The Linley Group

MICROPROCESSOR report

Insightful Analysis of Processor Technology

# QUALCOMM EXTENDS HEXAGON DSP

## *Hexagon v5 Adds Floating-Point Math, Dynamic Multithreading*

*By Linley Gwennap (August 26, 2013)*

Relying on its in-house DSP to spur new smartphone features, Qualcomm has improved the capabilities of its Hexagon architecture and is making it easier than ever for programmers to tap into. At today's Hot Chips conference, the company disclosed improvements such as floating-point support and dynamic multithreading that it implemented in the Hexagon v5 generation, which recently began shipping in the Snapdragon 800 processor. These improvements expand the DSP's range of applications to include image and video processing as well as computer vision and sensor analysis.

The new Moto X smartphone is the poster child for the kind of capabilities Hexagon enables. The phone implements "always on" voice recognition, which allows users to simply say "Hey Google Now" and then give a voice command without touching the device to wake it up. Moto X also recognizes a quick twisting gesture and places the phone into camera mode, bypassing the usual passcode and startup screen. The Hexagon DSP is essential for these features, operating in a low-power mode that can analyze voice and gesture input without waking the main CPUs. This approach helps Motorola claim a 24-hour battery life for the phone even with these new features continuously on.

Spurring innovation requires lots of programmers on the platform. Qualcomm first released the Hexagon architecture to partners in 2011, but it is now planning to offer a complete software-development kit (SDK) on its web site. Anyone can download and use these tools free of charge under a simple license agreement. Qualcomm will launch the new SDK at its Uplinq conference in September.

Dozens of companies are already developing algorithms or applications for Hexagon, including Acoustic Technologies, AM3D, Dolby, DTS, Morpho, Pelican Imaging, QSound, and TranSono. Most of these companies focus on voice and audio processing, but given the new Hexagon v5 capabilities, Qualcomm hopes to attract more computer-vision developers.

### Floating Point Expands Range

Hexagon appears in a variety of Qualcomm products, including the company's cellular-baseband processors, application processors, and femtocell processors. Most Snapdragon processors have at least two Hexagon cores: one for the cellular modem and one for application processing. The DSP uses a VLIW-style architecture that can execute four instructions per cycle. It is unusual in its multithreading capability (see *MPR 10/31/11*, "Qualcomm Aims Hexagon at Femtocells").

When we last wrote about Hexagon, the architecture was in its third version (QDSP6v3). Since then, Hexagon v4

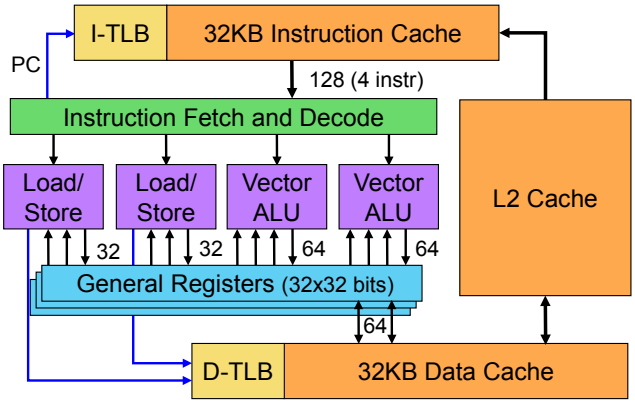


**Figure 1. Block diagram of Hexagon v5 DSP.** The VLIW-style design can issue four instructions per cycle. Each of the vector ALUs also has a single-precision FMAC unit, and each load/store unit can also perform simple 32-bit ALU operations.

made its debut with the MSM8960 in 2Q12, and the company moved to Hexagon v5 in its new 8974 processor. The basic microarchitecture, shown in Figure 1, is quite similar from v3 to v5. One minor change is that the second load unit in the v3 design became a load/store unit in v4, doubling the store bandwidth.

Note that each load/store unit has a 32-bit ALU for address generation, and the compiler can assign basic arithmetic operations to these units if they are not needed for memory operations on a given cycle. Although the basic register size is 32 bits, memory operations can transfer 64 bits (two registers) to or from memory.

Similarly, the vector ALU can operate on two paired registers at a time. These 64-bit values can be partitioned into 8-, 16-, or 32-bit values and operated on in SIMD fashion. Each unit includes an integer multiply-accumulate (MAC) unit that can produce up to four 16-bit results every cycle.

For Hexagon v5, Qualcomm added a floating-point unit to each vector ALU. Because most of the target applications, such as image and vision processing, require only single-precision operations, the designers implemented a 32-bit FP multiply-add (FMA) unit. (The hardware includes some features to accelerate software emulation of double-precision operations, if they are needed.) Each of the two FMA units can complete a multiply, add, or multiply-add operation every cycle, yielding peak throughput of 2.8Gflops at 700MHz.

## Multithreading Becomes Dynamic

In Hexagon v4, Qualcomm reduced the number of threads from six to three, for two main reasons. First, programmers had a hard time filling all six threads with useful work, so the new model is easier to program. Second, the instruction latency improved to the point that the extra threads were unneeded. Since each thread requires its own set of physical registers, this change also reduces the size of the register file.

As Figure 2 shows, Hexagon v4 uses a simple method called interleaved multithreading (IMT) that always shifts to the next thread on each cycle. As a result, each instruction has three clock cycles to complete before the next instruction in its thread is issued. Taking advantage of this leisurely pace, most instructions do not produce their result until the second or third execution cycle. Thus, the design is optimized for three threads and three execution cycles. Targeting a clock speed of 300MHz to 700MHz in 28nm, the design operates much more slowly than leading-edge CPUs. The Krait CPU in the Snapdragon 800, for example, clocks as fast as 2.3GHz.

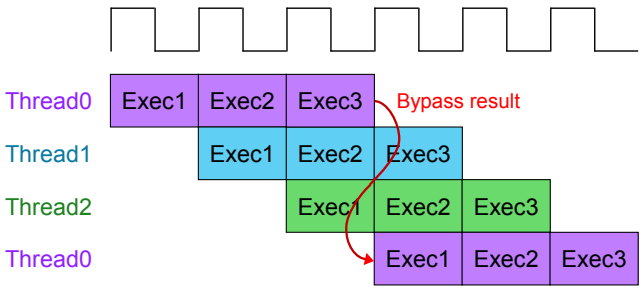


**Figure 2. Multithreading in Hexagon.** The DSP rotates among three threads, issuing one instruction bundle on each cycle. Most instructions require three execution cycles to generate a result.

Taking advantage of this extra time, Hexagon includes operations such as permutation, complex ($a$+$b$i) arithmetic, saturation, and rounding that often take extra cycles on a traditional design. In some cases, the compiler can even pair dependent operations in the same instruction bundle. For example, a condition instruction and a conditional branch can execute together, allowing a test-and-branch operation to issue in one bundle. Similarly, a basic add followed by a dependent add (for example, $x = a + b + c$) can be grouped into one bundle. These combinations reduce the number of bundles needed to encode a particular function, allowing it to complete sooner.

Using the IMT model, the execution time of a single thread is essentially the same as that of three threads, since each thread always issues instructions on one out of three cycles. The only difference comes from the fact that the three threads share the cache, reducing their hit rate when all are running.

To improve single-thread performance, Hexagon v5 includes a new mode called dynamic multithreading (DMT). In this mode, the DSP skips any threads that are processing an L2 cache miss, waiting for an interrupt, or disabled by software. In the degenerate case, a single thread could have full access to the machine.

One might expect DMT to triple the performance of a single thread. But remember that most Hexagon instructions actually take three cycles to execute, with two of those cycles hidden by the thread switching. Even in DMT mode, the DSP cannot issue the next instruction bundle until the previous one completes, so the program often proceeds at the same pace. Some instructions, however, complete in only two cycles. If a bundle contains only two-cycle instructions, the next bundle can issue after two cycles rather than three. Thus, if the processor is in DMT mode with only one or two active threads, these fast instructions occasionally save a cycle.

For applications with plenty of threads and few cache misses, DMT offers little or no benefit, but it accelerates some multithreaded applications by as much as 6%. Single-threaded code sees a bigger gain, of course. CoreMark performance jumps by 12%, and other programs gain 10% to 20%, according to Qualcomm's testing. DMT incurs little hardware overhead (just a tweak to the issue logic and some extra bypass logic for two-cycle instructions), so this gain is well worth the trouble.

## One Instruction Bundle, 29 Operations

To demonstrate Hexagon's power, Qualcomm offers the code example in Figure 3. This bundle contains four instructions—one for each of the function units—and thus can issue in a single cycle. The two memory units execute one load and one store, each moving data to two 32-bit registers at a time. (Qualcomm counts them as two operations each.) The load and store instructions also increment their memory-address registers (two more operations). One vector unit performs a complex multiply instruction that, with rounding and saturation, includes 16 operations. The other executes a vector instruction consisting of four 16-bit integer adds.

Because the DSP implements zero-overhead looping, this single-instruction loop also decrements the loop-count register, checks to see if it has reached zero, and branches back to the start of the loop (three more operations). Thus, this instruction bundle can be viewed as encoding a total of 29 simple operations. This count does not imply that Hexagon is 29 times more efficient than other architectures, many of which also include SIMD operations, 64-bit loads and stores, and (for DSPs) zero-overhead looping. But it illustrates the large amount of work the DSP can accomplish in one clock tick.

The DSP gurus at BDTI have tested Hexagon using their BDTImark2000. According to the fixed-point benchmark, the v4 and v5 designs each score 1,810 for a single thread running on a 300MHz DSP (100MHz per thread) in IMT mode. For three threads at the maximum DSP speed of 800MHz, this result projects to a best-case score of 14,520. This score exceeds that of the fastest C64x+ DSP from Texas Instruments. The C66x, which TI uses in some of its base-station processors, achieves a peak score of 20,030, however, and Freescale's SC3900 leads the pack at 37,460. Ceva, Hexagon's primary DSP competitor in mobile processors, has not certified BDTImark2000 scores for its recent designs.

At Hot Chips, Qualcomm also discussed an example of offloading functions from the CPU to the Hexagon DSP. The example is an augmented-reality application that uses feature detection to find objects in an image. Qualcomm first ran the application on the ARM (Krait) CPU, taking advantage of its Neon SIMD unit, then ported the feature-detection function to the DSP.

Offloading this critical task reduced CPU utilization by 52%. Even taking into account the added power of the DSP, total processor power dropped by 32%. This power savings carried no performance penalty; in fact, the application could detect features slightly faster using the DSP. Given the ratio of power savings to CPU utilization, we estimate Hexagon is three times more power efficient than the CPU for this particular task.

### For More Information

For more information about Hexagon v5, access the Hot Chips presentation at *www.hotchips.org*. The new Hexagon SDK will be available starting September 3 at the Qualcomm developers' web site (*http://developer.qualcomm.com*). A complete list of certified BDTImark2000 scores is available at *www.bdti.com/Resources/BenchmarkResults/BDTImark2000*.

## Taking Your Phone to the Next Level

Although most mobile processors include a small DSP core to offload audio and other tasks from the CPU, Qualcomm has taken this approach to a new level. The recent architecture upgrades enable Hexagon to offload more-complicated tasks from the main CPUs. For example, smartphone users today want to enhance their photos and videos right away, without transferring them to a PC. The hard-wired image processor (ISP) performs only the most common tasks, so the power-hungry CPU usually carries the burden. Hexagon's flexibility and high performance enables it to implement new video and imaging algorithms, offloading the CPU and reducing system power.

The next wave of smartphone applications could involve vision processing and sensors. Recognizing gestures, faces, and objects is computationally difficult, but it could enable exciting new capabilities. The new Hexagon architecture is well suited to these next-generation tasks. As in the example above, shifting these tasks from the CPU to a power-efficient DSP can reduce processor power by 30%. Mobile-processor vendors using small DSP cores, often licensed from a third party, may lack sufficient horsepower to handle these emerging applications.

Qualcomm also uses Hexagon in its processors for radio signal processing in small-cell base stations. These

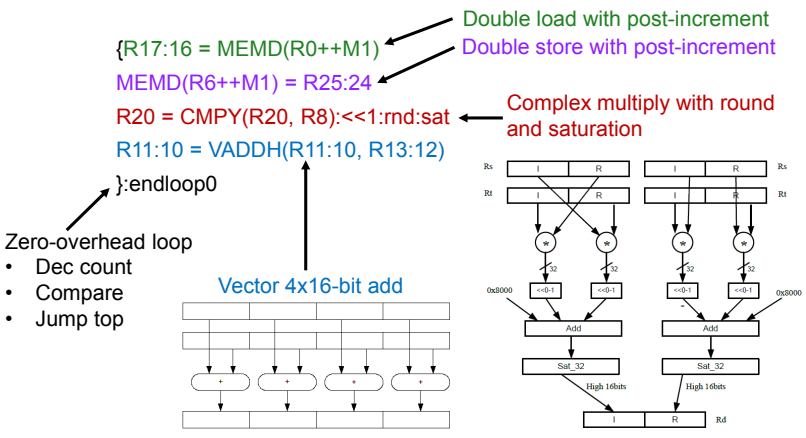


**Figure 3. Hexagon code example.** Taken from the inner loop of a fast Fourier transform (FFT), this instruction bundle issues in a single cycle but performs the equivalent of 29 simple operations. (Source: Qualcomm)

processors compete against chips from TI and Freescale, both of which have developed powerful in-house DSPs. As Qualcomm beefs up the performance and capabilities of its own DSP, it can better compete against these traditional DSP powerhouses. Just as in phones, limiting DSP power is increasingly important in small cells as well.

By deploying the same DSP architecture in both mobile clients and mobile infrastructure, Qualcomm benefits from economies of scale. No competitor has such a broad portfolio across which to amortize its DSP development costs. Owing to its potential advantages in cost and power, Qualcomm is making it difficult for its DSP competitors. ♦