Imperas Guide to using Virtual Platforms

Platform / Module Specific Information for safepower.ovpworld.org / Zynq_PL_TTELNoC_public_demonstrator

Imperas Software Limited

Imperas Buildings, North Weston
Thame, Oxfordshire, OX9 2HA, U.K.
docs@imperas.com

MULTICORE DESIGN SIMPLIFIED

imperas

Author	Imperas Software Limited
Version	20211118.0
Filename	Imperas_Platform_User_Guide_Zynq_PL_TTELNoC_public_demonstrator.pdf
Created	31 December 2021
Status	OVP Standard Release

www.imperas.com

Copyright Notice

Model Release Status

This model is released as part of OVP releases and is included in OVPworld packages. Please visit OVPworld.org.

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 2 of 18

Table Of Contents

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 3 of 18

1.0 Platform / Module: Zynq_PL_TTELNoC_public_demonstrator

This document provides the details of the usage of an Imperas OVP Virtual Platform / Module. The first half of the document covers specifics of this particular component. For more information about Imperas OVP virtual platforms, how they are built and used, please see the later sections in this document.

1.1 Virtual Platform / Module Type

Hardware described using OVP can either be a platform, module, processor, or peripheral.
This hardware component is described as being a module. A module is a component that is used in other modules, platforms, or test harnesses. It is normally used to encapsulate a layer in a hierarchical system.

1.2 Licensing

Open Source Apache 2.0

1.3 Description

This module implements the Top Level SafePower Public Demonstrator Xilinx Zynq Programmable Logic (PL).
This PL configuration instances SafePower Public Demonstrator Processing Node modules and a Sensor Actor Node module.
These modules are all interconnected by a TTEL NoC.
Also included is a TTEL NoC interface peripheral that is accessible from the Zynq_PS ARM processors.

1.4 Limitations

Provides a baremetal implementation.

1.5 Reference

SafePower Public Demonstrator

1.6 Location

The Zynq_PL_TTELNoC_public_demonstrator virtual platform / module is located in an Imperas/OVP installation at the VLNV: safepower.ovpworld.org / module / Zynq_PL_TTELNoC_public_demonstrator / 1.0.

1.7 Module Simulation Attributes

Table 1. Module Simulation Attributes

Attribute	Value	Description
stoponctrlc	stoponctrlc	Stop on control-C

2.0 External Ports for Module Zynq_PL_TTELNoC_public_demonstrator

Table 2. External Ports

Port Type	Port Name	Internal Connection

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 5 of 18

busport	extPort	extPortBus
netport	gpio_bank2_outP	gpio_bank2_out
netport	gpio_bank2_oen_outP	gpio_bank2_oen_out
netport	gpio_bank2_inP	gpio_bank2_in
netport	gpio_bank3_outP	gpio_bank3_out
netport	gpio_bank3_oen_outP	gpio_bank3_oen_out
netport	gpio_bank3_inP	gpio_bank3_in
netport	irqf2p0_outP	irqf2p0
netport	irqf2p1_outP	irqf2p1
netport	irqf2p2_outP	irqf2p2
netport	irqf2p3_outP	irqf2p3
netport	irqf2p4_outP	irqf2p4
netport	irqf2p5_outP	irqf2p5
netport	irqf2p6_outP	irqf2p6
netport	irqf2p7_outP	irqf2p7
netport	irqf2p8_outP	irqf2p8
netport	irqf2p9_outP	irqf2p9
netport	irqf2p10_outP	irqf2p10
netport	irqf2p11_outP	irqf2p11
netport	irqf2p12_outP	irqf2p12
netport	irqf2p13_outP	irqf2p13
netport	irqf2p14_outP	irqf2p14
netport	irqf2p15_outP	irqf2p15
netport	irqf2p16_outP	irqf2p16
netport	irqf2p17_outP	irqf2p17
netport	irqf2p18_outP	irqf2p18
netport	irqf2p19_outP	irqf2p19
netport	irqp2f0_inP	irqp2f0
netport	irqp2f1_inP	irqp2f1
netport	irqp2f2_inP	irqp2f2
netport	irqp2f3_inP	irqp2f3
netport	irqp2f4_inP	irqp2f4
netport	irqp2f5_inP	irqp2f5
netport	irqp2f6_inP	irqp2f6
netport	irqp2f7_inP	irqp2f7
netport	irqp2f8_inP	irqp2f8
netport	irqp2f9_inP	irqp2f9
netport	irqp2f10_inP	irqp2f10
netport	irqp2f11_inP	irqp2f11
netport	irqp2f12_inP	irqp2f12
netport	irqp2f13_inP	irqp2f13
netport	irqp2f14_inP	irqp2f14
netport	irqp2f15_inP	irqp2f15
netport	irqp2f16_inP	irqp2f16
netport	irqp2f17_inP	irqp2f17
netport	irqp2f18_inP	irqp2f18
netport	irqp2f19_inP	irqp2f19
netport	irqp2f20_inP	irqp2f20

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 6 of 18

netport	irqp2f21_inP	irqp2f21
netport	irqp2f22_inP	irqp2f22
netport	irqp2f23_inP	irqp2f23
netport	irqp2f24_inP	irqp2f24
netport	irqp2f25_inP	irqp2f25
netport	irqp2f26_inP	irqp2f26
netport	irqp2f27_inP	irqp2f27
netport	irqp2f28_inP	irqp2f28

3.0 Sub-Module [safepower.ovpworld.org/module/Zynq_PL_TTELNoC_sensor_actor_node_public_demonstrator/1.0] instance: NoC_0_1

Table 3. Sub-Module Instance 'NoC_0_1' Connections

Port Type	Port Name	Connection
packetnetport	networkNodePort	network

Table 4. Sub-Module Instance 'NoC_0_1' Parameters / Attributes set

Name	Value	Type	Expression
cluster	0	uns32	
tile	0	uns32	
node	1	uns32	

4.0 Sub-Module [safepower.ovpworld.org/module/Zynq_PL_TTELNoC_processing_node_public_demonstrator/1.0] instance: NoC_1_0

Table 5. Sub-Module Instance 'NoC_1_0' Connections

Port Type	Port Name	Connection
packetnetport	networkNodePort	network

Table 6. Sub-Module Instance 'NoC_1_0' Parameters / Attributes set

Name	Value	Type	Expression
cluster	0	uns32	
tile	0	uns32	
node	2	uns32	

5.0 Sub-Module [safepower.ovpworld.org/module/Zynq_PL_TTELNoC_processing_node_public_demonstrator/1.0] instance: NoC_1_1

Table 7. Sub-Module Instance 'NoC_1_1' Connections

Port Type	Port Name	Connection
packetnetport	networkNodePort	network

Table 8. Sub-Module Instance 'NoC_1_1' Parameters / Attributes set

Name	Value	Type	Expression
cluster	0	uns32	
tile	0	uns32	

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 7 of 18

node	3	uns32	

6.0 Peripheral Instances

6.1 Peripheral [safepower.ovpworld.org/peripheral/TTELNode/1.0] instance: NoC_0_0

6.1.1 Description
The TTEL Network on Chip (NoC) node peripheral for SafePower Project

6.1.2 Licensing
Open Source Apache 2.0

6.1.3 Limitations
This model implements the TTEL NoC node processor interface. It does not model any timing in the transfer of messages between nodes.

6.1.4 Reference
Generated using document TTEL Software Extensions ver 1.0 and D1.2.1 architectural style of dreams r1-0.

Table 9. Configuration options (attributes) set for instance 'NoC_0_0'

Attribute	Value	Type	Expression
cluster	0	uns32	
tile	0	uns32	
node	0	uns32	

6.2 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: dummy

6.2.1 Description
Open a port and allocate a region that is defined by parameters.
The region can be configured to act as standard memory or can report read/write accesses.

6.2.2 Licensing
Open Source Apache 2.0

6.2.3 Limitations
This peripheral cannot be used in a hardware description used to generate a TLM platform.

6.2.4 Reference
This is not based upon the operation of a real device but is intended to be used for bring up and development of new virtual platforms.

Table 10. Configuration options (attributes) set for instance 'dummy'

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 8 of 18

Attribute	Value	Type	Expression
portAddress	0x60000000	uns32	
portSize	0x30000	uns32	
cbEnable	1	bool	

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 9 of 18

7.0 Overview of Imperas OVP Virtual Platforms

This document provides the details of the usage of an Imperas OVP Virtual Platform / Module. The first half of the document covers specifics of this particular virtual platform / module.

This second part of the document, includes information about Imperas OVP virtual platforms and modules, how they are built and used.

The Imperas virtual platforms are designed to provide a base for you to run high-speed software simulations of CPU-based SoCs and platforms on any suitable PC. They are typically based on the functionality of vendors fixed or evaluation platforms, enabling you to simulate software on these reference platforms. Typically virtual platforms are fixed and require the vendor to modify or extend them. Imperas virtual platforms are different in that they enable you to extend the functionality of the virtual platform, to closer reflect your own platform, by adding more component models, running different operating systems or adding additional applications.

Imperas virtual platforms are created using the Imperas iGen technology, allowing them to be used with Imperas OVP based simulators and also with Accellera/OSCI compliant SystemC simulators and commercial EDA System Design environments that use SystemC.

Virtual platforms include simulation models of the target devices, including the processor model(s) for the target device plus enough peripheral models to boot an operating system or run bare metal applications. The platform and the peripheral models used in most of the virtual platforms are open source, so that you can easily add new models to the platform as well as modify the existing models. Some models are only provided as binary, normally because the IP owner has restricted the release of the model source. In this case, please contact Imperas for more information.

There are typically several generic flavors of the virtual platforms for specific processor families, some targeting full operating systems, such as Linux, and some which focus on Real Time Operating Systems (RTOS) such as Mentor Nucleus or freeRTOS. OVP models of the processor cores are included in the virtual platforms, and for those processors which support mulitple cores SMP Linux is often supported for that virtual platform. For all of these virtual platforms, many of the peripheral components of the platform are modeled, often including the Ethernet and USB components. The semi-hosting capability of the Imperas virtual platform simulator products enables connection via the Ethernet and USB components from the virtual platform to the real world via the x86 host machine.

The Imperas OVP CPU models are written using the OVP Virtual Machine Interface (VMI) API that defines the behavior of the processor. The VMI API makes a clear line between model and simulator allowing very good optimization and world class high speed performance. The processor models are Instruction Accurate and do not model the detailed cycle timing of the processor and they implement functionality at the level of a Programmers View of the processor and peripherals and the software running on them does not know it is not running on hardware. Many models are provided as a binary shared object and also as source. This allows the download and use of the model binary or the use of the source to explore

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 10 of 18

and modify the model. The models are run through an extensive QA and regression testing process and most processor model families are validated using technology provided by the processor IP owners. All the models in this platform are developed with the Open Virtual Platforms APIs and are implemented in C. A platform can be modeled as different levels of hierarchy using separately describable and compilable modules.

More information on modeling and APIs can be found on the www.OVPworld.org site.

8.0 Getting Started with Imperas OVP Virtual Platforms

Virtual platforms are downloadable from the OVPworld website OVPworld.org/downloads. You need to browse and look for '<platform processor name> Examples'. You do need to be registered and logged in on the OVP site to download. OVPworld currently provides 32 bit host versions of packages containing virtual platforms.

When downloading, choose, Linux or Windows host. 32 bit packages can be installed and executed on 32 bit or 64 bit hosts. If you require a 64 bit host version please contact Imperas.

For example, for the ARM Versatile Express platform booting Linux on Cortex-A15MP Single, Dual, and Quad core procesors, you would want the download package: 'OVPsim_demo_Linux_ArmVersatileExpress_arm_Cortex-A15MP'.

Most virtual platform packages contain the platform and all the processor and peripheral models needed. You will need to download a simulator to run the platform. You can use OVPsim, downloadable from OVPworld.org/downloads, or you can use one of the Imperas simulators (imperas.com/products) available commercially from Imperas.

9.0 Simulating Software

9.1 Getting a license key to run

After you have downloaded you will need a runtime license key before the simulators will run. For OVPsim please visit OVPworld.org/likey and provide the required information and an evaluation/demo license key will be automatically sent to you. If you are using Imperas, then please contact Imperas for a license key.

9.2 Normal runs

To run a platform, read the section below on command line control of the platform and the section on setting command line arguments.

9.3 Loading Software

For most virtual platforms the platform is already configured to run the default software application/program and there is normally a script to run that sets some arguments. You can then copy/edit this script to select your own applications etc.

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 11 of 18

The example application programs are typically .elf format files and are provided pre-compiled. There are normally makefiles and associated scripts to recompile the example applications.

To find more information about compiling and loading software, the following document should be looked at: Imperas_Installation_and_Getting_Started.pdf.

9.4 Semihosting
In a virtual platform, semihosting is not normally used as there is normally hardware that implements the appropriate functionality - for example I/O will be handled by UARTs etc.

9.5 Using a terminal (UART)
If the platform includes one or more UARTs you will need to connect a terminal program to it so that you can see output and type into the simulated program. Review the list of peripherals below and see what configuration options it has been set with. In most cases there is an option to set to instruct the simulator to 'pop up' a terminal window connected to the simulated UART.

9.6 Interacting with the simulation (keyboard and mouse)
If the platform has a simulated UART you can normally set a command to get the simulator to pop up a terminal window allowing you to see output from the simulated UART and also allowing you to type characters into the UART that can be processed by the simulated software.

If your simulated platform has an LCD device then you can often configure it to recognize mouse movements and mouse clicks - allowing full interaction.

To see these interactions in action, have a look at some of the available videos available at OVPworld.org/demosandvideos.

9.7 More Information (Documentation) on Simulation
To find more information about running simulations and more of the options the simulators provide, the following documents should be looked at:
Imperas_Installation_and_Getting_Started.pdf
Simulation_Control_of_Platforms_and_Modules_User_Guide.pdf
Advanced_Simulation_Control_of_Platforms_and_Modules_User_Guide.pdf
OVP_Control_File_User_Guide.pdf

A full list of the currently available OVP documentation is available: OVPworld.org/documentation.

10.0 Debugging Software running on an Imperas OVP Virtual Platform
The Imperas and OVP simulators have several different interfaces to debuggers. These include several proprietary formats and also the standard GNU RSP format is supported allowing many compatible debuggers to be used. Below are some examples that Imperas directly support.

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 12 of 18

10.1 Debugging with GDB
A GNU debugger (GDB) can be connected to a processor in a platform using the RSP protocol. This allows the application program running on a processor to be debugged using a specific GDB for the processor selected. When using the Imperas Professional products many connections can be made allowing a GDB to be connected to all the processors in the platform.

The use of GDB is documented: OVPsim_Debugging_Applications_with_GDB_User_Guide.pdf.

10.2 Debugging with Imperas M*DBG
The Imperas multi-processor debugger can be connected to a platform and through this connection you can debug application programs running on all of the processors instanced within the platform. It is also capable, within this single unified environment, to debug peripheral model behavioral code in conjunction with the processor application programs.

For more information please see the Imperas M*DBG user guide.

The Imperas multi-processor debugger is also capable of controlling the Imperas Verification Analysis aand Profiling (VAP) tools in real time, making them invaluable to application program development, debugging and analysis.

For more information please see the Imperas VAP tools user guide.

10.3 Debugging with the Imperas eGui and GDB
Imperas eGui gives a GUI front end to the use of the GDB debugger. It allows use of all the features of GDB including source level application program debugging on processors.

10.4 Debugging with the Imperas eGui and M*DBG
Imperas eGui gives a GUI front end to the Imperas multi-processor debugger. It provides all the features of this debugger but does so with source level application program debugging on processors and source level debugging of the behavioral code on peripheral components in the platform. A context view shows all the processor and peripheral components within the platform and allows switching between them to examine the state of each at the event at which the simulation was stopped

Imperas eGui provides a menu from which the Imperas VAP tools can be controlled.

10.5 Debugging with Imperas eGui and Eclipse
Imperas provide a GUI based on Eclipse called eGui. This provides a GUI front end to use with a standard GDB or the Imperas MPD (Multi-Processor Debugger).

The use of eGui is documented: eGui_Eclipse_User_Guide.pdf.

A standard Eclipse CDT development environment can be connected to one or more processors in a

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 13 of 18

platform (multiple processors require an Imperas professional product). The simulation platform is started remotely or using the external tool feature in Eclipse, opens a debug port and awaits the connection with Eclipse. All features provided by the Eclipse CDT development environment are available to be used to debug software applications executing on the processors in the platform.

The use of Eclipse is documented: OVPsim_Debugging_Applications_with_Eclipse_User_Guide.pdf.

10.6 Debugging applications running under a simulated operating system
If the simulated platform is running an Operating System and the platform has a UART or Ethernet etc connection then it is often possible to connect an external debugger and debug the applications running under the simulated operating system.

An example would be a simulated platform running the Linux operating system, such as the MIPS Malta, or ARM Versatile Express. Within the simulated Linux you can start a gdbserver that connects from within the simulation through a UART out to the host PC via a port. Within the host PC you start a terminal program and connect to the port with a debugger such as GDB and can then debug the simulated user application.

11.0 Modifying the Platform / Module

11.1 Platforms / Modules use C/C++ and OVP APIs
The Imperas and OVP simulators execute a platform / module that is written in C/C++ and that makes function calls into the simulator's APIs. Thus the virtual platform / module is compiled from C/C++ into a binary shared object that the simulator loads and runs. OVP provides the definition and documentation that defines the C APIs for modeling the platforms, modules, the peripherals, and the processors. You can find more information about these APIs on the OVP website and in the OVP API documentation.

11.2 Platforms/Modules/Peripherals can be easily built with iGen from Imperas
Imperas provides a product 'iGen' that takes an input script file and creates the C/C++ files needed for platforms, modules, and peripherals - it creates the C/C++ file that is compiled into the platform, module or peripheral that is needed as an object file by the simulator. iGen creates the C/C++ files, you then need to add any necessary behaviors or further details etc. For platforms iGen creates either a C platform or a C++ SystemC TLM2 platform. For peripherals or modules iGen creates the C files and also provides a native C++ SystemC TLM2 interface to allow the peripheral/module to be instantiated in SystemC TLM2 platforms.

Information on iGen is available from: imperas.com/products.

11.3 Re-configuring the platform
There will nornmally be several configuration options that you can set when running the platform without the neeed to change any source. Refer to the section above on command line arguments. If these do not allow you to make the changes you need, then you may need to edit and recompile the source of the platform.

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 14 of 18

The source of the platform, modules, and the source of the peripherals will be installed as part of the packages you are using. The sources are located in the Imperas/OVP installation VLNV source tree. The VLNV term refers to: Vendor (eg arm.ovpworld.org), Library (eg platform), Name, (eg ArmVersatileExpress-CA15), and Version (eg 1.0). To modify the platform, locate the platform source files.

If you are an Imperas user and have access to iGen, we recommend you modify the source script files and regenerate and recompile the C that makes up the platform. Refer to the Imperas iGen model generator guide and the Imperas platform generator guide.

If you are using the C or SystemC TLM2 platforms with OVPsim, then you can edit the C/C++ files, recompile the source directly using the supplied makefiles, and the run the simulator directly with the resultant shared object.

11.4 Replacing peripherals components
If you need to replace peripherals, find the appropriate place in the source of the platform, make the change you need, and recompile etc. Look in the library for documentation on available peripherals and their configuration options.

11.5 Adding new peripherals components
If you need to add peripherals, find the appropriate place in the source, make the additions you need, and recompile etc. Look in the library for documentation on available peripherals and their configuration options.

If you need to create new peripheral components then use iGen to very quickly create the necessary C/C++ files that get you started. With iGen you can create peripherals with register/memory state in a few lines of iGen source. When adding behavior to the peripherals refer to the OVP API documentation.

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 15 of 18

12.0 Available Virtual Platforms

Table 11. Imperas / OVP Extendable Platform Kits (13 available)

Name	Vendor
AlteraCycloneIII_3c120	altera.ovpworld.org
AlteraCycloneV_HPS	altera.ovpworld.org
ArmIntegratorCP	arm.ovpworld.org
ArmVersatileExpress	arm.ovpworld.org
ArmVersatileExpress-CA15	arm.ovpworld.org
ArmVersatileExpress-CA9	arm.ovpworld.org
AtmelAT91SAM7	atmel.ovpworld.org
FreescaleKinetis60	freescale.ovpworld.org
FreescaleKinetis64	freescale.ovpworld.org
FreescaleVybridVFxx	freescale.ovpworld.org
MipsMalta	mips.ovpworld.org
RenesasUPD70F3441	renesas.ovpworld.org
XilinxML505	xilinx.ovpworld.org

Table 12. Imperas General Virtual Platforms (6 available)

Name	Vendor
arm-ti-eabi	arm.imperas.com
armm-ti-coff	arm.imperas.com
armm-ti-eabi	arm.imperas.com
HeteroAlteraCycloneV_HPS_CycloneIII_3c120	imperas.ovpworld.org
HeteroArmNucleusMIPSLinux	imperas.ovpworld.org
SiFiveFU540	imperas.ovpworld.org

Table 13. Imperas Modules (component of other platforms) (55 available)

Name	Vendor
AlteraCycloneIII_3c120	altera.ovpworld.org
AlteraCycloneV_HPS	altera.ovpworld.org
AE350	andes.ovpworld.org
ARMv8-A-FMv1	arm.ovpworld.org
ArmIntegratorCP	arm.ovpworld.org
ArmVersatileExpress	arm.ovpworld.org
ArmVersatileExpress-CA15	arm.ovpworld.org
ArmVersatileExpress-CA9	arm.ovpworld.org
AtmelAT91SAM7	atmel.ovpworld.org
ArmCortexMFreeRTOS	imperas.ovpworld.org
ArmCortexMuCOS-II	imperas.ovpworld.org
ArmuKernel	imperas.ovpworld.org
ArmuKernelDual	imperas.ovpworld.org
BareMetalMIPS	imperas.ovpworld.org
Dual_ARMv8-A-FMv1_VLAN	imperas.ovpworld.org
Hetero_1xArm_3xMips32	imperas.ovpworld.org
Hetero_ARM_RISCV_NeuralNetwork	imperas.ovpworld.org

Hetero_ARMv8-A-FMv1_Cortex-M3	imperas.ovpworld.org
Hetero_ARMv8-A-FMv1_MIPS_microAptiv	imperas.ovpworld.org
Hetero_AlteraCycloneV_HPS_AlteraCycloneIII_3c120	imperas.ovpworld.org
Hetero_ArmIntegratorCP_XilinxMicroBlaze	imperas.ovpworld.org
Hetero_ArmVersatileExpress_MipsMalta	imperas.ovpworld.org
Hetero_ArmVersatileExpress_XilinxMicroBlaze	imperas.ovpworld.org
Quad_ArmVersatileExpress-CA15	imperas.ovpworld.org
RiscvRV32FreeRTOS	imperas.ovpworld.org
MipsMalta	mips.ovpworld.org
iMX6S	nxp.ovpworld.org
RenesasUPD70F3441	renesas.ovpworld.org
ghs-multi	renesas.ovpworld.org
virtio	riscv.ovpworld.org
FaultInjection	safepower.ovpworld.org
PublicDemonstrator	safepower.ovpworld.org
Zynq_PL_DualMicroblaze	safepower.ovpworld.org
Zynq_PL_NoC	safepower.ovpworld.org
Zynq_PL_NoC_node	safepower.ovpworld.org
Zynq_PL_NostrumNoC	safepower.ovpworld.org
Zynq_PL_NostrumNoC_node	safepower.ovpworld.org
Zynq_PL_RO	safepower.ovpworld.org
Zynq_PL_SingleMicroblaze	safepower.ovpworld.org
Zynq_PL_TTELNoC	safepower.ovpworld.org
Zynq_PL_TTELNoC_node	safepower.ovpworld.org
Zynq_PL_TTELNoC_processing_node_public_demonstrator	safepower.ovpworld.org
Zynq_PL_TTELNoC_public_demonstrator	safepower.ovpworld.org
Zynq_PL_TTELNoC_sensor_actor_node_public_demonstrator	safepower.ovpworld.org
FU540	sifive.ovpworld.org
S51CC	sifive.ovpworld.org
coreip-s51-arty	sifive.ovpworld.org
coreip-s51-rtl	sifive.ovpworld.org
dualFifo	vendor.com
XilinxML505	xilinx.ovpworld.org
Zynq	xilinx.ovpworld.org
Zynq_PL_Default	xilinx.ovpworld.org
Zynq_PS	xilinx.ovpworld.org
zc702	xilinx.ovpworld.org
zc706	xilinx.ovpworld.org

Table 14. Imperas / OVP Bare Metal Virtual Platforms (22 available)

Name	Vendor
BareMetalNios_IISingle	altera.ovpworld.org
BareMetalArcSingle	arc.ovpworld.org
BareMetalArm7Single	arm.ovpworld.org
BareMetalArmCortexADual	arm.ovpworld.org
BareMetalArmCortexASingle	arm.ovpworld.org
BareMetalArmCortexASingleAngelTrap	arm.ovpworld.org
BareMetalArmCortexMSingle	arm.ovpworld.org

Copyright (c) 2021 Imperas Software Limited   www.imperas.com

OVP License. Release 20211118.0   Page 17 of 18

ArmCortexMFreeRTOS	imperas.ovpworld.org
ArmCortexMuCOS-II	imperas.ovpworld.org
BareMetalArmx1Mips32x3	imperas.ovpworld.org
Or1kUclinux	imperas.ovpworld.org
BareMetalM14KSingle	mips.ovpworld.org
BareMetalMips32Dual	mips.ovpworld.org
BareMetalMips32Single	mips.ovpworld.org
BareMetalMips64Single	mips.ovpworld.org
BareMetalMipsDual	mips.ovpworld.org
BareMetalMipsSingle	mips.ovpworld.org
BareMetalOr1kSingle	ovpworld.org
BareMetalM16cSingle	posedgesoft.ovpworld.org
BareMetalPowerPc32Single	power.ovpworld.org
BareMetalV850Single	renesas.ovpworld.org
ghs-multi	renesas.ovpworld.org

#

www.imperas.com