

# SGRT: A Scalable Mobile GPU Architecture based on Ray Tracing

Won-Jong Lee\* Shi-Hwa Lee\* Jae-Ho Nah† Jin-Woo Kim† Youngsam Shin\* Jaedon Lee\* Seok-Yoon Jung\*  
 \*SAIT, Samsung Electronics, Korea †Yonsei University, Korea

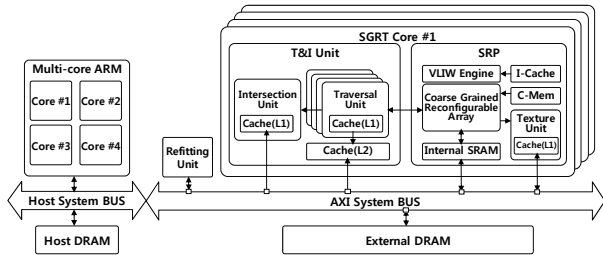


Figure 1: (a) Our system architecture including the SGRT cores and host processor. (b) Rendered images by the SGRT simulator: Ferrari (left, 210K triangles, 1 light source) and Fairy (right, 170K triangles, 2 light sources). The SGRT (4-core) is predicted to render at 67.83fps (Ferrari) and at 87.82fps (Fairy).

## 1. Introduction

Recently, with the increasing demand for photorealistic graphics and the rapid advances in desktop CPUs/GPUs, real-time raytracing has attracted considerable attention. Unfortunately, raytracing in the current mobile environment is difficult because of inadequate computing power, memory bandwidth, and flexibility in mobile GPUs. In this work, we present a novel mobile GPU architecture called the SGRT (Samsung reconfigurable GPU based on RayTracing) by enhancing our previous works with the following features: 1) a fast compact hardware engine that accelerates a traversal and intersection operation, 2) a flexible reconfigurable processor that supports software ray generation and shading, and 3) a parallelization framework that achieves scalable performance. Unlike our previous work, the current architecture is designed for both static and dynamic scenes with a smaller area. Experimental results show that the SGRT can be a versatile graphics solution, as it supports compatible performance compared to desktop GPU raytracers. To the best of our knowledge, the SGRT is the first mobile GPU based on full Whitted raytracing.

## 2. SGRT Core Architecture

**Dedicated Hardware for Traversal and Intersection:** The lack of computational power (<68GFLOPS) and memory bandwidth (<6.4GBPS) of current mobile GPUs motivated us to design a dedicated hardware for traversal and intersection, which are computation-intensive operations in raytracing. Our hardware, called the T&I engine, is based on our previous work [Nah et al. 2011]. However, unlike our previous work, the new T&I engine is designed for handling dynamic scenes with a bounding volume hierarchy (BVH). Moreover, the T&I engine has a smaller area (3.89 mm<sup>2</sup> per core, 65nm), because BVH is an object hierarchy, which negates the need for LIST units to manage primitives. High performance features like the MIMD architecture for incoherent rays and a ray accumulation unit for latency hiding are directly reused. We can selectively utilize a specific BVH between the variants (e.g. Full SAH, Binned, SBVH, and LBVH) that are supported by the T&I engine. The T&I engine also has other outstanding features such as ray-AABB intersection units and a compact node layout. A full paper version will be announced in the near future.

**Reconfigurable Processor for Shading:** We utilize a proprietary low-power DSP core developed in our previous work [Lee et al. 2011]; it is called the SRP (Samsung Reconfigurable Processor). The SRP is very flexible for supporting full programmability (standard “C” language); thus, various shaders (e.g. material and illumination) can be easily implemented. Unlike the conventional mobile GPU, the VLIW engine of the SRP can fully support control-flow such as branch, which make recursive raytracing possible. In addition, the SRP is capable of highly parallel data pro-

cessing. The coarse-grained reconfigurable array (CGRA) of the SRP makes full use of the software pipeline technique to allow loop acceleration. Therefore, the ray packet stream processing can be done in ray generation and shading kernels, which maximizes the utilization of the functional units. Furthermore, the use of the SRP’s reconfigurable feature might enable hybrid rendering that combines the OpenGL|ES rasterizer and raytracing.

**Parallelization Framework:** For scalable performance, we built a parallelization framework based on the Samsung Micro Kernel (SMK), a real-time operating system for embedded system. The SMK supports multi-tasking by systematic scheduling in the task queues, and it allows developers to create and use tasks easily. We define an individual task for each SGRT core that is responsible for different pixels (or pixel tiles), then the scheduler can distribute the next tasks to the idle SGRT core first, which results in dynamic load balancing. According to preliminary experiments, we could determine the performance scalability: 3.8x speedup on 4 SGRT cores compared to a single core.

## 3. Results

Figure 1(a) shows the overall system architecture including the SGRT cores and host CPUs. Our architecture is based on an asynchronous BVH that is a combination of the rebuild (CPU), the refit (H/W) and the rendering (SGRT). The validity of the SGRT is verified and its performance is evaluated during cycle accurate simulation. The Ferrari and Fairy scene has been thoroughly tested (Figure 1(b)). Table 1 lists the performance results of raytracing performed by the SGRT (4 cores), including shadow, reflection and refraction with WVGA (800x480) resolution at 1GHz clock speed. We achieve around 170M RPS (T&I engine), 255M RPS (SRP) and 87.82 fps (Fairy), which may be equivalent to the performance of recent desktop GPU ray tracers (200-300M RPS). We are now implementing the T&I engine at the RTL level, and we will release the complete product supporting fully dynamic scenes in the future.

Table 1. Performance results of the SGRT architecture

Scene	# of tri.	# of ray	T&I Engine (usage & cache hit ratio)			SRP		FPS
			Pipe	TRV %	IST %	MRPS	MRPS	
Fairy	170K	1.7M	87.27	93.83	96.53	171.32	255.72	87.82
Ferrari	210K	1.5M	79.75	92.56	92.92	122.48	319.56	67.83

## References

- NAH, J.-H., ET AL. 2011. T&I Engine: Traversal and Intersection Engine for Hardware Accelerated Ray Tracing. In *ACM Transaction on Graphics (Proceedings of SIGGRAPH ASIA 2011)*, 30, 6, 160:1-10.
- LEE, W.-J., ET AL. 2011. A Scalable GPU Architecture based on Dynamically Embedded Reconfigurable Processor. In *High Performance Graphics 2011, Posters*.

\*e-mail: joe.w.lee@samsung.com