

AMD APU Processors

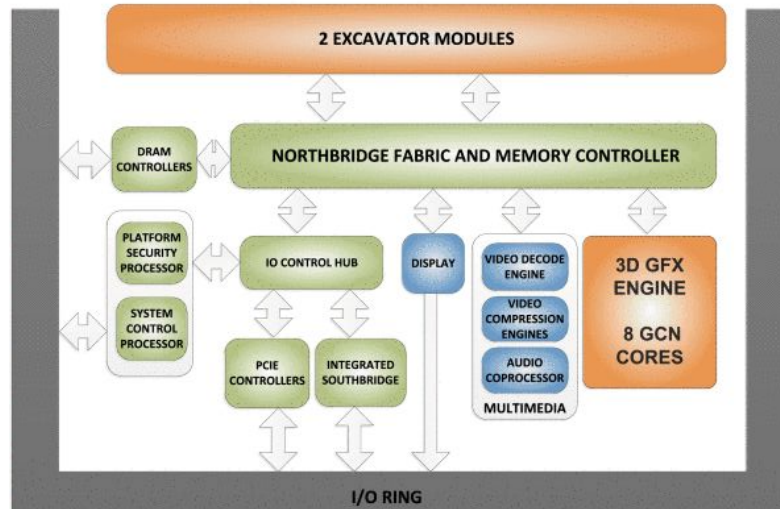
By Gregory Tyree and William Fanelli

Introduction to Heterogeneous Computing

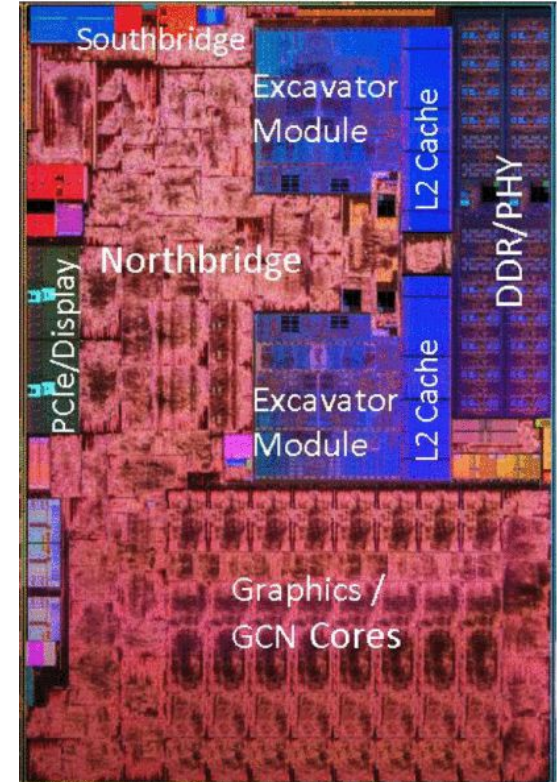
- Previously...
 - Computers commonly consist of a handful of standard cores on a processor
 - Graphics processing cores are included off board across a PCI Bus which causes
 - Communication Delays
 - More hardware necessary for a functioning device
 - Needs power for both the CPU and APU
- Today...
 - Companies like AMD and Intel have started adding graphics cores to processors
 - Less delay between cores
 - Removes the need for added hardware in the average device
 - Allows use of graphics cores for more efficient processing
 - Can help reduce power consumption in devices

Carrizo(2015) APU Architecture

- 6th Generation AMD APU Processor
- 2 Excavator Cores
- 8 Graphics Cores



Used in Mobile Devices



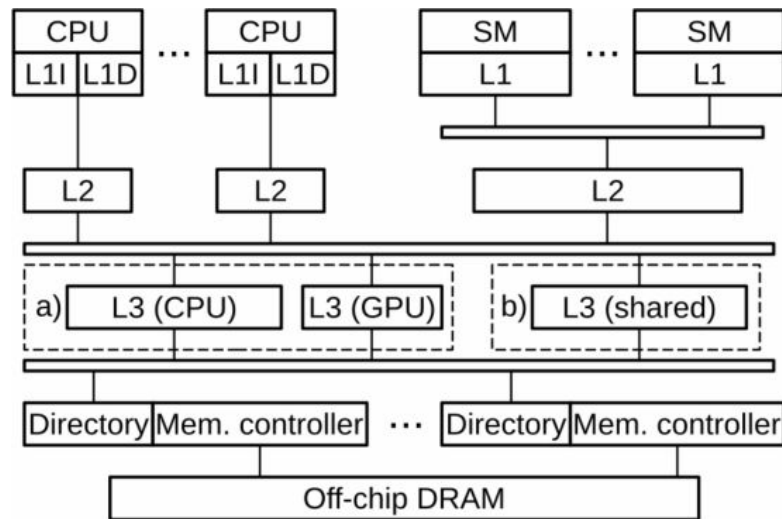
Advantages in Hardware

- Shared Cache
- Decreased Power Consumption
- Improved Performance
- New Coding opportunities

Cache Sharing

- Unified virtual address space
- Shared access to complex data structures
- No need for explicit memory management when programming
- Seamless Fine Grained Synchronization
 - Breaking down algorithms
 - Choose optimal hardware for individual “grains”
 - No PCIe delays
- Lower Level Cache (LLC) Sharing

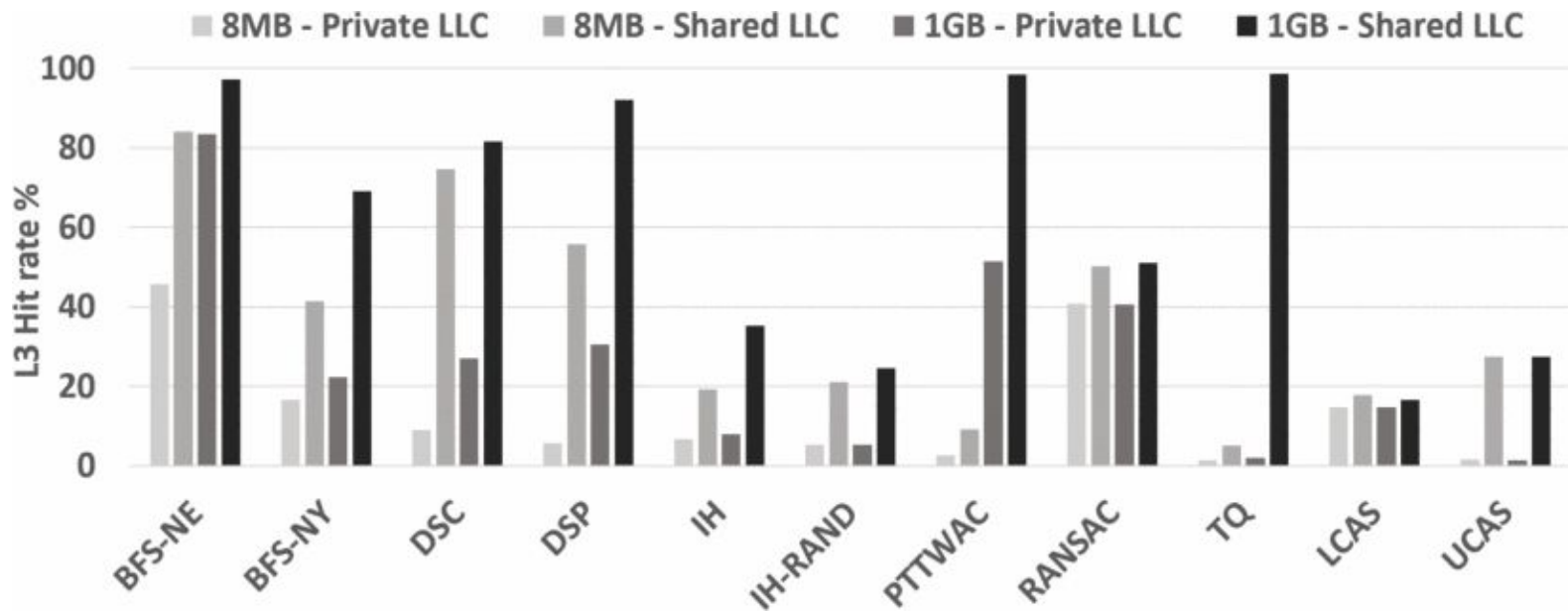
- a) Parted LLC
- b) Shared LLC



Last Level Shared Cache

Paper: Evaluating the effect of last-level cache sharing on integrated GPU-CPU systems with heterogeneous applications

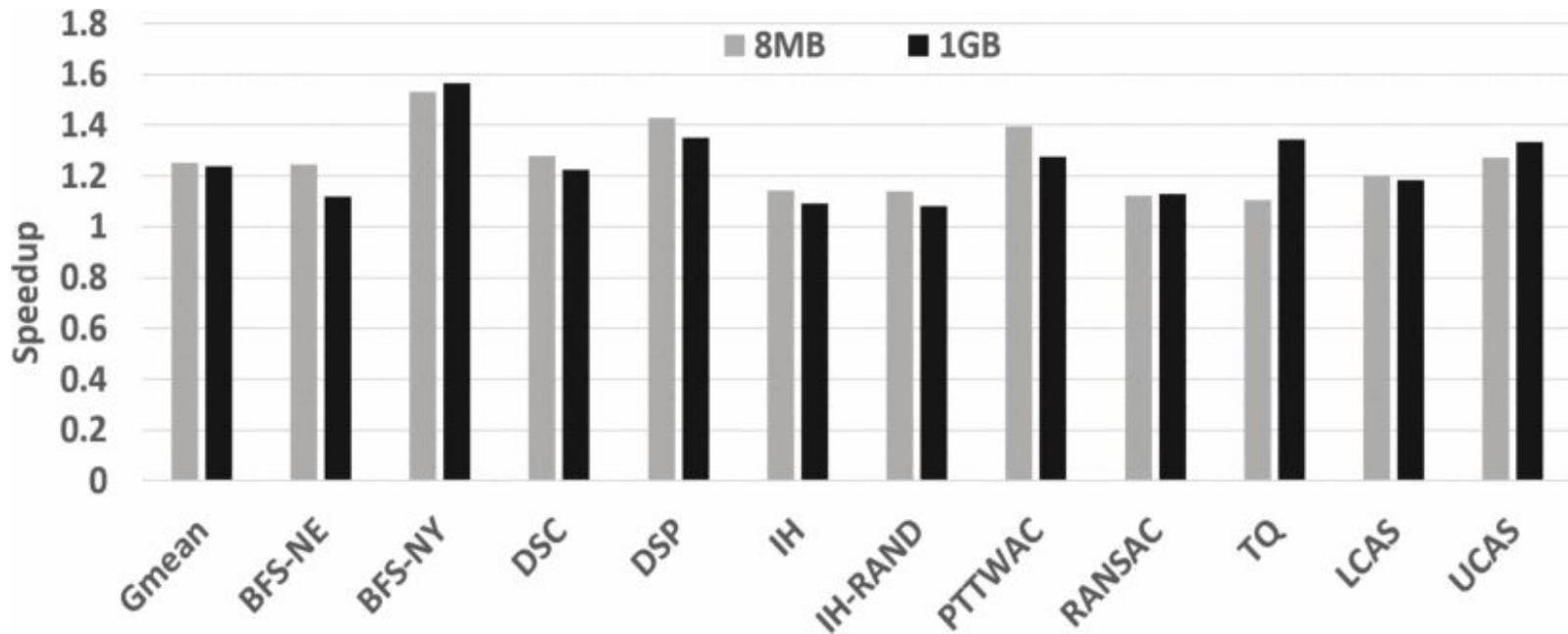
Benchmark	Short Name	Dataset
Backprop	RBP	256K nodes
Breadth-First Search	RBF	256K nodes
Gaussian	RGA	512 × 512 matrix
Hotspot	RHP	512 × 512 data points
LavaMD	RLA	10 boxes per dimension
LUD	RLU	2K × 2K matrix
NN	RNN	1024K data points
NW	RNW	8K × 8K data points
Particlefilter	RPF	10K particles
Pathfinder	RPA	100K × 10K data points
Srad	RSR	512 × 512 data points



Last Level Shared Cache

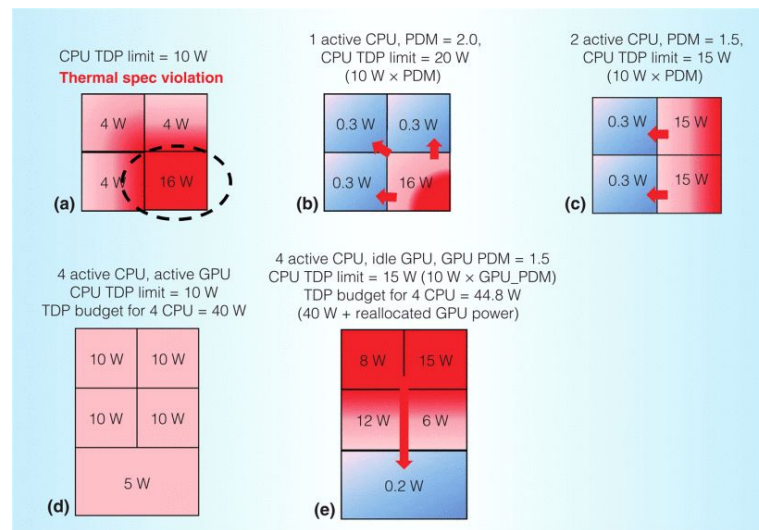
Paper: Evaluating the effect of last-level cache sharing on integrated GPU-CPU systems with heterogeneous applications

Benchmark	Short Name	Dataset
Backprop	RBP	256K nodes
Breadth-First Search	RBF	256K nodes
Gaussian	RGA	512 × 512 matrix
Hotspot	RHP	512 × 512 data points
LavaMD	RLA	10 boxes per dimension
LUD	RLU	2K × 2K matrix
NN	RNN	1024K data points
NW	RNW	8K × 8K data points
Particlefilter	RPF	10K particles
Pathfinder	RPA	100K × 10K data points
Srad	RSR	512 × 512 data points



Power Consumption

- One of the major advantages of the AMD APU architecture is the ability to do more with less power and less space.
- Tactics
 - Single die construction
 - Power and clock gating
 - Clock and voltage scaling
 - Precise power measurement
 - Unified memory space with configurable latency
- Results
 - Advanced power management is achievable through these techniques



Performance Benefits

- Understanding what operation should be run on what device
 - Example:
 - Integer add is 10x faster in CPU(1 cycle) then GPU(4 cycles)
 - GPU is optimized for memory bandwidth
 - In general the CPU is better for low latency tasks while the GPU is better for large data tasks
- Memory benefits of the APU architecture
 - Single combined memory space
 - Zero Copy feature
 - Due to single memory space copy operations can be performed by remapping
 - Uncached Speculative Write Combine
 - Very fast CPU streamed writes are paired with fast GPU reads mitigate the slow CPU read speeds and always provide the fastest path to memory

Utilization of APU benefits

- Benefits of APU are accessible through the AMD app SDK 3.0
- The OpenCL(compute library) exposes the features of the APU to the user through a few settings that allow full utilization if the hardware
- There are available C++ examples that show basic features, as well as more advanced features such as a custom accelerated OpenCV library
- The OpenCL is written for both 32 and 64 bit Windows and Linux systems

Current uses...

- Currently used in both Playstation 4 and Xbox One consoles
- Cheaper devices with improved graphics/processing abilities
- Applications in both mobile and desktop computing
- Used in conjunction with openCL to develop heterogeneous computing

Conclusions

- Advantages of using an APU
 - Physical size and power requirements
 - Tight battery and heat driven power budgets
 - More optimized task handling
 - Shared memory and cache spaces
 - Accelerated library and general use case support
 - Seamless sharing of data Structures
 - Single Die, less physical space required

The APU concept has paved the way for Heterogeneous Computing