Imperas Guide to using Virtual Platforms

Platform Specific Information for arm.ovpworld.org / ArmVersatileExpress-CA9

Imperas Software Limited

Imperas Buildings, North Weston
Thame, Oxfordshire, OX9 2HA, U.K.
docs@imperas.com.

Author	Imperas Software Limited
Version	20150901.0
Filename	Imperas_Platform_User_Guide_ArmVersatileExpress-CA9.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 Virtual Platform: ArmVersatileExpress-CA9

This document provides the details of the usage of an Imperas OVP Virtual Platform. The first half of the document covers specifics of this particular virtual platform. For more information about Imperas OVP virtual platforms, how they are built and used, please see the later sections in this document.

1.1 Licensing

Open Source Apache 2.0

1.2 Description

This platform models the ARM Versatile Express development board with a CoreTile Express A9x4 (V2P-CA9) Daughterboard.
See the ARM documents DUI0447G_v2m_p1_trm.pdf and DUI0448G_v2p_ca9_trm.pdf for details of the hardware being modeled.
Note this platform implements the motherboard's 'Legacy' memory map.

The default processor is an ARM Cortex-A9MPx4, which may be changed.

1.3 Limitations

This platform provides the peripherals required to boot and run Operating Systems such as Linux or Android.
Some of the peripherals are register-only, non-functional models. See the individual peripheral model documentation for details.

The TrustZone Protection Controller (TZPC) is modeled, the TrustZone Address Space Controller (TZASPC) is not modeled.

CoreSight software debug and trace ports are not modeled.

1.4 Reference

ARM Development Boards Versatile Express BaseBoard and ARM CoreTile Express A9x4

1.5 Location

The ArmVersatileExpress-CA9 virtual platform is located in an Imperas/OVP installation at the VLNV:
arm.ovpworld.org / platform / ArmVersatileExpress-CA9 / 1.0.

1.6 Platform Simulation Attributes

Table 1. Platform Simulation Attributes

Attribute	Value	Description
stoponctrlc	stoponctrlc	Stop on control-C

2.0 Command Line Control of the Platform

2.1 Built-in Arguments

Table 2. Platform Built-in Arguments

Attribute	Value	Description
allargs	allargs	The Command line parser will accept the complete imperas argument set. Note that this option is ignored in some Imperas products

When running a platform in a Windows or Linux shell several command arguments can be specified.
Typically there is a '-help' command which lists the commands available in the platforms. For example:
    myplatform.exe -help

Some command line arguments require a value to be provided. For example:
    myplatform.exe -program myimagefile.elf

2.2 Platform Specific Command Line Arguments

Table 3. Platform Arguments

Name	Type	Description
zimage	stringvar	Linux zImage file to load using smartLoader
zimageaddr	uns64var	Physical address to load zImage file (default:physicalbase + 0x00010000)
initrd	stringvar	Linux initrd file to load using smartLoader
initrdaddr	uns64var	Physical address to load initrd file (default:physicalbase + 0x00d00000)
linuxsym	stringvar	Linux ELF file with symbolic debug info (CpuManger only)
linuxcmd	stringvar	Linux command line (default: 'mem=1024M raid=noautodetect console=ttyAMA0,38400n8 vmalloc=256MB devtmpfs.mount=0'
boardid	int32var	Value to pass to Linux as the boardid (default (0x8e0)
linuxmem	uns64var	Amount of memory allocated to Linux (required in AMP mode)
boot	stringvar	Boot code object file (If specified, smartLoader will jump to this rather than zImage)
image0	stringvar	Image file to load on cpu0
image0addr	uns64var	load address for image on cpu0 (IMAGE0 must be specified)
image0sym	stringvar	Elf file with symbolic debug info for image on cpu0 (IMAGE0 must be specified, CpuManger only)
image1	stringvar	Image file to load on cpu1 to n
image1addr	uns64var	Load address for image on cpu1 to n (IMAGE1 must be specified)

image1sym	stringvar	Elf file with symbolic debug info for image on cpu1 to n (IMAGE1 must be specified, CpuManger only)
uart0port	stringvar	Uart0 port: 'auto' for automatic console, 0 for simulator chosen port #, or number of specific port
uart1port	stringvar	Uart1 port: 'auto' for automatic console, 0 for simulator chosen port #, or number of specific port
nographics	boolvar	Inhibit opening of the lcd graphics window

3.0 Processor [arm.ovpworld.org/processor/arm/1.0] instance: cpu

3.1 Processor model type: 'arm' variant 'Cortex-A9MPx4' definition

Imperas OVP processor models support multiple variants and details of the variants implemented in this model can be found in:
- the Imperas installation located at ImperasLib/source/arm.ovpworld.org/processor/arm/1.0/doc
- the OVP website: OVP_Model_Specific_Information_arm_Cortex-A9MPx4.pdf

3.1.1 Description

ARM Processor Model

3.1.2 Licensing

Usage of binary model under license governing simulator usage.
Note that for models of ARM CPUs the license includes the following terms:
Licensee is granted a non-exclusive, worldwide, non-transferable, revocable licence to:
If no source is being provided to the Licensee: use and copy only (no modifications rights are granted) the model for the sole purpose of designing, developing, analyzing, debugging, testing, verifying, validating and optimizing software which: (a) (i) is for ARM based systems; and (ii) does not incorporate the ARM Models or any part thereof; and (b) such ARM Models may not be used to emulate an ARM based system to run application software in a production or live environment.
If source code is being provided to the Licensee: use, copy and modify the model for the sole purpose of designing, developing, analyzing, debugging, testing, verifying, validating and optimizing software which: (a) (i) is for ARM based systems; and (ii) does not incorporate the ARM Models or any part thereof; and (b) such ARM Models may not be used to emulate an ARM based system to run application software in a production or live environment.
In the case of any Licensee who is either or both an academic or educational institution the purposes shall be limited to internal use.
Except to the extent that such activity is permitted by applicable law, Licensee shall not reverse engineer, decompile, or disassemble this model. If this model was provided to Licensee in Europe, Licensee shall not reverse engineer, decompile or disassemble the Model for the purposes of error correction.
The License agreement does not entitle Licensee to manufacture in silicon any product based on this model.
The License agreement does not entitle Licensee to use this model for evaluating the validity of any ARM patent.
Source of model available under separate Imperas Software License Agreement.

3.1.3 Limitations

Instruction pipelines are not modeled in any way. All instructions are assumed to complete immediately. This means that instruction barrier instructions (e.g. ISB, CP15ISB) are treated as NOPs, with the exception of any undefined instruction behavior, which is modeled. The model does not implement speculative fetch behavior. The branch cache is not modeled.
Caches and write buffers are not modeled in any way. All loads, fetches and stores complete immediately and in order, and are fully synchronous (as if the memory was of Strongly Ordered or Device-nGnRnE type). Data barrier instructions (e.g. DSB, CP15DSB) are treated as NOPs, with the exception of any undefined instruction behavior, which is modeled. Cache manipulation instructions are implemented as NOPs, with the exception of any undefined instruction behavior, which is modeled.
Real-world timing effects are not modeled: all instructions are assumed to complete in a single cycle.
Performance Monitors are implemented as a register interface only.
TLBs are architecturally-accurate but not device accurate. This means that all TLB maintenance and address translation operations are fully implemented but the cache is larger than in the real device.

3.1.4 Verification

Models have been extensively tested by Imperas. ARM Cortex-A models have been successfully used by customers to simulate SMP Linux, Ubuntu Desktop, VxWorks and ThreadX on Xilinx Zynq virtual platforms.

3.1.5 Features

Thumb-2 instructions are supported.
Trivial Jazelle extension is implemented.
SIMD instructions are implemented.
NEON is implemented.
VFP is implemented.
Security extensions are implemented (also known as TrustZone). Non-secure accesses can be made visible externally by connecting the processor to a 41-bit physical bus, in which case bits 39..0 give the true physical address and bit 40 is the NS bit.
VMSA secure and non-secure address translation is implemented.
GIC block is implemented (GICv1, including security extensions). Accesses to GIC registers can be viewed externally by connecting to the 32-bit GICRegisters bus port. Secure register accesses are at offset 0x0 on this bus; for example, a secure access to GIC register ICDDCR can be observed by monitoring address 0x00001000. Non-secure accesses are at offset 0x80000000 on this bus; for example, a non-secure access to GIC register ICDDCR can be observed by monitoring address 0x80001000

3.2 Instance Parameters

Several parameters can be specified when a processor is instanced in a platform. For this processor instance 'cpu' it has been instanced with the following parameters:

Table 4. Processor Instance 'cpu' Parameters (Configurations)

Parameter	Value	Description
endian	little	Select processor endian (big or little)
simulateexceptions	simulateexceptions	Causes the processor simulate exceptions instead of

		halting
mips	448.0	The nominal MIPS for the processor

Table 5. Processor Instance 'cpu' Parameters (Attributes)

Parameter Name	Value	Type
variant	Cortex-A9MPx4	
compatibility	ISA	
UAL	1	
showHiddenRegs	0	
override_CBAR	0x1e000000	

3.3 Memory Map for processor 'cpu' bus: 'tzBus'

Processor instance 'cpu' is connected to bus 'tzBus' using master port 'INSTRUCTION'.
Processor instance 'cpu' is connected to bus 'tzBus' using master port 'DATA'.

Table 6. Memory Map ('cpu' / 'tzBus' [width: 41])

Lo Address	Hi Address	Instance	Component
0x0	0xFFFFFFFF	secure	bridge
remappable	remappable	dmc1NS	DynamicBridge
remappable	remappable	faxi1NS	DynamicBridge
remappable	remappable	lcd1NS	DynamicBridge
remappable	remappable	saxi1NS	DynamicBridge
remappable	remappable	scc1NS	DynamicBridge
remappable	remappable	smc1NS	DynamicBridge
remappable	remappable	timer45NS	DynamicBridge
remappable	remappable	wdt2NS	DynamicBridge
0x10000000000	0x10003FFFFFF	ddr2RemapBridgeNS	bridge
0x10010000000	0x10010000FFF	sysRegsNS	bridge
0x10010001000	0x10010001FFF	sysCtrlNS	bridge
0x10010004000	0x10010004FFF	aac1NS	bridge
0x10010005000	0x10010005FFF	mmc1NS	bridge
0x10010006000	0x10010006FFF	kb1NS	bridge
0x10010007000	0x10010007FFF	ms1NS	bridge
0x10010009000	0x10010009FFF	uart0NS	bridge
0x1001000A000	0x1001000AFFF	uart1NS	bridge
0x1001000B000	0x1001000BFFF	uart2NS	bridge
0x1001000C000	0x1001000CFFF	uart3NS	bridge
0x1001000F000	0x1001000FFFF	wdt1NS	bridge
0x10010011000	0x10010011FFF	timer01NS	bridge
0x10010012000	0x10010012FFF	timer02NS	bridge
0x10010016000	0x10010016FFF	dvi1NS	bridge
0x10010017000	0x10010017FFF	rtc1NS	bridge
0x1001001A000	0x1001001AFFF	cf1NS	bridge
0x1001001F000	0x1001001FFFF	lcd2NS	bridge
0x1001E00A000	0x1001E00AFFF	l2regsNS	bridge
0x10040000000	0x10043FFFFFF	nor0NS	bridge
0x10044000000	0x10047FFFFFF	nor1NS	bridge
0x10048000000	0x1004BFFFFFF	sram1NS	bridge

0x1004C000000	0x1004C7FFFFF	vram1NS	bridge
0x1004E000000	0x1004E000FFF	eth0NS	bridge
0x1004F000000	0x1004F00FFFF	usb0NS	bridge
0x10060000000	0x1009FFFFFFF	ddr2ramNS	bridge

Table 7. Bridged Memory Map ('cpu' / 'secure' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
0x0	0x3FFFFFF	ddr2RemapBridge	bridge
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10000000	0x10000FFF	sysRegs	VexpressSysRegs
0x10001000	0x10001FFF	sysCtrl	SysCtrlSP810
0x10002000	0x10002FFF	sbpci0	dummyPort
0x10004000	0x10004FFF	aac1	AaciPL041
0x10005000	0x10005FFF	mmc1	MmciPL181
0x10006000	0x10006FFF	kb1	KbPL050
0x10007000	0x10007FFF	ms1	KbPL050
0x10009000	0x10009FFF	uart0	UartPL011
0x1000A000	0x1000AFFF	uart1	UartPL011
0x1000B000	0x1000BFFF	uart2	UartPL011
0x1000C000	0x1000CFFF	uart3	UartPL011
0x1000F000	0x1000FFFF	wdt1	WdtSP805
0x10011000	0x10011FFF	timer01	TimerSP804
0x10012000	0x10012FFF	timer23	TimerSP804
0x10016000	0x10016FFF	dvi1	SerBusDviRegs
0x10017000	0x10017FFF	rtc1	RtcPL031
0x1001A000	0x1001AFFF	cf1	CompactFlashRegs
0x1001F000	0x1001FFFF	clcd	LcdPL110
0x10020000	0x10020FFF	lcd1	LcdPL110
0x100E0000	0x100E0FFF	dmc1	DMemCtrlPL341
0x100E1000	0x100E1FFF	smc1	SMemCtrlPL354
0x100E2000	0x100E2FFF	scc1	dummyPort
0x100E4000	0x100E4FFF	timer45	TimerSP804
0x100E5000	0x100E5FFF	wdt2	WdtSP805
0x100E6000	0x100E6FFF	tzpc	TzpcBP147
0x100E8000	0x100E8FFF	gpio0	dummyPort
0x100E9000	0x100E9FFF	faxi1	dummyPort
0x100EA000	0x100EAFFF	saxi1	dummyPort
0x1E00A000	0x1E00AFFF	l2regs	L2CachePL310
0x40000000	0x43FFFFFF	nor0	NorFlash48F4400
0x44000000	0x47FFFFFF	nor1	NorFlash48F4400
0x48000000	0x4BFFFFFF	sram1	ram
0x4C000000	0x4C7FFFFF	vram1	ram
0x4E000000	0x4E000FFF	eth0	LAN9118
0x4F000000	0x4F00FFFF	usb0	ISP1761
0x60000000	0x9FFFFFFF	ddr2RamBridge	bridge

Table 8. Bridged Memory Map ('cpu' / 'ddr2RemapBridgeNS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
0x0	0x3FFFFFF	ddr2RemapBridge	bridge
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110

Table 9. Bridged Memory Map ('cpu' / 'sysRegsNS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10000000	0x10000FFF	sysRegs	VexpressSysRegs

Table 10. Bridged Memory Map ('cpu' / 'sysCtrlNS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10001000	0x10001FFF	sysCtrl	SysCtrlSP810

Table 11. Bridged Memory Map ('cpu' / 'aac1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10004000	0x10004FFF	aac1	AaciPL041

Table 12. Bridged Memory Map ('cpu' / 'mmc1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10005000	0x10005FFF	mmc1	MmciPL181

Table 13. Bridged Memory Map ('cpu' / 'kb1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10006000	0x10006FFF	kb1	KbPL050

Table 14. Bridged Memory Map ('cpu' / 'ms1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10007000	0x10007FFF	ms1	KbPL050

Table 15. Bridged Memory Map ('cpu' / 'uart0NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10009000	0x10009FFF	uart0	UartPL011

Table 16. Bridged Memory Map ('cpu' / 'uart1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x1000A000	0x1000AFFF	uart1	UartPL011

Table 17. Bridged Memory Map ('cpu' / 'uart2NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x1000B000	0x1000BFFF	uart2	UartPL011

Table 18. Bridged Memory Map ('cpu' / 'uart3NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x1000C000	0x1000CFFF	uart3	UartPL011

Table 19. Bridged Memory Map ('cpu' / 'wdt1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x1000F000	0x1000FFFF	wdt1	WdtSP805

Table 20. Bridged Memory Map ('cpu' / 'timer01NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10011000	0x10011FFF	timer01	TimerSP804

Table 21. Bridged Memory Map ('cpu' / 'timer02NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10012000	0x10012FFF	timer23	TimerSP804

Table 22. Bridged Memory Map ('cpu' / 'dvi1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10016000	0x10016FFF	dvi1	SerBusDviRegs

Table 23. Bridged Memory Map ('cpu' / 'rtc1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x10017000	0x10017FFF	rtc1	RtcPL031

Table 24. Bridged Memory Map ('cpu' / 'cf1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x1001A000	0x1001AFFF	cf1	CompactFlashRegs

Table 25. Bridged Memory Map ('cpu' / 'lcd2NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x1001F000	0x1001FFFF	clcd	LcdPL110

Table 26. Bridged Memory Map ('cpu' / 'l2regsNS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x1E00A000	0x1E00AFFF	l2regs	L2CachePL310

Table 27. Bridged Memory Map ('cpu' / 'nor0NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x40000000	0x43FFFFFF	nor0	NorFlash48F4400

Table 28. Bridged Memory Map ('cpu' / 'nor1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x44000000	0x47FFFFFF	nor1	NorFlash48F4400

Table 29. Bridged Memory Map ('cpu' / 'sram1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x48000000	0x4BFFFFFF	sram1	ram

Table 30. Bridged Memory Map ('cpu' / 'vram1NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x4C000000	0x4C7FFFFF	vram1	ram

Table 31. Bridged Memory Map ('cpu' / 'eth0NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x4E000000	0x4E000FFF	eth0	LAN9118

Table 32. Bridged Memory Map ('cpu' / 'usb0NS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x4F000000	0x4F00FFFF	usb0	ISP1761

Table 33. Bridged Memory Map ('cpu' / 'ddr2ramNS' / 'pBus' [width: 32])

Lo Address	Hi Address	Instance	Component
remappable	remappable	clcd	LcdPL110
remappable	remappable	lcd1	LcdPL110
0x60000000	0x9FFFFFFF	ddr2RamBridge	bridge

3.4 Net Connections to processor: 'cpu'

Table 34. Processor Net Connections ('cpu')

Net Port	Net	Instance	Component
SPI34	ir2	timer01	TimerSP804
SPI35	ir3	timer23	TimerSP804
SPI36	ir4	rtc1	RtcPL031
SPI37	ir5	uart0	UartPL011
SPI38	ir6	uart1	UartPL011
SPI39	ir7	uart2	UartPL011
SPI40	ir8	uart3	UartPL011

SPI41	ir9	mmc1	MmciPL181
SPI42	ir10	mmc1	MmciPL181
SPI44	ir12	kb1	KbPL050
SPI45	ir13	ms1	KbPL050
SPI46	ir14	clcd	LcdPL110
SPI47	ir15	eth0	LAN9118
SPI48	ir16	usb0	ISP1761
SPI76	ir44	lcd1	LcdPL110
SPI80	ir48	timer45	TimerSP804

4.0 Peripheral Instances

4.1 Peripheral [arm.ovpworld.org/peripheral/VexpressSysRegs/1.0] instance: sysRegs

4.1.1 Description
ARM Versatile Express System Registers

4.1.2 Limitations
Only select registers are modeled. See user.c for details.

4.1.3 Reference
A

4.1.4 Licensing
Open Source Apache 2.0

Table 35. Configuration options (attributes) set for instance 'sysRegs'

Attributes	Value
SYS_PROCID0	0x0c000191

4.2 Peripheral [arm.ovpworld.org/peripheral/SysCtrlSP810/1.0] instance: sysCtrl

4.2.1 Description
ARM SP810 System Control Registers

4.2.2 Limitations
Programmers View, register model only. Does NOT model functionality, just provides registers to allow code to run.

4.2.3 Reference
ARM PrimeCell System Controller (SP810) Technical Reference Manual (ARM DDI 0254)

4.2.4 Licensing
Open Source Apache 2.0

There are no configuration options set for this peripheral instance.

4.3 Peripheral [ovpworld.org/peripheral/dummyPort/1.0] instance: sbpci0

4.3.1 Description
Dummy peripheral that provides an area for accesses.

4.3.2 Limitations
Has no behavior. This peripheral defines a port through which a 4k byte memory area can be read and written.

4.3.3 Licensing
Open Source Apache 2.0

4.3.4 Reference
This is not based upon a real device

There are no configuration options set for this peripheral instance.

4.4 Peripheral [arm.ovpworld.org/peripheral/AaciPL041/1.0] instance: aac1

4.4.1 Description
ARM PL041 PrimeCell Advanced Audio CODEC Interface Registers

4.4.2 Limitations
Programmers View, register model only. Does NOT model functionality, just provides registers to allow code to run.

4.4.3 Reference
ARM PrimeCell Advanced Audio CODEC Interface (PL041) Technical Reference Manual (ARM DDI 0173)

4.4.4 Licensing
Open Source Apache 2.0

There are no configuration options set for this peripheral instance.

4.5 Peripheral [arm.ovpworld.org/peripheral/MmciPL181/1.0] instance: mmc1

4.5.1 Description
ARM PrimeCell Multimedia Card Interface (MMCI)

4.5.2 Limitations
None

4.5.3 Licensing
Open Source Apache 2.0

4.5.4 Reference
ARM PrimeCell Multimedia Card Interface (Pl180) Technical Reference Manual (ARM DDI 0172)

There are no configuration options set for this peripheral instance.

4.6 Peripheral [arm.ovpworld.org/peripheral/KbPL050/1.0] instance: kb1

4.6.1 Description
ARM PL050 PS2 Keyboard or mouse controller

4.6.2 Limitations
None

4.6.3 Reference
ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050) Technical Reference Manual (ARM DDI 0143)

4.6.4 Licensing
Open Source Apache 2.0

Table 36. Configuration options (attributes) set for instance 'kb1'

Attributes	Value
isKeyboard	1
grabDisable	1

4.7 Peripheral [arm.ovpworld.org/peripheral/KbPL050/1.0] instance: ms1

4.7.1 Description
ARM PL050 PS2 Keyboard or mouse controller

4.7.2 Limitations
None

4.7.3 Reference
ARM PrimeCell PS2 Keyboard/Mouse Interface (PL050) Technical Reference Manual (ARM DDI 0143)

4.7.4 Licensing
Open Source Apache 2.0

Table 37. Configuration options (attributes) set for instance 'ms1'

Attributes	Value
isMouse	1
grabDisable	1

4.8 Peripheral [arm.ovpworld.org/peripheral/UartPL011/1.0] instance: uart0

4.8.1 Description
ARM PL011 UART

4.8.2 Limitations
This is not a complete model of the PL011. There is no modeling of physical aspects of the UART, such as baud rates etc.

4.8.3 Reference
ARM PrimeCell UART (PL011) Technical Reference Manual (ARM DDI 0183)

4.8.4 Licensing
Open Source Apache 2.0

Table 38. Configuration options (attributes) set for instance 'uart0'

Attributes	Value
variant	ARM
log	log
outfile	uart0.log
portnum	portnum
console	console
finishOnDisconnect	1

4.9 Peripheral [arm.ovpworld.org/peripheral/UartPL011/1.0] instance: uart1

4.9.1 Description
ARM PL011 UART

4.9.2 Limitations
This is not a complete model of the PL011. There is no modeling of physical aspects of the UART, such as baud rates etc.

4.9.3 Reference
ARM PrimeCell UART (PL011) Technical Reference Manual (ARM DDI 0183)

4.9.4 Licensing
Open Source Apache 2.0

Table 39. Configuration options (attributes) set for instance 'uart1'

Attributes	Value
variant	ARM
log	log
outfile	uart1.log
portnum	portnum
console	console
finishOnDisconnect	finishOnDisconnect

4.10 Peripheral [arm.ovpworld.org/peripheral/UartPL011/1.0] instance: uart2

4.10.1 Description
ARM PL011 UART

4.10.2 Limitations
This is not a complete model of the PL011. There is no modeling of physical aspects of the UART, such as baud rates etc.

4.10.3 Reference
ARM PrimeCell UART (PL011) Technical Reference Manual (ARM DDI 0183)

4.10.4 Licensing
Open Source Apache 2.0

Table 40. Configuration options (attributes) set for instance 'uart2'

Attributes	Value
variant	ARM
log	log
outfile	outfile
portnum	portnum
console	console
finishOnDisconnect	finishOnDisconnect

4.11 Peripheral [arm.ovpworld.org/peripheral/UartPL011/1.0] instance: uart3

4.11.1 Description
ARM PL011 UART

4.11.2 Limitations
This is not a complete model of the PL011. There is no modeling of physical aspects of the UART, such as baud rates etc.

4.11.3 Reference

ARM PrimeCell UART (PL011) Technical Reference Manual (ARM DDI 0183)

4.11.4 Licensing
Open Source Apache 2.0

Table 41. Configuration options (attributes) set for instance 'uart3'

Attributes	Value
variant	ARM
log	log
outfile	outfile
portnum	portnum
console	console
finishOnDisconnect	finishOnDisconnect

4.12 Peripheral [arm.ovpworld.org/peripheral/WdtSP805/1.0] instance: wdt1

4.12.1 Description
ARM SP805 Watchdog Registers.

4.12.2 Limitations
Does NOT model watchdog functionality, just provides registers to allow code to run.

4.12.3 Reference
ARM Watchdog Module (SP805) Technical Reference Manual (ARM DDI 0270)

4.12.4 Licensing
Open Source Apache 2.0

There are no configuration options set for this peripheral instance.

4.13 Peripheral [arm.ovpworld.org/peripheral/TimerSP804/1.0] instance: timer01

4.13.1 Description
Timer SP804 Module

4.13.2 Licensing
Open Source Apache 2.0

4.13.3 Limitations
none

4.13.4 Reference
ARM Dual-Timer Module (SP804) Technical Reference Manual (ARM DDI 0271)

There are no configuration options set for this peripheral instance.

4.14 Peripheral [arm.ovpworld.org/peripheral/TimerSP804/1.0] instance: timer23

4.14.1 Description
Timer SP804 Module

4.14.2 Licensing
Open Source Apache 2.0

4.14.3 Limitations
none

4.14.4 Reference
ARM Dual-Timer Module (SP804) Technical Reference Manual (ARM DDI 0271)

There are no configuration options set for this peripheral instance.

4.15 Peripheral [arm.ovpworld.org/peripheral/SerBusDviRegs/1.0] instance: dvi1

4.15.1 Description
Versatile Express Serial Bus DVI Registers

4.15.2 Licensing
Open Source Apache 2.0

4.15.3 Limitations
Programmers View, register model only. Does NOT model functionality, just provides registers to allow code to run.

4.15.4 Reference
A

There are no configuration options set for this peripheral instance.

4.16 Peripheral [arm.ovpworld.org/peripheral/RtcPL031/1.0] instance: rtc1

4.16.1 Description
ARM PL031 Real Time Clock (RTC)

4.16.2 Limitations
none

4.16.3 Reference
ARM PrimeCell Real Time Clock (PL031) Technical Reference Manual (ARM DDI 0224)

4.16.4 Licensing
Open Source Apache 2.0

There are no configuration options set for this peripheral instance.

4.17 Peripheral [arm.ovpworld.org/peripheral/CompactFlashRegs/1.0] instance: cf1

4.17.1 Description
ARM Versatile Express Compact Flash Interface Registers

4.17.2 Limitations
Programmers View, register model only. Does NOT model functionality, just provides registers to allow code to run.

4.17.3 Reference
ARM Motherboard Express uATX (V2M-P1) Technical Reference Manual (ARM DDI 0447)

4.17.4 Licensing
Open Source Apache 2.0

There are no configuration options set for this peripheral instance.

4.18 Peripheral [arm.ovpworld.org/peripheral/LcdPL110/1.0] instance: clcd

4.18.1 Description
ARM PL110 LCD Controller

4.18.2 Limitations
The VGA display refresh is not optimised resulting in the VGA peripheral causing a limit on the maximum performance of a platform it contains to be around 300 MIPS (actual dependent upon refresh rate of LCD).
The LCD peripheral utilises memory watchpoints to optimise display refresh. This requires the use of ICM memory for the frame buffers, which currently may stop its use in SystemC TLM2 platforms.
Interrupts are not supported

4.18.3 Reference
ARM PrimeCell Color LCD Controller (PL111) Technical Reference Manual (ARM DDI 0293)

4.18.4 Licensing
Open Source Apache 2.0

Table 42. Configuration options (attributes) set for instance 'clcd'

Attributes	Value
resolution	xga
noGraphics	1

4.19 Peripheral [arm.ovpworld.org/peripheral/LcdPL110/1.0] instance: lcd1

4.19.1 Description

ARM PL110 LCD Controller

4.19.2 Limitations

The VGA display refresh is not optimised resulting in the VGA peripheral causing a limit on the maximum performance of a platform it contains to be around 300 MIPS (actual dependent upon refresh rate of LCD).
The LCD peripheral utilises memory watchpoints to optimise display refresh. This requires the use of ICM memory for the frame buffers, which currently may stop its use in SystemC TLM2 platforms.
Interrupts are not supported

4.19.3 Reference

ARM PrimeCell Color LCD Controller (PL111) Technical Reference Manual (ARM DDI 0293)

4.19.4 Licensing

Open Source Apache 2.0

Table 43. Configuration options (attributes) set for instance 'lcd1'

Attributes	Value
resolution	xga

4.20 Peripheral [ovpworld.org/peripheral/DynamicBridge/1.0] instance: lcd1NS

4.20.1 Description

DynamicBridge - Dynamically enable/disable a bus bridge from the input slave port to the output master port. The bridge is enabled when the input net is high, disabled when it is low. The size of the port is defined with the portSize parameter. The address on the input slave port is defined by the spLoAddress parameter. The address on the output master port is defined by the mpLoAddress parameter. All three parameters must be specified. The input and output ports may be connected to the same bus.

4.20.2 Licensing

Open Source Apache 2.0

4.20.3 Limitations

The range of the input slave port must not conflict with any exiting port connected to the bus. The output bus width is hard coded to be 32 bits.

4.20.4 Reference

This is not based upon the operation of a real device

Table 44. Configuration options (attributes) set for instance 'lcd1NS'

Attributes	Value
mpLoAddress	0x10020000
spLoAddress	1099780194304
portSize	0x1000

4.21 Peripheral [arm.ovpworld.org/peripheral/DMemCtrlPL341/1.0] instance: dmc1

4.21.1 Description

ARM PL341 Dynamic Memory Controller Registers

4.21.2 Limitations

Programmers View, register model only. Does NOT model functionality, just provides registers to allow code to run.

4.21.3 Reference

ARM CoreLink DDR2 Dynamic Memory Controller (DMC-341) Technical Reference Manual (ARM DDI 0418)

4.21.4 Licensing

Open Source Apache 2.0

There are no configuration options set for this peripheral instance.

4.22 Peripheral [ovpworld.org/peripheral/DynamicBridge/1.0] instance: dmc1NS

4.22.1 Description

DynamicBridge - Dynamically enable/disable a bus bridge from the input slave port to the output master port. The bridge is enabled when the input net is high, disabled when it is low. The size of the port is defined with the portSize parameter. The address on the input slave port is defined by the spLoAddress parameter. The address on the output master port is defined by the mpLoAddress parameter. All three parameters must be specified. The input and output ports may be connected to the same bus.

4.22.2 Licensing

Open Source Apache 2.0

4.22.3 Limitations

The range of the input slave port must not conflict with any exiting port connected to the bus. The output bus width is hard coded to be 32 bits.

4.22.4 Reference
This is not based upon the operation of a real device

Table 45. Configuration options (attributes) set for instance 'dmc1NS'

Attributes	Value
mpLoAddress	0x100e0000
spLoAddress	1099780980736
portSize	0x1000

4.23 Peripheral [arm.ovpworld.org/peripheral/SMemCtrlPL354/1.0] instance: smc1

4.23.1 Description
PL354 Static Memory Controller

4.23.2 Limitations
Programmers View, register model only. Does NOT model functionality, just provides registers to allow code to run.

4.23.3 Reference
ARM PrimeCell Static Memory Controller (PL350 series) Technical Reference Manual (ARM DDI 0380)

4.23.4 Licensing
Open Source Apache 2.0

There are no configuration options set for this peripheral instance.

4.24 Peripheral [ovpworld.org/peripheral/DynamicBridge/1.0] instance: smc1NS

4.24.1 Description
DynamicBridge - Dynamically enable/disable a bus bridge from the input slave port to the output master port. The bridge is enabled when the input net is high, disabled when it is low. The size of the port is defined with the portSize parameter. The address on the input slave port is defined by the spLoAddress parameter. The address on the output master port is defined by the mpLoAddress parameter. All three parameters must be specified. The input and output ports may be connected to the same bus.

4.24.2 Licensing
Open Source Apache 2.0

4.24.3 Limitations
The range of the input slave port must not conflict with any exiting port connected to the bus. The output bus width is hard coded to be 32 bits.

4.24.4 Reference

This is not based upon the operation of a real device

Table 46. Configuration options (attributes) set for instance 'smc1NS'

Attributes	Value
mpLoAddress	0x100e1000
spLoAddress	1099780984832
portSize	0x1000

4.25 Peripheral [ovpworld.org/peripheral/dummyPort/1.0] instance: scc1

4.25.1 Description
Dummy peripheral that provides an area for accesses.

4.25.2 Limitations
Has no behavior. This peripheral defines a port through which a 4k byte memory area can be read and written.

4.25.3 Licensing
Open Source Apache 2.0

4.25.4 Reference
This is not based upon a real device

There are no configuration options set for this peripheral instance.

4.26 Peripheral [ovpworld.org/peripheral/DynamicBridge/1.0] instance: scc1NS

4.26.1 Description
DynamicBridge - Dynamically enable/disable a bus bridge from the input slave port to the output master port. The bridge is enabled when the input net is high, disabled when it is low. The size of the port is defined with the portSize parameter. The address on the input slave port is defined by the spLoAddress parameter. The address on the output master port is defined by the mpLoAddress parameter. All three parameters must be specified. The input and output ports may be connected to the same bus.

4.26.2 Licensing
Open Source Apache 2.0

4.26.3 Limitations
The range of the input slave port must not conflict with any exiting port connected to the bus. The output bus width is hard coded to be 32 bits.

4.26.4 Reference
This is not based upon the operation of a real device

Table 47. Configuration options (attributes) set for instance 'scc1NS'

Attributes	Value
mpLoAddress	0x100e2000
spLoAddress	1099780988928
portSize	0x1000

4.27 Peripheral [arm.ovpworld.org/peripheral/TimerSP804/1.0] instance: timer45

4.27.1 Description
Timer SP804 Module

4.27.2 Licensing
Open Source Apache 2.0

4.27.3 Limitations
none

4.27.4 Reference
ARM Dual-Timer Module (SP804) Technical Reference Manual (ARM DDI 0271)

There are no configuration options set for this peripheral instance.

4.28 Peripheral [ovpworld.org/peripheral/DynamicBridge/1.0] instance: timer45NS

4.28.1 Description
DynamicBridge - Dynamically enable/disable a bus bridge from the input slave port to the output master port. The bridge is enabled when the input net is high, disabled when it is low. The size of the port is defined with the portSize parameter. The address on the input slave port is defined by the spLoAddress parameter. The address on the output master port is defined by the mpLoAddress parameter. All three parameters must be specified. The input and output ports may be connected to the same bus.

4.28.2 Licensing
Open Source Apache 2.0

4.28.3 Limitations
The range of the input slave port must not conflict with any exiting port connected to the bus. The output bus width is hard coded to be 32 bits.

4.28.4 Reference
This is not based upon the operation of a real device

Table 48. Configuration options (attributes) set for instance 'timer45NS'

Attributes	Value
mpLoAddress	0x100e4000
spLoAddress	1099780997120
portSize	0x1000

4.29 Peripheral [arm.ovpworld.org/peripheral/WdtSP805/1.0] instance: wdt2

4.29.1 Description
ARM SP805 Watchdog Registers.

4.29.2 Limitations
Does NOT model watchdog functionality, just provides registers to allow code to run.

4.29.3 Reference
ARM Watchdog Module (SP805) Technical Reference Manual (ARM DDI 0270)

4.29.4 Licensing
Open Source Apache 2.0

There are no configuration options set for this peripheral instance.

4.30 Peripheral [ovpworld.org/peripheral/DynamicBridge/1.0] instance: wdt2NS

4.30.1 Description
DynamicBridge - Dynamically enable/disable a bus bridge from the input slave port to the output master port. The bridge is enabled when the input net is high, disabled when it is low. The size of the port is defined with the portSize parameter. The address on the input slave port is defined by the spLoAddress parameter. The address on the output master port is defined by the mpLoAddress parameter. All three parameters must be specified. The input and output ports may be connected to the same bus.

4.30.2 Licensing
Open Source Apache 2.0

4.30.3 Limitations
The range of the input slave port must not conflict with any exiting port connected to the bus. The output bus width is hard coded to be 32 bits.

4.30.4 Reference
This is not based upon the operation of a real device

Table 49. Configuration options (attributes) set for instance 'wdt2NS'

Attributes	Value
mpLoAddress	0x100e5000

spLoAddress	1099781001216
portSize	0x1000

4.31 Peripheral [arm.ovpworld.org/peripheral/TzpcBP147/1.0] instance: tzpc

4.31.1 Description

ARM BP147 TrustZone Protection Controller.
There are 24 output net ports (TZPCDECPROT0_0 thru TZPCDECPROT2_7) corresponding to the 24 enables defined as 3 banks of 8 enables.
Each net port may be used to drive a net connected to a peripheral such as the DynamicBridge which can enable/disable a BusBridge mapping under control of an input net.

4.31.2 Limitations

none

4.31.3 Licensing

Open Source Apache 2.0

4.31.4 Reference

ARM PrimeCell Infrastructure AMBA3 TrustZone Protection Controller (BP147) Technical Overview (ARM DTO 0015)

There are no configuration options set for this peripheral instance.

4.32 Peripheral [ovpworld.org/peripheral/dummyPort/1.0] instance: gpio0

4.32.1 Description

Dummy peripheral that provides an area for accesses.

4.32.2 Limitations

Has no behavior. This peripheral defines a port through which a 4k byte memory area can be read and written.

4.32.3 Licensing

Open Source Apache 2.0

4.32.4 Reference

This is not based upon a real device

There are no configuration options set for this peripheral instance.

4.33 Peripheral [ovpworld.org/peripheral/dummyPort/1.0] instance: faxi1

4.33.1 Description

Dummy peripheral that provides an area for accesses.

4.33.2 Limitations
Has no behavior. This peripheral defines a port through which a 4k byte memory area can be read and written.

4.33.3 Licensing
Open Source Apache 2.0

4.33.4 Reference
This is not based upon a real device

There are no configuration options set for this peripheral instance.

4.34 Peripheral [ovpworld.org/peripheral/DynamicBridge/1.0] instance: faxi1NS

4.34.1 Description
DynamicBridge - Dynamically enable/disable a bus bridge from the input slave port to the output master port. The bridge is enabled when the input net is high, disabled when it is low. The size of the port is defined with the portSize parameter. The address on the input slave port is defined by the spLoAddress parameter. The address on the output master port is defined by the mpLoAddress parameter. All three parameters must be specified. The input and output ports may be connected to the same bus.

4.34.2 Licensing
Open Source Apache 2.0

4.34.3 Limitations
The range of the input slave port must not conflict with any exiting port connected to the bus. The output bus width is hard coded to be 32 bits.

4.34.4 Reference
This is not based upon the operation of a real device

Table 50. Configuration options (attributes) set for instance 'faxi1NS'

Attributes	Value
mpLoAddress	0x100e9000
spLoAddress	1099781017600
portSize	0x1000

4.35 Peripheral [ovpworld.org/peripheral/dummyPort/1.0] instance: saxi1

4.35.1 Description
Dummy peripheral that provides an area for accesses.

4.35.2 Limitations
Has no behavior. This peripheral defines a port through which a 4k byte memory area can be read and written.

4.35.3 Licensing
Open Source Apache 2.0

4.35.4 Reference
This is not based upon a real device

There are no configuration options set for this peripheral instance.

4.36 Peripheral [ovpworld.org/peripheral/DynamicBridge/1.0] instance: saxi1NS

4.36.1 Description
DynamicBridge - Dynamically enable/disable a bus bridge from the input slave port to the output master port. The bridge is enabled when the input net is high, disabled when it is low. The size of the port is defined with the portSize parameter. The address on the input slave port is defined by the spLoAddress parameter. The address on the output master port is defined by the mpLoAddress parameter. All three parameters must be specified. The input and output ports may be connected to the same bus.

4.36.2 Licensing
Open Source Apache 2.0

4.36.3 Limitations
The range of the input slave port must not conflict with any exiting port connected to the bus. The output bus width is hard coded to be 32 bits.

4.36.4 Reference
This is not based upon the operation of a real device

Table 51. Configuration options (attributes) set for instance 'saxi1NS'

Attributes	Value
mpLoAddress	0x100ea000
spLoAddress	1099781021696
portSize	0x1000

4.37 Peripheral [arm.ovpworld.org/peripheral/L2CachePL310/1.0] instance: l2regs

4.37.1 Description
ARM PL310 L2 Cache Control Registers

4.37.2 Licensing

Open Source Apache 2.0

4.37.3 Limitations

Programmers View, register model only. Does NOT model functionality, just provides registers to allow code to run.

4.37.4 Reference

ARM PrimeCell Level 2 Cache Controller (PL310) Technical Reference Manual (ARM DDI 0246)

There are no configuration options set for this peripheral instance.

4.38 Peripheral [intel.ovpworld.org/peripheral/NorFlash48F4400/1.0] instance: nor0

4.38.1 Description

Intel StrataFlash P30 Memory: 64MB (512-Mbit), 2x16, Top configuration.
Organized in a 2x16 configuration (2 16 bit wide data chips, requiring commands to be repeated in data bits 0:15 and 16:31).
Top configuration (last 4 blocks are 64KB parameter blocks).

4.38.2 Limitations

No flash program timing is modeled, all operations including erase take effect immediately, and suspend commands are NOPs.
Block Locking, Lock-down and OTP (One-Time Programmable) features are not modeled (Blocks are always reported to be unlocked).
Burst-Mode reads and the Read Configuration register are not modeled.
Buffered writes are written immediately, not held until confirm command is issued.

4.38.3 Licensing

Open Source Apache 2.0

4.38.4 Reference

Intel Nor Flash Datasheet as used on Xilinx ML505 board:
http://www.xilinx.com/products/boards/ml505/datasheets/30666604.pdf

There are no configuration options set for this peripheral instance.

4.39 Peripheral [intel.ovpworld.org/peripheral/NorFlash48F4400/1.0] instance: nor1

4.39.1 Description

Intel StrataFlash P30 Memory: 64MB (512-Mbit), 2x16, Top configuration.
Organized in a 2x16 configuration (2 16 bit wide data chips, requiring commands to be repeated in data bits 0:15 and 16:31).
Top configuration (last 4 blocks are 64KB parameter blocks).

4.39.2 Limitations

No flash program timing is modeled, all operations including erase take effect immediately, and suspend commands are NOPs.
Block Locking, Lock-down and OTP (One-Time Programmable) features are not modeled (Blocks are always reported to be unlocked).
Burst-Mode reads and the Read Configuration register are not modeled.
Buffered writes are written immediately, not held until confirm command is issued.

4.39.3 Licensing

Open Source Apache 2.0

4.39.4 Reference

Intel Nor Flash Datasheet as used on Xilinx ML505 board:
http://www.xilinx.com/products/boards/ml505/datasheets/30666604.pdf

There are no configuration options set for this peripheral instance.

4.40 Peripheral [smsc.ovpworld.org/peripheral/LAN9118/1.0] instance: eth0

4.40.1 Description

Fully functional Model of SMSC LAN9118 for Arm Versatile Express platforms. For full details please consult README-EMAC.txt

4.40.2 Licensing

Open Source Apache 2.0

4.40.3 Limitations

See README-EMAC.txt

4.40.4 Reference

SMSC LAN9118 High Performance single-chip 10/100 Non-PCI Ethernet Controller Datasheet Revision 1.5 (07-11-08)

There are no configuration options set for this peripheral instance.

4.41 Peripheral [philips.ovpworld.org/peripheral/ISP1761/1.0] instance: usb0

4.41.1 Description

Functional Model of USB Phillips ISP1761 for Arm Versatile Explress platforms. For full details please consult README-OTG.txt

4.41.2 Licensing

Open Source Apache 2.0

4.41.3 Limitations

- Only host mode is supported. - DMA modes are not supported for the moment, only the mandatory slave mode is implemented. - Control and bulk transfer types are currently implemented. No interrupt and isochronous transfers yet. - Tested only the attachment of a single host device. The HSOTG controller's root hub has a single port, so only one device can be attached to it. This device could be a hub, though. Currently we support only one non-hub device. - Hot plug events are currently unsupported.

4.41.4 Reference

Philips/NXP

There are no configuration options set for this peripheral instance.

4.42 Peripheral [arm.ovpworld.org/peripheral/SmartLoaderArmLinux/1.0] instance: smartLoader

4.42.1 Description

Psuedo-peripheral to perform memory initialisation for an ARM based Linux kernel boot:
Loads Linux kernel image file and (optional) initial ram disk image into memory.
Writes ATAG data into memory.
Writes tiny boot code at physical memory base that configures the registers as expected by Linux Kernel and then jumps to boot address (image load address by default).

4.42.2 Licensing

Open Source Apache 2.0

4.42.3 Limitations

Only supports little endian

4.42.4 Reference

See ARM Linux boot requirements in Linux source tree at documentation/arm/Booting

Table 52. Configuration options (attributes) set for instance 'smartLoader'

Attributes	Value
kernel	kernel
initrd	initrd
command	mem=1024M raid=noautodetect console=ttyAMA0,38400n8 vmalloc=256MB devtmpfs.mount=0
physicalbase	0x60000000
memsize	0x40000000
boardid	0x8e0
disable	disable

5.0 Overview of Imperas OVP Virtual Platforms

This document provides the details of the usage of an Imperas OVP Virtual Platform. The first half of the document covers specifics of this particular virtual platform.

This second part of the document, includes information about Imperas OVP virtual platforms, how they are built and used.

The Imperas virtual platforms are designed to provide a base for you to run high-speed software simulations of CPU-based SoCs and platforms on any suitable PC. They are typically based on the functionality of vendors fixed or evaluation platforms, enabling you to simulate software on these reference platforms. Typically virtual platforms are fixed and require the vendor to modify or extend them. Imperas virtual platforms are different in that they enable you to extend the functionality of the virtual platform, to closer reflect your own platform, by adding more component models, running different operating systems or adding additional applications.

Imperas virtual platforms are created using the Imperas iGen technology, allowing them to be used with Imperas OVP based simulators and also with Accellera/OSCI compliant SystemC simulators and commercial EDA System Design environments that use SystemC.

Virtual platforms include simulation models of the target devices, including the processor model(s) for the target device plus enough peripheral models to boot an operating system or run bare metal applications. The platform and the peripheral models used in most of the virtual platforms are open source, so that you can easily add new models to the platform as well as modify the existing models. Some models are only provided as binary, normally because the IP owner has restricted the release of the model source. In this case, please contact Imperas for more information.

There are typically several generic flavors of the virtual platforms for specific processor families, some targeting full operating systems, such as Linux, and some which focus on Real Time Operating Systems (RTOS) such as Mentor Nucleus or freeRTOS. OVP models of the processor cores are included in the virtual platforms, and for those processors which support mulitple cores SMP Linux is often supported for that virtual platform. For all of these virtual platforms, many of the peripheral components of the platform are modeled, often including the Ethernet and USB components. The semi-hosting capability of the Imperas virtual platform simulator products enables connection via the Ethernet and USB components from the virtual platform to the real world via the x86 host machine.

The Imperas OVP CPU models are written using the OVP Virtual Machine Interface (VMI) API that defines the behavior of the processor. The VMI API makes a clear line between model and simulator allowing very good optimization and world class high speed performance. The processor models are Instruction Accurate and do not model the detailed cycle timing of the processor and they implement functionality at the level of a Programmers View of the processor and peripherals and the software running on them does not know it is not running on hardware. Many models are provided as a binary shared object and also as source. This allows the download and use of the model binary or the use of the source to explore

and modify the model. The models are run through an extensive QA and regression testing process and most processor model families are validated using technology provided by the processor IP owners. All the models in this platform are developed with the Open Virtual Platforms APIs and are implemented in C.

More information on modeling and APIs can be found on the www.OVPworld.org site.

6.0 Getting Started with Imperas OVP Virtual Platforms

Virtual platforms are downloadable from the OVPworld website OVPworld.org/downloads. You need to browse and look for '<platform processor name> Examples'. You do need to be registered and logged in on the OVP site to download. OVPworld currently provides 32 bit host versions of packages containing virtual platforms.

When downloading, choose, Linux or Windows host. 32 bit packages can be installed and executed on 32 bit or 64 bit hosts. If you require a 64 bit host version please contact Imperas.

For example, for the ARM Versatile Express platform booting Linux on Cortex-A15MP Single, Dual, and Quad core procesors, you would want the download package:
'OVPsim_demo_Linux_ArmVersatileExpress_arm_Cortex-A15MP'.

Most virtual platform packages contain the platform and all the processor and peripheral models needed. You will need to download a simulator to run the platform. You can use OVPsim, downloadable from OVPworld.org/downloads, or you can use one of the Imperas simulators (imperas.com/products) available commercially from Imperas.

7.0 Simulating Software

7.1 Getting a license key to run

After you have downloaded you will need a runtime license key before the simulators will run. For OVPsim please visit OVPworld.org/likey and provide the required information and an evaluation/demo license key will be automatically sent to you. If you are using Imperas, then please contact Imperas for a license key.

7.2 Normal runs

To run a platform, read the section below on command line control of the platform and the section on setting command line arguments.

7.3 Loading Software

For most virtual platforms the platform is already configured to run the default software application/program and there is normally a script to run that sets some arguments. You can then copy/edit this script to select your own applications etc.

The example application programs are typically .elf format files and are provided pre-compiled. There are normally makefiles and associated scripts to recompile the example applications.

To find more information about compiling and loading software, the following document should be looked at: Imperas_Installation_and_Getting_Started.pdf.

7.4 Semihosting

In a virtual platform, semihosting is not normally used as there is normally hardware that implements the appropriate functionality - for example I/O will be handled by UARTs etc.

7.5 Using a terminal (UART)

If the platform includes one or more UARTs you will need to connect a terminal program to it so that you can see output and type into the simulated program. Review the list of peripherals below and see what configuration options it has been set with. In most cases there is an option to set to instruct the simulator to 'pop up' a terminal window connected to the simulated UART.

7.6 Interacting with the simulation (keyboard and mouse)

If the platform has a simulated UART you can normally set a command to get the simulator to pop up a terminal window allowing you to see output from the simulated UART and also allowing you to type characters into the UART that can be processed by the simulated software.

If your simulated platform has an LCD device then you can often configure it to recognize mouse movements and mouse clicks - allowing full interaction.

To see these interactions in action, have a look at some of the available videos available at OVPworld.org/demosandvideos.

7.7 More Information (Documentation) on Simulation

To find more information about running simulations and more of the options the simulators provide, the following documents should be looked at:
Imperas_Installation_and_Getting_Started.pdf
OVPsim_and_CpuManager_User_Guide.pdf
OVP_Control_File_User_Guide.pdf

A full list of the currently available OVP documentation is available: OVPworld.org/documentation.

8.0 Debugging Software running on an Imperas OVP Virtual Platform

The Imperas and OVP simulators have several different interfaces to debuggers. These include several proprietary formats and also the standard GNU RSP format is supported allowing many compatible debuggers to be used. Below are some examples that Imperas directly support.

8.1 Debugging with GDB

A GNU debugger (GDB) can be connected to a processor in a platform using the RSP protocol. This allows the application program running on a processor to be debugged using a specific GDB for the processor selected. When using the Imperas Professional products many connections can be made allowing a GDB to

be connected to all the processors in the platform.

The use of GDB is documented: OVPsim_Debugging_Applications_with_GDB_User_Guide.pdf.

*8.2 Debugging with Imperas M*DBG*

The Imperas multi-processor debugger can be connected to a platform and through this connection you can debug application programs running on all of the processors instanced within the platform. It is also capable, within this single unified environment, to debug peripheral model behavioral code in conjunction with the processor application programs.

For more information please see the Imperas M*DBG user guide.

The Imperas multi-processor debugger is also capable of controlling the Imperas Verification Analysis aand Profiling (VAP) tools in real time, making them invaluable to application program development, debugging and analysis.

For more information please see the Imperas VAP tools user guide.

8.3 Debugging with the Imperas iGui and GDB

Imperas iGui gives a GUI front end to the use of the GDB debugger. It allows use of all the features of GDB including source level application program debugging on processors.

*8.4 Debugging with the Imperas iGui and M*DBG*

Imperas iGui gives a GUI front end to the Imperas multi-processor debugger. It provides all the features of this debugger but does so with source level application program debugging on processors and source level debugging of the behavioral code on peripheral components in the platform. A context view shows all the processor and peripheral components within the platform and allows switching between them to examine the state of each at the event at which the simulation was stopped

Imperas iGui provides a menu from which the Imperas VAP tools can be controlled.

8.5 Debugging with Eclipse

A standard Eclipse CDT development environment can be connected to one or more processors in a platform (multiple processors require an Imperas professional product). The simulation platform is started remotely or using the external tool feature in Eclipse, opens a debug port and awaits the connection with Eclipse. All features provided by the Eclipse CDT development environment are available to be used to debug software applications executing on the processors in the platform.

The use of Eclipse is documented: OVPsim_Debugging_Applications_with_Eclipse_User_Guide.pdf.

8.6 Debugging applications running under a simulated operating system

If the simulated platform is running an Operating System and the platform has a UART or Ethernet etc

connection then it is often possible to connect an external debugger and debug the applications running under the simulated operating system.

An example would be a simulated platform running the Linux operating system, such as the MIPS Malta, or ARM Versatile Express. Within the simulated Linux you can start a gdbserver that connects from within the simulation through a UART out to the host PC via a port. Within the host PC you start a terminal program and connect to the port with a debugger such as GDB and can then debug the simulated user application.

9.0 Modifying the Platform

9.1 Platforms use C/C++ and OVP APIs

The Imperas and OVP simulators execute a platform that is written in C/C++ and that makes function calls into the simulator's APIs. Thus the virtual platform is compiled from C/C++ into a binary shared object that the simulator loads and runs. OVP provides the definition and documentation that defines the C APIs for modeling the platforms, the peripherals and the processors. You can find more information about these APIs on the OVP website and in the OVP API documentation.

9.2 Platforms/Peripherals can be easily built with iGen from Imperas

Imperas provides a product 'iGen' that takes an input script file and creates the C/C++ files needed for platforms and peripherals - it creates the C/C++ file that is compiled into the platform or peripheral that is needed as an object file by the simulator. iGen creates the C/C++ files, you then need to add any necessary behaviors or further details etc. For platforms iGen creates either a C platform or a C++ SystemC TLM2 platform. For peripherals iGen creates the C files and also provides a native C++ SystemC TLM2 interface to allow the peripheral to be instantiated in SystemC TLM2 platforms.

Information on iGen is available from: imperas.com/products.

9.3 Re-configuring the platform

There will nornmally be several configuration options that you can set when running the platform without the neeed to change any source. Refer to the section above on command line arguments. If these do not allow you to make the changes you need, then you may need to edit and recompile the source of the platform.

The source of the platform and the source of the peripherals will be installed as part of the packages you are using. The sources are located in the Imperas/OVP installation VLNV source tree. The VLNV term refers to: Vendor (eg arm.ovpworld.org), Library (eg platform), Name, (eg ArmVersatileExpress-CA15), and Version (eg 1.0). To modify the platform, locate the platform source files.

If you are an Imperas user and have access to iGen, we recommend you modify the source script files and regenerate and recompile the C that makes up the platform. Refer to the Imperas iGen model generator guide and the Imperas platform generator guide.

If you are using the C or SystemC TLM2 platforms with OVPsim, then you can edit the C/C++ files, recompile the source directly using the supplied makefiles, and the run the simulator directly with the resultant shared object.

9.4 Replacing peripherals components

If you need to replace peripherals, find the appropriate place in the source of the platform, make the change you need, and recompile etc. Look in the library for documentation on available peripherals and their configuration options.

9.5 Adding new peripherals components

If you need to add peripherals, find the appropriate place in the source, make the additions you need, and recompile etc. Look in the library for documentation on available peripherals and their configuration options.

If you need to create new peripheral components then use iGen to very quickly create the necessary C/C++ files that get you started. With iGen you can create peripherals with register/memory state in a few lines of iGen source. When adding behavior to the peripherals refer to the OVP API documentation.

10.0 Available Virtual Platforms

Table 53. Imperas / OVP Extendable Platform Kits (17 available)

Platform Name	Vendor
AlteraCycloneIII_3c120	altera.ovpworld.org
AlteraCycloneV_HPS	altera.ovpworld.org
AlteraCycloneV_HPS_TLM2	altera.ovpworld.org
ARMv8-A-FMv1	arm.ovpworld.org
ArmIntegratorCP	arm.ovpworld.org
ArmIntegratorCP_TLM2.0	arm.ovpworld.org
ArmVersatileExpress	arm.ovpworld.org
ArmVersatileExpress-CA15	arm.ovpworld.org
ArmVersatileExpress-CA9	arm.ovpworld.org
ArmVersatileExpress_CA9_TLM2	arm.ovpworld.org
AtmelAT91SAM7	atmel.ovpworld.org
FreescaleKinetis60	freescale.ovpworld.org
FreescaleVybridVF5	freescale.ovpworld.org
MipsMalta	mips.ovpworld.org
MipsMaltaLinux_TLM2.0	mips.ovpworld.org
RenesasUPD70F3441	renesas.ovpworld.org
XilinxML505	xilinx.ovpworld.org

Table 54. Imperas General Virtual Platforms (6 available)

Platform Name	Vendor
arm-ti-eabi	arm.imperas.com
armm-ti-coff	arm.imperas.com
armm-ti-eabi	arm.imperas.com
HeteroAlteraCycloneV_HPS_CycloneIII_3c120	imperas.ovpworld.org
HeteroArmNucleusMIPSLinux	imperas.ovpworld.org
QuadArmVersatileExpress	imperas.ovpworld.org

Table 55. Imperas / OVP Bare Metal Virtual Platforms (39 available)

Platform Name	Vendor
BareMetalNios_IISingle	altera.ovpworld.org
BareMetalNios_IISingle_TLM2.0	altera.ovpworld.org
BareMetalArcSingle	arc.ovpworld.org
BareMetalArcSingle_TLM2.0	arc.ovpworld.org
BareMetalArm7Single	arm.ovpworld.org
BareMetalArm7Single_TLM2.0	arm.ovpworld.org
BareMetalArmAArch64Single_TLM2.0	arm.ovpworld.org
BareMetalArmCortexADual	arm.ovpworld.org
BareMetalArmCortexASingle	arm.ovpworld.org
BareMetalArmCortexASingleAngelTrap	arm.ovpworld.org
BareMetalArmCortexASingle_TLM2.0	arm.ovpworld.org
BareMetalArmCortexMSingle	arm.ovpworld.org
BareMetalArmCortexMSingle_TLM2.0	arm.ovpworld.org

ArmCortexMFreeRTOS	imperas.ovpworld.org
ArmCortexMuCOS-II	imperas.ovpworld.org
BareMetalArcManycore24_TLM2.0	imperas.ovpworld.org
BareMetalArm7Dual_TLM2.0	imperas.ovpworld.org
BareMetalArmx1Mips32x3	imperas.ovpworld.org
BareMetalMips32Multicore2_TLM2.0	imperas.ovpworld.org
Or1kUclinux_TLM2.0	imperas.ovpworld.org
BareMetalM14KSingle	mips.ovpworld.org
BareMetalM14KSingle_TLM2.0	mips.ovpworld.org
BareMetalMips32Dual	mips.ovpworld.org
BareMetalMips32Single	mips.ovpworld.org
BareMetalMips32Single_TLM2.0	mips.ovpworld.org
BareMetalMips64Single	mips.ovpworld.org
BareMetalMips64Single_TLM2.0	mips.ovpworld.org
BareMetalMipsDual	mips.ovpworld.org
BareMetalMipsSingle	mips.ovpworld.org
BareMetalMipsSingle_TLM2.0	mips.ovpworld.org
BareMetalOr1kSingle	ovpworld.org
BareMetalOr1kSingle_TLM2.0	ovpworld.org
BareMetalM16cSingle	posedgesoft.ovpworld.org
BareMetalPowerPc32Single	power.ovpworld.org
BareMetalPowerPc32Single_TLM2.0	power.ovpworld.org
BareMetalV850Single	renesas.ovpworld.org
BareMetalV850Single_TLM2.0	renesas.ovpworld.org
ghs-multi	renesas.ovpworld.org
BareMetalMicroBlazeSingle_TLM2.0	xilinx.ovpworld.org

#

