Towards Task-Pipelined General Purpose Computing on GPUs
Shuai Mu, Yubei Chen, Dongdong Li, Yangdong Deng, Zhihua Wang
Institute of Microelectronics, Tsinghua University, Beijing, China

Motivation for Task-Pipelined on GPUs
- Many real-world applications, especially those following a stream processing pattern, feature interleaved task-pipelined and data parallelism
- Radar, Sonar, Wireless Base station, Router, Computational Finance...
- Current GPUs could not support task-pipelined pattern directly
  - The kernels are executed serially on most GPU architecture

Enhanced GPU Microarchitecture for Task-Pipelined Processing
- Prototype and Execution
- Dynamic kernel scheduling
  - Allocate GPU processors among multiple kernels for an optimal overall throughput
- Multi-kernels concurrent execution increases performance

Methodology
- Characteristic of task-pipeline application benchmarks

Results
- Performance improvement
  - Normalize IPC's of Fermi stream processing and our approach to those of serialized execution on base-line GPUs
  - Our architecture delivers an average 18% improvement compared to serial execution on average
  - Outperform Fermi by 10% on average
- Off-chip memory traffic reduction
  - Total off-chip memory access of our task-pipelined architecture reduces by 11%
- Little hardware overhead
  - Scheduler needs 1.75KB of SRAM and a 32-bit floating-point processing unit
  - Each cache line in the modified L2 cache needs an extra byte to store kernel ID and stream ID