Vitis™ HLS Webinar Series Q&A

Webinar FAQ's

Is Vivado™ HLS different from Vitis™ HLS?

Vivado™ HLS was the older generation of HLS and was retired in 2020. The next-generation Vitis HLS has several improvements, including a new compiler that uses an updated version of LLVM and supports C/C++ 11/14 for compilation and simulation. Please refer our user guide to learn more on migrating to Vitis HLS

Is Vitis HLS different from Vitis?

Vitis HLS is different from Vitis. Vitis HLS is used to develop acceleration kernels, while Vitis is the system level design tool, where kernels are integrated with other components of the design.

What AMD devices are supported by Vitis HLS?

Vitis HLS supports all devices 7-series and newer, including FPGAs and adaptive SoCs such as Versal™.

Is HLS task-level parallelism available for pure C programs?

hls::task are objects in C++, so they do require usage of the C++ compiler. That being said, C code can be compiled by the C++ compiler if you want other portions of your design to use C code.

Can Vitis HLS help me insert pragmas?

In the older Vitis HLS tool versions, the tool required to add pragmas inserted by the developer. Long time users of Vitis HLS may have noticed that the tool automatically inserts pragmas in more cases, when the tool can determine that the pragma is beneficial. In addition, the Performance pragma can be used to set a high-level performance constraint and let the tool infer even more pragmas.

What attributes of a task might disqualify a task from being data driven?

The only requirements for a data driven task are that all the data is passed to that task using hls::stream or hls::stream_of_blocks. One ramification of this is that memory mapped (e.g., AXI4) transactions cannot occur in a data driven task.

Can I combine Control-Driven and Data-driven?

Yes, you can design some portions of the design with Control-driven and some portions with Data-driven. In fact, there are advantages to combining both. If we continue from the previous question, if your design required a memory mapped interface and the engineer wanted to use hls::task, a control driven region could be created that encompasses a data driven region to utilize the benefits of both. You can refer to this example on how to combine them

What is a C-testbench?

The C test bench is used for verifying the design functionality. In terms of programming, this is as simple as defining the main () function in which you will instantiate the function that you want to test, feed it some inputs and verify the outputs. The C test bench will be used to create an RTL test bench. After the C and RTL simulations runs the results can be compared.

How does the intra-task handoff/signal occur?

It depends on what kind of Task Level Parallelism you are using. With control driven, there are block level control signals that connect tasks, which allow them to coordinate execution. With data driven, there is no control signal except for the presence or absence of data on a stream.

Are there any examples demonstrating the usage of channels?

You can refer to Introductory Examples on Channels here