



OVP Guide to Using Processor Models

Model specific information for `xilinx_microblaze_V8_10`

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



| | |
|----------|---|
| Author | Imperas Software Limited |
| Version | 20191106.0 |
| Filename | OVP_Model_Specific_Information_microblaze_V8_10.pdf |
| Created | 7 November 2019 |
| Status | OVP Standard Release |

Copyright Notice

Copyright (c) 2019 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 Features | 1 |
| 1.4 Limitations | 2 |
| 1.5 ConfigurationFeatures | 2 |
| 1.6 Verification | 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 | 8 |
| 8 Formal Parameters | 10 |
| 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 | 17 |
| 13.1 | Level 1 | 17 |
| 13.1.1 | User | 17 |
| 13.1.2 | System | 18 |
| 13.1.3 | Integration_support | 18 |

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

Microblaze Processor Model

1.2 Licensing

Apache 2.0 Open Source License

1.3 Features

Instruction Set: This model fully implements the instruction set upto and including V10.

Privileged Instructions: Implemented

Virtual-Memory Management: Implemented

Reset, Interrupts, Exceptions and Break: Implemented

Floating Point Unit: Implemented

Stream Link Interface: Implemented

1.4 Limitations

GDB: There is a known issue in the microblaze gdb, a lockup can occur when disassembling from address 0

1.5 ConfigurationFeatures

Barrel Shifter.

Hardware Divider.

Machine Status Set/Clear Instructions.

Hardware Exceptions.

Pattern Compare Instructions.

Floating Point Unit (FPU).

Disable Hardware Multiplier.

Processor Version Register (PVR).

Hardware Multiplier 64-bit Result.

Floating Point Conversion and Square Root Instructions.

Memory Management Unit.

Extended Stream Instructions .

Count Leading Zeros Instruction.

Memory Barrier Instruction.

Allow Stream Instructions in User Mode.

1.6 Verification

Models have been validated correct by running through extensive tests using test suites and technology provided by Xilinx

Chapter 2

Configuration

2.1 Location

This model's VLVN is `xilinx.ovpworld.org/processor/microblaze/1.0`.

The model source is usually at:

`$IMPERAS_HOME/ImperasLib/source/xilinx.ovpworld.org/processor/microblaze/1.0`

The model binary is usually at:

`$IMPERAS_HOME/lib/$IMPERAS_ARCH/ImperasLib/xilinx.ovpworld.org/processor/microblaze/1.0`

2.2 GDB Path

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

2.3 Semi-Host Library

The default semi-host library file is `xilinx.ovpworld.org/semihosting/microblazeNewlib/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

ELF codes supported by this model are: `0xbd` and `0xbaab`.

Chapter 3

All Variants in this model

This model has these variants

| Variant | Description |
|----------------|------------------------------|
| V7.00 | |
| V7.10 | |
| V7.20 | |
| V7.30 | |
| V8.00 | |
| V8.10 | (described in this document) |
| V8.20 | |
| V9.50 | |
| V10.00 | |
| ISA | |

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 |
|-------------|-------|----------|-------------|
| Interrupt | input | optional | |
| Reset | input | optional | |
| MB_Reset | input | optional | |
| Ext_BRK | input | optional | |
| Ext_NM_BRK | input | optional | |

Table 6.1: Net Ports

Chapter 7

FIFO Ports

This model has these FIFO ports.

| Name | Type | Bit width | Connect? | Description |
|--------|-------|-----------|----------|---|
| SFSL0 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL1 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL2 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL3 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL4 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL5 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL6 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL7 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL8 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL9 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL10 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL11 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL12 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL13 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |
| SFSL14 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C.FSL_LINKS |

| | | | | |
|--------|--------|----|----------|--|
| SFSL15 | input | 64 | optional | FSL Fifo Input port - Controlled by Parameter C_FSL_LINKS |
| MFSL0 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL1 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL2 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL3 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL4 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL5 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL6 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL7 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL8 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL9 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL10 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL11 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL12 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL13 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL14 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |
| MFSL15 | output | 64 | optional | FSL Fifo Output port - Controlled by Parameter C_FSL_LINKS |

Table 7.1: FIFO Ports

Chapter 8

Formal Parameters

| Name | Type | Description |
|-----------------------------|-------------|---|
| variant | Enumeration | Selects variant (either a generic ISA or a specific model) |
| verbose | Boolean | Specify verbose output messages |
| endian | Endian | Model endian |
| C_FAMILY | Uns32 | Target Family |
| C_AREA_OPTIMIZED | Uns32 | Select implementation to optimize area with lower instruction throughput |
| C_INTERCONNECT | Uns32 | Select interconnect 1 = PLBv46, 2 = AXI4 |
| C_ENDIANNES | Uns32 | Select endianness 0 = Big endian, 1 = Little endian |
| C_FAULT_TOLERANT | Uns32 | Implement fault tolerance |
| C_ECC_USE_CE_EXCEPTION | Uns32 | Generate Bus Error Exceptions for correctable errors |
| C_PVR | Uns32 | Processor version register mode selection |
| C_PVR_USER1 | Uns32 | Processor version register USER1 constant |
| C_PVR_USER2 | Uns32 | Processor version register USER2 constant |
| C_RESET_MSR | Enumeration | Reset value for MSR register (0x00, 0x20, 0x80 or 0xa0) |
| C_BASE_VECTORS | Uns32 | Location of Microblaze Vectors |
| C_D_PLB | Uns32 | Data side PLB interface |
| C_D_AXI | Uns32 | Data side AXI interface |
| C_D_LMB | Boolean | Data side LMB interface |
| C_I_PLB | Uns32 | Instruction side PLB interface |
| C_I_AXI | Uns32 | Instruction side AXI interface |
| C_I_LMB | Boolean | Instruction side LMB interface |
| C_M_AXI_DP_EXCLUSIVE_ACCESS | Uns32 | Data Peripheral AXI Interface uses AXI4 protocol, with support for exclusive access |
| C_M_AXI_DC_EXCLUSIVE_ACCESS | Uns32 | Data Cache AXI Interface uses AXI4 protocol, with support for exclusive access |
| C_USE_BARREL | Boolean | Include barrel shifter |
| C_USE_DIV | Boolean | Include hardware divider |
| C_USE_HW_MUL | Uns32 | Include hardware integer multiplier |
| C_USE_FPU | Uns32 | Include hardware floating integer point unit |
| C_USE_MSR_INSTR | Boolean | Enable use of instructions: integer MSRSET and MSRCLR |
| C_USE_PCMP_INSTR | Boolean | Enable use of instructions: integer CLZ, PCMPBF, PCMPEQ, and PCMPNE |
| C_USE_REORDER_INSTR | Uns32 | Enable use of instructions: LBUR, LHUR, LWR, SBR. SHR, SWR, SWAPB, SWAPH |
| C_UNALIGNED_EXCEPTIONS | Boolean | Enable exception handling for unaligned data accesses |
| C_ILL_OPCODE_EXCEPTION | Boolean | Enable exception handling for illegal op-code |
| C_IPLB_BUS_EXCEPTION | Uns32 | Enable exception handling for IPLB bus error |

| | | |
|---------------------------|-------------|---|
| C_DPLB_BUS_EXCEPTION | Uns32 | Enable exception handling for DPLB bus error |
| C_M_AXI_I_BUS_EXCEPTION | Uns32 | Enable exception handling for M_AXI_I bus error |
| C_M_AXI_D_BUS_EXCEPTION | Uns32 | Enable exception handling for M_AXI_D bus error |
| C_DIV_ZERO_EXCEPTION | Boolean | Enable exception handling for division by zero or division overflow |
| C_FPU_EXCEPTION | Boolean | Enable exception handling for hardware floating point unit exceptions |
| C_OPCODE_0x0_ILLEGAL | Boolean | Detect opcode 0x0 as an illegal instruction |
| C_FSL_EXCEPTION | Boolean | Enable exception handling for Stream Links |
| C_USE_STACK_PROTECTION | Uns32 | Generate exception for stack overflow or stack underflow |
| C_DEBUG_ENABLED | Boolean | MDM Debug interface |
| C_NUMBER_OF_PC_BRK | Uns32 | Number of hardware breakpoints |
| C_NUMBER_OF_RD_ADDR_BRK | Uns32 | Number of read address watchpoints |
| C_NUMBER_OF_WR_ADDR_BRK | Uns32 | Number of write address watchpoints |
| C_INTERRUPT_IS_EDGE | Uns32 | Level/Edge Interrupt |
| C_EDGE_IS_POSITIVE | Uns32 | Negative/Positive Edge integer Interrupt |
| C_FSL_LINKS | Uns32 | Number of stream interfaces (FSL or AXI) |
| C_USE_EXTENDED_FSL_INSTR | Boolean | Enable use of extended integer stream instructions |
| C_ICACHE_BASEADDR | Uns32 | Instruction cache base address |
| C_ICACHE_HIGHADDR | Uns32 | Instruction cache high address |
| C_USE_ICACHE | Uns32 | Instruction cache |
| C_ALLOW_ICACHE_WR | Uns32 | Instruction cache write enable |
| C_ICACHE_LINE_LEN | Enumeration | Instruction cache line length (4 or 8) |
| C_ICACHE_ALWAYS_USED | Uns32 | Instruction cache CacheLink used for all memory accesses |
| C_ICACHE_INTERFACE | Enumeration | Instruction cache CacheLink interface protocol (IXCL or IXCL2) |
| C_ICACHE_FORCE_TAG_LUTRAM | Uns32 | Instruction cache tag always implemented with distributed RAM |
| C_ICACHE_STREAMS | Uns32 | Instruction cache streams |
| C_ICACHE_VICTIMS | Enumeration | Instruction cache victims (0, 2, 4 or 8) |
| C_ICACHE_DATA_WIDTH | Uns32 | Instruction cache data width, 0 = 32 bits, 1 = Full cache line, 2 = 512 bits |
| C_ADDR_TAG_BITS | Uns32 | Instruction cache address tags |
| C_CACHE_BYTE_SIZE | Enumeration | Instruction cache size (64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768 or 65536) |
| C_ICACHE_USE_FSL | Uns32 | Cache over CacheLink instead of peripheral bus for instructions |
| C_DCACHE_BASEADDR | Uns32 | Data cache base address |
| C_DCACHE_HIGHADDR | Uns32 | Data cache high address |
| C_USE_DCACHE | Uns32 | Data cache |
| C_ALLOW_DCACHE_WR | Uns32 | Data cache write enable |
| C_DCACHE_LINE_LEN | Enumeration | Data cache line length (4 or 8) |
| C_DCACHE_ALWAYS_USED | Uns32 | Data cache CacheLink used for all memory accesses |
| C_DCACHE_INTERFACE | Enumeration | Data cache CacheLink interface protocol (DXCL or DXCL2) |
| C_DCACHE_FORCE_TAG_LUTRAM | Uns32 | Data cache tag always implemented with distributed RAM |
| C_DCACHE_USE_WRITEBACK | Uns32 | Data cache write-back storage policy used |
| C_DCACHE_VICTIMS | Enumeration | Data cache victims (0, 2, 4 or 8) |
| C_DCACHE_DATA_WIDTH | Uns32 | Data cache data width, 0 = 32 bits, 1 = Full cache line, 2 = 512 bits |
| C_DCACHE_ADDR_TAG | Uns32 | Data cache address tags |
| C_DCACHE_BYTE_SIZE | Enumeration | Data cache size (64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768 or 65536) |

| | | |
|----------------------------|-------|---|
| C.DCACHE_USE_FSL | Uns32 | Cache over CacheLink instead of peripheral bus for data |
| C.USE_MMU | Uns32 | 0 = None, 1 = Usermode, 2 = Protection, 3 = Virtual |
| C.MMU_DTLB_SIZE | Uns32 | Data shadow Translation Look-Aside Buffer size 1,2,4,8 |
| C.MMU_ITLB_SIZE | Uns32 | Instruction shadow Translation Look-Aside Buffer size 1,2,4,8 |
| C.MMU_TLB_ACCESS | Uns32 | Access to memory management special registers: 0 = Minimal, 1 = Read, 2 = Write, 3 = Full |
| C.MMU_ZONES | Uns32 | Number of memory protection zones |
| C.MMU_PRIVILEGED_INSTR | Uns32 | Privileged instructions 0 = Full protection, 1 = Allow stream instrs |
| C.USE_INTERRUPT | Uns32 | Enable interrupt handling |
| C.USE_EXT_BRK | Uns32 | Enable external break handling |
| C.USE_EXT_NM_BRK | Uns32 | Enable external non-maskable break handling |
| C.USE_BRANCH_TARGET_CACHE | Uns32 | Enable Branch Target Cache |
| C.BRANCH_TARGET_CACHE_SIZE | Uns32 | Branch Target Cache size: 0 = Default, 1 = 8 entries, 2 = 16 entries, 3 = 32 entries, 4 = 64 entries, 5 = 512 entries, 6 = 1024 entries, 7 = 2048 entries |
| C.STREAM_INTERCONNECT | Uns32 | Select AXI4-Stream integer interconnect |

Table 8.1: Parameters

Chapter 9

Execution Modes

| Mode | Code | Description |
|--------------|------|-------------------------|
| REAL | 0 | Real mode |
| VIRTUAL_PRIV | 1 | Virtual privileged mode |
| VIRTUAL_USER | 2 | Virtual user mode |

Table 9.1: Modes implemented in this processor

Chapter 10

Exceptions

| Exception | Code |
|--------------------------------------|-------------|
| STREAM_EXCEPTION | 0 |
| UNALIGNED_DATA_ACCESS | 1 |
| ILLEGAL_OPCODE_EXCEPTION | 2 |
| INSTRUCTION_BUS_ERROR_EXCEPTION | 3 |
| DATA_BUS_ERROR_EXCEPTION | 4 |
| DIVIDE_EXCEPTION | 5 |
| FLOATING_POINT_UNIT_EXCEPTION | 6 |
| PRIVILEGED_INSTRUCTION_EXCEPTION | 7 |
| STACK_PROTECTION_VIOLATION_EXCEPTION | 8 |
| DATA_STORAGE_EXCEPTION | 9 |
| INSTRUCTION_STORAGE_EXCEPTION | 10 |
| DATA_TLB_MISS_EXCEPTION | 11 |
| INSTRUCTION_TLB_MISS_EXCEPTION | 12 |
| RESET | 13 |
| INTERRUPT | 14 |

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 | 25 |
| 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 |
| -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 |

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 |
|------|------|-------------|----|-------------|
| R0 | 32 | 0 | r- | |
| R1 | 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 | |
| R24 | 32 | 0 | rw | |
| R25 | 32 | 0 | rw | |
| R26 | 32 | 0 | rw | |
| R27 | 32 | 0 | rw | |
| R28 | 32 | 0 | rw | |
| R29 | 32 | 0 | rw | |
| R30 | 32 | 0 | rw | |
| R31 | 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 |
|-----------|------|-------------|----|-----------------|
| SPR_PC | 32 | 0 | rw | program counter |
| SPR_MSR | 32 | 0 | rw | |
| SPR_EAR | 32 | 0 | rw | |
| SPR_ESR | 32 | 0 | rw | |
| SPR_FSR | 32 | 0 | rw | |
| SPR_BTR | 32 | 0 | rw | |
| SPR_PVR0 | 32 | 10001500 | r- | |
| SPR_PVR1 | 32 | 0 | r- | |
| SPR_PVR2 | 32 | 54831000 | r- | |
| SPR_PVR3 | 32 | 2000000 | r- | |
| SPR_PVR4 | 32 | 45000000 | r- | |
| SPR_PVR5 | 32 | 47000000 | r- | |
| SPR_PVR6 | 32 | 0 | r- | |
| SPR_PVR7 | 32 | 3fffffff | r- | |
| SPR_PVR8 | 32 | 0 | r- | |
| SPR_PVR9 | 32 | 3fffffff | r- | |
| SPR_PVR10 | 32 | 0 | r- | |
| SPR_PVR11 | 32 | ae000000 | r- | |
| SPR_EDR | 32 | 0 | rw | |
| SPR_PID | 32 | 0 | rw | |
| SPR_ZPR | 32 | 0 | rw | |
| SPR_TLBX | 32 | 0 | rw | |
| SPR_TLBSX | 32 | 0 | rw | |
| SPR_TLBLO | 32 | 0 | rw | |
| SPR_TLBHI | 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