Warped Gating: Gating Aware Scheduling and Power Gating for GPGPUs
Mohammad Abdel-Majeed, Daniel Wong and Murali Annavaram

Introduction

- GPGPU Execution units
  - GPU targets application with thousands of threads.
  - Large number of execution units in the GPGPU.
  - Each unit has an INT and FP pipelines.
  - 32/SM in Fermi and 192/SM in Kepler.

Motivation

- Scheduler greedily issues ready instructions (without considering instruction type)
- On average 16 warps are ready to execute any cycle
  - Good mix of INT and FP instructions are available each cycle
  - INT/FP units turn ON/OFF rather rapidly due to greedy scheduling
  - Power gating needs many consecutive cycles of idleness
  - So no opportunity to power gate
- Power Gating regions
  - A: Detect Idle periods (no Gating)
  - B: Gating overhead is higher than saving (Power gated)
  - C: Cycles spent in this region will Translate into savings

GATES

- Give priority to same instruction type during scheduling
  - Change the scheduling order based on the instruction mix of the benchmark.
- Idle periods are unable to go past break even time
  - Force idleness until break-even period once a unit goes idle and even if an instruction needs that unit
  - Performance Loss?
    - No because one can take advantage of other available resources and instruction mix
- GATES is able to increase the length of idle period but still not long enough to take advantage

Architectural Support

Simulation Setup
- GPGPU-sim cycle accurate simulator.
- Fermi architecture
- 14 cycles BET, 3 cycles wakeup latency, 5 cycles idle detect

Results

- Benchmarks GPGPU-sim simulator
  - 18
  - 1.5x Leakage power Reduction
  - ~0% Area overhead
  - 1% Performance overhead

abdelmaj@usc.edu, wongdani@usc.edu, annavara@usc.edu