Imperas Peripheral Model Guide

Model Specific Information for renesas.ovpworld.org / intc

Imperas Software Limited

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

Author	Imperas Software Limited
Version	20150901.0
Filename	OVP_Peripheral_Specific_Information_intc.pdf
Created	26 August 2015
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 Model Specific Information 4
1.1 Licensing 4
1.2 Description 4
1.3 Limitations 4
1.4 Reference 4
1.5 Location 4
2.0 Net Ports 4
3.0 Bus Slave Ports 8
3.1 Bus Slave Port: INTCP0 8
4.0 Platforms that use this peripheral component 11
5.0 Peripheral components in the library 12
6.0 General Information on Peripheral Models 14
6.1 Background 14
7.0 Building peripherals easily with Imperas iGen 14
8.0 Peripheral model internals 14
9.0 Parts of peripheral models 15
9.1 Configuring the Peripheral Instance with Parameters 15
9.2 Net Ports 15
9.3 Bus master ports 15
9.4 Bus slave ports 15
9.5 Packetnets 15
10.0 More information (documentation) on peripheral models and modeling 15

1.0 Model Specific Information

This document provides usage information for an Imperas OVP peripheral behavioral model.

The document is split into sections providing specific information for this peripheral, including any ports for connecting into a platform, registers, other component parts, and configuration options and general information for peripheral modeling with Imperas OVP.

1.1 Licensing

Open Source Apache 2.0

1.2 Description

Renesas INTC Interrupt Controller

1.3 Limitations

Register View Model Only

1.4 Reference

R01UH0128ED0700, Rev. 7.00, Oct 06, 2010

1.5 Location

The intc peripheral model is located in an Imperas/OVP installation at the VLNV: renesas.ovpworld.org / peripheral / intc / 1.0.

2.0 Net Ports

This model has the following net ports:

Table 1. Net Ports

Name	Type	Must Be Connected	Description
RESET	output	F (False)	
NMI0	output	F (False)	
NMI1	output	F (False)	
NMI2	output	F (False)	
INTP	output	F (False)	
INTACK	input	F (False)	
MIRETI	input	F (False)	
IRESET	input	F (False)	
NMI_00	input	F (False)	
NMI_01	input	F (False)	
NMI_02	input	F (False)	
NMI_03	input	F (False)	
NMI_04	input	F (False)	
NMI_05	input	F (False)	
NMI_06	input	F (False)	

NMI_07	input	F (False)	
NMI_08	input	F (False)	
NMI_09	input	F (False)	
NMI_10	input	F (False)	
NMI_11	input	F (False)	
NMI_12	input	F (False)	
NMI_13	input	F (False)	
NMI_14	input	F (False)	
NMI_15	input	F (False)	
NMI_16	input	F (False)	
NMI_17	input	F (False)	
NMI_18	input	F (False)	
NMI_19	input	F (False)	
NMI_20	input	F (False)	
NMI_21	input	F (False)	
NMI_22	input	F (False)	
NMI_23	input	F (False)	
NMI_24	input	F (False)	
NMI_25	input	F (False)	
NMI_26	input	F (False)	
NMI_27	input	F (False)	
NMI_28	input	F (False)	
NMI_29	input	F (False)	
NMI_30	input	F (False)	
NMI_31	input	F (False)	
NMI_32	input	F (False)	
NMI_33	input	F (False)	
NMI_34	input	F (False)	
NMI_35	input	F (False)	
NMI_36	input	F (False)	
NMI_37	input	F (False)	
NMI_38	input	F (False)	
NMI_39	input	F (False)	
NMI_40	input	F (False)	
NMI_41	input	F (False)	
NMI_42	input	F (False)	
NMI_43	input	F (False)	
NMI_44	input	F (False)	
NMI_45	input	F (False)	
NMI_46	input	F (False)	
NMI_47	input	F (False)	
NMI_48	input	F (False)	
NMI_49	input	F (False)	
NMI_50	input	F (False)	
NMI_51	input	F (False)	
NMI_52	input	F (False)	
NMI_53	input	F (False)	
NMI_54	input	F (False)	

NMI_55	input	F (False)	
NMI_56	input	F (False)	
NMI_57	input	F (False)	
NMI_58	input	F (False)	
NMI_59	input	F (False)	
NMI_60	input	F (False)	
NMI_61	input	F (False)	
NMI_62	input	F (False)	
NMI_63	input	F (False)	
INT_00	input	F (False)	
INT_01	input	F (False)	
INT_02	input	F (False)	
INT_03	input	F (False)	
INT_04	input	F (False)	
INT_05	input	F (False)	
INT_06	input	F (False)	
INT_07	input	F (False)	
INT_08	input	F (False)	
INT_09	input	F (False)	
INT_10	input	F (False)	
INT_11	input	F (False)	
INT_12	input	F (False)	
INT_13	input	F (False)	
INT_14	input	F (False)	
INT_15	input	F (False)	
INT_16	input	F (False)	
INT_17	input	F (False)	
INT_18	input	F (False)	
INT_19	input	F (False)	
INT_20	input	F (False)	
INT_21	input	F (False)	
INT_22	input	F (False)	
INT_23	input	F (False)	
INT_24	input	F (False)	
INT_25	input	F (False)	
INT_26	input	F (False)	
INT_27	input	F (False)	
INT_28	input	F (False)	
INT_29	input	F (False)	
INT_30	input	F (False)	
INT_31	input	F (False)	
INT_32	input	F (False)	
INT_33	input	F (False)	
INT_34	input	F (False)	
INT_35	input	F (False)	
INT_36	input	F (False)	
INT_37	input	F (False)	
INT_38	input	F (False)	

INT_39	input	F (False)	
INT_40	input	F (False)	
INT_41	input	F (False)	
INT_42	input	F (False)	
INT_43	input	F (False)	
INT_44	input	F (False)	
INT_45	input	F (False)	
INT_46	input	F (False)	
INT_47	input	F (False)	
INT_48	input	F (False)	
INT_49	input	F (False)	
INT_50	input	F (False)	
INT_51	input	F (False)	
INT_52	input	F (False)	
INT_53	input	F (False)	
INT_54	input	F (False)	
INT_55	input	F (False)	
INT_56	input	F (False)	
INT_57	input	F (False)	
INT_58	input	F (False)	
INT_59	input	F (False)	
INT_60	input	F (False)	
INT_61	input	F (False)	
INT_62	input	F (False)	
INT_63	input	F (False)	
INT_64	input	F (False)	
INT_65	input	F (False)	
INT_66	input	F (False)	
INT_67	input	F (False)	
INT_68	input	F (False)	
INT_69	input	F (False)	
INT_70	input	F (False)	
INT_71	input	F (False)	
INT_72	input	F (False)	
INT_73	input	F (False)	
INT_74	input	F (False)	
INT_75	input	F (False)	
INT_76	input	F (False)	
INT_77	input	F (False)	
INT_78	input	F (False)	
INT_79	input	F (False)	
INT_80	input	F (False)	
INT_81	input	F (False)	
INT_82	input	F (False)	
INT_83	input	F (False)	
INT_84	input	F (False)	
INT_85	input	F (False)	
INT_86	input	F (False)	

INT_87	input	F (False)	
INT_88	input	F (False)	
INT_89	input	F (False)	
INT_90	input	F (False)	
INT_91	input	F (False)	
INT_92	input	F (False)	
INT_93	input	F (False)	
INT_94	input	F (False)	
INT_95	input	F (False)	
INT_96	input	F (False)	
INT_97	input	F (False)	
INT_98	input	F (False)	
INT_99	input	F (False)	
INT_100	input	F (False)	
INT_101	input	F (False)	
INT_102	input	F (False)	
INT_103	input	F (False)	
INT_104	input	F (False)	
INT_105	input	F (False)	
INT_106	input	F (False)	
INT_107	input	F (False)	
INT_108	input	F (False)	
INT_109	input	F (False)	
INT_110	input	F (False)	
INT_111	input	F (False)	
INT_112	input	F (False)	
INT_113	input	F (False)	
INT_114	input	F (False)	
INT_115	input	F (False)	
INT_116	input	F (False)	

3.0 Bus Slave Ports

This model has the following bus slave ports:

3.1 Bus Slave Port: INTCP0

Table 2. Bus Slave Port: INTCP0

Name	Size (bytes)	Must Be Connected	Description
INTCP0	0xfc	F (False)	

Table 3. Bus Slave Port: INTCP0 Registers:

Name	Offset	Width (bits)	Description	R/W	is Volatile
reg0_IMR0	0x0	16			
reg0_IMR1	0x2	16			
reg0_IMR2	0x4	16			
reg0_IMR3	0x6	16			
reg0_IMR4	0x8	16			
reg0_IMR5	0xa	16			

reg0_IMR6	0xc	16			
reg0_IMR7	0xe	16			
reg1_IC000	0x10	8			
reg1_IC001	0x12	8			
reg1_IC002	0x14	8			
reg1_IC003	0x16	8			
reg1_IC004	0x18	8			
reg1_IC005	0x1a	8			
reg1_IC006	0x1c	8			
reg1_IC007	0x1e	8			
reg1_IC008	0x20	8			
reg1_IC009	0x22	8			
reg1_IC010	0x24	8			
reg1_IC011	0x26	8			
reg1_IC012	0x28	8			
reg1_IC013	0x2a	8			
reg1_IC014	0x2c	8			
reg1_IC015	0x2e	8			
reg1_IC016	0x30	8			
reg1_IC017	0x32	8			
reg1_IC018	0x34	8			
reg1_IC019	0x36	8			
reg1_IC020	0x38	8			
reg1_IC021	0x3a	8			
reg1_IC022	0x3c	8			
reg1_IC023	0x3e	8			
reg1_IC024	0x40	8			
reg1_IC025	0x42	8			
reg1_IC026	0x44	8			
reg1_IC027	0x46	8			
reg1_IC028	0x48	8			
reg1_IC029	0x4a	8			
reg1_IC030	0x4c	8			
reg1_IC031	0x4e	8			
reg1_IC032	0x50	8			
reg1_IC033	0x52	8			
reg1_IC034	0x54	8			
reg1_IC035	0x56	8			
reg1_IC036	0x58	8			
reg1_IC037	0x5a	8			
reg1_IC038	0x5c	8			
reg1_IC039	0x5e	8			
reg1_IC040	0x60	8			
reg1_IC041	0x62	8			
reg1_IC042	0x64	8			
reg1_IC043	0x66	8			
reg1_IC044	0x68	8			
reg1_IC045	0x6a	8			

reg1_IC046	0x6c	8			
reg1_IC047	0x6e	8			
reg1_IC048	0x70	8			
reg1_IC049	0x72	8			
reg1_IC050	0x74	8			
reg1_IC051	0x76	8			
reg1_IC052	0x78	8			
reg1_IC053	0x7a	8			
reg1_IC054	0x7c	8			
reg1_IC055	0x7e	8			
reg1_IC056	0x80	8			
reg1_IC057	0x82	8			
reg1_IC058	0x84	8			
reg1_IC059	0x86	8			
reg1_IC060	0x88	8			
reg1_IC061	0x8a	8			
reg1_IC062	0x8c	8			
reg1_IC063	0x8e	8			
reg1_IC064	0x90	8			
reg1_IC065	0x92	8			
reg1_IC066	0x94	8			
reg1_IC067	0x96	8			
reg1_IC068	0x98	8			
reg1_IC069	0x9a	8			
reg1_IC070	0x9c	8			
reg1_IC071	0x9e	8			
reg1_IC072	0xa0	8			
reg1_IC073	0xa2	8			
reg1_IC074	0xa4	8			
reg1_IC075	0xa6	8			
reg1_IC076	0xa8	8			
reg1_IC077	0xaa	8			
reg1_IC078	0xac	8			
reg1_IC079	0xae	8			
reg1_IC080	0xb0	8			
reg1_IC081	0xb2	8			
reg1_IC082	0xb4	8			
reg1_IC083	0xb6	8			
reg1_IC084	0xb8	8			
reg1_IC085	0xba	8			
reg1_IC086	0xbc	8			
reg1_IC087	0xbe	8			
reg1_IC088	0xc0	8			
reg1_IC089	0xc2	8			
reg1_IC090	0xc4	8			
reg1_IC091	0xc6	8			
reg1_IC092	0xc8	8			
reg1_IC093	0xca	8			

reg1_IC094	0xcc	8			
reg1_IC095	0xce	8			
reg1_IC096	0xd0	8			
reg1_IC097	0xd2	8			
reg1_IC098	0xd4	8			
reg1_IC099	0xd6	8			
reg1_IC100	0xd8	8			
reg1_IC101	0xda	8			
reg1_IC102	0xdc	8			
reg1_IC103	0xde	8			
reg1_IC104	0xe0	8			
reg1_IC105	0xe2	8			
reg1_IC106	0xe4	8			
reg1_IC107	0xe6	8			
reg1_IC108	0xe8	8			
reg1_IC109	0xea	8			
reg1_IC110	0xec	8			
reg1_IC111	0xee	8			
reg1_IC112	0xf0	8			
reg1_IC113	0xf2	8			
reg1_IC114	0xf4	8			
reg1_IC115	0xf6	8			
reg1_IC116	0xf8	8			
reg1_ISPR	0xfa	8			

4.0 Platforms that use this peripheral component

Peripheral components can be used in many different platforms, including those developed by Imperas or by other users of OVP. You can use this peripheral in your own platforms.

Table 4. Publicly available platforms using peripheral 'intc'

Platform Name	Vendor
RenesasUPD70F3441	renesas.ovpworld.org

This peripheral is used in some internal Imperas virtual platforms. Please contact Imperas for more information.

5.0 Peripheral components in the library

Table 5. Publicly available Imperas/OVP peripheral models (158 models)

Peripheral	Peripheral	Peripheral
renesas.ovpworld.org/memc	renesas.ovpworld.org/rng	renesas.ovpworld.org/taa
renesas.ovpworld.org/tms	renesas.ovpworld.org/tmt	renesas.ovpworld.org/uartc
renesas.ovpworld.org/UPD70F3441Logic	smsc.ovpworld.org/LAN9118	smsc.ovpworld.org/LAN91C111
ti.ovpworld.org/UartInterface	xilinx.ovpworld.org/mdm	xilinx.ovpworld.org/mpmc
xilinx.ovpworld.org/xps-gpio	xilinx.ovpworld.org/xps-iic	xilinx.ovpworld.org/xps-intc
xilinx.ovpworld.org/xps-ll-temac	xilinx.ovpworld.org/xps-mch-emc	xilinx.ovpworld.org/xps-sysace
xilinx.ovpworld.org/xps-timer	xilinx.ovpworld.org/xps-uartlite	altera.ovpworld.org/dw-apb-timer
altera.ovpworld.org/dw-apb-uart	altera.ovpworld.org/IntervalTimer32Core	altera.ovpworld.org/IntervalTimer64Core
altera.ovpworld.org/JtagUart	altera.ovpworld.org/PerformanceCounterCore	altera.ovpworld.org/RSTMGR
altera.ovpworld.org/SystemIDCore	altera.ovpworld.org/Uart	amd.ovpworld.org/79C970
arm.ovpworld.org/AaciPL041	arm.ovpworld.org/CompactFlashRegs	arm.ovpworld.org/CoreModule9x6
arm.ovpworld.org/DebugLedAndDipSwitch	arm.ovpworld.org/DMemCtrlPL341	arm.ovpworld.org/IcpControl
arm.ovpworld.org/IcpCounterTimer	arm.ovpworld.org/IntICP	arm.ovpworld.org/IntICP
arm.ovpworld.org/KbPL050	arm.ovpworld.org/L2CachePL310	arm.ovpworld.org/LcdPL110
arm.ovpworld.org/MmciPL181	arm.ovpworld.org/RtcPL031	arm.ovpworld.org/SerBusDviRegs
arm.ovpworld.org/SmartLoaderArm64Linux	arm.ovpworld.org/SmartLoaderArmLinux	arm.ovpworld.org/SMemCtrlPL354
arm.ovpworld.org/SysCtrlSP810	arm.ovpworld.org/TimerSP804	arm.ovpworld.org/TzpcBP147
arm.ovpworld.org/UartPL011	arm.ovpworld.org/VexpressSysRegs	arm.ovpworld.org/WdtSP805
atmel.ovpworld.org/AdvancedInterruptController	atmel.ovpworld.org/ParallelIOController	atmel.ovpworld.org/PowerSaving
atmel.ovpworld.org/SpecialFunction	atmel.ovpworld.org/TimerCounter	atmel.ovpworld.org/UsartInterface
atmel.ovpworld.org/WatchdogTimer	cirrus.ovpworld.org/GD5446	freescale.ovpworld.org/KinetisADC
freescale.ovpworld.org/KinetisAIPS	freescale.ovpworld.org/KinetisAXBS	freescale.ovpworld.org/KinetisCAN
freescale.ovpworld.org/KinetisCMP	freescale.ovpworld.org/KinetisCMT	freescale.ovpworld.org/KinetisCRC
freescale.ovpworld.org/KinetisDAC	freescale.ovpworld.org/KinetisDDR	freescale.ovpworld.org/KinetisDMA
freescale.ovpworld.org/KinetisDMAC	freescale.ovpworld.org/KinetisDMAMUX	freescale.ovpworld.org/KinetisENET
freescale.ovpworld.org/KinetisEWM	freescale.ovpworld.org/KinetisFB	freescale.ovpworld.org/KinetisFMC
freescale.ovpworld.org/KinetisFTFE	freescale.ovpworld.org/KinetisFTM	freescale.ovpworld.org/KinetisGPIO
freescale.ovpworld.org/KinetisI2C	freescale.ovpworld.org/KinetisI2S	freescale.ovpworld.org/KinetisLLWU
freescale.ovpworld.org/KinetisLPTMR	freescale.ovpworld.org/KinetisMCG	freescale.ovpworld.org/KinetisMPU
freescale.ovpworld.org/KinetisNFC	freescale.ovpworld.org/KinetisOSC	freescale.ovpworld.org/KinetisPDB
freescale.ovpworld.org/KinetisPIT	freescale.ovpworld.org/KinetisPMC	freescale.ovpworld.org/KinetisPORT
freescale.ovpworld.org/KinetisRCM	freescale.ovpworld.org/KinetisRFSYS	freescale.ovpworld.org/KinetisRFVBAT
freescale.ovpworld.org/KinetisRNG	freescale.ovpworld.org/KinetisRTC	freescale.ovpworld.org/KinetisSDHC
freescale.ovpworld.org/KinetisSIM	freescale.ovpworld.org/KinetisSMC	freescale.ovpworld.org/KinetisSPI
freescale.ovpworld.org/KinetisTSI	freescale.ovpworld.org/KinetisUART	freescale.ovpworld.org/KinetisUSB
freescale.ovpworld.org/KinetisUSBDCD	freescale.ovpworld.org/KinetisUSBHS	freescale.ovpworld.org/KinetisVREF
freescale.ovpworld.org/KinetisWDOG	freescale.ovpworld.org/Uart	freescale.ovpworld.org/VybridADC
freescale.ovpworld.org/VybridANADIG	freescale.ovpworld.org/VybridCCM	freescale.ovpworld.org/VybridDMA
freescale.ovpworld.org/VybridGPIO	freescale.ovpworld.org/VybridI2C	freescale.ovpworld.org/VybridLCD
freescale.ovpworld.org/VybridQUADSPI	freescale.ovpworld.org/VybridSDHC	freescale.ovpworld.org/VybridSPI
freescale.ovpworld.org/VybridUART	freescale.ovpworld.org/VybridUSB	intel.ovpworld.org/82077AA
intel.ovpworld.org/82371EB	intel.ovpworld.org/8253	intel.ovpworld.org/8259A

intel.ovpworld.org/NorFlash48F4400	intel.ovpworld.org/PciIDE	intel.ovpworld.org/PciPM
intel.ovpworld.org/PciUSB	intel.ovpworld.org/Ps2Control	marvell.ovpworld.org/GT6412x
mips.ovpworld.org/16450C	mips.ovpworld.org/MaltaFPGA	mips.ovpworld.org/SmartLoaderLinux
motorola.ovpworld.org/MC146818	national.ovpworld.org/16450	national.ovpworld.org/16550
ovpworld.org/Alpha2x16Display	ovpworld.org/dummyPort	ovpworld.org/DynamicBridge
ovpworld.org/FlashDevice	ovpworld.org/ledRegister	ovpworld.org/SerInt
ovpworld.org/SimpleDma	ovpworld.org/VirtioBlkMMIO	philips.ovpworld.org/ISP1761
renesas.ovpworld.org/adc	renesas.ovpworld.org/bcu	renesas.ovpworld.org/brg
renesas.ovpworld.org/can	renesas.ovpworld.org/can	renesas.ovpworld.org/clkgen
renesas.ovpworld.org/crc	renesas.ovpworld.org/csib	renesas.ovpworld.org/csie
renesas.ovpworld.org/dma	renesas.ovpworld.org/intc	

6.0 General Information on Peripheral Models

This document provides usage information for an Imperas OVP peripheral behavioral model.

The document is split into sections providing specific information for this peripheral, including any ports for connecting into a platform, registers etc. and configuration options and general information for peripheral modeling with Imperas OVP.

6.1 Background

Imperas OVP simulation technology enables very high performance simulation, debug and analysis of platforms containing multiple processors and peripheral models. The technology is designed to be extensible: you can create new models of processors, peripherals and other platform components using interfaces and libraries defined by OVP.

The peripheral models created using the OVP APIs run on the Peripheral Simulation Engine (PSE).

The model is typically written in C and compiled into an executable for the PSE processor architecture. The model is compiled for speed of execution and to protect IP. It is dynamically loaded by the simulator at run time.

7.0 Building peripherals easily with Imperas iGen

To aid with model creation, Imperas products include iGen, a model generation tool. iGen takes the laborious and error-prone task of constructing the various hardware model and software element files required for a typical model, and automates this process. iGen creates the needed C files. iGen also creates the C++ SystemC TLM2 interface files needed to run peripheral models in SystemC simulations.

iGen takes as input a simple script specification that includes device internals such as registers and memories, port information, component descriptors, and other elements. iGen then builds the C code model files and user editable templates. These include model frameworks with registers, function calls, memory map, and other items. It ensures that all component parts of the model are well-structured using best practices, and are consistent throughout the files, thus eliminating a common source of errors.

More information on iGen can be found: imperas.com/products.

Please contact Imperas to get access to the Imperas documents: Imperas_Model_Generator_Guide.pdf and Imperas_Peripheral_Generator_Guide.pdf.

8.0 Peripheral model internals

Each instance of a peripheral model runs on its own virtual machine with an address space large enough for the model. This processor (the PSE) and its memory are separate from any processors, memories and buses

in the platform being simulated; they exist only to execute the code of the peripheral model.

Interception of functions defined in the peripheral model allows the use of features of the host system in the implementation of the behavior of a peripheral. As an example, a real platform might contain a video display device. When simulating this system, it is generally more convenient not to simulate the complete video display device but to use a video package available on the host machine, such as SDL, and to use this to render to the host display. Also models of uarts, ethernet devices and USB components can make use of the host PC resources during simulation, to allow, for example, a simulation to browse the real internet, or the simulation to connect to a real USB device.

9.0 Parts of peripheral models

9.1 Configuring the Peripheral Instance with Parameters

A peripheral can include the behaviour of several configurations. These are controlled when the peripheral is instanced in the platform by setting parameters defined on the peripheral.

9.2 Net Ports

Peripherals may be connected to other peripherals or processors with signal wires (nets). These can be used to act as interrupt signals or used to control behavior between peripherals.
The wires are created in the platform as nets and this net is connected into the peripheral using a net port.

9.3 Bus master ports

A bus master port initiates (and controls the address of) a bus cycle. Bus cycles are generated by behavioral code within the peripheral model.

9.4 Bus slave ports

A peripheral can be defined as having several bus slave ports. The bus slave ports can be split into several address blocks. Each address block be either local memory or memory mapped registers. Both of these can have associated callback functions. A memory mapped register can also be defined as specific read/write access, whether it is volatile, and also whether it is associated with a reset pin and mask. A memory mapped register can also have specific bit fields defined.

9.5 Packetnets

A peripheral can be defined as being connected to packetnet ports. A packetnet is used to model packet based communication such as Ethernet, CAN bus or GSM. A packetnet is created in a platform, then connected to packetnet ports on model instances. A packetnet can have many connections, each able to send or receive packets. A packetnet is used as an efficient method of communication within OVP models.

For more information on modeling with packetnets, please see the peripheral modeling documentation: OVP_Peripheral_Modeling_Guide.pdf, OVPsim_and_CpuManager_User_Guide.pdf and the example: $IMPERAS_HOME/Examples/Models/Peripherals/packetnet.

10.0 More information (documentation) on peripheral models and modeling

More information on modeling and APIs can be found at: OVPworld.org/technology_apis.

Specifics on modeling peripherals can be found: OVP_Peripheral_Modeling_Guide.pdf.

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