Accelerating FPGA Development with C++

Intel® HLS Compiler is a high-level synthesis (HLS) tool that takes in untimed C++ as input and generates production-quality register transfer level (RTL) that is optimized for Intel FPGAs.

Figure 1 shows the Intel HLS Compiler tool flow, which enables an accelerated time for development that rivals hand-coded RTL. Creating FPGA accelerators can be cumbersome if customers wish to stick to traditional RTL flows. Therefore, we developed the Intel HLS Compiler tool. It is great for those who have already mastered the back-end flow, from high-level design to bit stream to run on the FPGA. Our high level synthesis compiler allows customers to generate RTL code loadable onto Platform Designer (formerly Qsys) using C++.

This tool accelerates verification time over RTL by raising the abstraction level for FPGA hardware design. Models developed in C++ are typically verified orders of magnitude faster than RTL and require 80% fewer lines of code†. The Intel HLS Compiler generates reusable, high-quality code that meets performance and is within 10%-15% of the area of hand-coded RTL.†

Faster Verification Time

The functional debug and verification iteration cycle for RTL simulation is long and limits iterations per day, resulting in longer development times for FPGAs. Due to this part of the development cycle, engineers using RTL alone struggle with meeting deadlines for frequent design changes, such as bigger devices, new application logic, and core frequency requirement changes.

Fortunately, verifying the design source in C++ is much faster than simulating in RTL. Table 1 shows orders of magnitude reduction in simulation time between C++ and RTL. For algorithms that require large test vector sets this can be the difference between 1 or 2 iterations per day to up to 50 to 100 iterations per day.

The Intel HLS Compiler accelerates the functional verification and debug cycle, allowing designers to find the errors and fix them in a fraction of the time.

---

Table 1. RTL vs. Untimes C++ Functional Verification Times†

<table>
<thead>
<tr>
<th>DESIGN</th>
<th>RTL SIMULATION TIME</th>
<th>C++ RUNTIME</th>
<th>ACCELERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Encryption System (AES)</td>
<td>22 minutes</td>
<td>46 ms</td>
<td>29,000x</td>
</tr>
<tr>
<td>Huffman Encoding</td>
<td>13 minutes</td>
<td>52 ms</td>
<td>15,000x</td>
</tr>
<tr>
<td>Complex FIR Filter</td>
<td>4.5 minutes</td>
<td>63 ms</td>
<td>4,200x</td>
</tr>
</tbody>
</table>

†

Figure 1. Development Flow
Intel HLS Compiler Key Features

The main features of the Intel HLS Compiler that make designing, implementing, and testing a new FPGA project easier are:

- **Supports C++ input** - The Intel HLS Compiler supports C++ for algorithmic development, which allows you to transfer existing programs that use the gcc compiler to the Intel HLS Compiler for FPGA development. This allows seamless verification of a user’s algorithm in the software world while using a software testbench.

- **Automatic RTL verification to C++** - Software testbench verification against the compiler generated hardware model (RTL) is automatically supported.

- **Interactive analysis reports** – The Intel HLS Compiler generates interactive analysis reports, with cross-probing of the source code allows for easy micro-architecture optimizations, such as loop-unrolling and variable dependency fixing.

- **Floating-point support** – The Intel HLS Compiler has native support for floating-point and fixed-point variables and operators. The floating-point support is especially exciting given the newly integrated hardened floating-point multiply-add units in Intel FPGAs. The fixed-point support is also useful for its ability to both cover legacy designs and enable the fastest, smallest design.

1Benchmark is performed using the following hardware and software—Intel HLS Compiler v0.9, ModelSim*, Intel FPGA SE-64 10.4d software, 2x8-core Intel Xeon® processor ES-2680 at 2.7 GHz, 256 GB RAM.

† Tests measure performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

© Intel Corporation. Intel, the Intel logo, the Intel Inside mark and logo, the Intel. Experience What’s Inside mark and logo, Altera, Arria, Cyclone, Enpirion, Intel Atom, Intel Core, Intel Xeon, MAX, Nios, Quartus and Stratix are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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