



OVP Guide to Using Processor Models

Model specific information for Altera Nios II_Nios_II_F

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



| | |
|----------|--|
| Author | Imperas Software Limited |
| Version | 20220722.0 |
| Filename | OVP_Model_Specific_Information_nios_ii_Nios_II_F.pdf |
| Created | 27 July 2022 |
| Status | OVP Standard Release |

Copyright Notice

Copyright (c) 2022 Imperas Software Limited. All rights reserved. This software and documentation contain information that is the property of Imperas Software Limited. The software and documentation are furnished under a license agreement and may be used or copied only in accordance with the terms of the license agreement. No part of the software and documentation may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without prior written permission of Imperas Software Limited, or as expressly provided by the license agreement.

Right to Copy Documentation

The license agreement with Imperas permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any.

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

IMPERAS SOFTWARE LIMITED, AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Model Release Status

This model is released as part of OVP releases and is included in OVPworld packages. Please visit OVPworld.org.

Contents

| | |
|---|-----------|
| 1 Overview | 1 |
| 1.1 Description | 1 |
| 1.2 Licensing | 1 |
| 1.3 Limitations | 1 |
| 1.4 Verification | 2 |
| 1.5 Features | 2 |
| 2 Configuration | 3 |
| 2.1 Location | 3 |
| 2.2 GDB Path | 3 |
| 2.3 Semi-Host Library | 3 |
| 2.4 Processor Endian-ness | 3 |
| 2.5 QuantumLeap Support | 3 |
| 2.6 Processor ELF code | 3 |
| 3 All Variants in this model | 4 |
| 4 Bus Master Ports | 5 |
| 5 Bus Slave Ports | 6 |
| 6 Net Ports | 7 |
| 7 FIFO Ports | 9 |
| 8 Formal Parameters | 10 |
| 8.1 Parameter values | 11 |
| 9 Execution Modes | 13 |
| 10 Exceptions | 14 |
| 11 Hierarchy of the model | 15 |
| 11.1 Level 1 | 15 |
| 12 Model Commands | 16 |
| 12.1 Level 1 | 16 |
| 12.1.1 dumpTLB | 16 |
| 12.1.1.1 Argument description | 16 |

| | | |
|-----------|---------------------|-----------|
| 12.1.2 | isync | 16 |
| 12.1.3 | itrace | 16 |
| 13 | Registers | 18 |
| 13.1 | Level 1 | 18 |
| 13.1.1 | User | 18 |
| 13.1.2 | System | 19 |
| 13.1.3 | Integration_support | 19 |

Chapter 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

Nios-II Family Processor Model.

1.2 Licensing

Open Source Apache 2.0

1.3 Limitations

No Custom instructions.

No Cache model.

No JTAG.

1.4 Verification

Models have been extensively tested by Imperas, and validated against tests from Altera.

1.5 Features

Barrel Shifter.

Configurable MPU.

Configurable MMU.

Shadow Register Sets.

Hardware Multiply.

Hardware Divide.

Hardware Extended Multiply.

Chapter 2

Configuration

2.1 Location

This model's VLVN is `altera.ovpworld.org/processor/nios_ii/1.0`.

The model source is usually at:

`$IMPERAS_HOME/ImperasLib/source/altera.ovpworld.org/processor/nios_ii/1.0`

The model binary is usually at:

`$IMPERAS_HOME/lib/$IMPERAS_ARCH/ImperasLib/altera.ovpworld.org/processor/nios_ii/1.0`

2.2 GDB Path

The default GDB for this model is: `$IMPERAS_HOME/lib/$IMPERAS_ARCH/gdb/nios2-elf-gdb`.

2.3 Semi-Host Library

The default semi-host library file is `altera.ovpworld.org/semihosting/nios_iiNewlib/1.0`

2.4 Processor Endian-ness

This is a LITTLE endian model.

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: `0x71`.

Chapter 3

All Variants in this model

This model has these variants

| Variant | Description |
|----------------|------------------------------|
| Nios_II.F | (described in this document) |
| Nios_II.S | |
| Nios_II.E | |

Table 3.1: All Variants in this model

Chapter 4

Bus Master Ports

This model has these bus master ports.

| Name | min | max | Connect? | Description |
|-------------|-----|-----|-----------|-------------|
| INSTRUCTION | 32 | 32 | mandatory | |
| DATA | 32 | 32 | optional | |

Table 4.1: Bus Master Ports

Chapter 5

Bus Slave Ports

This model has no bus slave ports.

Chapter 6

Net Ports

This model has these net ports.

| Name | Type | Connect? | Description |
|---------|-------|----------|-------------|
| reset_n | input | optional | |
| d_irq0 | input | optional | |
| d_irq1 | input | optional | |
| d_irq2 | input | optional | |
| d_irq3 | input | optional | |
| d_irq4 | input | optional | |
| d_irq5 | input | optional | |
| d_irq6 | input | optional | |
| d_irq7 | input | optional | |
| d_irq8 | input | optional | |
| d_irq9 | input | optional | |
| d_irq10 | input | optional | |
| d_irq11 | input | optional | |
| d_irq12 | input | optional | |
| d_irq13 | input | optional | |
| d_irq14 | input | optional | |
| d_irq15 | input | optional | |
| d_irq16 | input | optional | |
| d_irq17 | input | optional | |
| d_irq18 | input | optional | |
| d_irq19 | input | optional | |
| d_irq20 | input | optional | |
| d_irq21 | input | optional | |
| d_irq22 | input | optional | |
| d_irq23 | input | optional | |
| d_irq24 | input | optional | |
| d_irq25 | input | optional | |
| d_irq26 | input | optional | |
| d_irq27 | input | optional | |
| d_irq28 | input | optional | |
| d_irq29 | input | optional | |

| | | | |
|---------|-------|----------|--|
| d_irq30 | input | optional | |
| d_irq31 | input | optional | |

Table 6.1: Net Ports

Chapter 7

FIFO Ports

This model has no FIFO ports.

Chapter 8

Formal Parameters

| Name | Type | Description |
|------------------------------------|-------------|---|
| variant | Enumeration | Selects variant (either a generic UISA or a specific model) |
| verbose | Boolean | Specify verbose output messages |
| endian | Endian | Specify Model endian |
| TEST_MODE_EXIT | Boolean | Enable Test mode exit for instruction, cmpltui r0, r0, (0xabc1 0xabc2) |
| TEST_HALT_EXIT | Boolean | Enable a Halt on “br 0”, branch to self |
| INST_ADDR_WIDTH | Uns32 | Instruction bus Width |
| DATA_ADDR_WIDTH | Uns32 | Data bus Width |
| HW_MULTIPLY | Boolean | Hardware Multiply |
| HW_MULX | Boolean | Hardware Extended Multiply |
| HW_DIVIDE | Boolean | Hardware Divide |
| RESET_VECTOR | Uns32 | Reset Vector Address |
| EXCEPTION_VECTOR | Uns32 | Exception Vector Address |
| BREAK_VECTOR | Uns32 | Break Vector Address |
| INCLUDE_MMU | Boolean | MMU Available |
| FAST_TLB_MISS_EXCEPTION_VECTOR | Uns32 | Fast TLB Exception Vector Address |
| INCLUDE_MPU | Boolean | MPU Available |
| INCLUDE_CPURESETREQUEST | Boolean | CPU Reset Request Signal |
| INCLUDE_CPURESETTAKEN | Boolean | CPU Reset Taken Signal |
| CPUID_CONTROL_VALUE | Uns32 | CPUID Control Register Value |
| EXCEPTION_ILLEGAL_INSTRUCTION | Boolean | Generate Illegal Instruction Exception |
| EXCEPTION_DIVISION_ERROR | Boolean | Generate Division Error Exception |
| EXCEPTION_MISALIGNED_MEMORY_ACCESS | Boolean | Generate Misaligned Memory Access Exception |
| EXCEPTION_EXTRA_INFORMATION | Boolean | Generate Extra Exception information |
| INTERRUPT_CONTROLLER_INTERFACE | Enumeration | Interrupt Controller Interface (Internal or External) |
| NUMBER_SHADOW_REGISTER_SETS | Uns32 | Number of Shadow Register Sets |
| HARDCOPY_COMPATIBILITY | Boolean | Hardcopy Compatibility |
| MMU_PID_BITS | Uns32 | MMU Process ID (PID) Bits |
| MMU_OPTIMIZE_NUMBER_OF_TLB_ENTRIES | Boolean | MMU Optimize Number of TLB Entries based on device family |
| MMU_TLB_ENTRIES | Enumeration | MMU TLB Entries (128, 256, 512 or 1024) |
| MMU_TLB_SET_ASSOCIATIVITY | Enumeration | MMU TLB Entries (8 or 16) |
| MMU_MICRO_DTLB_ENTRIES | Uns32 | MMU Micro data TLB Entries |

| | | |
|-------------------------------------|---------|---|
| MMU_MICRO_ITLB_ENTRIES | Uns32 | MMU Micro instruction TLB Entries |
| MPU_USE_LIMIT_FOR_REGION_RANGE | Boolean | Controls Memory |
| MPU_NUMBER_DATA_REGIONS | Uns32 | Number of Data Regions to Allocate |
| MPU_MINIMUM_DATA_REGION_SIZE | Uns32 | Minimum Data Region Size 64Bytes to 1MBytes (power of 2) |
| MPU_NUMBER_INSTRUCTION_REGIONS | Uns32 | Number of Instruction Regions to Allocate |
| MPU_MINIMUM_INSTRUCTION_REGION_SIZE | Uns32 | Minimum Instruction Region Size 64Bytes to 1MBytes (power of 2) |
| CUSTOM_FP_DIVISION | Boolean | Enable Custom Hardware for FP Division Instruction |
| CUSTOM_BITSWAP | Boolean | Enable Custom Hardware for Bit Swap Instruction |
| CUSTOM_ENDIAN_CONVERT | Boolean | Enable Custom Hardware for Endian Conversion Instruction |
| CUSTOM_INTERRUPT_VECTOR | Boolean | Enable Custom Interrupt Vector Instruction |

Table 8.1: Parameters

8.1 Parameter values

These are the current parameter values.

| Name | Value |
|------------------------------------|-----------|
| (Others) | |
| variant | Nios_II_F |
| verbose | T |
| endian | none |
| TEST_MODE_EXIT | F |
| TEST_HALT_EXIT | F |
| INST_ADDR_WIDTH | 0 |
| DATA_ADDR_WIDTH | 0 |
| HW_MULTIPLY | F |
| HW_MULX | F |
| HW_DIVIDE | F |
| RESET_VECTOR | 0 |
| EXCEPTION_VECTOR | 0 |
| BREAK_VECTOR | 0 |
| INCLUDE_MMU | F |
| FAST_TLB_MISS_EXCEPTION_VECTOR | 0 |
| INCLUDE_MPU | F |
| INCLUDE_CPURESETREQUEST | F |
| INCLUDE_CPURESETTAKEN | F |
| CPUID_CONTROL_VALUE | 0 |
| EXCEPTION_ILLEGAL_INSTRUCTION | F |
| EXCEPTION_DIVISION_ERROR | F |
| EXCEPTION_MISALIGNED_MEMORY_ACCESS | F |
| EXCEPTION_EXTRA_INFORMATION | F |
| INTERRUPT_CONTROLLER_INTERFACE | Internal |

| | |
|-------------------------------------|-----|
| NUMBER_SHADOW_REGISTER_SETS | 0 |
| HARDCOPY_COMPATIBILITY | F |
| MMU_PID_BITS | 8 |
| MMU_OPTIMIZE_NUMBER_OF_TLB_ENTRIES | F |
| MMU_TLB_ENTRIES | 128 |
| MMU_TLB_SET_ASSOCIATIVITY | 8 |
| MMU_MICRO_DTLB_ENTRIES | 0 |
| MMU_MICRO_ITLB_ENTRIES | 0 |
| MPU_USE_LIMIT_FOR_REGION_RANGE | F |
| MPU_NUMBER_DATA_REGIONS | 2 |
| MPU_MINIMUM_DATA_REGION_SIZE | 0 |
| MPU_NUMBER_INSTRUCTION_REGIONS | 2 |
| MPU_MINIMUM_INSTRUCTION_REGION_SIZE | 0 |
| CUSTOM_FP_DIVISION | F |
| CUSTOM_BITSWAP | F |
| CUSTOM_ENDIAN_CONVERT | F |
| CUSTOM_INTERRUPT_VECTOR | F |

Table 8.2: Parameter values

Chapter 9

Execution Modes

| Mode | Code | Description |
|--------------------|------|---------------------|
| VM.MODE_KERNEL | 0 | Supervisor Mode |
| VM.MODE_USER | 1 | User Mode |
| VM.MODE_KERNEL_MPU | 2 | Supervisor Mode MPU |
| VM.MODE_USER_MPU | 3 | User Mode MPU |

Table 9.1: Modes implemented in this processor

Chapter 10

Exceptions

| Exception | Code |
|-------------------------------------|----------|
| NONE | 0 |
| RESET | 1 |
| HARDWARE_BREAK | 2 |
| PROCESSOR_ONLY_RESET_REQUEST | 4 |
| INTERNAL_INTERRUPT | 8 |
| EXTERNAL_NONMASKABLE_INTERRUPT | 16 |
| EXTERNAL_MASKABLE_INTERRUPT | 32 |
| SUPERVISOR_ONLY_INSTRUCTION_ADDRESS | 64 |
| FAST_TLB_MISS_INSTRUCTION | 128 |
| DOUBLE_TLB_MISS_INSTRUCTION | 256 |
| TLB_PERMISSION_VIOLATION_EXECUTE | 512 |
| MPU_REGION_VIOLATION_INSTRUCTION | 1024 |
| SUPERVISOR_ONLY_INSTRUCTION | 2048 |
| TRAP_INSTRUCTION | 4096 |
| ILLEGAL_INSTRUCTION | 8192 |
| UNIMPLEMENTED_INSTRUCTION | 16384 |
| BREAK_INSTRUCTION | 32768 |
| SUPERVISOR_ONLY_DATA_ADDRESS | 65536 |
| MISALIGNED_DATA_ADDRESS | 131072 |
| MISALIGNED_DESTINATION_ADDRESS | 262144 |
| DIVISION_ERROR | 524288 |
| FAST_TLB_MISS_DATA | 1048576 |
| DOUBLE_TLB_MISS_DATA | 2097152 |
| TLB_PERMISSION_VIOLATION_READ | 4194304 |
| TLB_PERMISSION_VIOLATION_WRITE | 8388608 |
| MPU_REGION_VIOLATION_DATA | 16777216 |

Table 10.1: Exceptions implemented by this processor

Chapter 11

Hierarchy of the model

A CPU core may be configured 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.

11.1 Level 1

This level in the model hierarchy has 3 commands.

This level in the model hierarchy has 3 register groups:

| Group name | Registers |
|---------------------|------------------|
| User | 32 |
| System | 17 |
| Integration_support | 1 |

Table 11.1: Register groups

This level in the model hierarchy has no children.

Chapter 12

Model Commands

A Processor model can implement one or more **Model Commands** available to be invoked from the simulator command line, from the OP API or from the Imperas Multiprocessor Debugger.

12.1 Level 1

12.1.1 dumpTLB

12.1.1.1 Argument description

Display the current contents of the TLB

12.1.2 isync

specify instruction address range for synchronous execution

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

Table 12.1: isync command arguments

12.1.3 itrace

enable or disable instruction tracing

| Argument | Type | Description |
|-------------------|---------|---|
| -after | Uns64 | apply after this many instructions |
| -enable | Boolean | enable instruction tracing |
| -instructioncount | Boolean | include the instruction number in each trace |
| -memory | String | show memory accesses by this instruction. Argument can be any combination of X (execute), A (load or store access) and S (system) |
| -mode | Boolean | show processor mode changes |
| -off | Boolean | disable instruction tracing |
| -on | Boolean | enable instruction tracing |
| -processorname | Boolean | Include processor name in all trace lines |

| | | |
|-----------------|---------|--|
| -registerchange | Boolean | show registers changed by this instruction |
| -registers | Boolean | show registers after each trace |

Table 12.2: itrace command arguments

Chapter 13

Registers

13.1 Level 1

13.1.1 User

Registers at level:1, group:User

| Name | Bits | Initial-Hex | RW | Description |
|------|------|-------------|----|-------------|
| zero | 32 | 0 | r- | |
| at | 32 | 0 | rw | |
| r2 | 32 | 0 | rw | |
| r3 | 32 | 0 | rw | |
| r4 | 32 | 0 | rw | |
| r5 | 32 | 0 | rw | |
| r6 | 32 | 0 | rw | |
| r7 | 32 | 0 | rw | |
| r8 | 32 | 0 | rw | |
| r9 | 32 | 0 | rw | |
| r10 | 32 | 0 | rw | |
| r11 | 32 | 0 | rw | |
| r12 | 32 | 0 | rw | |
| r13 | 32 | 0 | rw | |
| r14 | 32 | 0 | rw | |
| r15 | 32 | 0 | rw | |
| r16 | 32 | 0 | rw | |
| r17 | 32 | 0 | rw | |
| r18 | 32 | 0 | rw | |
| r19 | 32 | 0 | rw | |
| r20 | 32 | 0 | rw | |
| r21 | 32 | 0 | rw | |
| r22 | 32 | 0 | rw | |
| r23 | 32 | 0 | rw | |
| et | 32 | 0 | rw | |
| bt | 32 | 0 | rw | |
| gp | 32 | 0 | rw | |
| sp | 32 | 0 | rw | |
| fp | 32 | 0 | rw | |
| ea | 32 | 0 | rw | |
| ba | 32 | 0 | rw | |
| ra | 32 | 0 | rw | |

Table 13.1: Registers at level 1, group:User

13.1.2 System

Registers at level:1, group:System

| Name | Bits | Initial-Hex | RW | Description |
|----------|------|-------------|----|-----------------|
| PC | 32 | 0 | rw | program counter |
| status | 32 | 0 | rw | |
| estatus | 32 | 0 | rw | |
| bstatus | 32 | 0 | rw | |
| ienable | 32 | 0 | rw | |
| ipending | 32 | 0 | rw | |
| cpuid | 32 | 0 | rw | |
| ctl6 | 32 | 0 | rw | |
| except | 32 | 0 | rw | |
| pteaddr | 32 | 0 | rw | |
| tlbacc | 32 | 0 | rw | |
| tlbmisc | 32 | 0 | rw | |
| eccinj | 32 | 0 | rw | |
| badaddr | 32 | 0 | rw | |
| config | 32 | 0 | rw | |
| mpubase | 32 | 0 | rw | |
| mpuacc | 32 | 0 | rw | |

Table 13.2: Registers at level 1, group:System

13.1.3 Integration_support

Registers at level:1, group:Integration_support

| Name | Bits | Initial-Hex | RW | Description |
|------|------|-------------|----|-------------|
| stop | 32 | 0 | rw | |

Table 13.3: Registers at level 1, group:Integration_support