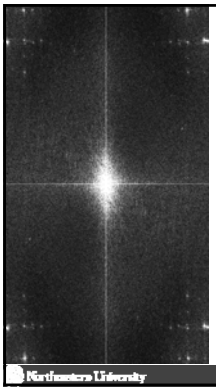


ANALYZING OPTIMIZATION TECHNIQUES FOR POWER EFFICIENCY ON HETEROGENEOUS PLATFORMS

Yash Ukidave and David Kaeli
Department of Electrical and Computer Engineering
Northeastern University, Boston, USA

AsHES 2013
Boston, MA
20th May, 2013

WHAT IS THIS TALK ABOUT ?

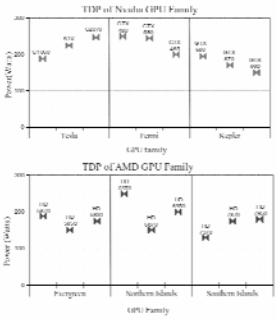
- Study of power and energy profile of different optimization techniques used in heterogeneous applications
- Evaluation of power/performance of such optimization techniques on heterogeneous applications such as FFT

TOPICS

- Applications using OpenCL and power consumption of heterogeneous devices
- Fast Fourier Transforms (FFT) & evaluation methodology
- Optimization techniques used for analysis
- Results for power-performance of FFT implementations
- Analysis of power-performance of different optimization techniques
- Energy profile of different optimization techniques
- Conclusion
- Future work

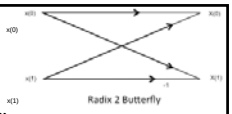
MOTIVATION

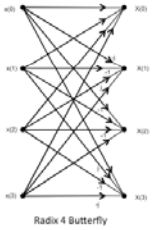
- Increasing use of the CPU-GPU environment to accelerate data-parallel heterogeneous applications
- **Thermal Design Power (TDP)** of latest generation of GPUs used for heterogeneous compute
- Understanding the effects of software design methods contributing to power consumption
- Power and Energy aspect of different optimization techniques for heterogeneous platforms



FAST FOURIER TRANSFORM (FFT)

- FFT is an algorithm to compute Discrete Fourier Transforms (DFT)
  - Reduces time complexity to ***O(nlogn)*** from ***O(n²)***
- FFT classified as Decimation in Time(DIT) or Decimation in Frequency (DIF)
  - **DIT :** Operates on odd and even components of signal
  - **DIF :** Operates on two halves of the signal
- **Butterfly** structure is a major component of FFT of a given Radix
- Each FFT work item performs computes butterfly on given data points





FFT IMPLEMENTATIONS

- **MR-SC FFT :** Multi-Radix single kernel call, based on the FFT implementation in AMD SDK.
- **MR-MC FFT :** Based on Cooley-Tukey algorithm, uses Multiple kernel calls and Multiple Radix combinations for compute
- **Stockham FFT :** Based on the Stockham algorithm for FFT. Single Radix and Single Kernel call computation
- **Apple FFT :** A Multiple Kernel call based FFT provided by Apple Inc. using OpenCL

FFT Implementation	Kernel Calls	Memory Access Patterns	Twiddle factor Computation
MR-SC	Single	Global & Local Memory	Kernel Compile-Time
Stockham FFT		Global Memory	Kernel Run-Time
Apple FFT	Multiple		
MR-MC FFT			

PLATFORMS FOR EVALUATION

- Shared Memory APUs

Device Features	Intel Core i7 3300	AMD Fusion A8 APU
Device Generation	Ivy Bridge	Evergreen
Compute Units(CU)	**16**	**5**
Processing Elements(PE)/CU	4	16
TDP(Watts)	**70**	**100**
Memory Bandwidth(GB/s)	26	17
Register File Per CU(KB)	64	64

- Discrete GPUs

Device Features	Nvidia GTX 680	AMD Radeon HD 7770
Device Generation	Kepler	Southern Islands
Compute Units(CU)	**8**	**10**
Processing Elements(PE)/CU	192	64
TDP(Watts)	**195**	**80**
Memory Bandwidth(GB/s)	192	72
Register File Per CU(KB)	256	256

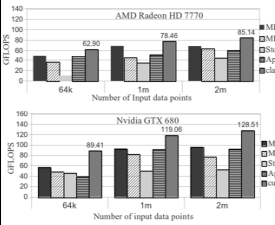
OPTIMIZATION TECHNIQUES CLASSIFIED IN SETS

- Set S0
  - No modifications
  - (*"out-of-box")*
  - *performance*
- Set S1
  - Coalesced Global Memory Accesses
  - Loop Unrolling
- Set S2
  - Data Transformation
  - *float → float2*
  - *float → float4*
  - *float → float8*
- Set S3
  - Local Memory Usage
  - Stage overlapping for stage-based compute

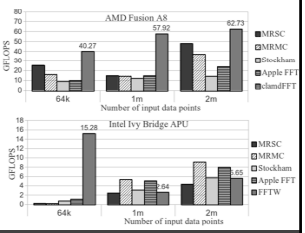
EXECUTION PERFORMANCE OF FFT

- Performance is compared to the baseline on each platform
- Evaluation is done for three data set sizes 64K, 1M and 2M data points

Execution Performance on GPUs



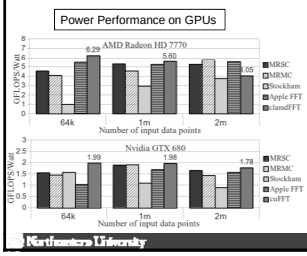
Execution Performance on APUs



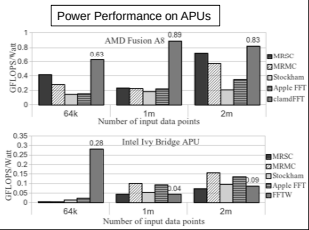
POWER PERFORMANCE OF NON-OPTIMIZED FFT

- High power consumption of Nvidia GPUs affect their power efficiency over AMD GPUs
- Low compute capability of APUs exhibit decrease in performance over GPUs

Power Performance on GPUs

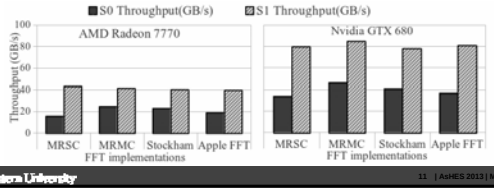


Power Performance on APUs



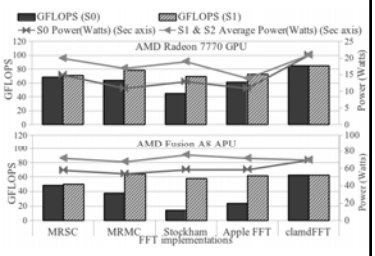
ANALYSIS OF MEMORY BASED OPTIMIZATIONS

- S1 & S2 optimizations use coalesced memory accesses in FFT kernels
- FFTs are modified to perform coalesced memory accesses and loop structures are unrolled
- Improved throughput on devices leads to increase in performance
- Average improvement in throughput is 2X on discrete GPUs



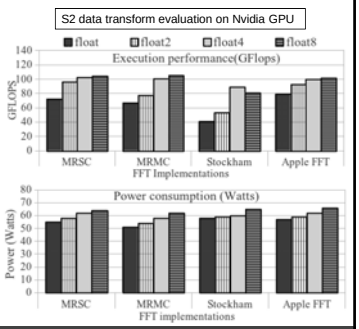
ANALYSIS OF MEMORY BASED OPTIMIZATIONS

- Improved throughput causes increase in execution performance
- Coalesced accesses are handled using specialized hardware paths
- Coalesced accesses increase power consumption
- Average increase in power is 32%



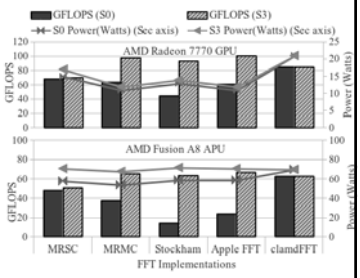
ANALYSIS OF MEMORY BASED OPTIMIZATIONS

- S2 optimizations use data transformation
- Data transforms in OpenCL allow contiguous memory access
- This increases coalesced accesses for the kernel
- Power consumption increases due to coalesced access
- Per-workitem compute increases due to increase in input data access



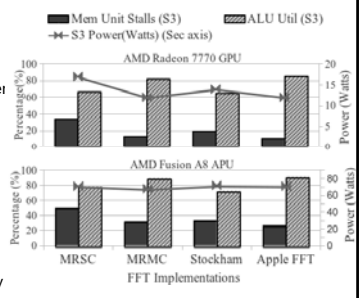
EFFECTS OF LOCAL MEMORY USAGE AND STAGE-OVERLAPPING

- S3 optimizations use local memory and stage-overlapping
- Stages of multi-kernel implementation are overlapped in single kernel call
- MRMC and Apple FFT are modified for stage overlap
- Kernel call overhead is avoided and coalesced access are also used
- Large performance gains are observed on GPUs and APUs



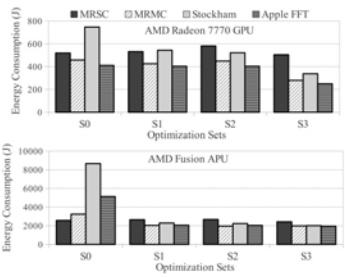
ARCHITECTURAL FACTORS CONTRIBUTING TO POWER

- S3 optimizations are evaluated for this analysis
- Memory stalls and ALU utilization is evaluated for power consumption
- Memory unit stalls directly affects power consumption of GPUs and APUs
- Number of in-flight memory accesses can cause memory-unit stalls
- ALU utilization does not directly affect power consumption



ENERGY PROFILE OF OPTIMIZATION TECHNIQUES

- Energy relates power consumption and execution performance directly
- Power consumption can increase due to increase in performance
- Energy consumption exposes this trade-off
- Variation in energy across optimizations is 11% on average
- S2 optimizations show 13% increase in energy over other optimization sets



RESULTS SUMMARY

GPUs	APUs
• S1 & S2 improve performance efficiency with cost for power consumption • Local memory in S3 improve power-performance • S3 are compute and power efficient	• S1 & S2 are not power efficient • Local memory improves compute efficiency and power efficiency • Stage overlapping increases load on resources and increases power consumption

RESULTS SUMMARY

✓ Highly Efficient	× Less Efficient	ℙ Moderately Efficient
More than 40% improvement	Less than 10% improvement	10-40% improvement

Optimization Techniques	Power Efficiency		Performance Efficiency		Power Performance (Gflops/Watts)	
	GPUs	APUs	GPUs	APUs	GPUs	APUs
Coalesced Memory Access	×	×	✓	✓	ℙ	ℙ
Loop Unrolling	×	×	✓	✓	ℙ	ℙ
Data Transformation	×	×	✓	✓	×	×
Local Memory Usage	✓	✓	✓	✓	✓	✓
Stage Overlapping	✓	ℙ	✓	✓	✓	ℙ

CONCLUSION

- Analyzed different optimization techniques for their power-performance on once class of applications
- Study helps developer identify potential of power-aware kernel development
- Optimizations related to coalesced memory accesses exhibit increase in power consumption on GPUs and APUs
- Local Memory utilization is observed as the most power efficient optimization technique
- Power increment due to performance improvement is captured effectively in the energy profile

FUTURE WORK

- Analyze power consumption on SoC (System-on-Chip) devices with GPUs, such as ***TI OMAP4, Samsung Exynos, Qualcomm Snapdragon***
- Power-performance analysis to multi-GPU environments such as clusters
- Study of microarchitectural features responsible for power consumption on GPUs
- Extend study to different heterogeneous multi-core devices such as ***Adapteva Ephiphany Processor***, ***Tilera TilePro processors***

THANK YOU !

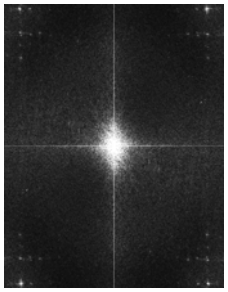
QUESTIONS ? COMMENTS ?

Yash Ukidave
yukidave@ece.neu.edu

EXTRA SLIDES

TITLE PAGE PICTURE

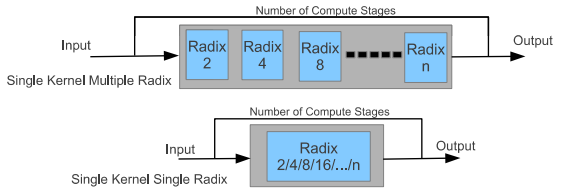
FFT



Source: http://commons.wikimedia.org/wiki/File:Mona_Lisa_bw_square.jpeg

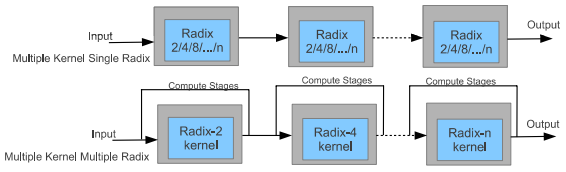
FFT IMPLEMENTATIONS WITH SINGLE KERNEL CALL

- **MR-SC FFT :** FFT implementation based on AMD SDK. Multiple Radix sizes used
- **Stockham FFT :** Stockham algorithm for FFT computation. Single Radix computation



FFT IMPLEMENTATIONS WITH MULTIPLE KERNEL CALLS

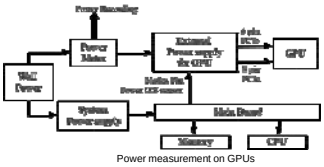
- **MR-MC FFT :** Multiple kernel calls and Multiple Radix combinations for compute
- **Apple FFT :** Multiple Kernel call based FFT by Apple Inc. using OpenCL



POWER MEASUREMENT SETUP

Power measurement on GPU:

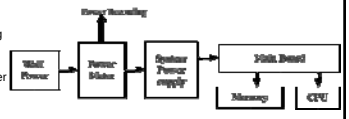
- Discrete GPUs were isolated from the system for power measurement
- A dedicated external Power supply is used for Discrete GPUs
- Power consumption of GPU is measured by profiling the External PSU



Power measurement on GPUs

Power Measurement for APU:

- Power supply for APUs is measured using a similar power meter
- System power is measured to record power consumption on APUs
- Graphic processor of APUs cannot be completely isolated off the host CPU device



Power measurement on APUs