NEW SILICON

Siliconarts' RayCore ray-tracing processor

Disruptive technology from a little startup?

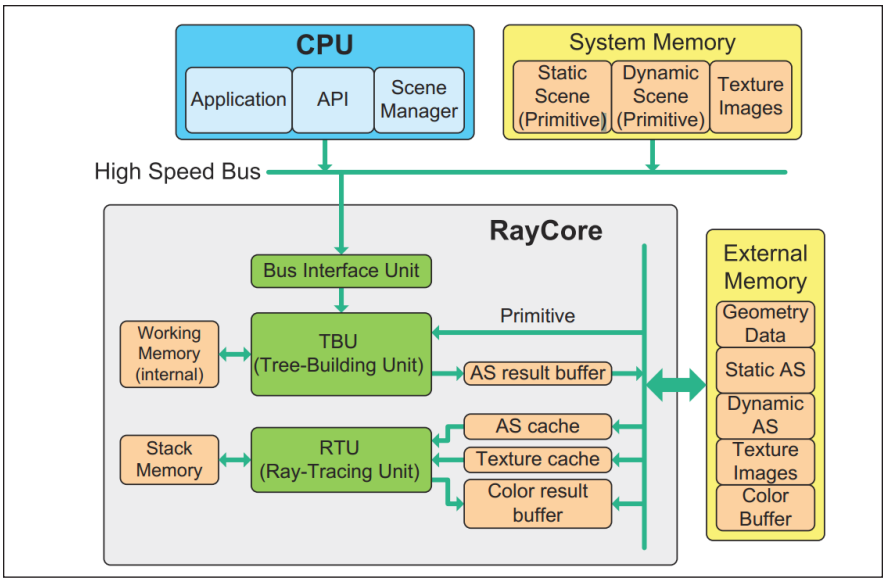
By Jon Peddie

Seemingly out of nowhere—well, South Korea, actually—a four-year-old startup founded by Dr. Hyung Min Yoon, formerly at Samsung, Hee Jin Shin from LG, Byoung Ok Lee from MtekVision, and Woo Chan Park from Sejong University burst on the scene at Hot Chips and simultaneously at Siggraph this year to show off a ray-tracing (RT) chip, their RayCore.

It is not, however, as the company claims, the world's first ray-tracing chip, but they qualify their claim by calling it the world's first ray-tracing GPU IP. The first actual RT chip was built by Advanced Rendering Technology (ART) in Cambridge, U.K., 1995, and the company still operates but gave up its silicon solution about six years ago. Caustic Graphics (now owned by Imagination Technologies) was started in 2006 and built prototype hardware RT accelerators using FPGAs.

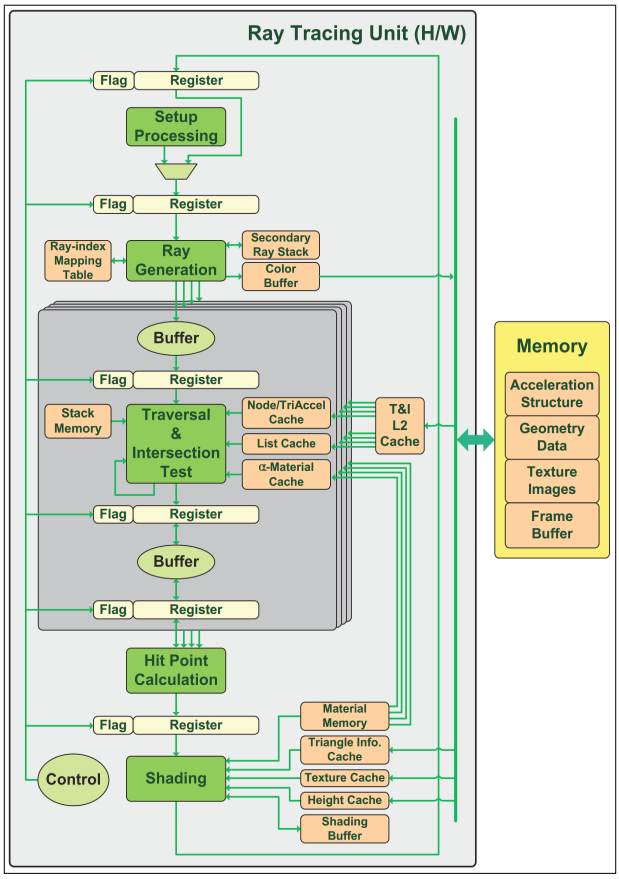
Siliconarts is offering IP, not a chip. What they demonstrated at Siggraph and Hot Chips was a proof of concept, not a product. (Although they do claim to have an RT chip—the RayCore built in 55-nm.) No matter how it's delivered, they have an interesting and impressive piece of work. In fact, the company has already been licensing out its ray-tracing IP to its OEM partners since 2011, and is currently working with a multinational mobile AP manufacturer to develop a next-generation mobile application processor.

RayCore consists of two major components, the ray-tracing units (RTUs) based on a unified traversal and intersection pipeline, and a tree-building unit (TBU) for dynamic scenes. Unlike SIMD-based GPUs, the The RayCore architecture is based on a multiple instruction, multiple data (MIMD) architecture, and has been developed as RT IP. The RTU's unified traversal and intersection (T&I) pipelines MIMD execution model was chosen to meet power efficiency and Si area goals targeting mobile devices. According to Siliconarts, the ASIC evaluation, with six RTUs, can achieve up to 239 Mrays/second using an area of 18 mm^2 and 1 watt power consumption using a 28-nm process. The company says the RTU is designed to maintain ray-tracing performance regardless of ray coherence or scene characteristics. The TBU uses a K-D tree construction to test models with up to 64K triangles, and the company says it can run such a test in 20 ms.



Source: Siliconarts

BASIC ARCHITECTURE of Siliconarts RayCore processor.



Source: Siliconarts

THE OVERALL architecture of the ray-tracing unit.

To reduce performance degradation due to off-chip memory accesses, Silconarts uses a novel latency-hiding technique they call "looping for the next chance" on the T&I pipelines combined with efficient memory systems of T&I units and the TBU and texture mip-mapping, to minimize off-chip memory accesses. As a result, says Siliconarts, real-time Whitted ray tracing with six RTUs and k-D-tree construction with one TBU requires less than 1.1 GB/second of memory bandwidth (in their benchmarks), which is much less than the bandwidth of mobile LPDDR3 memory (12.8 GB/second).

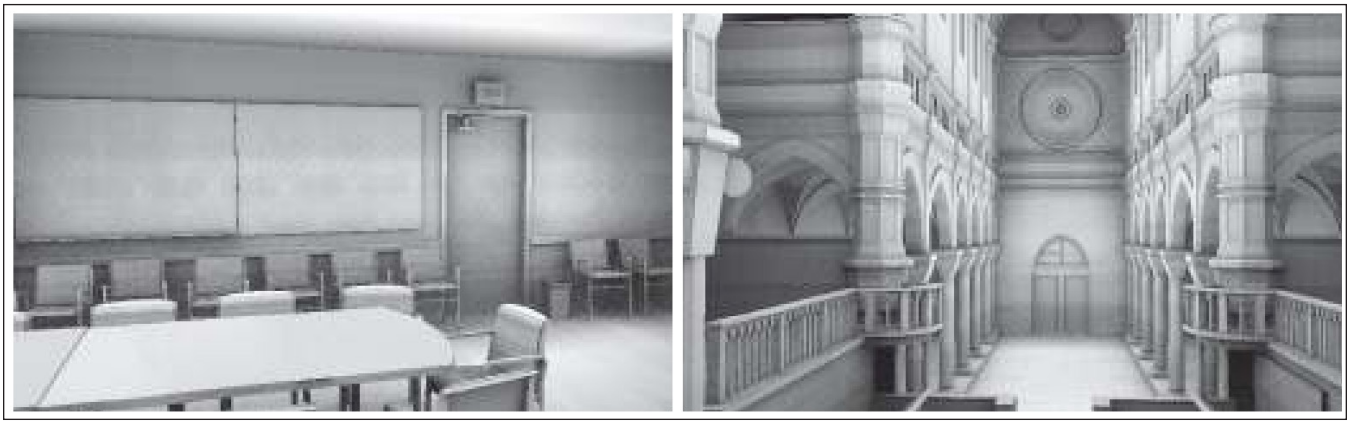
The ray-tracing engine is depicted in the diagram to the left.

Siliconarts says they provide OpenGL ES 1.1-like API extensions to separate static and dynamic objects. Static objects are retained for subsequent frames, and dynamic objects are transferred to the tree builder via vertex arrays to reconstruct dynamic sub-trees during each frame. With regard to S/W programming, the OpenGL ES API can be the medium that con-

NEW SILICON

Source: Siliconarts

THREE STATIC test scenes for Whitted ray tracing: Kitchen, Room with moving light, and Living room rendered at interactive frame rates on FPGA prototype.



SAMPLE IMAGES from two static test scenes: Conference (courtesy of Anat Grynberg and Greg Ward) and Sibenik (courtesy of Marko Dabrovic). These images were rendered with ambient occlusion.

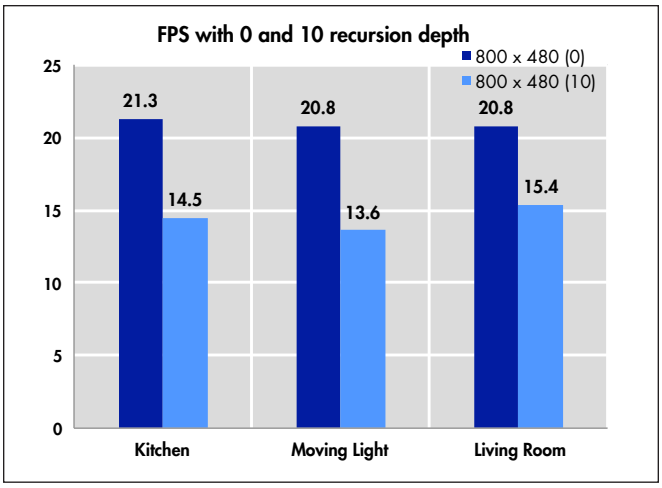
nects RayCore and the mobile GPU cores in the same AP.

This research was supported by Siliconarts and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education.

Results

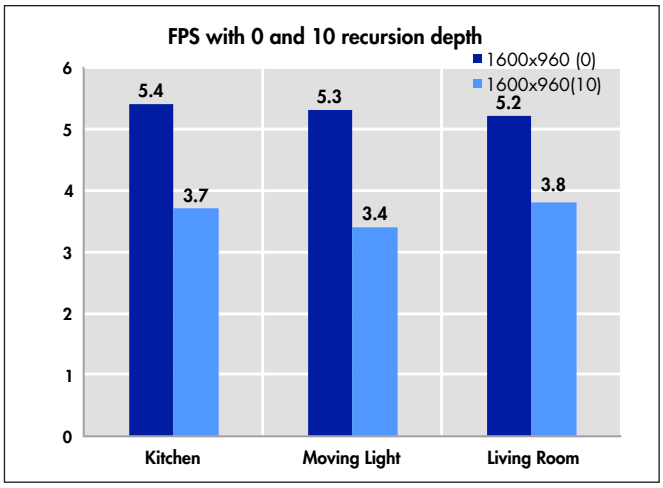
Siliconarts used two different scene setups for benchmark testing. They designed three scenes, as shown above: Kitchen (296 K triangles), Room with moving light (240 K triangles), and Living room (360 K triangles).

The Kitchen and Living room scenes each have one moving camera and two static light sources. In contrast, the Room with moving light scene has one static camera, one static light source, and one dynamic light source. To measure ray coherence, they set the ray recursion depths to 0 and 10. In the lat-



Source: Siliconarts

FPGA PERFORMANCE results for Whitted ray tracing at 800 x 480.



Source: Siliconarts

FPGA PERFORMANCE results for Whitted ray tracing at 1600 x 960.

ter case, coherence between the rays is quite low; rays of different types (primary, shadow, reflection, and refraction rays) and different depths are processed simultaneously in a T&I pipeline. The company notes that reflection or refraction rays were spawned only if the material on the hit point was reflective or refractive. The screen resolutions on this benchmark are 800 × 480 and 1600 × 960.

The second benchmark was structured in order to analyze the performance of different ray types. In this benchmark, two scenes were selected conference (282 K triangles) and Sibenik (M) K triangles).

In this benchmark, they set the ray types to a primary ray (the most coherent type), an AO ray, and a diffuse inter-reflection ray (the least coherent type). One light source used for all. There were 32 samples per pixel for AO and diffuse inter-reflection rays. AO rays were terminated by the cut-off value of 5.0 for the maximum distance.

The results were impressive.

Remember, these are test results using a FPGA. A tightly coupled IP block in a 22-nm or smaller SoC would be significantly faster.

The business idea is similar to Caustic's/IMG to offer an IP block that can be part of an SoC. Caustic is offering a fully functional AIB as an RT accelerator, while IMG is offering the IP as part of their library. WIth the firm located in South Korea, you can guess who Siliconarts' target customers are.

Siliconarts, also like Caustic, offers a blended RT capability using RT for only certain parts of the scene or image. That saves enormous amounts of time and gives beautiful (and hopefully) physically accurate results.

Speed

Ray-tracing is one of the holy grails of CG, and has always been difficult to realize quickly because of the enormous computational load. GPUs are often used, but because of the organization of GPUs, they are not the ultimate solution, even with their thousands of processors. However, at Siggraph AMD had a technology demo using four high-end Radeon AIBs running RT. An RT image is said to "resolve" or converge, meaning it processes (and processes and processes) until the viewer is satisfied (or accepting of the image). You've seen the effect; it is the scintillation of the image as it builds up all the ray paths. With four AIBs, the AMD demo could resolve an acceptable image in about 2 to 3 seconds, which is pretty damn fast, but not as fast as 33 ms, which is what is required for a 30 frames per second real-time display such as we get in FPS HD games today—with 16 ms being the desired goal.

To get the frame rate up, several tricks are employed. One is blending, as mentioned. Other methods are limiting the light sources or limiting the secondary bounces of the rays; another is limiting the number of ops, or converge steps. One measurement is an OpenCL benchmark called RAT. Look at the convergence times—many, many seconds.

Siliconarts looks damn interesting. They claim to be able to do 239 m rays/second. If you use an HD screen as the filter (assuming their SW is clever enough to not launch rays that can't be seen), then the RayCore 2000 could generate 300 rays/second in an HD screen, which would seem to be fast enough. However, a ray has to bounce off of several surfaces to create a realistic image, and Siliconarts says they will allow up to 15 bounces (which is a lot), so even at that max, you could theoretically be realize 30 frames a second, which would be astounding and earth-shaking.

What do we think?

If this is as good as it looks right now, it is indeed going to be disruptive; however, there are some obstacles.

It is indeed innovative technology; however, it is based on OpenGL ES 1.1, which is a fixed pipeline architecture. As everybody knows, now is the age of OpenGL ES 2.0, which does not have a fixed pipeline architecture, but is programmable shader–based. Siliconarts technology is somewhat mismatched with OpenGL ES 2.0 and, of course, OpenGL ES 3.0. They will have to overcome these obstacles to get into new SoCs. However, it may come from their basic design philosophy, and therefore may be a serious obstacle. Based on an older philosophy (fixed pipeline), the company may encounter obstacles for standardization, too. Although Siliconarts wants to, it will be hard to add their APIs as an extension of OpenGL ES 1.1 because nobody in Khronos will want to discuss it—they're planning the next generation, which may merge standard OpenGL with OpenGL ES.

Just getting started

However, this is Siliconarts' first implementation, developed with limited funds from Siliconarts and a grant. It is possible, and we're trying to find this out from the company, that the next generation will embrace a programmable pipeline and be more compatible with the newer and future versions of OpenGL ES. It's always a tricky process trying to get the timing right on standards.

Another consideration is the whole premise of mobile. Accelerated RT on its own is not enough to turn the tide on mobile, at least in the near term. Look how good UE4 looks using DX/GL rasterization on a single GPU when done right: https://www.youtube.com/watch?v=UwEuSxAEXPA.

Next steps

To get to the next step they will need true path tracing, which will likely always be cheaper and simpler to stream from the cloud or bake into a lightfield environment like what Otoy showed at Siggraph. All of the work Otoy is doing with path tracing, cloud rendering, and light fields is being added in the next few months to the UE4 engine to give developers the tools to leverage this tech on the latest mobile GPU platforms.

So Siliconarts has its work cut out for them, but they have a great start and opportunity.

As for the future, in speaking with Silionarts they told us they plan to develop a combined GPU that supports both ray tracing and OpenGL ES 2.X/3.X. The Siliconarts folks tell us because their ray-tracing API looks similar to that of OpenGL ES 1.1, it is easier for them to design an OpenGL ES vendor-specific extension. This combined GPU supports full-screen ray tracing as well as hybrid ray tracing (full screen of OpenGL ES + partial screen of ray tracing).

With regard to the UE4 demo that shows the light effects, the Siliconarts folks think it is a great example of what they should do next—real-time global illumination. In the company's roadmap, they have laid out a concrete plan to develop real-time path tracing. This will make a significant impact in mobile and embedded space. To overcome the limitations of noise in path tracing, they plan to apply several noise filter techniques to make sure it provides life-like graphics in gaming, UI, AR, VR, and many other real-time applications.

We think these guys are really going to make a difference. I suspect they will be acquired in less than five years.