OVP Guide to Using Processor Models

Model Specific Information
for variant powerpc_mpc82x

Imperas Software Limited
Imperas Buildings, North Weston
Thame, Oxfordshire, OX9 2HA, UK
docs@imperas.com

Author	Imperas Software Limited
Version	0.5
Filename	OVP_Model_Specific_Information_powerpc32_mpc82x.pdf
Created	27 February 2018
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 Overview

This document provides the details of an OVP Fast Processor Model variant.

OVP Fast Processor Models are written in C and provide a C API for use in C based platforms. The models also provide a native interface for use in SystemC TLM2 platforms.

The models are written using the OVP VMI API that provides a Virtual Machine Interface 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.

Most 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 model families are validated using technology provided by the processor IP owners.

There is a companion document (OVP Guide to Using Processor Models) which explains the general concepts of OVP Fast Processor Models and their use. It is downloadable from the OVPworld website documentation pages.

1.1 Description

PPC32 Family Processor Model.

1.2 Licensing

Open Source Apache 2.0

1.3 Limitations

This model is currently under development. The FPU is incomplete.

1.4 Verification

Basic verification of ISA against reference has been performed.

1.5 Features

2 Configuration

2.1 Location

The model source and object file is found in the VLNV tree at:

power.ovpworld.org/processor/powerpc32/1.0

2.2 GDB Path

The default GDB for this model is found at:

$IMPERAS_HOME/lib/$IMPERAS_ARCH/gdb/powerpc-elf-gdb

2.3 Semi-Host Library

The default semi-host library file is found in the VLNV tree at :

power.ovpworld.org/semihosting/powerpc32Newlib/1.0

2.4 Processor Endian-ness

This model can be set to either endian-ness (normally by a pin, or the ELF code).

2.5 QuantumLeap Support

This processor is qualified to run in a QuantumLeap enabled simulator.

2.6 Processor ELF Code

The ELF code supported by this model is: 0x14

3 Other Variants in this Model

Table 1. All variants in this model

Variant
mpc82x
UISA

4 Bus Ports

Table 2. Bus Ports

Type	Name	min	max
master (initiator)	INSTRUCTION	32	32
master (initiator)	DATA	32	32

5 Net Ports

Table 3. Net Ports

Name	Type
reset	input

6 FIFO Ports

No FIFO Ports in this model.

7 Parameters

Table 4. Parameters that can be set in the model, type:

Name	Type	Description
verbose	Boolean	Specify verbose output messages

UISA_I_B	Boolean	UISA Feature UISA_I_B
UISA_I_BCDA	Boolean	UISA Feature UISA_I_BCDA
UISA_I_S	Boolean	UISA Feature UISA_I_S
UISA_I_E	Boolean	UISA Feature UISA_I_E
UISA_I_E_PC	Boolean	UISA Feature UISA_I_E_PC
UISA_I_E_PD	Boolean	UISA Feature UISA_I_E_PD
UISA_I_EC	Boolean	UISA Feature UISA_I_EC
UISA_I_FP	Boolean	UISA Feature UISA_I_FP
UISA_I_DFP	Boolean	UISA Feature UISA_I_DFP
UISA_I_MA	Boolean	UISA Feature UISA_I_MA
UISA_I_SP	Boolean	UISA Feature UISA_I_SP
UISA_I_V	Boolean	UISA Feature UISA_I_V
UISA_I_LMA	Boolean	UISA Feature UISA_I_LMA
UISA_I_WT	Boolean	UISA Feature UISA_I_WT
UISA_I_VLE	Boolean	UISA Feature UISA_I_VLE
ENABLE_FPU	Uns32	Enable FPU At Startup
UNIMP_TO_NOP	Boolean	Map Unimplemented Instructions to NOP
WARN_NOP	Boolean	Warn when executing nop-mapped instructions

8 Execution Modes

No execution modes.

9 Exceptions

Table 5. Exceptions handled by the model, type:

Name	Code
Reset	0
Undefined	1
Arith	2
RdAlign	3
WrAlign	4
RdAbort	5
WrAbort	6
RdPriv	7
WrPriv	8
Fetch	9

10 Hierarchy of the model

A CPU core may allow the user to configure it to instance many processors of a Symmetrical Multi Processor (SMP). A CPU core may also have sub elements within a processor, for example hardware threading blocks.
OVP processor models can be written to include SMP blocks and to have many levels of hierarchy.

Some OVP CPU models may have a fixed hierarchy, and some may be configured by settings in a configuration register. Please see the register definitions of this model.

This model documentation shows the settings and hierarchy of the default settings for this model variant.

10.1 Level 1:

This level in the model hierarchy has 2 commands.

This level in the model hierarchy has 3 register groups:

Table 6. Register groups

Group name	Registers
User	32
FloatingPoint	32
System	7

This level in the model hierarchy has no children.

11 Model Commands

11.1 Level 1:

11.1.1 isync

specify instruction address range for synchronous execution

Table 7. isync command arguments

Argument	Type	Description
-addresshi	Uns64	end address of synchronous execution range
-addresslo	Uns64	start address of synchronous execution range

11.1.2 itrace

enable or disable instruction tracing

Table 8. itrace command arguments

Argument	Type	Description
-after	Uns64	apply after this many instructions
-enable	Boolean	enable instruction tracing
-instructioncount	Boolean	include the instruction number in each trace
-off	Boolean	disable instruction tracing
-on	Boolean	enable instruction tracing
-registerchange	Boolean	show registers changed by this instruction
-registers	Boolean	show registers after each trace

12 Registers

12.1 Level 1:

12.1.1 User

Table 9. Registers at level 1, type: , register group: 'User'

Name	Bits	Initial value (Hex)		Description
GPR0	32	0	rw	
GPR1	32	0	rw	
GPR2	32	0	rw	
GPR3	32	0	rw	
GPR4	32	0	rw	
GPR5	32	0	rw	
GPR6	32	0	rw	
GPR7	32	0	rw	

GPR8	32	0	rw	
GPR9	32	0	rw	
GPR10	32	0	rw	
GPR11	32	0	rw	
GPR12	32	0	rw	
GPR13	32	0	rw	
GPR14	32	0	rw	
GPR15	32	0	rw	
GPR16	32	0	rw	
GPR17	32	0	rw	
GPR18	32	0	rw	
GPR19	32	0	rw	
GPR20	32	0	rw	
GPR21	32	0	rw	
GPR22	32	0	rw	
GPR23	32	0	rw	
GPR24	32	0	rw	
GPR25	32	0	rw	
GPR26	32	0	rw	
GPR27	32	0	rw	
GPR28	32	0	rw	
GPR29	32	0	rw	
GPR30	32	0	rw	
GPR31	32	0	rw	

12.1.2 FloatingPoint

Table 10. Registers at level 1, type: , register group: 'FloatingPoint'

Name	Bits	Initial value (Hex)		Description
FPR0	64	0	rw	
FPR1	64	0	rw	
FPR2	64	0	rw	
FPR3	64	0	rw	
FPR4	64	0	rw	
FPR5	64	0	rw	
FPR6	64	0	rw	
FPR7	64	0	rw	
FPR8	64	0	rw	
FPR9	64	0	rw	
FPR10	64	0	rw	
FPR11	64	0	rw	
FPR12	64	0	rw	

FPR13	64	0	rw	
FPR14	64	0	rw	
FPR15	64	0	rw	
FPR16	64	0	rw	
FPR17	64	0	rw	
FPR18	64	0	rw	
FPR19	64	0	rw	
FPR20	64	0	rw	
FPR21	64	0	rw	
FPR22	64	0	rw	
FPR23	64	0	rw	
FPR24	64	0	rw	
FPR25	64	0	rw	
FPR26	64	0	rw	
FPR27	64	0	rw	
FPR28	64	0	rw	
FPR29	64	0	rw	
FPR30	64	0	rw	
FPR31	64	0	rw	

12.1.3 System

Table 11. Registers at level 1, type: , register group: 'System'

Name	Bits	Initial value (Hex)		Description
PC	32	0	rw	program counter
MSR	32	0	rw	
CR	32	0	rw	
LR	32	0	rw	
CTR	32	0	rw	
XER	32	0	rw	
FPSCR	32	0	rw	

#

