

# Zynq UltraScale+ MPSoC: RPU Introduction Demo Script

## Introduction

This demonstration provides high-level instructions on how to run a bare-metal application on an Arm® Cortex®-R5 processor using QEMU command line options.

### Preparation:

- Required files: `UED_zcu104.xsa`  
The above file can be found in the `$CustEdIP` directory.
- Required hardware: None
- Required software: Vitis™ Unified IDE and QEMU

## Zynq UltraScale+ MPSoC: RPU Introduction

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"><li>• Click the <b>Vitis</b> icon from the taskbar to launch the Vitis Unified IDE.</li></ul>	<ul style="list-style-type: none"><li>• The Workspace Launcher opens after a moment.</li></ul>
<ul style="list-style-type: none"><li>• Close the Welcome tab if it is open.</li><li>• From the Vitis Components window, click <b>Open Workspace</b>.</li></ul>	<ul style="list-style-type: none"><li>• The Vitis Unified IDE creates a workspace environment that initially only contains a thin structure that tracks tool settings and maintains the log file. As projects are added, this workspace will update to include hardware projects, BSPs, and your software applications.</li></ul>

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> <li>• Select <b>File &gt; New Component &gt; Platform</b> to create a new platform component.</li> <li>• Enter the platform name as <b>rpu_intro_plat</b>.</li> <li>• Ensure that <b>Hardware Design</b> is selected.</li> <li>• Click <b>Browse</b>, navigate to \$TRAINING_PATH/CustEdIP, and select <b>UED_zcu104.xsa</b>.</li> <li>• Click <b>Next</b>.</li> <li>• Select <b>psu_cortexr5_[0]1</b> as the processor.</li> <li>• Select <b>standalone</b> as the operating system.</li> <li>• Click <b>Next</b> and then click <b>Finish</b> to generate the platform component.</li> </ul>	<ul style="list-style-type: none"> <li>• You are targeting the Zynq™ UltraScale+™ MPSoC platform for the ZCU104 board. Since there is no predefined hardware platform provided in the Vitis Unified IDE, you will need to create a new hardware platform using the provided XSA.</li> <li>• The platform component contains the domain and BSP, along with the FSBL and PMU firmware. <ul style="list-style-type: none"> <li>• The domain contains the BSP for the specific processor for which the application is being targeted.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Select <b>File &gt; New Example &gt; Embedded Software Examples</b> to create a new application component.</li> <li>• Select <b>Hello World</b> from the available examples and click the "+" symbol.</li> <li>• Enter the application name as <b>hello_world</b>.</li> <li>• Click <b>Next</b>.</li> </ul>	<ul style="list-style-type: none"> <li>• This opens the Create Application Component Wizard.</li> <li>• You will find many applications available as examples: <ul style="list-style-type: none"> <li>• DDR Self Refresh</li> <li>• Empty Application</li> <li>• Hello World</li> <li>• Libmetal AMP Demo</li> <li>• lwIP Echo Server</li> <li>• Memory Tests</li> <li>• OpenAMP echo-test</li> <li>• OpenAMP Matrix Multiplication Demo</li> <li>• OpenAMP RPC Demo</li> <li>• Peripheral Tests</li> <li>• Zynq MP FSBL, and many more</li> </ul> </li> </ul>

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> <li>Select <b>rpu_intro_plat</b> as the platform project for this application.</li> <li>Click <b>Next</b>.</li> <li>The domain settings can be left at default, so click <b>Next</b>.</li> <li>Click <b>Finish</b>.</li> </ul>	<ul style="list-style-type: none"> <li>Using the Create Application Component Wizard is a quick way to set up a software application component that targets an existing processor and OS platform (standalone or Linux).</li> <li>You will find two components in the Vitis Components window: <ul style="list-style-type: none"> <li>Platform component (rpu_intro_plat)</li> <li>Application component (hello_world)</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>From the Flow view, select <b>hello_world</b> and click <b>Build</b>.</li> </ul>	<ul style="list-style-type: none"> <li>This will build the platform and application components and generate the ELF file used for running the application.</li> </ul>
<ul style="list-style-type: none"> <li>Select <b>Terminal &gt; New Terminal</b>.</li> </ul>	<ul style="list-style-type: none"> <li>This will open the Vitis terminal. QEMU commands are run from here.</li> <li>The current release of the Vitis Unified IDE does not support QEMU. A Tcl and a makefile are provided as a workaround to view the output had the application run on QEMU.</li> </ul>
<ul style="list-style-type: none"> <li>Run the following commands to launch QEMU:  <pre>[host] \$ cd \$TRAINING_PATH/rpu_intro/support</pre> <pre>[host] \$ ./rpu_intro_demo_QemuRun.sh r5</pre> </li> <li>This will print:  <pre>Hello World</pre> </li> </ul>	<ul style="list-style-type: none"> <li>The files required for running QEMU are provided in the <code>CustEdIP</code> directory.</li> <li>These files need to be copied to the workspace directory for the workaround QEMU to run.</li> <li>Since the real-time processor (RPU) is being targeted, the Arm® Cortex®-R5 processor is chosen.</li> </ul>

## Summary

You learned how to configure QEMU in the Vitis Unified IDE and saw a bare-metal application run on the Arm Cortex-R5 processor.