

Benefits of Smart High-Level Synthesis for FPGA Design

Introduction

As FPGA designs continue to get larger and more complex, engineers need to improve their productivity to meet tight design schedules. This white paper explains how engineers can speed up their FPGA development by using the SmartHLS tool to generate their hardware blocks from the C++ software.

Smart High-Level Synthesis (SmartHLS[™]) provides an integrated development environment tool that enables engineers to compile the C/C++ software into Verilog targeting a Microchip[®] FPGA device, improving productivity, and time-to-market. For more information, visit the SmartHLS webpage.

High-Level Synthesis Design Flow for FPGA

- 1. In the high-level synthesis design flow, the engineer implements the design in C++ software and verifies the functionality with software tests.
- Next, they specify a top-level C++ function, which SmartHLS will compile into an equivalent Verilog hardware
 module. SmartHLS can run co-simulation to verify whether the hardware module behaviour matches the
 software. SmartHLS uses Libero SoC to generate the post-layout timing and resource reports for the Verilog
 module.
- Finally, SmartHLS generates a SmartDesign IP component that the engineer can instantiate into their SmartDesign system in Libero SoC.

The following figure shows the high-level synthesis FPGA design flow for targeting a Microchip PolarFire® FPGA.

Figure 1. Smart High-Level Synthesis FPGA Design Flow Targeting a PolarFire FPGA

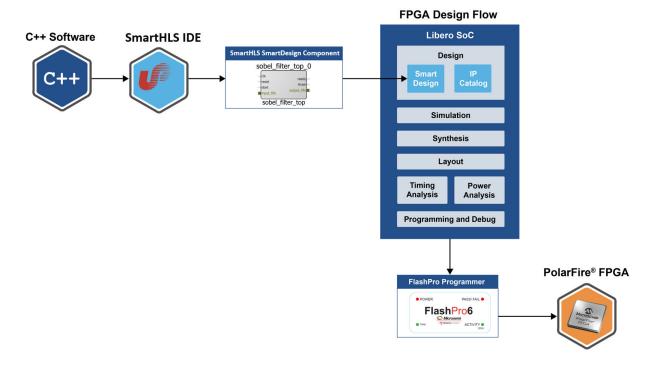


Table of Contents

Intr	oducti	on	1				
1.	FPGA Design: RTL vs. HLS						
2.	Benefits of High-Level Synthesis						
	2.1.	HLS Productivity Gain	4				
	2.2.	Port Existing Software to FPGA	4				
	2.3.	Faster Verification and Less Bugs	5				
	2.4.	Design Space Exploration	5				
	2.5.	FPGA Device Portability	5				
3.	Conc	lusion	6				
	3.1.	Key Takeaways	6				
The	e Micro	chip Website	7				
Pro	oduct C	hange Notification Service	7				
Cu	stomer	Support	7				
Mic	crochip	Devices Code Protection Feature	7				
Leç	gal Not	ice	8				
Tra	ıdemar	ks	8				
Qu	ality M	anagement System	9				
۱۸۱۰	Vorldwide Sales and Service						

1. FPGA Design: RTL vs. HLS

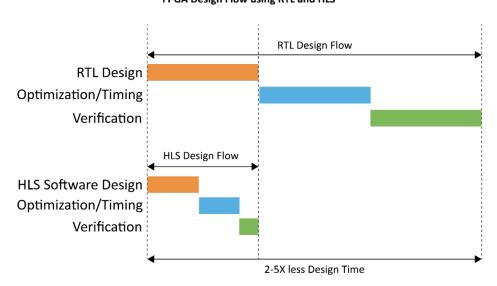
For readers unfamiliar to high-level synthesis, it compares the traditional FPGA design flow using a register transfer level (RTL) language like Verilog/VHDL to the newer high-level synthesis design approach using the C++ software.

Traditional RTL Design Flow for FPGAs:

- RTL design: Hardware engineer decides the hardware microarchitecture and manually writes the hardware block in RTL language like Verilog/VHDL.
- Optimization & Timing closure: Hardware engineer iterates on design to meet the clock period and area constraints by adding pipeline registers and restructuring the RTL.
- 3. Verification: Hardware engineer writes a test bench to verify the RTL block in simulation. Acceptance testing compares functionality to the original software model.

The following figure displays using high-level synthesis to design FPGA hardware blocks with the C++ software code. This can save engineers 2-5x design time compared to traditional RTL design.

Figure 1-1. FPGA Design Flow Using RTL vs. HLS



FPGA Design Flow using RTL and HLS

High-Level Synthesis Design Flow for FPGAs

- 1. HLS Software Design: Engineer writes a software model of the design in the C++ software. Software is tested to verify the desired functionality. Engineer generates a hardware block from the C++ software using high-level synthesis. Engineer rewrites the C++ in HLS style, as necessary. For example, using FIFO types for dataflow computation to get the desired throughput.
- 2. Optimization & timing closure: Engineer adds HLS constraints for the desired clock period and tunes the HLS settings. For example, to reduce area by sharing hardware operators. Engineer runs the Libero SoC design suite to verify area/timing.
- 3. Verification: The generated circuit will be corrected by construction. Engineer can use co-simulation to verify the generated RTL block in Mentor Graphics ModelSim[®] simulation. Co-simulation creates a test bench for the tests from the original software model.

2. Benefits of High-Level Synthesis

2.1 HLS Productivity Gain

Engineers developing an FPGA hardware in C++ using high-level synthesis design tools reduce their design effort significantly compared to engineers writing hardware in RTL. Writing software code is much easier for engineers than writing hardware in RTL because software code is more concise, with 5-10x less lines of C++ required than RTL, as shown in the following figure¹.

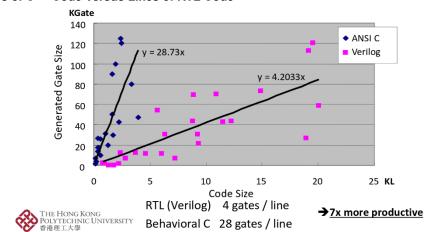


Figure 2-1. Lines of C++ Code Versus Lines of RTL Code

Software code is also much easier to read and understand for future improvements or maintenance compared to RTL. Software conciseness and readability mean less bugs in your FPGA design. Engineers will see a 2-5x reduction in the FPGA design time by using HLS as shown in Figure 1-1. Better designer productivity leads to faster time-to-market. The C++ software code describes an algorithm, which is at a much higher level of abstraction than hardware descriptive languages. For example, in C++ code the programmer does not need to specify any hardware timing constructs like clock cycles. In contrast, a hardware engineer writing RTL must specify cycle-by-cycle hardware behavior using finite state machines (FSM)s, and other control logic. If any control logic is off by one cycle, then circuit functionality will be interrupted. High-level synthesis also supports C++ data types to represent floating or fixed-point computation. These data types are not natively supported in Verilog/VHDL languages, therefore the engineer must handle these operations manually.

High-level synthesis also offers much more flexible hardware module parameterization compared to Verilog/VHDL. In a software description of the design, the C++ template parameters offer more fine-grained control versus simple RTL parameters. Furthermore, HLS user-provided constraints can fine-tune the generated hardware microarchitecture for different latency, timing, and area design constraints. Meanwhile, for an RTL design the engineer must manually add pipeline registers to the data path to meet different timing constraints. Last-minute pipeline changes are painful and difficult to parameterize in RTL.

2.2 Port Existing Software to FPGA

SmartHLS allows engineers to reuse a pre-existing software implementation of their algorithm and port this software to an FPGA. In many cases, this existing software implementation has been verified and tested extensively. For an engineer to manually re-implement the software design in Verilog/VHDL could introduce subtle errors that can be tedious to catch and time consuming to verify. High-level synthesis greatly simplifies this process and avoids manual reimplementation by saving design time and engineering resources.

For example, a real-time control application might include software or firmware running on a microcontroller that can no longer meet the required response time. The solution is to migrate the software to an FPGA, which offers deterministic latency. High-level synthesis allows the engineer to automatically generate Verilog/VHDL from the

¹ Innovative Technology Series (ITS)- High Level Synthesis

existing C++ software. See the customer case-study white paper: Migrating Motor Controller C++ Software from a Microcontroller to a PolarFire FPGA with Smart High-Level Synthesis.

2.3 Faster Verification and Less Bugs

For engineers, verifying the functional correctness of a hardware design written in RTL is usually the most time-consuming part of designing for an FPGA. The engineer must manually write test benches in RTL to simulate their design under various input/output test stimulus. In contrast, an engineer designing hardware in C++ with high-level synthesis can significantly reduce their time spent on verification. While using high-level synthesis, the engineer can write tests in software to verify their design, since the circuit generated from HLS will be correct by construction. Writing tests in the C++ software is simpler than writing hardware test benches in RTL, which usually leads to writing more comprehensive tests increasing test coverage.

SmartHLS supports automatic co-simulation of the generated hardware with ModelSim, by re-using the C++ software tests. During co-simulation, the SmartHLS tool runs the software program with instrumentation added around the top-level C++ function to collect golden inputs/outputs for each function argument. Then SmartHLS automatically generates a hardware test bench that reads the golden inputs/outputs from test vector files. The test bench verifies that the generated hardware module behaviour matches the software model. Engineers can also write their own custom test bench with a more restricted set of tests to be run on the HLS-generated RTL as a sanity check. Much less RTL simulation is necessary with HLS design. RTL simulation times can become long and impractical for larger FPGA designs as the number of tests increase. High-level synthesis allows for software-based testing and verification, which has 100-1000X faster runtime than RTL simulation. Faster software-based verification runtimes mean that engineers using HLS can still verify hardware after last-minute design changes if requirements change late in the design process.

2.4 Design Space Exploration

Design space exploration is the process of making hardware microarchitecture design trade-offs within a set of constraints on FPGA clock frequency, throughput, latency, and area. An RTL engineer will usually decide on a hardware microarchitecture early in the design process. After the RTL is written, an engineer will typically avoid making microarchitecture changes, such as adding pipeline stages, because RTL redesign is a time-consuming process. In contrast, high-level synthesis simplifies design space exploration and allows continuous refinement of the hardware microarchitecture throughout the design process. For example, the HLS tool automatically inserts pipeline registers based on the HLS target clock period constraint. The engineer can easily modify the HLS target clock period to achieve different performance/area targets without manual redesign. HLS typically runs in a few minutes, so the designer can quickly get feedback on resource and throughput estimates without waiting for a time-consuming FPGA synthesis run.

SmartHLS offers a rich set of user-constraints and pragmas for the engineer to specify their desired hardware microarchitecture based on the software description. For example, loops can be pipelined to improve performance, or inner loops can be unrolled. Functions with FIFO dataflow streaming inputs/outputs can be pipelined. C++ arrays can be partitioned into registers or RAMs to achieve better memory bandwidth and performance. High-level synthesis also supports sharing larger hardware operations, such as floating-point cores. HLS constraints give the designer the ability to easily perform more design space exploration, which can lead to better trade-offs between performance and area for their FPGA designs.

2.5 FPGA Device Portability

A verified hardware IP block can be reused for many years. Therefore, future proofing is important if the hardware block could later target another FPGA device family. RTL designs have a specific microarchitecture targeting one FPGA family. Therefore, RTL redesign will be required if the timing constraints are not met while porting. RTL designs may also use primitive blocks specific to the FPGA family, which will need to be rewritten.

High-level synthesis automatically adds pipelining stages to your hardware depending on the target FPGA device and clock period constraints. HLS settings can easily be changed to re-generate a hardware block targeting a new FPGA family while still meeting the timing constraints. Portability is simplified because there are no FPGA family-specific C++ design constructs.

3. Conclusion

In conclusion, FPGA design with C++ using high-level synthesis can offer 2-5x better design productivity compared to writing in RTL. The resulting C++ code is 5-10v shorter than an equivalent RTL design and C++ is expressed at a higher level of abstraction. This results in less bugs and easier readability. HLS software-based verification and testing saves significant design effort compared to RTL verification. HLS also enables easier design space exploration and FPGA device portability.

3.1 **Key Takeaways**

- SmartHLS simplifies FPGA design by allowing you to program the FPGA using C/C++ software
- 2-5X better design productivity
- Higher abstraction level means less code and less bugs
- Faster verification and testing
- Design space exploration
- · FPGA device portability

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- Product Support Data sheets and errata, application notes and sample programs, design resources, user's
 guides and hardware support documents, latest software releases and archived software
- General Technical Support Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- Business of Microchip Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- · Distributor or Representative
- · Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal
 conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features
 of the Microchip devices. We believe that these methods require using the Microchip products in a manner
 outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code
 protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code
 protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly
 evolving. We at Microchip are committed to continuously improving the code protection features of our products.
 Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act.
 If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue
 for relief under that Act.

© 2021 Microchip Technology Inc. White Paper DS50003181A-page 7

Legal Notice

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, Anyln, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2021, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-8676-3

Quality Management System							
For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.							



Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	Australia - Sydney	India - Bangalore	Austria - Wels
2355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	China - Beijing	India - New Delhi	Fax: 43-7242-2244-393
Tel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	Denmark - Copenhagen
Fax: 480-792-7277	China - Chengdu	India - Pune	Tel: 45-4485-5910
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
www.microchip.com/support	China - Chongqing	Japan - Osaka	Finland - Espoo
Web Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
www.microchip.com	China - Dongguan	Japan - Tokyo	France - Paris
Atlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Duluth, GA	China - Guangzhou	Korea - Daegu	Fax: 33-1-69-30-90-79
Tel: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	Germany - Garching
Fax: 678-957-1455	China - Hangzhou	Korea - Seoul	Tel: 49-8931-9700
Austin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	Germany - Haan
Tel: 512-257-3370	China - Hong Kong SAR	Malaysia - Kuala Lumpur	Tel: 49-2129-3766400
Boston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	Germany - Heilbronn
Westborough, MA	China - Nanjing	Malaysia - Penang	Tel: 49-7131-72400
Tel: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	Germany - Karlsruhe
Fax: 774-760-0088	China - Qingdao	Philippines - Manila	Tel: 49-721-625370
Chicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	Germany - Munich
Itasca, IL	China - Shanghai	Singapore	Tel: 49-89-627-144-0
Tel: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Fax: 630-285-0075	China - Shenyang	Taiwan - Hsin Chu	Germany - Rosenheim
Dallas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
Addison, TX	China - Shenzhen	Taiwan - Kaohsiung	Israel - Ra'anana
Tel: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
Fax: 972-818-2924	China - Suzhou	Taiwan - Taipei	Italy - Milan
Detroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
Novi, MI	China - Wuhan	Thailand - Bangkok	Fax: 39-0331-466781
Tel: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	Italy - Padova
Houston, TX	China - Xian	Vietnam - Ho Chi Minh	Tel: 39-049-7625286
Tel: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	Netherlands - Drunen
Indianapolis	China - Xiamen		Tel: 31-416-690399
Noblesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
Tel: 317-773-8323	China - Zhuhai		Norway - Trondheim
Fax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
Tel: 317-536-2380			Poland - Warsaw
Los Angeles			Tel: 48-22-3325737
Mission Viejo, CA			Romania - Bucharest
Tel: 949-462-9523			Tel: 40-21-407-87-50
Fax: 949-462-9608			Spain - Madrid
Tel: 951-273-7800 Raleigh, NC			Tel: 34-91-708-08-90
• .			Fax: 34-91-708-08-91
Tel: 919-844-7510 New York, NY			Sweden - Gothenberg Tel: 46-31-704-60-40
Tel: 631-435-6000			Sweden - Stockholm
San Jose, CA			Tel: 46-8-5090-4654
Tel: 408-735-9110			UK - Wokingham
Tel: 408-436-4270			Tel: 44-118-921-5800
Canada - Toronto			Fax: 44-118-921-5820
Tel: 905-695-1980			1 ax. 77-110-321-0020
Fax: 905-695-2078			
1 dx. 505-050-2010			