Imperas Guide to using Virtual Platforms

Platform / Module Specific Information for nxp.ovpworld.org / iMX6S

Imperas Software Limited

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

Author	Imperas Software Limited
Version	20211118.0
Filename	Imperas_Platform_User_Guide_iMX6S.pdf
Created	31 December 2021
Status	OVP Standard Release

Copyright Notice

Right to Copy Documentation

The license agreement with Imperas permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any.

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the readers responsibility to determine the applicable regulations and to comply with them.

Disclaimer

IMPERAS SOFTWARE LIMITED, AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Model Release Status

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

Table Of Contents

1.0 Platform / Module: iMX6S

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 NXP i.MX 6 Solo application processor

The i.MX6S integrates a single ARM Cortex-A9 MPCore application processor, memories and peripherals.

1.4 Limitations

Some of the peripherals are register-only, non-functional models. See the individual peripheral model documentation for details.

1.5 Reference

i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf)

From: https://www.nxp.com/products/microcontrollers-and-processors/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-6-processors/i.mx-6solo-processors-single-core-multimedia-3d-graphics-arm-cortex-a9-core:i.MX6S

1.6 Location

The iMX6S virtual platform / module is located in an Imperas/OVP installation at the VLNV:
nxp.ovpworld.org / module / iMX6S / 1.0.

1.7 Module Simulation Attributes

Table 1. Module Simulation Attributes

Attribute	Value	Description
stoponctrlc	stoponctrlc	Stop on control-C

2.0 Processor [arm.ovpworld.org/processor/arm/1.0] instance: cpu

2.1 Processor model type: 'arm' variant 'Cortex-A9MPx1' definition

Imperas OVP processor models support multiple variants and details of the variants implemented in this model can be found in:
- the Imperas installation located at ImperasLib/source/arm.ovpworld.org/processor/arm/1.0/doc
- the OVP website: OVP_Model_Specific_Information_arm_Cortex-A9MPx1.pdf

2.1.1 Description

ARM Processor Model

2.1.2 Licensing

Usage of binary model under license governing simulator usage.
Note that for models of ARM CPUs the license includes the following terms:
Licensee is granted a non-exclusive, worldwide, non-transferable, revocable licence to:
If no source is being provided to the Licensee: use and copy only (no modifications rights are granted) the model for the sole purpose of designing, developing, analyzing, debugging, testing, verifying, validating and optimizing software which: (a) (i) is for ARM based systems; and (ii) does not incorporate the ARM Models or any part thereof; and (b) such ARM Models may not be used to emulate an ARM based system to run application software in a production or live environment.
If source code is being provided to the Licensee: use, copy and modify the model for the sole purpose of designing, developing, analyzing, debugging, testing, verifying, validating and optimizing software which: (a) (i) is for ARM based systems; and (ii) does not incorporate the ARM Models or any part thereof; and (b) such ARM Models may not be used to emulate an ARM based system to run application software in a production or live environment.
In the case of any Licensee who is either or both an academic or educational institution the purposes shall be limited to internal use.
Except to the extent that such activity is permitted by applicable law, Licensee shall not reverse engineer, decompile, or disassemble this model. If this model was provided to Licensee in Europe, Licensee shall not reverse engineer, decompile or disassemble the Model for the purposes of error correction.
The License agreement does not entitle Licensee to manufacture in silicon any product based on this model.
The License agreement does not entitle Licensee to use this model for evaluating the validity of any ARM patent.
Source of model available under separate Imperas Software License Agreement.

2.1.3 Limitations

Instruction pipelines are not modeled in any way. All instructions are assumed to complete immediately. This means that instruction barrier instructions (e.g. ISB, CP15ISB) are treated as NOPs, with the exception of any undefined instruction behavior, which is modeled. The model does not implement speculative fetch behavior. The branch cache is not modeled.
Caches and write buffers are not modeled in any way. All loads, fetches and stores complete immediately and in order, and are fully synchronous (as if the memory was of Strongly Ordered or Device-nGnRnE type). Data barrier instructions (e.g. DSB, CP15DSB) are treated as NOPs, with the exception of any undefined instruction behavior, which is modeled. Cache manipulation instructions are implemented as NOPs, with the exception of any undefined instruction behavior, which is modeled.
Real-world timing effects are not modeled: all instructions are assumed to complete in a single cycle.

Performance Monitors are implemented as a register interface only except for the cycle counter, which is implemented assuming one instruction per cycle.
TLBs are architecturally-accurate but not device accurate. This means that all TLB maintenance and address translation operations are fully implemented but the cache is larger than in the real device.

2.1.4 Verification

Models have been extensively tested by Imperas. ARM Cortex-A models have been successfully used by customers to simulate SMP Linux, Ubuntu Desktop, VxWorks and ThreadX on Xilinx Zynq virtual platforms.

2.1.5 Core Features

Thumb-2 instructions are supported.
Trivial Jazelle extension is implemented.

2.1.6 Memory System

Security extensions are implemented (also known as TrustZone). Non-secure accesses can be made visible externally by connecting the processor to a 41-bit physical bus, in which case bits 39..0 give the true physical address and bit 40 is the NS bit.
VMSA secure and non-secure address translation is implemented.
TLB behavior is controlled by parameter ASIDCacheSize. If this parameter is 0, then an unlimited number of TLB entries will be maintained concurrently. If this parameter is non-zero, then only TLB entries for up to ASIDCacheSize different ASIDs will be maintained concurrently initially; as new ASIDs are used, TLB entries for less-recently used ASIDs are deleted, which improves model performance in some cases (especially when 16-bit ASIDs are in use). If the model detects that the TLB entry cache is too small (entry ejections are very frequent), it will increase the cache size automatically. In this variant, ASIDCacheSize is 8

2.1.7 Advanced SIMD and Floating-Point Features

SIMD and VFP instructions are implemented.
The model implements trapped exceptions if FPTrap is set to 1 in MVFR0 (for AArch32) or MVFR0_EL1 (for AArch64). When floating point exception traps are taken, cumulative exception flags are not updated (in other words, cumulative flag state is always the same as prior to instruction execution, even for SIMD instructions). When multiple enabled exceptions are raised by a single floating point operation, the exception reported is the one in least-significant bit position in FPSCR (for AArch32) or FPCR (for AArch64). When multiple enabled exceptions are raised by different SIMD element computations, the exception reported is selected from the lowest-index-number SIMD operation. Contact Imperas if requirements for exception reporting differ from these.
Trapped exceptions not are implemented in this variant (FPTrap=0)

2.1.8 Generic Interrupt Controller

GIC block is implemented (GICv1, including security extensions). Accesses to GIC registers can be viewed externally by connecting to the 32-bit GICRegisters bus port. Secure register accesses are at offset 0x0 on this bus; for example, a secure access to GIC register ICDDCR can be observed by monitoring address

0x00001000. Non-secure accesses are at offset 0x80000000 on this bus; for example, a non-secure access to GIC register ICDDCR can be observed by monitoring address 0x80001000

2.1.9 Debug Mask

It is possible to enable model debug features in various categories. This can be done statically using the "override_debugMask" parameter, or dynamically using the "debugflags" command. Enabled debug features are specified using a bitmask value, as follows:
Value 0x004: enable debugging of MMU/MPU mappings.
Value 0x020: enable debugging of reads and writes of GIC block registers.
Value 0x040: enable debugging of exception routing via the GIC model component.
Value 0x080: enable debugging of all system register accesses.
Value 0x100: enable debugging of all traps of system register accesses.
Value 0x200: enable verbose debugging of other miscellaneous behavior (for example, the reason why a particular instruction is undefined).
Value 0x400: enable debugging of Performance Monitor timers
Value 0x800: enable dynamic validation of TLB entries against in-memory page table contents (finds some classes of error where page table entries are updated without a subsequent flush of affected TLB entries).
All other bits in the debug bitmask are reserved and must not be set to non-zero values.

2.1.10 AArch32 Unpredictable Behavior

Many AArch32 instruction behaviors are described in the ARM ARM as CONSTRAINED UNPREDICTABLE. This section describes how such situations are handled by this model.

2.1.11 Equal Target Registers

Some instructions allow the specification of two target registers (for example, double-width SMULL, or some VMOV variants), and such instructions are CONSTRAINED UNPREDICTABLE if the same target register is specified in both positions. In this model, such instructions are treated as UNDEFINED.

2.1.12 Floating Point Load/Store Multiple Lists

Instructions that load or store a list of floating point registers (e.g. VSTM, VLDM, VPUSH, VPOP) are CONSTRAINED UNPREDICTABLE if either the uppermost register in the specified range is greater than 32 or (for 64-bit registers) if more than 16 registers are specified. In this model, such instructions are treated as UNDEFINED.

2.1.13 Floating Point VLD[2-4]/VST[2-4] Range Overflow

Instructions that load or store a fixed number of floating point registers (e.g. VST2, VLD2) are CONSTRAINED UNPREDICTABLE if the upper register bound exceeds the number of implemented floating point registers. In this model, these instructions load and store using modulo 32 indexing (consistent with AArch64 instructions with similar behavior).

2.1.14 If-Then (IT) Block Constraints

Where the behavior of an instruction in an if-then (IT) block is described as CONSTRAINED UNPREDICTABLE, this model treats that instruction as UNDEFINED.

2.1.15 Use of R13

In architecture variants before ARMv8, use of R13 was described as CONSTRAINED UNPREDICTABLE in many circumstances. From ARMv8, most of these situations are no longer considered unpredictable. This model allows R13 to be used like any other GPR, consistent with the ARMv8 specification.

2.1.16 Use of R15

Use of R15 is described as CONSTRAINED UNPREDICTABLE in many circumstances. This model allows such use to be configured using the parameter "unpredictableR15" as follows:
Value "undefined": any reference to R15 in such a situation is treated as UNDEFINED;
Value "nop": any reference to R15 in such a situation causes the instruction to be treated as a NOP;
Value "raz_wi": any reference to R15 in such a situation causes the instruction to be treated as a RAZ/WI (that is, R15 is read as zero and write-ignored);
Value "execute": any reference to R15 in such a situation is executed using the current value of R15 on read, and writes to R15 are allowed (but are not interworking).
Value "assert": any reference to R15 in such a situation causes the simulation to halt with an assertion message (allowing any such unpredictable uses to be easily identified).
In this variant, the default value of "unpredictableR15" is "undefined".

2.1.17 Unpredictable Instructions in Some Modes

Some instructions are described as CONSTRAINED UNPREDICTABLE in some modes only (for example, MSR accessing SPSR is CONSTRAINED UNPREDICTABLE in User and System modes). This model allows such use to be configured using the parameter "unpredictableModal", which can have values "undefined" or "nop". See the previous section for more information about the meaning of these values.
In this variant, the default value of "unpredictableModal" is "nop".

2.1.18 Integration Support

This model implements a number of non-architectural pseudo-registers and other features to facilitate integration.

2.1.19 Memory Transaction Query

Two registers are intended for use within memory callback functions to provide additional information about the current memory access. Register transactPL indicates the processor execution level of the current access (0-3). Note that for load/store translate instructions (e.g. LDRT, STRT) the reported execution level will be 0, indicating an EL0 access. Register transactAT indicates the type of memory access: 0 for a normal read or write; and 1 for a physical access resulting from a page table walk.

2.1.20 Page Table Walk Query

A banked set of registers provides information about the most recently completed page table walk. There are up to six banks of registers: bank 0 is for stage 1 walks, bank 1 is for stage 2 walks, and banks 2-5 are for stage 2 walks initiated by stage 1 level 0-3 entry lookups, respectively. Banks 1-5 are present only for processors with virtualization extensions. The currently active bank can be set using register PTWBankSelect. Register PTWBankValid is a bitmask indicating which banks contain valid data: for example, the value 0xb indicates that banks 0, 1 and 3 contain valid data.

Within each bank, there are registers that record addresses and values read during that page table walk. Register PTWBase records the table base address, register PTWInput contains the input address that starts a walk, register PTWOutput contains the result address and register PTWPgSize contains the page size (PTWOutput and PTWPgSize are valid only if the page table walk completes). Registers PTWAddressL0-PTWAddressL3 record the addresses of level 0 to level 3 entries read, respectively. Register PTWAddressValid is a bitmask indicating which address registers contain valid data: bits 0-3 indicate PTWAddressL0-PTWAddressL3, respectively, bit 4 indicates PTWBase, bit 5 indicates PTWInput, bit 6 indicates both PTWOutput and PTWPgSize. For example, the value 0x73 indicates that PTWBase, PTWInput, PTWOutput, PTWPgSize and PTWAddressL0-L1 are valid but PTWAddressL2-L3 are not. Register PTWAddressNS is a bitmask indicating whether an address is in non-secure memory: bits 0-3 indicate PTWAddressL0-PTWAddressL3, respectively, bit 4 indicates PTWBase, bit 6 indicates PTWOutput (PTWInput is a VA and thus has no secure/non-secure info). Registers PTWValueL0-PTWValueL3 contain page table entry values read at level 0 to level 3. Register PTWValueValid is a bitmask indicating which value registers contain valid data: bits 0-3 indicate PTWValueL0-PTWValueL3, respectively.

2.1.21 Artifact Page Table Walks

Registers are also available to enable a simulation environment to initiate an artifact page table walk (for example, to determine the ultimate PA corresponding to a given VA). Register PTWI_EL1S initiates a secure EL1 table walk for a fetch. Register PTWD_EL1S initiates a secure EL1 table walk for a load or store (note that current ARM processors have unified TLBs, so these registers are synonymous). Registers PTW[ID]_EL1NS initiate walks for non-secure EL1 accesses. Registers PTW[ID]_EL2 initiate EL2 walks. Registers PTW[ID]_S2 initiate stage 2 walks. Registers PTW[ID]_EL3 initiate AArch64 EL3 walks. Finally, registers PTW[ID]_current initiate current-mode walks (useful in a memory callback context). Each walk fills the query registers described above.

2.1.22 MMU and Page Table Walk Events

Two events are available that allow a simulation environment to be notified on MMU and page table walk actions. Event mmuEnable triggers when any MMU is enabled or disabled. Event pageTableWalk triggers on completion of any page table walk (including artifact walks).

2.1.23 Artifact Address Translations

A simulation environment can trigger an artifact address translation operation by writing to the architectural address translation registers (e.g. ATS1CPR). The results of such translations are written to an integration support register artifactPAR, instead of the architectural PAR register. This means that such artifact writes will not perturb architectural state.

2.1.24 TLB Invalidation

A simulation environment can cause TLB state for one or more address translation regimes in the processor to be flushed by writing to the artifact register ResetTLBs. The argument is a bitmask value, in which non-zero bits select the TLBs to be flushed, as follows:
Bit 0: EL0/EL1 stage 1 secure TLB
Bit 1: EL0/EL1 stage 1 non-secure TLB

2.1.25 Halt Reason Introspection

An artifact register HaltReason can be read to determine the reason or reasons that a processor is halted. This register is a bitfield, with the following encoding: bit 0 indicates the processor has executed a wait-for-event (WFE) instruction; bit 1 indicates the processor has executed a wait-for-interrupt (WFI) instruction; and bit 2 indicates the processor is held in reset.

2.1.26 System Register Access Monitor

If parameter "enableSystemMonitorBus" is True, an artifact 32-bit bus "SystemMonitor" is enabled for each PE. Every system register read or write by that PE is then visible as a read or write on this artifact bus, and can therefore be monitored using callbacks installed in the client environment (use opBusReadMonitorAdd/opBusWriteMonitorAdd or icmAddBusReadCallback/icmAddBusWriteCallback, depending on the client API). The format of the address on the bus is as follows:
bits 31:26 - zero
bit 25 - 1 if AArch64 access, 0 if AArch32 access
bit 24 - 1 if non-secure access, 0 if secure access
bits 23:20 - CRm value
bits 19:16 - CRn value
bits 15:12 - op2 value
bits 11:8 - op1 value
bits 7:4 - op0 value (AArch64) or coprocessor number (AArch32)
bits 3:0 - zero
As an example, to view non-secure writes to writes to CNTFRQ_EL0 in AArch64 state, install a write monitor on address range 0x020e0330:0x020e0333.

2.1.27 System Register Implementation

If parameter "enableSystemBus" is True, an artifact 32-bit bus "System" is enabled for each PE. Slave callbacks installed on this bus can be used to implement modified system register behavior (use opBusSlaveNew or icmMapExternalMemory, depending on the client API). The format of the address on the bus is the same as for the system monitor bus, described above.

2.2 Instance Parameters

Several parameters can be specified when a processor is instanced in a platform. For this processor instance 'cpu' it has been instanced with the following parameters:

Table 2. Processor Instance 'cpu' Parameters (Configurations)

Parameter	Value	Description
endian	little	Select processor endian (big or little)
simulateexceptions	simulateexceptions	Causes the processor simulate exceptions instead of halting
mips	800	The nominal MIPS for the processor

Table 3. Processor Instance 'cpu' Parameters (Attributes)

Parameter Name	Value	Type
variant	Cortex-A9MPx1	enum

compatibility	ISA	enum
UAL	1	boolean
override_CBAR	0x00a00000	Uns32
override_MIDR	0x411fc090	Uns32
override_GICD_TYPER_ITLines	4	uns32

2.3 Memory Map for processor 'cpu' bus: 'pBus'

Processor instance 'cpu' is connected to bus 'pBus' using master port 'INSTRUCTION'.
Processor instance 'cpu' is connected to bus 'pBus' using master port 'DATA'.

Table 4. Memory Map ('cpu' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	AIPS1_Cfg	trap
remappable	remappable	AIPS2_Cfg	trap
remappable	remappable	APBH_DMA	trap
remappable	remappable	ASRC	trap
remappable	remappable	AUDMUX	trap
remappable	remappable	CAAM	trap
remappable	remappable	CAN1	trap
remappable	remappable	CAN2	trap
remappable	remappable	CSU	trap
remappable	remappable	DCIC1	trap
remappable	remappable	DCIC2	trap
remappable	remappable	EIM	trap
remappable	remappable	ENET	trap
remappable	remappable	EPDC	trap
remappable	remappable	EPIT1	trap
remappable	remappable	EPIT2	trap
remappable	remappable	ESAI	trap
remappable	remappable	GPU2D	trap
remappable	remappable	GPU3D	trap
remappable	remappable	HDMI	trap
remappable	remappable	I2C1	trap
remappable	remappable	I2C2	trap
remappable	remappable	I2C3	trap
remappable	remappable	I2C4	trap
remappable	remappable	IOMUXC	trap
remappable	remappable	IPU	trap
remappable	remappable	KPP	trap
remappable	remappable	MIPI_CSI	trap
remappable	remappable	MIPI_DSI	trap
remappable	remappable	MLB150	trap
remappable	remappable	MMDCp1	trap
remappable	remappable	OCOTP_CTRL	trap
remappable	remappable	PGC_ARM	trap
remappable	remappable	PGC_PU	trap
remappable	remappable	PWM1	trap
remappable	remappable	PWM2	trap

remappable	remappable	PWM3	trap
remappable	remappable	PWM4	trap
remappable	remappable	PXP	trap
remappable	remappable	ROMCP	trap
remappable	remappable	SDMA	trap
remappable	remappable	SNVS_HP	trap
remappable	remappable	SPBA	trap
remappable	remappable	SPDIF	trap
remappable	remappable	SSI1	trap
remappable	remappable	SSI2	trap
remappable	remappable	SSI3	trap
remappable	remappable	TZASC1	trap
remappable	remappable	TZASC2	trap
remappable	remappable	UART3	trap
remappable	remappable	UART4	trap
remappable	remappable	UART5	trap
remappable	remappable	USBOH3_PL301	trap
remappable	remappable	USBOH3_USB	trap
remappable	remappable	USBPHY1	trap
remappable	remappable	USBPHY2	trap
remappable	remappable	VDOA	trap
remappable	remappable	eCSPI1	trap
remappable	remappable	eCSPI2	trap
remappable	remappable	eCSPI3	trap
remappable	remappable	eCSPI4	trap
remappable	remappable	uSDHC1	trap
remappable	remappable	uSDHC2	trap
remappable	remappable	uSDHC3	trap
remappable	remappable	uSDHC4	trap
0x900000	0x91FFFF	OCRAM	ram
0xA02000	0xA02FFF	PL310	L2CachePL310
0x2020000	0x2023FFF	UART1	iMX6_UART
0x2098000	0x2098FFF	GPT	iMX6_GPT
0x209C000	0x209FFFF	GPIO1	iMX6_GPIO
0x20A0000	0x20A3FFF	GPIO2	iMX6_GPIO
0x20A4000	0x20A7FFF	GPIO3	iMX6_GPIO
0x20A8000	0x20ABFFF	GPIO4	iMX6_GPIO
0x20AC000	0x20AFFFF	GPIO5	iMX6_GPIO
0x20B0000	0x20B3FFF	GPIO6	iMX6_GPIO
0x20B4000	0x20B7FFF	GPIO7	iMX6_GPIO
0x20BC000	0x20BFFFF	WDOG1	iMX6_WDOG
0x20C0000	0x20C3FFF	WDOG2	iMX6_WDOG
0x20C4000	0x20C4FFF	CCM	iMX6_CCM
0x20C8000	0x20C8FFF	ANALOG	iMX6_Analog
0x20D8000	0x20DBFFF	SRC	iMX6_SRC
0x20DC000	0x20DC25F	GPC	iMX6_GPC
0x21B0000	0x21B3FFF	MMDC	iMX6_MMDC
0x21E8000	0x21EBFFF	UART2	iMX6_UART

0x21FF000	0x21FF1FF	VBD0	VirtioBlkMMIO
0x8000000	0xFFFFFFF	EIM-RAM	ram
0x10000000	0x2FFFFFFF	DRAM	ram

2.4 Net Connections to processor: 'cpu'

Table 5. Processor Net Connections ('cpu')

Net Port	Net	Instance	Component
SPI32	IOMUXC_spi	GPC	iMX6_GPC
SPI33	DAP_spi	GPC	iMX6_GPC
SPI34	SDMA_spi	GPC	iMX6_GPC
SPI35	VPU_spi	GPC	iMX6_GPC
SPI36	SNVS_spi	GPC	iMX6_GPC
SPI37	IPU_spi	GPC	iMX6_GPC
SPI38	IPU1_spi	GPC	iMX6_GPC
SPI41	GPU3D_spi	GPC	iMX6_GPC
SPI42	R2D_spi	GPC	iMX6_GPC
SPI44	VPU_jpeg_spi	GPC	iMX6_GPC
SPI45	APBH_spi	GPC	iMX6_GPC
SPI46	EIM_spi	GPC	iMX6_GPC
SPI47	BCH_spi	GPC	iMX6_GPC
SPI48	GPMI_spi	GPC	iMX6_GPC
SPI49	DTCP_spi	GPC	iMX6_GPC
SPI50	VDOA_spi	GPC	iMX6_GPC
SPI51	SNVS_consolidated_spi	GPC	iMX6_GPC
SPI52	SNVS_security_spi	GPC	iMX6_GPC
SPI53	CSU_spi	GPC	iMX6_GPC
SPI58	UART1_spi	GPC	iMX6_GPC
SPI58	UART1_spi	UART1	iMX6_UART
SPI59	UART2_spi	GPC	iMX6_GPC
SPI59	UART2_spi	UART2	iMX6_UART
SPI60	UART3_spi	GPC	iMX6_GPC
SPI61	UART4_spi	GPC	iMX6_GPC
SPI62	UART5_spi	GPC	iMX6_GPC
SPI63	eCSPI1_spi	GPC	iMX6_GPC
SPI64	eCSPI2_spi	GPC	iMX6_GPC
SPI65	eCSPI3_spi	GPC	iMX6_GPC
SPI66	eCSPI4_spi	GPC	iMX6_GPC
SPI67	I2C4_spi	GPC	iMX6_GPC
SPI68	I2C1_spi	GPC	iMX6_GPC
SPI69	I2C2_spi	GPC	iMX6_GPC
SPI70	I2C3_spi	GPC	iMX6_GPC
SPI72	USBHost1_spi	GPC	iMX6_GPC
SPI73	USBHost2_spi	GPC	iMX6_GPC
SPI74	USBHost3_spi	GPC	iMX6_GPC
SPI75	USBOTG_spi	GPC	iMX6_GPC
SPI76	USB_PHYUTMI0_spi	GPC	iMX6_GPC

SPI77	USB_PHYUTMI1_spi	GPC	iMX6_GPC
SPI78	SSI1_spi	GPC	iMX6_GPC
SPI79	SSI2_spi	GPC	iMX6_GPC
SPI80	SSI3_spi	GPC	iMX6_GPC
SPI81	TMTS_spi	GPC	iMX6_GPC
SPI82	ASRC_spi	GPC	iMX6_GPC
SPI83	ESAI_spi	GPC	iMX6_GPC
SPI84	SPDIF_spi	GPC	iMX6_GPC
SPI85	MLB150_err_spi	GPC	iMX6_GPC
SPI86	PMU_analog_spi	GPC	iMX6_GPC
SPI87	GPT_spi	GPC	iMX6_GPC
SPI87	GPT_spi	GPT	iMX6_GPT
SPI88	EPIT1_spi	GPC	iMX6_GPC
SPI89	EPIT2_spi	GPC	iMX6_GPC
SPI90	GPIO1_int7_spi	GPC	iMX6_GPC
SPI91	GPIO1_int6_spi	GPC	iMX6_GPC
SPI92	GPIO1_int5_spi	GPC	iMX6_GPC
SPI93	GPIO1_int4_spi	GPC	iMX6_GPC
SPI94	GPIO1_int3_spi	GPC	iMX6_GPC
SPI95	GPIO1_int2_spi	GPC	iMX6_GPC
SPI96	GPIO1_int1_spi	GPC	iMX6_GPC
SPI97	GPIO1_int0_spi	GPC	iMX6_GPC
SPI98	GPIO1_s0_15_spi	GPC	iMX6_GPC
SPI99	GPIO1_s16_31_spi	GPC	iMX6_GPC
SPI100	GPIO2_s0_15_spi	GPC	iMX6_GPC
SPI101	GPIO2_s16_31_spi	GPC	iMX6_GPC
SPI102	GPIO3_s0_15_spi	GPC	iMX6_GPC
SPI103	GPIO3_s16_31_spi	GPC	iMX6_GPC
SPI104	GPIO4_s0_15_spi	GPC	iMX6_GPC
SPI105	GPIO4_s16_31_spi	GPC	iMX6_GPC
SPI106	GPIO5_s0_15_spi	GPC	iMX6_GPC
SPI107	GPIO5_s16_31_spi	GPC	iMX6_GPC
SPI108	GPIO6_s0_15_spi	GPC	iMX6_GPC
SPI109	GPIO6_s16_31_spi	GPC	iMX6_GPC
SPI110	GPIO7_s0_15_spi	GPC	iMX6_GPC
SPI111	GPIO7_s16_31_spi	GPC	iMX6_GPC
SPI112	WDOG1_spi	GPC	iMX6_GPC
SPI112	WDOG1_spi	WDOG1	iMX6_WDOG
SPI113	WDOG2_spi	GPC	iMX6_GPC
SPI113	WDOG2_spi	WDOG2	iMX6_WDOG
SPI114	KPP_spi	GPC	iMX6_GPC
SPI115	PWM1_spi	GPC	iMX6_GPC
SPI116	PWM2_spi	GPC	iMX6_GPC
SPI117	PWM3_spi	GPC	iMX6_GPC
SPI118	PWM4_spi	GPC	iMX6_GPC
SPI119	CCM1_spi	GPC	iMX6_GPC
SPI120	CCM2_spi	GPC	iMX6_GPC
SPI121	GPC1_spi	GPC	iMX6_GPC

SPI123	SRC_spi	GPC	iMX6_GPC
SPI124	CPU_L2_spi	GPC	iMX6_GPC
SPI125	CPU_ParityCheckError_spi	GPC	iMX6_GPC
SPI126	CPU_Performance_Unit_spi	GPC	iMX6_GPC
SPI127	CPU_CTI_spi	GPC	iMX6_GPC
SPI128	SRC_wdog_spi	GPC	iMX6_GPC
SPI129	EPDC_spi	GPC	iMX6_GPC
SPI130	PXP_spi	GPC	iMX6_GPC
SPI132	MIPI_CSI1_spi	GPC	iMX6_GPC
SPI133	MIPI_CSI2_spi	GPC	iMX6_GPC
SPI134	MIPI_DSI_spi	GPC	iMX6_GPC
SPI135	MIPI_HSI_spi	GPC	iMX6_GPC
SPI136	SJC_spi	GPC	iMX6_GPC
SPI137	CAAM0_spi	GPC	iMX6_GPC
SPI138	CAAM1_spi	GPC	iMX6_GPC
SPI140	TZASC1_spi	GPC	iMX6_GPC
SPI141	TZASC2_spi	GPC	iMX6_GPC
SPI142	FLEXCAN1_spi	GPC	iMX6_GPC
SPI143	FLEXCAN2_spi	GPC	iMX6_GPC
SPI147	HDMI_spi	GPC	iMX6_GPC
SPI148	HDMICEC_spi	GPC	iMX6_GPC
SPI149	MLB150_irq0_31_spi	GPC	iMX6_GPC
SPI150	ENET_spi	GPC	iMX6_GPC
SPI151	ENET_timer_spi	GPC	iMX6_GPC
SPI152	PCIe1_spi	GPC	iMX6_GPC
SPI153	PCIe2_spi	GPC	iMX6_GPC
SPI154	PCIe3_spi	GPC	iMX6_GPC
SPI155	PCIe4_spi	GPC	iMX6_GPC
SPI156	DCIC1_spi	GPC	iMX6_GPC
SPI157	DCIC2_spi	GPC	iMX6_GPC
SPI158	MLB150_irq32_63_spi	GPC	iMX6_GPC
SPI159	PMU_digital_spi	GPC	iMX6_GPC
reset_CPU0	reset_A9_CPU0	SRC	iMX6_SRC
SPI39	VBD0_spi	VBD0	VirtioBlkMMIO

3.0 Peripheral Instances

3.1 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPC/1.0] instance: GPC

3.1.1 Description

NXP i.MX6 GPC General Power Controller

3.1.2 Licensing

Open Source Apache 2.0

3.1.3 Limitations

This is a register only model

3.1.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.2 Peripheral [arm.ovpworld.org/peripheral/L2CachePL310/1.0] instance: PL310

3.2.1 Description
ARM PL310 L2 Cache Control Registers

3.2.2 Licensing
Open Source Apache 2.0

3.2.3 Limitations
Programmers View, register model only. Does NOT model functionality, just provides registers to allow code to run.

3.2.4 Reference
ARM PrimeCell Level 2 Cache Controller (PL310) Technical Reference Manual (ARM DDI 0246)

There are no configuration options set for this peripheral instance.

3.3 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPT/1.0] instance: GPT

3.3.1 Description
NXP i.MX6 GPT General Purpose Timer

3.3.2 Licensing
Open Source Apache 2.0

3.3.3 Limitations
Resolution of this timer is limited to the simulation time slice (aka quantum) size

3.3.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.4 Peripheral [nxp.ovpworld.org/peripheral/iMX6_UART/1.0] instance: UART1

3.4.1 Description
iMX6 UART

3.4.2 Licensing
Open Source Apache 2.0

3.4.3 Limitations
This is an incomplete model of the UART.
It has basic functionality to support the iMX6 platform, Rx and Tx of data only.
There is no modeling of physical aspects of the UART, such as baud rates etc.

3.4.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

Table 6. Configuration options (attributes) set for instance 'UART1'

Attribute	Value	Type	Expression
outfile	UART1.log	string	
finishOnDisconnect	1	boolean	
console	1	boolean	

3.5 Peripheral [nxp.ovpworld.org/peripheral/iMX6_UART/1.0] instance: UART2

3.5.1 Description
iMX6 UART

3.5.2 Licensing
Open Source Apache 2.0

3.5.3 Limitations
This is an incomplete model of the UART.
It has basic functionality to support the iMX6 platform, Rx and Tx of data only.
There is no modeling of physical aspects of the UART, such as baud rates etc.

3.5.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

Table 7. Configuration options (attributes) set for instance 'UART2'

Attribute	Value	Type	Expression
outfile	UART2.log	string	
finishOnDisconnect	1	boolean	
console	0	boolean	

3.6 Peripheral [nxp.ovpworld.org/peripheral/iMX6_CCM/1.0] instance: CCM

3.6.1 Description
NXP i.MX6 CCM Clock Controller Module

3.6.2 Licensing
Open Source Apache 2.0

3.6.3 Limitations
This is a register only interface model. No functionality is implemented. The reset value for the CCM_CSCDR2 epdc_pix_clk_sel field (bits 11:9) has been modified from the documented value. The documented value (0x5) is reserved and causes a Linux Kernel Panic

3.6.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.7 Peripheral [nxp.ovpworld.org/peripheral/iMX6_Analog/1.0] instance: ANALOG

3.7.1 Description
NXP i.MX6 ANALOG: (PLLs, PFDs, Regulators, LDOs, Temp Sensor) Registers

3.7.2 Licensing
Open Source Apache 2.0

3.7.3 Limitations
This is a register only interface model. No functionality is implemented. The reset values for registers have been modified from those specified in the documentation to set the lock bit (bit 31) on PLL registers.

3.7.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.8 Peripheral [nxp.ovpworld.org/peripheral/iMX6_SRC/1.0] instance: SRC

3.8.1 Description
NXP i.MX6 System Reset Control (SRC)

3.8.2 Licensing
Open Source Apache 2.0

3.8.3 Limitations
Implements the power on reset and core1 enable requirements.

3.8.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.9 Peripheral [nxp.ovpworld.org/peripheral/iMX6_MMDC/1.0] instance: MMDC

3.9.1 Description
NXP i.MX6 MMDC

3.9.2 Licensing
Open Source Apache 2.0

3.9.3 Limitations
This is a register only model with acknowledgement of auto power saving

3.9.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.10 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPIO/1.0] instance: GPIO1

3.10.1 Description
NXP i.MX6 GPIO

3.10.2 Licensing
Open Source Apache 2.0

3.10.3 Limitations
No behaviour is implemented.

3.10.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.11 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPIO/1.0] instance: GPIO2

3.11.1 Description
NXP i.MX6 GPIO

3.11.2 Licensing
Open Source Apache 2.0

3.11.3 Limitations
No behaviour is implemented.

3.11.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.12 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPIO/1.0] instance: GPIO3

3.12.1 Description
NXP i.MX6 GPIO

3.12.2 Licensing
Open Source Apache 2.0

3.12.3 Limitations
No behaviour is implemented.

3.12.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.13 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPIO/1.0] instance: GPIO4

3.13.1 Description
NXP i.MX6 GPIO

3.13.2 Licensing
Open Source Apache 2.0

3.13.3 Limitations
No behaviour is implemented.

3.13.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.14 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPIO/1.0] instance: GPIO5

3.14.1 Description

NXP i.MX6 GPIO

3.14.2 Licensing
Open Source Apache 2.0

3.14.3 Limitations
No behaviour is implemented.

3.14.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.15 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPIO/1.0] instance: GPIO6

3.15.1 Description
NXP i.MX6 GPIO

3.15.2 Licensing
Open Source Apache 2.0

3.15.3 Limitations
No behaviour is implemented.

3.15.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.16 Peripheral [nxp.ovpworld.org/peripheral/iMX6_GPIO/1.0] instance: GPIO7

3.16.1 Description
NXP i.MX6 GPIO

3.16.2 Licensing
Open Source Apache 2.0

3.16.3 Limitations
No behaviour is implemented.

3.16.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.17 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: uSDHC1

3.17.1 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.

3.17.2 Licensing

Open Source Apache 2.0

3.17.3 Limitations

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

3.17.4 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 8. Configuration options (attributes) set for instance 'uSDHC1'

Attribute	Value	Type	Expression
portAddress	0x02190000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.18 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: uSDHC2

3.18.1 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.

3.18.2 Licensing

Open Source Apache 2.0

3.18.3 Limitations

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

3.18.4 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 9. Configuration options (attributes) set for instance 'uSDHC2'

Attribute	Value	Type	Expression

portAddress	0x02194000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.19 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: uSDHC3

3.19.1 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.

3.19.2 Licensing

Open Source Apache 2.0

3.19.3 Limitations

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

3.19.4 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 'uSDHC3'

Attribute	Value	Type	Expression
portAddress	0x02198000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.20 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: uSDHC4

3.20.1 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.

3.20.2 Licensing

Open Source Apache 2.0

3.20.3 Limitations

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

3.20.4 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 11. Configuration options (attributes) set for instance 'uSDHC4'

Attribute	Value	Type	Expression
portAddress	0x0219C000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.21 Peripheral [nxp.ovpworld.org/peripheral/iMX6_WDOG/1.0] instance: WDOG1

3.21.1 Description
iMX6 WDOG

3.21.2 Licensing
Open Source Apache 2.0

3.21.3 Limitations
This is an incomplete model of the WDOG.
It has basic functionality to support the iMX6 platform.

3.21.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.22 Peripheral [nxp.ovpworld.org/peripheral/iMX6_WDOG/1.0] instance: WDOG2

3.22.1 Description
iMX6 WDOG

3.22.2 Licensing
Open Source Apache 2.0

3.22.3 Limitations
This is an incomplete model of the WDOG.
It has basic functionality to support the iMX6 platform.

3.22.4 Reference
i.MX 6Solo/6DualLite Applications Processor Reference Manual (IMX6SDLRM_Ref_Manual.pdf

There are no configuration options set for this peripheral instance.

3.23 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: EPDC

3.23.1 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.

3.23.2 Licensing
Open Source Apache 2.0

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

3.23.4 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 12. Configuration options (attributes) set for instance 'EPDC'

Attribute	Value	Type	Expression
portAddress	0x020F4000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.24 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: PXP

3.24.1 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.

3.24.2 Licensing
Open Source Apache 2.0

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

3.24.4 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 13. Configuration options (attributes) set for instance 'PXP'

Attribute	Value	Type	Expression
portAddress	0x020F0000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.25 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: SDMA

3.25.1 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.

3.25.2 Licensing
Open Source Apache 2.0

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

3.25.4 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 14. Configuration options (attributes) set for instance 'SDMA'

Attribute	Value	Type	Expression
portAddress	0x020EC000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.26 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: DCIC2

3.26.1 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.

3.26.2 Licensing
Open Source Apache 2.0

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

3.26.4 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 15. Configuration options (attributes) set for instance 'DCIC2'

Attribute	Value	Type	Expression
portAddress	0x020E8000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.27 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: DCIC1

3.27.1 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.

3.27.2 Licensing
Open Source Apache 2.0

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

3.27.4 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 16. Configuration options (attributes) set for instance 'DCIC1'

Attribute	Value	Type	Expression
portAddress	0x020E4000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.28 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: IOMUXC

3.28.1 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.

3.28.2 Licensing
Open Source Apache 2.0

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

3.28.4 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 17. Configuration options (attributes) set for instance 'IOMUXC'

Attribute	Value	Type	Expression
portAddress	0x020E0000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.29 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: PGC_ARM

3.29.1 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.

3.29.2 Licensing
Open Source Apache 2.0

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

3.29.4 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 18. Configuration options (attributes) set for instance 'PGC_ARM'

Attribute	Value	Type	Expression
portAddress	0x020DC2A0	uns32	
portSize	32	uns32	
cbEnable	True	boolean	

3.30 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: PGC_PU

3.30.1 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.

3.30.2 Licensing
Open Source Apache 2.0

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

3.30.4 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 19. Configuration options (attributes) set for instance 'PGC_PU'

Attribute	Value	Type	Expression
portAddress	0x020DC260	uns32	
portSize	32	uns32	
cbEnable	True	boolean	

3.31 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: EPIT2

3.31.1 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.

3.31.2 Licensing
Open Source Apache 2.0

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

3.31.4 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 20. Configuration options (attributes) set for instance 'EPIT2'

Attribute	Value	Type	Expression
portAddress	0x020D4000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.32 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: EPIT1

3.32.1 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.

3.32.2 Licensing
Open Source Apache 2.0

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

3.32.4 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 21. Configuration options (attributes) set for instance 'EPIT1'

Attribute	Value	Type	Expression
portAddress	0x020D0000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.33 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: SNVS_HP

3.33.1 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.

3.33.2 Licensing
Open Source Apache 2.0

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

3.33.4 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 22. Configuration options (attributes) set for instance 'SNVS_HP'

Attribute	Value	Type	Expression
portAddress	0x020CC000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.34 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: USBPHY2

3.34.1 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.

3.34.2 Licensing
Open Source Apache 2.0

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

3.34.4 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 23. Configuration options (attributes) set for instance 'USBPHY2'

Attribute	Value	Type	Expression
portAddress	0x020CA000	uns32	
portSize	4096	uns32	
cbEnable	True	boolean	

3.35 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: USBPHY1

3.35.1 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.

3.35.2 Licensing
Open Source Apache 2.0

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

3.35.4 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 24. Configuration options (attributes) set for instance 'USBPHY1'

Attribute	Value	Type	Expression
portAddress	0x020C9000	uns32	
portSize	4096	uns32	
cbEnable	True	boolean	

3.36 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: KPP

3.36.1 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.

3.36.2 Licensing
Open Source Apache 2.0

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

3.36.4 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 25. Configuration options (attributes) set for instance 'KPP'

Attribute	Value	Type	Expression
portAddress	0x020B8000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.37 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: CAN2

3.37.1 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.

3.37.2 Licensing
Open Source Apache 2.0

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

3.37.4 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 26. Configuration options (attributes) set for instance 'CAN2'

Attribute	Value	Type	Expression
portAddress	0x02094000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.38 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: CAN1

3.38.1 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.

3.38.2 Licensing
Open Source Apache 2.0

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

3.38.4 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 27. Configuration options (attributes) set for instance 'CAN1'

Attribute	Value	Type	Expression
portAddress	0x02090000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.39 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: PWM4

3.39.1 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.

3.39.2 Licensing
Open Source Apache 2.0

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

3.39.4 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 28. Configuration options (attributes) set for instance 'PWM4'

Attribute	Value	Type	Expression
portAddress	0x0208C000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.40 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: PWM3

3.40.1 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.

3.40.2 Licensing
Open Source Apache 2.0

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

3.40.4 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 29. Configuration options (attributes) set for instance 'PWM3'

Attribute	Value	Type	Expression
portAddress	0x02088000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.41 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: PWM2

3.41.1 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.

3.41.2 Licensing
Open Source Apache 2.0

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

3.41.4 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 30. Configuration options (attributes) set for instance 'PWM2'

Attribute	Value	Type	Expression
portAddress	0x02084000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.42 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: PWM1

3.42.1 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.

3.42.2 Licensing
Open Source Apache 2.0

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

3.42.4 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 31. Configuration options (attributes) set for instance 'PWM1'

Attribute	Value	Type	Expression
portAddress	0x02080000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.43 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: AIPS1_Cfg

3.43.1 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.

3.43.2 Licensing
Open Source Apache 2.0

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

3.43.4 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 32. Configuration options (attributes) set for instance 'AIPS1_Cfg'

Attribute	Value	Type	Expression
portAddress	0x0207C000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.44 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: SPBA

3.44.1 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.

3.44.2 Licensing
Open Source Apache 2.0

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

3.44.4 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 33. Configuration options (attributes) set for instance 'SPBA'

Attribute	Value	Type	Expression
portAddress	0x0203C000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.45 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: ASRC

3.45.1 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.

3.45.2 Licensing
Open Source Apache 2.0

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

3.45.4 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 34. Configuration options (attributes) set for instance 'ASRC'

Attribute	Value	Type	Expression
portAddress	0x02034000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.46 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: SSI3

3.46.1 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.

3.46.2 Licensing
Open Source Apache 2.0

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

3.46.4 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 35. Configuration options (attributes) set for instance 'SSI3'

Attribute	Value	Type	Expression
portAddress	0x02030000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.47 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: SSI2

3.47.1 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.

3.47.2 Licensing
Open Source Apache 2.0

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

3.47.4 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 36. Configuration options (attributes) set for instance 'SSI2'

Attribute	Value	Type	Expression
portAddress	0x0202C000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.48 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: SSI1

3.48.1 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.

3.48.2 Licensing
Open Source Apache 2.0

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

3.48.4 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 37. Configuration options (attributes) set for instance 'SSI1'

Attribute	Value	Type	Expression
portAddress	0x02028000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.49 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: ESAI

3.49.1 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.

3.49.2 Licensing
Open Source Apache 2.0

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

3.49.4 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 38. Configuration options (attributes) set for instance 'ESAI'

Attribute	Value	Type	Expression
portAddress	0x02024000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.50 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: eCSPI4

3.50.1 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.

3.50.2 Licensing
Open Source Apache 2.0

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

3.50.4 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 39. Configuration options (attributes) set for instance 'eCSPI4'

Attribute	Value	Type	Expression
portAddress	0x02014000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.51 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: eCSPI3

3.51.1 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.

3.51.2 Licensing
Open Source Apache 2.0

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

3.51.4 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 40. Configuration options (attributes) set for instance 'eCSPI3'

Attribute	Value	Type	Expression
portAddress	0x02010000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.52 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: eCSPI2

3.52.1 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.

3.52.2 Licensing
Open Source Apache 2.0

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

3.52.4 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 41. Configuration options (attributes) set for instance 'eCSPI2'

Attribute	Value	Type	Expression
portAddress	0x0200C000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.53 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: eCSPI1

3.53.1 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.

3.53.2 Licensing
Open Source Apache 2.0

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

3.53.4 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 42. Configuration options (attributes) set for instance 'eCSPI1'

Attribute	Value	Type	Expression
portAddress	0x02008000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.54 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: SPDIF

3.54.1 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.

3.54.2 Licensing
Open Source Apache 2.0

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

3.54.4 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 43. Configuration options (attributes) set for instance 'SPDIF'

Attribute	Value	Type	Expression
portAddress	0x02004000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.55 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: I2C4

3.55.1 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.

3.55.2 Licensing
Open Source Apache 2.0

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

3.55.4 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 44. Configuration options (attributes) set for instance 'I2C4'

Attribute	Value	Type	Expression
portAddress	0x021F8000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.56 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: UART5

3.56.1 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.

3.56.2 Licensing
Open Source Apache 2.0

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

3.56.4 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 45. Configuration options (attributes) set for instance 'UART5'

Attribute	Value	Type	Expression
portAddress	0x021F4000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.57 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: UART4

3.57.1 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.

3.57.2 Licensing
Open Source Apache 2.0

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

3.57.4 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 46. Configuration options (attributes) set for instance 'UART4'

Attribute	Value	Type	Expression
portAddress	0x021F0000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.58 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: UART3

3.58.1 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.

3.58.2 Licensing
Open Source Apache 2.0

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

3.58.4 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 47. Configuration options (attributes) set for instance 'UART3'

Attribute	Value	Type	Expression
portAddress	0x021EC000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.59 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: VDOA

3.59.1 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.

3.59.2 Licensing
Open Source Apache 2.0

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

3.59.4 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 48. Configuration options (attributes) set for instance 'VDOA'

Attribute	Value	Type	Expression
portAddress	0x021E4000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.60 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: MIPI_DSI

3.60.1 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.

3.60.2 Licensing
Open Source Apache 2.0

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

3.60.4 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 49. Configuration options (attributes) set for instance 'MIPI_DSI'

Attribute	Value	Type	Expression
portAddress	0x021E0000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.61 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: MIPI_CSI

3.61.1 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.

3.61.2 Licensing
Open Source Apache 2.0

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

3.61.4 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 50. Configuration options (attributes) set for instance 'MIPI_CSI'

Attribute	Value	Type	Expression
portAddress	0x021DC000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.62 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: AUDMUX

3.62.1 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.

3.62.2 Licensing
Open Source Apache 2.0

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

3.62.4 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 51. Configuration options (attributes) set for instance 'AUDMUX'

Attribute	Value	Type	Expression
portAddress	0x021D8000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.63 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: TZASC2

3.63.1 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.

3.63.2 Licensing
Open Source Apache 2.0

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

3.63.4 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 52. Configuration options (attributes) set for instance 'TZASC2'

Attribute	Value	Type	Expression
portAddress	0x021D4000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.64 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: TZASC1

3.64.1 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.

3.64.2 Licensing
Open Source Apache 2.0

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

3.64.4 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 53. Configuration options (attributes) set for instance 'TZASC1'

Attribute	Value	Type	Expression
portAddress	0x021D0000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.65 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: CSU

3.65.1 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.

3.65.2 Licensing
Open Source Apache 2.0

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

3.65.4 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 54. Configuration options (attributes) set for instance 'CSU'

Attribute	Value	Type	Expression
portAddress	0x021C0000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.66 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: OCOTP_CTRL

3.66.1 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.

3.66.2 Licensing
Open Source Apache 2.0

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

3.66.4 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 55. Configuration options (attributes) set for instance 'OCOTP_CTRL'

Attribute	Value	Type	Expression
portAddress	0x021BC000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.67 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: EIM

3.67.1 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.

3.67.2 Licensing
Open Source Apache 2.0

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

3.67.4 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 56. Configuration options (attributes) set for instance 'EIM'

Attribute	Value	Type	Expression
portAddress	0x021B8000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.68 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: MMDCp1

3.68.1 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.

3.68.2 Licensing
Open Source Apache 2.0

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

3.68.4 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 57. Configuration options (attributes) set for instance 'MMDCp1'

Attribute	Value	Type	Expression
portAddress	0x021B4000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.69 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: ROMCP

3.69.1 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.

3.69.2 Licensing
Open Source Apache 2.0

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

3.69.4 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 58. Configuration options (attributes) set for instance 'ROMCP'

Attribute	Value	Type	Expression
portAddress	0x021AC000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.70 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: I2C3

3.70.1 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.

3.70.2 Licensing
Open Source Apache 2.0

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

3.70.4 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 59. Configuration options (attributes) set for instance 'I2C3'

Attribute	Value	Type	Expression
portAddress	0x021A8000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.71 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: I2C2

3.71.1 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.

3.71.2 Licensing
Open Source Apache 2.0

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

3.71.4 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 60. Configuration options (attributes) set for instance 'I2C2'

Attribute	Value	Type	Expression
portAddress	0x021A4000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.72 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: I2C1

3.72.1 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.

3.72.2 Licensing
Open Source Apache 2.0

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

3.72.4 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 61. Configuration options (attributes) set for instance 'I2C1'

Attribute	Value	Type	Expression
portAddress	0x021A0000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.73 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: MLB150

3.73.1 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.

3.73.2 Licensing
Open Source Apache 2.0

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

3.73.4 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 62. Configuration options (attributes) set for instance 'MLB150'

Attribute	Value	Type	Expression
portAddress	0x0218C000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.74 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: ENET

3.74.1 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.

3.74.2 Licensing
Open Source Apache 2.0

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

3.74.4 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 63. Configuration options (attributes) set for instance 'ENET'

Attribute	Value	Type	Expression
portAddress	0x02188000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.75 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: USBOH3_USB

3.75.1 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.

3.75.2 Licensing
Open Source Apache 2.0

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

3.75.4 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 64. Configuration options (attributes) set for instance 'USBOH3_USB'

Attribute	Value	Type	Expression
portAddress	0x02184000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.76 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: USBOH3_PL301

3.76.1 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.

3.76.2 Licensing
Open Source Apache 2.0

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

3.76.4 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 65. Configuration options (attributes) set for instance 'USBOH3_PL301'

Attribute	Value	Type	Expression
portAddress	0x02180000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.77 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: AIPS2_Cfg

3.77.1 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.

3.77.2 Licensing
Open Source Apache 2.0

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

3.77.4 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 66. Configuration options (attributes) set for instance 'AIPS2_Cfg'

Attribute	Value	Type	Expression
portAddress	0x0217C000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.78 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: CAAM

3.78.1 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.

3.78.2 Licensing
Open Source Apache 2.0

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

3.78.4 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 67. Configuration options (attributes) set for instance 'CAAM'

Attribute	Value	Type	Expression
portAddress	0x02100000	uns32	
portSize	65536	uns32	
cbEnable	True	boolean	

3.79 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: IPU

3.79.1 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.

3.79.2 Licensing
Open Source Apache 2.0

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

3.79.4 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 68. Configuration options (attributes) set for instance 'IPU'

Attribute	Value	Type	Expression
portAddress	0x02600000	uns32	
portSize	4194304	uns32	
cbEnable	True	boolean	

3.80 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: GPU2D

3.80.1 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.

3.80.2 Licensing
Open Source Apache 2.0

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

3.80.4 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 69. Configuration options (attributes) set for instance 'GPU2D'

Attribute	Value	Type	Expression
portAddress	0x00134000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.81 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: GPU3D

3.81.1 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.

3.81.2 Licensing
Open Source Apache 2.0

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

3.81.4 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 70. Configuration options (attributes) set for instance 'GPU3D'

Attribute	Value	Type	Expression
portAddress	0x00130000	uns32	
portSize	16384	uns32	
cbEnable	True	boolean	

3.82 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: HDMI

3.82.1 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.

3.82.2 Licensing
Open Source Apache 2.0

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

3.82.4 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 71. Configuration options (attributes) set for instance 'HDMI'

Attribute	Value	Type	Expression
portAddress	0x00120000	uns32	
portSize	36864	uns32	
cbEnable	True	boolean	

3.83 Peripheral [ovpworld.org/peripheral/trap/1.0] instance: APBH_DMA

3.83.1 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.

3.83.2 Licensing
Open Source Apache 2.0

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

3.83.4 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 72. Configuration options (attributes) set for instance 'APBH_DMA'

Attribute	Value	Type	Expression
portAddress	0x00110000	uns32	
portSize	8192	uns32	
cbEnable	True	boolean	

3.84 Peripheral [arm.ovpworld.org/peripheral/SmartLoaderArmLinux/1.0] instance: smartLoader

3.84.1 Licensing
Open Source Apache 2.0

3.84.2 Description
Psuedo-peripheral to perform memory initialisation for an ARM based Linux kernel boot: Loads Linux kernel image file and (optional) initial ram disk image into memory. Writes ATAG data into memory. Writes tiny boot code at physical memory base that configures the registers as expected by Linux Kernel and then jumps to boot address (image load address by default).

3.84.3 Limitations
Only supports little endian

3.84.4 Reference
See ARM Linux boot requirements in Linux source tree at documentation/arm/Booting

Table 73. Configuration options (attributes) set for instance 'smartLoader'

Attribute	Value	Type	Expression
physicalbase	0x10000000	uns32	
memsize	0x20000000	uns32	
kerneladdr	0x11000000	uns32	

3.85 Peripheral [ovpworld.org/peripheral/VirtioBlkMMIO/1.0] instance: VBD0

3.85.1 Description
VIRTIO version 1 mmio block device This model implements a VIRTIO MMIO block device as

described in: http://docs.oasis-open.org/virtio/virtio/v1.0/virtio-v1.0.pdf. Use the VB_DRIVE parameter to specify the disk image file to use. Set the VB_DRIVE_DELTA parameter to 1 to prevent writes to disk during simulation from changing the image file.

3.85.2 Limitations

Only supports the Legacy (Device Version 1) interface. Only little endian guests are supported.

3.85.3 Licensing

Open Source Apache 2.0

3.85.4 Reference

http://docs.oasis-open.org/virtio/virtio/v1.0/virtio-v1.0.pdf

There are no configuration options set for this peripheral instance.

4.0 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 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.

5.0 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.

6.0 Simulating Software

6.1 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.

6.2 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.

6.3 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.

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.

6.4 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.

6.5 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.

6.6 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.

6.7 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.

7.0 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.

7.1 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.

*7.2 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.

7.3 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.

*7.4 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.

7.5 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

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.

7.6 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.

8.0 Modifying the Platform / Module

8.1 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.

8.2 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.

8.3 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.

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.

8.4 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.

8.5 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.

9.0 Available Virtual Platforms

Table 74. 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 75. 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 76. 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 77. 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

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

#