OVP Guide to Using Processor Models

Model Specific Information for variant renesas_v850_V850E1F

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

Author	Imperas Software Limited
Version	0.4
Filename	OVP_Model_Specific_Information_v850_V850E1F.pdf
Created	25 August 2015

Copyright Notice

Right to Copy Documentation

Destination Control Statement

Disclaimer

Table of Contents

1.0 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

V850 Family Processor Model.

1.2 Licensing

Open Source Apache 2.0

1.3 Limitations

The following Debug Registers are non-functional DIR, BPC0, BPC1, ASID BPAV0, BPAV1, BPAM0, BPAM1 BPDV0, BPDV1, BPDM0, BPDM1

1.4 Verification

Models have been extensively tested by Imperas, In addition Verification suites have been supplied by Renesas for CORE validation

1.5 Features

All v850e1 single precision FPU Instructions are supported.

All v850e1 Instructions are supported.

All Program and System Registers are supported.

2.0 Configuration

2.1 Location

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

renesas.ovpworld.org/processor/v850/1.0

2.2 GDB Path

The default GDB for this model is found at:

$IMPERAS_HOME/lib/$IMPERAS_ARCH/gdb/v850-elf-gdb

2.3 Semi-Host Library

The default semi-host library file is found in the VLNV tree at :
renesas.ovpworld.org/semihosting/v850Newlib/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

ELF codes supported by this model are: , 0x57, 0x24, 0x70f1, 0x70ff and 0x747b.

3.0 Other Variants in this Model

Table 1.

Variant
V850
V850E1
V850E1F
V850ES
V850E2
V850E2M
V850E2R

4.0 Bus Ports

Table 2.

Type	Name	Bits
master (initiator)	INSTRUCTION	32
master (initiator)	DATA	32

5.0 Net Ports

Table 3.

Name	Type	Description
intp	input	Interrupt Port
nmi0	input	Non-Maskable Interrupt Port
nmi1	input	Non-Maskable Interrupt Port

nmi2	input	Non-Maskable Interrupt Port
reset	input	Reset Port
mireti	output	Return from Interrupt Port
intack	output	Interrupt Acknowledge Port

6.0 FIFO Ports

No FIFO Ports in this model.

7.0 Parameters

Table 4.

Name	Type	Description
verbose	Boolean	Specify verbose output messages
GDBSIMMODE	Boolean	GDB Simulator Compatibility Mode

8.0 Execution Modes

No execution modes.

9.0 Exceptions

Table 5.

Name	Code	Description
reset	0	Reset Signal Exception
nmi0	16	Non Maskable Interrupt(0) Exception
nmi1	32	Non Maskable Interrupt(1) Exception
nmi2	48	Non Maskable Interrupt(2) Exception
intp	65535	Maskable Interrupt Exception - Vector value = (0x0000ffff AND intp)
trap0	64	TRAP0 Exception
trap1	80	TRAP1 Exception
ilgop	96	Illegal OPCODE Exception

10.0 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 2 register groups:

Table 6.

Group name	Registers
User	32
System	34

This level in the model hierarchy has no children.

11.0 Model Commands

11.1 Level 1:

Table 7.

Name	Arguments
isync	specify instruction address range for synchronous execution
itrace	enable or disable instruction tracing

12.0 Registers

12.1 Level 1:

12.1.1 User

Table 8.

Name	Bits	Initial value (Hex)		Description
R0	32	0	r-	Zero Register
R1	32	0	rw	Assembler-reserved register
R2	32	0	rw	Address/data variable register (when the real-time OS to be used is not using r2)
R3	32	0	rw	Stack pointer (SP)
R4	32	0	rw	Global pointer (GP)
R5	32	0	rw	Test pointer (TP)
R6	32	0	rw	Address/data variable registers
R7	32	0	rw	Address/data variable registers
R8	32	0	rw	Address/data variable registers
R9	32	0	rw	Address/data variable registers
R10	32	0	rw	Address/data variable registers
R11	32	0	rw	Address/data variable registers
R12	32	0	rw	Address/data variable registers
R13	32	0	rw	Address/data variable registers
R14	32	0	rw	Address/data variable registers
R15	32	0	rw	Address/data variable registers
R16	32	0	rw	Address/data variable registers
R17	32	0	rw	Address/data variable registers
R18	32	0	rw	Address/data variable registers
R19	32	0	rw	Address/data variable registers
R20	32	0	rw	Address/data variable registers
R21	32	0	rw	Address/data variable registers
R22	32	0	rw	Address/data variable registers

R23	32	0	rw	Address/data variable registers
R24	32	0	rw	Address/data variable registers
R25	32	0	rw	Address/data variable registers
R26	32	0	rw	Address/data variable registers
R27	32	0	rw	Address/data variable registers
R28	32	0	rw	Address/data variable registers
R29	32	0	rw	Address/data variable registers
R30	32	0	rw	Element pointer (EP)
R31	32	0	rw	Link pointer (LP)

12.1.2 System

Table 9.

Name	Bits	Initial value (Hex)		Description
EIPC	32	0	r-	Interrupt status-saving register PC
EIPSW	32	0	r-	Interrupt status-saving register PSW
FEPC	32	0	r-	NMI status-saving register PC
FEPSW	32	0	r-	NMI status-saving register PSW
ECR	32	0	r-	Exception cause register
PSW	32	20	r-	Program status word
SR6	32	0	r-	_UNIMPLEMENTED_
SR7	32	0	r-	_UNIMPLEMENTED_
SR8	32	0	r-	_UNIMPLEMENTED_
SR9	32	0	r-	_UNIMPLEMENTED_
SR10	32	0	r-	_UNIMPLEMENTED_
SR11	32	0	r-	_UNIMPLEMENTED_
SR12	32	0	r-	_UNIMPLEMENTED_
SR13	32	0	r-	_UNIMPLEMENTED_
SR14	32	0	r-	_UNIMPLEMENTED_
SR15	32	0	r-	_UNIMPLEMENTED_
CTPC	32	0	r-	CALLT status-saving register PC
CTPSW	32	0	r-	CALLT status-saving register PSW
DBPC	32	0	r-	Exception/Debug trap status-saving register PC
DBPSW	32	0	r-	Exception/Debug trap status-saving register PSW
CTBP	32	0	r-	CALLT base pointer
DIR	32	0	r-	Debug Interface register
SR22	32	0	r-	_UNIMPLEMENTED_
SR23	32	0	r-	_UNIMPLEMENTED_
SR24	32	0	r-	_UNIMPLEMENTED_
SR25	32	0	r-	_UNIMPLEMENTED_
SR26	32	0	r-	_UNIMPLEMENTED_
SR27	32	0	r-	_UNIMPLEMENTED_

SR28	32	0	r-	_UNIMPLEMENTED_
SR29	32	0	r-	_UNIMPLEMENTED_
SR30	32	0	r-	_UNIMPLEMENTED_
SR31	32	0	r-	_UNIMPLEMENTED_
PC	32	0	rw	Program Counter
FP	32	0	r-	

#