Imperas Peripheral Model Guide

Model Specific Information for freescale.ovpworld.org / KinetisDMAMUX

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_KinetisDMAMUX.pdf
Created	26 August 2015
Status	OVP Standard Release

Copyright Notice

Right to Copy Documentation

Destination Control Statement

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

2.0 Peripheral Instance Parameters

This model accepts the following parameters:

Table 1. Peripheral Parameters

Name	Type	Description
startDMAChannel	uns32	Starting channel number, 0, 16, 32.. (default 0)

3.0 Net Ports

This model has the following net ports:

Table 2. Net Ports

Name	Type	Must Be Connected	Description
Reset	input	F (False)	
eDMARequest	output	F (False)	
dmaSrc0	input	F (False)	
dmaSrc1	input	F (False)	
dmaSrc2	input	F (False)	
dmaSrc3	input	F (False)	
dmaSrc4	input	F (False)	
dmaSrc5	input	F (False)	
dmaSrc6	input	F (False)	
dmaSrc7	input	F (False)	
dmaSrc8	input	F (False)	
dmaSrc9	input	F (False)	
dmaSrc10	input	F (False)	
dmaSrc11	input	F (False)	
dmaSrc12	input	F (False)	
dmaSrc13	input	F (False)	
dmaSrc14	input	F (False)	
dmaSrc15	input	F (False)	

dmaSrc16	input	F (False)	
dmaSrc17	input	F (False)	
dmaSrc18	input	F (False)	
dmaSrc19	input	F (False)	
dmaSrc20	input	F (False)	
dmaSrc21	input	F (False)	
dmaSrc22	input	F (False)	
dmaSrc23	input	F (False)	
dmaSrc24	input	F (False)	
dmaSrc25	input	F (False)	
dmaSrc26	input	F (False)	
dmaSrc27	input	F (False)	
dmaSrc28	input	F (False)	
dmaSrc29	input	F (False)	
dmaSrc30	input	F (False)	
dmaSrc31	input	F (False)	
dmaSrc32	input	F (False)	
dmaSrc33	input	F (False)	
dmaSrc34	input	F (False)	
dmaSrc35	input	F (False)	
dmaSrc36	input	F (False)	
dmaSrc37	input	F (False)	
dmaSrc38	input	F (False)	
dmaSrc39	input	F (False)	
dmaSrc40	input	F (False)	
dmaSrc41	input	F (False)	
dmaSrc42	input	F (False)	
dmaSrc43	input	F (False)	
dmaSrc44	input	F (False)	
dmaSrc45	input	F (False)	
dmaSrc46	input	F (False)	
dmaSrc47	input	F (False)	
dmaSrc48	input	F (False)	
dmaSrc49	input	F (False)	
dmaSrc50	input	F (False)	
dmaSrc51	input	F (False)	
dmaSrc52	input	F (False)	
dmaSrc53	input	F (False)	
dmaSrc54	input	F (False)	
dmaSrc55	input	F (False)	
dmaSrc56	input	F (False)	
dmaSrc57	input	F (False)	
dmaSrc58	input	F (False)	
dmaSrc59	input	F (False)	
dmaSrc60	input	F (False)	
dmaSrc61	input	F (False)	
dmaSrc62	input	F (False)	
dmaSrc63	input	F (False)	

trg1	input	F (False)	
trg2	input	F (False)	
trg3	input	F (False)	
trg4	input	F (False)	

4.0 Bus Slave Ports

This model has the following bus slave ports:

4.1 Bus Slave Port: bport1

Table 3. Bus Slave Port: bport1

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

Table 4. Bus Slave Port: bport1 Registers:

Name	Offset	Width (bits)	Description	R/W	is Volatile
ab_CHCFG0	0x0	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG1	0x1	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG2	0x2	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG3	0x3	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG4	0x4	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG5	0x5	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG6	0x6	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG7	0x7	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG8	0x8	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG9	0x9	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG10	0xa	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG11	0xb	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG12	0xc	8	Channel Configuration Register, array offset:		

			0x0, array step: 0x1		
ab_CHCFG13	0xd	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG14	0xe	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		
ab_CHCFG15	0xf	8	Channel Configuration Register, array offset: 0x0, array step: 0x1		

5.0 Peripheral components in the library

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

Peripheral	Peripheral	Peripheral
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	freescale.ovpworld.org/KinetisDMA
freescale.ovpworld.org/KinetisDMAC	freescale.ovpworld.org/KinetisDMAMUX	

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.
#