Imperas Peripheral Model Guide

Model Specific Information for freescale.ovpworld.org / KinetisENET

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_KinetisENET.pdf
Created	31 December 2021
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 Description

Model of the ENET peripheral used on the Freescale Kinetis platform

1.2 Limitations

Provides the base behaviour for the OVP Freescale Kinetis platforms

1.3 Reference

www.freescale.com/Kinetis

1.4 Licensing

Open Source Apache 2.0

1.5 Location

The KinetisENET peripheral model is located in an Imperas/OVP installation at the VLNV: freescale.ovpworld.org / peripheral / KinetisENET / 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_EIR	0x4	32	Interrupt Event Register, offset: 0x4		

ab_EIMR	0x8	32	Interrupt Mask Register, offset: 0x8		
ab_RDAR	0x10	32	Receive Descriptor Active Register, offset: 0x10		
ab_TDAR	0x14	32	Transmit Descriptor Active Register, offset: 0x14		
ab_ECR	0x24	32	Ethernet Control Register, offset: 0x24		
ab_MMFR	0x40	32	MII Management Frame Register, offset: 0x40		
ab_MSCR	0x44	32	MII Speed Control Register, offset: 0x44		
ab_MIBC	0x64	32	MIB Control Register, offset: 0x64		
ab_RCR	0x84	32	Receive Control Register, offset: 0x84		
ab_TCR	0xc4	32	Transmit Control Register, offset: 0xC4		
ab_PALR	0xe4	32	Physical Address Lower Register, offset: 0xE4		
ab_PAUR	0xe8	32	Physical Address Upper Register, offset: 0xE8		
ab_OPD	0xec	32	Opcode/Pause Duration Register, offset: 0xEC		
ab_IAUR	0x118	32	Descriptor Individual Upper Address Register, offset: 0x118		
ab_IALR	0x11c	32	Descriptor Individual Lower Address Register, offset: 0x11C		
ab_GAUR	0x120	32	Descriptor Group Upper Address Register, offset: 0x120		
ab_GALR	0x124	32	Descriptor Group Lower Address Register, offset: 0x124		
ab_TFWR	0x144	32	Transmit FIFO Watermark Register, offset: 0x144		
ab_RDSR	0x180	32	Receive Descriptor Ring Start Register, offset: 0x180		
ab_TDSR	0x184	32	Transmit Buffer Descriptor Ring Start Register, offset: 0x184		
ab_MRBR	0x188	32	Maximum Receive Buffer Size Register, offset: 0x188		
ab_RSFL	0x190	32	Receive FIFO Section Full Threshold, offset: 0x190		
ab_RSEM	0x194	32	Receive FIFO Section Empty Threshold, offset: 0x194		

ab_RAEM	0x198	32	Receive FIFO Almost Empty Threshold, offset: 0x198		
ab_RAFL	0x19c	32	Receive FIFO Almost Full Threshold, offset: 0x19C		
ab_TSEM	0x1a0	32	Transmit FIFO Section Empty Threshold, offset: 0x1A0		
ab_TAEM	0x1a4	32	Transmit FIFO Almost Empty Threshold, offset: 0x1A4		
ab_TAFL	0x1a8	32	Transmit FIFO Almost Full Threshold, offset: 0x1A8		
ab_TIPG	0x1ac	32	Transmit Inter-Packet Gap, offset: 0x1AC		
ab_FTRL	0x1b0	32	Frame Truncation Length, offset: 0x1B0		
ab_TACC	0x1c0	32	Transmit Accelerator Function Configuration, offset: 0x1C0		
ab_RACC	0x1c4	32	Receive Accelerator Function Configuration, offset: 0x1C4		
ab_RMON_T_DROP	0x200	32	Count of frames not counted correctly (RMON_T_DROP). NOTE: Counter not implemented (read 0 always) as not applicable., offset: 0x200		
ab_RMON_T_PACKETS	0x204	32	RMON Tx packet count (RMON_T_PACKETS), offset: 0x204		
ab_RMON_T_BC_PKT	0x208	32	RMON Tx Broadcast Packets (RMON_T_BC_PKT), offset: 0x208		
ab_RMON_T_MC_PKT	0x20c	32	RMON Tx Multicast Packets (RMON_T_MC_PKT), offset: 0x20C		
ab_RMON_T_CRC_ALIGN	0x210	32	RMON Tx Packets w CRC/Align error (RMON_T_CRC_ALIGN), offset: 0x210		
ab_RMON_T_UNDERSIZE	0x214	32	RMON Tx Packets < 64 bytes, good CRC (RMON_T_UNDERSIZE), offset: 0x214		
ab_RMON_T_OVERSIZE	0x218	32	RMON Tx Packets > MAX_FL bytes, good CRC (RMON_T_OVERSIZE), offset: 0x218		
ab_RMON_T_FRAG	0x21c	32	RMON Tx Packets < 64 bytes, bad CRC (RMON_T_FRAG),		

			offset: 0x21C		
ab_RMON_T_JAB	0x220	32	RMON Tx Packets > MAX_FL bytes, bad CRC (RMON_T_JAB), offset: 0x220		
ab_RMON_T_COL	0x224	32	RMON Tx collision count (RMON_T_COL), offset: 0x224		
ab_RMON_T_P64	0x228	32	RMON Tx 64 byte packets (RMON_T_P64), offset: 0x228		
ab_RMON_T_P65TO127	0x22c	32	RMON Tx 65 to 127 byte packets (RMON_T_P65TO127), offset: 0x22C		
ab_RMON_T_P128TO255	0x230	32	RMON Tx 128 to 255 byte packets (RMON_T_P128TO255), offset: 0x230		
ab_RMON_T_P256TO511	0x234	32	RMON Tx 256 to 511 byte packets (RMON_T_P256TO511), offset: 0x234		
ab_RMON_T_P512TO1023	0x238	32	RMON Tx 512 to 1023 byte packets (RMON_T_P512TO1023), offset: 0x238		
ab_RMON_T_P1024TO2047	0x23c	32	RMON Tx 1024 to 2047 byte packets (RMON_T_P1024TO2047), offset: 0x23C		
ab_RMON_T_P_GTE2048	0x240	32	RMON Tx packets w > 2048 bytes (RMON_T_P_GTE2048), offset: 0x240		
ab_RMON_T_OCTETS	0x244	32	RMON Tx Octets (RMON_T_OCTETS), offset: 0x244		
ab_IEEE_T_DROP	0x248	32	Count of frames not counted correctly (IEEE_T_DROP). NOTE: Counter not implemented (read 0 always) as not applicable., offset: 0x248		
ab_IEEE_T_FRAME_OK	0x24c	32	Frames Transmitted OK (IEEE_T_FRAME_OK), offset: 0x24C		
ab_IEEE_T_1COL	0x250	32	Frames Transmitted with Single Collision (IEEE_T_1COL), offset: 0x250		
ab_IEEE_T_MCOL	0x254	32	Frames Transmitted with Multiple Collisions (IEEE_T_MCOL), offset: 0x254		
ab_IEEE_T_DEF	0x258	32	Frames Transmitted after Deferral Delay (IEEE_T_DEF), offset: 0x258		

ab_IEEE_T_LCOL	0x25c	32	Frames Transmitted with Late Collision (IEEE_T_LCOL), offset: 0x25C		
ab_IEEE_T_EXCOL	0x260	32	Frames Transmitted with Excessive Collisions (IEEE_T_EXCOL), offset: 0x260		
ab_IEEE_T_MACERR	0x264	32	Frames Transmitted with Tx FIFO Underrun (IEEE_T_MACERR), offset: 0x264		
ab_IEEE_T_CSERR	0x268	32	Frames Transmitted with Carrier Sense Error (IEEE_T_CSERR), offset: 0x268		
ab_IEEE_T_SQE	0x26c	32	Frames Transmitted with SQE Error (IEEE_T_SQE). NOTE: Counter not implemented (read 0 always) as no SQE information is available., offset: 0x26C		
ab_IEEE_T_FDXFC	0x270	32	Flow Control Pause frames transmitted (IEEE_T_FDXFC), offset: 0x270		
ab_IEEE_T_OCTETS_OK	0x274	32	Octet count for Frames Transmitted w/o Error (IEEE_T_OCTETS_OK). NOTE: Counts total octets (includes header and FCS fields)., offset: 0x274		
ab_RMON_R_PACKETS	0x284	32	RMON Rx packet count (RMON_R_PACKETS), offset: 0x284		
ab_RMON_R_BC_PKT	0x288	32	RMON Rx Broadcast Packets (RMON_R_BC_PKT), offset: 0x288		
ab_RMON_R_MC_PKT	0x28c	32	RMON Rx Multicast Packets (RMON_R_MC_PKT), offset: 0x28C		
ab_RMON_R_CRC_ALIGN	0x290	32	RMON Rx Packets w CRC/Align error (RMON_R_CRC_ALIGN), offset: 0x290		
ab_RMON_R_UNDERSIZE	0x294	32	RMON Rx Packets < 64 bytes, good CRC (RMON_R_UNDERSIZE), offset: 0x294		
ab_RMON_R_OVERSIZE	0x298	32	RMON Rx Packets > MAX_FL bytes, good CRC (RMON_R_OVERSIZE), offset: 0x298		
ab_RMON_R_FRAG	0x29c	32	RMON Rx Packets < 64 bytes, bad CRC		

			(RMON_R_FRAG), offset: 0x29C		
ab_RMON_R_JAB	0x2a0	32	RMON Rx Packets > MAX_FL bytes, bad CRC (RMON_R_JAB), offset: 0x2A0		
ab_RMON_R_RESVD_0	0x2a4	32	Reserved (RMON_R_RESVD_0), offset: 0x2A4		
ab_RMON_R_P64	0x2a8	32	RMON Rx 64 byte packets (RMON_R_P64), offset: 0x2A8		
ab_RMON_R_P65TO127	0x2ac	32	RMON Rx 65 to 127 byte packets (RMON_R_P65TO127), offset: 0x2AC		
ab_RMON_R_P128TO255	0x2b0	32	RMON Rx 128 to 255 byte packets (RMON_R_P128TO255), offset: 0x2B0		
ab_RMON_R_P256TO511	0x2b4	32	RMON Rx 256 to 511 byte packets (RMON_R_P256TO511), offset: 0x2B4		
ab_RMON_R_P512TO1023	0x2b8	32	RMON Rx 512 to 1023 byte packets (RMON_R_P512TO1023), offset: 0x2B8		
ab_RMON_R_P1024TO2047	0x2bc	32	RMON Rx 1024 to 2047 byte packets (RMON_R_P1024TO2047), offset: 0x2BC		
ab_RMON_R_P_GTE2048	0x2c0	32	RMON Rx packets w > 2048 bytes (RMON_R_P_GTE2048), offset: 0x2C0		
ab_RMON_R_OCTETS	0x2c4	32	RMON Rx Octets (RMON_R_OCTETS), offset: 0x2C4		
ab_RMON_R_DROP	0x2c8	32	Count of frames not counted correctly (IEEE_R_DROP). NOTE: Counter increments if a frame with valid/missing SFD character is detected and has been dropped. None of the other counters increments if this counter increments., offset: 0x2C8		
ab_RMON_R_FRAME_OK	0x2cc	32	Frames Received OK (IEEE_R_FRAME_OK), offset: 0x2CC		
ab_IEEE_R_CRC	0x2d0	32	Frames Received with CRC Error (IEEE_R_CRC), offset: 0x2D0		
ab_IEEE_R_ALIGN	0x2d4	32	Frames Received with		

			Alignment Error (IEEE_R_ALIGN), offset: 0x2D4		
ab_IEEE_R_MACERR	0x2d8	32	Receive Fifo Overflow count (IEEE_R_MACERR), offset: 0x2D8		
ab_IEEE_R_FDXFC	0x2dc	32	Flow Control Pause frames received (IEEE_R_FDXFC), offset: 0x2DC		
ab_IEEE_R_OCTETS_OK	0x2e0	32	Octet count for Frames Rcvd w/o Error (IEEE_R_OCTETS_OK). Counts total octets (includes header and FCS fields)., offset: 0x2E0		
ab_ATCR	0x400	32	Timer Control Register, offset: 0x400		
ab_ATVR	0x404	32	Timer Value Register, offset: 0x404		
ab_ATOFF	0x408	32	Timer Offset Register, offset: 0x408		
ab_ATPER	0x40c	32	Timer Period Register, offset: 0x40C		
ab_ATCOR	0x410	32	Timer Correction Register, offset: 0x410		
ab_ATINC	0x414	32	Time-Stamping Clock Period Register, offset: 0x414		
ab_ATSTMP	0x418	32	Timestamp of Last Transmitted Frame, offset: 0x418		
ab_TGSR	0x604	32	Timer Global Status Register, offset: 0x604		
ab_TCSR0	0x608	32	Timer Control Status Register, array offset: 0x608, array step: 0x8		
ab_TCCR0	0x60c	32	Timer Compare Capture Register, array offset: 0x60C, array step: 0x8		
ab_TCSR1	0x610	32	Timer Control Status Register, array offset: 0x610, array step: 0x8		
ab_TCCR1	0x614	32	Timer Compare Capture Register, array offset: 0x614, array step: 0x8		
ab_TCSR2	0x618	32	Timer Control Status Register, array offset: 0x618, array step: 0x8		
ab_TCCR2	0x61c	32	Timer Compare Capture Register, array offset: 0x61c, array step: 0x8		
ab_TCSR3	0x620	32	Timer Control Status Register, array offset: 0x620, array step: 0x8		
ab_TCCR3	0x624	32	Timer Compare Capture Register, array offset:		

			0x624, array step: 0x8		

4.0 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 4. Publicly available platforms using peripheral 'KinetisENET'

Platform Name	Vendor
FreescaleKinetis60	freescale.ovpworld.org
FreescaleKinetis64	freescale.ovpworld.org

5.0 Peripheral components in the library

Table 5. Publicly available Imperas/OVP peripheral models (227 models)

Peripheral	Peripheral	Peripheral
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	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	

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.

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.
#