Imperas Peripheral Model Guide

Model Specific Information for freescale.ovpworld.org / KinetisDDR

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_KinetisDDR.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 KinetisDDR peripheral model is located in an Imperas/OVP installation at the VLNV: freescale.ovpworld.org / peripheral / KinetisDDR / 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_CR00	0x0	32	DDR Control Register 0, offset: 0x0		
ab_CR01	0x4	32	DDR Control Register 1, offset: 0x4		
ab_CR02	0x8	32	DDR Control Register 2, offset: 0x8		
ab_CR03	0xc	32	DDR Control Register 3, offset: 0xC		
ab_CR04	0x10	32	DDR Control Register 4, offset: 0x10		
ab_CR05	0x14	32	DDR Control Register 5, offset: 0x14		
ab_CR06	0x18	32	DDR Control Register 6, offset: 0x18		

ab_CR07	0x1c	32	DDR Control Register 7, offset: 0x1C		
ab_CR08	0x20	32	DDR Control Register 8, offset: 0x20		
ab_CR09	0x24	32	DDR Control Register 9, offset: 0x24		
ab_CR10	0x28	32	DDR Control Register 10, offset: 0x28		
ab_CR11	0x2c	32	DDR Control Register 11, offset: 0x2C		
ab_CR12	0x30	32	DDR Control Register 12, offset: 0x30		
ab_CR13	0x34	32	DDR Control Register 13, offset: 0x34		
ab_CR14	0x38	32	DDR Control Register 14, offset: 0x38		
ab_CR15	0x3c	32	DDR Control Register 15, offset: 0x3C		
ab_CR16	0x40	32	DDR Control Register 16, offset: 0x40		
ab_CR17	0x44	32	DDR Control Register 17, offset: 0x44		
ab_CR18	0x48	32	DDR Control Register 18, offset: 0x48		
ab_CR19	0x4c	32	DDR Control Register 19, offset: 0x4C		
ab_CR20	0x50	32	DDR Control Register 20, offset: 0x50		
ab_CR21	0x54	32	DDR Control Register 21, offset: 0x54		
ab_CR22	0x58	32	DDR Control Register 22, offset: 0x58		
ab_CR23	0x5c	32	DDR Control Register 23, offset: 0x5C		
ab_CR24	0x60	32	DDR Control Register 24, offset: 0x60		
ab_CR25	0x64	32	DDR Control Register 25, offset: 0x64		
ab_CR26	0x68	32	DDR Control Register 26, offset: 0x68		
ab_CR27	0x6c	32	DDR Control Register 27, offset: 0x6C		
ab_CR28	0x70	32	DDR Control Register 28, offset: 0x70		
ab_CR29	0x74	32	DDR Control Register 29, offset: 0x74		
ab_CR30	0x78	32	DDR Control Register 30, offset: 0x78		
ab_CR31	0x7c	32	DDR Control Register 31, offset: 0x7C		
ab_CR32	0x80	32	DDR Control Register 32, offset: 0x80		
ab_CR33	0x84	32	DDR Control Register 33, offset: 0x84		
ab_CR34	0x88	32	DDR Control Register 34,		

			offset: 0x88		
ab_CR35	0x8c	32	DDR Control Register 35, offset: 0x8C		
ab_CR36	0x90	32	DDR Control Register 36, offset: 0x90		
ab_CR37	0x94	32	DDR Control Register 37, offset: 0x94		
ab_CR38	0x98	32	DDR Control Register 38, offset: 0x98		
ab_CR39	0x9c	32	DDR Control Register 39, offset: 0x9C		
ab_CR40	0xa0	32	DDR Control Register 40, offset: 0xA0		
ab_CR41	0xa4	32	DDR Control Register 41, offset: 0xA4		
ab_CR42	0xa8	32	DDR Control Register 42, offset: 0xA8		
ab_CR43	0xac	32	DDR Control Register 43, offset: 0xAC		
ab_CR44	0xb0	32	DDR Control Register 44, offset: 0xB0		
ab_CR45	0xb4	32	DDR Control Register 45, offset: 0xB4		
ab_CR46	0xb8	32	DDR Control Register 46, offset: 0xB8		
ab_CR47	0xbc	32	DDR Control Register 47, offset: 0xBC		
ab_CR48	0xc0	32	DDR Control Register 48, offset: 0xC0		
ab_CR49	0xc4	32	DDR Control Register 49, offset: 0xC4		
ab_CR50	0xc8	32	DDR Control Register 50, offset: 0xC8		
ab_CR51	0xcc	32	DDR Control Register 51, offset: 0xCC		
ab_CR52	0xd0	32	DDR Control Register 52, offset: 0xD0		
ab_CR53	0xd4	32	DDR Control Register 53, offset: 0xD4		
ab_CR54	0xd8	32	DDR Control Register 54, offset: 0xD8		
ab_CR55	0xdc	32	DDR Control Register 55, offset: 0xDC		
ab_CR56	0xe0	32	DDR Control Register 56, offset: 0xE0		
ab_CR57	0xe4	32	DDR Control Register 57, offset: 0xE4		
ab_CR58	0xe8	32	DDR Control Register 58, offset: 0xE8		
ab_CR59	0xec	32	DDR Control Register 59, offset: 0xEC		
ab_CR60	0xf0	32	DDR Control Register 60, offset: 0xF0		
ab_CR61	0xf4	32	DDR Control Register 61, offset: 0xF4		

ab_CR62	0xf8	32	DDR Control Register 62, offset: 0xF8		
ab_CR63	0xfc	32	DDR Control Register 63, offset: 0xFC		
ab_RCR	0x180	32	RCR Control Register, offset: 0x180		

4.0 Peripheral components in the library

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

Peripheral	Peripheral	Peripheral
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
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	

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.
#