

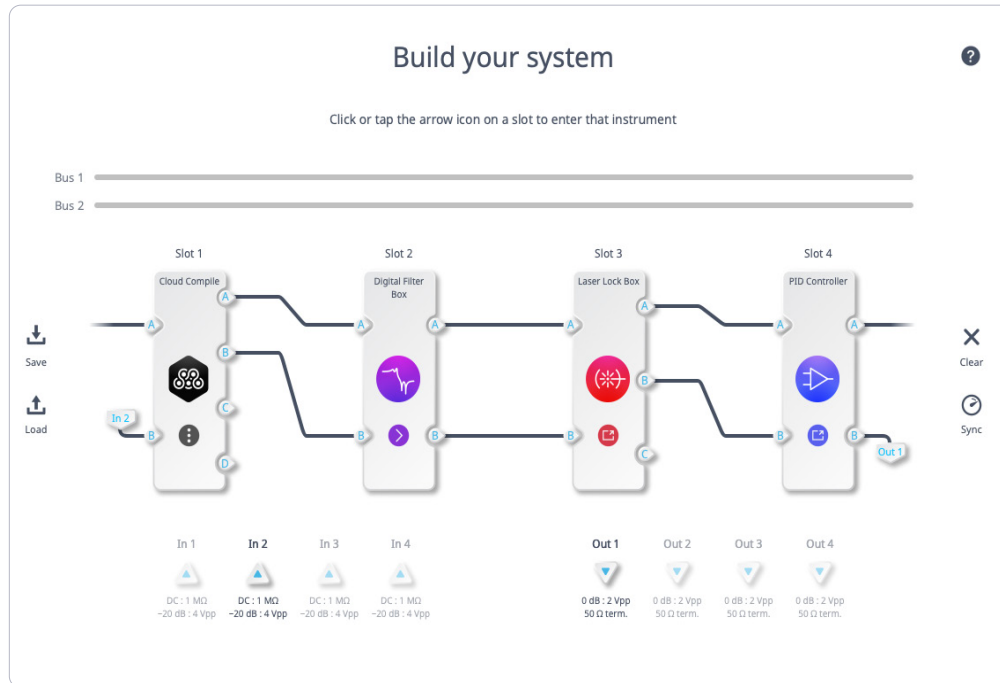


Moku Cloud Compile

Deploy custom VHDL code to Moku



Moku Cloud Compile (MCC) enables users to deploy custom code directly on Moku hardware, allowing for unprecedented customization capabilities with minimal development time. A custom wrapper is available to ensure Moku Cloud Compile creations can interface directly with the front end of any Moku. Run custom designs alongside any supported instrument in the Moku suite in Multi-instrument Mode. By opening access to the FPGA that enables Moku hardware, new features or functions can be deployed in minutes. The cloud-based workflow greatly simplifies setup and deployment.



Example use cases

- Create custom math functions, gain, or offset and create more complex signal processing flows
- Design and deploy a custom instrument like a boxcar averager, specific DC sequencing, or more advanced cross correlation functions
- Create custom pulse or function sequences
- Digitally sum, subtract, or otherwise combine signals, or create gaussian noise for testing
- Combine with MathWorks® Simulink and MATLAB HDL Coder to produce noise generators, random sequences, or square root functions

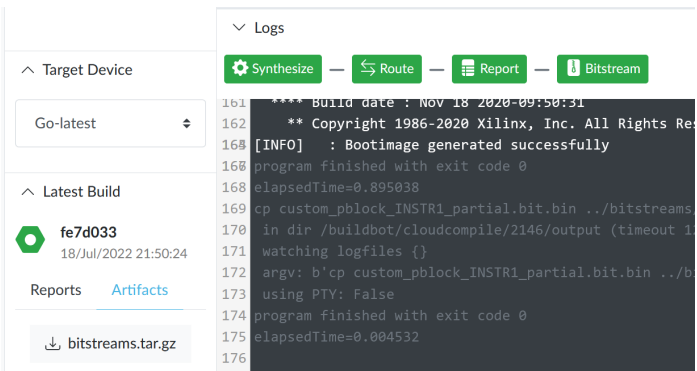
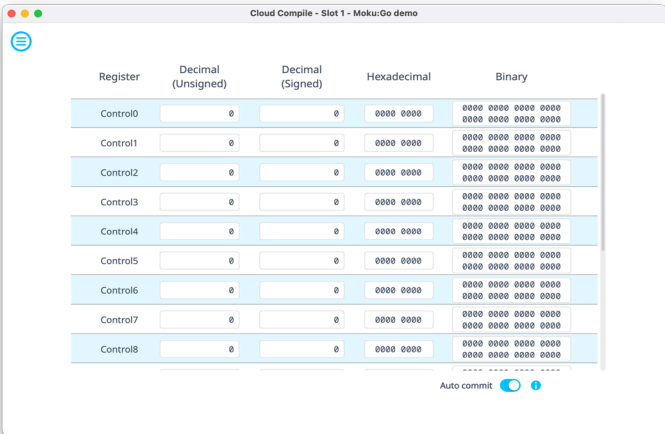
```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.numeric_std.ALL;
4
5  ARCHITECTURE Behavioural OF CustomWrapper IS
6  -- Component Declarations
7  COMPONENT AWGN -- Check if this matches your instrument name
8  PORT( Clk
9        : IN      std_logic;
10       Reset
11       : IN      std_logic;
12       InputA
13       : IN      signed(15 DOWNTO 0); -- int16
14       InputB
15       : IN      signed(15 DOWNTO 0); -- int16
16       Control0
17       : IN      signed(15 DOWNTO 0); -- int16
18       Control1
19       : IN      signed(15 DOWNTO 0); -- int16
20       OutputA
21       : OUT     signed(15 DOWNTO 0); -- int16
22       OutputB
23       : OUT     signed(15 DOWNTO 0); -- int16
24     );
25  END COMPONENT;
26
27 BEGIN
28   u_AWGN : AWGN -- Check if this matches your instrument name
29   PORT MAP( Clk=>Clk,
30             Reset=>Reset,
31             InputA=>InputA,
32             InputB=>InputB,
33             Control0=>Control0(15 downto 0),
34             Control1=>Control1(15 downto 0),
35             OutputA=>OutputA,
36             OutputB=>OutputB
37           );
38 END Behavioural;
```

Get started with free example code available at <https://compile.liquidinstruments.com/docs/>

Powerful customization made easy

Multi-instrument Mode allows Moku Cloud Compile to interact with the Moku suite of standard instruments. For example, the Oscilloscope can be used to observe the custom code outputs, or use the custom code to drive modulation onto a Waveform Generator signal. By harnessing the power of the existing instrument suite, users can create, compile, and validate their designs on one hardware platform.

Control custom code via app-based registers to adjust gains or change operating modes in real time. Register changes are reflected immediately in the running VHDL for quick adjustment of design parameters.



Learn, explore, deploy

Compile, synthesize, and route entirely in the cloud. Users can code in VHDL or use third-party code generators like HDL Coder from Mathworks.

The synthesized design forms a bitstream file that is hardware deployable via Moku: Apps. Cloud-based synthesis allows users to avoid the need to install, debug, and maintain complex local tools.

Choose your hardware device

Moku:Pro




Customize advanced test setups by deploying custom DSP alongside up to three additional instruments in a 4-slot Multi-instrument Mode configuration, enabling you to tailor your test equipment to your requirements.

Moku:Lab

Run two instruments simultaneously to build custom test systems. Deploy custom code through Moku Cloud Compile in one slot, then test your design with any Moku instrument in the other.

Moku:Go

Quickly iterate on VHDL programs without complex software installations. Deploy Moku Cloud Compile creations alongside another instrument in Multi-instrument Mode to test designs in realtime.

	 Moku:Pro	 Moku:Lab	 Moku:Go
FPGA	Ultrascale+	Zynq 7020	Zynq 7020
Core clock	312.5 MHz	125 MHz	31.25 MHz
Look-up tables	48,400	19,600	20,000
Flip-flops	96,800	39,200	40,000
Block RAM (36k)	154	60	50
DSP	432	100	100

All specifications are per slot