Imperas Peripheral Model Guide

Model Specific Information for nxp.ovpworld.org / iMX6_Analog

Imperas Software Limited

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

Author	Imperas Software Limited
Version	20211118.0
Filename	OVP_Peripheral_Specific_Information_iMX6_Analog.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 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 Description

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

1.2 Licensing

Open Source Apache 2.0

1.3 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.

1.4 Reference

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

1.5 Location

The iMX6_Analog peripheral model is located in an Imperas/OVP installation at the VLNV: nxp.ovpworld.org / peripheral / iMX6_Analog / 1.0.

2.0 Bus Slave Ports

This model has the following bus slave ports:

2.1 Bus Slave Port: bport1

Table 1. Bus Slave Port: bport1

Name	Size (bytes)	Must Be Connected	Description
bport1	0x1000	T (True)	

Table 2. Bus Slave Port: bport1 Registers:

Name	Offset	Width (bits)	Description	R/W	is Volatile
ab_CCM_ANALOG_PLL_ARM	0x0	32	Analog ARM PLL control Register		
ab_CCM_ANALOG_PLL_ARM_SET	0x4	32			
ab_CCM_ANALOG_PLL_ARM_CLR	0x8	32			
ab_CCM_ANALOG_PLL_ARM_TOG	0xc	32			
ab_CCM_ANALOG_PL	0x10	32	Analog USB1 480MHz		

L_USB1			PLL Control Register		
ab_CCM_ANALOG_PL L_USB1_SET	0x14	32			
ab_CCM_ANALOG_PL L_USB1_CLR	0x18	32			
ab_CCM_ANALOG_PL L_USB1_TOG	0x1c	32			
ab_CCM_ANALOG_PL L_USB2	0x20	32	Analog USB2 480MHz PLL Control Register		
ab_CCM_ANALOG_PL L_USB2_SET	0x24	32			
ab_CCM_ANALOG_PL L_USB2_CLR	0x28	32			
ab_CCM_ANALOG_PL L_USB2_TOG	0x2c	32			
ab_CCM_ANALOG_PL L_SYS	0x30	32	Analog System PLL Control Register (modified reset value)		
ab_CCM_ANALOG_PL L_SYS_SET	0x34	32			
ab_CCM_ANALOG_PL L_SYS_CLR	0x38	32			
ab_CCM_ANALOG_PL L_SYS_TOG	0x3c	32			
ab_CCM_ANALOG_PL L_SYS_SS	0x40	32	528MHz System PLL Spread Spectrum Register		
ab_CCM_ANALOG_PL L_SYS_SYS_NUM	0x50	32	Numerator of 528MHz System PLL Fractional Loop Divider Register		
ab_CCM_ANALOG_PL L_SYS_SYS_DENOM	0x60	32	Denominator of 528MHz System PLL Fractional Loop Divider Register		
ab_CCM_ANALOG_PL L_AUDIO	0x70	32	Analog Audio PLL control Register		
ab_CCM_ANALOG_PL L_AUDIO_SET	0x74	32			
ab_CCM_ANALOG_PL L_AUDIO_CLR	0x78	32			
ab_CCM_ANALOG_PL L_AUDIO_TOG	0x7c	32			
ab_CCM_ANALOG_PL L_AUDIO_NUM	0x80	32	Numerator of Audio PLL Fractional Loop Divider Register		
ab_CCM_ANALOG_PL L_AUDIO_NUM_SET	0x84	32			
ab_CCM_ANALOG_PL L_AUDIO_NUM_CLR	0x88	32			
ab_CCM_ANALOG_PL L_AUDIO_NUM_TOG	0x8c	32			
ab_CCM_ANALOG_PL L_AUDIO_DENOM	0x90	32	Denominator of Audio PLL Fractional Loop Divider Register		
ab_CCM_ANALOG_PL L_AUDIO_DENOM_SE T	0x94	32			
ab_CCM_ANALOG_PL	0x98	32			

L_AUDIO_DENOM_CLR					
ab_CCM_ANALOG_PLL_AUDIO_DENOM_TOG	0x9c	32			
ab_CCM_ANALOG_PLL_VIDEO	0xa0	32	Analog Video PLL control Register		
ab_CCM_ANALOG_PLL_VIDEO_SET	0xa4	32			
ab_CCM_ANALOG_PLL_VIDEO_CLR	0xa8	32			
ab_CCM_ANALOG_PLL_VIDEO_TOG	0xac	32			
ab_CCM_ANALOG_PLL_VIDEO_NUM	0xb0	32	Numerator of Video PLL Fractional Loop Divider Register		
ab_CCM_ANALOG_PLL_VIDEO_NUM_SET	0xb4	32			
ab_CCM_ANALOG_PLL_VIDEO_NUM_CLR	0xb8	32			
ab_CCM_ANALOG_PLL_VIDEO_NUM_TOG	0xbc	32			
ab_CCM_ANALOG_PLL_VIDEO_DENOM	0xc0	32	Denominator of Video PLL Fractional Loop Divider Register		
ab_CCM_ANALOG_PLL_VIDEO_DENOM_SET	0xc4	32			
ab_CCM_ANALOG_PLL_VIDEO_DENOM_CLR	0xc8	32			
ab_CCM_ANALOG_PLL_VIDEO_DENOM_TOG	0xcc	32			
ab_CCM_ANALOG_PLL_MLB	0xd0	32	Analog MLB PLL Control Register		
ab_CCM_ANALOG_PLL_MLB_SET	0xd4	32			
ab_CCM_ANALOG_PLL_MLB_CLR	0xd8	32			
ab_CCM_ANALOG_PLL_MLB_TOG	0xdc	32			
ab_CCM_ANALOG_PLL_ENET	0xe0	32	Analog ENET PLL Control Register		
ab_CCM_ANALOG_PLL_ENET_SET	0xe4	32			
ab_CCM_ANALOG_PLL_ENET_CLR	0xe8	32			
ab_CCM_ANALOG_PLL_ENET_TOG	0xec	32			
ab_CCM_ANALOG_PFD_480	0xf0	32	480MHz Clock (PLL3) Phase Fractional Divider Control Register		
ab_CCM_ANALOG_PFD_480_SET	0xf4	32			
ab_CCM_ANALOG_PF	0xf8	32			

D_480_CLR					
ab_CCM_ANALOG_PFD_480_TOG	0xfc	32			
ab_CCM_ANALOG_PFD_528	0x100	32	528MHz Clock (PLL2) Phase Fractional Divider Control Register		
ab_CCM_ANALOG_PFD_528_SET	0x104	32			
ab_CCM_ANALOG_PFD_528_CLR	0x108	32			
ab_CCM_ANALOG_PFD_528_TOG	0x10c	32			
ab_PMU_REG_1P1	0x110	32	Regulator 1P1 Register		
ab_PMU_REG_1P1_SET	0x114	32			
ab_PMU_REG_1P1_CLR	0x118	32			
ab_PMU_REG_1P1_TOG	0x11c	32			
ab_PMU_REG_3P0	0x120	32	Regulator 3P0 Register		
ab_PMU_REG_3P0_SET	0x124	32			
ab_PMU_REG_3P0_CLR	0x128	32			
ab_PMU_REG_3P0_TOG	0x12c	32			
ab_PMU_REG_2P5	0x130	32	Regulator 2P5 Register		
ab_PMU_REG_2P5_SET	0x134	32			
ab_PMU_REG_2P5_CLR	0x138	32			
ab_PMU_REG_2P5_TOG	0x13c	32			
ab_PMU_REG_CORE	0x140	32	Digital Regulator Core Registerr		
ab_PMU_REG_CORE_SET	0x144	32			
ab_PMU_REG_CORE_CLR	0x148	32			
ab_PMU_REG_CORE_TOG	0x14c	32			
ab_MISC0	0x150	32	Miscellaneous Register 0		
ab_MISC0_SET	0x154	32			
ab_MISC0_CLR	0x158	32			
ab_MISC0_TOG	0x15c	32			
ab_MISC1	0x160	32	Miscellaneous Register 1		
ab_MISC1_SET	0x164	32			
ab_MISC1_CLR	0x168	32			
ab_MISC1_TOG	0x16c	32			
ab_MISC2	0x170	32	Miscellaneous Register 2		
ab_MISC2_SET	0x174	32			
ab_MISC2_CLR	0x178	32			
ab_MISC2_TOG	0x17c	32			
ab_USB_ANALOG_USB1_VBUS_DETECT	0x1a0	32	USB VBUS Detect Register		

ab_USB_ANALOG_USB1_VBUS_DETECT_SET	0x1a4	32			
ab_USB_ANALOG_USB1_VBUS_DETECT_CLR	0x1a8	32			
ab_USB_ANALOG_USB1_VBUS_DETECT_TOG	0x1ac	32			
ab_USB_ANALOG_USB1_CHRG_DETECT	0x1b0	32	USB Charger Detect Register		
ab_USB_ANALOG_USB1_CHRG_DETECT_SET	0x1b4	32			
ab_USB_ANALOG_USB1_CHRG_DETECT_CLR	0x1b8	32			
ab_USB_ANALOG_USB1_CHRG_DETECT_TOG	0x1bc	32			
ab_USB_ANALOG_USB1_VBUS_DETECT_STAT	0x1c0	32	USB VBUS Detect Status Register		
ab_USB_ANALOG_USB1_CHRG_DETECT_STAT	0x1d0	32	USB Charger Detect Status Register		
ab_USB_ANALOG_USB1_MISC	0x1f0	32	USB Misc Register		
ab_USB_ANALOG_USB1_MISC_SET	0x1f4	32			
ab_USB_ANALOG_USB1_MISC_CLR	0x1f8	32			
ab_USB_ANALOG_USB1_MISC_TOG	0x1fc	32			
ab_USB_ANALOG_USB2_VBUS_DETECT	0x200	32	USB VBUS Detect Register		
ab_USB_ANALOG_USB2_VBUS_DETECT_SET	0x204	32			
ab_USB_ANALOG_USB2_VBUS_DETECT_CLR	0x208	32			
ab_USB_ANALOG_USB2_VBUS_DETECT_TOG	0x20c	32			
ab_USB_ANALOG_USB2_CHRG_DETECT	0x210	32	USB Charger Detect Register		
ab_USB_ANALOG_USB2_CHRG_DETECT_SET	0x214	32			
ab_USB_ANALOG_USB2_CHRG_DETECT_CLR	0x218	32			
ab_USB_ANALOG_USB2_CHRG_DETECT_TOG	0x21c	32			
ab_USB_ANALOG_USB2_VBUS_DETECT_STAT	0x220	32	USB VBUS Detect Status Register		
ab_USB_ANALOG_USB2_CHRG_DETECT_STAT	0x230	32	USB Charger Detect Status Register		
ab_USB_ANALOG_USB2_MISC	0x250	32	USB Misc Register		
ab_USB_ANALOG_USB2_MISC_SET	0x254	32			

ab_USB_ANALOG_USB2_MISC_CLR	0x258	32			
ab_USB_ANALOG_USB2_MISC_TOG	0x25c	32			
ab_USB_ANALOG_DIGPROG	0x260	32	Chip Silicon Version (modified reset value)		
ab_USB_ANALOG_DIGPROG_SET	0x264	32			
ab_USB_ANALOG_DIGPROG_CLR	0x268	32			
ab_USB_ANALOG_DIGPROG_TOG	0x26c	32			

3.0 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 3. Publicly available platforms using peripheral 'iMX6_Analog'

Platform Name	Vendor
iMX6S	nxp.ovpworld.org

4.0 Peripheral components in the library

Table 4. Publicly available Imperas/OVP peripheral models (227 models)

Peripheral	Peripheral	Peripheral
nxp.ovpworld.org/iMX6_CCM	nxp.ovpworld.org/iMX6_GPC	nxp.ovpworld.org/iMX6_GPIO
nxp.ovpworld.org/iMX6_GPT	nxp.ovpworld.org/iMX6_MMDC	nxp.ovpworld.org/iMX6_SDHC
nxp.ovpworld.org/iMX6_SRC	nxp.ovpworld.org/iMX6_UART	nxp.ovpworld.org/iMX6_WDOG
ovpworld.org/Alpha2x16Display	ovpworld.org/DynamicBridge	ovpworld.org/FlashDevice
ovpworld.org/ledRegister	ovpworld.org/SerInt	ovpworld.org/SimpleDma
ovpworld.org/switchRegister	ovpworld.org/temperatureSensor	ovpworld.org/trap
ovpworld.org/trap4K	ovpworld.org/vEthernet_Bridge	ovpworld.org/VirtioBlkMMIO
ovpworld.org/VirtioNetMMIO	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	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	riscv.ovpworld.org/CLINT
riscv.ovpworld.org/PLIC	riscv.ovpworld.org/SmartLoaderRV64Linux	safepower.ovpworld.org/node
safepower.ovpworld.org/NostrumNode	safepower.ovpworld.org/ring_oscillator	safepower.ovpworld.org/TTELNode
sifive.ovpworld.org/artyIO	sifive.ovpworld.org/DDRCTL	sifive.ovpworld.org/gpio
sifive.ovpworld.org/MSEL	sifive.ovpworld.org/PLIC	sifive.ovpworld.org/PRCI
sifive.ovpworld.org/pwm	sifive.ovpworld.org/spi	sifive.ovpworld.org/teststatus
sifive.ovpworld.org/UART	smsc.ovpworld.org/LAN9118	smsc.ovpworld.org/LAN91C111
ti.ovpworld.org/tca6416a	ti.ovpworld.org/UartInterface	ti.ovpworld.org/ucd9012a
ti.ovpworld.org/ucd9248	vendor.com/fifo	xilinx.ovpworld.org/axi-gpio
xilinx.ovpworld.org/axi-intc	xilinx.ovpworld.org/axi-pcie	xilinx.ovpworld.org/axi-timer
xilinx.ovpworld.org/logicore-fit	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	xilinx.ovpworld.org/zynq_7000-can
xilinx.ovpworld.org/zynq_7000-ddrc	xilinx.ovpworld.org/zynq_7000-devcfg	xilinx.ovpworld.org/zynq_7000-dmac
xilinx.ovpworld.org/zynq_7000-gpio	xilinx.ovpworld.org/zynq_7000-iic	xilinx.ovpworld.org/zynq_7000-ocm
xilinx.ovpworld.org/zynq_7000-qos301	xilinx.ovpworld.org/zynq_7000-qspi	xilinx.ovpworld.org/zynq_7000-sdio
xilinx.ovpworld.org/zynq_7000-slcr	xilinx.ovpworld.org/zynq_7000-spi	xilinx.ovpworld.org/zynq_7000-swdt
xilinx.ovpworld.org/zynq_7000-ttc	xilinx.ovpworld.org/zynq_7000-tz_GPVsecurity	xilinx.ovpworld.org/zynq_7000-tz_security
xilinx.ovpworld.org/zynq_7000-usb	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	andes.ovpworld.org/ATCUART100
andes.ovpworld.org/NCEPLIC100	andes.ovpworld.org/NCEPLMT100	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
cadence.ovpworld.org/gem	cadence.ovpworld.org/uart	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
imperas.ovpworld.org/frameBuffer	imperas.ovpworld.org/uart	imperas.ovpworld.org/usecCounter
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	maxim.ovpworld.org/max673x	microsemi.ovpworld.org/CoreUARTapb
mips.ovpworld.org/16450C	mips.ovpworld.org/MaltaFPGA	mips.ovpworld.org/SmartLoaderLinux
motorola.ovpworld.org/MC146818	national.ovpworld.org/16450	national.ovpworld.org/16550
national.ovpworld.org/16550_4bytes	nxp.ovpworld.org/iMX6_Analog	

5.0 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.

5.1 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.

6.0 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.

7.0 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.

8.0 Parts of peripheral models

8.1 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.

8.2 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.

8.3 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.

8.4 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.

8.5 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.

9.0 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.
#