Imperas Peripheral Model Guide

Model Specific Information for freescale.ovpworld.org / KinetisMPU

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_KinetisMPU.pdf
Created	26 August 2015
Status	OVP Standard Release

Copyright Notice

Right to Copy Documentation

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

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 Licensing

Open Source Apache 2.0

1.2 Location

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

2.0 Net Ports

This model has the following net ports:

Table 1. Net Ports

Name	Type	Must Be Connected	Description
Reset	input	F (False)	

3.0 Bus Slave Ports

This model has the following bus slave ports:

3.1 Bus Slave Port: bport1

Table 2. Bus Slave Port: bport1

Name	Size (bytes)	Must Be Connected	Description
bport1	0x1000	F (False)	

Table 3. Bus Slave Port: bport1 Registers:

Name	Offset	Width (bits)	Description	R/W	is Volatile
ab_CESR	0x0	32	Control/Error Status Register, offset: 0x0		
ab_EAR0	0x10	32	Error Address Register, Slave Port n, array offset: 0x10, array step: 0x8		
ab_EDR0	0x14	32	Error Detail Register, Slave Port n, array offset: 0x14, array step: 0x8		
ab_EAR1	0x18	32	Error Address Register, Slave Port n, array offset: 0x10, array step: 0x8		
ab_EDR1	0x1c	32	Error Detail Register, Slave Port n, array offset: 0x14, array step: 0x8		

ab_EAR2	0x20	32	Error Address Register, Slave Port n, array offset: 0x10, array step: 0x8		
ab_EDR2	0x24	32	Error Detail Register, Slave Port n, array offset: 0x14, array step: 0x8		
ab_EAR3	0x28	32	Error Address Register, Slave Port n, array offset: 0x10, array step: 0x8		
ab_EDR3	0x2c	32	Error Detail Register, Slave Port n, array offset: 0x14, array step: 0x8		
ab_EAR4	0x30	32	Error Address Register, Slave Port n, array offset: 0x10, array step: 0x8		
ab_EDR4	0x34	32	Error Detail Register, Slave Port n, array offset: 0x14, array step: 0x8		
ab_RGD0_WORD0	0x400	32	Region Descriptor 0, Word 0, offset: 0x400		
ab_RGD0_WORD1	0x404	32	Region Descriptor 0, Word 1, offset: 0x404		
ab_RGD0_WORD2	0x408	32	Region Descriptor 0, Word 2, offset: 0x408		
ab_RGD0_WORD3	0x40c	32	Region Descriptor 0, Word 3, offset: 0x40c		
ab_RGD1_WORD0	0x410	32	Region Descriptor 1, Word 0, offset: 0x410		
ab_RGD1_WORD1	0x414	32	Region Descriptor 1, Word 1, offset: 0x414		
ab_RGD1_WORD2	0x418	32	Region Descriptor 1, Word 2, offset: 0x418		
ab_RGD1_WORD3	0x41c	32	Region Descriptor 1, Word 3, offset: 0x41c		
ab_RGD2_WORD0	0x420	32	Region Descriptor 2, Word 0, offset: 0x420		
ab_RGD2_WORD1	0x424	32	Region Descriptor 2, Word 1, offset: 0x424		
ab_RGD2_WORD2	0x428	32	Region Descriptor 2, Word 2, offset: 0x428		
ab_RGD2_WORD3	0x42c	32	Region Descriptor 2, Word 3, offset: 0x42c		
ab_RGD3_WORD0	0x430	32	Region Descriptor 3, Word 0, offset: 0x430		
ab_RGD3_WORD1	0x434	32	Region Descriptor 3, Word 1, offset: 0x434		
ab_RGD3_WORD2	0x438	32	Region Descriptor 3, Word 2, offset: 0x438		
ab_RGD3_WORD3	0x43c	32	Region Descriptor 3, Word 3, offset: 0x43c		
ab_RGD4_WORD0	0x440	32	Region Descriptor 4, Word 0, offset: 0x440		
ab_RGD4_WORD1	0x444	32	Region Descriptor 4, Word 1, offset: 0x444		
ab_RGD4_WORD2	0x448	32	Region Descriptor 4, Word 2, offset: 0x448		

ab_RGD4_WORD3	0x44c	32	Region Descriptor 4, Word 3, offset: 0x44c		
ab_RGD5_WORD0	0x450	32	Region Descriptor 5, Word 0, offset: 0x450		
ab_RGD5_WORD1	0x454	32	Region Descriptor 5, Word 1, offset: 0x454		
ab_RGD5_WORD2	0x458	32	Region Descriptor 5, Word 2, offset: 0x458		
ab_RGD5_WORD3	0x45c	32	Region Descriptor 5, Word 3, offset: 0x45c		
ab_RGD6_WORD0	0x460	32	Region Descriptor 6, Word 0, offset: 0x460		
ab_RGD6_WORD1	0x464	32	Region Descriptor 6, Word 1, offset: 0x464		
ab_RGD6_WORD2	0x468	32	Region Descriptor 6, Word 2, offset: 0x468		
ab_RGD6_WORD3	0x46c	32	Region Descriptor 6, Word 3, offset: 0x46c		
ab_RGD7_WORD0	0x470	32	Region Descriptor 7, Word 0, offset: 0x470		
ab_RGD7_WORD1	0x474	32	Region Descriptor 7, Word 1, offset: 0x474		
ab_RGD7_WORD2	0x478	32	Region Descriptor 7, Word 2, offset: 0x478		
ab_RGD7_WORD3	0x47c	32	Region Descriptor 7, Word 3, offset: 0x47c		
ab_RGD8_WORD0	0x480	32	Region Descriptor 8, Word 0, offset: 0x480		
ab_RGD8_WORD1	0x484	32	Region Descriptor 8, Word 1, offset: 0x484		
ab_RGD8_WORD2	0x488	32	Region Descriptor 8, Word 2, offset: 0x488		
ab_RGD8_WORD3	0x48c	32	Region Descriptor 8, Word 3, offset: 0x48c		
ab_RGD9_WORD0	0x490	32	Region Descriptor 9, Word 0, offset: 0x490		
ab_RGD9_WORD1	0x494	32	Region Descriptor 9, Word 1, offset: 0x494		
ab_RGD9_WORD2	0x498	32	Region Descriptor 9, Word 2, offset: 0x498		
ab_RGD9_WORD3	0x49c	32	Region Descriptor 9, Word 3, offset: 0x49c		
ab_RGD10_WORD0	0x4a0	32	Region Descriptor 10, Word 0, offset: 0x4a0		
ab_RGD10_WORD1	0x4a4	32	Region Descriptor 10, Word 1, offset: 0x4a4		
ab_RGD10_WORD2	0x4a8	32	Region Descriptor 10, Word 2, offset: 0x4a8		
ab_RGD10_WORD3	0x4ac	32	Region Descriptor 10, Word 3, offset: 0x4ac		
ab_RGD11_WORD0	0x4b0	32	Region Descriptor 11, Word 0, offset: 0x4b0		
ab_RGD11_WORD1	0x4b4	32	Region Descriptor 11, Word 1, offset: 0x4b4		
ab_RGD11_WORD2	0x4b8	32	Region Descriptor 11,		

			Word 2, offset: 0x4b8		
ab_RGD11_WORD3	0x4bc	32	Region Descriptor 11, Word 3, offset: 0x4bc		
ab_RGD12_WORD0	0x4c0	32	Region Descriptor 12, Word 0, offset: 0x4c0		
ab_RGD12_WORD1	0x4c4	32	Region Descriptor 12, Word 1, offset: 0x4c4		
ab_RGD12_WORD2	0x4c8	32	Region Descriptor 12, Word 2, offset: 0x4c8		
ab_RGD12_WORD3	0x4cc	32	Region Descriptor 12, Word 3, offset: 0x4cc		
ab_RGD13_WORD0	0x4d0	32	Region Descriptor 13, Word 0, offset: 0x4d0		
ab_RGD13_WORD1	0x4d4	32	Region Descriptor 13, Word 1, offset: 0x4d4		
ab_RGD13_WORD2	0x4d8	32	Region Descriptor 13, Word 2, offset: 0x4d8		
ab_RGD13_WORD3	0x4dc	32	Region Descriptor 13, Word 3, offset: 0x4dc		
ab_RGD14_WORD0	0x4e0	32	Region Descriptor 14, Word 0, offset: 0x4e0		
ab_RGD14_WORD1	0x4e4	32	Region Descriptor 14, Word 1, offset: 0x4e4		
ab_RGD14_WORD2	0x4e8	32	Region Descriptor 14, Word 2, offset: 0x4e8		
ab_RGD14_WORD3	0x4ec	32	Region Descriptor 14, Word 3, offset: 0x4ec		
ab_RGD15_WORD0	0x4f0	32	Region Descriptor 15, Word 0, offset: 0x4f0		
ab_RGD15_WORD1	0x4f4	32	Region Descriptor 15, Word 1, offset: 0x4f4		
ab_RGD15_WORD2	0x4f8	32	Region Descriptor 15, Word 2, offset: 0x4f8		
ab_RGD15_WORD3	0x4fc	32	Region Descriptor 15, Word 3, offset: 0x4fc		
ab_RGDAAC0	0x800	32	Region Descriptor Alternate Access Control 0, offset: 0x800		
ab_RGDAAC1	0x804	32	Region Descriptor Alternate Access Control 1, offset: 0x804		
ab_RGDAAC2	0x808	32	Region Descriptor Alternate Access Control 2, offset: 0x808		
ab_RGDAAC3	0x80c	32	Region Descriptor Alternate Access Control 3, offset: 0x80c		
ab_RGDAAC4	0x810	32	Region Descriptor Alternate Access Control 4, offset: 0x810		
ab_RGDAAC5	0x814	32	Region Descriptor Alternate Access Control 5, offset: 0x814		
ab_RGDAAC6	0x818	32	Region Descriptor Alternate Access Control 6, offset: 0x818		

ab_RGDAAC7	0x81c	32	Region Descriptor Alternate Access Control 7, offset: 0x81c		
ab_RGDAAC8	0x820	32	Region Descriptor Alternate Access Control 8, offset: 0x820		
ab_RGDAAC9	0x824	32	Region Descriptor Alternate Access Control 9, offset: 0x824		
ab_RGDAAC10	0x828	32	Region Descriptor Alternate Access Control 10, offset: 0x828		
ab_RGDAAC11	0x82c	32	Region Descriptor Alternate Access Control 11, offset: 0x82c		
ab_RGDAAC12	0x830	32	Region Descriptor Alternate Access Control 12, offset: 0x830		
ab_RGDAAC13	0x834	32	Region Descriptor Alternate Access Control 13, offset: 0x834		
ab_RGDAAC14	0x838	32	Region Descriptor Alternate Access Control 14, offset: 0x838		
ab_RGDAAC15	0x83c	32	Region Descriptor Alternate Access Control 15, offset: 0x83c		

4.0 Peripheral components in the library

Table 4. Publicly available Imperas/OVP peripheral models (158 models)

Peripheral	Peripheral	Peripheral
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	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	

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.

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

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.
#