

# SPIR-V to ISPC: GPU Compute on the CPU

There is a growing trend within the games industry to move compute work to the graphics processing unit (GPU) resulting in engines and/or studios developing large portfolios of GPU compute shaders for many different compute tasks. However, there are times when it would be convenient to run those compute shaders on the CPU without having to re-invest in developing C/C++ variants of them. There are many reasons for doing this, including simple experimentation and debug, utilizing spare CPU cycles and encouraging CPU-based content scaling, CPU-based interaction with other CPU side game assets, for deterministic consistency of results, and so on.

To help address this opportunity while also utilizing the single instruction, multiple data (SIMD) vector units built into modern CPU cores, we have started developing a prototype translator, based on the open source Khronos [SPIRV-Cross](#) project, that will take Standard Portable Intermediate Representation ([SPIR-V<sup>1</sup>](#)) as input and produce [Intel® SPMD Program Compiler \(a.k.a. ISPC\)](#) kernels as output. ISPC takes C-style kernels and generates highly vectorized CPU object files targeting multiple ISAs such as Intel® Streaming SIMD Essentials (Intel® SSE), Intel® Advanced Vector Extensions 2 (Intel® AVX2), and Intel® Advanced Vector Extensions 512 (Intel® AVX-512).

The project should be considered as a starting point for conversion to ISPC rather than a fully featured and performant solution. The project currently supports a subset of the standard SPIR-V intrinsics, built-ins and types, but it was designed to utilize a core performance feature of ISPC, which is the notion of uniform (scalar) or varying (vector) variables. This allows optimizations such as avoiding expensive and divergent vector branches if the test condition is scalar.

The code has been tested on a handful of shaders, such as the compute examples in [Sascha Willems' Vulkan\\*](#) repo and the particle system compute shaders in [Microsoft DirectX\\*12 MiniEngine](#) sample.

The code can be downloaded from our [GitHub\\* repo](#) and has currently only been tested on Windows\* systems. The GitHub Readme contains more detailed documentation about the implementation and supported features.

## Usage:

```
glslangValidator.exe -H -V -o test.spv test.comp
spirv-cross.exe --ispc --output test.ispc test.spv
ispc.exe -O2 test.ispc -o test.ispc.obj -h test.ispc.h --target=avx2 --opt=fast-math
```

## Example API Usage:

```
ispc::raytracing_get_workgroup_size(workgroupSize[0], workgroupSize[1],
workgroupSize[2]);

int32_t dispatch[3] = { textureComputeTarget.width / workgroupSize[0],
textureComputeTarget.height / workgroupSize[1], 1 };
int32_t dispatch_count = dispatch[0] * dispatch[1];

concurrency::parallel_for<uint32_t>(0, dispatch_count, [&](uint32_t dispatchID)
{
    int32_t workgroupID[3] = { dispatchID % dispatch[0], dispatchID / dispatch[1], 0 };
    ispc::raytracing_dispatch_single(workgroupID, dispatch, planeCount, *pPlanes,
sphereCount, *pSpheres, *pUBO, resultImage);
});
```

This project is open sourced under the original SPIRV-Cross [Apache 2.0 license](#) and we welcome any comments and contributions.

Further information on using ISPC in games can be found in the article, [Use the Intel® SPMD Program Compiler for CPU Vectorization in Games.](#)

## Notes:

<sup>1</sup> SPIR-V is the default shader language for Vulkan and can be generated from OpenGL\* Shading Language (GLSL) and High-Level Shading Language (HLSL) shaders by tools such as the [glslangValidator](#) and [shaderc](#) compilers.

## Notices

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting [www.intel.com/design/literature.htm](http://www.intel.com/design/literature.htm).

This sample source code is released under the [Intel Sample Source Code License Agreement](#).

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

\*Other names and brands may be claimed as the property of others.

© 2018 Intel Corporation