



## Imperas Peripheral Model Guide

### Model Specific Information for [freescale.ovpworld.org](http://freescale.ovpworld.org) / KinetisDAC

#### Imperas Software Limited

Imperas Buildings, North Weston  
Thame, Oxfordshire, OX9 2HA, U.K.  
[docs@imperas.com](mailto:docs@imperas.com).



Author	Imperas Software Limited
Version	20150901.0
Filename	OVP_Peripheral_Specific_Information_KinetisDAC.pdf
Created	26 August 2015
Status	OVP Standard Release

## Copyright Notice

Copyright 2015 Imperas Software Limited. All rights reserved. This software and documentation contain information that is the property of Imperas Software Limited. The software and documentation are furnished under a license agreement and may be used or copied only in accordance with the terms of the license agreement. No part of the software and documentation may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without prior written permission of Imperas Software Limited, or as expressly provided by the license agreement.

## 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](http://OVPworld.org).

## Table Of Contents

<b>1.0 Model Specific Information</b>	4
1.1 Licensing	4
1.2 Location	4
<b>2.0 Net Ports</b>	4
<b>3.0 Bus Slave Ports</b>	4
3.1 Bus Slave Port: bport1	4
<b>4.0 Peripheral components in the library</b>	7
<b>5.0 General Information on Peripheral Models</b>	9
5.1 Background	9
<b>6.0 Building peripherals easily with Imperas iGen</b>	9
<b>7.0 Peripheral model internals</b>	9
<b>8.0 Parts of peripheral models</b>	10
8.1 Configuring the Peripheral Instance with Parameters	10
8.2 Net Ports	10
8.3 Bus master ports	10
8.4 Bus slave ports	10
8.5 Packetnets	10
<b>9.0 More information (documentation) on peripheral models and modeling</b>	10

## 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 KinetisDAC peripheral model is located in an Imperas/OVP installation at the VLNV: [freescale.ovpworld.org / peripheral / KinetisDAC / 1.0](http://freescale.ovpworld.org/peripheral/KinetisDAC/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_DAT0L	0x0	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT0H	0x1	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT1L	0x2	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT1H	0x3	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT2L	0x4	8	DAC Data Low Register, array offset: 0x0, array		

			step: 0x2		
ab_DAT2H	0x5	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT3L	0x6	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT3H	0x7	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT4L	0x8	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT4H	0x9	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT5L	0xa	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT5H	0xb	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT6L	0xc	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT6H	0xd	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT7L	0xe	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT7H	0xf	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT8L	0x10	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT8H	0x11	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT9L	0x12	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT9H	0x13	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT10L	0x14	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT10H	0x15	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT11L	0x16	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT11H	0x17	8	DAC Data High Register, array offset: 0x1, array step: 0x2		

ab_DAT12L	0x18	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT12H	0x19	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT13L	0x1a	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT13H	0x1b	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT14L	0x1c	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT14H	0x1d	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_DAT15L	0x1e	8	DAC Data Low Register, array offset: 0x0, array step: 0x2		
ab_DAT15H	0x1f	8	DAC Data High Register, array offset: 0x1, array step: 0x2		
ab_SR	0x20	8	DAC Status Register, offset: 0x20		
ab_C0	0x21	8	DAC Control Register, offset: 0x21		
ab_C1	0x22	8	DAC Control Register 1, offset: 0x22		
ab_C2	0x23	8	DAC Control Register 2, offset: 0x23		

## 4.0 Peripheral components in the library

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

Peripheral	Peripheral	Peripheral
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	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	smcsc.ovpworld.org/LAN9118
smcsc.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	



## 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](http://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](http://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](http://OVPworld.org/documentation).

#