x+3:	0	1	1	1	0	1	1	0
Cycle x+4:	1	1	1	1	1	1	1	1
Cycle x+5:	1	1	1	1	1	1	1	1

Less than 32 group

Equal to 32 group

Lane Shifting

- Shift the threads to the lower order SIMT lanes
  - Done at the cluster level to reduce overhead

Lane#:	0	1	2	3	4	5	6	7
Cycle x:	1	1	1	0	1	0	0	0
Cycle x+1:	1	1	1	0	1	0	0	0
Cycle x+2:	1	1	0	0	1	1	0	0
Cycle x+3:	1	1	1	0	1	1	0	0
Cycle x+4:	1	1	1	1	1	1	1	1
Cycle x+5:	1	1	1	1	1	1	1	1

Less than 32 group (Cycle x – Cycle x+3)

Equal to 32 group (Cycle x+4 – Cycle x+5)

Origami Scheduling

- Runtime Warp Folding Algorithm
  - Folds warps long enough to guarantee savings

$$N_{phase} = N_{pipelineflush} + N_{idledetect} + N_{breakeventime}$$

  - Adaptive Folding
    - Aggressive folding for warps with lower instruction count
    - Conservative folding for warps with higher instruction count
    - Change folding frequency based on application utilization
    - See paper for more detail!

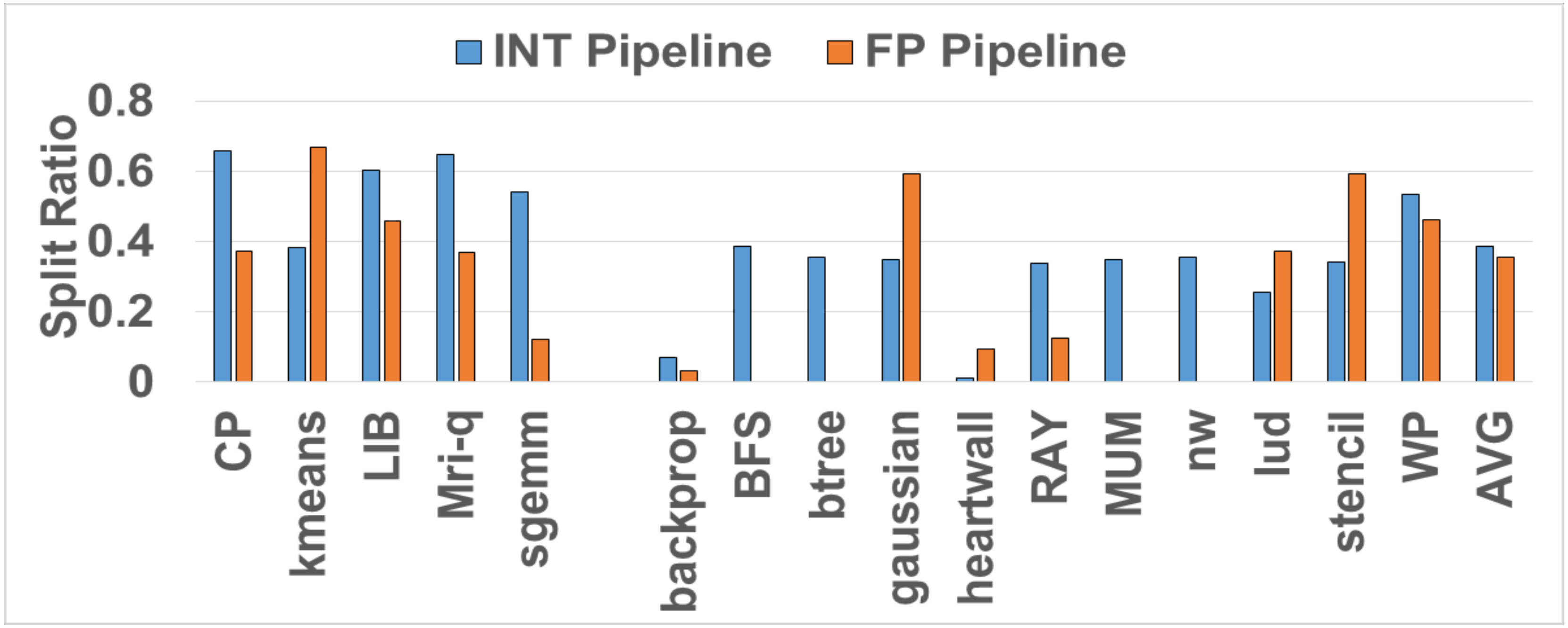
EVALUATION

Evaluation Methodology

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

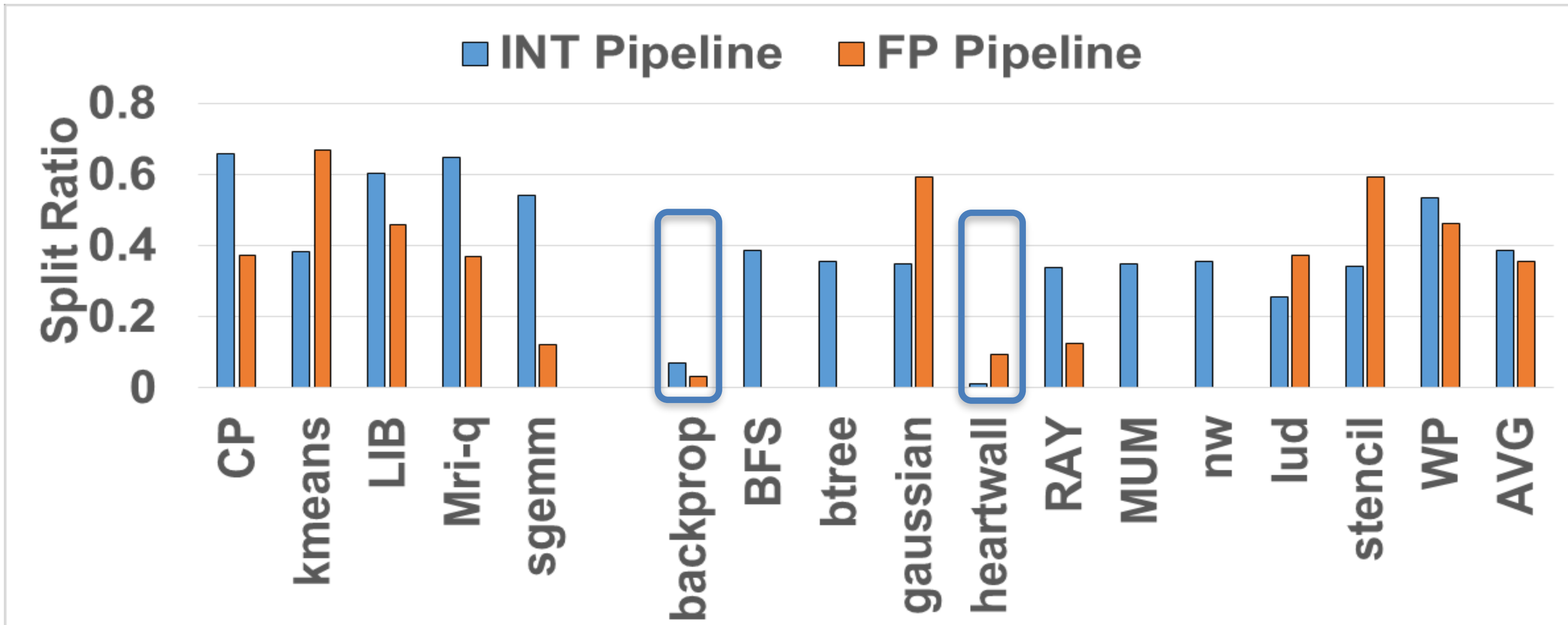
Folding Ratio

- Folding frequency is application dependent



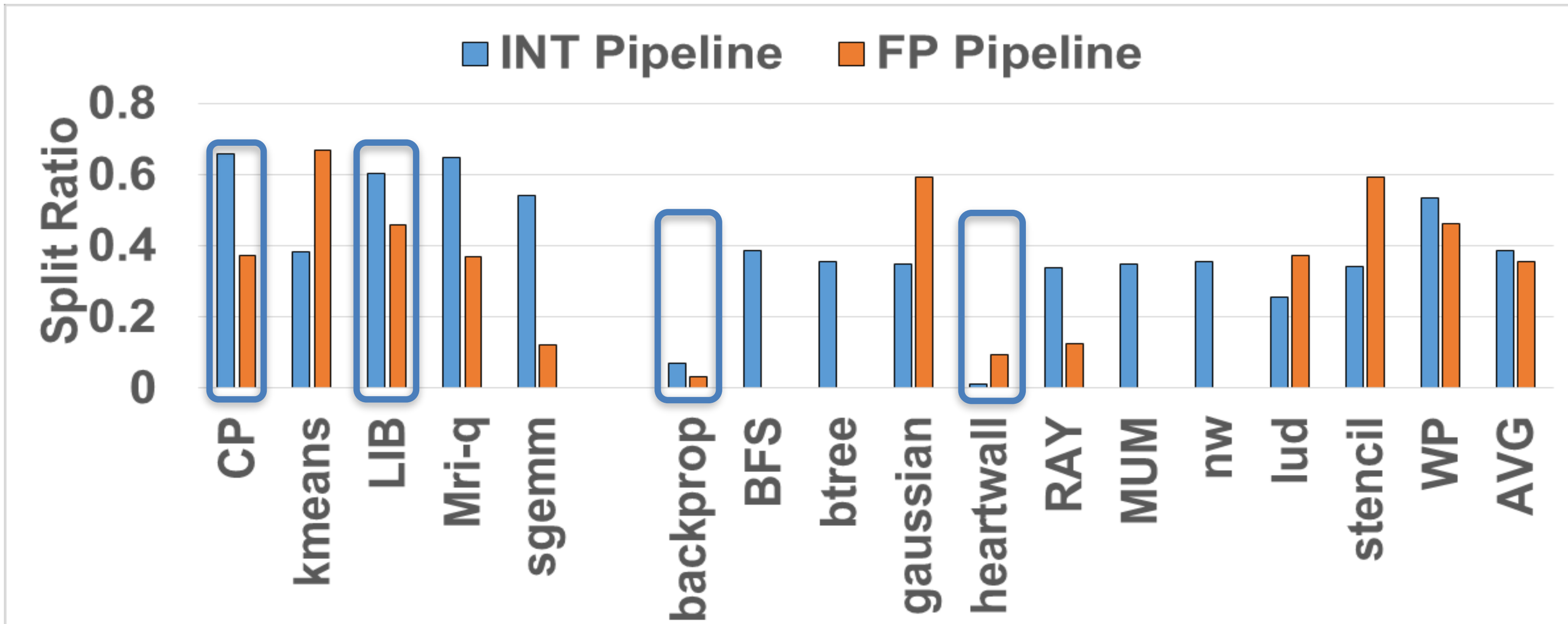
Folding Ratio

- Folding frequency is application dependent



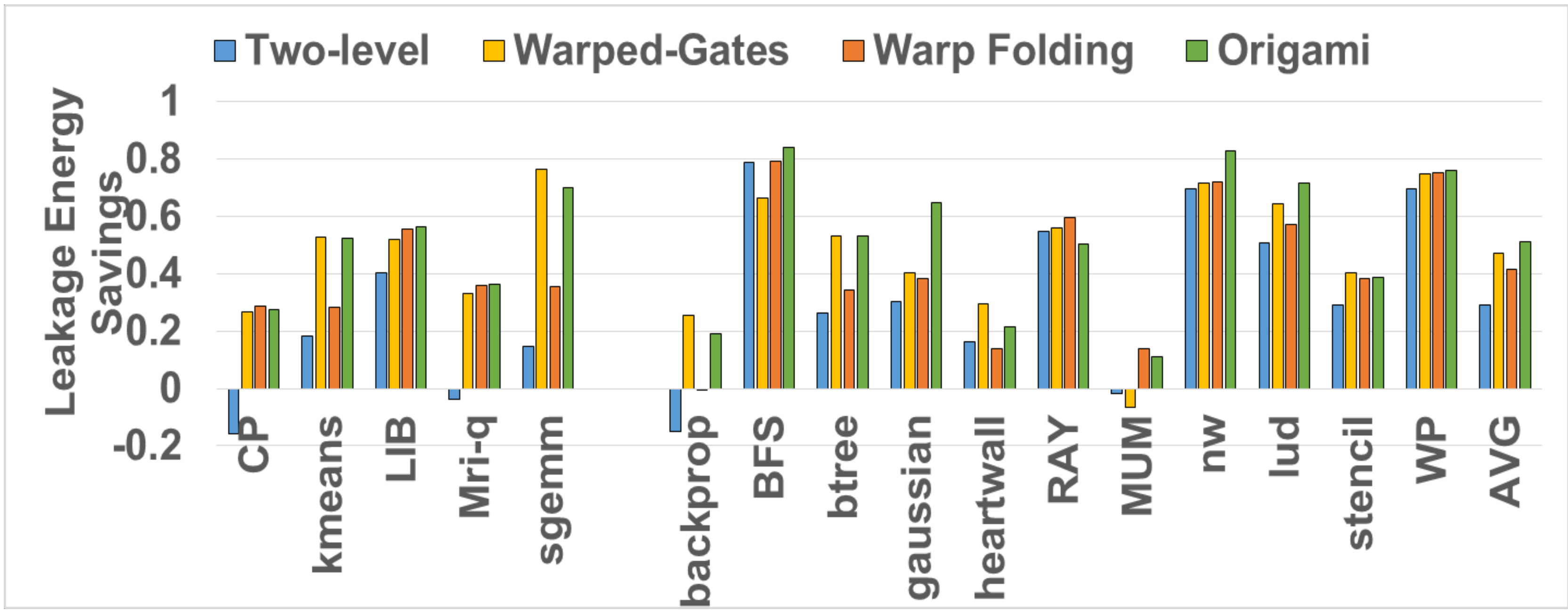
Folding Ratio

- Folding frequency is application dependent



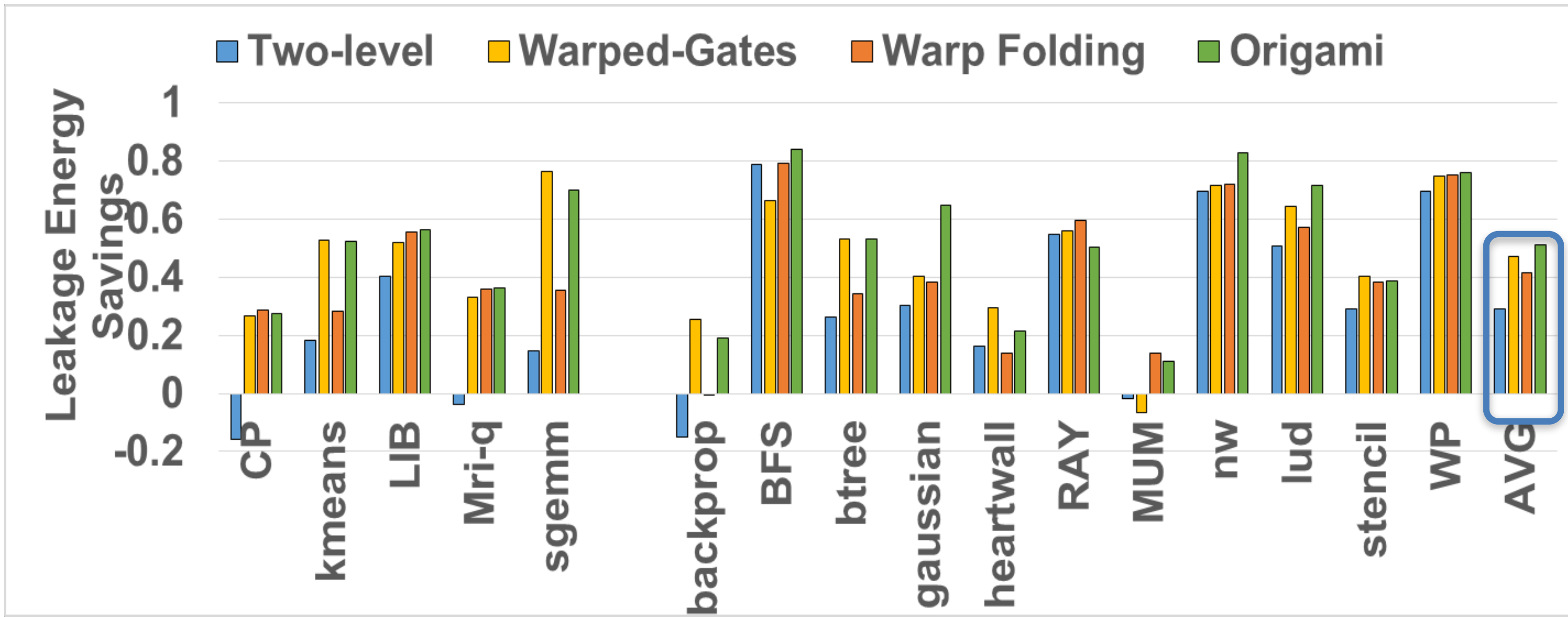
Energy Savings/INT

- Eliminates negative energy savings
- Origami scheduler able to amplify folding benefits
- Origami is able to save 49%



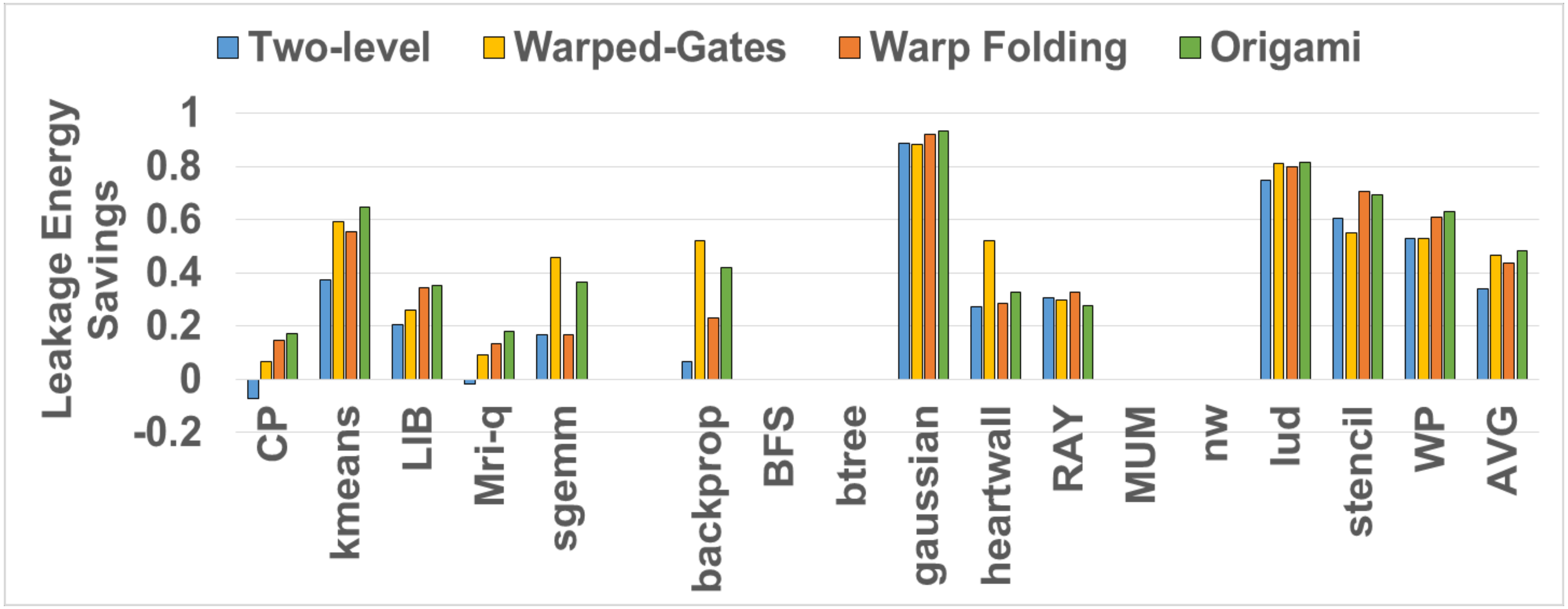
Energy Savings/INT

- Eliminates negative energy savings
- Origami scheduler able to amplify folding benefits
- Origami is able to save 49%



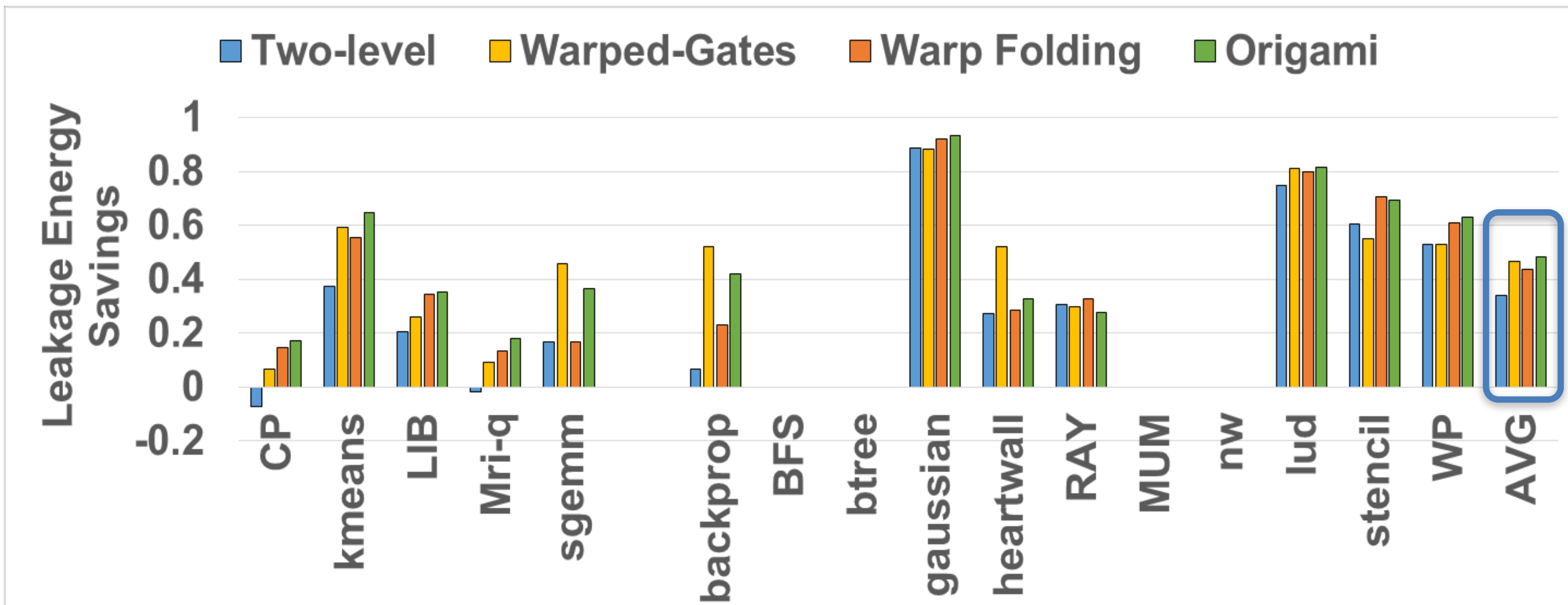
Energy Savings/FP

- Eliminates negative energy savings
- Origami scheduler able to amplify folding benefits
- Origami is able to save 46%



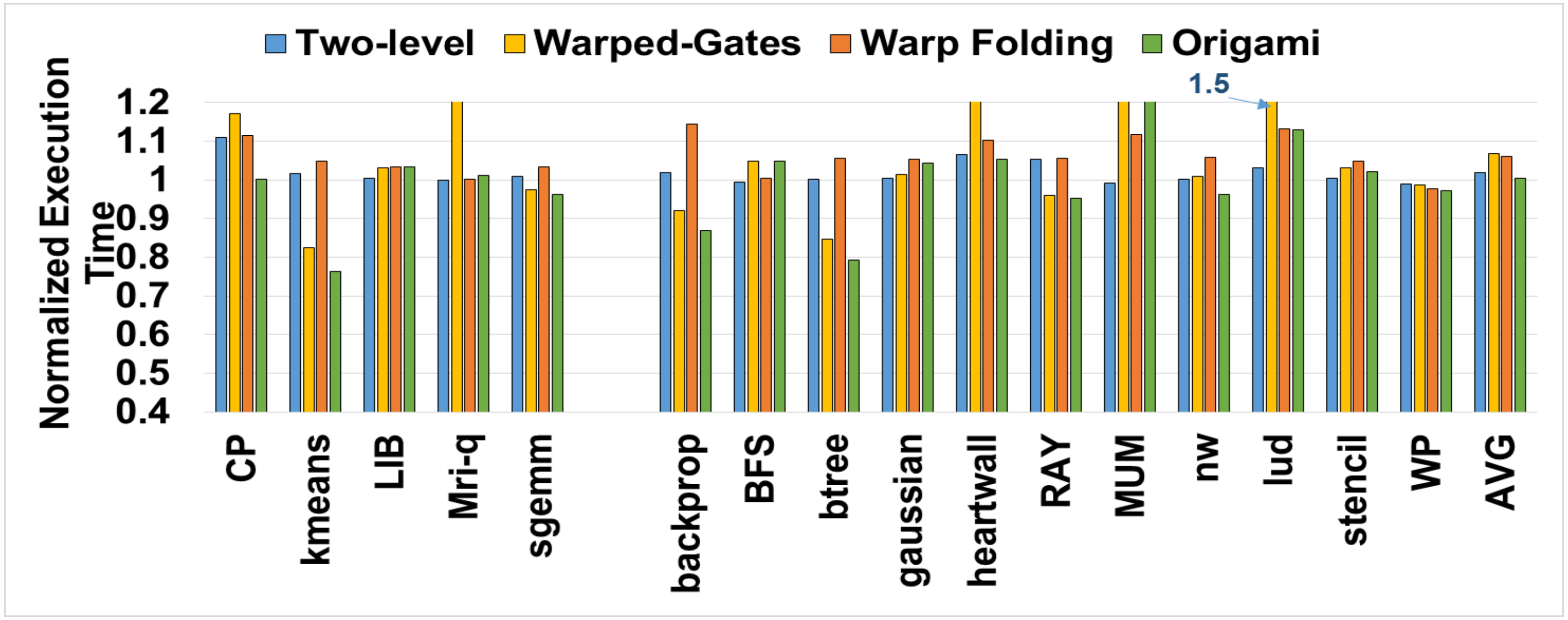
Energy Savings/FP

- Eliminates negative energy savings
- Origami scheduler able to amplify folding benefits
- Origami is able to save 46%



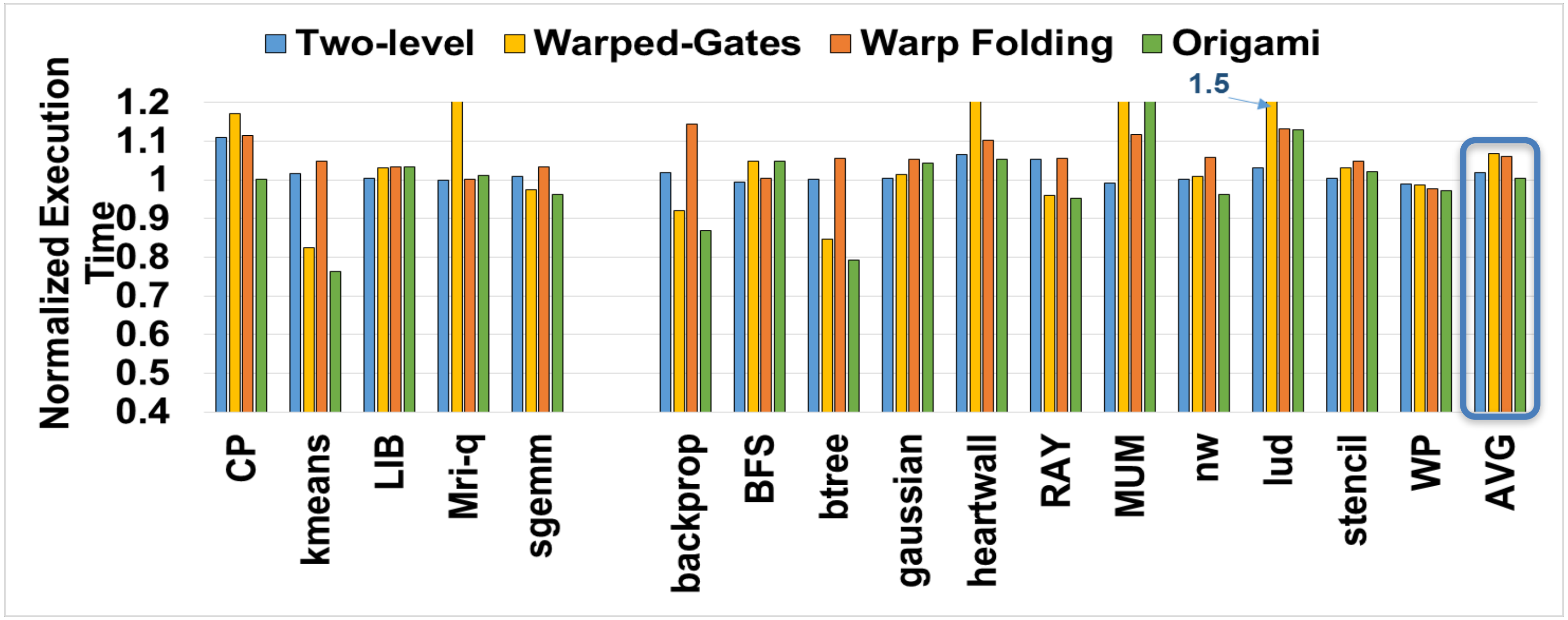
Performance

- Origami is able to reduce the performance overhead significantly over Warped-Gates
- Origami scheduler has positive impact on performance for some workloads



Performance

- Origami is able to reduce the performance overhead significantly over Warped-Gates
- Origami scheduler has positive impact on performance for some workloads



Conclusion

- Execution units energy efficiency is critical
- Take advantage of the spatial and temporal idleness to Improve the power gating potential
- Warp folding
  – Adaptively fold warp to coalesce bubbles
- Origmai scheduler
  – Scheduler warps based on the threads activity and type.
- Able to save 49% and 46% of the execution units leakage energy
- Negligible performance overhead

Questions?

Origami:
Folding Warps for Energy Efficient GPUs

Mohammad Abdel-Majeed, Daniel Wong†, Justin Huang‡ and Murali Annavaram**
abdelmaj@usc.edu, dwong@ece.ucr.edu *annavara@usc.edu*
** University of Southern California*
†University of California, Riverside
‡Stanford University

THANK YOU!