Imperas Peripheral Model Guide

Model Specific Information for atmel.ovpworld.org / AdvancedInterruptController

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_AdvancedInterruptController.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 Description

AIC: Advanced Interrupt Controller This model contains an accurate Register set interface. The functionality has only been implemented to sufficiently boot uClinux The Advanced Interrupt Controller has an 8-level priority, individually maskable, vectored interrupt controller, and drives the NIRQ and NFIQ pins of the ARM7TDMI from: The external fast interrupt line (FIQ) The three external interrupt request lines (IRQ0 - IRQ2) The interrupt signals from the on-chip peripherals The AIC is extensively programmable offering maximum flexibility, and its vectoring features reduce the real-time overhead in handling interrupts. The AIC also features a spurious vector detection feature, which reduces spurious interrupt handling to a minimum, and a protect mode that facilitates the debug capabilities.

1.2 Licensing

Open Source Apache 2.0

1.3 Limitations

This model is sufficient to boot Linux

1.4 Reference

Rev. 1354D–ATARM–08/02

1.5 Location

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

2.0 Net Ports

This model has the following net ports:

Table 1. Net Ports

Name	Type	Must Be Connected	Description
NFIQ	output	F (False)	
NIRQ	output	F (False)	
FIQ	input	F (False)	
SWIRQ	input	F (False)	
US0IRQ	input	F (False)	
US1IRQ	input	F (False)	
TC0IRQ	input	F (False)	

TC1IRQ	input	F (False)	
TC2IRQ	input	F (False)	
WDIRQ	input	F (False)	
PIOIRQ	input	F (False)	
IRQ0	input	F (False)	
IRQ1	input	F (False)	
IRQ2	input	F (False)	

3.0 Bus Slave Ports

This model has the following bus slave ports:

3.1 Bus Slave Port: bp1

Table 2. Bus Slave Port: bp1

Name	Size (bytes)	Must Be Connected	Description
bp1	0x1000	T (True)	

Table 3. Bus Slave Port: bp1 Registers:

Name	Offset	Width (bits)	Description	R/W	is Volatile
REG_AIC_SMR0	0x0	32			
REG_AIC_SMR1	0x4	32			
REG_AIC_SMR2	0x8	32			
REG_AIC_SMR3	0xc	32			
REG_AIC_SMR4	0x10	32			
REG_AIC_SMR5	0x14	32			
REG_AIC_SMR6	0x18	32			
REG_AIC_SMR7	0x1c	32			
REG_AIC_SMR8	0x20	32			
REG_AIC_SMR9	0x24	32			
REG_AIC_SMR10	0x28	32			
REG_AIC_SMR11	0x2c	32			
REG_AIC_SMR12	0x30	32			
REG_AIC_SMR13	0x34	32			
REG_AIC_SMR14	0x38	32			
REG_AIC_SMR15	0x3c	32			
REG_AIC_SMR16	0x40	32			
REG_AIC_SMR17	0x44	32			
REG_AIC_SMR18	0x48	32			
REG_AIC_SMR19	0x4c	32			
REG_AIC_SMR20	0x50	32			
REG_AIC_SMR21	0x54	32			
REG_AIC_SMR22	0x58	32			
REG_AIC_SMR23	0x5c	32			
REG_AIC_SMR24	0x60	32			
REG_AIC_SMR25	0x64	32			
REG_AIC_SMR26	0x68	32			
REG_AIC_SMR27	0x6c	32			
REG_AIC_SMR28	0x70	32			

REG_AIC_SMR29	0x74	32			
REG_AIC_SMR30	0x78	32			
REG_AIC_SMR31	0x7c	32			
REG_AIC_SVR0	0x80	32			
REG_AIC_SVR1	0x84	32			
REG_AIC_SVR2	0x88	32			
REG_AIC_SVR3	0x8c	32			
REG_AIC_SVR4	0x90	32			
REG_AIC_SVR5	0x94	32			
REG_AIC_SVR6	0x98	32			
REG_AIC_SVR7	0x9c	32			
REG_AIC_SVR8	0xa0	32			
REG_AIC_SVR9	0xa4	32			
REG_AIC_SVR10	0xa8	32			
REG_AIC_SVR11	0xac	32			
REG_AIC_SVR12	0xb0	32			
REG_AIC_SVR13	0xb4	32			
REG_AIC_SVR14	0xb8	32			
REG_AIC_SVR15	0xbc	32			
REG_AIC_SVR16	0xc0	32			
REG_AIC_SVR17	0xc4	32			
REG_AIC_SVR18	0xc8	32			
REG_AIC_SVR19	0xcc	32			
REG_AIC_SVR20	0xd0	32			
REG_AIC_SVR21	0xd4	32			
REG_AIC_SVR22	0xd8	32			
REG_AIC_SVR23	0xdc	32			
REG_AIC_SVR24	0xe0	32			
REG_AIC_SVR25	0xe4	32			
REG_AIC_SVR26	0xe8	32			
REG_AIC_SVR27	0xec	32			
REG_AIC_SVR28	0xf0	32			
REG_AIC_SVR29	0xf4	32			
REG_AIC_SVR30	0xf8	32			
REG_AIC_SVR31	0xfc	32			
REG_AIC_IVR	0x100	32			
REG_AIC_FVR	0x104	32			
REG_AIC_ISR	0x108	32			
REG_AIC_IPR	0x10c	32			
REG_AIC_IMR	0x110	32			
REG_AIC_CISR	0x114	32			
REG_AIC_IECR	0x120	32			
REG_AIC_IDCR	0x124	32			
REG_AIC_ICCR	0x128	32			
REG_AIC_ISCR	0x12c	32			
REG_AIC_EOICR	0x130	32			
REG_AIC_SPU	0x134	32			

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 'AdvancedInterruptController'

Platform Name	Vendor
AtmelAT91SAM7	atmel.ovpworld.org

5.0 Peripheral components in the library

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

Peripheral	Peripheral	Peripheral
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	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	

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.
#