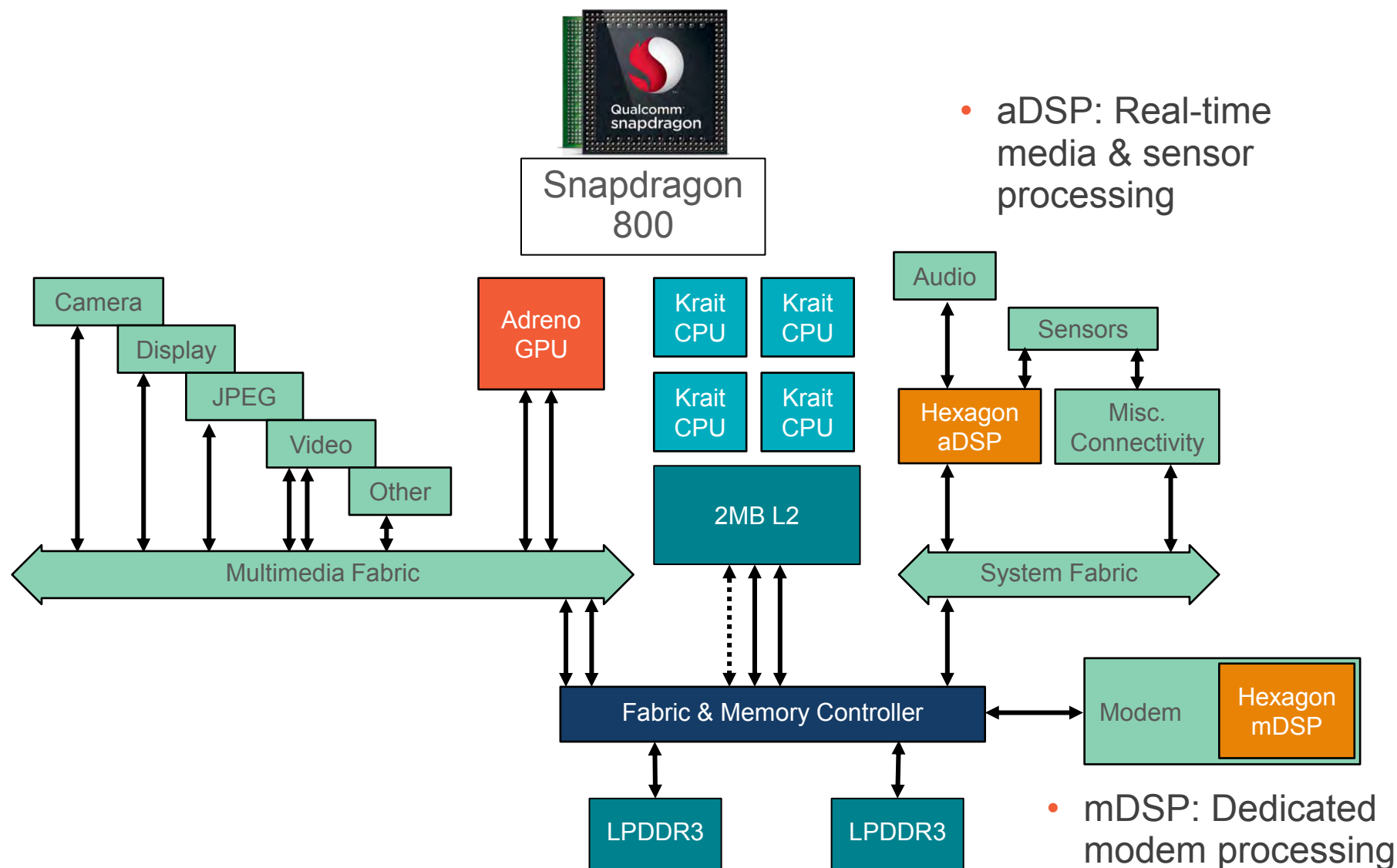


Lucian Codrescu
Sr. Director, Technology
Qualcomm Technologies, Inc.

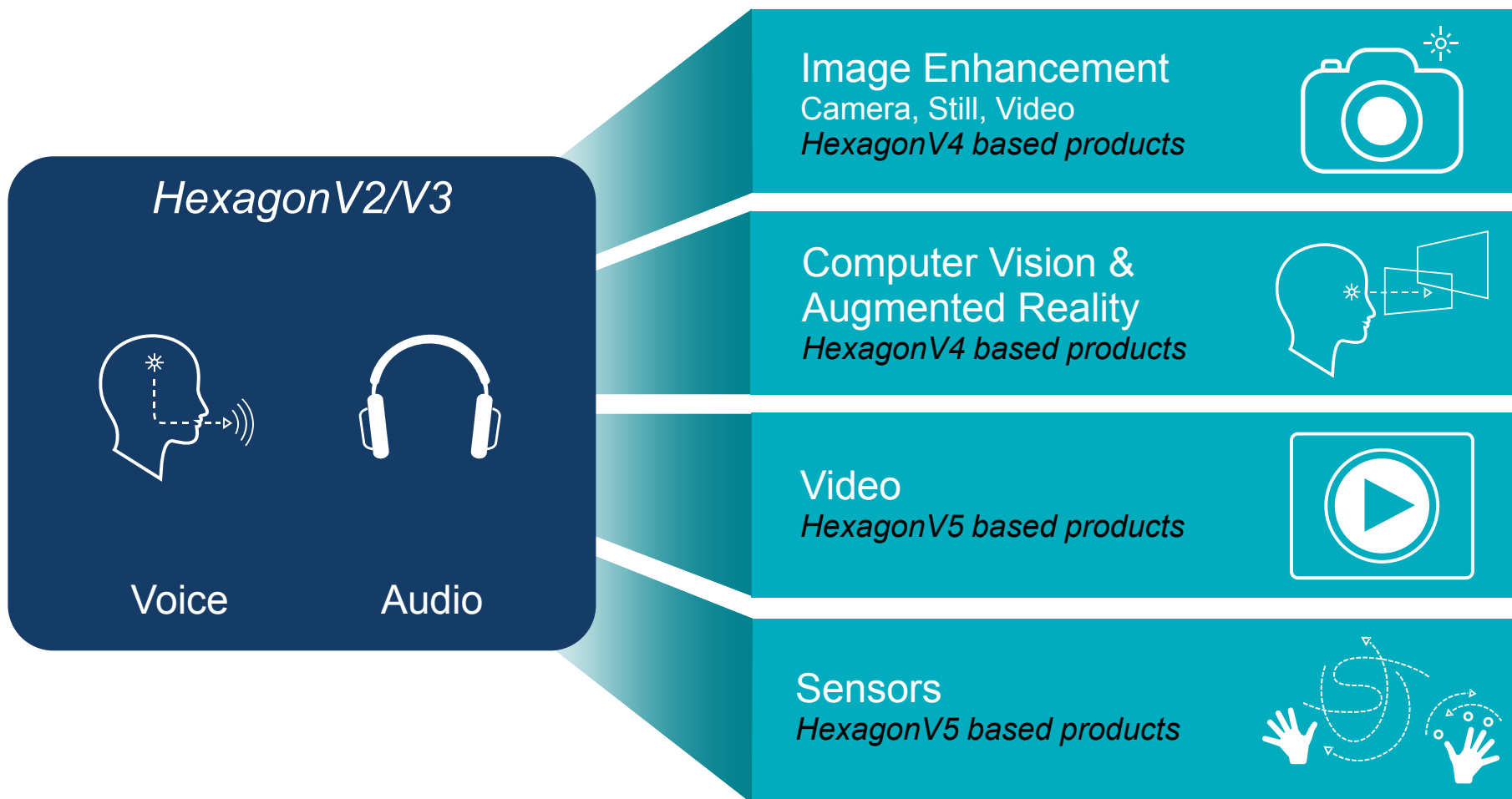
**Qualcomm Hexagon
DSP: An architecture
optimized for mobile
multimedia and
communications**



Hexagon™ DSP processors in Snapdragon products



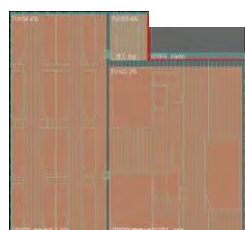
Expansion of Hexagon DSP use cases beyond audio



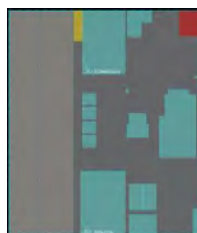
Hexagon DSP is evolving for use beyond voice and audio to computer vision, video and imaging features

The Hexagon DSP evolution

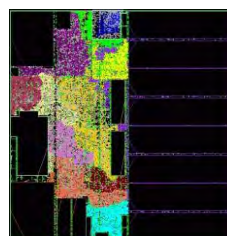
Generational improvements in performance and power efficiency driven by both architecture and implementation



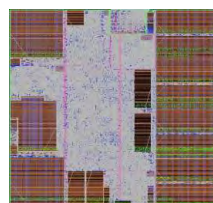
V1
65nm
Oct 2006



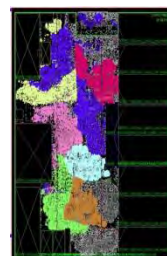
V2
65nm
Dec 2007



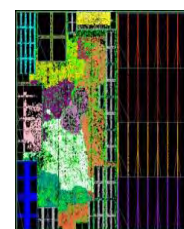
V3M
45nm
June 2009



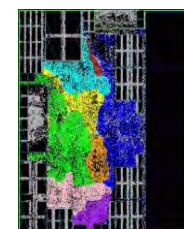
V3C
45nm Aug
2009



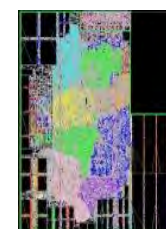
V3L
45nm Nov
2009



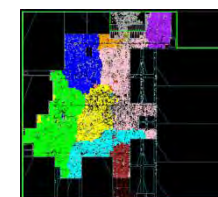
V4M
28nm
Dec 2010



V4C
28nm
Dec 2010



V4L
28nm
Apr 2011



V5A
28nm
Dec 2012



V5H
28nm
Dec 2012

Time



Key characteristics of modem & multimedia applications

Requirements

- Require fixed real-time performance level (fps, Mbit/sec, etc.)
- Extremely aggressive power & area targets

Characteristics

- Mix of signal processing & control code
 - For modem, Qualcomm does not use a split CPU/DSP architecture. All processing is done on Hexagon DSP
 - Multimedia apps have significant control in the RTOS & frameworks
- Heavy L2\$ misses
 - Multimedia is data intensive
 - Modem is code intensive

Hexagon DSP blends features targeted to modem & multimedia

VLIW

- Need multi-issue to meet performance
- Low complexity for Area & Power

Multi-Threading

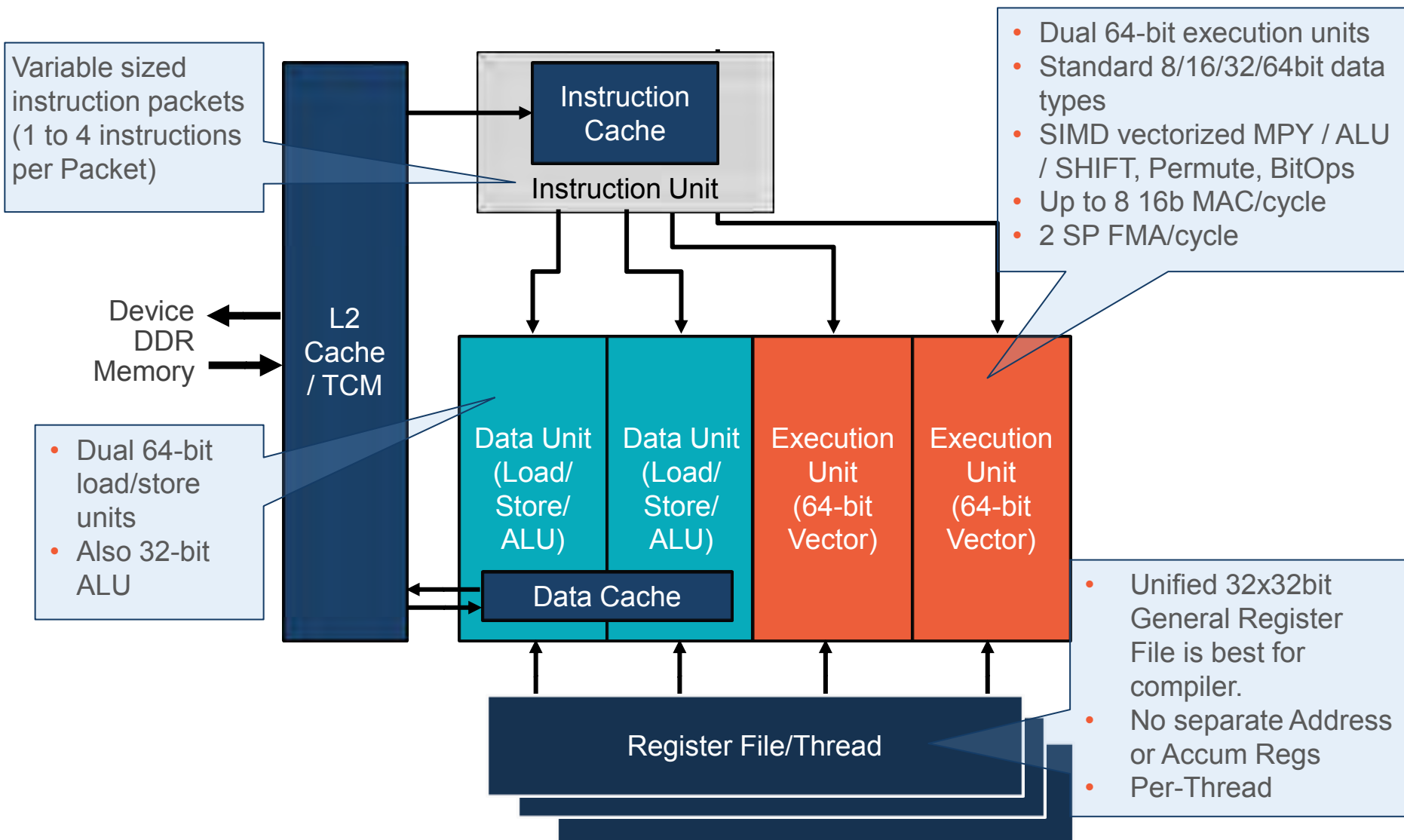
- To reduce L2\$ miss penalty without the need for a large L2
- Increases instructions/VLIW packet because compiler doesn't need to schedule latency



Innovate in ISA to maximize IPC

- More work/VLIW packet reduces energy/instruction
- Keep the pipelines full for MIPS/mm2
- Target both Signal Processing & Control code

VLIW: Area & power efficient multi-issue



Maximizing the signal processing code work/packet

Example from inner loop of FFT: Executing 29 “simple RISC ops” in 1 cycle

64-bit Load and

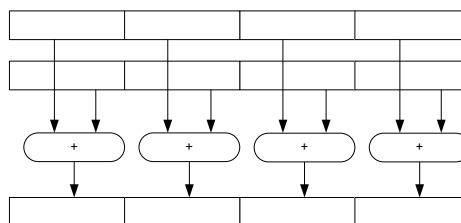
64-bit Store with
post-update
addressing

```
{ R17:16 = MEMD(R0++M1)
  MEMD(R6++M1) = R25:24
  R20 = CMPY(R20, R8):<<1:rnd:sat
  R11:10 = VADDH(R11:10, R13:12)
};endloop0
```

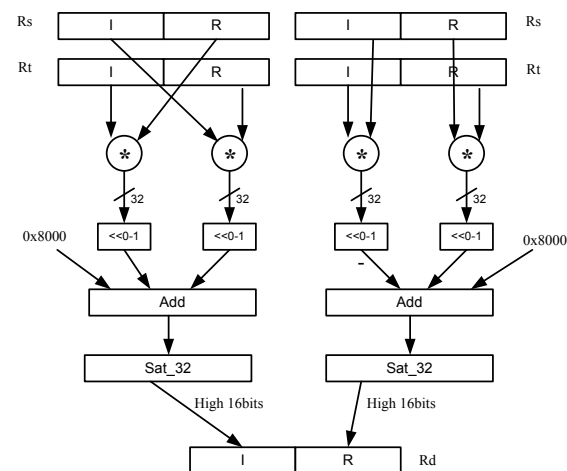
Zero-overhead loops

- Dec count
- Compare
- Jump top

Vector 4x16-bit Add



Complex multiply with
round and saturation



Maximizing the control code work/packet

Hexagon DSP ISA improves control code efficiency over traditional VLIW

Example C code

```
void example(int *ptr, int val) {
    if (ptr!=0) {
        *ptr = *ptr + val + 2;
    }
}
```

Traditional VLIW Assembly Code

```

1  p0 = cmp.eq(r0,#0)
   {
2  if (!p0) r2=memw(r0)
   if (p0) jumpr:nt r31
   }
3  r2 = add(r2,#2)
4  r1 = add(r1,r2)
   {
5  memw(r0) = r1
   jumpr r31
   }
```

Hexagon DSP: Dot-New Predication

```

{
1  p0 = cmp.eq (r0,#0)
   if (!p0.new) r2=memw(r0)
   if (p0.new) jumpr:nt r31
2  }
2  r2 = add(r2,#2)
3  r1 = add(r1,r2)
   {
4  memw(r0) = r1
   jumpr r31
   }
```

Hexagon DSP: Compound ALU

```

{
1  p0 = cmp.eq(r0,#0)
   if (!p0.new) r2=memw(r0)
   if (p0.new) jumpr:nt r31
2  }
2  r1 = add(r1,add(r2,#2))
   {
3  memw(r0) = r1
   jumpr r31
   }
```

Hexagon DSP: New-Value Store

```

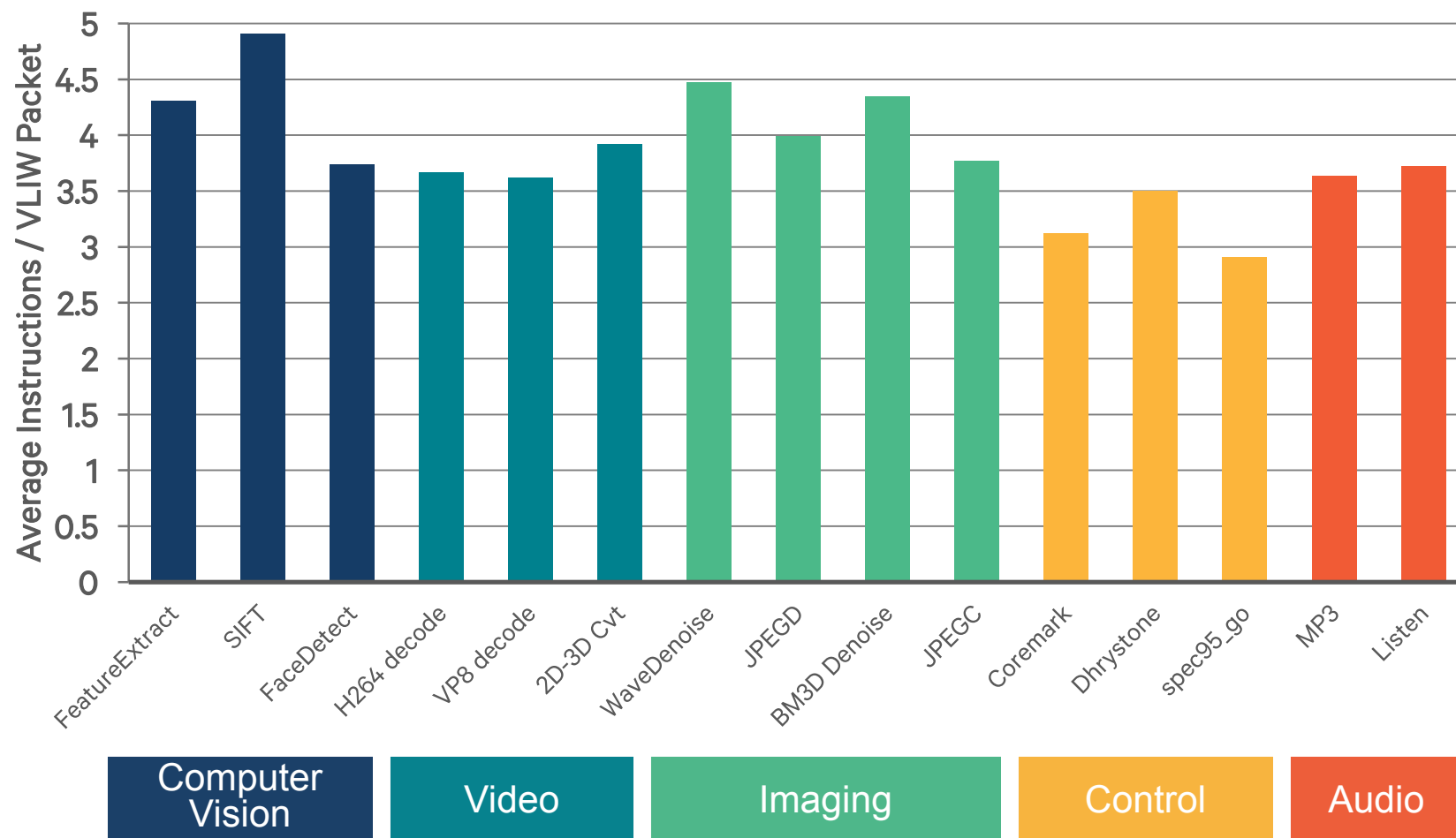
{
1  p0 = cmp.eq(r0,#0)
   if (!p0.new) r2=memw(r0)
   if (p0.new) jumpr:nt r31
2  }
2  r1 = add(r1,add(r2,#2))
   {
2  memw(r0) = r1.new
   jumpr r31
   }
```

Instr/Packet =
7 instr/5 packets = 1.4

Instr/Packet =
7 instr/2packets = 3.5

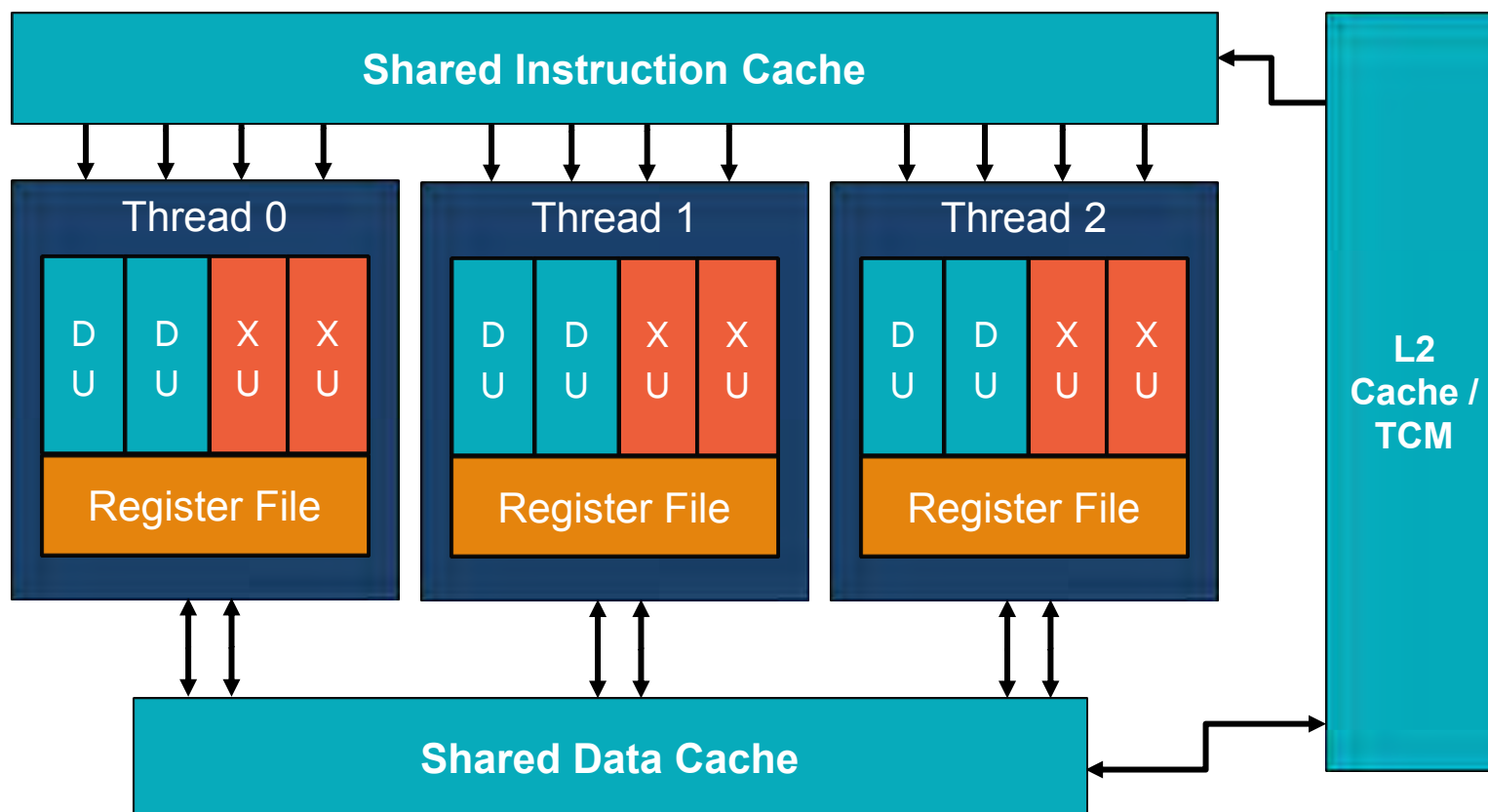
High avg. instructions/packet for targeted use cases

Compound instructions count as 2



Programmer's view of Hexagon DSP HW multi-threading

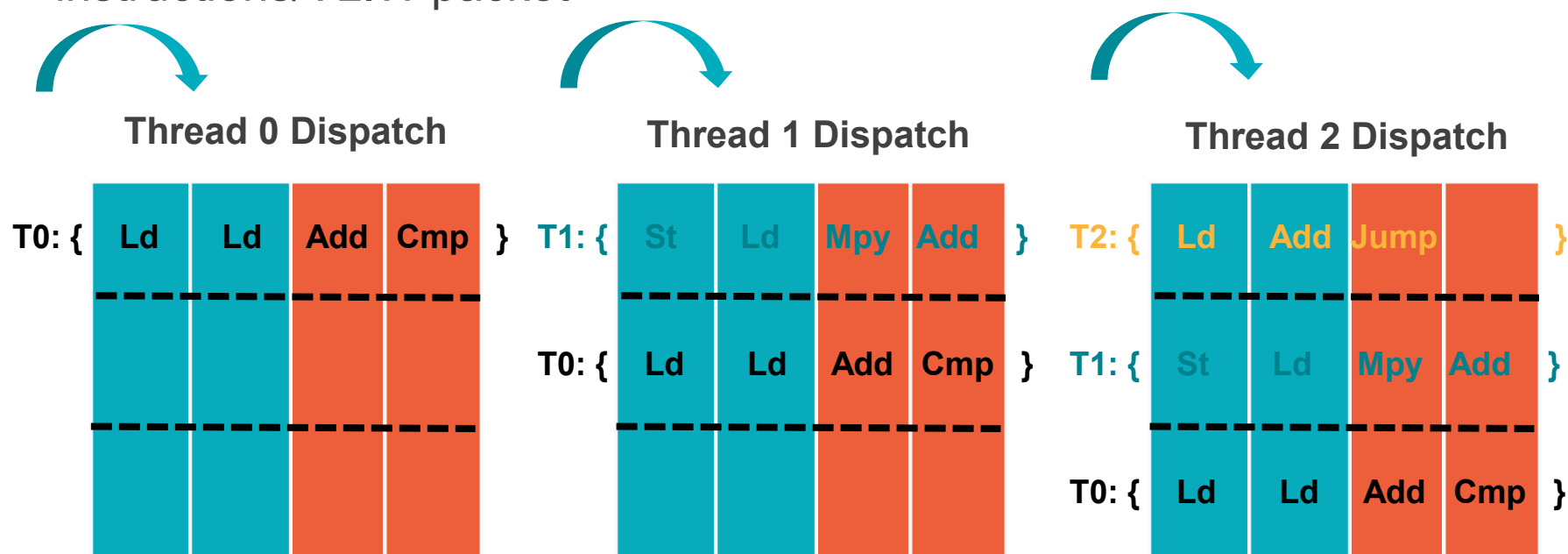
- Hexagon V5 includes three hardware threads
- Architected to look like a multi-core with communication through shared memory



Hexagon DSP V1-V4: Interleaved multi-threading

Simple round-robin thread scheduling

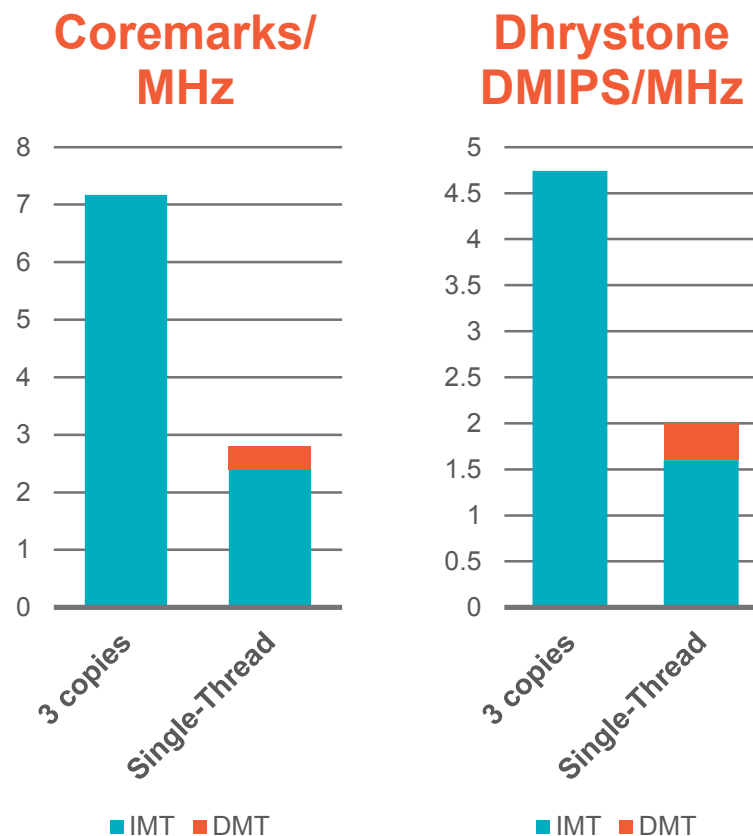
- Number of threads match execution pipe depth (three threads → three execute stages)
- All instructions complete before next packet dispatch
- Compiler schedules for zero-latency which helps to increase instructions/VLIW packet



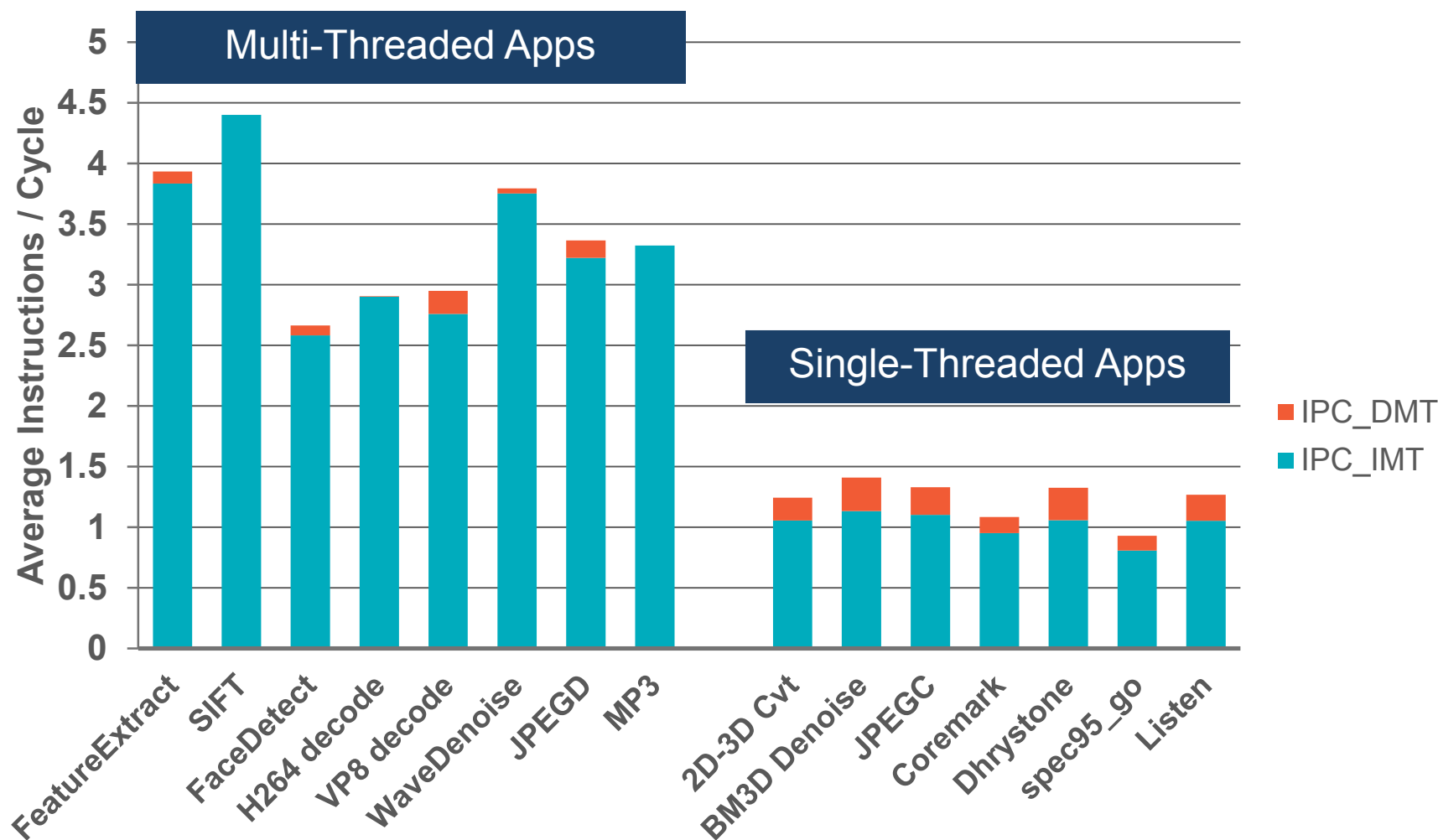
Hexagon DSP V5: Dynamic HW multi-threading

Recover some performance when threads idle or stalled

- Remove a thread from IMT rotation
 - On L2 cache misses
 - When in wait-for-interrupt or off mode
- Additional forwarding to support 2-cycle packets
- VLIW packets with dependencies between long latency instructions will stall
 - But many VLIW packets with simple instructions can complete in 2 processor clocks

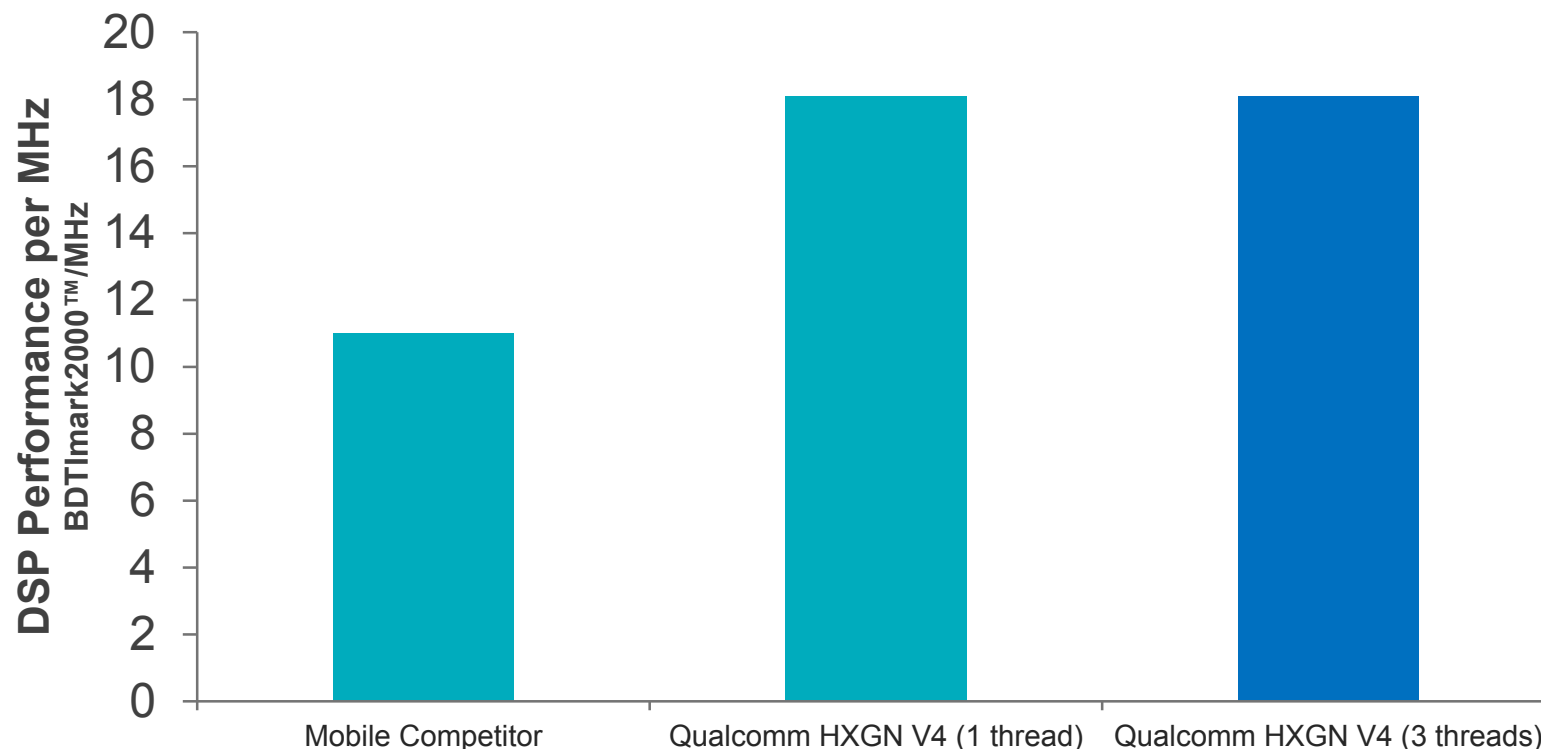


Hexagon DSP instructions per cycle



Qualcomm Hexagon DSP architecture

Highly efficient mobile application processor—designed for more performance per MHz



Clock Rate (MHz)

430-520

100-233

300-700

DSP Performance (BDTImark2000)

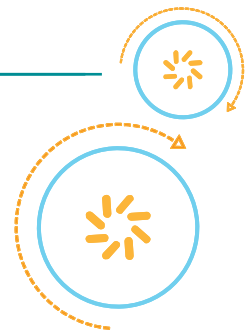
4730-5720

1810-4220

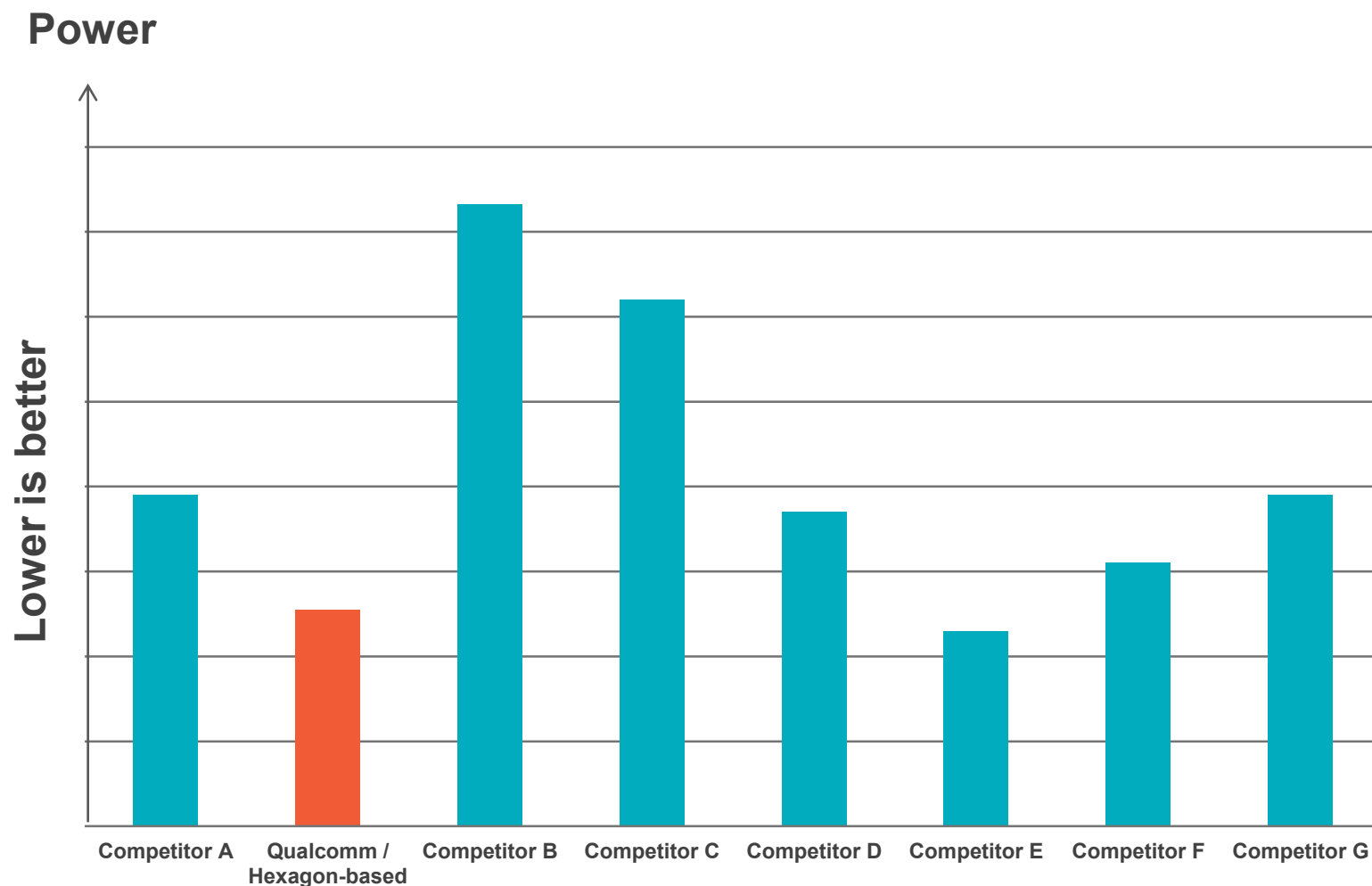
5440-12660*

* - Projected best case score for 3-threads

Hexagon DSP Power Benefits

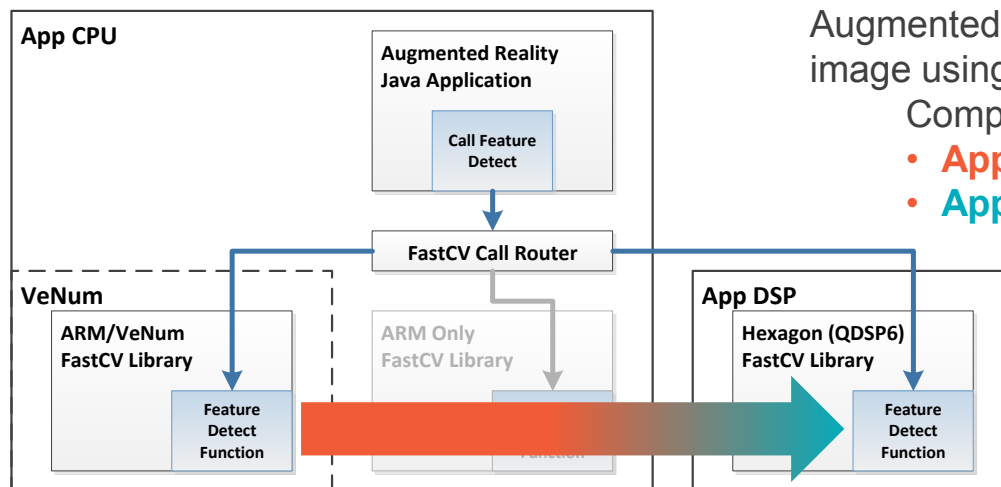


MP3 playback power for competitive smartphones



- Power measured at the battery for various phones
- Includes everything: DSP, CPU, memory, analog components, etc

Computer vision offload – ARM/neon to Hexagon DSP

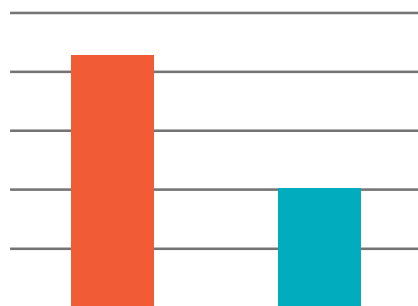


Augmented Reality Java App finding objects in image using **FastCV Feature Detect**

Comparison of Feature Detect run on:

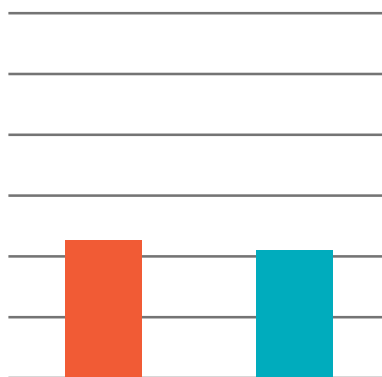
- **App CPU (ARM/Neon)**
- **App DSP (Hexagon)**

CPU Utilization (%)



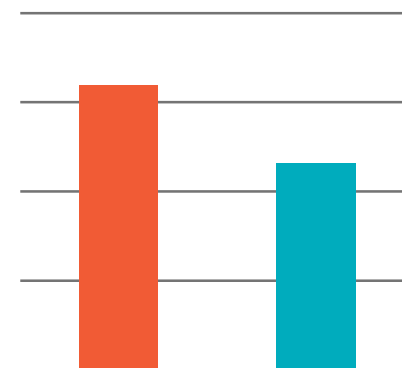
52% Less CPU

Detection Time (%)



7% Less Time

Total Device Power (%)

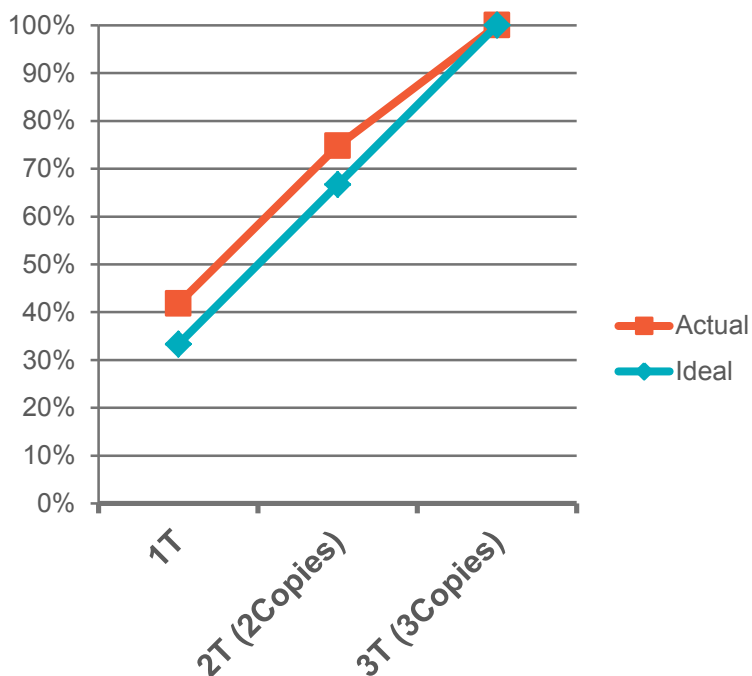


32% Less Power*

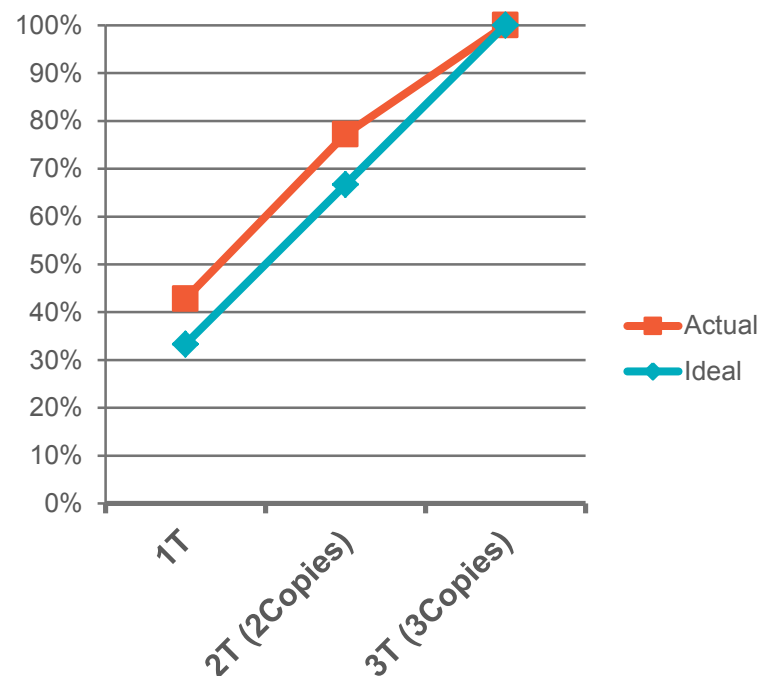
Hexagon DSP power for different thread utilizations

- Excellent near-linear power scalability
(as threads go idle, power used by the thread is nearly eliminated)
- Achieved through optimized clock tree design & clock gating

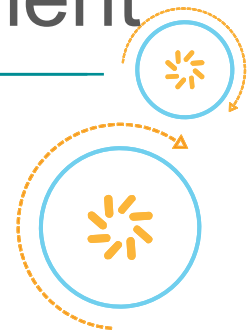
**Dhrystone Power,
IMT Mode**



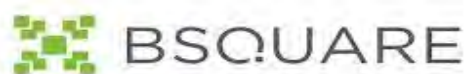
**FIR Power,
IMT Mode**



Hexagon DSP Software Development



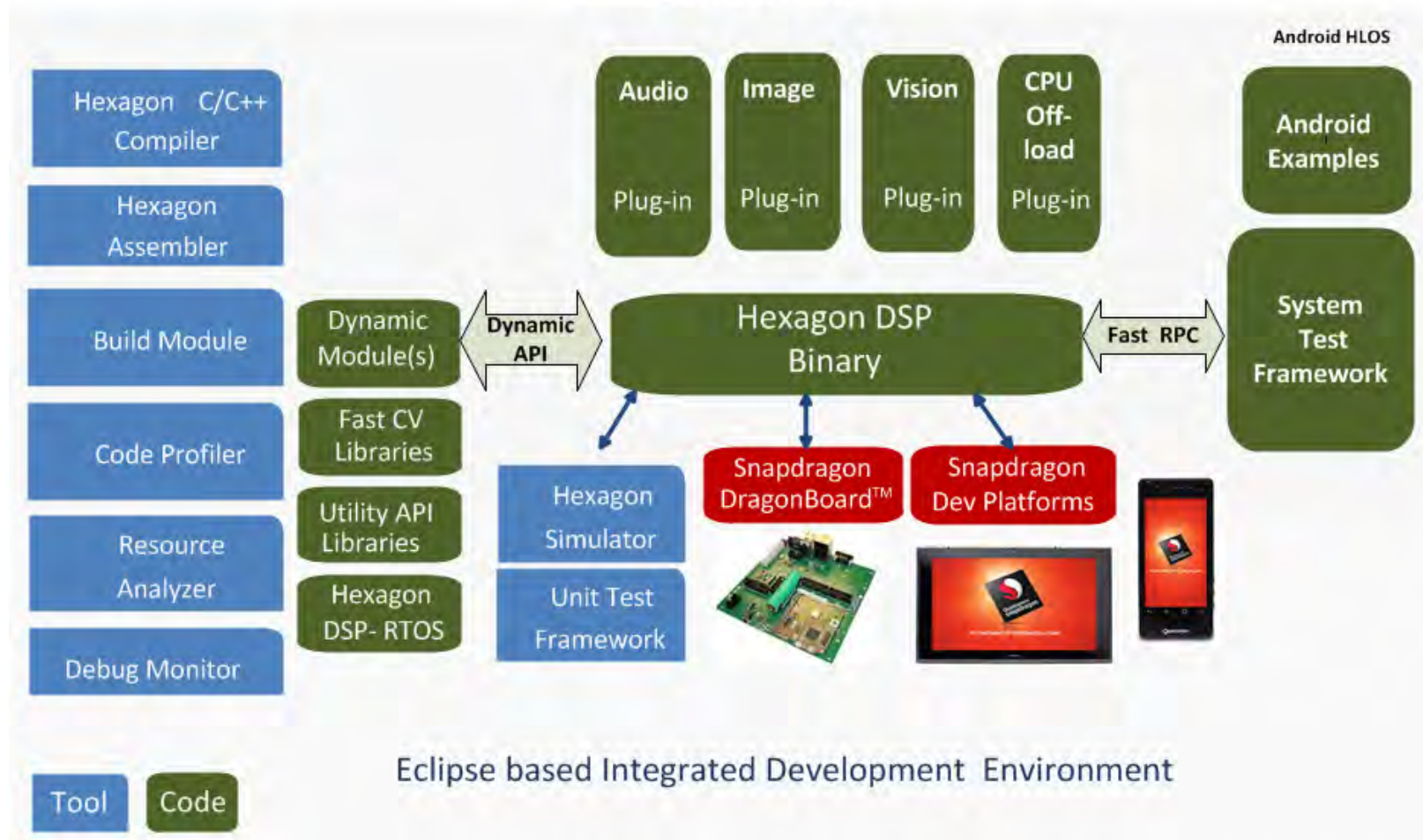
Independent Algorithm Developers on Hexagon DSP



Announcing the Hexagon DSP SDK

See the *Hexagon DSP SDK* in action at *Uplinq2013* (www.uplinq.com)

Hexagon DSP SDK



Visit <http://developer.qualcomm.com> for more information.

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.

