2011 is going to be a year dominated by multi-core smartphone launches, but there always has to be a first. So just like that, we have our first example of said category of smartphone, the LG Optimus 2X, with Nvidia's dual-core 1 GHz Tegra 2 AP20H at its heart. The Optimus 2X (simply the 2X henceforth) hasn’t changed much since we saw it at CES—the hardware is aesthetically the same, and software at first glance is the same as well.

Read on for our full review.

Page 1

In many ways, the smartphone platform has evolved following the same kinds of steps we saw in the early days of the PC—lots of different software and hardware platforms, rapidly changing lead players, faster and faster platform update cadence, the slow emergence of obvious majority leaders. Anand and I have talked extensively about just how striking the similarities are between the PC evolution and the current mobile one, but one of the striking differences is just how much faster that evolution is happening in the mobile space. The reason is simple—nearly all the hard lessons have already been learned in the previous PC evolution, it's just a matter of porting that knowledge to mobile under a different set of constraints.

2011 is going to be a year dominated by multi-core smartphone launches, but there always has to be a first. So just like that, we have our first example of said category of smartphone, the LG Optimus 2X, with Nvidia’s dual-core 1 GHz Tegra 2 AP20H at its heart. The Optimus 2X (simply the 2X henceforth) hasn’t changed much since we saw it at CES—the hardware is aesthetically the same, and software at first glance is the same as well. We weren’t able to publish benchmarks at that time purely because LG hadn’t finalized the software build on that test hardware, but we definitely can do so now.

First off are the hardware specs. There’s a version of the 2X already shipping on South Korea Telecom which is similar but not identical to the version we were sampled—what we’re reviewing is the LG PP990 rather than the SU660. You can look at the specs of that Korean version and compare yourself, but the differences boil down to a few things. The South Korean version ships with 16 GB of internal storage compared to 8 GB like ours, Xenon versus LED flash, likely a different build of Android (more on that later), and a physically different set of Android buttons. The Korean version also has T-DMB for mobile TV. LG hasn’t
officially announced what carrier the 2X will launch with stateside, nor has it been specific about what UMTS or GSM bands that final version will work with, I'd expect that announcement to happen at MWC. Needless to say, I was surprised that the 2X immediately hopped on HSPA when I inserted my personal AT&T SIM. Regardless, just know that what we're reviewing here is something between the international model and what will be launched in the US. The 2X will launch running Android 2.2.1 and is already slated to move to Android 2.3 at some time in the future.

<table>
<thead>
<tr>
<th>Physical Comparison</th>
<th>Apple iPhone 4</th>
<th>Motorola Droid 2</th>
<th>Samsung Galaxy S Fascinate</th>
<th>Google Nexus S</th>
<th>LG Optimus 2X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>115.2 mm (4.5&quot;)</td>
<td>116.3 mm (4.6&quot;)</td>
<td>106.17 mm (4.18&quot;)</td>
<td>123.9 mm (4.8&quot;)</td>
<td>123.9 mm (4.87&quot;)</td>
</tr>
<tr>
<td>Width</td>
<td>58.6 mm (2.31&quot;)</td>
<td>60.5 mm (2.4&quot;)</td>
<td>63.5 mm (2.5&quot;)</td>
<td>63.0 mm (2.48&quot;)</td>
<td>63.2 mm (2.48&quot;)</td>
</tr>
<tr>
<td>Depth</td>
<td>9.3 mm (0.37&quot;)</td>
<td>13.7 mm (0.54&quot;)</td>
<td>9.91 mm (0.39&quot;)</td>
<td>10.88 mm (0.43&quot;)</td>
<td>10.9 mm (0.43&quot;)</td>
</tr>
<tr>
<td>Weight</td>
<td>137 g (4.8 oz)</td>
<td>169 g (5.9 oz)</td>
<td>127 grams (4.5 oz)</td>
<td>129 grams (4.6 oz)</td>
<td>139.0 grams (4.90 oz)</td>
</tr>
<tr>
<td>CPU</td>
<td>Apple A4 @~800MHz</td>
<td>Texas Instruments OMAP 3630 @ 1 GHz</td>
<td>1 GHz Samsung Hummingbird</td>
<td>1 GHz Samsung Hummingbird</td>
<td>NVIDIA Tegra ULV GeForce @ 100-300 MHz</td>
</tr>
<tr>
<td>GPU</td>
<td>PowerVR SGX 555</td>
<td>PowerVR SGX 530</td>
<td>PowerVR SGX 540</td>
<td>PowerVR SGX 540</td>
<td>8 GB integrated (5.51 GB internal SD, 1.12 phone storage), up to 32 microSD</td>
</tr>
<tr>
<td>RAM</td>
<td>512MB LPDDR1 (?)</td>
<td>512 MB LPDDR1</td>
<td>512 MB LPDDR1</td>
<td>512 MB LPDDR1</td>
<td>8 GB integrated, 5 MP with autofocus, LED flash, 1080p24 video recording, 1.3 MP front facing</td>
</tr>
<tr>
<td>NAND</td>
<td>16GB or 32GB integrated</td>
<td>8 GB integrated, preinstalled 8 GB microSD</td>
<td>2 GB, 16 GB microSD (Class 2)</td>
<td>16 GB Integrated</td>
<td></td>
</tr>
<tr>
<td>Camera</td>
<td>5MP with LED Flash + Front Facing Camera</td>
<td>5 MP with dual LED flash and autofocus</td>
<td>5 MP with Autofocus, LED Flash, VGA front facing, 720P Video</td>
<td>5 MP with Autofocus, LED Flash, VGA front facing, 720P Video</td>
<td></td>
</tr>
<tr>
<td>Screen</td>
<td>3.5&quot; 640 x 960 LED backlit LCD</td>
<td>3.7&quot; 854 x 480</td>
<td>4&quot; Super AMOLED 800 x 480</td>
<td>4&quot; Super AMOLED 800 x 480</td>
<td>4&quot; IPS-LCD 800x480</td>
</tr>
</tbody>
</table>

On paper, the 2X is impressive. Highlights are obviously the AP20H Tegra 2 SoC, 4-inch IPS display, 8 MP rear camera and 1.3 MP front facing camera, and 1080p24 H.264 (Baseline) video capture. We're going to go over everything in detail, but starting out the review are our hardware impressions.

Page 2

The 2X we received came in packaging that obviously isn't final, but that isn't what's important at this point. Included is a microHDMI type D to full size HDMI type A cable about 6 feet in length, USB power adapter with type C AC plug, microUSB cable, and of course the smartphone itself with included battery. There's no microSD card provided, but that's admittedly somewhat mitigated by the presence of 8 GB of internal storage which looks just like an SD card to Android.
Hardware feel is just how we left it at CES. The 2X's back is a brown soft touch material with a metallic strip running along the center. The "with Google" text is engraved into the strip, which has a gentle slope leading to the camera bulge.

The camera area is raised a few mm above the rest of the device—it's reminiscent of the Droid X's camera bulge but not nearly as dramatic. The 2X's battery cover is removed by jamming your thumb in a slot at the bottom and tearing it off—plastic snaps around the edge hold it in place.

The 2X puts the battery cover in-between the last vertex of the camera and object space, one of the things we've complained about the Nexus One and other devices doing which basically provides an additional two more surfaces for fingerprints and grime to happen and add glare. The odd part is that the LED flash port on the battery cover is literally just a hole—this is a perfect place for dirt and pocket lint to get under and inside the battery cover.

With the battery cover off you can see the 2X's 5.6 watt-hour battery, SIM slot, and microSD card slot. The microSD position means that you can take the SD card
out with the device turned on, and the slot is of the click-to-eject sort. There's no obvious port for a rear microphone, so there's likely no dual-microphone noise rejection for calling on the 2X.

The 2X is ringed by a silvery metallic plastic, in fact all of the 2X exterior is plastic. The right side of the 2X is home to the volume up and down buttons, which are discrete, clicky, and otherwise perfect.

At the bottom is the microUSB port, and two meshed ports which are reminiscent of the iPhone's design language. The left port is the microphone, right side is the speaker port.

Up top is the power and lock button, in-between is the microHDMI port for HDMI mirroring. There's a snap-off door held on by another piece of plastic which lets the door come off and swivel without totally detaching. To the right is the 1/8" headset jack. The rear of the jack is at a bit of an angle, so some of the connector shows through, but it's not a big deal and the jack works fine.

The front of the 2X is one continuous glass surface. The left and right sides of the display are curved gently in one axis, but nonetheless a noticeable amount. The majority of the display surface used for actual display and interaction is actually flat. Only at the extreme edges is there curvature. It's a bit reminiscent of the Dell Venue Pro, but nowhere near as extreme or pronounced.
The curvature, while attractive, has a downside. That downside is that the surface of the display is essentially defined by the two ridges formed right at the curvy parts—already vertical scratches have been accumulating right at those places. Obviously, this is only a problem if you set the phone front-down.

The 2X's front facing camera is up at the top right. On the left, faintly visible are the two proximity sensor and ambient light sensors. At the top extreme is the main earpiece.

There are no status LEDs on the front of the device or tucked away under the speaker grille on the front like the EVO.
Unlike the Korean version, the P990 model has all capacitive buttons below the display. They're in the same order as the Galaxy S Fascinate, but a different order from everything else except other LG phones. They're all backlit, and there's no light leakage around the icons. There's a generous amount of space above and below the buttons, almost too much extra space. Either LG is leaving room for a carrier logo silkscreen, or else just making things easier for people with gigantic fingers.

The 2X is dimensioned appropriately for the class of 4-inch screened displays it fits in. Thickness is just shy of 11 mm, which is essentially identical to the Nexus S we just reviewed. It's still thinner than the iPhone 4, but not dramatically so. The camera bulge is really what contributes to thickness—the vast majority of the 2X is a millimeter thinner. Weight is specced at 139 grams, we measured it at a slightly heavier 144 grams. It isn't unnervingly light like the Galaxy S series of phones (sans the Epic) were, but rather just right. In-hand feel is actually excellent—I found that the phone naturally sat in the hand so that my index finger rested on the curved ridge just below the camera, balance also was just fine in the palm.

The 2X's build is quite good, there's no flexing or creaking, no rattling when SMSes come in and the phone vibrates, either. It doesn't feel fragile, but it still lacks that hand-on-metal solidness that phones like the Nexus One misgive.

**Page 3**

**NVIDIA's Tegra 2**

The old definition of a computer is a microprocessor, memory, storage, input and output. The modern definition expands it a bit to include coprocessors (e.g. GPUs) as well as multiple types and levels of memory and storage. The type of input and output devices have changed as well. In smartphones keyboards are virtual and output is integrated into the phone. Although the definition of a computer has evolved, it's not all that different.

In the old days, almost each one of these parts of a computer was a discrete component. You had a CPU, memory, a hard drive, a video processor (before they were GPUs) all independent of one another. Go back prior to the 486 and you'll even find that your CPU had to rely on an external FPU for all floating point math.
Moore’s Law has given us bigger, faster, better in all of these areas. Intel’s 486 was the first million transistor x86 processor, introduced in 1989. Sandy Bridge, introduced in January, is just shy of a billion transistors. Sixteen megabytes of memory was a big deal 20 years ago, today high end PCs ship with several gigabytes of memory. Another side effect of Moore’s Law however is integration.

At first integration brought things like the FPU and a second level cache onto the processor die. Over time more components were brought in to the fold. AMD integrated the memory controller in its Athlon 64 processor. Intel brought graphics on-package with Clarkdale/Arrandale, and later on-die with Sandy Bridge. AMD is doing the same with Fusion.

In the smartphone space, the integration is even more pronounced. With physical space and power as major constraints, smartphone chip makers have been forced to further trade performance for integration. The level of integration is so high within a smartphone that you almost never hear about what CPU a phone uses, but rather what Application Processor it uses otherwise known as an SoC (System on Chip or System on a Chip).

Integrate a CPU, GPU, memory controller, video decoder, audio decoder, video encoder (sometimes), camera processor, system memory and maybe even a modem onto a single chip and you’ve got something that can only be described as a System on a Chip. It’s a single physical chip that integrates nearly all of the functions of the entire computer. Nearly all of the aforementioned components are on a single piece of silicon, with the exception of any integrated memory. To save board real estate and enable smaller form factors, it’s not uncommon to stack DRAM on top of the SoC package instead of beside it. The SoC in a PoP (Package on Package) stack has contacts on its top surface that line up with the balls on the DRAM for power and signaling. PoP stacks work because the SoC underneath doesn’t dissipate much heat and thus doesn’t mind being insulated by some DRAM up top.

Examples of SoCs are Qualcomm’s Snapdragon, Texas Instruments’ OMAP 4 and of course the subject at hand, NVIDIA’s Tegra 2. Although this is a review of LG’s Optimus 2X, it’s just as much a review of NVIDIA’s Tegra 2.

**Tegra 2: The SoC**

As a System on a Chip, NVIDIA’s Tegra 2 has a number of processors that make up the whole. Having its roots in the PC industry and being used to briefing inquisitive press, NVIDIA put together this handy die shot that shows the various parts of the Tegra 2:
With the exception of two blocks, the Tegra 2 die is entirely NVIDIA's own creation. The A7 and Cortex A9 blocks are IP licensed from ARM. The entire die is manufactured at TSMC on a 40nm process, similar to NVIDIA's high end GPUs. While its GPUs are built on TSMC's 40nm "G" process, Tegra 2 is a slightly different beast.

Most foundries offer two variants of the same manufacturing process: general purpose (G) and low power (LP). The feature size is the same, however the transistors are tuned differently. TSMC's general purpose process transistors are very fast and low voltage, unfortunately they have very high leakage current. Transistors, as you may know, are electrical on/off switches. Apply a voltage to them and current flows, remove the voltage and current stops flowing. In reality sometimes current flows when you don't want it to, and this is referred to as leakage. High leakage is a side effect of the nice high performance transistors we need to run the fastest processors.

TSMC's 40nm LP process uses lower voltage, slower switching transistors (can't run at as high of a clock speed) that, as a result, have very low leakage characteristics. The lower your leakage and the lower your voltage, the lower your overall power is.

For an SoC, you'd assume that the whole thing would be built at the 40nm LP process. See those two Cortex A9 cores in the diagram above? Remember how they're licensed from ARM? Those things are pretty high performance, they run at 1GHz. Everything else in the chip runs at 300MHz or less for the most part. On top of that, the libraries ARM provides are optimized for TSMC's 40nm G process.

As a result, Tegra 2 uses a mixture of G and LP 40nm transistors on two separate voltage rails. The two Cortex A9 cores and the L2 cache are built using TSMC's 40nm G process transistors, while the rest of the SoC (including the GPU) is built using 40nm LP transistors. The pair of A9s can be powered down together although not independently. We'll get to a deeper discussion of the ARM Cortex A9 shortly.

The ISP (Image Signal Processor), located in the upper left of the die shot above, is responsible for taking the output from a camera (still/video) controller and processing into a usable video state. The Tegra 2 ISP is capable of processing images at a rate of 80 megapixels per second. The ISP supports two cameras, a 12 megapixel primary and a 5 megapixel secondary. The math works out to a maximum of 6 frames per second captured from the primary sensor at 12MP. LG uses the Tegra 2's ISP to enable a 6 fps burst mode as you'll see later on in the review, unfortunately it only works at a 2MP resolution. NVIDIA's strong ISP looks better on paper than in practice it seems.

The video encode processor does real time H.264 video encoding, which is used when capturing video from the camera sensor. NVIDIA doesn't provide any specs on what the encoder is capable of, but we're not too impressed with the quality of its output (again, you'll see more later).

The audio processor is dedicated hardware for audio encoding and decoding. This is used for audio capture as well as audio playback. Even MP3 playback is done on the dedicated audio processor so the Cortex A9s remain powered down to maximize battery life.

The Tegra 2 video decoder can fully accelerate the decode of 1080p H.264 Baseline profile videos at up to 20Mbps. The specs mostly look good on paper as you won't be playing anything near that bitrate on your smartphone. NVIDIA includes dual-display capabilities with Tegra 2. The SoC can output the frame buffer to the smartphone's display as well as an external display via HDMI out.

The ARM A7 nestled in between the video decoder and the L2 cache is used as a system management core. It handles communication between blocks, power management and general SoC management. The remaining blocks (outside of I/O) are the two CPU cores and the GPU. Those both require a lot more detail.

---

The CPU Comparison: NVIDIA, TI & Qualcomm in 2011

NVIDIA makes two versions of the Tegra 2, one for tablets and one for smartphones. The difference between the two boils down to packaging size and TDP. NVIDIA hasn't been too forthcoming with information but here's what I know thus far:
FPU, in the A9 the FPU is fully pipelined—so performance is much improved. The A9's FPU however is still not as quick at math as the optional SIMD MPE.

While all Cortex A8 designs incorporated ARM's NEON SIMD engine, A9 gives you the option of integrating either a SIMD engine (ARM's Media Processing Engine, aka NEON) or a non-vector floating point unit (VFPv3-D16). NVIDIA chose not to include the A9's MPE and instead opted for the FPU. Unlike the A8's MPE, the Scorpion has some big advantages on the floating point side. Qualcomm implements ARM's VFPv3 vector floating point instruction set on Scorpion, the same instructions supported by the Cortex A8. The Cortex A8's FPU isn't pipelined. A single instruction had to make it through the FP pipeline before the next instruction could be issued. For those of you who remember desktop processors, the Cortex A8's non-pipelined FPU is reminiscent of Intel's 486 and AMD's K6. It wasn't until the Pentium processor that Intel gained a pipelined FPU, and for AMD that came with the Athlon. As a result, floating point code runs rather slowly on the Cortex A8.

The Scorpion's VFPv3 FPU is fully pipelined. As a result, floating point performance is much improved. Qualcomm also implements support for NEON, but with a wider 128-bit datapath (compared to 64-bit in the A8 and A9). As a result, Qualcomm should have much higher VFP and NEON performance than the Cortex A8 (we see a good example of this in our Linpack performance results).

While all Cortex A8 designs incorporated ARM's NEON SIMD engine, A9 gives you the option of integrating either a SIMD engine (ARM's Media Processing Engine, aka NEON) or a non-vector floating point unit (VFPv3-D16). NVIDIA chose not to include the A9's MPE and instead opted for the FPU. Unlike the A8's FPU, in the A9 the FPU is fully pipelined—so performance is much improved. The A9's FPU however is still not as quick at math as the optional SIMD MPE.
### Minimum Instruction Latencies (Single Precision)

<table>
<thead>
<tr>
<th>Instruction</th>
<th>FADD</th>
<th>FSUB</th>
<th>FMUL</th>
<th>FMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM Cortex A8 (FPU)</td>
<td>9 cycles</td>
<td>9 cycles</td>
<td>10 cycles</td>
<td>18 cycles</td>
</tr>
<tr>
<td>ARM Cortex A9 (FPU)</td>
<td>4 cycles</td>
<td>4 cycles</td>
<td>5 cycles</td>
<td>8 cycles</td>
</tr>
<tr>
<td>ARM Cortex A8 (NEON)</td>
<td>5 cycles</td>
<td>5 cycles</td>
<td>5 cycles</td>
<td>8 cycles</td>
</tr>
<tr>
<td>ARM Cortex A9 (NEON)</td>
<td>5 cycles</td>
<td>5 cycles</td>
<td>5 cycles</td>
<td>9 cycles</td>
</tr>
</tbody>
</table>

Remember that the A8's FPU isn't pipelined so it can't complete these instructions every cycle, resulting throughput that's nearly equal to the instruction latency. The A9's FPU by comparison is fully pipelined, giving it much higher instruction throughput compared to the A8.

NVIDIA claims implementing MPE would incur a 30% die penalty for a performance improvement that impacts only a minimal amount of code. It admits that at some point integrating a SIMD engine makes sense, just not yet. The table above shows a comparison of instruction latency on various floating point and SIMD engines in A8 and A9. TI's OMAP 4 on the other hand will integrate ARM's Cortex A9 MPE. Depending on the code being run, OMAP 4 could have a significant performance advantage vs. the Tegra 2. Qualcomm's FP/NEON performance should remain class leading in non-bandwidth constrained applications.

Unfortunately for Qualcomm, much of what impacts smartphone application performance today isn't bound by floating point performance. Future apps and workloads will definitely appreciate Qualcomm's attention to detail, but loading a web page won't touch the FPU.

The Scorpion core remains largely unchanged between SoC generations. It won't be until 28nm in 2012 that Qualcomm introduces a new microprocessor architecture. Remember that as an architecture licensee Qualcomm is going to want to create architectures that last as long as possible in order to recover its initial investment. Microprocessor licensees however have less invested into each generation and can move to new architectures much faster.

### Cache & Memory Controller Comparison

NVIDIA outfits the Tegra 2 with a 1MB L2 cache shared between the two Cortex A9 cores. A shared L2/private L1 structure makes the most sense for a dual-core CPU as we've learned from following desktop CPUs for years. It's only once you make the transition to 3 or 4 cores that it makes sense to have private L2s and introduce a large, catch-all shared L3 cache.

Qualcomm's upcoming QSD8660 only has a 512KB L2 cache shared between its two cores, while TI's OMAP 4 has a Tegra 2-like 1MB L2. In these low power parts, having a large L2 with a good hit rate is very important. Moving data around a chip is always very power intensive. The close proximity of the L2 cache to the CPU cores helps keep power consumption down. Any data that has to be fetched from main memory requires waking up the external memory interface as well as the external or on-package DRAMs. A trip to main memory easily requires an order of magnitude more power than pulling data from a local cache.

While the OMAP 4 and Tegra 2 both have a larger L2 cache, it's unclear what frequency the cache operates at. Qualcomm's L2 operates at core frequency and as a result could offer higher bandwidth/lower latency operation.

NVIDIA opted for a single-channel 32-bit LPDDR2 memory interface. Qualcomm's QSD8660/MSM8x60 and TI's upcoming OMAP 4 have two LPDDR2 channels. NVIDIA claims that a narrower memory bus with more efficient arbitration logic is the best balance for power/performance at the 40nm process node. In order to feed the data hungry CPUs and GPU, NVIDIA specs the Tegra 2 for use with 600MHz datarate LPDDR2 memory (although the LG Optimus 2X actually has 800MHz datarate DRAM on package, it still only runs at 600MHz).

Assuming all memory controllers are equally efficient (which is an incorrect assumption), NVIDIA should have half the bandwidth of TI's OMAP 4. NVIDIA's larger L2 cache gives it an advantage over the QSD8660, giving Tegra 2 an effective memory latency advantage for a percentage of memory requests. Depending on the operating frequency of NVIDIA's L2, Qualcomm could have a cache bandwidth advantage. The take away point here is that there's no clear winner in this battle of specifications, just a comparison of tradeoffs.

### The Dual-Core Comparison in 2011

In 2011 Qualcomm will introduce the QSD8660, a Snapdragon SoC with two 45nm Scorpion cores running at 1.2GHz. With a deeper pipeline, smaller cache and a largely in-order architecture, the QSD8660 should still trail NVIDIA's Cortex A9 based Tegra 2 at the same clock speed. However Tegra 2 launches at 1GHz and it won't be until Tegra 2 3D that we see 1.2GHz parts. Depending on timing we could see dual-core Qualcomm phones running at 1.2GHz competing with dual-core NVIDIA phones running at 1.0GHz. The winner between those two may not be as clear—it'll likely vary based on workload.
At 1.2GHz I'd expect the Tegra 2 3D to be the fastest SoC for the entirety of 2011. Once 2012 rolls around we'll reset the clock as Qualcomm introduces its next-generation microprocessor architecture.

NVIDIA clearly has an execution advantage as it is the first SoC maker to ship an ARM Cortex A9. NVIDIA’s biggest weakness on the CPU side is the lack of NEON support in Tegra 2, something that doesn’t appear to be an issue today but could be a problem in the future depending on how widespread NEON code becomes in apps. TI’s OMAP 4 includes both a NEON unit and a wider memory bus, the latter could be a performance advantage depending on how well designed the memory controller is.

Qualcomm is a bit behind on the architecture side. The Scorpion core began shipping in Snapdragon SoCs at the end of 2008 and the architecture won’t be refreshed until late 2011/2012. As Intel discovered with NetBurst, 4—5 year runs for a single microprocessor architecture are generally too long. Luckily Qualcomm appears to be on a ~3 year cadence at this point.

The QSD8660 running at 1.2GHz should be sufficient to at least keep Qualcomm competitive until the Scorpion’s replacement arrives (although I suspect NVIDIA/TI will take the crown with their A9 designs). One aspect we haven’t talked about (mostly because there isn’t any good data available) is power consumption. It’s unclear how the Scorpion core compares to the Cortex A9 in terms of power consumption at the same process node.

The GeForce ULV

Complementing the three major CPU architectures in the mobile applications processor market for 2011 there are three major GPUs you’ll see crop up in devices this year: Imagination Technologies’ PowerVR SGX Series5 and Series5XT, Qualcomm’s Adreno 205/220 and NVIDIA’s GeForce ULV. There are other players but these three are the ones that will show up in the most exciting devices this year.

ImgTec licenses its GPUs for use in a number of SoCs. Apple’s A4, TI’s OMAP 3 and 4 and Samsung’s Hummingbird all use ImgTec GPUs. The currently used high end from ImgTec is the PowerVR SGX 540, which features four unified shader pipelines capable of handling both pixel and vertex shader operations. The PowerVR SGX 543 is widely expected to be used in Apple’s 5th generation SoC.

The PowerVR SGX as well as Qualcomm’s Adreno GPUs both implement tile based deferred rendering architectures. In the early days of the PC GPU race deferred renderers were quite competitive. As geometry complexity in games increased, ATI and NVIDIA’s immediate mode rendering + hidden surface removal proved to be the better option. Given the lack of serious 3D gaming, much less geometry heavy titles on smartphones today the tile based approach makes a lot of sense. Tile based renderers conserve both power and memory bandwidth, two things that are in very short supply on smartphones. Remember from our CPU discussions that in many cases only a single 32-bit LPDDR2 memory channel has to feed two CPU cores as well as the GPU. By comparison, even PCs from 10 years ago had a 64-bit memory bus just for the CPU and a 128-bit memory bus for the GPU.

NVIDIA believes that the future of GPUs on smartphones is no different than the future of PC GPUs: immediate mode renderers. As a result, the GeForce ULV GPU in NVIDIA’s Tegra 2 looks very similar to a desktop GPU—just a lot smaller, and a lot lower power. It’s also worth pointing out that until we get PC-like content on smartphones, NVIDIA’s approach to ultra mobile GPU architectures may not always make the most sense for power efficiency.

(Note that some of what follows below is borrowed from our earlier coverage of NVIDIA’s Tegra 2):

At a high level NVIDIA is calling the GeForce ULV an 8-core GPU, however its not a unified shader GPU. Each core is an ALU but half of them are used for vertex shaders and the other half are for pixel shaders. You can expect the GeForce ULV line to take a similar evolutionary path to desktop GeForce in the future (meaning it’ll eventually be a unified shader architecture).
The four vertex shader cores/ALUs can do a total of 4 MADDs per clock, the same is true for the four pixel shader ALUs (4 MADDs per clock).

The GeForce ULV in NVIDIA’s Tegra 2 runs at a minimum of 100MHz but it can scale up to 400MHz depending on the SoC version:

<table>
<thead>
<tr>
<th>SoC</th>
<th>Part Number</th>
<th>CPU Clock</th>
<th>GPU Clock</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA Tegra 2</td>
<td>T20</td>
<td>1GHz</td>
<td>333MHz</td>
<td>Now</td>
</tr>
<tr>
<td>NVIDIA Tegra 2</td>
<td>AP20H</td>
<td>1GHz</td>
<td>300MHz</td>
<td>Now</td>
</tr>
<tr>
<td>NVIDIA Tegra 2 3D</td>
<td>T25</td>
<td>1.2GHz</td>
<td>400MHz</td>
<td>Q2 2011</td>
</tr>
<tr>
<td>NVIDIA Tegra 2 3D</td>
<td>AP25</td>
<td>1.2GHz</td>
<td>400MHz</td>
<td>Q2 2011</td>
</tr>
</tbody>
</table>

The AP20H runs at up to 300MHz, while the tablet version runs at a faster 333MHz.

Architecturally, the GeForce ULV borrows several technologies that only recently debuted on desktop GPUs. GeForce ULV has a pixel cache, a feature that wasn’t introduced in GeForce on the desktop until Fermi. This is purely an efficiency play as saving any trips to main memory reduces power consumption considerably (firing up external interfaces always burns watts quicker than having data on die).

NVIDIA also moved the register files closer to the math units, again in the pursuit of low power consumption. GeForce ULV is also extremely clock gated although it’s not something we’re able to quantify.

NVIDIA did reduce the number of pipeline stages compared to its desktop GPUs by a factor of 2.5 to keep power consumption down.

The GeForce ULV supports Early Z culling, a feature first introduced on the desktop with G80. While G80 could throw away around 64 pixels per clock, early Z on GeForce ULV can throw away 4 pixels per clock. While early Z isn’t the equivalent of a tile based renderer, it can close the efficiency gap between immediate mode renderers and TBRs.
messages. It’s nothing new from a UI perspective—keep the thread drawing the screen active and speedy while longer processes run in the background. The even

multithreaded scenario on Android, the main thread communicates to child threads using a handler class, and hums along until they come back with results and delivering speedy UI. The idea is to render the UI using one thread and distribute slow tasks into a background threads as necessary. In the best case

Before we start that discussion however, we need to talk about multithreading in Android. Android itself already is multithreaded natively, in fact, that’s part of

Obviously now we don’t have such limitations, and won’t keep you waiting any longer to see how Tegra 2 compares to other phones we’ve benchmarked already.

When I sat down with the 2X at CES, naturally the first thing we did was run our usual suite of benchmarking tools on the phone. The phone was running Android 2.2.1 then, and even though the numbers were good, LG hadn’t quite finalized the software and didn’t think those numbers were representative. We didn’t publish,

2.2.1 then, and even though the numbers were good, LG hadn’t quite finalized the software and didn’t think those numbers were representative. We didn’t publish,

2.2.1 then, and even though the numbers were good, LG hadn’t quite finalized the software and didn’t think those numbers were representative. We didn’t publish,
better part is that multiprocessor smartphones can immediately take advantage of multiple cores and distribute threads appropriately with Android. That said, Android 3.x (Honeycomb) brings a much lighter focus on multithreading and bringing things like garbage collecting off of the first CPU and onto the second. In case you haven't figured it out by now, Android releases generally pair with and are tailored to a specific SoC. If you had to assign those, it'd look something like this: 2.0-2.1—TI OMAP3, 2.2—Qualcomm Snapdragon, 2.3—Samsung Hummingbird, 3.0—Tegra 2.

Back to the point however, the same caveats we saw with multithreading on the PC apply in the mobile space. Applications need to be developed with the expressed intent of being multithreaded to feel faster. The big question on everyone’s mind is whether Android 2.2.x can take advantage of those multiple cores. Turns out, the answer is yes.

First off, we can check that Android 2.2.1 on the 2X is indeed seeing the two Cortex-A9 cores by checking dmesg, which thankfully is quite easy to do over adb shell after a fresh boot. Sure enough, inside we can see two cores being brought up during boot by the kernel:

```
<4>[  118.962880] CPU1: Booted secondary processor
<6>[  118.962989] Brought up 2 CPUs
<6>[  118.963003] SMP: Total of 2 processors activated (3997.69 BogoMIPS).
<7>[  118.963025] CPU0 attaching sched-domain:
<7>[  118.963036]  domain 0: span 0-1 level CPU
<7>[  118.963046]   groups: 0 1
<7>[  118.963063] CPU1 attaching sched-domain:
<7>[  118.963072]  domain 0: span 0-1 level CPU
<7>[  118.963079]   groups: 1 0
<6>[  118.966500] regulator: core version 0.5
```

The 2X runs the same 2.6.32.9 linux kernel common to all of Android 2.2.x, but in a different mode. Check out the first line of dmesg from the Nexus One:

```
<5>[ 0.000000] Linux version 2.6.32.9-27240-gbca5320 (android-build@apa26.mtv.corp.google.com) (gcc version 4.4.0 (GCC) ) #1 PREEMPT Tue Aug 10 16:42:38 PDT 2010
```

Compare that to the 2X:

```
<5>[ 0.000000] Linux version 2.6.32.9 (sp9pm_9@sp9pm2pl3) (gcc version 4.4.0 (GCC) ) #1 SMP PREEMPT Sun Jan 16 20:58:43 KST 2011
```

The major difference is the inclusion of “SMP” which shows definitively that Symmetric Multi-Processor support is enabled on the kernel, which means the entire platform can use both CPUs. PREEMPT of course shows that kernel preemption is enabled, which both have turned on. Again, having a kernel that supports multithreading isn’t going to magically make everything faster, but it lets applications with multiple threads automatically spread them out across multiple cores.

Though there are task managers on Android, seeing how many threads a given process has running isn’t quite as easy as it is on the desktop, however there still are ways of gauging multithreading. The two tools we have are both checking “dumpsys cpuinfo” from over adb shell, and simply looking at the historical CPU use reported in a monitoring program we use called System Panel which likely looks at the same thing.

The other interesting gem we can glean from the dmesg output are the clocks NVIDIA has set for most of the interesting bits of Tegra 2 in the 2X. There’s a section of output during boot which looks like the following:

```
<4>[ 119.026337] ADJUSTED CLOCKS:
<4>[ 119.026354] HC clock is set to 300000 KHz
<4>[ 119.026365] EMC clock is set to 600000 KHz (DDR clock is at 300000 KHz)
<4>[ 119.026373] PLLXO clock is set to 1000000 KHz
<4>[ 119.026379] PLLCO clock is set to 600000 KHz
<4>[ 119.026385] CPU clock is set to 1000000 KHz
<4>[ 119.026391] System and AVP clock is set to 240000 KHz
<4>[ 119.026400] GraphicsHost clock is set to 100000 KHz
<4>[ 119.026408] 3D clock is set to 100000 KHz
<4>[ 119.026415] 2D clock is set to 100000 KHz
<4>[ 119.026423] Epp clock is set to 100000 KHz
<4>[ 119.026430] Hpe clock is set to 100000 KHz
<4>[ 119.026436] Vde clock is set to 240000 KHz
```

We can see the CPU set to 1 GHz, but the interesting bits are that LPDDR2 runs at 300 MHz (thus with DDR we get to 600 MHz), and the GPU is clocked at a relatively conservative 100 MHz, compared to the majority of PowerVR GPUs which run somewhere around 200 MHz. It turns out that AP20H lets the GPU clock up to 300 MHz under load as we'll discuss later. The other clocks are a bit more mysterious, Vde could be the Video Decode Engine, Mpe could be Media Processing Engine (which is odd since Tegra 2 uses the A9 FPU instead of MPE), the others are even less clear.

The other interesting bit is how RAM is allocated on the 2X—there’s 512 MB of it, of which 384 MB is accessible by applications and Android. 128 MB is dedicated entirely to the GPU. You can pull that directly out of some other dmesg trickery as well:

```
mem=384M@0M nvmem=128M@384M
```

The first 384 are for general RAM, the remaining 128 MB is NVIDIA memory which we can only assume is dedicated entirely to the GPU.
Performance Analysis

Android’s browser is multithreaded, but again certain workloads and pages lend themselves to parallelization much better than others. When it comes to our page load suite, the 2X is almost class leading, though our results are starting to get very cramped and clustered among the same grouping of Android versions. All of the Android browsers in these tests have Flash enabled as well. Anand and I are working on setting up a much more comprehensive and sophisticated page load test with much better repeatability, as the current version has become a bit unwieldy. Expect to see that soon, but for now we’re using the old method.

Panning and zooming on the 2X is speedy, even if it’s still choppier than the Galaxy S’ hardware accelerated browser. What’s really noticeable on the 2X is how fast Adobe Flash is. Generally while browsing I can feel when Flash ads are really slowing a page down—the 2X almost never felt that way. The 2X was running Flash version 10.1.120.10 which is newer than the version available in the market at current writing. The Flash version in the market has a number of warnings cautioning against installing the current market version on the 2X or a Tegra 2 powered phone as only the preloaded version has Tegra 2 optimizations. Thankfully these are being rolled into the next market release so there won’t be any Flash version fragmentation.
We tested with GUIMark2’s mobile Flash charting test which has a nice framerate counter and test mode. The results speak for themselves, Flash feels dramatically faster on here than any other platform we’ve tested. We still don’t have a Galaxy S device with 2.2 officially on it, and therefore don’t have a comparison from Hummingbird-based phones.

You can see how snappy the browser is (including page load, pan and zoom speed, and Flash performance) at 12:40 in our LG Optimus 2X Overview video:

Moving along is the SunSpider Javascript benchmark, which is a regular fixture in our smartphone reviews. The benchmark measures the performance of various Javascript code snippets designed to represent real world js usage. When run on a smartphone it gives us the idea of how fast the browser, OS and hardware platform (including SoC) is at running these Javascript tests. Good js performance alone isn’t enough to provide a fast web browsing experience, but it’s one component.

Google has been on a Javascript performance optimization rampage since Android’s initial release. You get a hint of that here when you look at the clustering of Android scores. The move from Android 1.6 to 2.1 was responsible for a big jump in performance, and we saw another major improvement with the move to Froyo
The SunSpider benchmark isn’t explicitly multithreaded, although some of the tests within the benchmark will take advantage of more than one core. As a result, some of the performance gain here over a Cortex A8 is due to the out-of-order execution engine and shorter pipeline of the Cortex A9 and not just the two cores.

The Motorola Droid 2 uses a TI OMAP 3620 with a single 1GHz Cortex A8 core. As such it is the best hardware comparison to the LG Optimus 2X with its dual Cortex A9 cores. The performance improvement here is very good. NVIDIA’s Tegra 2 manages a 64% faster score under SunSpider than the OMAP 3620 based Droid 2. The advantage over Samsung’s Hummingbird SoC is narrower but still an impressive 44%.

The comparison to T-Mobile’s myTouch 4G is very telling. The myTouch 4G uses the latest 45nm SoC from Qualcomm. Architecturally this SoC still has the same Scorpion CPU core as previous Snapdragons, but the difference here is the memory interface. The 45nm Snapdragons feature a LP-DDR2 memory interface vs. the LP-DDR1 interface from the 65nm versions. The faster memory interface is likely responsible for the improved performance here.

With a faster memory bus, Qualcomm’s 45nm Snapdragon closes the gap between it and the Optimus 2X’s Tegra 2. NVIDIA only holds a 12.5% performance advantage here vs. Qualcomm’s single core 45nm Snapdragon. A 1.2GHz dual-core Snapdragon (the 8660) would likely outperform NVIDIA’s Tegra 2 in this test.

In terms of currently available hardware however, the LG Optimus 2X is clearly the fastest device we have running SunSpider. And NVIDIA’s Tegra 2 is the fastest SoC in our SunSpider benchmarks.

Rightware’s BrowserMark is another JavaScript performance benchmark. Rightware modeled its benchmark after the JavaScript frameworks and functions used by websites like Facebook, Amazon and Gmail among others. The results are simply one aspect of web browsing performance, but an important one:

BrowserMark shows the Optimus 2X with a 57% performance advantage over the Motorola Droid 2. The performance advantage is similar to what we saw in our SunSpider results. The gap doesn’t close by much as we look at Samsung Hummingbird and Qualcomm Snapdragon based SoCs. NVIDIA and LG maintain a 41.5% performance advantage over the 45nm Snapdragon based myTouch 4G.

It’s still difficult to draw real conclusions about why some SoCs perform better than others, however in this case I’d guess that it’s probably equal parts A9 and dual-core giving the Tegra 2 its performance advantage in BrowserMark.

The combination of these two js benchmarks proves one point: the LG Optimus 2X and NVIDIA’s Tegra 2 provide the fastest Android web browsing experience we’ve seen thus far. A message that is continuously echoed in our day to day use of the phone. The Optimus 2X is the first Android phone to render web pages as quickly, if not quicker, than the iPhone 4.

WiFi performance is often surprisingly bad on smartphones we test for some reason. Thankfully, the LG Optimus 2X did very well in our tests:
At 30.3Mbps, the Optimus 2X all but tied the iPad for local network WiFi performance. It’s not only the fastest Android phone we’ve run through our WiFi test, but also the fastest WiFi we’ve tested on a smartphone in general. Congrats to LG.

One of the original Android 3D benchmarks, Neocore has lost much of its usefulness over the past year. With the move to PowerVR SGX 535/540 and Adreno 205 GPUs in newer SoCs we run into a curious problem—nearly all of the high end GPUs converge at around 60 fps:

This isn't some mystic performance convergence, rather it's the OS-wide frame rate cap both Qualcommm and Imagination Technologies implement on their drivers. Neocore can't physically run any faster because the graphics driver won't allow it. The idea behind the cap is simple. Running at higher frame rates would simply run the GPU faster, which would require data from the CPU more quickly and in turn fire the memory interface more often and just draw a lot more power. Limiting frame rate to 60 fps keeps performance nice and smooth but conserves battery life at the same time.

With a 60 fps limit, we can't use Neocore to tell the difference between GPUs and SoCs capable of running the benchmark at higher frame rates—everything looks the same. So why does the LG Optimus 2X score 74.9 fps? NVIDIA is no stranger to benchmarking—it simply raised the frame rate limit to 117.5 fps across the OS, and as a result we get a meaningful score in Neocore from at least one high end GPU.

Unfortunately this tells us nothing about how Tegra 2 stacks up against the PowerVR SGX 535/540 or Qualcomm's Adreno 205. The only things we can conclude are: 1) NVIDIA is going to eat the competition for breakfast once more smartphone benchmarks are available and we start playing optimization games, and 2) Neocore is a meaningless benchmark and should be retired.

This next benchmark needs no introduction. Quake 3 graced the pages of AnandTech years ago, serving as a great GPU and eventually an even better CPU benchmark as our systems grew to be more powerful. While hardly representative of the type of 3D games that are being ported to smartphones today, Quake 3 remains an interesting measure of GPU and SoC performance.
NVIDIA's Tegra 2 is clearly faster than Qualcomm's MSM7230, however it plays second fiddle to the PowerVR SGX 540 in Samsung's Hummingbird. The why is difficult to explain. Quake 3 is a fairly simple game by today's standards. There are several layers of texturing but no pixel or vertex shaders and very little geometry. Running on modern hardware, Quake 3 is a GPU texturing/ROP test and a triangle setup test. Performance here should be influenced by the memory interface, cache performance, CPU-GPU interface as well as the GPU's ability to churn out textured pixels. I suspect that NVIDIA's driver and GPU itself just aren't optimized for this type of a game. The performance is decent, but it's no faster than a PowerVR SGX 540.

The Cortex A8's FPU wasn't pipelined and thus presented serious performance limitations when used. The Cortex A9 corrects this and thus we see a huge performance increase in Linpack. Qualcomm however appears to have always had a pretty decent FP setup in its Scorpion core, resulting in comparable performance to what NVIDIA delivered with Tegra 2.

It's unclear to us how much of the Linpack Android port uses VFP code vs. NEON code. Based on the results here it would appear to be a VFP benchmark, not touching NEON at all. In addition, the Linpack test we use from the Android market appears to be single-threaded.

**Quadrant**

Quadrant is one of the unfortunate realities of Android smartphone benchmarking today: we have a lot of synthetic benchmarks that give us relatively little insight into real world app performance. While I expect the smartphone benchmarking tools to improve over the coming 24 months, we'll have to make do for now.
Quadrant provides an overall score as well as individual category scores. As you'd expect, NVIDIA's Tegra 2 is on top. CPU performance, as measured in Quadrant, is 38% faster than the fastest SoCs on the market today. Given that this is a purely synthetic test I'd expect to see smaller gains in real world apps. Quadrant's CPU benchmark, like linpack, also is just a single thread:

What is surprising is just how well the Tegra 2 does in Quadrant's memory and I/O benchmarks:
The Quadrant developers aren't very open about what these tests do other than obviously stress the memory interface and stream data over the I/O bus. NVIDIA has claimed that it has the best memory controller and arbitration logic of any of the SoC players. These advantages could be the cause of NVIDIA's solid showing in Quadrant's memory and I/O tests.
GLBenchmark 2.0—as its name implies—tests OpenGL ES 2.0 performance on compatible devices. The suite includes two long benchmarking scenarios with a demanding combination of OpenGL ES 2.0 effects, and individual tests such as swap buffer speed (for determining the framerate cap), texture fill, triangle, and geometric tests. GLBenchmark 2.0 also leverages texture based and direct lighting, bump, environment, and radiance mapping, soft shadows, vertex shader based skinning, level of detail support, multi-pass deferred rendering, noise textures, and ETC1 texture compression.

GLBenchmark 2.0 is the best example of an even remotely current 3D game running on Android—and even then this is a stretch. If you want an idea of how the Tegra 2 GPU stacks up to the competition however, GLBenchmark 2.0 is probably going to be our best bet (at least until we get Epic to finally release an Unreal Engine benchmark for Android).
The first benchmark, Egypt, tests OpenGL ES 2.0 and represents the newest and most demanding benchmark. The second—GLBenchmark PRO—also tests OpenGL ES 2.0 features, and is a port of an earlier GLBenchmark 1.1 test which focused on OpenGL ES 1.1 performance.

<table>
<thead>
<tr>
<th>GLBenchmark 2.0 - Egypt</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenGL ES 2.0 Test - Native Resolution - FPS (higher is better)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Optimus 2X (2.2.1 - 1 GHz Tegra 2 - ULP GeForce)</td>
<td>25.2</td>
</tr>
<tr>
<td>Samsung Fascinate (2.1.1 GHz Hummingbird- SGX 540)</td>
<td>21.5</td>
</tr>
<tr>
<td>Nexus S (2.3.1 GHz Hummingbird- SGX 540)</td>
<td>20.9</td>
</tr>
<tr>
<td>T-Mobile G2 (2.2.2 - 800 MHz MSM7230- Adreno 205)</td>
<td>17.3</td>
</tr>
<tr>
<td>myTouch 4G (2.2.1 GHz MSM8255- Adreno 205)</td>
<td>16.3</td>
</tr>
<tr>
<td>iPhone 3GS (4.2.1- 600 MHz S5PC100- SGX 535)</td>
<td>14.5</td>
</tr>
<tr>
<td>Optimus One (2.2.2 - 600 MHz MSM7277- Adreno 200)</td>
<td>9.8</td>
</tr>
<tr>
<td>HTC EVO 4G (2.2.1 - 1 GHz QSD8650- Adreno 200)</td>
<td>7.5</td>
</tr>
<tr>
<td>Nokia N900 (600 MHz OMAP 3430- SGX 530)</td>
<td>7.3</td>
</tr>
<tr>
<td>Nexus One (2.2.1 - 1 GHz QSD8250- Adreno 200)</td>
<td>7.1</td>
</tr>
<tr>
<td>Motorola Droid (2.2.2 - 550 MHz OMAP 3430- SGX 530)</td>
<td>6.9</td>
</tr>
<tr>
<td>iPhone 4 (4.2.1- ~850 Mhz Apple A4- SGX 535)</td>
<td>5.9</td>
</tr>
<tr>
<td>iPad (4.2.1- 1 GHz Apple A4- SGX 535)</td>
<td>4.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GLBenchmark 2.0 - PRO</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenGL ES 2.0 PRO - Native Resolution - FPS (higher is better)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Optimus 2X (2.2.1 - 1 GHz Tegra 2 - ULP GeForce)</td>
<td>51.2</td>
</tr>
<tr>
<td>Nexus S (2.3.1 GHz Hummingbird- SGX 540)</td>
<td>41.9</td>
</tr>
<tr>
<td>Samsung Fascinate (2.1.1 GHz Hummingbird- SGX 540)</td>
<td>40.7</td>
</tr>
<tr>
<td>myTouch 4G (2.2.1 GHz MSM8255- Adreno 205)</td>
<td>25.2</td>
</tr>
<tr>
<td>T-Mobile G2 (2.2.2 - 800 MHz MSM7230- Adreno 205)</td>
<td>24.5</td>
</tr>
<tr>
<td>iPhone 3GS (4.2.1- 600 MHz S5PC100- SGX 535)</td>
<td>24.5</td>
</tr>
<tr>
<td>iPhone 4 (4.2.1- ~850 Mhz Apple A4- SGX 535)</td>
<td>16.2</td>
</tr>
<tr>
<td>Nokia N900 (600 MHz OMAP 3430- SGX 530)</td>
<td>12.8</td>
</tr>
<tr>
<td>iPad (4.2.1- 1 GHz Apple A4- SGX 535)</td>
<td>12.7</td>
</tr>
<tr>
<td>Motorola Droid (2.2.2 - 550 MHz OMAP 3430- SGX 530)</td>
<td>11.3</td>
</tr>
<tr>
<td>Optimus One (2.2.2 - 600 MHz MSM7277- Adreno 200)</td>
<td>9.0</td>
</tr>
<tr>
<td>Nexus One (2.2.1 - 1 GHz QSD8250- Adreno 200)</td>
<td>5.8</td>
</tr>
<tr>
<td>HTC EVO 4G (2.2.1 - 1 GHz QSD8850- Adreno 200)</td>
<td>5.8</td>
</tr>
</tbody>
</table>

In both OpenGL ES 2.0 benchmarks NVIDIA’s Tegra 2 comes out ahead of PowerVR’s SGX 540. Granted it’s impossible to know how much of this is driver optimization for the benchmark vs. pure hardware capability, but I’d expect that these results are reasonably believable. NVIDIA’s Tegra 2 looks to be 15—25% faster than the PowerVR SGX 540 based Samsung Hummingbird SoC. Not all of this is going to be due to the GPU as even the most taxing games run on smartphones are still quite CPU bound at times.

BaseMark GUI Benchmark
Basemark divides benchmarking into two categories—feature tests and the final overall user interface test. We couldn’t get the UI test to run reliably on the Optimus 2X so we’ll be looking at the feature tests exclusively. These are named animation, vertex streaming, texture streaming, and alpha blending as shown in the menu above. Each tests a different collection of what Rightware believes are representative graphics tasks that will be leveraged in user interfaces moving forward.

First up among those feature tests is animation, which animates a graphics element (in this case, a robot moving through a set of actions) by stepping through a table of keyframes and interpolating the character's movement between using splines. This particular benchmark uses ES 2.0 APIs and per vertex lighting, but the purpose of this test is to be as CPU-bound as possible and specifically stress floating point performance.

NVIDIA’s Tegra 2 has no SIMD FP engine (NEON) and instead relies on ARM’s first fully pipelined FPU. The result is competitive performance to the NEON equipped Hummingbird and Snapdragon SoCs, but not better performance. NVIDIA told us that to implement NEON would be to incur a 30% die penalty for a performance advantage that would only impact a small portion of your code. It remains to be seen how much the NEON tradeoff will actually cost NVIDIA this generation. If BaseMark’s Animation benchmark is any indication, the penalty may not be all that bad.

Next are the vertex and texture streaming tests, both of which measure asset streaming performance—according to RightWare, effectively memory bandwidth. The vertex test loads geometry data into GPU memory and frees it when no longer needed. The scene itself involves a lot of geometry—ten repeating city blocks which the camera moves through with increasing speed and draw distance. The test ramps from around 3k vertices to 15k vertices per frame, and 190k to 250k triangles per frame. There’s a single texture material, fog, two lights, and OpenGL ES 2.0 shaders which use per vertex lighting.
The Vertex IO test is an important test of performance as it measures the CPU-GPU interface bandwidth, vertex shader performance and memory bandwidth. NVIDIA does well here but and remains competitive with Samsung & Qualcomm, although definitely not faster.

The texture test is a bit more straightforward, quickly loading images from RAM into the GPU memory and discarding them.

These asset streaming tests effectively test memory bandwidth from disk into RAM and into GPU memory. Although Kanzi supports ETC texture compression (and Basemark exposes this option for testing), we’ve initially tested with the default configuration which has ETC texture compression disabled.
Quadrant implied that NVIDIA's paths to NAND and main memory were both ridiculously quick. BaseMark's Texture IO test corroborates the implication. The Optimus 2X manages a score here that's nearly twice that of the closest competitor. A fast memory and I/O interface never hurts. I am curious to know how much of the performance here is due to NVIDIA's new ROPs that reuse pixel shader hardware.

Blend testing—as its name implies—tests alpha blended rendering by drawing a number of semi-transparent contact cards atop each other. These are overlaid sequentially until we reach a desired number of layers. This test actually runs twice, first with front to back ordering of these contact cards, and then with back to front ordering.

This ordering difference shouldn't be confused with the fact that feature test actually runs in both back to front and front to back rendering orders and are combined later.
Next is the composition test, which composites interface screens rendered off-scene into one large tiled user interface. The example here are applications rendering in the background being composited into one view. This test combines 16 off-scene objects.

![Composition Test](image)

The composition test puts NVIDIA's Tegra 2 and the Optimus 2X at the top of the chart. As a result, Android UI performance feels very snappy running on Tegra 2. All we need now is Gingerbread.

### BaseMark GUI Benchmark - Composition

<table>
<thead>
<tr>
<th>Device</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Optimus 2X (2.2.1 - 1 GHz Tegra 2 - ULP GeForce)</td>
<td>23.38</td>
</tr>
<tr>
<td>myTouch 4G (2.2 - 1 GHz MSM8255 Adreno 205)</td>
<td>23.74</td>
</tr>
<tr>
<td>Nexus S (2.3.1 - 1 GHz Hummingbird SGX540)</td>
<td>22.42</td>
</tr>
<tr>
<td>Epic 4G (2.1 - 1 GHz Hummingbird SGX540)</td>
<td>21.68</td>
</tr>
<tr>
<td>G2 (2.2 - 800 MHz MSM7230 Adreno 205)</td>
<td>20.11</td>
</tr>
<tr>
<td>Galaxy Tab (2.2 - 1 GHz Hummingbird SGX540)</td>
<td>18.74</td>
</tr>
<tr>
<td>LG Optimus One (2.2 - 600 MHz MSM7227 Adreno 200)</td>
<td>12.62</td>
</tr>
<tr>
<td>Dell Streak (2.2 - 1 GHz QSD8250 Adreno 200)</td>
<td>10.29</td>
</tr>
<tr>
<td>Nexus One (2.2.1 - 1 GHz QSD8250 Adreno 200)</td>
<td>10.1</td>
</tr>
<tr>
<td>HTC EVO 4G (2.2 - 1 GHz QSD8650 Adreno 200)</td>
<td>9.64</td>
</tr>
</tbody>
</table>

Page 9

**Baseband and Antenna**

After I got the 2X out of the box and removed its protective shipping plastic, I popped the back cover off, ready to insert a SIM. It struck me as odd at the time, but the 2X had another piece of protective plastic over another region at the bottom covering the antenna traces at the very bottom. I removed that, but there was another piece of blue protective plastic which also looked like it was intended to come off. After getting those up, it was very obvious that the 2X has the cellular antennas at the bottom.
I took the 2X apart, but before I even did so I figured out what radio hardware was inside. The baseband in the 2X is actually pretty interesting—it's the Infineon X-Gold XG616 baseband which is from the same family as what's in the iPhone 4 (which uses an X-Gold 618), and also the same as what's in the Samsung i9000 (Galaxy S) and Nexus S. It supports HSDPA 7.2 Mbps and HSUPA 5.7 Mbps as well as GSM and EDGE class 33. Before I found out what baseband was in the 2X, I noticed a similar level of receive sensitivity to the iPhone 4 in areas I'm familiar with that are troublesome for other devices. Anecdotally this behavior confirms our earlier stipulations that the iPhone 4 baseband is much better than its predecessors in low-signal areas. The 2X fares equally as well, with the advantage of being totally same deathgrip and in line with the 15 or so dB of attenuation from cupping the phone right over the antennas:

<table>
<thead>
<tr>
<th></th>
<th>Cupping Tightly</th>
<th>Holding Naturally</th>
<th>On an Open Palm</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG Optimus 2X</td>
<td>13.7</td>
<td>9.3</td>
<td>5.9</td>
</tr>
<tr>
<td>Nexus S</td>
<td>13.3</td>
<td>6.1</td>
<td>4.3</td>
</tr>
<tr>
<td>Droid 2</td>
<td>11.5</td>
<td>5.1</td>
<td>4.5</td>
</tr>
<tr>
<td>BlackBerry Torch</td>
<td>15.9</td>
<td>7.1</td>
<td>3.7</td>
</tr>
<tr>
<td>Dell Streak</td>
<td>14.0</td>
<td>8.7</td>
<td>4.0</td>
</tr>
<tr>
<td>Droid X</td>
<td>15.0</td>
<td>5.1</td>
<td>4.5</td>
</tr>
<tr>
<td>iPhone 4</td>
<td>24.6</td>
<td>19.8</td>
<td>9.2</td>
</tr>
<tr>
<td>iPhone 3GS</td>
<td>14.3</td>
<td>1.9</td>
<td>0.2</td>
</tr>
<tr>
<td>HTC Nexus One</td>
<td>17.7</td>
<td>10.7</td>
<td>6.7</td>
</tr>
</tbody>
</table>

LG disclosed what bands the 2X we were sampled tunes, which include WCDMA (3G) support for PCS 1900 and AWS 2100, in addition to GSM 900. PCS 1900 explains why the AT&T SIM I popped in works, in my market all of AT&T is PCS 1900 (UMTS band 2). Even better, you can confirm that fact by poking around inside the 2X’s excellent hidden menus under the modem’s engineering area.

The absence of 850 MHz support for AT&T would make T-Mobile the obvious conclusion if the 2X is coming to the states in this exact same modem configuration. Even though HSDPA 7.2 is still decently fast, it’d obviously be a disappointing to not see a slightly faster baseband sneak in before launch, especially given T-Mobile’s aggressive HSPA+-is-4G marketing storm. At this point, however, 7.2 Mbps looks like what’s coming.
Maximum speeds on the 2X were around 3 Mbps down, and 1.5 Mbps up. I ran well over 200 speedtests before I factory reset the device (more on why later), which I unfortunately did that wipe before exporting that data. In the few days after wiping, I ran over 120 speedtests got an average of 2.1 Mbps down, 1.02 Mbps up. I’d say that's relatively standard for AT&T here. Regardless, in the time I spent with the 2X the speeds I got were typical of what I’m used to seeing on the iPhone 4 and Nexus One on AT&T 3G. I’ve yet to play with an HSPA+ device on AT&T for comparison. I didn’t experience any erratic handing off between EDGE and HSPA either, all in all the cellular part of the 2X seems solid.

Likewise, calls on the 2X sounded good on the earpiece. On speaker, there’s some distortion and a little saturation at the two highest volume settings. Turning it down one more notch makes it go away and brings audio quality way up subjectively.

The good side of that tradeoff is that the 2X’s speakerphone is very loud, even if by the numbers it isn’t the absolute loudest we’ve tested. No doubt that’s at least partly due to the fact the we put the microphone 6 inches above the display surface.

On the far left side you can see a depressed region which leads out to the meshed speaker port.

This is the speakerphone and antenna assembly, you can see two contacts at the bottom left which make contact near the antenna connector on the top photo. The speaker is to the right below the fine mesh, and aligns with the mixing chamber in the above photo.

Internally, the 2X actually has a nice large audio chamber for the speakerphone port. The speaker is oriented facing out the back, through some mesh, into the mixing chamber, and then ported out through the bottom.

**Disassembly**

The 2X is shockingly simple to take apart—5 Phillips screws sit between the top plastic cover on the backside and the main PCB underneath, three hold the antenna and speaker assembly to the main case. Unscrew them, then the plastic covers easily pop out revealing the interesting goodies underneath.
There are 7 flex connectors that easily pop off. Before you even go any further, it’s obvious where Tegra 2 sits on the board.

After you get those off, the EMI can which doubles as heatspreader lifts off very easily, revealing the Tegra 2 with its package-on-package Hynix LPDDR2. The part is marked “H8TBR00U0MLR-0DM” which corresponds to a 4 gigabit (512 MB) 400 MHz LPDDR2 PoP part. It’s not uncommon to use memory that’s rated for higher frequency operation, in this case the Tegra 2 only runs its memory interface at 300MHz (600MHz data rate). Underneath that is the Tegra 2 AP20H SoC, but obviously it’d be difficult to impossible to remove the RAM stacked on top without making the device non-functional.
There's been some discussion as to whether current smartphone SoCs are already TDP limited, and moreover whether the new round of dual-core parts run noticeably warmer than the previous generation. There's obviously a bit more attention paid to thermals with the 2X, but that beefy EMI can is still nowhere near being a robust copper heatsink. While running our BaseMark GUI battery test which runs 3D endlessly until the battery dies, I took temperature measurements on the exterior using an IR thermometer and saw temperatures between 96 and 98 degrees F at maximum. This is completely on-par with what I measured with all the other smartphones we put through the same test.

To the right is the 2X's baseband and attached Hynx 1 GB of LPDDR1 and NAND combo marked “H8BCS0QG0MMR-46M.” Right below it is a test point, and to the right, the antenna connector. You can see the connector and cable which leads out and down the side of the phone to the antenna assembly at the bottom.

Removing some extra cables and flipping it over, the bottom of the PCB is home to the 8 GB of Toshiba eMMC internal storage. Disassembly further was much more difficult because of some beefy wires connecting the main PCB to the front display assembly.
WiFi Performance

The 2X has a BCM4329 WiFi 802.11n, Bluetooth 2.1 and FM solution onboard. The antenna for 2.4 GHz Bluetooth and WiFi is likely at the top of the device. WiFi paired at 72 Mbps with our 802.11n router. Though the BCM4329 has 5 GHz support, the 2X lacks the radio hardware to enable it, and we only saw 2.4 GHz networks. WiFi range and reliability is almost exactly the same as other similar smartphones, if not slightly better—I make it a few more feet beyond the point most smartphones drop off WiFi. As an aside, it’s amazing that there’s so much consistency here.

As we’ve already shown, WiFi throughput on the 2X is class-leading for some reason, second only to the iPad in our PDF download test. I’m tempted to say that the 2X is class leading in terms of WiFi range just because I can make it those extra couple of feet and still break 2 Mbps running that test before falling off completely.

GPS Performance

The other big consideration for smartphones is GPS. The 2X has the BCM4751 A-GPS radio stack from Broadcom. First off, the 2X shipped to us with WiFi augmented location services turned off. If you fire up maps, it tells you to go enable it, and that works fine and dandy. The problem is that GPS fixes by default take a long, long time.

Remember that awesome hidden menu I mentioned earlier?

There’s a lot buried in there, including a test program for GPS as well as links to the internal GPS configuration. Out of the box, GPS fixes are taking a comparatively long time because they’re cold fixes—the GPS has to search for both almanac data which is broadcast every 12.5 minutes, and ephemeris data which contains the up to date GPS constellation position. What A-GPS does is enable almanac, ephemeris, network time, and rough location information downloaded over the data connection to speed fixes up dramatically, which is what we’re pretty spoiled with most of the time. On average, a proper A-GPS fix should be around 10 seconds. Cold fixes where the phone has to sit and wait take minutes.
From the poking around I've done, I don't believe the 2X we received was properly provisioned to go grab A-GPS data. That sort of makes sense since the phone came with no APN configured, and there's absolutely no hint from LG about what carrier it's headed to. How it is now, the first fix will take a long time, subsequent fixes happen much faster because the phone will have alamac and will only be waiting for ephemeris sent every 30 seconds. The long and short of it is that this is something I fully expect to see properly provisioned on a carrier-backed 2X. SNR isn't a problem on the 2X, it's comparable to other devices, it's all just about speeding that initial time to first fix up.

Page 10

Display

I've already talked a bit about the 2X's display in the intro—it's a WVGA (800x480) IPS panel topped with a capacitive digitizer. Touch detection is nice and snappy, and supports up to 10 contact points simultaneously which was confirmed with some testing using the LG touch screen test application. The same surface is also home to the capacitive buttons just below the display edge. They're responsive and work perfectly fine. In the video review, I struggle on camera with taps a few times, which is purely a function of using the phone at a weird angle some distance away—the buttons are actually very responsive.
Though the 2X display is indeed IPS, it doesn’t score too highly in our display tests. I found contrast to be more than adequate, even if it isn’t AMOLED. At maximum brightness there’s just a bit too much brightness in the blacks which hurts that contrast score. I prefer LCD over AMOLED personally because of how grainy and off-white PenTile looks on most devices, so it’s nice to see an LCD. The automatic brightness dynamic range on the 2X could stand to be way bigger, it doesn’t go as dark as the darkest setting in pitch blackness, or as bright as maximum. That’s more of a complaint about how Android in general handles auto brightness by default than something LG is guilty of doing wrong. It’s a shame we don’t have the same kind of color calibration checks on smartphones as we do displays, because subjectively the display looked just about perfect. Auto brightness also sometimes seems to change brightness abruptly, without a gradual transition.
We talked in the physical impressions area about how the display surface is curved. It’s a gentle curve out at the edge which slopes down to meet flush the plastic lip running around the edge. If you put the phone face down, it seems that these areas are slightly raised, and thus the area that scratches is out at the start of that curve. What’s more interesting though is that several times I’ve noticed that with the sun or bright light at the side, light will enter through the raised curve and totally internally reflect all the way across the surface of the display. The result is that you see a ton of repeated vertical lines across the phone from light reflecting off the front and back surfaces of the glass. It’s a weird effect to describe, but I’ve seen it happen twice now during brief stints outside.

Viewing angles are decent, though there’s a bit more color distortion when viewed from extreme horizontal angles. The slight curve doesn’t really affect viewing angles in that direction—it’s nowhere as extreme as the Dell Venue Pro. Likewise, viewing angles in the vertical direction are great.

Outdoors there’s a bit of glare, including an odd secondary reflection from the second (back) surface of the glass. The result is that when you get glare outdoors, there are two images. One much more visible reflection from the first surface, another fainter image from the second surface. I haven’t really noticed it as much on any other device as I do with the 2X. It isn’t honestly a problem outdoors, just something worth noting. It’s too bad that LG couldn’t toss their super bright “Nova” display on the 2X, but as it stands right now LG has a pretty decent IPS on the phone.

The final interesting thing is an odd option under the display settings page. At the bottom is the option to change the Android display font. There are 7 fonts options, including the default Droid Sans font, two other sans-serif fonts (one of which appears monospace), three nightmarish script fonts, and a serif font.

It’s definitely not something I’d change yet (mostly because the alternatives to Droid Sans are visual atrocities), but this is the first time I’ve seen font changing on Android outside of the rooted/modding crowd.

Page 11
Camera Analysis

The 2X has a 1.3 MP front facing camera and 8 MP rear camera, unfortunately I haven’t been able to determine the camera source, but pixels on the rear camera are 1.4 µm square, and 1.75 µm on the front facing camera. That pixel size would make the 2X’s CMOS sensor most likely 1/3.2” in size.
LG’s camera app is familiar territory coming from other LG Android phones. We’ve still got the LG Optimus One, which has almost the exact same layout and design camera application. Honestly, I think LG’s got the best menus I’ve seen yet for a smartphone. There are image size settings for 8 MP (3264x2488), 5 MP, 3 MP, 2 MP, 1 MP, VGA and QVGA, three compression settings (Super Fine, Fine, and Normal), some optional focus controls (Auto, Macro, Face Tracking, and Manual Focus), ISO (Auto, 100, 200, 400, 800), scene modes, effects, ability to disable the shutter sound, and the shot mode. There’s the option of recording images either on the “internal SD card” (which is really that eMMC) or an external SD card if you’ve got one inserted. I noticed that writing to internal storage enabled a bit faster cycling between images compared to my SD card. There’s a full overview of all the camera and video capture settings in the gallery below.

Shot mode is interesting because under it is a setting called “continuous shot” which enables 6 photos to be captured in rapid succession, which you can see in action in our overview video. When switching to that mode, image size drops from 8 MP to a maximum of 2 MP, and a couple of other options grey out. Tapping the capture button immediately takes those 6 images in rapid-fire succession and saves them. It’s cool that you can take a bunch of images quickly, but what would be more useful is being able to mash the capture button in the normal shooting mode and capture as quickly as you would on a decent DSLR. That still isn’t possible, but we’re clearly getting there. NVIDIA advertises that its image signal processor (ISP) is capable of doing 12 MP at 11 FPS, and JPEG encoding at 80 MP/s, which would mean that (negating integration time on the camera sensor) 10 FPS shooting should be possible. That’s obviously a bit ambitious, and isn’t what we see here—maybe someday though.

The OSD LG has put together for the camera is again probably the best I’ve seen on any Android—heck smartphone in general—yet. Icons rotate between landscape and portrait orientation modes, up at the top are shots remaining, image size, and other iconography for modes that have been set elsewhere in settings. Tapping or waiting a few seconds eliminates everything save the live preview to give a less cluttered view for image composition. The 2X lacks tap to focus and expose, instead it’s always center-weighted. There’s also no dedicated capture button on the 2X, so the on-screen button is the only option.

Image quality from photos taken with the rear facing camera is actually pretty good. There’s a lot of spatial detail in our test images with surprisingly little noise. I’d say that the 2X actually takes some of the best 8 MP smartphone images I’ve seen to date. The problem, however, is saturation. Colors are almost universally under-saturated by default—compare almost any of the shots we took at our bench location to other devices, and it’s readily apparent. It’s still outclassed by phones like the Nokia N8, but not bad.

With the lights off, the 2X runs autofocus with the LED lit up, which is awesome. Unfortunately, the flash exposure over illuminates our test scene and creates a very overexposed image. Otherwise the LED flash is nice and powerful, yet more proof that having two isn’t necessarily better.

Interestingly enough the other strong suit of the 2X is that the preview image is one of the most fluid and high framerate we’ve come across. In situations with adequate light, the rear facing camera preview is easily higher framerate than the iPhone 4. Oddly enough, the front sensor preview framerate is quite low unless
you’re imaging something extremely well-lit. The limitation on the front-facing camera framerate is one of integration time on the sensor rather than ISP bandwidth, however.

Tapping the top left button switches to the front facing camera. What’s odd however is that the front facing camera is actually rotated 90 degrees. That’s not to say that the image is rotated 90 degrees, but rather that the longer side of the sensor is parallel to the shorter axis of the phone. The result is that there are black bars on both sides of the preview. It’s as if the sensor was aligned with the intention of being used with the phone primarily in portrait—instead of orienting the sensor to match the aspect ratio of the phone. It’s just an odd choice considering all the other smartphones we’ve looked at thus far are the other way around.

The 1.3 MP front facing camera can capture at 1280x960, VGA, and QVGA resolution. Quality isn’t too good—there’s noticeable lack of detail and blurring in our test image. It’s likely more than enough quality for a video chat that’s going to crop and decimate image size to a much smaller size. The 2X also mirrors images horizontally on the front facing camera.

Page 12

Video Capture

The other big part of the 2X is that it’s the first smartphone to do H.264 1080P video capture. We took the 2X out to our usual test site and recorded video on the 2X at every quality setting at 1080p, and at maximum quality at 720P and VGA resolutions, and one final video with the front facing camera. I looked at the videos and then had Ganesh, our resident media center and video expert, as well as Anand take a look at the same original videos and compare to our other devices. It’s hard to argue that the iPhone 4 and Nokia N8 aren’t the devices to beat, both of them cranking out impressively sharp 720P video. We’ve done the usual thing and uploaded all the test videos to YouTube in addition to making a big zip for comparison in their original glory—links are in the table below.
Before we get to our comparison, a little background. First off, the 2X records 1920x1088 video in H.264 Baseline profile at an average of around 12 Mbps, audio is 1 channel AAC at 64 Kbps. The specifications for the 2x say 1080p24, in practice I’ve seen some framerate variability between 24 and 30 depending on lighting conditions. These videos are close to but not exactly 30 FPS, two videos I shot with the 2X at CES are clearly 24 FPS. Why the extra 8 pixels of vertical resolution, you might be wondering? The reason is simple—1088 is an even factor of 16, and macroblocks are 16x16 pixels.

LG Optimus 2X Video Capture Samples

Rear Facing 8 MP Camera

1080P—SuperFine
1080P—Fine
1080P—Normal

720P—SuperFine

Front Facing 1.3 MP Camera

480P—SuperFine
VGA—SuperFine

LG Optimus 2X vs iPhone 4 at 720P
Mashup—YouTube, MP4 (zip), iPhone 720P (zip)

LG Optimus 2X Original Videos

Original Videos (153.6 MB zip)

So how does 1080p24 video shot on the 2X compare to the iPhone 4 and Nokia N8? Unfortunately, not all that well. At 1080P there's noticeable softness and loss of high spatial frequency detail. At about the 3 second mark in the first video I took (1080p at Super Fine) there’s also some noticeable glare from light flaring off of the glass surface between the camera’s last vertex and the plastic battery door. It’s that kind of stuff that’s a bit frustrating to still see going on with smartphones. The video has noticeable macro-blocking artifacts in the dark regions as well, which is disappointing. Though the Tegra 2 ISP is competent as shown by still image quality, clearly the video encode engine needs a bit more work. SuperFine as we already mentioned corresponds to around 12 Mbps, Fine corresponds to 8.5 Mbps, Normal quality seems to hover around 6 Mbps. You can fit a little over an hour of SuperFine quality 1080P video on the user-accessible 6 GB partition of the 2X’s 8 GB internal storage.

The obvious comparison really is at 720P, where we can directly compare the 2X’s video quality to the N8 and iPhone 4. I don’t have the N8 anymore, our comparison video is still what’s in bench. I do still have an iPhone 4, and captured a video taken at the exact same time as the 2X held carefully above the other phone. You can view both for yourself or compare with a mashup I put together showing both at the same time. The video I made showing both has a bit of downscaling and is at 30 FPS (so the 2X video occasionally looks like it’s dropping frames when it really isn’t), but still illustrates the differences.

Watching both at the same time, it’s readily apparent that the iPhone 4 does a noticeably better job with high frequency spatial detail, where the 2X seems to have softening. The 2X does do a better job with the dark areas of the intersection when panning back, but there’s still macroblocking visible. It’s obvious that there’s a combination of encoder and optics holding the 2X back from having dramatically higher quality video.
The 2X in the form we reviewed it was running Android 2.2.1 and a custom LG skin. Eventually the 2X will get a major software update that brings it up to 2.3. There are some design elements which are common between the 2X and the other LG Android phone we've tested, the LG Optimus One. Those similarities are the way applications are grouped, inclusion of power options in the notifications shade, but pretty much stops right there.

Most of the LG skin is actually pretty tasteful. Status icons for signal, battery, and network status are colorful and non-stock, and the color-on-black theme is vaguely reminiscent of Android 2.3. One of the things I’m a fan of is that LG made the network status indicator show H for “HSPA,” and 3G for UMTS—there’s a distinction between those two network modes that stock Android doesn’t draw unless you dig down into “About” or use a widget.

The theme emulates some of iOS in a rather annoying fashion, including a four-icon dock at the bottom that persists across all pages and isn’t readily user-customizable. Phone, contacts, messaging, and applications are what always are going to be at the bottom. Even if you bring up the applications launcher, those elements persist, though “applications” changes to “home.”

The launcher organizes applications in a slightly odd way with two discrete categories. System applications include those that come preinstalled as part of LG’s preload. You can’t change or remove these, they’re special, and there are 43 of them. I’m actually a bit put-off by how much bloat there is here, including “browsing protection” and another separate “f-secure” mobile security application, all of which can’t be removed. I was hoping that “preloaded apps” would let me uninstall preloaded apps, instead it appears to inexplicably do absolutely nothing. There’s Facebook, Twitter, and MySpace all of which are “for LG,” presumably to accompany preinstalled widgets. Why you’d use these instead of the official versions of the respective apps is beyond me.

Down below “system applications” is a category simply named “Downloads.” It’s under this category and completely discrete section that things from the marketplace, internet, or otherwise installed will appear. Applications are ordered in here in the order they’re installed by default, which is puzzling. Notice that some applications also have a blue N above their logo—N ostensibly stands for New, as the bubble goes away after initial launch. You can force things to be organized alphabetically by performing a category reset in the menu. There’s also the option to view apps in a horizontal grid or list.

It’s a strange dichotomy that LG sets up with this launcher scheme that divides “downloaded” apps from “system applications.” one that’s made on no other Android device I’ve ever seen but the Optimus One. The end result is that most of the stuff I want (you know, since I just installed it) is at the very last page or very bottom of the list, resulting in guaranteed scrolling every single time. If you’re a power user, just replace the launcher with something else entirely.
The theme inside the messaging application is also honestly somewhat of an obvious attempt to emulate iOS. The dialog takes place in chat bubbles that are strikingly similar. What’s frustrating here is that the font is huge, and the shading and space between those bubbles is positively gigantic, which makes it hard to see the whole dialog.

The 2X ships with just LG’s own custom keyboard as an available option. The keyboard isn’t multitouch and offers only basic correction, but is visually more appealing than the stock Android keyboard. It works but it’d be nice to see Swype on here, but you can grab that yourself if you’re lucky enough to be in the beta program. I immediately went and grabbed Swype just out of force of habit, but found that occasionally input from the keyboard in the messaging application didn’t actually make it into the compose box. I could tell the keyboard was receiving input, but characters simply wouldn’t appear until I switched input methods back and forth at least once. Other small glitches included messages I sent sometimes not resulting in the dialog auto-scrolling to the latest. SMS is one of those things every phone has to really get right to be usable, and I feel like there’s a bit more refinement needed here.

The dialer also has some LG theme going on, but the differences between it and stock are purely aesthetic. White buttons, a link to messaging instead of voicemail, and moved backspace key are really it.
Back on the home screen, by default LG has a pretty decent preload of widgets and shortcuts. Their weather and clock widget is actually the best I've seen yet and has already been ripped from system dumps for use on other Android phones—after time with it, I can honestly understand why. Another extra is the ability to totally clear all the widgets and shortcuts from a particular homescreen from the longpress menu.

There’s also an FM radio application which works off the previously mentioned BCM4329 stack. Standard fare here with the requirement that you have some headsets plugged in to double as the antenna. The radio application has an auto-scan function for detecting local channels, and also pulls down RDS text.

Alongside the normal marketplace also is NVIDIA’s Tegra Zone marketplace which is home to apps and games optimized for Tegra 2 based smartphones. In general, they’re differentiated from normal applications by having more detailed geometry, textures, shaders, and content. Tegra Zone isn’t fully launched yet (it
goes live when the 2X starts shipping stateside, I’m not sure whether it’s live on the Korean version), so we couldn’t install applications from it directly, but could still explore the application and try sideloaded copies of some games from the market.

One of the interesting things done in the Tegra Zone marketplace is drawing a distinction between user contributed reviews and professional ones. As games get closer to their console and desktop counterparts (and probably more expensive), that could definitely be useful.

We were able to try Galaxy on Fire 2 and Dungeon Defenders, two of a number of titles that will be available through Tegra Zone. Galaxy on Fire is essentially space trader reimagined for 3D (I remember playing Space Trader an eternity ago on a Palm III) and uses OpenGL ES 2.0 shaders, specular and bump mapping, and 4x higher texture resolution and geometry over the normal version. It’s definitely visually compelling, especially for a mobile device. I’ve played Galaxy on Fire 2 on iOS a little bit and definitely can appreciate the difference in visuals between the two versions.

Dungeon Defenders is sort of a 3D tower defense game with online and RPG components. What makes the title interesting is that it’s Unreal 3 engine based and looks visually compelling.

Accelerometer Gestures
At CES I spoke to a number of people from Kionix, who make MEMS accelerometers for a wide variety of consumer electronics. What they were most excited about, however, were accelerometers with hardware support for detecting a variety of gestures. Things like detecting where touch events are on the screen purely from accelerometer data (even multiple accelerometers), and gestures such as waving or tapping on all sides of a phone. All that support is baked into the hardware—just watching a register is all that's required on the software side.

They told me to expect to see it in smartphones soon, but I didn’t think soon meant this early. When I saw the gestures tab in the 2X, I suspected Kionix. After a bit more digging, I found that the 2X definitely has a Kionix KXTF9 accelerometer inside, and the software support for a number of the gestures I saw demoed, specifically directional tap and double tap.

Under settings is a field appropriately labeled "gesture" where you can see all the available accelerometer gestures.

<table>
<thead>
<tr>
<th>Gesture settings</th>
<th>Gesture settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text input field</td>
<td>Snooze (turn off)</td>
</tr>
<tr>
<td>Home screen</td>
<td>Flip the phone over to snooze (turn off)</td>
</tr>
<tr>
<td>Rearrange item</td>
<td>the alarm. (Only if ringtone set to sound)</td>
</tr>
<tr>
<td>Incoming call</td>
<td>Previous/next Picture</td>
</tr>
<tr>
<td>Mute</td>
<td>Tap left/right side of the phone to view previous/next picture.</td>
</tr>
<tr>
<td>Alarm</td>
<td>Panning</td>
</tr>
<tr>
<td>Snooze (turn off)</td>
<td>Tilt the phone up/down/left/right to pan the screen to the nibed direction.</td>
</tr>
<tr>
<td>Gallery</td>
<td>Previous/next Music</td>
</tr>
<tr>
<td></td>
<td>Double-tap left/right side of the phone to play previous/next music.</td>
</tr>
<tr>
<td></td>
<td>Video player</td>
</tr>
<tr>
<td></td>
<td>Pause and mute</td>
</tr>
</tbody>
</table>

Back in the windows mobile days, I remember a number of homebrew applications enabling similar simple things, like putting the phone face down to silence a call or alarm, and rotating the phone clockwise in the air to lock, counterclockwise to unlock. This is sort of the extension of those, except with a few extras and much more polished and accurate detection.

Most of the elements are pretty self explanatory—you can tap on the sides of the phone to move the cursor, go to the next or previous photo, or double tap to change the track. It works surprisingly well. There's also those two gestures I mentioned earlier for silencing an alarm or pausing video and music playback. I show the tap left and right gestures in the overview video, which definitely works well. I think we're at the cusp of finally leveraging the accelerometer in some completely new and exciting ways.

Software Instability

One of the things I've been a huge proponent of is actually using every device in for review in place of my own. That was easy with the 2X since it worked with my AT&T SIM perfectly. The Optimus 2X is an excellent device save one huge problem—the software build on it as tested is extremely unstable. To be totally fair, the software LG has running on the 2X right now isn't what's shipping in Korea on their version, and isn't totally final. The first time the instability issues I'm going to mention cropped up, I reset completely to defaults and asked LG and NVIDIA about what was going on.

When I played with the 2X at CES, I had some instability issues, including two random reboots while doing relatively mundane things. The software on our sampled 2X hasn't randomly restarted, instead, applications crash in the background and go into a force close loop later. There are three main offenders I've seen do this on a regular basis, as frequent as once every three to four hours—the browser, messaging, and an LG background process named "omadmclient." My own speculation is that omadmclient is crashing almost on schedule because it's an OMA DM client checking for updates at a server that isn't live yet. That's forgivable but annoying.
The other two—browser and messaging—are not.

What happens is that periodically, when you try to launch either the browser or messaging, you'll get a force close loop that doesn't stop. Killing everything in the background with a task killer doesn't fix the problem or make the force close loop stop either. A reboot is the only way to get things back up and working. It's frustrating when that happens to the browser, but an even more serious problem when messaging does it.

When messaging decides to crash, you lose messages. The first time it happened, I assumed my friends were leaving me alone whilst I was working on a four hour long lab assignment. Instead, the entire messaging stack simply died in my pocket, silently. What's curious is that when this happens, messages are received from the network side but never make it into the messaging database. They completely disappear—the result is that when you pull it out of your pocket to check and experience the force closing, you've already permanently lost messages. Another instance that sticks out in my mind was when I was working on this actual article—I assumed friends knew I was busy writing and were consciously making an attempt not to distract. Instead, I ended up losing a couple hours of messages—thankfully most people already are used to me bouncing between devices and periodically discovering things like this.

The other much less urgent thing is a recurring audio "pop" which occurs even muted very now and then when using the phone. It happens at random but just frequently enough to frustrate.

Again, the build of the software on the 2X we tested isn't final and will definitely change before launch in the US. Nor is this the same software running on the 2X already for sale in South Korea. We pinged NVIDIA and LG about the aforementioned instability issues; NVIDIA confirmed that they saw the same stability issues, but LG told us they haven't seen or are aware of any crashing or instability. Getting these fixed before launch should be priority one—I'd be using the 2X right now were it not for the constant liability of losing another couple of hours of messages randomly.
One of Tegra 2’s most interesting features is support for multiple displays—HDMI 1.3 at 1080p mirroring is supported. The implementation on the 2X is how other Android phones with HDMI ports should have worked, you plug the HDMI cable in, and everything on the phone is instantly mirrored on the connected display. Android isn’t suddenly rendered at higher resolution, it’s just scaled up to whatever resolution of HDMI device you connect to, but that looks surprisingly good.

In portrait mode, there are black bars at the left and right, but rotate to landscape and the WVGA Android screen fills 1080P displays. WVGA (800x480) isn’t exactly 16:9, but it’s close, so there’s a little stretching in landscape but nothing noticeable.

The result is that you can use the 2X to play angry birds on a 55” TV without waiting for the console version, browse the web, give a PDF or PPT presentation, or do anything you’d do on the phone on a different screen. I put together a reasonably comprehensive video showing off HDMI mirroring.
There’s a tiny bit of input lag. In the video I shot showing off HDMI mirroring, it’s entirely possible some of that is just the result of my Onkyo TX-SR608 A/V receiver which seems to add a consistent 100ms of lag to almost everything, even in game mode. The supplied microHDMI cable is just long enough to stretch from the receiver to my couch, I could use a few more feet to be comfortable however.

You can also play videos over the HDMI connection, while doing so the 2X shows a “showing on second display” message:

HDMI mirroring works shockingly well, and sends all audio over HDMI. It’s a bit difficult to look at the TV and interact with the phone’s touchscreen, but not impossible. WebOS and others have drawn circles on the screen to show where fingers are. The tradeoff there is that it’s one more element to clutter display.

**Video Playback**

The big question is how well the X2 (or any Tegra 2 smartphone) could work as a mini-HTPC. NVIDIA advertises a big long list of codecs that Tegra 2 can decode:
LG's own spec list (what's below is actually for the Korean version, but the video codec support is the same) is much closer to the truth for the X2 because of Android's player framework and other limitations.

---

You can play back H.264 1080p30 content, but it has to be Baseline profile—no B frames, two reference frames. I used handbrake and messed around with a variety of other encode profiles and eventually settled on a bitrate of around 10 Mbps. That puts a 2 hour movie at around 8 GB total, which is too big to fit on a FAT32 microSD card. If you're going to fit 2 hours of video on that SD card and stay under 4 GB, bitrate should be around 4 Mbps. Tegra 2 can decode H.264 1080P baseline at a maximum of 20 Mbps.

Interestingly enough, I tried the iPhone 4 preset in handbrake which is H.264 960x400 High profile and noticed some stuttering and dropped frames. Media playback on Tegra 2 as it stands definitely works best with H.264 baseline, it's just a matter of having gobs of storage to park video on.

The 2X didn't do very well in our media streamer test suite. Some of that is because the software lacks the ability to open mkvs and a huge number of our files. The two that did open and playback successfully were test 3, an 8 Mbps 1080p WMV9 video with 5.1 WMA audio, and file 19, a simple m4v container test. Unfortunately we're still not at the point where you can dump just about anything you'd stick on an HTPC on your mobile device without a transcode in-between, it's no pirate phone.

---

Battery Life

There's been a lot of speculation about whether dual-core phones would be battery hogs or not. Turns out that voltage scaling does win, and \( P=V^2/R \) does indeed apply here. The 2X delivers middle of the road 3G and WiFi web browsing battery life numbers, and above average 3G talk time numbers.
We’ve also got another new test. Gaming battery life under constant load is a use scenario we haven’t really been able to measure in the past, but are now able to. Our BaseMark GUI benchmark includes a battery test which runs the composition feature test endlessly, simultaneously taxing the CPU and GPU. It’s an aggressive test that nearly emulates constant 3D gaming. For this test we leave WiFi and cellular interfaces enabled, bluetooth off, and display brightness at 50%.
I'm a bit disappointed we don't have more numbers to compare to, but the 2X does come out on top in this category. Anand and I both tested the Galaxy S devices we have kicking around (an Epic 4G and Fascinate), but both continually locked up, crashed, or displayed graphical corruption while running the test. Our constant 3D gaming test looks like a winner for sifting out platform instability.

**Conclusion**

The 2X is somewhat of a dichotomy. On one side, you've got moderately aesthetically pleasing hardware, class-leading performance from Tegra 2 that doesn't sacrifice battery life at the stake, and a bunch of notable and useful extras like HDMI mirroring. On the other, you've got some serious experience-killing instability issues (which need to be fixed by launch), a relatively mundane baseband launching at a time when we're on the cusp of 4G, and perhaps most notably a host of even better-specced Tegra 2 based smartphones with more RAM, better screens, and 4G slated to arrive very soon.

It's really frustrating for me to have to make all those qualifications before talking about how much I like the 2X, because the 2X is without a doubt the best Android phone I've used to date. Android is finally fast enough that for a lot of the tasks I care about (especially web browsing) it's appreciably faster than the iPhone 4. At the same time, battery doesn't take a gigantic hit, and the IPS display is awesome. The software instability issues (which are admittedly pre-launch bugs) are the only thing holding me back from using it 24/7. How the 2X fares when Gingerbread gets ported to it will also make a huge difference, one we're going to cover when that time comes.

The other part of the story is Tegra 2.

Google clearly chose NVIDIA's Tegra 2 as the Honeycomb platform of choice for a reason. It is a well executed piece of hardware that beat both Qualcomm and TI's dual-core solutions to market. The original Tegra was severely underpowered in the CPU department, which NVIDIA promptly fixed with Tegra 2. The pair of Cortex A9s in the AP20H make it the fastest general purpose SoC in an Android phone today.

NVIDIA's GeForce ULV performance also looks pretty good. In GLBenchmark 2.0 NVIDIA manages to hold a 20% performance advantage over the PowerVR SGX 540, our previous king of the hill.

Power efficiency also appears to be competitive both in our GPU and general use battery life tests. Our initial concern about Tegra 2 battery life was unnecessary.

It's the rest of the Tegra 2 SoC that we're not completely sure about. Video encode quality on the LG Optimus 2X isn't very good, and despite NVIDIA's beefy ISP we're only able to capture stills at 6 fps with the camera is set to a 2MP resolution. NVIDIA tells us that the Tegra 2 SoC is fully capable of a faster capture rate for stills and that LG simply chose 2MP as its burst mode resolution. For comparison, other phones with burst modes capture at either 1 MP or VGA. That said, unfortunately for NVIDIA, a significant technological advantage is almost meaningless if no one takes advantage of it. It'll be interesting to see if the other Tegra 2 phones coming will enable full resolution burst capture.

Then there's the forthcoming competition. TI's OMAP 4 will add the missing MPE to the Cortex A9s and feed them via a wider memory bus. Qualcomm's QSD8660 will retain its NEON performance advantages and perhaps make up for its architectural deficits with a higher clock speed, at least initially. Let's not forget that the QSD8660 will bring a new GPU core to the table as well (Adreno 220).

Tegra 2 is a great first step for NVIDIA, however the competition is not only experienced but also well equipped. It will be months before we can truly crown an overall winner, and then another year before we get to do this all over again with Qualcomm's MSM8960 and TI's OMAP 5. How well NVIDIA executes Tegra 3 and 4 will determine how strong of a competitor it will be in the SoC space. Between the performance we're seeing and the design wins (both announced and rumored) NVIDIA is off to a great start. I will say that I'm pleasantly surprised.