FPGA Resource Manager

# Problem 1: Integrating Multiple Custom Devices in to One Bitfile

## Overview

When developing Custom Devices that use an R-Series FPGA, there is the FPGA code and a host portion that runs in the FPGA engine. It is uncommon for a single Custom Device to require all the IO on an FPGA, so it is possible to use the same FPGA as the engine for multiple Custom Devices.

Currently, several of the partners have distributed their Custom Devices with a bitfile rather than a VI to protect their IP, making this impossible.

## Proposed Solution

To get around this, we will specify how the engine for a Custom Device should be developed. Namely, there will be an “FPGA Engine” VI that is the internal logic of the Custom Device.

### FPGA Engine VI

The device developer will create and distribute an FPGA VI that can be placed inside a loop with other FPGA VIs. The VI will have several different kinds of inputs:

#### User Inputs

These are Read/Write Controls that will be passed up to the VeriStand System Explorer and selected by the end user of the Custom Device.

#### Integrator Inputs

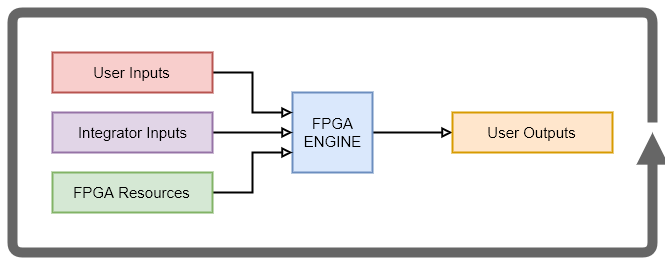
These are constant values that the integrator creating the top-level FPGA VI provides.

#### FPGA Resources

These are FPGA resources such as DMA FIFOs, Global Registers, and other FPGA hardware resources.

#### User Outputs

These are processing results that will be passed up to the host via Read/Write Controls



To further simplify the development of the top-level FPGA VI, we will also encourage the use of a Placer VI. This VI goes on the palette, and when selected places the engine and all of the associated controls and indicators.

# Problem 2: Mapping the Correct Resource in the System Definition

## Overview Example Proposed Solution

# Problem 3: Ensuring Engine Compatibility Between Custom Devices

## Overview Example Proposed Solution