Architecture of the Hexagon™ 680 DSP for Mobile Imaging and Computer Vision
Hexagon DSPs in Snapdragon™ 820

COMPUTING ENGINES
- CPU Complex
- ADRENO™ GPU (Games, GUI, GPGPU Compute)
- COMPUTE DSP
  - HVX (Vector Extensions Coprocessor)
  - (Low Power Compute, Audio, Voice, Image/Video Processing, Computer Vision)

MULTIMEDIA ENGINES
- Video
  - (Playback, Broadcast, Streaming, Camcorder, Video Conference)
- Imaging
  - (ISP, JPEG, Image Quality Enhancement)
- Display
  - (Display Composition, Wireless Display, Picture Quality Improvement)
- System Security
  - (Snapdragon StudioAccess™ content protection, Security for Enterprise and BYOD)
- Memory
  - (Caches, custom MMUs, LP/PCDDR controller)

CELLULAR
- (Global LTE & Multimode, Global Carrier Aggregation)

MODEM DSP

LOCATION
- IZat™

Low Power Island
- (Always-on Sensors)

LOW POWER DSP

Snapdragon, Adreno, and IZAT are products of Qualcomm Technologies, Inc.
Hexagon Vector eXtensions (HVX)

- DSP Extensions for Advanced Imaging and Computer Vision
- Achieve Performance / mW substantially better than CPU
Low Light Video & Photos

Algorithm
HVX accelerated local tone mapping and temporal noise reduction to brighten dark videos and photos

Benefits
Adaptively brightening dark areas of video/images
Content Adaptive Detail Enhancement

Original

After HVX Processing
Hexagon Vector eXtensions (HVX)

Domain Specific Architecture

Familiar Programming Model

Tight System Integration
HVX Architecture – SIMD Extensions

- Large SIMD Extensions
  - 1024b SIMD * 4 vector-slot VLIW
  - 4096 result bits / cycle
- 256 8x8 mpy, 64 16x16 mpy
- 32 1024-bit vector registers
- 8/16/32 bit fixed point
- NO floating-point
  - Smaller & Lower Energy Design
  - Algorithmically not needed for majority of CV/Imaging Apps
- Special ISA: Sliding window filters, LUTs, Histograms
- Performance is sufficient for UHD video post-processing, 20Mpix camera burst mode processing … and more

Example shows 1 of 32 lanes of vector-byte-by-scalar multiply reduction

Two such instructions can be done in a packet
HVX Architecture – Threading Model

- 4 Parallel Scalar Threads each with 4-way VLIW and shared L1/L2
  - 500MHz per Thread
  - 2GHz total scalar performance
- 2 HVX Contexts, controllable by any two scalar threads
  - 500MHz per Thread
  - 1GHz total vector performance
- Other 2 threads can do scalar work in parallel
HVX Architecture – Memory

- L2 is the first level memory for the vector units
  - Large primary memory to hold image data reduces tiling overheads seen on small L1
  - Single cycle Load to Use
  - Supports full BW
  - Simplifies programming
- L1/L2 is kept HW coherent
- Streaming prefetch from DDR to L2
- Vector units support variety of Load/Store instructions:
  - Unaligned
  - Per-Byte Conditional
Hexagon Vector eXtensions (HVX)

Domain Specific Architecture

- Wide 1024-bit SIMD (for pixel data parallelism)
- Emphasis on low precision fixed-point + Special ISA
- Parallel and coordinated Scalar & Vector Threads
- Large primary cache for Imaging Working Sets

Familiar Programming Model

Tight System Integration
## DSP with HVX has a CPU-like Programming Model

<table>
<thead>
<tr>
<th>Quad CPU w/ Neon</th>
<th>Hexagon DSP with HVX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Thread Programming w/ cache-based coherent shared memory</td>
<td>Multi-Thread Programming w/ cache-based coherent shared memory</td>
</tr>
<tr>
<td>SIMD Instructions on Vector RF</td>
<td>SIMD Instructions on Vector RF</td>
</tr>
<tr>
<td>Scalar Instructions on Scalar RF</td>
<td>Scalar Instructions on Scalar RF</td>
</tr>
<tr>
<td>Good Control Performance for mixed vector/scalar</td>
<td>Good Control Performance for mixed vector/scalar</td>
</tr>
</tbody>
</table>

**Diagram:**
- **Quad CPU w/ Neon**
  - PC
  - Scalar RF
  - Vector RF
  - Single Shared Memory

- **Hexagon DSP with HVX**
  - PC
  - Scalar RF
  - Vector RF
  - Single Shared Memory
With Key Differences for Performance & Power

<table>
<thead>
<tr>
<th>Quad CPU with Neon</th>
<th>Hexagon DSP with HVX</th>
<th>HVX Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>128 bit SIMD with 1 SIMD pipeline/CPU is common</td>
<td>1024 bit SIMD, 4 Pipelines</td>
<td>8x compute/cycle</td>
</tr>
<tr>
<td>SIMD thread on 32KB L1</td>
<td>SIMD threads share 512KB “L1”</td>
<td>8x more “L1” memory/thread</td>
</tr>
<tr>
<td>Floating-Point in SIMD</td>
<td>Only Fixed-Point in SIMD</td>
<td>Efficient Data Sharing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lower area &amp; power</td>
</tr>
</tbody>
</table>

**Diagram:**
- **Thread 0:** 128bit SIMD, 32KB L1, L2 (1MB+)
- **Thread 1:** 128bit SIMD, 32KB L1, 1024bit SIMD, 1024bit SIMD, 1024bit SIMD, 1024bit SIMD, 512KB
- **Thread 2:** 128bit SIMD, 32KB L1, 1024bit SIMD, 1024bit SIMD, 1024bit SIMD, 1024bit SIMD, 512KB
- **Thread 3:** 128bit SIMD, 32KB L1, 1024bit SIMD, 1024bit SIMD, 1024bit SIMD, 1024bit SIMD, 512KB
Familiar Programming Model

• Conventional Tools & Techniques
  – Shared memory POSIX-like threads (on DSP RTOS)
  – LLVM compiler
  – Program with C/C++ and Intrinsics
  – Pre-Optimized Libraries for common filters & algorithms

• Easy DSP offload from Android
  – DSP code is dynamically loaded on invocation of synchronous Remote Procedure Call
  – Higher Level Android Frameworks Available for
    – Camera Pre-Processing in Real-Time
    – Video Post-Processing in Real-Time
    – Offline Camera Post-Proc and Computer Vision

• On-going Research: Halide Language on HVX
Hexagon Vector eXtensions (HVX)

Domain Specific Architecture
- Wide 1024-bit SIMD (for pixel data parallelism)
- Emphasis on low precision fixed-point + Special ISA
- Parallel and coordinated Scalar & Vector Threads
- Large primary cache for Imaging Working Sets

Familiar Programming Model
- SIMD + Multi-Thread
- Cache based coherent shared memory
- Programming in C/C++ with Intrinsics
- Pre-optimized libraries for common functions

Tight System Integration
System Features – Streaming Interface

- Custom Camera Pre-Processing without DDR access saves power
- Raw data from Image Sensor is streamed into L2 at up to 1.2Gpixels/sec
- Streamer formats / pads / aligns data for convenient vector processing
- Data held in input & output circular buffers
- HVX processed pixels can be streamed out to ISP HW
- Synchronization through memory-mapped control registers
System Features – SMMU

- ARM Compliant SMMU allows for Zero-Copy data sharing with CPU
- Multi-Threaded DSP can be servicing multiple offload sessions (concurrent apps for Audio, Camera, Computer Vision (CV), etc.)
- SMMU supports multiple Context Banks to allow sharing with multiple different address spaces on CPU
- SMMU can be used to support processing on Secure Content managed outside of HLOS
System Features – One Way Coherency

- Bus writes from the DSP snoop-invalidate the CPU caches
  - Avoids expensive SW cache maintenance on CPU, saving time & energy
  - DSP side requires cache maintenance, but this is handled transparently in the RPC software interface layer
System Features – Quality of Service

- DSP services multiple real-time clients (Audio, Camera, CV), each with their own timelines
  - Imaging Algorithms can consume large amounts of external bandwidth and cause congestion for other clients
- L2 partitioning: L2 can be soft partitioned into regions assigned to different threads
- Internal QoS: HW-based prioritization of memory requests; SW assigns priorities to threads
- External QoS: System Level HW algorithms can throttle DSP traffic to ensure system performance
Hexagon Vector eXtensions (HVX)

**Domain Specific Architecture**
- Wide 1024-bit SIMD (for pixel data parallelism)
- Emphasis on low precision fixed-point + Special ISA
- Parallel and coordinated Scalar & Vector Threads
- Large primary cache for Imaging Working Sets

**Familiar Programming Model**
- SIMD + Multi-Thread
- Cache based coherent shared memory
- Programming in C/C++ with Intrinsics
- Pre-optimized libraries for common functions

**Tight System Integration**
- Camera Streaming Interface
- SMMU
- One way Coherency
- System & Internal QoS
Imaging & Vision Kernel Benchmarks

- DSP with HVX vs Quad Krait CPU with full Neon-Optimization
  - Quad Krait CPU clocked at 2.65GHz
  - Single DSP/HVX clocked at 725MHz
  - Core power only excluding SoC infrastructure, DDR, etc.

Source Qualcomm Internal Performance Evaluation
Full Application Low Light Video Enhancement

Source Qualcomm Internal Performance Evaluation
DSP vs CPU power

- Data for a typical imaging application
- Unlike CPUs, the bulk of the power is spent in compute datapath

DSP w/ HVX

Why power savings?

<table>
<thead>
<tr>
<th>CPU</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOO/Superscalar</td>
<td>In-order/VLIW</td>
</tr>
<tr>
<td>Short Vector have high overhead</td>
<td>Long Vectors amortize overhead</td>
</tr>
<tr>
<td>L1 &amp; L2 used for image data</td>
<td>Only L2 used for image data</td>
</tr>
<tr>
<td>Design Optimized for high MHz incurs high clock power overheads &amp; high leakage</td>
<td>Design Optimized for low-power &amp; lower MHz reduces overheads &amp; leakage</td>
</tr>
</tbody>
</table>
HVX Architecture – Scalar core

- Many target applications have both scalar and vector components

- Good control performance means less need to move control parts of an application to the CPU:
  - Easier to offload a full algorithm rather than partition it
  - Support apps with vector→scalar→vector dependency loops
  - Keep data local in cache
  - Avoid CPU power

Source Qualcomm Internal Performance Evaluation
Visit developer.qualcomm.com to request the latest Hexagon HVX SDK
Hexagon Licensees Deploying HVX Solutions

*Not a complete listing, simple for OEM to enable their own visual partners*
Thank you

Follow us on:  

For more information on Qualcomm, visit us at:  
www.qualcomm.com & www.qualcomm.com/blog

©2013 Qualcomm Technologies, Inc.  
Qualcomm and Hexagon are trademarks of QUALCOMM Incorporated, registered in the United States and other countries. All QUALCOMM Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners. Hexagon is a product of Qualcomm Technologies, Inc.